IMAQ™

IMAQ Vision for G Reference Manual

NATIONAL INSTRUMENTS™

June 1999 Edition
Part Number 322366A-01

Worldwide Technical Support and Product Information

www.natinst.com

National Instruments Corporate Headquarters

11500 North Mopac Expressway   Austin, Texas 78759-3504   USA   Tel: 512 794 0100

Worldwide Offices

Australia 03 9879 5166, Austria 0662 45 79 90 0, Belgium 02 757 00 20, Brazil 011 284 5011, Canada (Ontario) 905 785 0085, Canada (Québec) 514 694 8521, China 0755 3904939, Denmark 45 76 26 00, Finland 09 725 725 11, France 01 48 14 24 24, Germany 089 741 31 30, Hong Kong 2645 3186, India 91805275406, Israel 03 6120092, Italy 02 413091, Japan 03 5472 2970, Korea 02 596 7456, Mexico (D.F.) 5 280 7625, Mexico (Monterrey) 8 357 7695, Netherlands 0348 433466, Norway 32 27 73 00, Singapore 2265886, Spain (Madrid) 91 640 0085, Spain (Barcelona) 93 582 0251, Sweden 08 587 895 00, Switzerland 056 200 51 51, Taiwan 02 2377 1200, United Kingdom 01635 523545

For further support information, see the *Technical Support Resources* appendix. To comment on the documentation, send e-mail to techpubs@natinst.com.

Important Information

Warranty

The media on which you receive National Instruments software are warranted not to fail to execute programming instructions, due to defects in materials and workmanship, for a period of 90 days from date of shipment, as evidenced by receipts or other documentation. National Instruments will, at its option, repair or replace software media that do not execute programming instructions if National Instruments receives notice of such defects during the warranty period. National Instruments does not warrant that the operation of the software shall be uninterrupted or error free.

A Return Material Authorization (RMA) number must be obtained from the factory and clearly marked on the outside of the package before any equipment will be accepted for warranty work. National Instruments will pay the shipping costs of returning to the owner parts which are covered by warranty.

National Instruments believes that the information in this document is accurate. The document has been carefully reviewed for technical accuracy. In the event that technical or typographical errors exist, National Instruments reserves the right to make changes to subsequent editions of this document without prior notice to holders of this edition. The reader should consult National Instruments if errors are suspected. In no event shall National Instruments be liable for any damages arising out of or related to this document or the information contained in it.

EXCEPT AS SPECIFIED HEREIN, NATIONAL INSTRUMENTS MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AND SPECIFICALLY DISCLAIMS ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. CUSTOMER'S RIGHT TO RECOVER DAMAGES CAUSED BY FAULT OR NEGLIGENCE ON THE PART OF NATIONAL INSTRUMENTS SHALL BE LIMITED TO THE AMOUNT THERETOFORE PAID BY THE CUSTOMER. NATIONAL INSTRUMENTS WILL NOT BE LIABLE FOR DAMAGES RESULTING FROM LOSS OF DATA, PROFITS, USE OF PRODUCTS, OR INCIDENTAL OR CONSEQUENTIAL DAMAGES, EVEN IF ADVISED OF THE POSSIBILITY THEREOF. This limitation of the liability of National Instruments will apply regardless of the form of action, whether in contract or tort, including negligence. Any action against National Instruments must be brought within one year after the cause of action accrues. National Instruments shall not be liable for any delay in performance due to causes beyond its reasonable control. The warranty provided herein does not cover damages, defects, malfunctions, or service failures caused by owner's failure to follow the National Instruments installation, operation, or maintenance instructions; owner's modification of the product; owner's abuse, misuse, or negligent acts; and power failure or surges, fire, flood, accident, actions of third parties, or other events outside reasonable control.

Trademarks

BridgeVIEW™, IMAQ™, LabVIEW™, National Instruments™, and `natinst.com`™ are trademarks of National Instruments Corporation.

Product and company names mentioned herein are trademarks or trade names of their respective companies.

WARNING REGARDING MEDICAL AND CLINICAL USE OF NATIONAL INSTRUMENTS PRODUCTS

National Instruments products are not designed with components and testing for a level of reliability suitable for use in or in connection with surgical implants or as critical components in any life support systems whose failure to perform can reasonably be expected to cause significant injury to a human. Applications of National Instruments products involving medical or clinical treatment can create a potential for death or bodily injury caused by product failure, or by errors on the part of the user or application designer. Because each end-user system is customized and differs from National Instruments testing platforms and because a user or application designer may use National Instruments products in combination with other products in a manner not evaluated or contemplated by National Instruments, the user or application designer is ultimately responsible for verifying and validating the suitability of National Instruments products whenever National Instruments products are incorporated in a system or application, including, without limitation, the appropriate design, process and safety level of such system or application.

Contents

Chapter 4
Display VIs

Chapter 5
Tool VIs

Chapter 6
Conversion VIs

Chapter 7
Operator VIs

Chapter 8
Processing VIs

Chapter 9
Filter VIs

Chapter 10
Morphology VIs

Chapter 11
Analysis VIs

Chapter 12
Geometry VIs

Chapter 13
Complex VIs

Chapter 14
Color VIs

Chapter 15
External Library Support VIs

Chapter 16
Browser VIs

Chapter 17
Inspection Tool VIs

Figures

Tables

About This Manual

The *IMAQ Vision for G Reference Manual* describes the features, functions, and operation of IMAQ Vision for G. To use this manual effectively, you should be familiar with image processing, your image capture hardware, and LabVIEW or BridgeVIEW.

Conventions Used in This Manual

The following conventions are used in this manual:

<>	Angle brackets enclose the name of a key on the keyboard—for example, <Shift>.
	This icon to the left of bold italicized text denotes a note, which alerts you to important information.
	This icon denotes a caution, which advises you of precautions to take to avoid injury, data loss, or a system crash.
bold	Bold text denotes items that you must select or click on in the software, such as menu items and dialog box options. Bold text also denotes parameter names.
italic	Italic text denotes variables, emphasis, a cross reference, or an introduction to a key concept.
`monospace`	Text in this font denotes text or characters that you should enter from the keyboard, sections of code, programming examples, and syntax examples. This font is also used for the proper names of disk drives, paths, directories, programs, subprograms, subroutines, device names, functions, operations, variables, filenames and extensions, and code excerpts.

Related Documentation

The following documents contain information that you may find helpful as you read this manual:

- *LabVIEW User Manual*
- *LabVIEW Tutorial*
- *BridgeVIEW User Manual*
- *G Programming Reference Manual*
- *IMAQ Vision User Manual*

1

VI Overview and Programming Concepts

This chapter contains an overview of IMAQ Vision programming concepts, describes the Base and Advanced versions of IMAQ Vision, and lists the VIs included in these versions. It also summarizes the icons used in this manual.

Images

An *image* is a function of the light intensity $f(x, y)$ where x and y represent the spatial coordinates of a point in an image, and f is the brightness of the point (x, y).

The pixel depth and the number of planes in an image determines the image type. IMAQ Vision supports multiple image types.

Whether you encode an image in 8 bits, 16 bits, or in a floating-point value is influenced by several factors: the nature of the image, the type of image processing you need to use, and the type of analysis you need to perform. For example, 8-bit encoding is sufficient if you plan to perform a morphology analysis (for example, surface or elongation factor). On the other hand, if your goal is to obtain a highly precise quantification of the light intensity from an image or a region of an image, you must use 16-bit or floating-point encoding.

VIs that perform frequency-domain operations can be applied to images that are Fourier transformed. Each pixel in a Fourier-transformed image, called a *complex image*, is encoded as two single-precision floating-point values.

You can also process color images. Color images can be stored in IMAQ Vision in either RGB or HSL formats. *RGB* images store color information using 8 bits each for the red, green, and blue planes. *HSL* images store color information using 8 bits each for hue, saturation, and luminance. Both formats also have an additional unused 8 bits per pixel, yielding a total of 32 bits per pixel.

All of these image types work with IMAQ Vision. However, certain operations on specific image types are not practical (for example, applying the logic operator AND to a complex image).

IMAQ Vision does not directly support other image types, particularly images encoded in files as 1-bit, 2-bit, or 4-bit images. In these cases, IMAQ Vision automatically transforms the image into an 8-bit image (the minimum for IMAQ Vision) when opening the image file. This transformation is transparent and has no effect on the use of these image types in IMAQ Vision.

In IMAQ Vision, the IMAQ Create VI defines the image type at the creation of the image object. The default image type is 8-bit (a single image plane encoded in 8 bits per pixel), which is the most prevalent image type for the scientific and industrial fields. IMAQ Vision, however, is designed to process images encoded in 10-bit, 12-bit, or 16-bit as well as in floating point and color.

IMAQ Vision VIs

This section describes the organization of the IMAQ Vision VIs. It also describes the icons used in both IMAQ Vision and the VI reference chapters of this manual.

Image-Type Icons

Table 1-1 shows the icons used to describe the image types that work with the VIs in IMAQ Vision.

Table 1-1. IMAQ Vision Image-Type Icons

Icon	Type	Description
8	0	8 bits per pixel (unsigned, standard monochrome)
16	1	16 bits per pixel (signed)
F	2	32 bits (floating point) per pixel
C	3	2 × 32 bits (floating point) per pixel (native format after a Fast Fourier Transform, or FFT)

Table 1-1. IMAQ Vision Image-Type Icons (Continued)

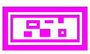
Icon	Type	Description
RGB	4	32 bits per pixel (RGB, standard color)
HSL	5	32 bits per pixel (HSL, hue, saturation, luminance)

An IMAQ Vision image has other attributes in addition to its type and size. The calibration attributes define the physical horizontal and vertical dimensions of the pixels. The ability to independently calibrate two different axes permits you to correct common defects resulting from the camera. These values are used only when performing calculations (for example, surface or perimeter) based on morphological transformations. They have no effect on either processing or operations between images.

For optimization reasons, IMAQ Vision uses borders. This border is a space that is physically reserved in the image, although it is completely transparent to you. This border is necessary when you want to perform a morphological transformation, a convolution, or particle analysis. These processes all use neighboring operations between pixels. These operations consist of applying a new value to a pixel in relation to the value of its neighbor. The advantage of the border is that all pixels can be treated the same when performing these types of operations.

IMAQ VI Error Clusters

IMAQ Vision VIs use a standard control and indicator (**error in** and **error out**) to notify you that an error has occurred.



error in (no error) is a cluster that describes the error status before the VI executes. If **error in** indicates that an error occurred before the VI was called, the VI might choose not to execute its function but just pass the error through to its **error out** cluster. If no error has occurred, the VI executes normally and sets its own error status in **error out**. Use the Error Handler VIs to look up the error code and to display the corresponding error message. Using **error in** and **error out** clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.

status is TRUE if an error occurred before the VI was called or FALSE if not. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** can be 0 or a warning code.

code is the number identifying an error or warning. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** can be 0 or a warning code. Use the Error Handler VIs to look up the meaning of this code and to display the corresponding error message.

source is a string that indicates the origin of the error, if any. Usually, **source** is the name of the VI in which the error occurred.

error out is a cluster that describes the error status after the VI executes. If an error occurred before the VI was called, **error out** is the same as **error in**. Otherwise, **error out** shows the error, if any, that occurred in the VI. Use the Error Handler VIs to look up the error code and to display the corresponding error message. Using **error in** and **error out** clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.

status is TRUE if an error occurred or FALSE if not. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** can be 0 or a warning code.

code is the number identifying an error or warning. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** can be 0 or a warning code. Use the Error Handler VIs to look up the meaning of this code and to display the corresponding error message.

source is a string that indicates the origin of the error, if any. Usually, **source** is the name of the VI in which the error occurred.

Base and Advanced Versions of IMAQ Vision

IMAQ Vision is available in both a Base version and an Advanced version. The Base version is a subset of the Advanced version.

Refer to the following tables to determine whether the VI is included in both the Base and Advanced versions. As Table 1-4 indicates, some of the IMAQ Vision VI families contain VIs that are in both the Base and Advanced versions along with others that are only in the Advanced version.

Table 1-2 lists the VI families found in both versions of IMAQ Vision.

Table 1-2. VIs in the Base and Advanced Versions

Name of VI Family	Chapter	Functionality of VIs
Management	2	Creating, listing, and disposing of image structures
Files	3	Reading and writing images to and from disk files
Display (Basics, Special, Tools, and User)	4	All aspects of image visualization Tools for creating and manipulating a region of interest
Tools (Pixels, Image, and Diverse)	5	Manipulating images Transforming the contents of an image to and from a LabVIEW array
Analysis	11	Analysis of the contents of an image
Geometry	12	3D view, rotate, shift, and symmetry
Color	14	Color image processing and analysis (histogram, threshold) Manipulating color image planes (conversions)
External Library Support	15	Access to information about image pixel organization

The Advanced version of IMAQ Vision contains all the functions found in the Base version as well as an additional set of VIs. Table 1-3 lists the VI families found only in the Advanced version.

Table 1-3. VIs in the Advanced Version Only

Name of VI Family	Chapter	Functionality of VIs
Conversion	6	Conversions from one image type into another
Operators (Arithmetic, Logic, and Comparison)	7	Addition, subtraction, multiplication, division, ratio, and modulo between two images or between one image and a constant AND, NAND, OR, NOR, XOR, XNOR, and LogDiff between two images or between one image and a constant. Clear or set as a function of a relational operator between two images or between one image and a constant Masking and the extraction of a minimum, maximum, or average can be performed between two images or between an image and a constant
Processing	8	Threshold, Label, LUT (look-up table) transformation
Filters	9	Convolutions, construction, choosing of user-defined kernels, and edge detection
Morphology	10	Morphology functions for editing binary images, including erosions, dilations, closings, openings, edge detection, thinning, thickening, hole filling, lowpass, highpass, distance mapping, rejection of particles touching the border, and filtering particles according to morphological measurements Morphology functions for modifying grayscale images, including erosions, dilations, closings, openings, and auto-median
Complex	13	Frequency processing including FFT, Inverse FFT, truncation, attenuation, addition, subtraction, multiplication, and division for complex images Functions for extraction and manipulation of planes
Color	15	Color learning and matching functions
Browser	16	Display and selection of a set of images

Table 1-3. VIs in the Advanced Version Only (Continued)

Name of VI Family	Chapter	Functionality of VIs
Inspection	17	Edge detection, caliper and other tools for gauging, measurement and inspection application
Searching and Matching	18	Tools for learning and locating reference patterns or templates in images

Some of the VI families in the Base version contain VIs that are found only in the Advanced version. Table 1-4 describes the functionality of the Advanced VIs in the Tools and Analysis VI families.

Table 1-4. Advanced Version VIs Found in Existing VI Families

Name of VI Family	Chapter	Functionality of VIs
Tools	5	Calibration, control of offset, and the ability to create a mask starting from a user-selected point and a user-defined tolerance value
Analysis	11	Simple and complex particle detection Extraction of measurement and morphological coefficients for each object in an image
Color	14	Color inspection

Manipulation of Images by IMAQ Vision

An 8-bit encoded image with a 512 × 512 resolution occupies 262,144 bytes or 256 KB of memory. IMAQ Vision is responsible for managing these image spaces.

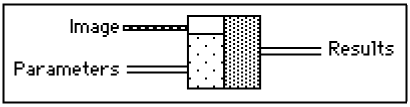
Inherent in all VIs belonging to the IMAQ Vision library is an input of one or more image structures. The IMAQ Create VI manages these structures directly. Each image requires a unique name. An image structure can contain different data or information. The image structure is dependent on the image processing and type of functions that you need to perform.

This image cluster input that is wired to each VI is a specific data type (a cluster in the *G* programming language), which results from the execution of IMAQ Create. Before it can execute, a VI must have information about which image to process and which image (the original or another) should receive the results. This image cluster provides this information when wired to a VI.

You can create multiple images by executing IMAQ Create as many times as you want, but each image you create requires a unique name. Determine the number of required images through an analysis of your intended application. The decision is based on different processing phases and your need to keep the original image (after each processing step).

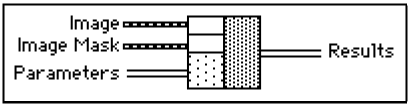
Depending on the type of function a VI performs, different combinations of input and output are possible. You can use this flexibility to decide which image to process and where to store the resulting image. If no destination image is wired, the source image is used and passed to the destination output.

The following graphics show several connection types used in IMAQ Vision.



This connection pane applies only to VIs that analyze an image and therefore do not modify either the size or contents of the image. Examples of these types of operations include particle analysis and histogram calculations.

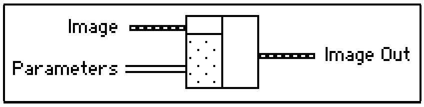
The following pane introduces an **Image Mask**.



The presence of an **Image Mask** input indicates that the processing or analysis is dependent on the contents of another image (the **Image Mask**). The only pixels in **Image** that are processed are those whose corresponding pixels in **Image Mask** are non-zero. If an **Image Mask** pixel is 0, the corresponding **Image** pixel is not changed. **Image Mask** must be an 8-bit image.

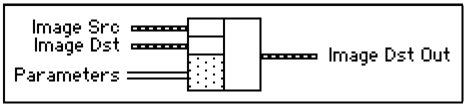
If you want to apply a processing or analysis function to the entire image, do not connect the **Image Mask** input. Connecting the same image to both inputs **Image** and **Image Mask** also gives the same effect as leaving the input **Image Mask** unconnected, except in this case the **Image** must be an 8-bit image.

The following connection pane applies to VIs performing an operation that fills an image.



Examples of this type of operation include reading a file or transforming a 2D array into an image. This type of VI can modify the size of an image.

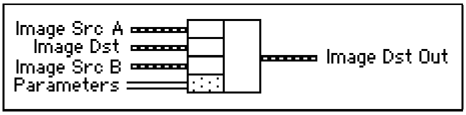
The following connection pane applies to VIs that process an image.



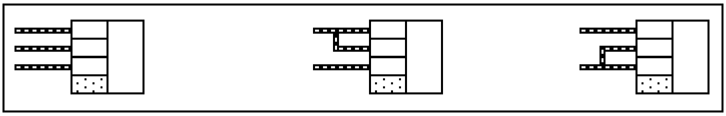
This connection is the most common type in IMAQ Vision. The **Image Src** input receives the image to process. The **Image Dst** output can receive either another image or the original, depending on your goals. If two different images are connected to the two inputs, the original **Image Src** image is not modified. As shown in the following diagrams, if the **Image Dst** and **Image Src** inputs receive the same image, or if nothing is connected to **Image Dst**, the processed image is placed into the original image, and the original image data is lost.

The **Image Dst** image is the image that receives the processing results. Depending on the functionality of the VI, this image can be either the same or a different image type as that of the source image. The VI descriptions in this manual include the type of image that can be connected to the **Image** inputs. The image connected to **Image Dst** will be resized to the source image size.

The following connection pane applies to VIs that perform arithmetic or logical operations between two images.



Two source images exist for the destination image. You can perform an operation between two images A and B and then either store the result in another image (**Dst**) or in one of the two source images, A or B. In the latter case, you can consider the original data to be unnecessary after the processing has occurred. The following combinations are possible in this pane.



In the pane on the left, the three images are all different. **Image Src A** and **Image Src B** are intact after processing and the results from this operation are stored in **Image Dst**.

In the pane in the center, **Image Src A** also is connected to the **Image Dst**, which therefore receives the results from the operation. In this operation the source data for **Image Src A** is overwritten.

In the pane on the right, **Image Src B** receives the results from the operation and its source data is overwritten.

Most operations between two images require that the images have the same size. However, arithmetic operations can work between two different types of images (for example, 8 bit and 16 bit).

Certain other data structures are frequently used in IMAQ Vision. All VIs that use coordinates (for example, line or rectangle) use an array of integers.

Rectangle

[I32]

The entity **Rectangle** is composed of four coordinates (Left/Top/Right/Bottom). A rectangle is specified by constructing an array of integers containing the following information:

Rectangle[0] = *L*, where *L* is the left-horizontal position.
Rectangle[1] = *T*, where *T* is the top-vertical position.
Rectangle[2] = *R*, where *R* is the right-horizontal position.
Rectangle[3] = *B*, where *B* is the bottom-vertical position.

An image with a resolution of 640 × 480 is composed of the points [0, 0] to [639, 479], but the rectangle takes into account the entirety of the image [0, 0, 640, 480]. The right-horizontal and the bottom-vertical positions must be 1 greater than the last specified column and line. The default coordinates for a rectangle are [0, 0, 32,767, 32,767]. If these coordinate values are shown (in the front panel of the VI), the rectangle input is not connected. In this case the entire image is taken into account when the operation is performed.

Line

[I32]

The entity **Line** is composed of four coordinates distributed in two points. Each point contains horizontal and vertical information. An array of integers must be constructed to specify a line. This includes the following information:

Line[0] = x_1, where x_1 is the horizontal starting position.
Line[1] = y_1, where y_1 is the vertical starting position.
Line[2] = x_2, where x_2 is the horizontal end-point position.
Line[3] = y_2, where y_2 is the vertical end-point.

No default vector is defined. In executing this type of VI, you must connect an array of four elements.

Array of Pixels

The entity **array of pixels** is represented as a 2D array. The first dimension in a G array is the vertical axis, and the second dimension is the horizontal axis. In memory, the pixels are stored in the order of the X axis. The array is formed as shown below:

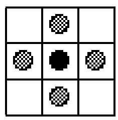
Y Dimension (**I32**)
X Dimension (**I32**)

$$\begin{bmatrix} \mathit{Array}\,[0][0] & \cdots\ \mathit{Array}\,[1][0]\ \cdots & \mathit{Array}\,[\mathit{X\ Dimension}-1][0] \\ \vdots & & \vdots \\ \mathit{Array}\,[0][\mathit{Y\ Dimension}-1] & \cdots & \mathit{Array}\,[\mathit{X\ Dimension}-1][\mathit{Y\ Dimension}-1] \end{bmatrix}$$

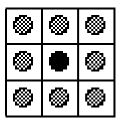
Connectivity 4/8

Specific label and particle-measurement VIs possess the input **Connectivity 4/8**. This parameter specifies how the algorithm determines whether two adjacent pixels are part of the same particle.

In Connectivity-4, only the neighboring pixels above, below, to the left, and to the right are considered to be in the same particle, as shown below.



In Connectivity-8, all touching pixels are considered to be part of the same particle, as shown below.



Structuring Element

A *structuring element* is a 2D array. It is used specifically for morphological transformations. The values contained in this array are either 0 or 1. These values dictate which pixels to take into account during processing.

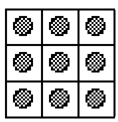
The use of a structuring element requires that the image contain a border. The application of a 3 × 3 structuring element requires a minimal border size of 1. In the same way, structuring elements of 5 × 5 and 7 × 7 require a minimal border size of 2 and 3, respectively. Structuring elements greater than these sizes require corresponding increases in the image border.

Note The default border size used when creating an image with IMAQ Vision is 3. This value enables the use of kernels up to 7 × 7 in size without any modification. If you plan to use kernels larger than 7 × 7 in your process, you need to specify a larger border when creating your image.

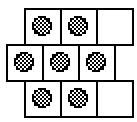
Square/Hexagon

A digital image is a 2D array of pixels arranged in a regular rectangular grid. In image processing, this grid can have two different (pixel) frames: square or hexagonal. Therefore you can choose whether the structuring element to apply during a morphological transformation has a *square* frame or *hexagonal* frame. This decision affects how the algorithm perceives the image during processing, when using those functions that use this concept of a frame. The chosen pixel frame directly affects the output from morphological measurements (for example, perimeter and surface). Notice, however, that the frame has no effect on the availability of the pixel in memory.

By default, IMAQ Vision uses a square frame. A hexagonal frame is recommended when you need to obtain highly precise results. As shown below, every other line has shifted a half pixel to the right. The hexagonal frame places the pixels in a configuration approaching a true circle. In those cases when the hexagonal frame is used, not all the structuring element values are used. For example, the hexagonal frame below ignores the unfilled pixels. All VIs that use this information have the input **Square/Hexa**.



Square 3 × 3



Hexagonal 3 × 3

The size of the structuring element directly determines the speed of the morphological transformation. Different results occur when the contents of the structuring element are changed. It is recommended that you understand morphology or learn how to use these elements before changing the standard structuring element. You can learn more about morphology by reading Chapter 7, *Morphology Analysis*, in the *IMAQ Vision User Manual*.

2
Management VIs

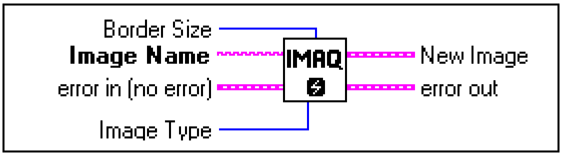
This chapter describes the functionality of the IMAQ Vision Management VIs.

IMAQ Create

Creates an image.

Note Use IMAQ Create in conjunction with the IMAQ Dispose VI to create/dispose IMAQ Vision images in LabVIEW or BridgeVIEW.

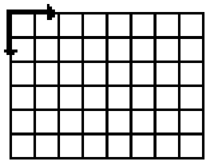
8 16 F C RGB HSL



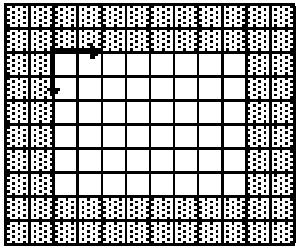
I32

Border Size determines the width, in pixels, of the border to create around an image. These pixels are used only for specific VIs. Create a border at the beginning of your application if an image is to be processed later using functions that require a border (for example, labeling and morphology). The default border value is 3. With a border of three pixels, you can use kernels up to 7 × 7 with no change. If you plan to use kernels larger than 7 × 7 in your process, you will need to specify a larger border when creating your image.

The following graphic illustrates an 8 × 6 image with a border equal to 0.



In the following 8 × 6 image, the border equals 2, allowing the use of kernels up to 5 × 5.



Note The border of an image is taken into account only for the process. It is never displayed or stored to a file.

Image Name is the name associated with the created image. Each image created must have a unique name.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Image Type specifies the image type. This input can accept the following values:

8	**8 bits**	8 bits per pixel (unsigned, standard monochrome)
16	**16 bits**	16 bits per pixel (signed)
F	**float**	32 bits (floating point) per pixel
C	**complex**	2 × 32 bits (floating point) per pixel (native format after an FFT)
RGB	**RGB**	32 bits per pixel (RGB, color)
HSL	**HSL**	32 bits per pixel (HSL, color)

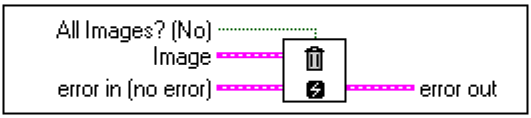
New Image is the **Image** structure that is supplied as input to all subsequent (downstream) functions used by IMAQ Vision. Multiple images can be created in a LabVIEW or BridgeVIEW application.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ Dispose

Destroys an image and frees the space it occupied in memory. This VI is required for each image created in an application to free the memory allocated to IMAQ Create. Execute IMAQ Dispose only when the image is no longer needed in your application. You can use IMAQ Dispose for each call to IMAQ Create or just once for all images created with IMAQ Create.

8 16 F C RGB HSL



All Images? (No) specifies whether to destroy a single image or all previously created images. Giving a TRUE value on input destroys all images previously created. The default is FALSE. Be sure to use this function at the end of an application to free the memory occupied by the images.

Image specifies the image to destroy.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

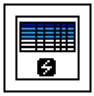
error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Note When a LabVIEW or BridgeVIEW application is aborted, the image space remains occupied.

IMAQ Status

Lists all the images created and the memory space they occupy.

8 16 F C RGB HSL



This VI cannot be used as a subVI. It must be executed from its front panel. This VI gives the total space, in bytes, occupied by the existing images. You can use this VI while writing an application to retrieve information about the images currently in memory and monitor the image creation and disposal mechanism.

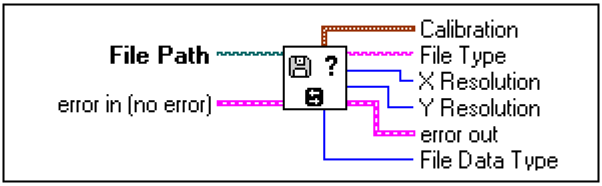
3

File VIs

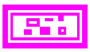
This chapter describes the VIs dedicated to read, write, and retrieve image file information. The file formats IMAQ Vision supports are BMP, TIFF, JPEG, PNG, and AIPD (internal file format).

IMAQ GetFileInfo

Obtains information regarding the contents of the file. This information is supplied for standard file formats only: BMP, TIFF, JPEG, PNG, or AIPD.



File Path is the complete pathname, including drive, directory, and filename, for the file to be examined. This path can be supplied by either the user or the VI **File Dialog** from LabVIEW or BridgeVIEW.

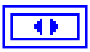


error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Calibration is a cluster containing the following elements:

X Step is the horizontal distance separating two adjacent pixels (in units specified by **Unit**).

Y Step is the vertical distance separating two adjacent pixels (in units specified by **Unit**).



Unit is the measuring unit associated with the image.

If calibration information was not written into the file, **X Step** and **Y Step** will be set to 0 and **Unit** will be set to **Undefined**.

File Type indicates the file type that is read: BMP, TIFF, JPEG, PNG, or AIPD (internal file format).

X Resolution indicates the horizontal resolution, in pixels, of the image file.

Y Resolution indicates the vertical resolution, in pixels, of the image file.

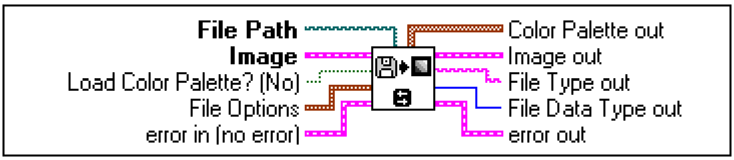
error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

File Data Type indicates the pixel type defined in the header of the file.

IMAQ ReadFile

Reads an image file. The file format can be a standard format (BMP, TIFF, JPEG, PNG, and AIPD) or a non-standard format known to the user. In all cases, the read pixels are automatically converted into the image type passed by **Image**.

8 16 F C RGB HSL



File Path is the complete pathname, including drive, directory, and filename, for the file to be loaded.

Image is the reference to the image structure to which the data from the image file is applied.

Load Color Palette? (No) determines whether to load the color table present in the file (if it exists). If loaded, this table is read and made available to the output **Color Palette**. The default is FALSE.

File Options is a cluster of user-optional values that you can use to read non-standard file formats. The file structure must be known to the user. This cluster consists of the following elements:

Read Raw File indicates whether the file to be read has a non-standard file format. If so, the remaining options in this cluster describe how to read the data.

File Data Type indicates how the image file is encoded.

Offset to Data specifies the size, in bytes, of the file header. This part of the file is not taken into account when read. The pixel values are read from the byte immediately after the offset size. The default is 0.

Use Min Max determines if the user is using a predetermined minimum and maximum. The technique to determine this minimum and maximum depends on the following input values:

0	**Don't use min max**	Minimum and maximum are dependent on the type of image. For an 8-bit image, min = 0 and max = 255.
1	**Use file values**	Pixel values from the file are scanned one time to determine the minimum and maximum, and a linear interpolation is performed before loading the image.
2	**Use optional values**	Uses the two optional values described below.

Optional Min Value is the minimum value of the pixels if **Use Min Max** is selected in mode 2 (**Use optional values**). In this case, pixels with a smaller value are altered to match the chosen minimum. The default is 0.

Optional Max Value is the maximum value of the pixels if **Use Min Max** was selected in mode 2 (**Use optional values**). In this case, pixels with a greater value are truncated to match the chosen maximum. The default is 255.

Byte Order determines if the byte weight is to be swapped (Intel or Motorola). The default is FALSE, which specifies big endian (Motorola). TRUE specifies little endian (Intel). This input is useful only if the pixels are encoded with more than 8 bits.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Color Palette out contains the RGB color table (if the file has one) read from the file when the user passes the value TRUE for the input **Load Color Palette? (No)**.

Image out is the reference to the image structure containing the data read from the image file.

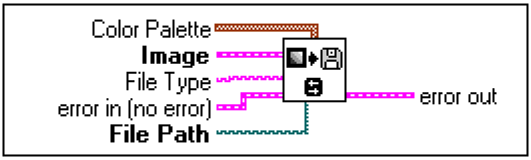
File Type out indicates the file type that is read. This string returns an identifier of the file format, which can be BMP, TIFF, JPEG, PNG, or AIPD (internal file format). **File Type** returns `xxx` if the file format is unknown.

File Data Type out indicates the pixel size defined in the header for standard image file types. **File Options** are not necessary for reading standard image files. For other types of image files, the returned values are passed from **File Options/File Data Type**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ WriteFile

Writes an image to a file.



Color Palette is used to apply colors to a monochrome image. **Color Palette** is an array of clusters constructed by the user or supplied by IMAQ GetPalette. This palette is composed of 256 elements for each of the three color planes. A specific color is the result of applying a value between 0 and 255 for each of the three color planes (red, green, and blue). If the three planes have the identical value, a gray level is obtained (0 specifies black and 255 specifies white). If the image type requires a color palette and it is not supplied, a grayscale color palette is generated and written.

Image is the reference to the image structure to be written to an image file.

File Type describes the file type to be written. The default file type is BMP. Other file types supported are AIPD, TIFF, JPEG, and PNG.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

File Path is the complete pathname, including drive, directory, and filename, of the file to be written. This path can be supplied by either the user or the VI **File Dialog** from LabVIEW or BridgeVIEW.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Table 3-1 indicates the image types that each image file format supports.

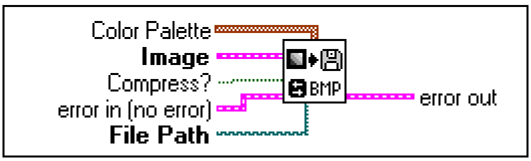
Table 3-1. Image Types Supported by Each File Format

File Type	Image Types Supported
BMP	8, RGB
JPEG	8, RGB
PNG	8, 16, RGB
TIFF	8, 16, RGB
AIPD	8, 16, F, C, RGB, HSL

IMAQ Write BMP File

Writes an image to a file in BMP format.

8 RGB



Color Palette is used to apply a color palette to an image. **Color Palette** is an array of clusters constructed by the user or supplied by IMAQ GetPalette. This palette is composed of 256 elements for each of the three color planes. A specific color is the result of applying a value between 0 and 255 for each of the three color planes (red, green, and blue). If the three planes have the identical value, a gray level is obtained (0 specifies black and 255 specifies white). If the image type requires a color palette and it is not supplied, a grayscale color palette is generated and written.

Image is the reference to the image structure to be written as an image file.

Compress? designates whether to compress the BMP file. By default, the image file is not compressed. If it is compressed, it uses the **Run Length Encoded** compression type.

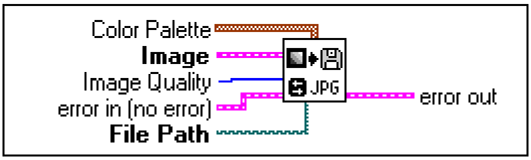
error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

File Path is the complete pathname, including drive, directory, and filename, for the image file to be written. This path can be supplied by either the user or the VI **File Dialog** from LabVIEW or BridgeVIEW.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ Write JPEG File

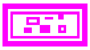
Writes an image to a file in JPEG format.



Color Palette is used to apply a color palette to an image. **Color Palette** is an array of clusters constructed by the user or supplied by IMAQ GetPalette. This palette is composed of 256 elements for each of the three color planes. A specific color is the result of applying a value between 0 and 255 for each of the three color planes (red, green, and blue). If the three planes have the identical value, a gray level is obtained (0 specifies black and 255 specifies white). If the image type requires a color palette and it is not supplied, a grayscale color palette is generated and written.

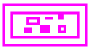
Image is the reference to the image structure to be written as an image file.

Image Quality specifies the quality of the image written to file. This is a measure of the compression. A higher quality means a lower compression factor. The allowable range is 0 to 1,000. The default quality is 750. Note that JPEG compression is lossy, which means that the lower the quality specified, the more loss takes place during the compression. Also, a small amount of loss can occur even when the quality is 1,000.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

File Path is the complete pathname, including drive, directory, and filename, for the file to be written. This path can be supplied by either the user or the VI **File Dialog** from LabVIEW or BridgeVIEW.

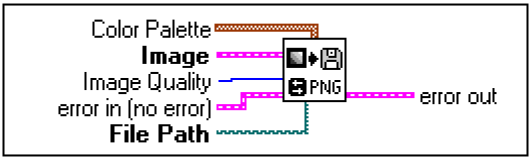


error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Note JPEG format is not recommended if you want to make precise gauging or inspection measurements on the image.

IMAQ Write PNG File

Writes an image to a file in PNG format.



Color Palette is used to apply a color palette to an image. **Color Palette** is an array of clusters constructed by the user or supplied by IMAQ GetPalette. This palette is composed of 256 elements for each of the three color planes. A specific color is the result of applying a value between 0 and 255 for each of the three color planes (red, green, and blue). If the three planes have the identical value, a gray level is obtained (0 specifies black and 255 specifies white). If the image type requires a color palette and it is not supplied, a grayscale color palette is generated and written.

Image is the reference to the image structure to be written as an image file.

Image Quality specifies the quality of the image written to the file. This is a measure of the compression. A higher quality means a lower compression factor. The allowable range is 0 to 1,000. The default quality is 750. Note that PNG compression is not lossy. Therefore, unlike JPEG images, the image does not lose information even at a low image-quality factor. Instead, the image is compressed as much as possible without losing any image information.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

File Path is the complete pathname, including drive, directory, and filename, for the file to be written. This path can be supplied by either the user or the VI **File Dialog** from LabVIEW or BridgeVIEW.

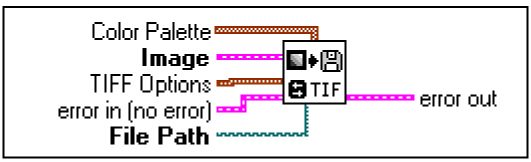
error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Note PNG format is not recommended if you want to make precise gauging or inspection measurements on the image.

IMAQ Write TIFF File

Writes an image to a file in TIFF format.

8 16 RGB



Color Palette is used to apply a color palette to an image. **Color Palette** is an array of clusters constructed by the user or supplied by IMAQ GetPalette. This palette is composed of 256 elements for each of the three color planes. A specific color is the result of applying a value between 0 and 255 for each of the three color planes (red, green, and blue). If the three planes have the identical value, a gray level is obtained (0 specifies black and 255 specifies white). If the image type requires a color palette and it is not supplied, a grayscale color palette is generated and written.

Image is the reference to the image structure to be written as an image file.

TIFF Options is a cluster of user-optional values that you can use to set TIFF options. This cluster consists of the following elements:

Rows Per Strip designates the number of rows you want to designate per strip of data. The default value of 0 specifies to write the entire data as one strip.

Photometric designates whether to interpret white or black as a zero value.

Compression Type determines the compression type. By default, no compression is used. You can choose among **JPEG**, **Run Length Encoded**, or **Zip**.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

File Path is the complete pathname, including drive, directory, and filename, for the file to be written. This path can be supplied by either the user or the VI **File Dialog** from LabVIEW or BridgeVIEW.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Note Saving 16-bit images is a nonstandard extension to the TIFF standard. Most third-party applications cannot read 16-bit TIFF files. For compatibility with most applications, write 16-bit images into PNG files.

4

Display VIs

This chapter describes the Display VIs in IMAQ Vision.

Introduction

The control of image visualization is of primary importance in an imaging application. *Image processing* and *image visualization* are distinct and separate elements that should not be confused. Image processing refers to the creation, acquisition, and analysis of images. On the other hand, image visualization refers to how the image data is presented to you and how you can work with the visualized images. A typical imaging application uses many more images than the number of display windows.

Because IMAQ Vision is designed to answer a wide variety of imaging needs, the Display VIs are arranged in four groups to accommodate users with varying skill levels. The novice user can easily access the basic Display functions, while more experienced users can create imaging applications containing sophisticated display and control capabilities.

The **Display (basics)** library contains VIs that control the display of images in image windows as well as the positioning, opening, and closing of these windows on the display screen. You can resize these image windows, remove scrollbars, and set up the display color palette. Note that these image windows are not LabVIEW or BridgeVIEW panels. They are directly managed by IMAQ Vision.

The **Display (tools)** library contains VIs for controlling image window tools. These tools include points, lines, rectangles, ovals, and freehand contours that you can use to access the image data displayed in the image window. Then you can convert this data into a region of interest or *ROI*. These VIs also regulate user interaction in the IMAQ Vision image windows as well as the events that occur in these image windows.

The **Display (user)** library enables the advanced user to create and manipulate user windows. These palettes (user windows) are user defined and can be used to create sophisticated applications.

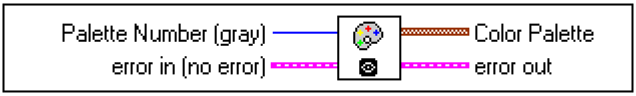
The **Display (special)** library contains advanced functionality and user-interface management.

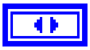
Display (Basics)

Use these functions to display images in windows and manipulate those windows.

IMAQ GetPalette

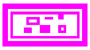
Selects a display palette. Five predefined palettes are available. To activate a color palette choose a code for **Palette Number** and connect the **Color Palette** output to the input **Color Palette** of IMAQ WindDraw.





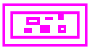
Palette Number (gray) gives you a choice of five predefined palettes. You can choose from the following values:

Gray	Grayscale is the default palette. The color tables are all identical.
Binary	Binary palette is designed especially for binary images.
Gradient	Gradient palette.
Rainbow	Rainbow palette.
Temperature	Temperature palette.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

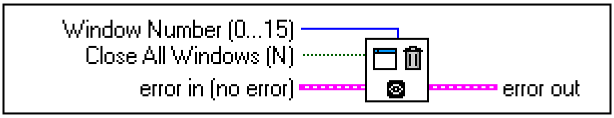
Color Palette indicates an array of clusters composed of 256 elements for each of the three color planes. A specific color is the result of applying a value between 0 and 255 for each of the three color planes (red, green, and blue). If the three planes have the identical value, a gray level is obtained (0 specifies black and 255 specifies white). This output is to be directly connected to the input **Color Palette** of IMAQ WindDraw.

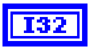


error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ WindClose

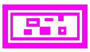
Closes an image window.



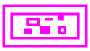


Window Number (0…15) specifies the image window to close. It is specified by a number from 0 to 15. The default value is 0.

Close All Windows (N) specifies whether all the image windows are to be closed. The default value FALSE (No) closes only the specified window. Setting this value to TRUE closes all windows simultaneously.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

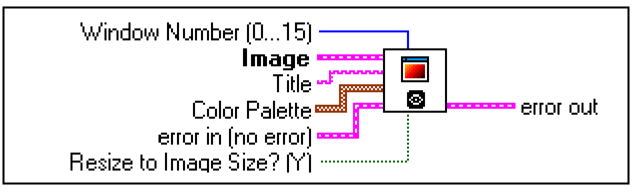


error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Note You should close all windows at the end of your application.

IMAQ WindDraw

Displays an image in an image window. The image window appears automatically when the VI is executed. Note that the default image window does not have scrollbars. You can add scrollbars by using the IMAQ WindSize VI.



Window Number (0…15) specifies the image window in which the image is displayed. As many as 16 windows can be displayed simultaneously. Each window is specified with an indicator ranging from 0 to 15. Only the specified image window is affected, and all other image windows remain the same. The default value is 0.

Image specifies the image reference for the displayed image.

Note Floating-point and 16-bit images are displayed by scaling the data to 8 bits, calculated as a function of the dynamic range from the image source. The minimum value (min) and the maximum value (max) are calculated automatically. Then the following formula is applied to each pixel:

$$Display(x, y) = (Src(x, y) - \text{min}) \times 255/(\text{max} - \text{min})$$

Title is an image window name. When a string is attached to this input, the image window automatically takes that name.

Color Palette is used to apply a color palette to an image window. **Color Palette** is an array of clusters constructed by the user or supplied by IMAQ GetPalette. This palette is composed of 256 elements for each of the three color planes. A specific color is the result of applying a value between 0 and 255 for each of the three color planes (red, green, and blue). If the three planes have the identical value, a gray level is obtained (0 specifies black and 255 specifies white). The color palette can be used only for 8-bit images.

Note For best results, set your video adapter to high color or true color.

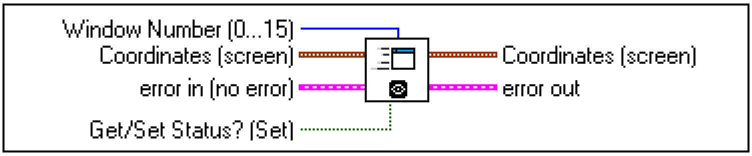
error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

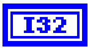
Resize to Image Size? (Y) specifies whether the user wants to resize the image window automatically to fit the image size. The default is TRUE (yes), in which case the user does not have to know the size of a source image before displaying it.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ WindMove

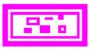
Indicates and sets the position of an image window.





Window Number (0…15) is a number from 0 to 15 that specifies the image window. The default value is 0.

Coordinates (screen) is a cluster that contains the screen coordinates, in X and Y positions, where the image window is located or where the image window will be placed. This input is necessary only when the input **Get/Set Status** is TRUE (Set).



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

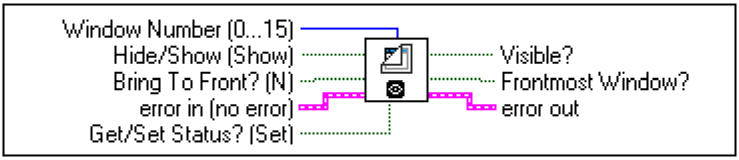
Get/Set Status? (Set) specifies whether you want to know the coordinates of an image window or change the position of an image window. The default is TRUE (Set).

Coordinates (screen) returns the present coordinates (X and Y) of an image window.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ WindShow

Shows or hides an image window.



Window Number (0…15) specifies the image window to show or hide. It is specified by a number from 0 to 15. The default value is 0.

Hide/Show (Show) specifies whether an image window is visible. This input is used only when **Get/Set Status?** is TRUE (Set).

Bring To Front? (N) determines whether to bring a window to the foreground. This input is used only when **Get/Set Status?** is TRUE (Set) and **Hide/Show** is also TRUE.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Get/Set Status? (Set) specifies whether you want to know if the image window is visible or change the visibility status of an image window. The default is TRUE (Set).

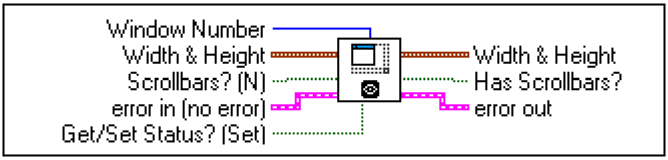
Visible? returns the present visibility status of the window. A visible image window returns TRUE.

Frontmost Window? returns TRUE if an image window is in the front.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ WindSize

Indicates and sets the size of an image window. You also can use this VI to add or remove scrollbars for image windows and test for the presence of scrollbars in an image window.



Window Number is a number from 0 to 15 that specifies the image window. The default value is 0.

Width & Height is a cluster containing two elements. When the input **Get/Set Status?** is TRUE (Set), you can specify the width and height of an image window. If the input is not connected, or if the value is (0, 0), the image window is resized automatically to the image associated with it.

Note This value includes the size of the scrollbars and title bar.

Scrollbars? (N) controls the presence of scrollbars in an image window. By default, scrollbars are not used. You can resize and move an image window whether scrollbars are present or absent.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Get/Set Status? (Set) specifies whether you want to know the position of an image window or change the position of an image window. The default value is TRUE (Set).

Width & Height returns the present width and height of an image window.

Note The returned value includes the size of the scrollbars and title bar.

Has Scrollbars? returns the present scrollbar status for an image window.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Display (Tools)

You can use the VIs in this library to perform the following functions:

- Select a region tool for defining an ROI
- Manage a standard palette of display tools
- Retrieve the events generated by a user and the associated data from an image window

With IMAQ WindToolsSelect you can select from a number of region tools including point, line, rectangle, oval, polygon, and freehand.

Using these tools you can decide which region of an image to analyze or process. Then you can transform this selected region into an image mask with IMAQ WinGetROI and IMAQ ROIToMask.

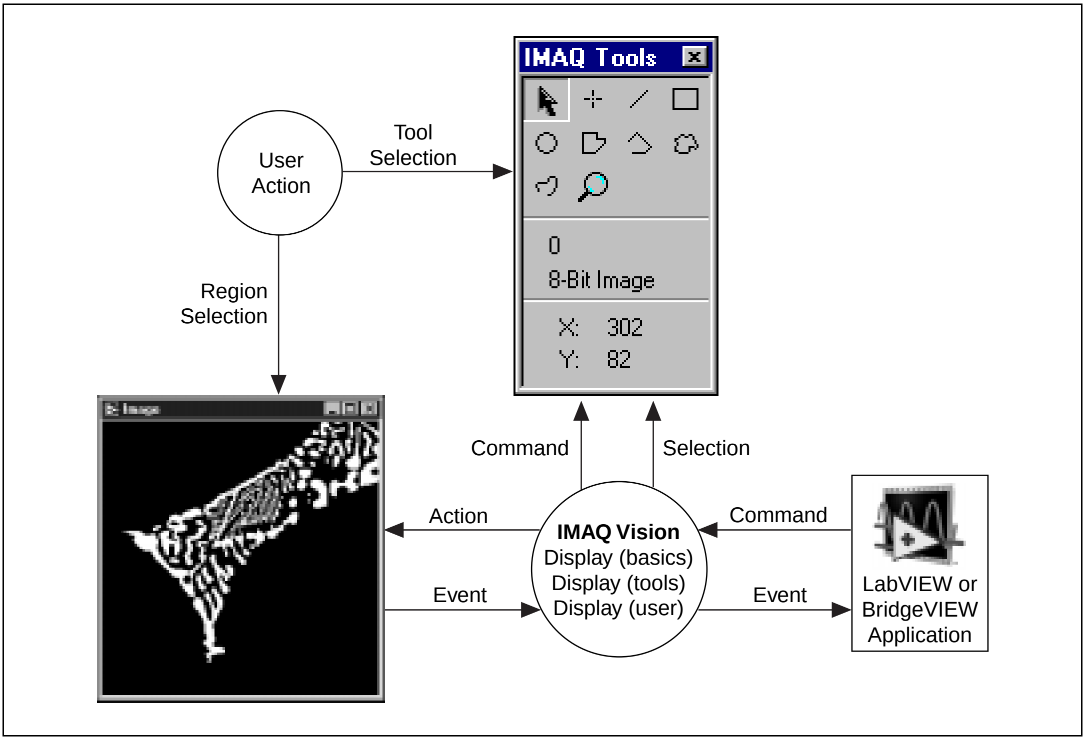
You can program a region by using the IMAQ MaskToROI and IMAQ WindSetROI VIs.

Also, you can configure a floating palette of tools from which you can choose a tool by clicking its icon. This palette displays the coordinates of the cursor within the image and the parameters of the active region.

You can also magnify (zoom) an image.

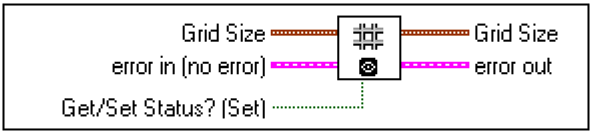
IMAQ WindLastEvent can retrieve and manage the events resulting from the interaction in an image window.

The following figure illustrates the possible interactions found among a user, IMAQ Vision, and LabVIEW or BridgeVIEW.

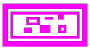


IMAQ WindGrid

Obtains or modifies the status of the grid. If enabled, the grid is invisible and drawn ROIs will be constrained to it. The grid can be used to help trace a region of interest accurately.



Grid Size is a cluster containing two elements that encode the size of the horizontal and vertical steps for the grid. The cursor is moved by steps, as defined in this VI, when tracing a region of interest. The default value is (1, 1).



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

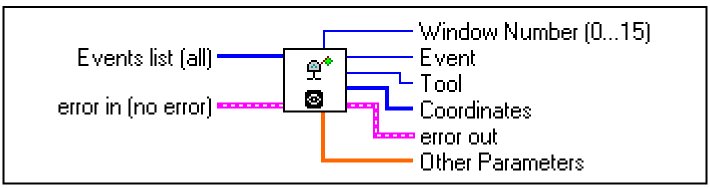
Get/Set Status? (Set) specifies whether to determine the present status or modify the step values for the grid. The default is TRUE (Set).

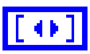
Grid Size returns the present grid-step size.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ WindLastEvent

Returns the events generated through the image windows as well as the data associated with them.





Events list (all) specifies which events to obtain. The default case returns all events generated through the image windows as well as the data associated with them. This VI enables you to specify the image window events that interest you.

Event	Description
No event	No event.
Click event	A user has clicked in an image window.
Draw event	A user has drawn in an image window.
Move event	A user has moved an image window.
Size event	A user has resized an image window.
Scroll event	A user has moved the scrollbars in an image window.
Activate event	A user has chosen (clicked once to activate) an image window.
Close event	A user has closed an image window.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Window Number (0...15) indicates the image window that is queried for events.

Event indicates the type of event.

Tool returns a code indicating the tool used.

Coordinates indicates the relative position of the event.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Other Parameters supplies information associated with an event, such as positioning and region distances.

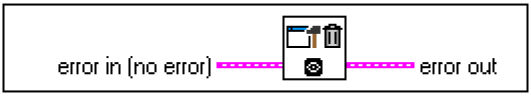
The following table describes the possible values for the **Event**, **Tool**, **Coordinates**, and **Other Parameters** indicators.

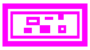
Event	Tool	Coordinates	Other Parameters
None	—	empty	empty
Click	**Cursor**	[0, 1] position (x, y) of click	[0, 1, 2] pixel value*
	Zoom	[0, 1] position of click [2, 3] position of image center	[0] zoom factor
Draw	**Line**	[0, 1] position of starting point [2, 3] position of ending point	[0, 1] width and height [2] vertical segment angle [3] segment length
	Rectangle	[0...3] bounding rectangle	[0, 1] width and height
	Oval	[0...3] bounding rectangle	[0, 1] width and height
	Polygon	[0...3] bounding rectangle	[0, 1] width and height
	Freehand	[0...3] bounding rectangle	[0, 1] width and height
Move	—	[0, 1] position of image window	empty
Size	—	[0, 1] width and height of image window	empty
Scroll	—	[0, 1] center position of image	empty
Activate	—	empty	empty
Close	—	empty	empty

* Pixel values are stored in the first element of the array for 8-bit, 16-bit, and floating-point images. The RGB and HSL values of color images are stored in the order [0, 1, 2]. The real and imaginary values of a complex image are stored in the order [0, 1].

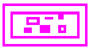
IMAQ WindToolsClose

Closes the WindTools window. This VI functions in the same way as IMAQ WindClose, which is used for closing image windows.





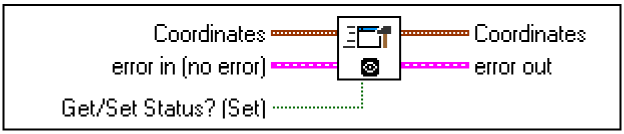
error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.



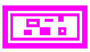
error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ WindToolsMove

Obtains or sets the position of the WindTools palette. This VI functions in the same way as IMAQ WindMove, which is used for moving image windows. You can use this VI even if the WindTools palette is not displayed.

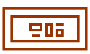


Coordinates is a cluster that contains the screen coordinates where the tools palette will be placed. This input is necessary only when **Get/Set Status? (Set)** is TRUE (Set).



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Get/Set Status? (Set) specifies whether to determine the present status or modify the status of the available region tools. The default is TRUE (Set).

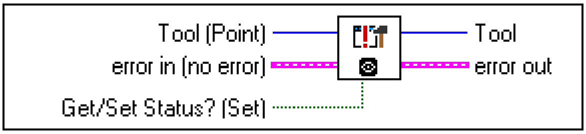


Coordinates is a cluster returning the screen coordinates (in X and Y positions) where the tools palette is located.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ WindToolsSelect

Obtains or modifies the status of the region tools.



Tool (Point) can have the following values:

Number	Icon	Tool Name	Function
0		Selection Tool	Select an ROI in the image.
1		Point	Select a pixel in the image.
2		Line	Draw a line in the image.
3		Rectangle	Draw a rectangle or square in the image.
4		Oval	Draw an oval or circle in the image.
5		Polygon	Draw a polygon in the image.
6		Free	Draw a freehand region in the image.
7	—	Unused 1	—
8		Zoom	Zoom-in or zoom-out in an image.
9	—	Unused 2	—
10		Broken Line	Draw a broken line in the image.
11		Free Hand Line	Draw a free hand line in the image.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Get/Set Status? (Set) specifies whether to determine the present status or modify the status of the available region tools. The default is TRUE (Set).

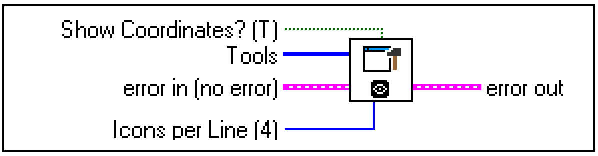
Tool returns the chosen region tool.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Note You can use this VI even if the WindTools palette is not displayed.

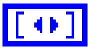
IMAQ WindToolsSetup

Configures the appearance and availability of the region tools found in the IMAQ Tools palette. With no input connections, the palette displays all nine region tools. The WindTools palette is a floating palette and is always visible.



Show Coordinates? (T) specifies whether to show the active pixel coordinates. Coordinates are shown (TRUE) by default.

Note The IMAQ Tools window is not visible unless activated by calling IMAQ WindToolsShow.



Tools specifies which icons are displayed in the IMAQ Tools window. The following tools are available:

Number	Icon	Tool Name	Function
0		Selection Tool	Select an existing ROI in the image.
1		Point	Select a pixel in the image. Action: Click on the desired position.
2		Line	Draw a line in the image. Action: Click on the initial position and click again on the final position.
3		Rectangle	Draw a rectangle (or square) in the image. Action: Click on one corner and drag to the opposite corner.
4		Oval	Draw an oval (or circle) in the image. Action: Click on the center position and drag to the desired size.
5		Polygon	Draw a polygon in the image. Action: Click to place a new vertex and double-click to complete the ROI element.
6		Free	Draw a freehand region in the image. Action: Click on the initial position, drag to the desired shape and release the mouse button to complete the shape.
7	—	Unused 1	—
8		Zoom	Zoom-in or zoom-out in an image. Action: Click on the image.
9	—	Unused 2	—
10		Broken Line	Draw a broken line in the image. Action: Click on the initial position and click again on the final position.
11		Free Hand Line	Draw a free hand line in the image. Action: Click on the initial position and click again on the final position.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Icons per Line (4) determines the number of icons per line. The subsequent lines are set as a function of the number of remaining available icons.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Note The WindTools palette automatically displays cursor information if the input **Icons per Line** is set to 4 (or higher).

Use IMAQ WindLastEvent to retrieve the draw events on the window and find the coordinates of a selected region.

You can alter the functionality of region tools by using a tool while pressing certain keyboard keys.

<Shift> while drawing constrains x and y dimensions of an ROI to be the same. This forces rectangles into squares, ellipses into circles, and line segments into horizontal or vertical segments.

<Control> <Click> adds an ROI without erasing the previous ROI elements. The previous elements are erased if you do not use <Control> when starting a new element.

<Click> <Click> (double-click) while drawing produces the last point of a polygon or broken line.

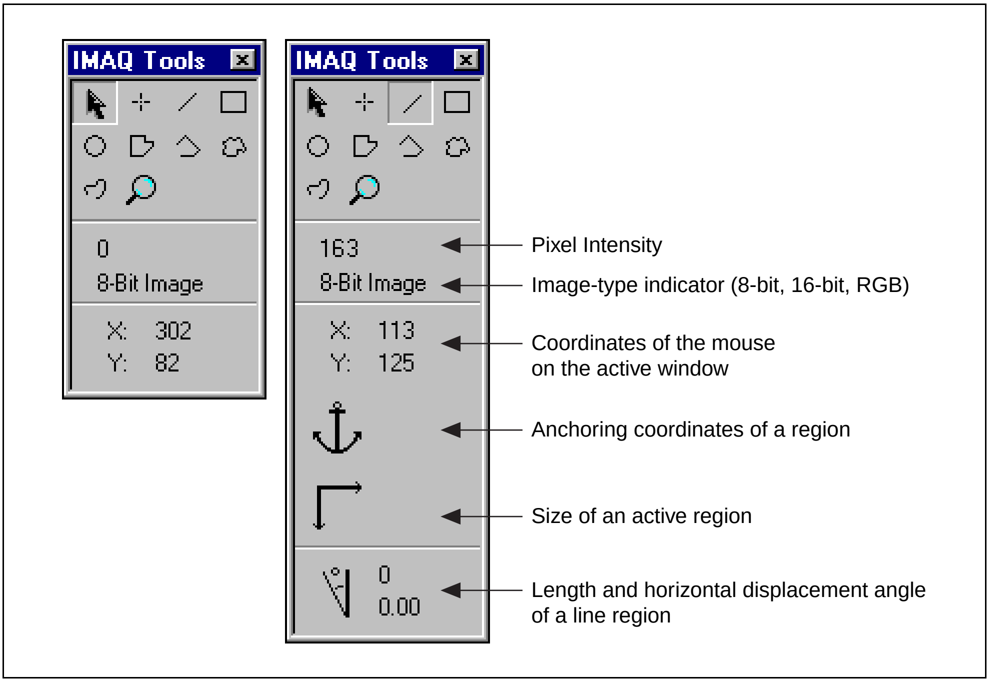
Note You can erase an ROI in an image window by selecting it and pressing <Delete>.

Use the selection tool to select any currently existing ROI by clicking its border. You can then manipulate the element in the following ways:

- To resize a rectangle or ellipse, click in a grab handle and drag it to a new location.
- To reposition a vertex in a broken line, polygon, or line, click in a grab handle and move it to a new location.
- To reposition a rectangle or ellipse, click in the interior and drag it to a new location.
- To reposition a point, click on it and drag it to a new location.

- To reposition lines, broken lines, and polygons, click on any segment and drag it to a new location.
- To reposition freehand lines and closed freehand lines, click anywhere on the line and drag it to a new location.

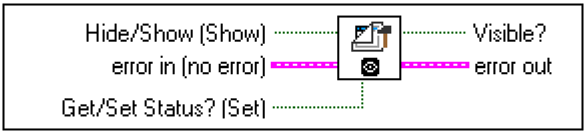
The following examples of the Tools palette have three icons per line.



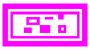
The Tools palette on the left automatically transforms to the palette on the right when you manipulate a region tool in an image window.

IMAQ WindToolsShow

Shows or hides the WindTools palette. This VI functions in the same way as IMAQ WindShow, which is used for displaying image windows.



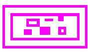
Hide/Show (Show) specifies whether the tools palette is visible. Use this input only when **Get/Set Status? (Set)** is TRUE (Set).



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Get/Set Status? (Set) specifies whether to determine the present status or modify the status of the available region tools. The default is TRUE (Set).

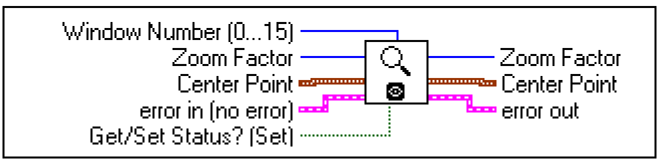
Visible? returns the present visibility status of the tools palette. A visible tools palette returns TRUE.



error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ WindZoom

Obtains or modifies the status of the zoom factor. An image can be displayed using either a positive zoom factor, which represents the image larger than actual size, or a negative zoom factor, which represents the image smaller than actual size.



Window Number (0…15) is a number from 0 to 15 that specifies the image window. The default value is 0.

Zoom Factor can have the following values: 1 to 16 and –1 to –16. The default value is 1 (image is displayed at its original size).

Center Point is a cluster containing two elements that describe the (*x*, *y*) coordinates used to center the image in the image window. Using **Center Point**, you can center an image with respect to a chosen region.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Get/Set Status? (Set) specifies whether to determine the present status or modify the **Zoom Factor** and **Center Point**. The default is TRUE (Set).

Zoom Factor returns the present zoom factor.

Center Point returns the present coordinates of the **Center Point.**

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Regions of Interest

You can use ROIs to focus your processing and analysis on part of an image. You can trace an ROI using standard contours (oval, rectangle, and so forth) or free contours (freehand). You can also perform any of the following options:

- Associate an ROI with an image window
- Extract an ROI associated with an image window
- Erase the current ROI from an image window
- Transform an ROI into an image mask
- Transform an image mask into an ROI

An image mask is an 8-bit image of the same size as or smaller than the image to process. Pixels in the mask image determine whether the corresponding pixel in the image to process needs to be processed. If a pixel in the image mask is non-zero, the corresponding pixel in the image to process is processed. If a pixel in the image mask has a value of 0, the corresponding pixel in the image to process is left unchanged.

The image mask can be limited to the bounding rectangle of the shape you want to mask. This technique is used to save memory, limiting the image mask to only the part containing information. To keep track of the location of this ROI in regard to the original image, IMAQ Vision sets a specific record of the image mask called an *offset*.

Figure 4-1 shows how an offset defines the upper-left corner coordinates (*x*, *y*) for the bounding rectangle belonging to the ROI. The default value of the offset is (0, 0).

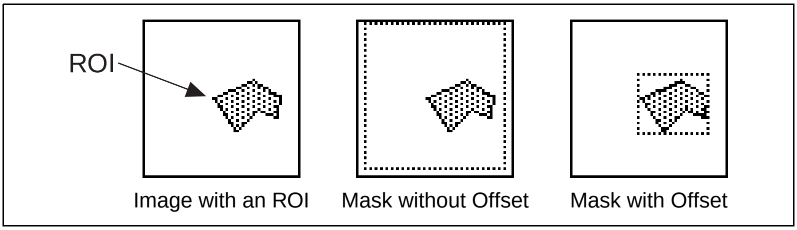


Figure 4-1. Using an Offset to Define an Image Mask

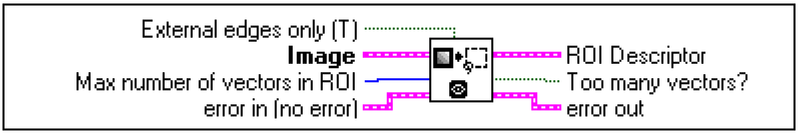
(Advanced users only) The **ROI Descriptor** cluster contains the following two elements:

- **Bounding rectangle** for an ROI.
- **Regions list**. This is an array of clusters containing the following elements:
  - **contour identifier**, where 0 specifies an exterior contour and 1 specifies an interior contour
  - **contour type** (point, line, rectangle, oval, freehand, and so on)
  - **list of points** (*x*, *y*) describing the contour

IMAQ MaskToROI

Transforms an image mask into a region of interest descriptor.

8



External edges only (T) specifies whether only the external edges are transformed. The default is TRUE.

Image is the image containing the image mask that is transformed into a region of interest. This image must be an 8-bit image.

Max number of vectors in ROI is the limit of points that define the contour of a region of interest. This value is 2500 by default.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

ROI Descriptor returns the descriptor for a region of interest.

Too many vectors? returns TRUE if the number of vectors needed to represent the ROI exceeds the value specified by **Max number of vectors in ROI**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ ROIToMask

Transforms a region of interest into a mask.

8

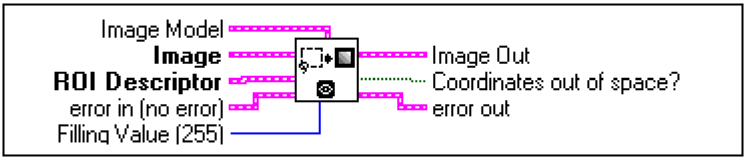


Image Model serves as a template for the destination image where the mask is placed. This can be any image type that IMAQ Vision supports. **Image** takes the characteristics of **Image Model** (size and location of ROI) when **Image Model** is connected. However, the connection of **Image Model** is optional. When no **Image Model** is connected, the size of the image mask generated is limited to the bounding rectangle of the **ROI Descriptor**, which reduces the amount of memory used. The offset of the image mask generated is set to reflect the real position of the ROI.

Image is the destination image where the mask is created. This image must be an 8-bit image type.

ROI Descriptor is the descriptor that defines the region of interest.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

I32

Filling Value (255) is the pixel value of the mask. All pixels inside the region of interest take this value. The default value is 255.

Image Out is the reference to the destination (output) image.

TF

Coordinates out of space? returns TRUE if any ROI data is found outside the space associated with the **Image Model**.

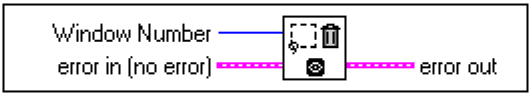
error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

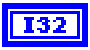
You can use this VI in two ways. The simplest technique is to connect the input **Image Model**. In this case you can use the source image, in which the image ROI was drawn, as a template for the final destination image by connecting it to **Image Model**. The output image (**Image Out**) automatically acquires the size of the image and location of the ROI as found in the original source image.

However, you do not have to connect an **Image Model**. In this case the ROI requires an offset that is determined automatically from the upper-left corner of the bounding rectangle described by the ROI. These offset values are automatically set to the image mask.

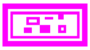
IMAQ WindEraseROI

Erases the active region of interest associated with an image window.

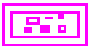




Window Number (0…15) is a number from 0 to 15 that specifies the image window. The default value is 0.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

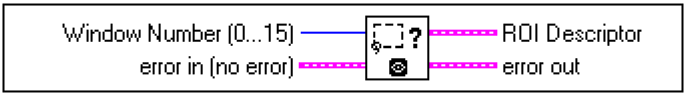


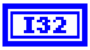
error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Note You can erase an ROI in an image window by selecting it and pressing <Delete>.

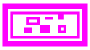
IMAQ WindGetROI

Returns the descriptor for an ROI.

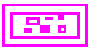




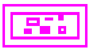
Window Number (0…15) is a number from 0 to 15 that specifies the image window. The default value is 0.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.



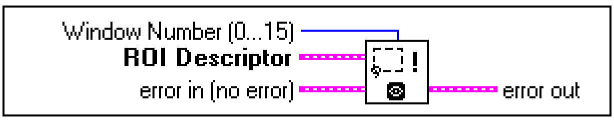
ROI Descriptor returns the ROI descriptor associated with the window.



error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ WindSetROI

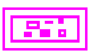
Associates an ROI with an image window.



Window Number (0…15) is a number from 0 to 15 that specifies the image window. The default value is 0.

ROI Descriptor is the descriptor that defines the region of interest that has to be associated with an image window.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.



error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Display (User)

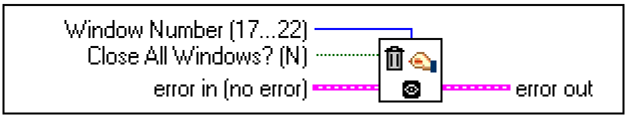
Advanced users can use this library to create and manipulate user windows. These palettes (user windows) are defined by the user and can be used to create sophisticated applications. The user window is constructed from two images that are dynamically loaded. Within these images are defined *zones* that respond to a user click, just like the buttons in LabVIEW or BridgeVIEW.

These palettes are created in the following manner:

1. Load a foreground image that appears when a zone has not been chosen.
2. Load a background image that appears when a zone has been chosen.
3. Specify the coordinates of the zones and their *mechanical action* (how they function).

IMAQ WindUserClose

Closes a user window. This VI functions in the same way as IMAQ WindClose, which is used for closing image windows.



Window Number (17…22) is a number from 17 to 22 that specifies the user window. The default value is 17.

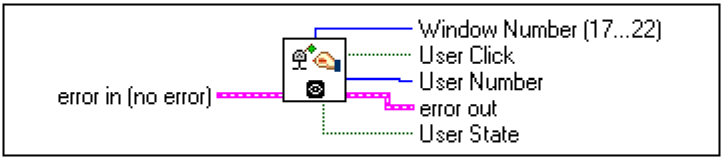
Close All Windows? (N) specifies whether to close all the image windows. The default value FALSE (No) closes only the specified window. Setting this value to TRUE closes all windows simultaneously.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ WindUserEvent

Returns the events generated through the user windows and the data associated with them.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Window Number (17…22) indicates the image window that is queried for events.

User Click returns TRUE if a zone has been chosen.

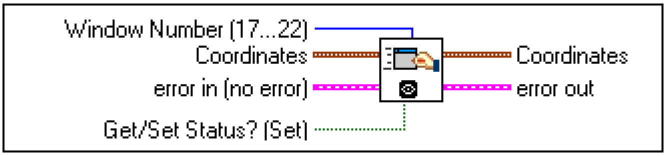
User Number returns the zone number chosen.

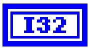
error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

User State returns the present status (TRUE or FALSE) of each zone after a click has been registered. This output is by definition TRUE when the mechanical action of the zone is latch. Reading this event causes the zone to pass to FALSE.

IMAQ WindUserMove

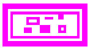
Obtains or sets the position of a user window. This VI functions in the same way as IMAQ WindMove, which is used for moving image windows.





Window Number (17…22) is a number from 17 to 22 that specifies the user window. The default value is 17.

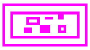
Coordinates is a cluster that contains the screen coordinates (X and Y) where the tools palette will be placed. This input is necessary only when **Get/Set Status? (Set)** is TRUE (Set).



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Get/Set Status? (Set) specifies whether to determine the present status or modify the status of the zones. The default is TRUE (Set).

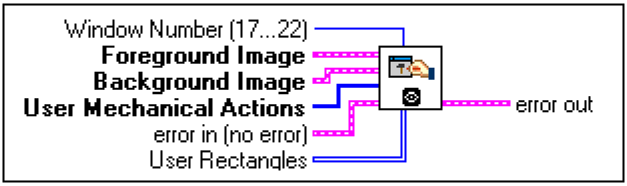
Coordinates returns the present coordinates (X and Y) of a user window.

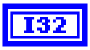


error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

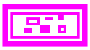
IMAQ WindUserSetup

Loads and configures the user window.

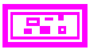




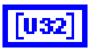
Window Number (17…22) is a number from 17 to 22 that specifies the user window. It is possible to manipulate six different user windows. The default value is 17.



Foreground Image is an 8-bit or RGB image. The corresponding part of the image is displayed when a zone within this image is FALSE.



Background Image is an 8-bit or RGB image. The corresponding part of the image is displayed when a zone within this image is TRUE.

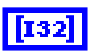


User Mechanical Actions specifies the method of operation of each zone. The following two modes are possible:

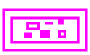
0	**Switch**	The first click causes the zone to change to TRUE. A second click on the same zone causes it to change to FALSE.
1	**Latch**	A click on the zone causes it to change to TRUE temporarily.

Note In both cases the status of the zone can be determined using IMAQ WindUserEvent or IMAQ WindUserStatus.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.



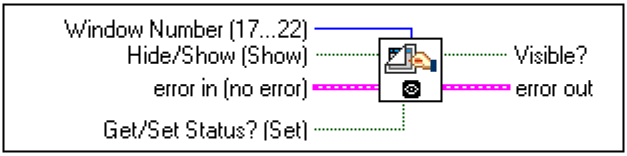
User Rectangles is a 2D array that defines the coordinates of each zone in the user window. Each line in this array must contain the four coordinates that specify the position of the zone.



error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ WindUserShow

Obtains or modifies the status regarding the visibility of a user window. This VI functions in the same way as IMAQ WindShow, which is used for displaying image windows.



Window Number (17…22) is a number from 17 to 22 that specifies the user window. The default value is 17.

Hide/Show (Show) specifies whether the tools palette is visible. Use this input only when **Get/Set Status? (Set)** is TRUE (Set).

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

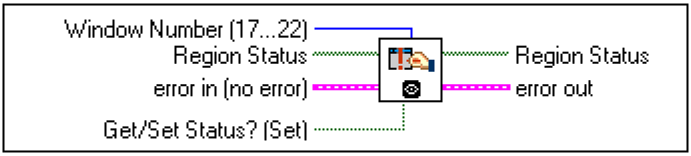
Get/Set Status? (Set) specifies whether to determine the present status or modify the status of the zones. The default is TRUE (Set).

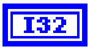
Visible? returns the present visibility status of the tools palette. A visible tools palette returns TRUE.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ WindUserStatus

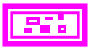
Obtains or modifies the status of each zone in a user window.





Window Number (17…22) is a number from 17 to 22 that specifies the user window. The default value is 17.

Region Status modifies the status of a user zone (TRUE or FALSE) when the input **Get/Set Status?** is TRUE (Set).



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Get/Set Status? (Set) specifies whether to determine the present status or modify the status of the zones. The default is TRUE (Set).

Region Status returns the present status (TRUE or FALSE) of each zone.

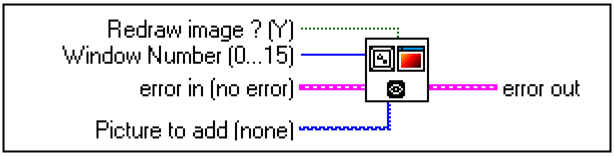
error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Display (Special)

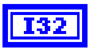
The Display (special) library contains new VIs that help you make more sophisticated user front panels.

IMAQ AddPictToWindow

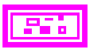
Superimposes objects (rectangles, ovals, lines, text, and so on) on an image window. These superimposed objects are drawn in a LabVIEW or BridgeVIEW picture using the Picture functions (located in the **Graphics & Sound** subpalette).



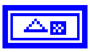
Redraw image? (Y) specifies whether to refresh the window after the picture is set.



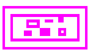
Window Number (0...15) specifies the window in which the picture is superimposed.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.



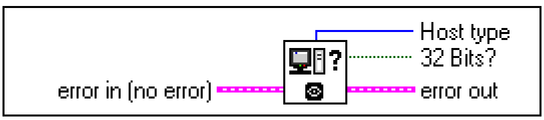
Picture to add (none) contains the objects (rectangles, ovals, lines, text, and so forth) that are superimposed on the window. These superimposed objects normally are drawn in a LabVIEW or BridgeVIEW picture using the Picture Functions.



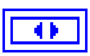
error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ GetHostType

Returns information about the host computer.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.



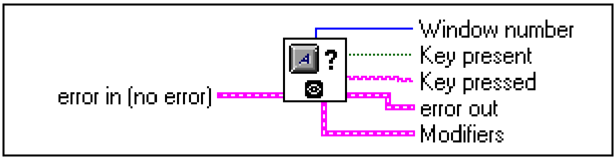
Host type returns the type of the host computer. The returned type is Windows.

32 Bits? returns TRUE.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ GetLastKey

Returns the last key pressed when the focus was on an image window.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Window number indicates the window in which the key was caught.

Key present. If TRUE, a new key was pressed. If FALSE, no new keys were pressed, and the VI returns the last key pressed.

Key pressed indicates the last key pressed.

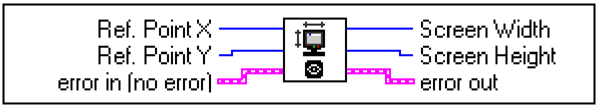
error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

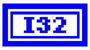
Modifiers specifies a set of flags that identifies the modifiers. Some flags are platform dependent.

- Alt
- Shift
- Caps Lock
- Ctrl

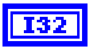
IMAQ GetScreenSize

Returns the screen size in pixels.

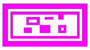




Ref. Point X. Unused.

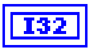


Ref. Point Y. Unused.

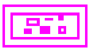


error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Screen Width gives the X size of screen.



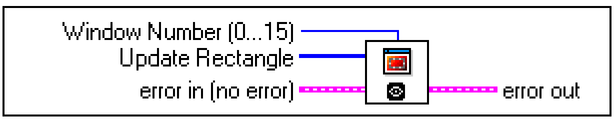
Screen Height gives the Y size of screen.



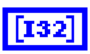
error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ WindDrawRect

Refreshes a rectangular portion of an image window. The advantage of this VI is that refreshing part of an image is faster than redrawing the whole image.



Window Number (0…15) selects the window to refresh. The default is 0.



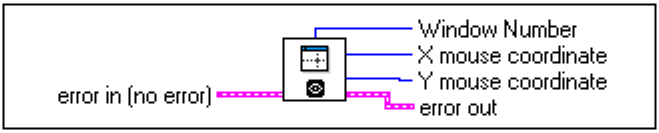
Update Rectangle is an array of numbers that represent the coordinates of the rectangle to be refreshed (Left/Top/Right/Bottom).

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ WindGetMouse

When the mouse is moved over an active window, this VI returns the window number and the mouse coordinates.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Window Number gives the window number of the active window.

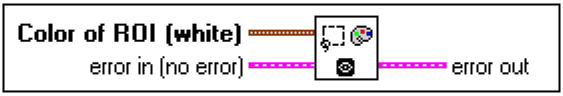
X mouse coordinate gives the X coordinate of the mouse in the active image window.

Y mouse coordinate gives the Y coordinate of the mouse in the active image window.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ WindROIColor

Selects the color to use to depict ROI contours in the image window.

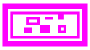


Color of ROI (white) is a cluster that specifies the color of the ROI. The default color is white.

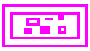
Red gives the red plane intensity. The default is 255.

Green gives the green plane intensity. The default is 255.

Blue gives the blue plane intensity. The default is 255.



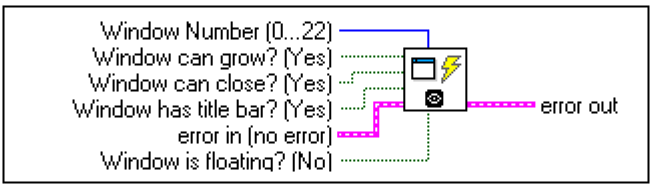
error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.



error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ WindSetup

Configures the look and attributes of a window.



Window Number (0…22) selects the window to configure. The default is 0.

Window can grow? (Yes) enables or disables the capability for the user to resize the image window. Default is TRUE, which indicates the user can resize the image window.

Window can close? (Yes) shows or does not show the close box of the window. The default is TRUE, which shows the close box.

Window has title bar? (Yes) shows or does not show the title bar. The default is TRUE, which shows the title bar.

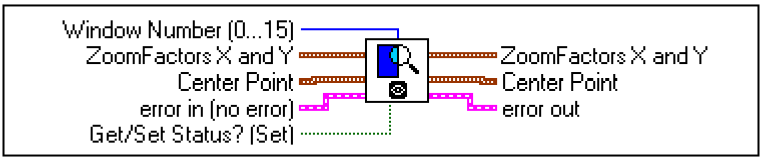
error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Window is floating? (No) produces either a normal or a floating (always on top) window. The default is FALSE, which produces a normal (non-floating) window.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ WindXYZoom

This VI is similar to IMAQ WindZoom but allows you to zoom the image at different scales in X and Y. IMAQ WindXYZoom produces rectangular pixels in displaying the image.



Window Number (0…15) is a number that specifies the image window. The default value is 0.

ZoomFactors X and Y is a cluster containing the zoom factors for X and Y scale.

Zoom Factor X ranges from –16 to +16.

Zoom Factor Y ranges from –16 to +16.

Center Point is a cluster containing two elements that describe the (*x*, *y*) coordinates used to center the image in the image window. Using **Center Point**, you can center an image with respect to a user-chosen region. Additionally, you can use **Center Point** to place only a part of an image into an image window.

X is the horizontal coordinate of the center point.

Y is the vertical coordinate of the center point.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Get/Set Status? (Set) specifies whether to determine the present status or modify the **Zoom Factor** and **Center Point**.

ZoomFactors X and Y returns the actual **Zoom Factor** in both the axes.

Zoom Factor X returns the horizontal **Zoom Factor.**

Zoom Factor Y returns the vertical **Zoom Factor**.

Center Point returns the actual **Center Point**.

X is the horizontal coordinate.

Y is the vertical coordinate.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

5

Tool VIs

This chapter describes the Tool VIs used in IMAQ Vision for G. IMAQ Vision groups the Tool VIs into three palettes:

- Use the Image tools to manipulate image.
- Use the Pixel tools to manipulate pixels.
- Use the Diverse tools to perform other various functions.

Tools (Image)

Use these tools to manipulate images, such as copying, resizing, and querying for size.

IMAQ Copy

Copies the specifications and pixels of one image into another image of the same type. You can use this function to keep an original copy of an image (for example, before processing an image). The full definition of the source image as well as the pixel data are copied to the destination image. The border size of the destination image also is modified to be equal to that of the source image.

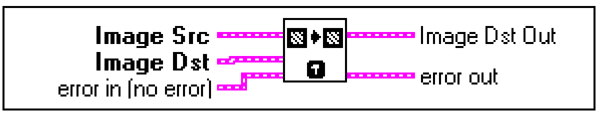


Image Src is the reference to the source (input) image.

Image Dst is the reference to the destination image. If it is connected, it must be the same type as the **Image Src**.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

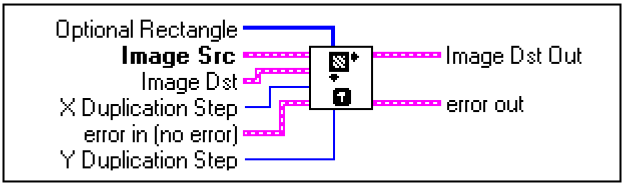
Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst.** Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ Expand

Expands (duplicates) an image or part of an image by adjusting the horizontal and vertical resolution.

8 16 F RGB HSL



Optional Rectangle defines a four-element array that contains the coordinates (Left/Top/Right/Bottom) of the region to expand. The operation applies to the entire image if the input is empty or not connected.

Image Src is the reference to the source (input) image.

Image Dst is the reference to the destination image. If it is connected, it must be the same type as the **Image Src**.

X Duplication Step specifies the number of pixel duplications per column. The column is recopied if the default value (1) is used.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Y Duplication Step specifies the number of pixel duplications per line. The row is recopied if the default value (1) is used.

Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

For example, if a 256 × 256 image is connected and the **X Duplication Step** and **Y Duplication Step** are both equal to 2, the resulting image has a resolution of 512 × 512. Each pixel in the original image now is represented by four pixels in new image (2 × 2).

Figure 5-1 illustrates an expansion of an image where **X Duplication Step** equals 2 and **Y Duplication Step** equals 3.

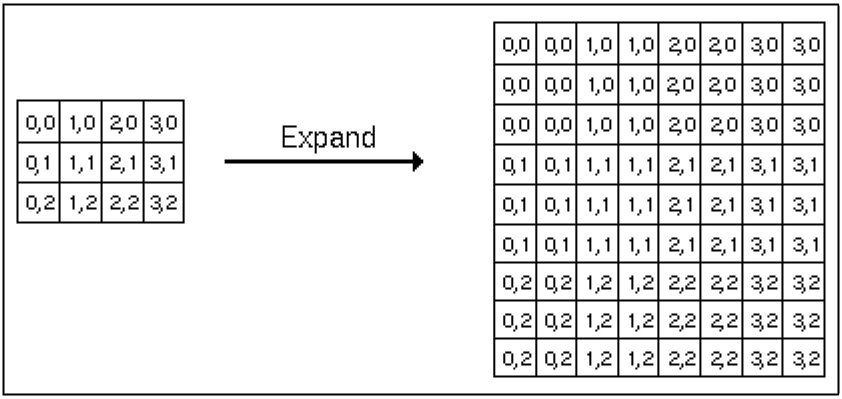
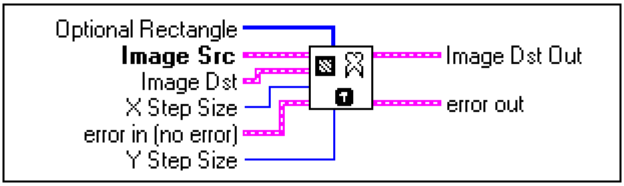


Figure 5-1. Example of Image Expansion

IMAQ Extract

Extracts (reduces) an image or part of an image with adjustment of the horizontal and vertical resolution.



Optional Rectangle defines a four-element array that contains the coordinates (Left/Top/Right/Bottom) of the region to extract. The operation applies to the entire image if the input is empty or not connected.

Image Src is the reference to the source (input) image.

Image Dst is the reference to the destination image. If it is connected, it must be the same type as the **Image Src**.

X Step Size is the vertical sampling step, which defines the columns to be extracted (the horizontal reduction ratio). For example, with an **X Step Size** equal to 3, one out of every three columns is extracted from the **Image Src** into the **Image Dst**. Each column is extracted if the default value (1) is used.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Y Step Size is the horizontal sampling step, which defines the lines to be extracted (the vertical reduction ratio). Each row is extracted if the default value (1) is used.

Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

For example, if a 512 × 512 image is connected and the **X Step Size** and **Y Step Size** are both equal to 2, the resulting image has a resolution of 256 × 256. The resulting image contains the lines from the **Image Src** 0, 2, 4, …, 510 and the columns 0, 2, 4, …, 510 from the **Image Src**.

Image Src and **Image Dst** must be of the same image type.

Figure 5-2 illustrates an extraction of an image where **X Step Size** equals 2 and **Y Step Size** equals 3.

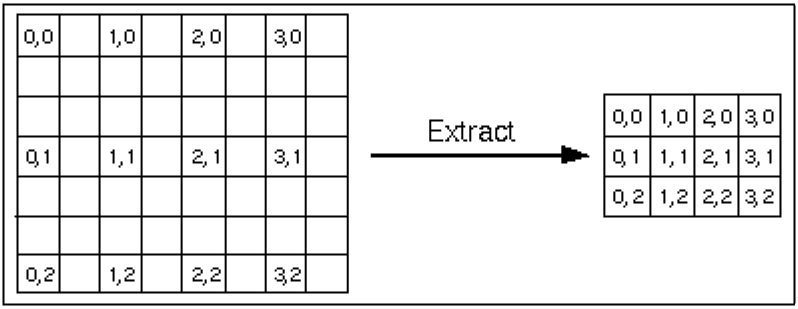
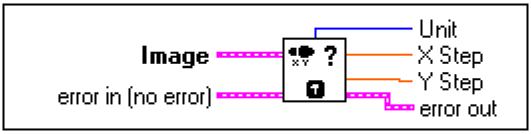


Figure 5-2. Example of Image Extraction

IMAQ GetCalibration

Obtains the present image calibration.



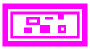
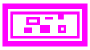
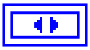


Image is the reference to the source (input) image.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.



Unit is the measuring unit associated with the image. It can have the following values:

undefined

angstrom

micrometer

millimeter

centimeter

meter

kilometer

microinch

inch

feet

nautical miles

standard miles

X Step specifies the horizontal distance separating two adjacent pixels in the specified **Unit**.

Y Step specifies the vertical distance separating two adjacent pixels in the specified **Unit**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ GetImageInfo

Gives different characteristics of the image.

8 16 F C RGB HSL

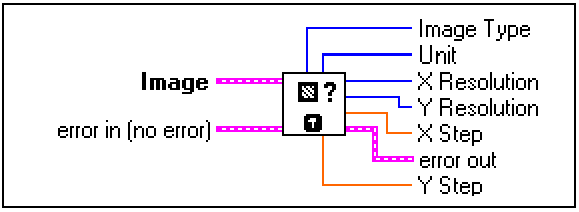


Image is the reference to the image you want to retrieve information about.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Image Type is the type of the image.

Unit is the measuring unit associated with the image. It can have the following values:

undefined

angstrom

micrometer

millimeter

centimeter

meter

kilometer

microinch

inch

feet

nautical miles

standard miles

X Resolution gives the horizontal resolution of the image.

Y Resolution gives the vertical resolution of the image.

X Step specifies the horizontal distance separating two adjacent pixels in the specified **Unit**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Y Step specifies the vertical distance separating two adjacent pixels in the specified **Unit**.

IMAQ GetImageSize

Gives information regarding the size (resolution) of the image.

8 16 F C RGB HSL

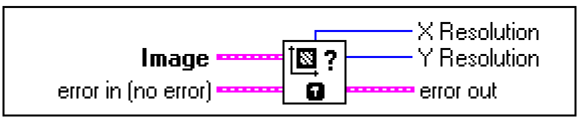


Image is the reference to the image whose size you want to determine.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

X Resolution gives the number of pixels per line.

Y Resolution gives the number of pixels per column.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ GetOffset

Returns the position of an image mask in relation to the origin of the coordinate system (0, 0). The default offset value [0, 0] is established when the image is initially created by IMAQ Create. The offset is used only for mask images. With this offset, the mask can be moved to any location in the image without having to create a new image for each mask.

8

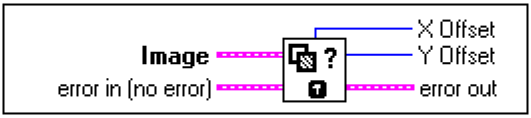


Image is the reference to the source (input) image.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

X Offset specifies the horizontal offset of the image mask.

Y Offset specifies the vertical offset of the image mask.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Figure 5-3 illustrates the use of a mask with two different offsets [0, 0] and [3, 1]. A VI processing **Image A** and using the **Image Mask** with an offset of [0, 0] and [3, 1] gives the results as shown in **Image B** and **Image C** respectively. Notice the location of the pixels.

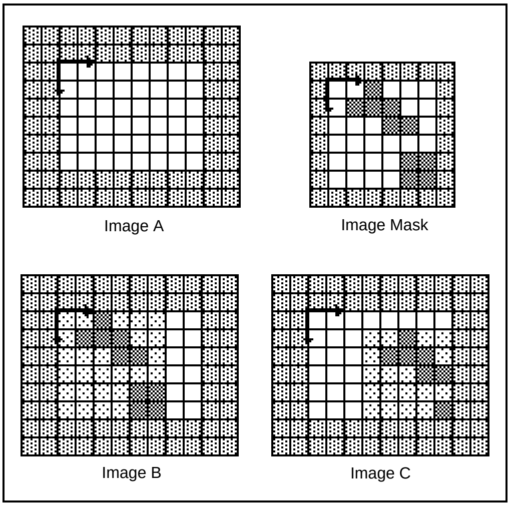


Figure 5-3. Effect of Applying Different Masks to an Image

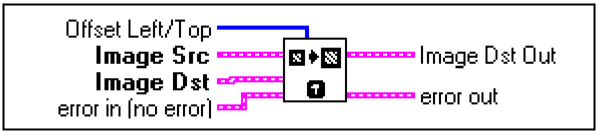
Pixels from the border

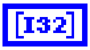
Pixels outside the mask

Pixels from the **Image Mask**

IMAQ ImageToImage

Copies a small image into part of another larger image. This VI is useful for making thumbnail sketches from multiple miniature images.





Offset Left/Top is an array specifying the **Image Dst** pixel coordinates that receive the image copied from **Image Src**.

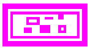


Image Src is the reference to the source (input) image, which will be copied into the larger image.

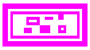
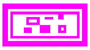


Image Dst is the reference to the destination image.


error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

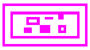
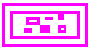


Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI.


error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

For example, an **Image Dst** with a resolution of 512 × 512 and an **Image Src** with a resolution of 256 × 256, having an **Offset Left/Top** value [256, 256], produce the operation shown in Figure 5-4.

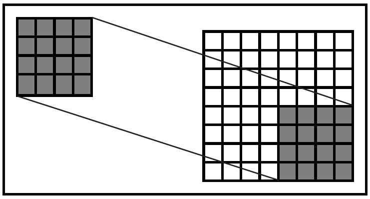


Figure 5-4. All of Source Image Copied into Destination Image

However, using an **Offset Left/Top** value [256, 256] and a resolution of 384 × 384 for the **Image Src** produces the operation shown in Figure 5-5. Notice that part of the **Image Src** is lost during the operation because it extends beyond the resolution of **Image Dst**.

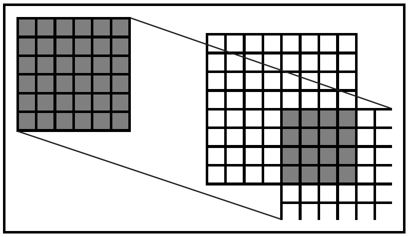
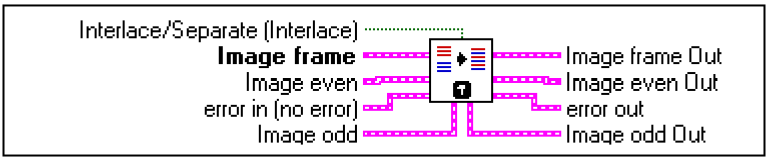


Figure 5-5. Part of Source Image Lost when Copied into Destination Image

IMAQ Interlace

Extracts odd and even fields from an interlaced image or builds an image using two field images.



Interlace/Separate (Interlace). The default is the interlace mode, which specifies that an interlaced image is built using two field images (**Image even** and **Image odd**). In the separate mode, the odd and even fields from an interlaced image (**Image frame**) are extracted.

Image frame is the reference to the image in which odd and even fields have to be extracted.

Image even is the reference to the image that forms the even lines of the interlaced image.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Image odd is the reference to the image that forms the odd lines of the interlaced image.

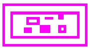


Image frame Out contains the interlaced image.

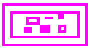
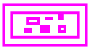


Image even Out contains the even lines of the input image.



error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

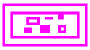
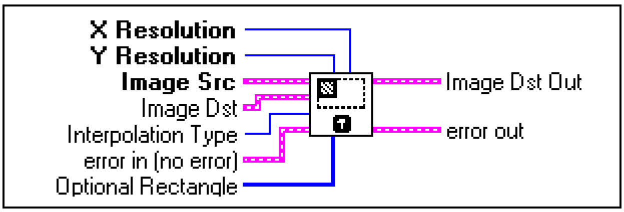


Image odd Out contains the odd lines of the input image.

Note When two fields are interlaced, the first line in the resulting frame comes from the even field, and the second comes from the odd field.

IMAQ Resample

Resamples an image to a user-defined size. You can use this VI to display a reduced or enlarged image.



X Resolution gives the final horizontal size of the image.

Y Resolution gives the final vertical size of the image.

Image Src is the reference to the source (input) image.

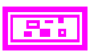
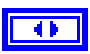


Image Dst is the reference to the destination image. If it is connected, it must be the same type as the **Image Src**.



Interpolation Type specifies the type of interpolation used to resample the image.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Optional Rectangle defines a four-element array that contains the coordinates (Left/Top/Right/Bottom) of the region to redraw. The operation applies to the entire image if the input is empty or not connected.

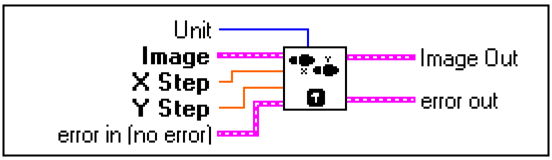
Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ SetCalibration

Sets the calibration scale for an image.

8 16 F C RGB HSL



Unit is the measuring unit associated with the image. It can have the following values:

undefined

angstrom

micrometer

millimeter

centimeter

meter

kilometer

microinch

inch

feet

nautical miles

standard miles

Image is the reference to the source (input) image.

X Step specifies the horizontal distance separating two adjacent pixels in the specified **Unit**.

Y Step specifies the vertical distance separating two adjacent pixels in the specified **Unit**.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Image Out is the reference to the destination (output) image.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ SetImageSize

Modifies the resolution of an image.

8 16 F C RGB HSL

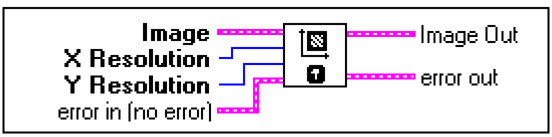


Image is the reference to the image whose size you want to modify.

X Resolution sets the new horizontal resolution of the image.

Y Resolution sets the new vertical resolution of the image.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Image Out is the reference to the image whose size is modified to the resolution specified by the **X Resolution** and **Y Resolution** parameters.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ SetOffset

Defines the position of an image mask in relation to the origin of the coordinate system (0, 0).

8

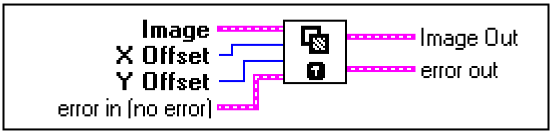


Image is the reference to the source (input) image.

X Offset specifies the horizontal offset of the image mask.

Y Offset specifies the vertical offset of the image mask.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Image Out is the reference to the destination (output) image.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

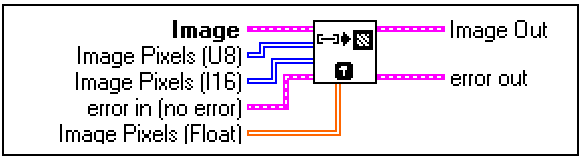
Tools (Pixel)

Use the tools in this library to manipulate values of pixels and sets of pixels.

IMAQ ArrayToImage

Creates an image from a 2D array.

8 16 F



Note Each **Image Pixels** input is specific for a particular type of data.

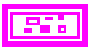


Image is the reference to the source (input) image.

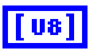


Image Pixels (U8) is a 2D array containing all the pixel values that form the image. The first index corresponds to the vertical axis and the second to the horizontal index. The final size of the image is equal to the size of the array. The image passed in the input image is forced to the same size as the array encoded by **Image Pixels**. Use this input only to create an 8-bit image.

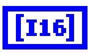
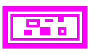


Image Pixels (I16) is a 2D array of 16-bit integers. This input is required if the image connected is a 16-bit image. Use this input only to create a 16-bit signed image.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Image Pixels (Float) is a 2D array of floating-point values. This input is required if the image connected is a floating-point image. Use this input only to create floating-point images.

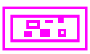


Image Out is the reference to the destination (output) image.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

See the additional VIs in Chapter 13, *Complex VIs*, for performing array-to-image transformations with complex images.

IMAQ GetPixelLine

Extracts the intensity values of a line of pixels.

8 16 F

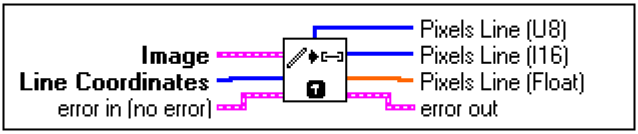


Image is the reference to the source (input) image.

Line Coordinates specifies the coordinates of the line to extract. These coordinates are in the form of an array specifying the endpoints of the line. Notice that a line with the coordinates (0, 0, 0, 255) is formed from 256 pixels. The output **Pixels Line** is an array containing the intensity values of the pixels in the selected line. Any pixels designated by the **Line Coordinates** found outside the actual image are set to zero in **Pixels Line**.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Pixels Line (U8) returns the intensity values for the specified line of pixels. This output is used only for an 8-bit image.

Pixels Line (I16) returns the intensity values for the specified line of pixels. This output is used only for a 16-bit image.

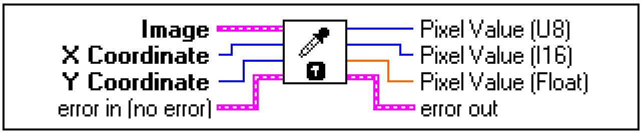
Pixels Line (Float) returns the intensity values for the specified line of pixels. This output is used only for a 32-bit floating-point image.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ GetPixelValue

Reads a pixel value from an image.

8 16 F



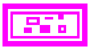
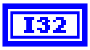
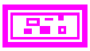


Image is the reference to the source (input) image.



X Coordinate is the horizontal coordinate of the pixel to read.

Y Coordinate is the vertical coordinate of the pixel to read.

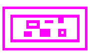


error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Pixel Value (U8) returns the specified pixel value. This output is used only for an 8-bit image.

Pixel Value (I16) returns the specified pixel value. This output is used only for an 8-bit or 16-bit image.

Pixel Value (Float) returns the specified pixel value. The float format can accept values from all supported image types (8-bit, 16-bit, or 32-bit floating point).

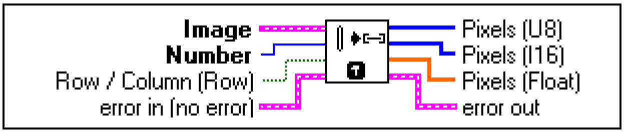


error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ GetRowCol

Extracts a range of pixel values, either a row or column, from an image.

8 **16** **F**



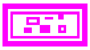
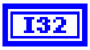
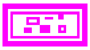


Image is the reference to the source (input) image.



Number is the row or column number to be extracted.

Row/Column (Row) defines **Number** as a row number when false and as a column number when true.

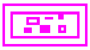


error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Pixels (U8) returns the intensity values for the specified row or column of pixels. This output is used only for an 8-bit image.

Pixels (I16) returns the intensity values for the specified row or column of pixels. This output is used only for a 16-bit image.

Pixels (Float) returns the intensity values for the specified row or column of pixels. This output is used only for a 32-bit floating-point image.

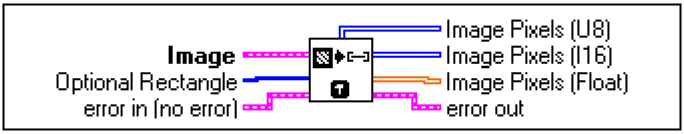


error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ ImageToArray

Extracts (copies) the pixels from an image, or part of an image, into a LabVIEW or BridgeVIEW 2D array. This array is encoded in 8 bits, 16 bits, or floating point, as determined by the type of input image.

8 16 F



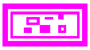
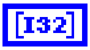
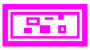


Image is the reference to the source (input) image.



Optional Rectangle defines a four-element array that contains the coordinates (Left/Top/Right/Bottom) of the region to extract. The operation applies to the entire image if the input is empty or not connected.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Image Pixels (U8) returns the extracted pixel values into a 2D array. This output is used only for an 8-bit image.

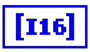
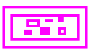


Image Pixels (I16) returns the extracted pixel values into a 2D array. This output is used only for a 16-bit image.

Image Pixels (Float) returns the extracted pixel values into a 2D array. This output is used only for a 32-bit floating-point image.

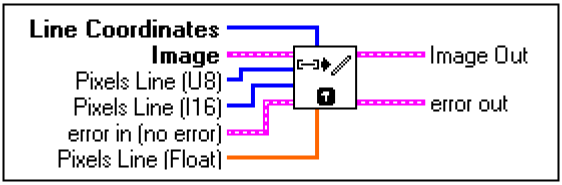


error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

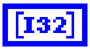
IMAQ SetPixelLine

Changes the intensity values in a line of pixels of an image.

8 16 F



Note Each **Pixels Line** input is specific for a particular type of data.



Line Coordinates specifies the coordinates of the line to change. These coordinates are in the form of an array specifying the endpoints of the line. Any pixels designated by the **Line Coordinates** found outside the actual image are not replaced.

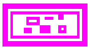
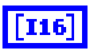
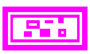


Image is the reference to the source (input) image.

Pixels Line (U8) is an array containing the coordinates of the pixel line to be drawn. This input is required if the image connected is an 8-bit image. The drawing is made between the endpoints of the line and contains the values supplied from **Pixels Line**.



Pixels Line (I16) is an array containing the coordinates of the pixel line to be drawn. This input is required if the image connected is a 16-bit image. The drawing is made between the endpoints of the line and contains the values supplied from **Pixels Line**.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Pixels Line (Float) is an array containing the coordinates of the pixel line to be drawn. This input is required if the image connected is a 32-bit floating-point image. The drawing is made between the endpoints of the line and contains the values supplied from **Pixels Line**.

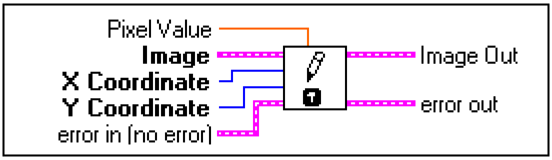
Image Out is the reference to the destination (output) image.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ SetPixelValue

Changes a pixel value in an image.

8 16 F



Pixel Value contains the replacement pixel value.

Image is the reference to the source (input) image.

X Coordinate is the horizontal coordinate of the pixel to modify.

Y Coordinate is the vertical coordinate of the pixel to modify.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

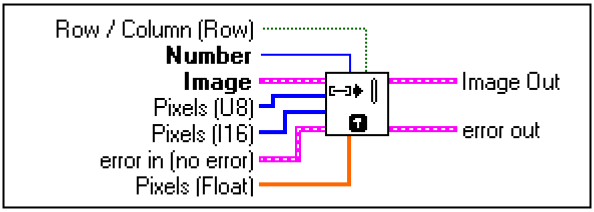
Image Out is the reference to the destination (output) image.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ SetRowCol

Changes the intensity values in either a row or a column of pixels in an image.

8 16 F



Note Each **Pixels** input is specific for a particular type of data.

TF **Row/Column (Row)** defines **Number** as a row number when false and as a column number when true.

I32 **Number** is the row or column number to be replaced in the image.

Image is the reference to the source (input) image.

[U8] **Pixels (U8)** is an array containing the values to put in the specified row or column of the image. This input is required if the image connected is an 8-bit image.

[I16] **Pixels (I16)** is an array of 16-bit integers containing the values to put in the specified row or column of the image. This input is required if the image connected is a 16-bit image.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

[SGL] **Pixels (Float)** is an array of floating-point values containing the values to put in the specified row or column of the image. This input is required if the image connected is a floating-point image.

Image Out is the reference to the destination (output) image.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Tools (Diverse)

The tools described in this section are used for miscellaneous manipulations of images.

IMAQ ClipboardToImage

Copies the clipboard data into an image.

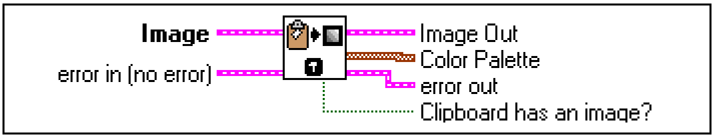


Image is the reference to the source (input) image.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Image Out contains a copy of the clipboard if the clipboard is an image.

Color Palette is the color palette that is stored on the clipboard. A grayscale ramp is returned if no color palette is found on the clipboard.

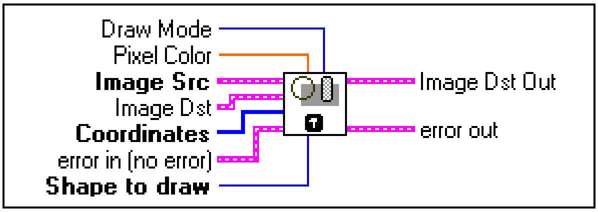
error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

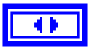
Clipboard has an image? returns a TRUE value if the clipboard contains an image.

IMAQ Draw

Draws geometric objects in an image.

8 **16** **F**





Draw Mode defines how to draw the object and has the following choices:

Frame	(Default) Specifies to use **Pixel Color** when tracing the contour
Paint	Specifies to use **Pixel Color** when tracing the contour and the interior of the shape
Invert Frame	Specifies to use the inverse of the pixel values when drawing the contour
Invert Paint	Specifies to use the inverse of the pixel values when drawing the contour and the interior of the shape

Pixel Color is the pixel value used for tracing the design. This value is not used when in the mode **Invert Frame** or **Invert Paint**. The default is 0.

Image Src is the reference to the source (input) image.

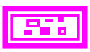
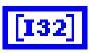


Image Dst is the reference to the destination image. If it is connected, it must be the same type as the **Image Src**.



Coordinates is an array of four elements. A line is specified by the two points forming it. Rectangles and ovals are specified by their bounding rectangle, with the format (Left/Top/Right/Bottom). In these cases, the tracing of a rectangle or oval stops at the column (*Right* – 1) and at the row (*Bottom* – 1). The values by default are (0, 0, *SizeX*, *SizeY*) where (*SizeX*, *SizeY*) is the resolution of the image. The default is used if the input is 0 or is not connected.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Shape to draw is the form to draw. The following shapes are available:

Line	(Default) Defined by the two points specified in the array **Coordinates**
Rectangle	Defined by the bounding rectangle specified in the array **Coordinates**
Oval	Defined by the bounding rectangle specified in the array **Coordinates**

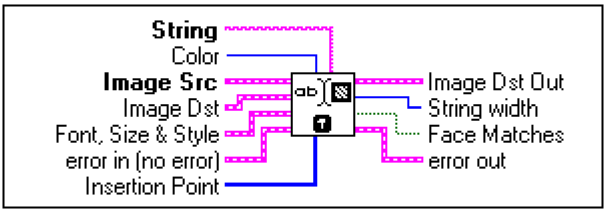
Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ DrawText

Inserts text in an image.

8 RGB



String (empty by default) is the text to write in an image. The string can be composed of multiple lines separated by a hard return.

Color is the mode for writing the text. The default is 0, which specifies white. The following values are possible:

White	(Default) White on the image background
Black	Black on the image background
Inverted	Text inverted on the image background
Black on White	
White on Black	

Image Src is the image reference source. It must be an 8-bit or RGB image.

Image Dst is the reference of the image destination. If it is connected, it must be the same type as the **Image Src**.

Font, Size & Style is a cluster that enables the user to choose the font, size, style, and alignment and contains the following elements:

desired font (Application) specifies the character type of the text. The following values are possible:

User-specified font

(Default) Application font

System font

Dialog font

user-specified font is a cluster containing the specific font characteristics for the text to draw. This specification is ignored unless the **desired font** control is set to user-specified font.

Font Name is the name of the user-specified font.

Strikeout? If TRUE, text appears in strikeout.

Italic? If TRUE, text appears in italic.

Underline? If TRUE, text appears underlined.

Outline? This field is reserved.

Shadow? This field is reserved.

Bold? If TRUE, text appears in bold.

Size is the size of the font. The default is 9.

Alignment specifies the alignment of the text. The following values are possible: Left (default), Center, and Right.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Insertion Point is an array (*x* and *y*) specifying the location in which the text is inserted. The text position depends on the alignment mode chosen. The default is (0, 0).

Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.

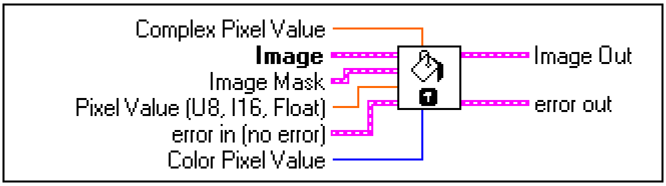
String width returns the string length of the text.

Face Matches returns TRUE if the user-specified font name was found. Returns FALSE if the font name was not found. Note that if the requested font name is not found, the operating system chooses the best match.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ FillImage

Fills an image and its border with a specified value.



Complex Pixel Value specifies the value used for filling a complex image.

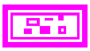


Image is the reference to the source (input) image.

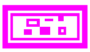


Image Mask is an 8-bit image that specifies the region in the image to modify. Only those pixels in the original image that correspond to an equivalent pixel in the mask different than 0 are replaced by the **Pixel Value**. All other pixels keep their original value. The complete image is modified if **Image Mask** is not connected.

Pixel Value (U8, I16, Float) specifies the value with which the image is to be filled. This value is used for 8-bit, 16-bit, and 32-bit floating-point images.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Color Pixel Value specifies the value used for filling a color image.

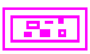


Image Out contains the image that has been filled with the specified pixel value.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ GetPointsOnContour

Finds the number of edge segments in an image and returns the coordinates of the pixels in each segment. Any pixel that is greater than zero is considered as an edge location.

This VI joins adjoining edge pixels into edge segments. An edge segment is considered closed if it forms a loop. Each edge segment is given a weight based on the pixel gray values along that edge. An edge segment with high gray values has a higher weight.

8 16 F

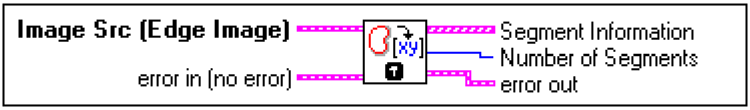


Image Src (Edge Image) is the reference to the source (input) image.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Segment Information is an array of clusters containing the following information:

- I16 **Number of Points** is the number of points that make up the edge segment.
- I16 **Open/Closed** indicates whether the edge segment is open or closed.
- DBL **Weight** indicates the significance of the edge in terms of the gray values that constitute the edge.
- **Contour Points** is an array of clusters, where each point on the edge is represented by the following coordinates:
  - DBL **X** returns the X coordinate value of the edge pixel in the image.
  - DBL **Y** returns the Y coordinate value of the edge pixel in the image.
  - DBL **Curvature** returns the change in slope at this edge point of the segment.

DBL **X Displacement** returns the X displacement of the current edge pixel from a cubic spline fit of the current edge segment.

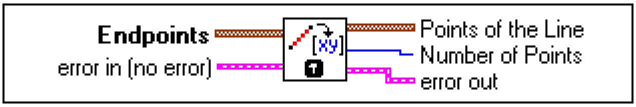
DBL **Y Displacement** returns the Y displacement of the current edge pixel from a cubic spline fit of the current edge segment.

I32 **Number of Segments** returns the number of edge segments in the image.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ GetPointsOnLine

Given the endpoints of a line, returns an array of all the points comprising the line.



Endpoints specifies the points at each end of the line.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

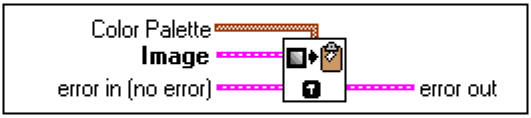
Points of the Line returns an array of clusters, where each point of the line is represented by its X and Y coordinates.

I32 **Number of Points** returns the number of points found on the line.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ ImageToClipboard

Copies the image to the operating system clipboard.



Color Palette can be applied to an 8-bit image. It can be taken directly from the output of IMAQ GetPalette or specified by the user. It is formed from an array of 256-element clusters for each of the three color planes (red, green, and blue). A specific color is the result of affecting a value between 0 and 255 for each of the three color planes. If the three planes have the identical value, a gray level is obtained (0 specifies black and 255 specifies white). By default the palette is a grayscale ramp.

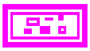
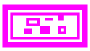


Image is the reference to the source (input) image.



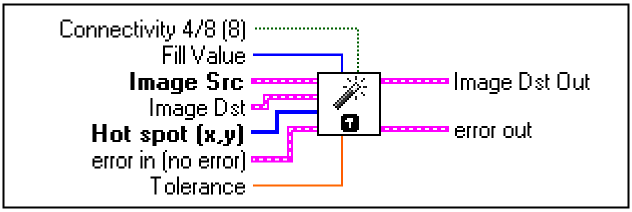
error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ MagicWand

Creates an image mask by extracting a region surrounding a reference pixel, called the origin, and using a tolerance (+ or –) of intensity variations based on this reference pixel. Using this origin, the VI searches for its neighbors with an intensity that equals or falls within the tolerance value of the point of reference. The resulting image is binary. The image passed as input for **Image Dst** must be an 8-bit image. If the same image is entered for **Image Src** and **Image Dst**, both must be 8-bit images.

8 16 F



TF — **Connectivity 4/8 (8)** specifies the type of connectivity the algorithm uses to create the mask. The default is Connectivity-8.

U8 — **Fill Value** is the value used for the highlighted pixels in the destination image. The default is 1.

Image Src is the image reference source. It must be an 8-bit or RGB image.

Image Dst is the reference of the image destination. It must be an 8-bit image.

[I32] — **Hot spot (x,y)** is an array counting the (*x*, *y*) coordinates of the origin pixel chosen from the image source.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

SGL — **Tolerance** is the maximum authorized deviation from the origin. All pixels satisfying the tolerance criteria (origin pixel – tolerance/origin pixel + tolerance) and connectivity criteria, as specified in **Connectivity 4/8 (8)**, are highlighted and all other pixels are turned off. The default is 20.

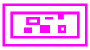
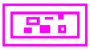


Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.



error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

6

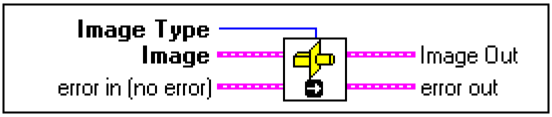
Conversion VIs

This chapter describes the Conversion VIs in IMAQ Vision.

IMAQ Cast

Converts the current image type of an image to the image type specified by **Image Type**. IMAQ Cast uses the same conversion rules as IMAQ Convert.

8 16 F C RGB HSL



I32

Image Type specifies the image type into which the input **Image** is converted. The following values are valid:

8	**8 bits**	8 bits per pixel (unsigned, standard monochrome)
16	**16 bits**	16 bits per pixel (signed)
F	**float**	32 bits (floating point) per pixel
C	**complex**	2 × 32 bits (floating point) per pixel
RGB	**RGB**	32 bits per pixel (RGB, standard color)
HSL	**HSL**	32 bits per pixel (color)

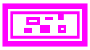


Image is both the image to be converted (input) and the image that receives the conversion (output). This VI changes only the image type of the image. The conversion rules are the same as described in IMAQ Convert.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Image Out is the reference to the input image with the new image type.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ Convert

Converts the image type specified by **Image Src** into the image type specified by **Image Dst**.

8 16 F C RGB HSL

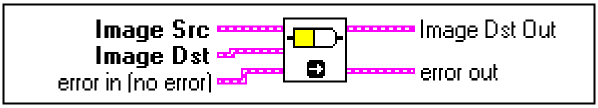


Image Src is the reference to the source (input) image.

Image Dst is the reference to the destination image.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

The conversion rules are performed as a function of the image type specified by **Image Src** and **Image Dst**. The image type encoded by **Image Dst** defines the how the conversion works. Table 6-1 describes the conversion rules.

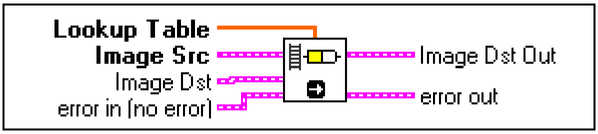
Table 6-1. Image Type Conversion Rules

Image Type to Convert	Conversion Rules
8 to 16 or F	Pixel values are recopied (0 to 255).
8 to RGB or HSL	Pixel values are copied into each of the three color planes (red, green, and blue) for RGB images, or into the luminance plane for HSL images.
16 to 8	Pixel values less than 0 are set to 0. Pixel values between 0 and 255 are recopied. Pixel values greater than 255 are set to 255.
16 to F	Pixel values are recopied (–32,768 to 32,767).
16 to RGB or HSL	Pixel values are copied into each of the three color planes with the same conversion rule as 16-bit to 8-bit.
F to 8	Pixel values less than 0 are set to 0. Pixel values between 0 and 255 are recopied. Pixel values greater than 255 are set to 255.
F to 16	Pixel values less than –32,768 are set to –32,768. Pixel values between –32,768 and 32,767 are recopied. Pixel values greater than 32,767 are set to 32,767.
F to RGB or HSL	Same conversion rule as 16-bit to RGB or HSL.
RGB to 8 or 16 or F	The pixel values are assigned the average of the three color planes (red, green, and blue).
RGB to HSL	The RGB values are converted into equivalent HSL values.
HSL to 8 or 16 or F	The pixel values are assigned the value of the luminance plane.
HSL to RGB	The HSL values are converted into equivalent RGB values.

IMAQ ConvertByLookup

Converts an image by using a look-up table that is encoded in floating-point values.

8 **16**



Lookup Table is an array consisting of a maximum of 256 elements if **Image Src** has an 8-bit image, or a maximum of 65536 elements if the **Image Src** has a 16-bit image. This array is filled with values equal to the index if it has fewer elements than the amount demanded by the image type in **Image Src**. The look-up table can be used to calculate a polynomial giving a relation between a gray-level value and a user value. VIs capable of analyzing floating-point type images can be used to directly quantify an image, or regions from an image, in user values after converting the image into a floating-point type image.

Image Src is the image to be converted. It must be an 8-bit or 16-bit image.

Image Dst is the image that receives the conversion. The image type for **Image Dst** can take the following values:

16-bit if **Image Src** has an 8-bit image

32-bit floating point if **Image Src** has an 8-bit or 16-bit image

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

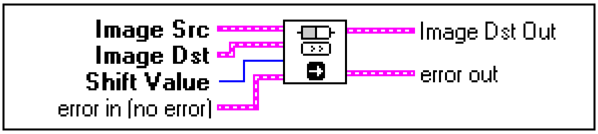
Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ Shift16To8

Converts a 16-bit image to an 8-bit image. The VI executes this conversion by shifting the 16-bit pixel values to the right by the specified number of shift operations and then truncating to get an 8-bit value.

16



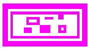


Image Src is the reference to the 16-bit image.

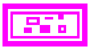
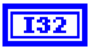
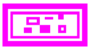


Image Dst is the reference to the 8-bit output image.



Shift Value specifies the number of right shifts by which each pixel value in the input image is shifted.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

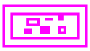
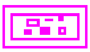


Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI.



error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

7

Operator VIs

This chapter describes the Operator VIs in IMAQ Vision. You can use arithmetic and logic operators.

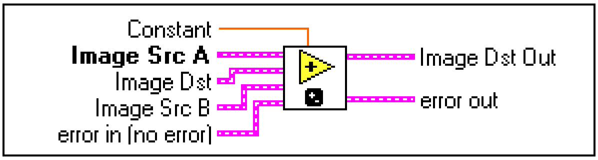
Arithmetic Operators

These VIs perform arithmetic operations in IMAQ Vision.

IMAQ Add

Adds two images or an image and a constant.

8 16 F RGB



Constant is the value added to the input **Image Src A** for image-constant operations. The constant is rounded down in the cases in which the image is encoded as an integer. The default is 0.

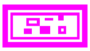


Image Src A is the source (input) image A.

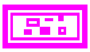


Image Dst is the reference to the destination image.

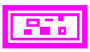
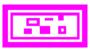


Image Src B is the reference to the source (input) image B.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

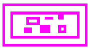
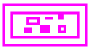


Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src A**.



error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

An operation between an image and a constant occurs when the input **Image Src B** is not connected. The two possibilities are distinguished in the following equations:

$$Dst(x, y) = SrcA(x, y) + SrcB(x, y)$$

or

$$Dst(x, y) = SrcA(x, y) + Constant$$

The different image-type combinations supported by this VI are described in the following equations. The first symbol represents the image connected to **Image Src A**, and the second symbol represents the image type connected to **Image Src B**. The third symbol represents the image type that should be connected to the output **Image Dst**.

8 + 8 = 8	8 + 16 = 16	8 + F = F
16 + 8 = 16	16 + 16 = 16	16 + F = F
F + 8 = F	F + 16 = F	F + F = F
RGB + 8 = RGB	RGB + RGB = RGB	

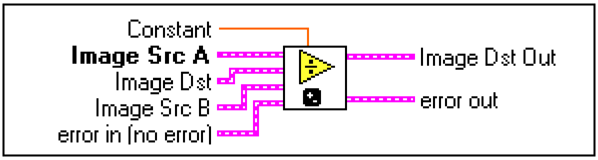
To add a constant to an image, the output **Image Dst** must be connected to the same image type as the input **Image Src A**.

When an 8-bit image or constant is added to an RGB image, it is added to each plane of the RGB image. When IMAQ Add is performed on two RGB images, each color plane (red, green, and blue) of **Image Src A** is added to the corresponding color plane of **Image Src B**. Each of these additions is similar to an IMAQ Add operation on two 8-bit images.

IMAQ Divide

Divides one image by another or an image by a constant.

8 16 F RGB



Constant. The input **Image Src A** is divided by the **Constant** value for image-constant operations. The default is 1.

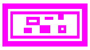


Image Src A is the reference to the source (input) image A.

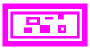


Image Dst is the reference to the destination image.

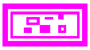
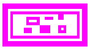


Image Src B is the reference to the source (input) image B.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

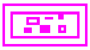
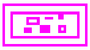


Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src A**.



error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

An operation between an image and a constant occurs when the input **Image Src B** is not connected. The two possibilities are distinguished in the following equations.

$$Dst(x, y) = SrcA(x, y) \div SrcB(x, y)$$

or

$$Dst(x, y) = SrcA(x, y) \div Constant$$

The different image-type combinations supported by this VI are described in the following equations. The first symbol represents the image connected to **Image Src A**, and the second symbol represents the image type connected to **Image Src B**. The third symbol represents the image type that should be connected to the output **Image Dst**.

$\boxed{8} \div \boxed{8} = \boxed{8}$

$\boxed{16} \div \boxed{8} = \boxed{16}$ $\boxed{16} \div \boxed{16} = \boxed{16}$

$\boxed{F} \div \boxed{8} = \boxed{F}$ $\boxed{F} \div \boxed{16} = \boxed{F}$ $\boxed{F} \div \boxed{F} = \boxed{F}$

$\boxed{RGB} \div \boxed{8} = \boxed{RGB}$ $\boxed{RGB} \div \boxed{RGB} = \boxed{RGB}$

To divide an image by a constant, the output **Image Dst** must be connected to the same image type as the input **Image Src A**.

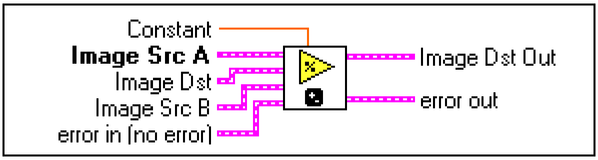
Division by 0 is not allowed. If the constant is 0 it automatically is replaced by 1. If one of the two source images is empty, the result is a copy of the other.

When an RGB image is divided by an 8-bit image or constant, each plane of the RGB image is divided by the 8-bit image or constant. When IMAQ Divide is performed on two RGB images, each color plane (red, green, and blue) of **Image Src A** is divided by the corresponding color plane of **Image Src B**. Each of these divisions is similar to an IMAQ Divide operation on two 8-bit images.

IMAQ Modulo

Executes *modulo division* (remainder) of one image by another or an image by a constant.

8 16 F RGB



SGL **Constant**. The input **Image Src A** is divided by the **Constant** value for image-constant operations. The default is 1.

Image Src A is the reference to the source (input) image A.

Image Dst is the reference to the destination image. If it is connected, it must be the same type as the **Image Src A**.

Image Src B is the reference to the source (input) image B.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src A**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

An operation between an image and a constant occurs when the input **Image Src B** is not connected. The two possibilities are distinguished in the following equations:

$$Dst(x, y) = SrcA(x, y) \% SrcB(x, y)$$

or

$$Dst(x, y) = SrcA(x, y) \% Constant$$

If **Image Src A** is a 32-bit floating-point image, the following operation is performed:

$$Dst(x, y) = SrcA(x, y) - SrcB(x, y) \times E(SrcA(x, y) \div SrcB(x, y))$$

or

$$Dst(x, y) = SrcA(x, y) - Constant \times E(SrcA(x, y) \div Constant)$$

where $E(x)$ is the integer part of x.

The different image-type combinations supported by this VI are described in the following equations. The first symbol represents the image connected to **Image Src A**, and the second symbol represents the image type connected to **Image Src B**. The third symbol represents the image type that should be connected to the output **Image Dst**.

$\boxed{8}\ \%\ \boxed{8} = \boxed{8}$

$\boxed{16}\ \%\ \boxed{8} = \boxed{16}$ $\boxed{16}\ \%\ \boxed{16} = \boxed{16}$

$\boxed{F}\ \%\ \boxed{8} = \boxed{F}$ $\boxed{F}\ \%\ \boxed{16} = \boxed{F}$ $\boxed{F}\ \%\ \boxed{F} = \boxed{F}$

$\boxed{RGB}\ \%\ \boxed{8} = \boxed{RGB}$ $\boxed{RGB}\ \%\ \boxed{RGB} = \boxed{RGB}$

To modulo-divide an image by a constant, the output **Image Dst** must be connected to the same image type as the input **Image Src A**.

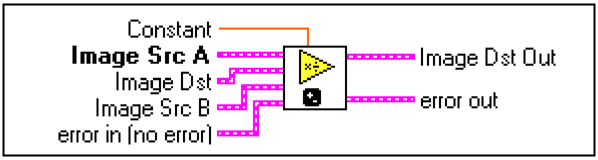
Division by 0 is not allowed. If 0 is found in the divider, it automatically is replaced by 1. If one of the two source images is empty, the result is a copy of the other.

When an RGB image is modulo-divided by an 8-bit image or constant, each plane of the RGB image is modulo-divided by the 8-bit image or constant. When IMAQ Modulo is performed on two RGB images, each color plane (red, green, and blue) of **Image Src A** is modulo-divided by the corresponding color plane of **Image Src B**. Each of these modulo-divisions is similar to an IMAQ Modulo operation on two 8-bit images.

IMAQ MulDiv

Computes a ratio between two images. Each pixel in input **Image Src A** is multiplied by the integer value specified in the input **Constant** before being divided by the equivalent pixel found in input **Image Src B**. To avoid losing information, a temporary variable giving higher definition is used to perform the operation. If the background is lighter than the image, this function can correct the background. In a background correction image, **Image Src A** is the acquired image, and **Image Src B** is the light background.

8 16 F RGB



SGL

Constant. Each pixel in **Image Src A** is multiplied by the **Constant** value before being divided by the equivalent pixel in **Image Src B**. The default is 255, which corresponds to the maximum value for a pixel encoded in an 8-bit image.

Image Src A is the reference to the source (input) image A.

Image Dst is the reference to the destination image. If it is connected, it must be the same type as the **Image Src A**.

Image Src B is the reference to the source (input) image B.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src A**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

This VI performs the following equation:

$$Dst(x, y) = (SrcA(x, y) \times Constant) \div SrcB(x, y)$$

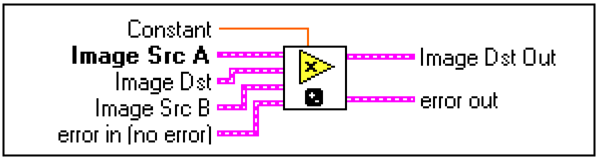
All input images must of be the same image type.

Division by 0 is not allowed. If this value is found in **Image Src B**, the equivalent pixel value from **Image Src A** is directly applied to **Image Dst**. If one of the two source images is empty, the result is a copy of the other.

IMAQ Multiply

Multiplies two images or an image and a constant.

8 16 F RGB



Constant. The input **Image Src A** is multiplied by the **Constant** value for image-constant operations. The default is 1.

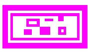


Image Src A is the reference to the source (input) image A.

Image Dst is the reference to the destination image.

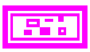


Image Src B is the reference to the source (input) image B.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src A**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

An operation between an image and a constant occurs when the input **Image Src B** is not connected. The two possibilities are distinguished in the following equations:

$$Dst(x, y) = SrcA(x, y) \times SrcB(x, y)$$
or
$$Dst(x, y) = SrcA(x, y) \times Constant$$

The different image-type combinations supported by this VI are described in the following equations. The first symbol represents the image connected to **Image Src A**, and the second symbol represents the image type connected to **Image Src B**. The third symbol represents the image type that should be connected to the output **Image Dst**.

$\boxed{8} \times \boxed{8} = \boxed{8}$	$\boxed{8} \times \boxed{16} = \boxed{16}$	$\boxed{8} \times \boxed{F} = \boxed{F}$
$\boxed{16} \times \boxed{8} = \boxed{16}$	$\boxed{16} \times \boxed{16} = \boxed{16}$	$\boxed{16} \times \boxed{F} = \boxed{F}$
$\boxed{F} \times \boxed{8} = \boxed{F}$	$\boxed{F} \times \boxed{16} = \boxed{F}$	$\boxed{F} \times \boxed{F} = \boxed{F}$
$\boxed{RGB} \times \boxed{8} = \boxed{RGB}$	$\boxed{RGB} \times \boxed{RGB} = \boxed{RGB}$	

To multiply a constant and an image, the output **Image Dst** must be connected to the same image type as the input **Image Src A**.

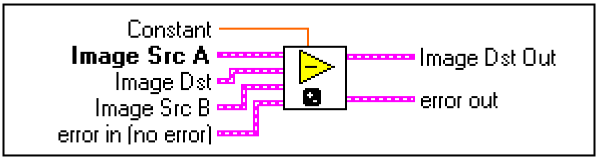
If one of the two source images is empty, the result is a copy of the other.

When an RGB image is multiplied by an 8-bit image or constant, each plane of the RGB image is multiplied by the 8-bit image or constant. When IMAQ Multiply is performed on two RGB images, each color plane (red, green, and blue) of **Image Src A** is multiplied by the corresponding color plane of **Image Src B**. Each of these multiplications is similar to an IMAQ Multiply operation on two 8-bit images.

IMAQ Subtract

Subtracts one image from another or a constant from an image.

8 16 F RGB



Constant is the value subtracted from the input **Image Src A** for image-constant operations. The constant is rounded down in the cases in which the image is encoded as an integer. The default is 0.

Image Src A is the reference to the source (input) image A.

Image Dst is the reference to the destination image.

Image Src B is the reference to the source (input) image B.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src A**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

An operation between an image and a constant occurs when the input **Image Src B** is not connected. The two possibilities are distinguished in the following equations:

$$Dst(x, y) = SrcA(x, y) - SrcB(x, y)$$

or

$$Dst(x, y) = SrcA(x, y) - Constant$$

The different image-type combinations supported by this VI are described in the following equations. The first symbol represents the image connected to **Image Src A**, and the second symbol represents the image type connected to **Image Src B**. The third symbol represents the image type that should be connected to the output **Image Dst**.

8 – 8 = 8

16 – 8 = 16 16 – 16 = 16

F – 8 = F F – 16 = F F – F = F

RGB – 8 = RGB RGB – RGB = RGB

To subtract a constant from an image, the output **Image Dst** must be connected to the same image type as the input **Image Src A**.

If one of the two source images is empty, the result is a copy of the other.

When an 8-bit image or constant is subtracted from an RGB image, it is subtracted from each plane of the RGB image. When IMAQ Subtract is performed on two RGB images, each color plane (red, green, and blue) of **Image Src A** subtracts the corresponding color plane of **Image Src B**. Each of these subtractions is similar to an IMAQ Subtract operation on two 8-bit images.

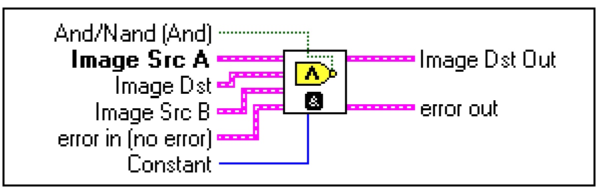
Logic Operators

These VIs perform bit-wise operations on images.

IMAQ And

Performs an AND or NAND operation on two images or an image and a constant.

8 16 RGB HSL



And/Nand (And) is the result from a logic operation. If set to TRUE, the result of the AND is inverted, producing a NAND. The default is FALSE, which specifies an AND.

Image Src A is the reference to the source (input) image A.

Image Dst is the reference to the destination image. If it is connected, it must be the same type as the **Image Src A**.

Image Src B is the reference to the source (input) image B.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Constant is a binary constant used for image-constant operations. The default is 0.

Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src A**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

All connected images must be the same image type. An operation between an image and a constant occurs when the input **Image Src B** is not connected.

This VI is performed for each pixel (x, y) in the following manner:

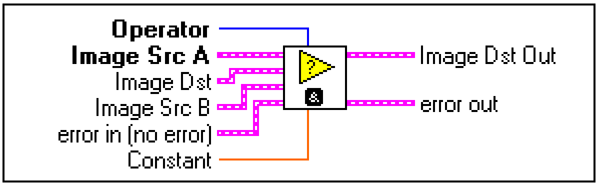
If two images are connected on input, then $Dst(x, y) = SrcA(x, y)$ AND $SrcB(x, y)$.

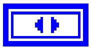
If the input **Image Src B** is not connected, then $Dst(x, y) = SrcA(x, y)$ AND *Constant*.

IMAQ Compare

Performs comparison operations between two images or an image and a constant. An operation between an image and a constant occurs when the input **Image Src B** is not connected.

8 **16** **F**





Operator specifies the comparison operator to use. You can choose from the following valid operators:

Average	Calculates the average
Min	Extracts the smallest value *Min* [*SrcA*(*x*, *y*), *SrcB*(*x*, *y*)]
Max	Extracts the largest value *Max* [*SrcA*(*x*, *y*), *SrcB*(*x*, *y*)]
Clear if <	If $SrcA(x, y) < SrcB(x, y)$ or a constant, then $Dst\ (x, y) = 0$, else $Dst(x, y) = SrcA(x, y)$
Clear if < or =	If $SrcA(x, y) \leq SrcB(x, y)$ or a constant, then $Dst\ (x, y) = 0$, else $Dst(x, y) = SrcA(x, y)$
Clear if =	If $SrcA(x, y) = SrcB(x, y)$ or a constant, then $Dst\ (x, y) = 0$, else $Dst(x, y) = Src\ A(x, y)$
Clear if > or =	If $SrcA(x, y) \geq SrcB(x, y)$ or a constant, then $Dst\ (x, y) = 0$, else $Dst(x, y) = SrcA(x, y)$
Clear if >	If $Src\ A(x, y) > SrcB(x, y)$ or a constant, then $Dst\ (x, y) = 0$, else $Dst(x, y) = SrcA(x, y)$

Image Src A is the reference to the source (input) image A.

Image Dst is the reference to the destination image. If it is connected, it must be the same type as the **Image Src A**.

Image Src B is the reference to the source (input) image B.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Constant is the value used in comparison with **Image Src A** for image-constant operations. The default is 0.

Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

The different image-type combinations supported by this VI are described in the following equations. The first symbol represents the image connected to **Image Src A**, and the second symbol represents the image type connected to **Image Src B**. The third symbol represents the image type that should be connected to the output **Image Dst**.

$\boxed{8} \# \boxed{8} = \boxed{8}$

$\boxed{16} \# \boxed{8} = \boxed{16}$ $\boxed{16} \# \boxed{16} = \boxed{16}$

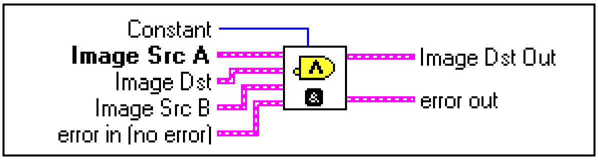
$\boxed{F} \# \boxed{8} = \boxed{F}$ $\boxed{F} \# \boxed{16} = \boxed{F}$ $\boxed{F} \# \boxed{F} = \boxed{F}$

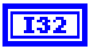
For all comparison operations, the output **Image Dst** must be connected to the same image type as the input **Image Src A.**

If one of the two source images is empty, the result is a copy of the other.

IMAQ LogDiff

Keeps bits found in **Image Src A** that are absent from **Image Src B.**





Constant is a constant value that can replace **Image Src B** for image-constant operations. The default is 0.

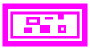


Image Src A is the reference to the source (input) image A.

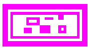


Image Dst is the reference to the destination image. If it is connected, it must be the same type as the **Image Src A**.

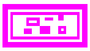
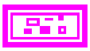
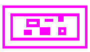


Image Src B is the reference to the source (input) image B.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src A**.



error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

All connected images must be the same image type. An operation between an image and a constant occurs when the input **Image Src B** is not connected.

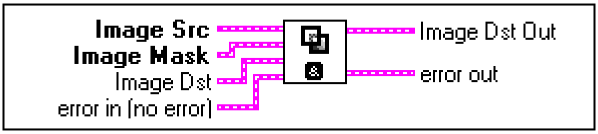
This VI is performed for each pixel (x, y) in the following manner:

If two images are connected on input, then $Dst(x, y) = SrcA(x, y)$ And Not $(SrcB(x, y))$.

If the input **Image Src B** is not connected, then $Dst(x, y) = SrcA(x, y)$ And Not ($Constant$).

IMAQ Mask

Recopies the **Image Src** into the **Image Dst**. If a pixel value is 0 (OFF) in the **Image Mask**, all corresponding pixels in **Image Dst** are reset to 0.



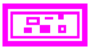


Image Src is the reference to the source (input) image.

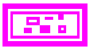


Image Mask is an 8-bit image that specifies the region in the image to modify. The pixels in the original image whose equivalent pixels in the Image Mask have a value of 0 are set to 0. All other pixels keep their original value. The complete image is modified if **Image Mask** is not connected. For more information about an image mask, refer to the *Regions of Interest* section in Chapter 4, *Display VIs*.

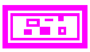


Image Dst is the reference to the destination image. If it is connected, it must be the same type as the **Image Src**.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

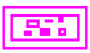
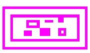


Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.



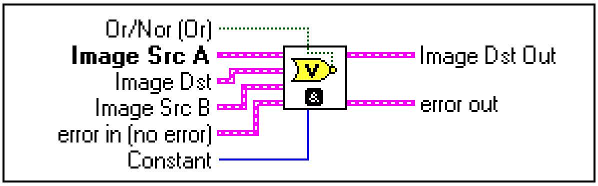
error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

The **Image Mask** contents are considered to be binary. All pixel values other than zero are highlighted and all pixel values of 0 are turned off. **Image Mask** must be an 8-bit image if it is different than the **Image Src**. **Image Dst** must be the same image type as **Image Src**.

IMAQ Or

Performs an OR or NOR operation on two images or an image and a constant.

8 16 RGB HSL



Or/Nor (Or) is the result from a logic operation. If set to TRUE, the result of the OR is inverted, giving a NOR. The default is FALSE, which specifies an OR operation.

Image Src A is the reference to the source (input) image A.

Image Dst is the reference to the destination image. If it is connected, it must be the same type as the **Image Src A**.

Image Src B is the reference to the source (input) image B.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Constant is a binary constant used for image-constant operations. The default is 0.

Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src A**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

All connected images must be the same image type. An operation between an image and a constant occurs when the input **Image Src B** is not connected.

This VI is performed for each pixel (*x*, *y*) in the following manner:

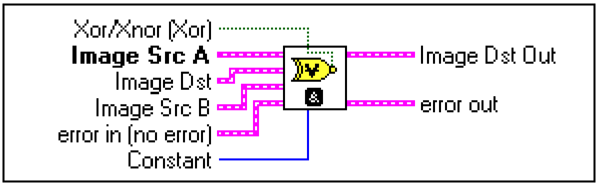
If two images are connected on input, then *Dst(x, y)* = *SrcA(x, y)* OR *SrcB(x, y)*.

If the input **Image Src B** is not connected, then *Dst(x, y)* = *SrcA(x, y)* OR *Constant*.

IMAQ Xor

Performs an XOR or XNOR operation on two images or an image and a constant.

8 16 RGB HSL



Xor/Xnor (Xor) is the result from a logic operation. If set to TRUE, the result of the XOR is inverted, giving an XNOR. The default is FALSE, which specifies an XOR operation.

Image Src A is the reference to the source (input) image A.

Image Dst is the reference to the destination image. If it is connected, it must be the same type as the **Image Src A**.

Image Src B is the reference to the source (input) image B.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Constant is a binary constant used for image-constant operations. The default is 0.

Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src A**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

All connected images must be the same image type. An operation between an image and a constant occurs when the input **Image Src B** is not connected.

This VI is performed for each pixel (*x*, *y*) in the following manner:

If two images are connected on input, then $Dst(x, y) = SrcA(x, y)$ XOR $SrcB(x, y)$.

If the input **Image Src B** is not connected, then $Dst(x, y) = SrcA(x, y)$ XOR *Constant*.

8

Processing VIs

This chapter describes the Processing VIs in IMAQ Vision.

IMAQ AutoBThreshold

Automatically thresholds an image into two classes based on the requested statistical method.

8

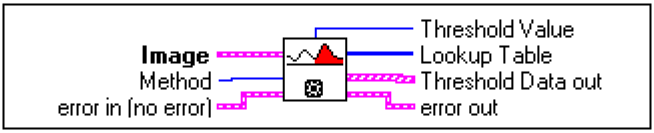


Image is the reference to the source (input) image.

Method is the threshold method used. The following values are valid:

0 **clustering**

1 **entropy**

2 **metric**

3 **moments**

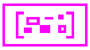
4 **inter-class variance**

Note See the *Thresholding* section of Chapter 7, *Morphology Analysis*, in the *IMAQ Vision User Manual* for more information about these methods.

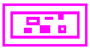
error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Threshold Value outputs the threshold value. This value can be directly connected to **Lower value** from IMAQ Threshold, provided that 255 is connected to **Upper value**.

Lookup Table outputs a look-up table containing 256 elements encoded in 0 and 1. If the threshold value is 160, the values between 0 and 159 become 0 and the values between 160 and 255 become 1. IMAQ UserLookup can use this value directly.



Threshold Data out outputs an array containing two clusters compatible with IMAQ MultiThreshold. The elements in this array define a set of intervals equivalent to the LUT output by **Lookup Table**.



error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

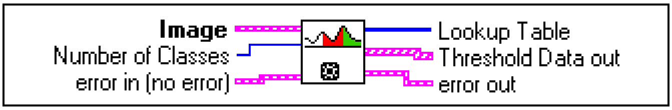
The VI outputs the threshold data in three forms:

- The threshold data directly (**Threshold Value**)
- A LUT directly usable by IMAQ UserLookup
- An array directly usable by IMAQ MultiThreshold (**Threshold Data**)

IMAQ AutoMThreshold

Applies an automatic multi-threshold by using a variant of the classification by clustering method. Starting from a random sort, a clustering algorithm is iterated until a stable and reliable result is found.

8



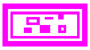
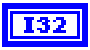
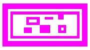


Image is the reference to the source (input) image.



Number of Classes is the number of desired phases. This algorithm uses a clustering method and can use any value between 2 and 256. The default is 2.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Lookup Table is an array containing the values of the 256 transformed elements encoded between 0 and (n – 1), where n is the **Number of Classes**. This array can be connected directly to IMAQ UserLookup.

Threshold Data out outputs an array containing the **Number of Classes** compatible with IMAQ MultiThreshold. The results range from 0 to (n – 1), where n is the **Number of Classes**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

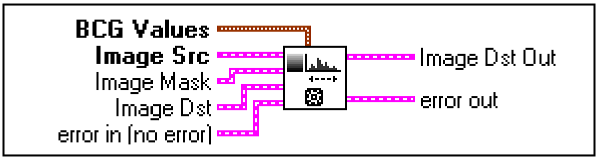
This method is based on an iterated measurement of an histogram. After finding the best result (a very rapid process), the histogram is segmented into n groups. These groups are based on the fact that each point in a group is closer to the *barycenter* of its own group than the other group. The VI outputs the threshold data in two forms:

- A LUT directly usable by IMAQ UserLookup
- An array directly usable by IMAQ MultiThreshold (**Threshold Data**)

IMAQ BCGLookup

Applies a brightness, contrast, and gamma correction to an image. The correction is performed by computing and applying a look-up table. **Brightness**, **Contrast**, and **Gamma** control the changes made to the transfer function represented by the look-up table.

8



BCG Values is a cluster of the following three values to adjust:

Brightness (default = 128) sets the brightness of the image. This value is used as the X intercept of the transfer function in the look-up table. The neutral value is 128 (no change in the image). A higher value returns a brighter image. A value less than 128 decreases the overall brightness of the image.

Contrast (default = 45.0) sets the contrast of the image. This control is used as the slope of the transfer function in the look-up table and is expressed in degrees. A slope of 45 degrees is neutral. A higher value returns a more contrasted image. A value smaller than 45 decreases the contrast of the image.

Gamma (default = 1.0) sets the gamma correction applied to the image. The neutral value is 1. A value greater than 1 gives extended contrast for small pixel values and less contrast for large pixel values. A value smaller than 1 returns an image with less contrast for small pixel values and extended contrast for large pixel values.

Image Src is the reference to the source (input) image.

Image Mask is an 8-bit image that specifies the region in the image to modify. Only those pixels in the original image that correspond to an equivalent pixel in the mask different than 0 are replaced by the values in the look-up table. All other pixels keep their original value. The complete image is modified if **Image Mask** is not connected.

Image Dst is the reference to the destination image. If it is connected, it must be the same type as the **Image Src**.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

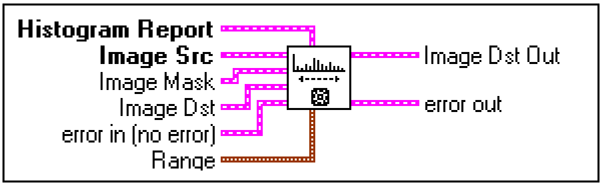
Image Dst Out is the reference to the destination (output) image that receives the processing result of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise **Image Dst Out** refers to the image referenced by **Image Src**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ Equalize

Produces a histogram equalization of an image. This VI redistributes the pixel values of an image to linearize the accumulated histogram. Execute IMAQ Histogram before this VI to supply **Histogram Report** as input. The precision of the VI is dependent on the histogram precision, which in turn is dependent on the number of classes used in the histogram.

8 16 F



Histogram Report is the histogram from the source image. This histogram is supplied from the output of the IMAQ Histogram VI. Because no processing occurs if this input is not connected, you need to connect the same image to both IMAQ Histogram and this VI.

Image Src is the reference to the source (input) image.

Image Mask is an 8-bit image that specifies the region in the image to modify. Only those pixels in the original image that correspond to an equivalent pixel in the mask different than 0 are replaced by the values in the look-up table. All other pixels keep their original value. The complete image is modified if **Image Mask** is not connected.

Image Dst is the reference to the destination image. If it is connected, it must be the same type as the **Image Src**.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Range is a cluster containing the minimum and maximum values for the range to equalize. The equalization of the entire image occurs if this cluster is not connected or the defaults 0 and 0 are used as input. In this case, the **Minimal Value** and **Maximal Value** contained in **Histogram Report** are considered to be the minimum and maximum. The default is (0, 0).

The following elements are specified in this cluster:

Minimum is the smallest value used for processing. After processing, all pixel values that are less than or equal to the **Minimum** in the original image are set to 0 for an 8-bit image. In 16-bit and floating-point images, these pixel values are set to the smallest pixel value found in the original image.

Maximum is the largest value used for processing. After processing, all pixel values that are greater than or equal to the **Maximum** in the original image are set to 255 for an 8-bit image. In 16-bit and floating-point images, these pixel values are set to the largest pixel value found in the original image.

Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

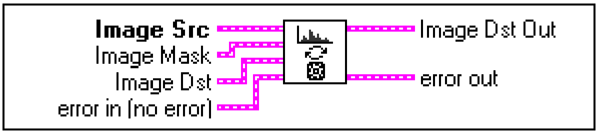
Note The modification to the pixel value depends on the histogram contents, regardless of the image type used. All pixels entering into the same histogram class have an identical value after equalization.

IMAQ Inverse

Inverts the pixel intensities of an image to compute the negative of an image.

For 8-bit images, this VI performs an 8-bit inversion of the pixel intensities in the [0, 255] range. For 16-bit and floating-point image types, the inversion is performed within the minimum and maximum intensity values found in the image.

8 16 F



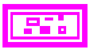


Image Src is the reference to the source (input) image.

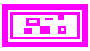


Image Mask is an 8-bit image that specifies the region in the image to modify. Only those pixels in the original image that correspond to an equivalent pixel in the mask different than 0 are processed. All other pixels keep their original value. The complete image is modified if **Image Mask** is not connected.

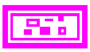
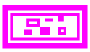


Image Dst is the reference to the destination image. If it is connected, it must be the same type as the **Image Src**.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Image Dst Out is the reference to the destination (output) image that receives the processing result of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise **Image Dst Out** refers to the image referenced by **Image Src**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ Label

Labels the particles in a binary image.

8 16

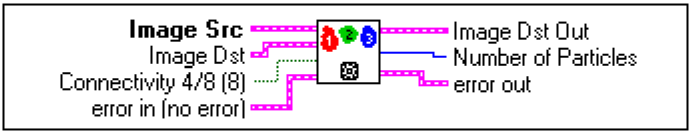


Image Src is the reference to the source (input) image.

Image Dst is the reference to the destination image. If it is connected, it must be the same type as the **Image Src**.

Connectivity 4/8 (8) specifies the connectivity used for particle detection. The default is 8.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.

Number of Particles indicates the number of particles detected in the image.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

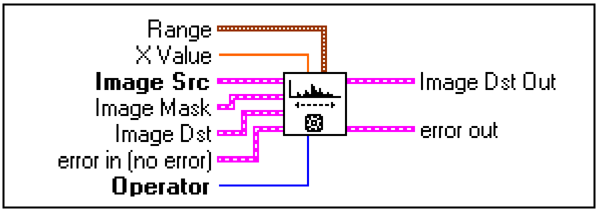
This operation applies a color to all pixels composing the same group of pixels (a particle). This color level is encoded in 8 or 16 bits, depending on the image type. Therefore, 255 particles can be labeled in an 8-bit image and 65,535 particles in a 16-bit image. If you want to label more than 255 particles in an 8-bit image, you need to perform a threshold operation with an interval of [255, 255] after processing the first 254 particles. The goal of this threshold operation is to eliminate the first 254 particles in order to visualize the next 254 particles.

Note This operation requires that **Image Src** and **Image Dst** be the same image type and that the border for these images be greater or equal to 2.

IMAQ MathLookup

Converts the pixel values of an image by replacing them with values from a defined look-up table. This VI modifies the dynamic range of either part of an image or the complete image, depending on the type of curve chosen.

8 **16** **F**



Range is a cluster containing the minimum and maximum values for the range to modify. The dynamic range of the entire image is modified if this cluster is not connected or if the defaults 0 and 0 are used as input. The dynamic range of the destination image is dependent on the type of input image. The dynamic range for an 8-bit image is between 0 and 255. The dynamic range for 16-bit and floating-point images is the smallest and largest pixel value contained in the original image before processing. The default is (0, 0).

Note The dynamic range for 16-bit and floating-point images is not modified. Only the distribution of the values is changed.

The following elements are specified in the **Range** cluster:

Minimum is the smallest value used for processing. After processing, all pixel values that are less than or equal to **Minimum** in the original image are set to 0 for an 8-bit image. In 16-bit and floating-point images, these pixel values are set to the smallest pixel value found in the original image.

Maximum is the largest value used for processing. After processing, all pixel values that are greater than or equal to **Maximum** in the original image are set to 255 for an 8-bit image. In 16-bit and floating-point images, these pixel values are set to the largest pixel value found in the original image.

X Value is a value used only for the operators Power X and Power 1/X.

Image Src is the reference to the source (input) image.

Image Mask is an 8-bit image that specifies the region in the image to modify. Only those pixels in the original image that correspond to an equivalent pixel in the mask different than 0 are replaced by the values in the look-up table. All other pixels keep their original value. The complete image is modified if **Image Mask** is not connected.

Image Dst is the reference to the destination image. If it is connected, it must be the same type as the **Image Src**.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Operator specifies the remapping procedure used. The horizontal axis represents the pixel values before processing (between **Minimum** and **Maximum**), and the vertical axis represents the pixel values after processing. The default is 0, which specifies linear remapping.

Linear	Linear remapping.
Log	A logarithmic remapping operation that gives extended contrast for small pixel values and less contrast for large pixel values.
Exp	An exponential remapping operation that gives extended contrast for large pixel values and less contrast for small pixel values.
Square	Similar to exponential but with a more gradual effect.
Square Root	Similar to logarithmic but with a more gradual effect.
Power X	Gives variable effects depending on the value of *X*. The default value of *X* is 1.5.
Power 1/X	Gives variable effects depending on the value of *X*. The default value of *X* is 1.5.

Note For an 8-bit image, the minimum is always 0 and the maximum is always 255. For 16-bit and floating-point images, the minimum and maximum are the endpoint values found in the image before processing.

Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ MultiThreshold

Performs thresholds of multiple intensity ranges to an image.

8 16 F

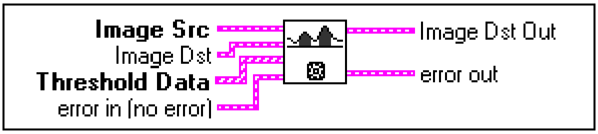


Image Src is the reference to the source (input) image.

Image Dst is the reference to the destination image. If it is connected, it must be the same type as the **Image Src**.

Threshold Data is an array of clusters specifying the mode and threshold range. This operation is analogous to the process in IMAQ Threshold. Each cluster is composed of the following elements:

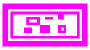
Lower value is the lowest pixel value to be taken into account during a threshold. The default is 128.

Upper value is the highest pixel value to be taken into account during a threshold. The default is 255.

Replace Value is the value used to replace pixels between the **Lower value** and **Upper value**. This operation requires that **Keep/Replace Value (Replace)** is TRUE.

Keep/Replace Value (Replace) determines whether to replace the value of the pixels existing in the range between **Lower value** and **Upper value**. The default TRUE replaces these pixel values and the status FALSE keeps the original values.

All pixels outside the range between **Lower value** and **Upper value** are set to 0. All values found between this range are replaced by the value entered in **Replace Value** if **Keep/Replace Value (Replace)** is TRUE.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

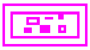
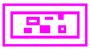


Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.



error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

The threshold operations are performed in the order that the data is received from **Threshold Data**. A pixel can be taken into account only once, even if the pixel is included in the threshold range of two different thresholds by **Threshold Data**.

For example, a VI contains two clusters on input:

Cluster 1	**Lower value** = 80, **Upper value** = 150, **Keep/Replace Value** = TRUE, **Replace Value** = 255.
Cluster 2	**Lower value** = 120, **Upper value** = 200, **Keep/Replace Value** = FALSE.

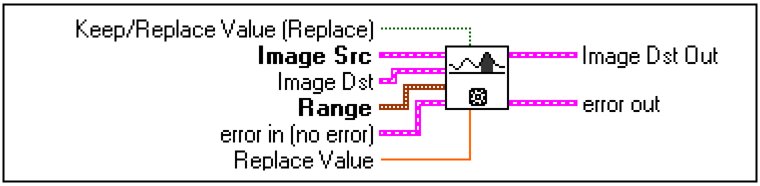
This example shows two threshold ranges with an overlap between 120 and 150. Therefore, the pixels between 120 and 150 are treated only by the first threshold. The following results occur after execution of this VI:

- Pixel values between 0 and 79 are replaced by 0.
- Pixel values between 80 and 150 are replaced by 255.
- Pixel values between 151 and 200 keep their original values.
- Pixel values greater than 200 are set to 0.

IMAQ Threshold

Applies a threshold to an image.

8 16 F



Keep/Replace Value (Replace) determines whether to replace the value of the pixels existing in the range between **Lower value** and **Upper value**. The default TRUE replaces these pixel values, and the status FALSE keeps the original values.

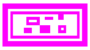


Image Src is the reference to the source (input) image.

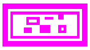


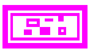
Image Dst is the reference to the destination image. If it is connected, it must be the same type as the **Image Src**.

Range is a cluster specifying the threshold range. It is composed of the following elements:

Lower value is the lowest pixel value used during a threshold. The default is 128.

Upper value is the highest pixel value used during a threshold. The default is 255.

All pixels outside the range between **Lower value** and **Upper value** are set to 0. All values found between this range are replaced by the value entered in **Replace Value** if **Keep/Replace Value (Replace)** is TRUE.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Replace Value is the value used to replace pixels between the **Lower value** and **Upper value**. This operation requires that **Keep/Replace Value (Replace)** is TRUE.

Note Use a binary palette when you plan to visualize an image to which a threshold has been applied in Replace mode. However, which palette to use for visualization depends on the value of **Replace Value**. For example, the visualization of a threshold image could be performed with a gray palette. However, in this case it is advised that you use a replacement value of 255 (white) to see the threshold image better.

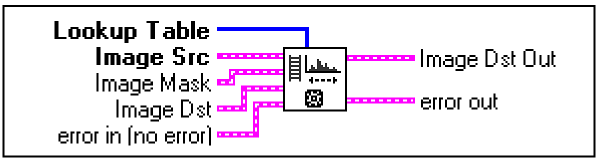
Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ UserLookup

Performs a user-chosen look-up table transformation by remapping the pixel values in an image.

8 16



[I16] **Lookup Table** is a grayscale replacement table. This array can contain 256 elements (8-bit) or 65,536 elements (16-bit) depending on the type of image. Individual pixels within the image are not modified in cases in which the look-up table is missing a corresponding value.

Image Src is the reference to the source (input) image.

Image Mask is an 8-bit image that specifies the region in the image to modify. Only those pixels in the original image that correspond to an equivalent pixel in the mask different than 0 are replaced by the values in the look-up table. All other pixels keep their original value. The complete image is modified if **Image Mask** is not connected.

Image Dst is the reference to the destination image. If it is connected, it must be the same type as the **Image Src**.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

9

Filter VIs

This chapter describes the Filter VIs in IMAQ Vision. The filters are divided into two types: linear (also called *convolution*) and nonlinear.

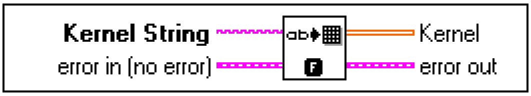
A convolution is a special algorithm that consists of recalculating the value of a pixel based on its own pixel value and the pixel values of its neighbors. The sum of this calculation is divided by the sum of the elements in the matrix to obtain a new pixel value. The size of the *convolution matrix* (or *kernel*) does not have a theoretical limit and can be either square or rectangular (3 × 3, 5 × 5, 5 × 7, 9 × 3, 127 × 127, and so on). Convolutions are divided into four families: gradient, Laplacian, smoothing, and Gaussian. This grouping is determined by the convolution matrix contents or the weight assigned to each pixel, which depends on the geographical position of that pixel in relation to the central matrix pixel.

IMAQ Vision features a set of standard convolution kernels for each family and for the usual sizes (3 × 3, 5 × 5, and 7 × 7). These convolution kernels are accessible from the IMAQ GetKernel VI. You also can create your own kernels and choose what to put into them. The size of the user-defined kernel is virtually unlimited. With this capability, you can create special-effect filters.

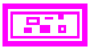
The purpose of the nonlinear filters is to either extract the contours (edge detection) or remove the isolated pixels. The IMAQ EdgeDetection VI has six different methods you can use for contour extraction (Differentiation, Gradient, Prewitt, Roberts, Sigma, or Sobel). The IMAQ CannyEdgeDetection VI is a specialized edge detection method that locates edges accurately even under low signal-to-noise conditions in an image. To harmonize pixel values, you can choose between two VIs, each of which uses a different method: IMAQ NthOrder and IMAQ LowPass. These VIs require that either a kernel size and order number (IMAQ NthOrder) or percentage (IMAQ LowPass) is specified on input.

IMAQ BuildKernel

Constructs a convolution matrix by converting a string. This string can represent either integers or floating-point values.

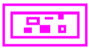


Kernel String is a string listing the coefficients that form the matrix.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Kernel is the resulting matrix converted from the input string. You can connect this output directly to the input **Kernel** in IMAQ Convolute.



error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

The column separator can be either a comma, a semicolon, or a blank space. The line separator is a linefeed character. For example, the string `1 1 1 1 1 1 1 1 1` produces a 3×3 matrix with all coefficients set to `1` as follows:

$$\begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}$$

IMAQ CannyEdgeDetection

Uses a specialized edge detection method to accurately estimate the location of edges even under conditions of poor signal-to-noise ratios.

8

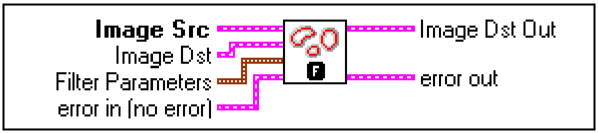


Image Src is the image reference source.

Image Dst is the reference of the image destination. If it is connected, it must be the same type as the **Image Src**.

Filter Parameters is a cluster containing the following filter parameters:

SGL **Sigma** is the sigma of the gaussian smoothing filter that is applied to the image before the edge detection is done.

SGL **H Thresh** (High Threshold) defines the upper percentage of pixel values in the image from which the seed or starting point of an edge segment is chosen. It takes values from 0 to 1.

SGL **L Thresh** (Low Threshold) is multiplied by the **H Thresh** value to define a lower threshold for all the pixels in an edge segment.

I32 **Window Size** defines the size of the gaussian filter that is applied to the image. This size must be odd.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

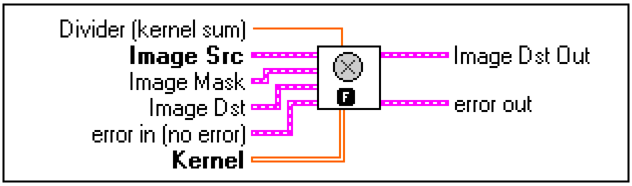
Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ Convolute

Filters an image using a linear filter. The calculations are performed with either integers or floating points, depending on the image type and the contents of the kernel.

8 16 F



Divider (kernel sum) is a normalization factor that can be applied to the sum of the obtained products. Under normal conditions the divider should not be connected. If connected and not equal to 0, the elements internal to the matrix are summed and then divided by this normalization factor.

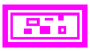


Image Src is the image reference source. It must be an 8-bit or RGB image.

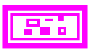


Image Mask is an 8-bit image that specifies the region in the image to modify. Only those pixels in the original image that correspond to an equivalent pixel in the mask different than 0 are replaced by the values in the convolution process. All other pixels keep their original value. The complete image is modified if **Image Mask** is not connected.

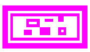
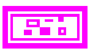


Image Dst is the reference of the image destination. If it is connected, it must be the same type as the **Image Src**.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Kernel is a 2D array that contains the convolution matrix to be applied to the image. The size of the convolution is fixed by the size of this array. The array can be generated by using standard G programming techniques or the IMAQ GetKernel or IMAQ BuildKernel VIs. If the kernel contains fewer than three rows or three columns, no convolution is performed.

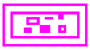
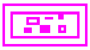


Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.



error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Any image connected to the input **Image Dst** must be the same image type connected to **Image Src**. The image type connected to the input **Image Mask** must be an 8-bit image.

The connected source image must have been created with a border capable of supporting the size of the convolution matrix. A 3 × 3 matrix must have a minimum border of 1, a 5 × 5 matrix must have a minimum border of 2, and so on. The border size of the destination image is not important.

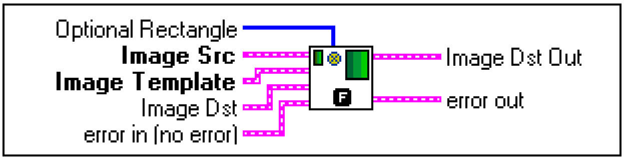
A convolution matrix must have odd-sized dimensions so that it contains a central pixel. The function does not take into account the odd boundary, farthest out on the matrix, if one of the **Kernel** dimensions is even. For example, if the input **Kernel** is 6 × 4 (X = 6 and Y = 4), the actual convolution is 5 × 3. Both the sixth line and the fourth are ignored.

Calculations using an 8-bit or 16-bit **Image Src** input are made in integer mode provided that the kernel contains only integers. Calculations using a 32-bit floating-point **Image Src** input are made in floating-point mode. Notice that the processing speed corresponds to the size of the kernel. A 3 × 3 convolution processes nine pixels and a 5 × 5 convolution processes 25 pixels.

IMAQ Correlate

Computes the normalized cross correlation between the source image and the template image.

8



[I32] **Optional Rectangle** defines an array containing the coordinates (Left/Top/Right/Bottom) of the region in the source image that is used for the correlation process. Correlation is applied to the entire image if the input is empty or not connected.

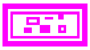


Image Src is a reference to the source image. The normalized cross correlation is performed between this image and the template image. This image must be an 8-bit image.

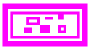


Image Template is a reference to a template image. This image must be an 8-bit image. For the correlation, the center of the template image is used as the origin.

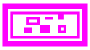
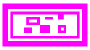


Image Dst is the reference of the image destination. If it is connected, it must be the same type as the **Image Src**.


error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

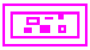
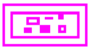


Image Dst Out is an 8-bit image that contains the cross-correlation values normalized to lie in the range [0, 255]. A value of 255 indicates a very high correlation and a value of 0 indicates no correlation.

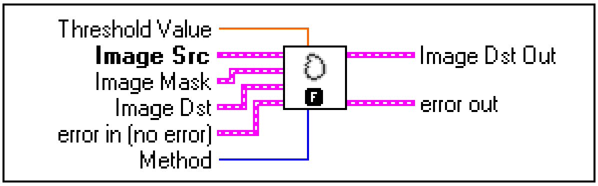

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Note Correlation is a computer-intensive operation. On an MMX machine, you can reduce the time required to perform a correlation by keeping the x size of the **Image Template** at a multiple of 4, by keeping the template size small, and by using the optional rectangle to reduce the search area in the source image.

IMAQ EdgeDetection

Extracts the contours (detects edges) in gray-level values. Any image connected to the input **Image Dst** must be the same image type connected to **Image Src**. The image type connected to the input **Image Mask** must be an 8-bit image. The connected source image must have been created with a border capable of supporting the size of the processing matrix. For example, a 3 × 3 matrix has a minimum border size of 1. The border size of the destination image is not important.

8 16 F



Threshold Value is the minimum pixel value to appear in the resulting image. It is rare to use a value greater than 0 for this type of processing because the results from this processing are usually very dark and are not very dynamic. The default is 0.

Image Src is the image reference source.

Image Mask is an 8-bit image that specifies the region in the image to modify. Only those pixels in the original image that correspond to an equivalent pixel in the mask different than 0 are replaced by the values of the edge detection process. All other pixels keep their original value. The complete image is modified if **Image Mask** is not connected.

Image Dst is the reference of the image destination. If it is connected, it must be the same type as the **Image Src**.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Method specifies the type of edge-detection filter to use. The following filters are available:

Differentiation	(Default) Processing with a 2 × 2 matrix
Gradient	Processing with a 2 × 2 matrix
Prewitt	Processing with a 3 × 3 matrix
Roberts	Processing with a 2 × 2 matrix
Sigma	Processing with a 3 × 3 matrix
Sobel	Processing with a 3 × 3 matrix

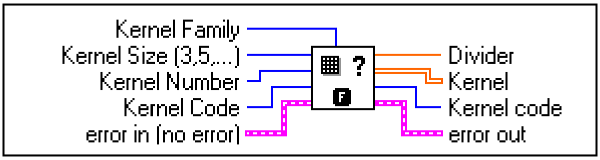
Note See the *Nonlinear Filters* section of Chapter 5, *Spatial Filtering*, in the *IMAQ Vision User Manual* for more information about these filters.

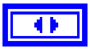
Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ GetKernel

Reads a predefined kernel. This code consists of three separate units: **Kernel Family**, **Kernel Size**, and **Kernel Number**. If you already know the code, you can enter it directly with **Kernel Code**.





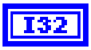
Kernel Family determines the type of matrix. This value corresponds to the thousandth unit in the researched code.

Gradient

Laplacian

Smoothing

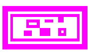
Gaussian



Kernel Size (3,5,...) determines the horizontal and vertical matrix size. The values are 3, 5, and 7, corresponding to the convolutions 3 × 3, 5 × 5, and 7 × 7 supplied in the matrix catalog. This value corresponds to the hundredth unit in the researched code.

Kernel Number is the matrix family number. It is a two-digit number, between 0 and *n*, belonging to a family and a size. A number of predefined matrices are available for each type and size.

Kernel Code is a code you can use to directly access a convolution matrix. Each code specifies a specific convolution matrix. You can use this input if it is connected and is not 0. The kernel located in the file then is transcribed into a 2D array that is available from the output **Kernel**. You can use the codes to specify a predefined kernel.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Divider is the normalization factor associated with the retrieved kernel.

Kernel is the resulting matrix. It corresponds to a kernel encoded by a code specified from the inputs **Kernel Family**, **Kernel Size**, and **Kernel Number** or a from a code directly passed through the input **Kernel Code**. You can connect this output directly to the input **Kernel** in IMAQ Convolute.

Kernel code indicates the code that was used to retrieve the kernel.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Example

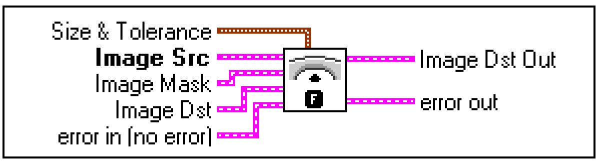
For the kernel code 1300, the kernel family is gradient, the kernel size is 3 × 3, and the kernel number (*nn*) is 00. The matrix is as follows:

$$\begin{array}{ccc} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{array}$$

IMAQ LowPass

Calculates the inter-pixel variation between the pixel being processed and those pixels surrounding it. If the pixel being processed has a variation greater than a specified percentage, it is set to the average pixel value as calculated from the neighboring pixels.

8 16 F



Size & Tolerance is a cluster that specifies the following variables:

X Size is the size of the horizontal matrix axis. The default is 3.

Y Size is the size of the vertical matrix axis. The default is 3.

% Tolerance is the maximum variation allowed. The default is 40 percent.

Image Src is the image reference source.

Image Mask is an 8-bit image that specifies the region in the image to modify. Only pixels in the original image that correspond to an equivalent pixel in the mask different than 0 are replaced by the values of the lowpass process. All other pixels keep their original value. The complete image is modified if **Image Mask** is not connected.

Image Dst is the reference of the image destination. If it is connected, it must be the same type as the **Image Src**.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Note See the *Nonlinear Filters* section of Chapter 5, *Spatial Filtering*, in the *IMAQ Vision User Manual* for more information about the lowpass filter.

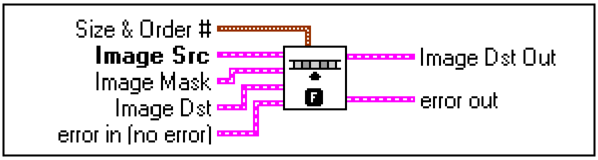
Any image connected to the input **Image Dst** must be the same image type connected to **Image Src**. The image type connected to the input **Image Mask** must be an 8-bit image.

The connected source image must have been created with a border capable of supporting the size of the convolution matrix. A 3 × 3 matrix must have a minimum border of 1, a 5 × 5 matrix must have a minimum border of 2, and so on. The border size of the destination image is not important.

IMAQ NthOrder

Orders, or classifies, the pixel values surrounding the pixel being processed. The data is placed into an array and the pixel being processed is set to the *N*th pixel value, the *N*th pixel being the ordered number.

8 16 F



Size & Order # is a cluster that specifies the following variables. The default values correspond to a 3 x 3 median filter.

I32 **X Size** is the size of the horizontal matrix axis. The default is 3.

I32 **Y Size** is the size of the vertical matrix axis. The default is 3.

I32 **Order #** is the order number chosen after classing the values. The default is 4.

Image Src is the image reference source.

Image Mask is an 8-bit image that specifies the region in the image to modify. Only those pixels in the original image that correspond to an equivalent pixel in the mask different than 0 are replaced by the values of the Nth order process. All other pixels keep their original value. The complete image is modified if **Image Mask** is not connected.

Image Dst is the reference of the image destination. If it is connected, it must be the same type as the **Image Src**.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Note See the *Nonlinear Filters* section of Chapter 5, *Spatial Filtering*, in the *IMAQ Vision User Manual* for more information about the Nth order filter.

Any image connected to the input **Image Dst** must be the same image type connected to **Image Src**. The image type connected to the input **Image Mask** must be an 8-bit image.

The connected source image must have been created with a border capable of supporting the size of the convolution matrix. A 3 × 3 matrix must have a minimum border of 1, a 5 × 5 matrix must have a minimum border of 2, and so on. The border size of the destination image is not important.

The default for this VI is a 3 × 3 *Median operation* with X = 3, Y = 3, and **Order** = 4. To change to a 5 × 5 median operation, the cluster must take the values X = 5, Y = 5, and **Order** = 12. In this last example, the order number is determined by calculating the central pixel number in the array. For a 5 × 5 convolution, **Order** = 12 (the 13th pixel) because that pixel is the center pixel number for a 2D array of 25 pixels.

A lighter image results when using a higher-order number (such as 7 in a 3 × 3 matrix). Darker images result when using a lower-order number (such as 1 in a 3 × 3 matrix).

A median (center-pixel) operation is advantageous because it standardizes the gray-level values without significantly modifying the form of the objects or the overall brightness in the image.

If the order value that is entered is 0, the image obtained is representative of the local minimum from the source image. This operation is equivalent to a gray morphology erosion.

If the order value that is passed is equal to [(X Size × Y Size) – 1], the obtained image is representative of the local maximum from the source image. This operation is equivalent to a gray morphology dilation.

10

Morphology VIs

This chapter describes the Morphology VIs in IMAQ Vision. The morphological transformations fall into two groups: gray-level morphology and binary morphology.

Introduction

In *gray-level* morphology, a pixel is compared to those pixels surrounding it to keep those pixel values that are the smallest (erosion) or the largest (dilation). The VIs responsible for binary morphological transformations accept only 8-bit images while the VI for gray-level morphological transformations (IMAQ GrayMorphology) accepts 8-bit, 16-bit, or 32-bit floating-point images.

In *binary* morphology, the pixels are considered to exist in one of two states. The pixels are present (for pixel values other than 0) or absent (for pixel values equal to 0). The two levels of binary processing available—primary and advanced—either activate or deactivate pixels.

An image is considered to be binary after it has undergone a threshold (using such VIs as IMAQ Threshold, IMAQ AutoBThreshold, and so on). Binary morphology is divided into two groups in IMAQ Vision. A single VI (IMAQ Morphology) performs all the primary operations, such as erosions, dilations, openings, closings, and contour extractions. The advanced operations are performed by multiple VIs, each responsible for a single type of operation. These operations can separate particles, remove either small or large particles, keep or remove filter particles according to morphological parameters, fill holes in particles, remove particles that touch the boundary of the image border, and create the skeleton of particles.

Morphological transformations use an object known as a *structuring element*, which helps you control the effect of the functions on the shape and the boundary of object. In IMAQ Vision, the structuring element is a 2D array that specifies, by its size and contents, which pixels the process has to take into account to determine the new value of a pixel. A structuring element must have a center pixel and therefore must have an odd-sized axis. The contents of the structuring element are also considered to be binary (0 or not 0). The most often used structuring element is 3 × 3 and contains only values of 1. This is usually the default model for binary and gray-level morphological transformations. You need at least a basic understanding of structuring elements before experimenting with user-chosen sizes and contents. The majority of the VIs for advanced morphology do not have a structuring element input because only the standard 3 × 3 default is useful.

The connected source image for a morphological transformation must have been created with a border capable of supporting the size of the structuring element. A 3 × 3 structuring element requires a minimal border of 1, a 5 × 5 structuring element requires a minimal border of 2, and so on.

The input **Square/Hexa** is available for certain VIs that perform morphological transformations. This concept introduces a variable for the perception of an *image frame* (aligned or shifted), which influences whether to include pixels in the processing. Figure 10-1 illustrates the difference between a 3 × 3 and 5 × 5 structuring element in a square frame and a hexagonal frame.

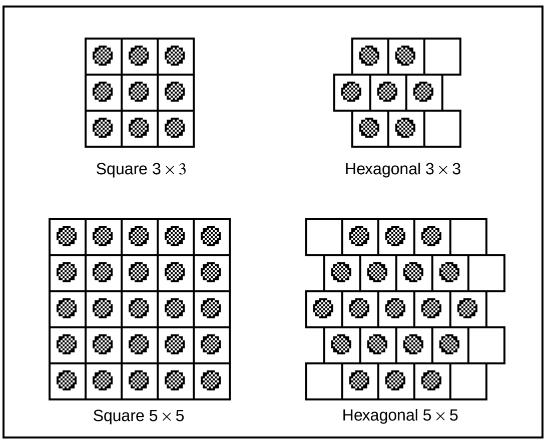


Figure 10-1. Using the Square/Hexa Input Parameter

Hexagonal mode does not use the elements [2, 0] and [2, 2] from the 3 × 3 structuring element. The same holds true for the elements [0, 0], [4, 0], [4, 1], [4, 3], [0, 4] and [4, 4] if the transformation uses a 5 × 5 structuring element.

The advanced morphology VIs IMAQ RemoveParticle, IMAQ RejectBorder, IMAQ FillHole, and IMAQ Particle Filter use the input **Connectivity 4/8** (default is 8) to determine whether a neighboring pixel is considered to be part of same particle.

Figure 10-2 illustrates the difference between Connectivity 4 and Connectivity 8.

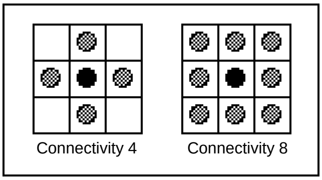


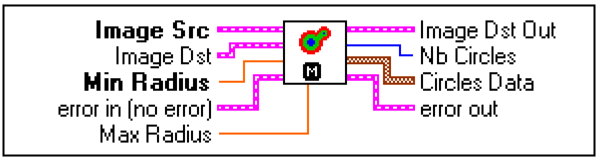
Figure 10-2. Using the Connectivity 4/8 Input Parameter

For more information about morphology, refer to Chapter 7, *Morphology Analysis*, in the *IMAQ Vision User Manual*.

IMAQ Circles

Separates overlapping circular objects and classifies them based on their radius, surface area, and perimeter. Starting from a binary image, it finds the radius and center of the circular objects even when multiple circular objects overlap. In addition, this VI can trace the circles in the destination image. It constructs and uses a Danielsson distance map to determine the radius of each object.

8



Note **IMAQ Circles** works correctly only for circles that have a radius less than or equal to 256 pixels.

Image Src is the reference to the source (input) image.

Image Dst is the reference to the destination image. If it is connected, it must be the same type as the **Image Src**.

Min Radius specifies the smallest radius, in pixels, to detect. Circles with a radius smaller than this value do not appear in the destination image and have a negative radius value in the output **Circles Data**. The default is 1.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Max Radius specifies the largest radius, in pixels, to detect. Circles with a radius larger than this value do not appear in the destination image and have a negative radius value in the output **Circles Data**. The default is 10.

Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, then **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.

Nb Circles returns the number of detected circles in the image.

Note This number also includes circles with a radius outside the limits of **Min Radius** or **Max Radius**.

Circles Data returns an array of measurements for all detected circles. Each element in the array has a structure containing the following elements:

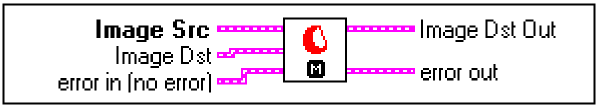
- **Pos. X** is the horizontal position, in pixels, of the center of the circle.
- **Pos. Y** is the vertical position, in pixels, of the center of the circle.
- **Radius** is the radius of the circle in pixels. Circles with a radius outside the limits of **Min Radius** or **Max Radius** are returned with negative radius values.
- **Core Area** is the surface area, in pixels, of the nucleus of the circle as defined by the Danielsson distance map.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ Convex

Calculates a convex envelope for particles that are labeled in an image. Execute IMAQ Label to label the objects in the image before using this VI.

8 16



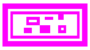


Image Src is the reference to the source (input) image.

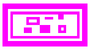
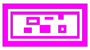


Image Dst is the reference to the destination image. If it is connected, it must be the same type as the **Image Src**.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

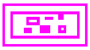
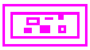


Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, then **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.

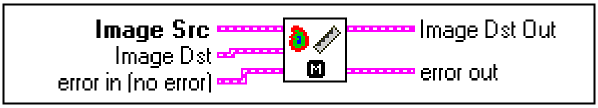


error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ Danielsson

Returns a distance map based on the algorithms of Danielsson. The *Danielsson distance map* produces images and data that are similar to IMAQ Distance but are much more accurate. In most cases it is recommended that you use this function instead of IMAQ Distance.

8



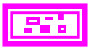


Image Src is the reference to the source (input) image.

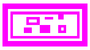
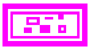


Image Dst is the reference to the destination image. If it is connected, it must be the same type as the **Image Src**.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

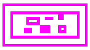
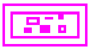


Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, then **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.



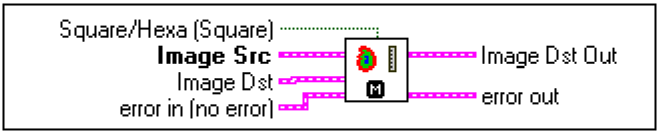
error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ Distance

Encodes a pixel value of a particle as a function of the location of that pixel in relation to the distance to the border of the particle. The source image must have been created with a border size of at least 1 and must be an 8-bit binary image.

Note In most cases it is recommended that you use IMAQ Danielsson instead of IMAQ Distance because it produces more accurate results.

8



Square/Hexa (Square) specifies whether to treat the pixel frame as square or hexagonal during the transformation. The default is square.

Image Src is the reference to the source (input) image.

Image Dst is the reference to the destination image. If it is connected, it must be the same type as the **Image Src**.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

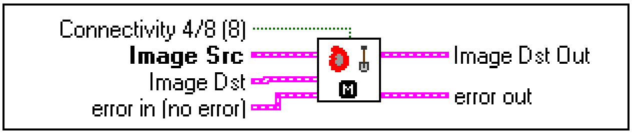
Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, then **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ FillHole

Fills the holes found in a particle. The holes are filled with a pixel value of 1. The source image must be an 8-bit binary image. The holes found in contact with the image border are never filled because it is impossible to determine whether these holes are part of a particle.

8



Connectivity 4/8 (8) specifies how the algorithm determines whether an adjacent pixel belongs to the same or to a different particle. The default is **Connectivity 8**.

Image Src is the reference to the source (input) image.

Image Dst is the reference to the destination image. If it is connected, it must be the same type as the **Image Src**.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

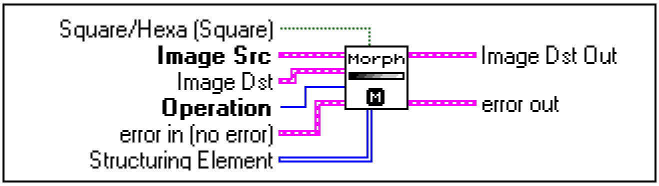
Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, then **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ GrayMorphology

Performs grayscale morphological transformations. All source and destination image types must be the same. The connected source image for a morphological transformation must have been created with a border capable of supporting the size of the structuring element. A 3 × 3 structuring element requires a minimal border of 1, a 5 × 5 structuring element requires a minimal border of 2, and so on. The border size of the destination image is not important.

8 16 F



Square/Hexa (Square) specifies whether to treat the pixel frame as square or hexagonal during the transformation. The default is square.

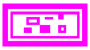


Image Src is the reference to the source (input) image.

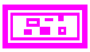
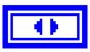


Image Dst is the reference to the destination image. If it is connected, it must be the same type as the **Image Src**.



Operation specifies the type of morphological transformation procedure to use. The default is **AutoM**. You can choose from the following values:

AutoM	(Default) Auto median
Close	Dilation followed by an erosion
Dilate	Dilation
Erode	Erosion
Open	Erosion followed by a dilation
PClose	A succession of seven closings and openings
POpen	A succession of seven openings and closings

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Structuring Element is a 2D array that contains the structuring element to apply to the image. The size of the structuring element (the size of this array) determines the processing size. A structuring element of 3 × 3 is used if this input is not connected.

Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, then **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.

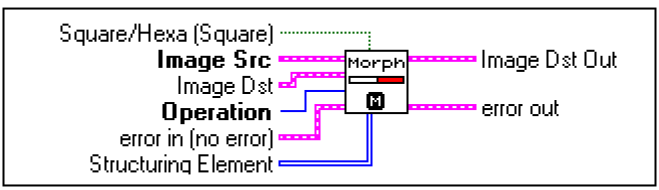
error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

A structuring element must have odd-sized dimensions so that it contains a central pixel. The function does not take into account the odd boundary, farthest out on the matrix, if one of the dimensions for the structuring element is even. For example, if the input structuring element is 6 × 4 (X = 6 and Y = 4), the actual processing is performed at 5 × 3. Both the sixth line and the fourth row are ignored. The processing speed is correlated with the size of the structuring element. For example, a 3 × 3 structuring element processes nine pixels, and a 5 × 5 structuring element processes 25 pixels.

IMAQ Morphology

Performs primary morphological transformations. All source images must be 8-bit binary images. The connected source image for a morphological transformation must have been created with a border capable of supporting the size of the structuring element. A 3 × 3 structuring element requires a minimal border of 1, a 5 × 5 structuring element requires a minimal border of 2, and so on. The border size of the destination image is not important.

8



Square/Hexa (Square) specifies whether to treat the pixel frame as square or hexagonal during the transformation. The default is square.

Image Src is the reference to the source (input) image.

Image Dst is the reference to the destination image. If it is connected, it must be the same type as the **Image Src**.

Operation specifies the type of morphological transformation procedure to use. The default is **AutoM**. You can choose from the following values:

AutoM	(Default) Auto median
Close	Dilation followed by an erosion
Dilate	Dilation (the opposite of an erosion)
Erode	Erosion that eliminates isolated background pixels
Gradient	Extraction of internal and external contours of a particle
Gradient out	Extraction of exterior contours of a particle
Gradient in	Extraction of interior contours of a particle
Hit miss	Elimination of all pixels that do not have the same pattern as found in the structuring element
Open	Erosion followed by a dilation
PClose	A succession of seven closings and openings
POpen	A succession of seven openings and closings
Thick	Activation of all pixels matching the pattern in the structuring element
Thin	Activation of all pixels matching the pattern in the structuring element

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Structuring Element is a 2D array that contains the structuring element to apply to the image. The size of the structuring element (the size of this array) determines the processing size. A structuring element of 3 × 3 is used if this input is not connected.

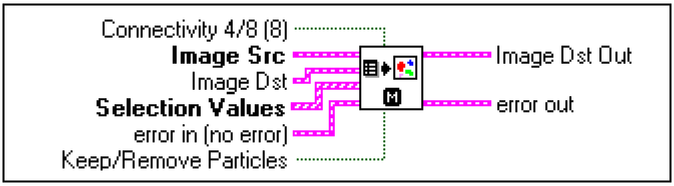
Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, then **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

A structuring element must have odd-sized dimensions so that it contains a central pixel. The function does not take into account the odd boundary, farthest out on the matrix, if one of the dimensions for the structuring element is even. For example, if the input structuring element is 6 × 4 (X = 6 and Y = 4), the actual processing is performed at 5 × 3. Both the sixth line and the fourth row are ignored. The processing speed is correlated with the size of the structuring element. For example, a 3 × 3 structuring element processes nine pixels, and a 5 × 5 structuring element processes 25 pixels.

IMAQ Particle Filter

Filters (keeps or removes) particles in an image according to their morphological measurements.



Connectivity 4/8 (8) specifies the type of connectivity used by the algorithm for particle detection. The connectivity mode directly determines whether an adjacent pixel belongs to the same particle or a different particle. The default value is Connectivity 8. The following values are possible:

TRUE	**Connectivity 8**	(Default) Particle detection is performed in connectivity mode 8.
FALSE	**Connectivity 4**	Particle detection is performed in connectivity mode 4.

Image Src is the reference to the source (input) image.

Image Dst is the reference to the destination image. If it is connected, it must be the same type as the **Image Src**.

Selection Values controls the criteria that will be used to filter the particle in the image. This control is made of an array of clusters composed of the following values:

Parameter selects the measurement.

Lower Value specifies the lower value of the range for the chosen parameter.

Upper Value specifies the upper value of the range for the chosen parameter.

Interval specifies if the process selects particles inside or outside the range.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Keep/Remove Particles sets the action that is performed on the objects filling the criteria specified by the control **Selection Values**. TRUE removes the particles filling the criteria. FALSE keeps the objects filling the criteria.

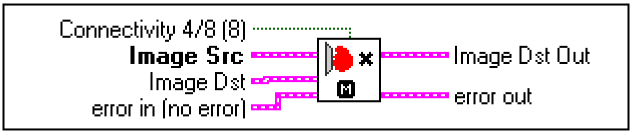
Image Dst Out is the reference to the destination (output) image that receives the processing result of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise **Image Dst Out** refers to the image referenced by **Image Src**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ RejectBorder

Eliminates particles that touch the border of an image. The source image must be an 8-bit binary image.

8



Connectivity 4/8 (8) specifies how the algorithm determines whether an adjacent pixel belongs to the same or to a different particle. The default is Connectivity 8.

Image Src is the reference to the source (input) image.

Image Dst is the reference to the destination image. If it is connected, it must be the same type as the **Image Src**.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

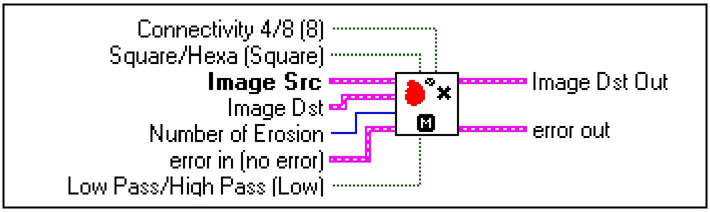
Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, then **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ RemoveParticle

Eliminates or keeps particles resistant to a specified number of 3 × 3 erosions. The particles that are kept are exactly the same as those found in the original source image. The source image must be an 8-bit binary image.

8



Connectivity 4/8 (8) specifies how the algorithm determines whether an adjacent pixel belongs to the same or to a different particle. The default is Connectivity 8.

Square/Hexa (Square) specifies whether to treat the pixel frame as square or hexagonal during the transformation. The default is square.

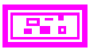


Image Src is the reference to the source (input) image.

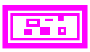


Image Dst is the reference to the destination image. If it is connected, it must be the same type as the **Image Src**.

Number of Erosion specifies the number of 3 × 3 erosions to apply to the image. The default is 2.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Low Pass/High Pass (Low) specifies whether the objects resistant to *n* erosions are discarded or kept (default).

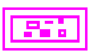


Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ Segmentation

Starting from a labeled image, calculates the zones of influence between particles. Each labeled particle grows until the particles reach their neighbors, at which time this growth is stopped. The source image must have a border greater than or equal to 1.

8 **16**

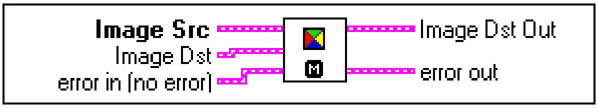


Image Src is the reference to the source (input) image.

Image Dst is the reference to the destination image. If it is connected, it must be the same type as the **Image Src**.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

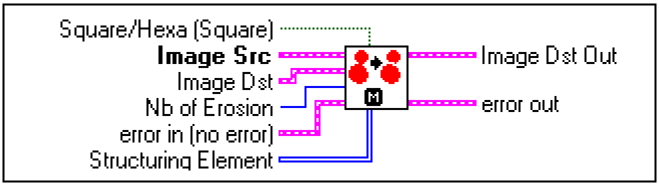
Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ Separation

Separates touching particles, particularly small isthmuses found between particles. It performs *n* erosions (*n* = **Nb of Erosion**) and then reconstructs the final image based on the results of the erosion. If an existing isthmus is broken or removed during the erosion process, the particles are reconstructed without the isthmus. The reconstructed particles, however, have the same size as the initial particles except that they are separated. If no isthmus is broken during the erosion process, the particles are reconstructed as they were initially found. No change is made. The source image must be an 8-bit binary image. The source image must have a border greater than or equal to 1.

8



Square/Hexa (Square) specifies whether to treat the pixel frame as square or hexagonal during the transformation. The default is square.

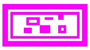


Image Src is the reference to the source (input) image.

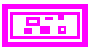
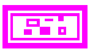
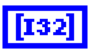


Image Dst is the reference to the destination image. If it is connected, it must be the same type as the **Image Src**.

Nb of Erosion specifies the number of erosions that are used to separate the particles. The default is 1.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.



Structuring Element is a 2D array that contains the structuring element to apply to the image. The size of the structuring element (the size of this array) determines the processing size. A structuring element of 3 × 3 is used if this input is not connected.

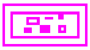
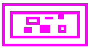


Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.

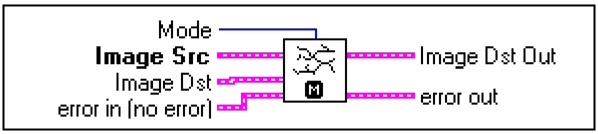


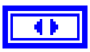
error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ Skeleton

Starting from a binary image, calculates the skeletons of the particles within an image or the lines delineating the zones of influence of the objects (skeleton of an inverse image). The source image must have a border greater than or equal to 1.

8





Mode specifies the type of skeleton to perform. The default is Skeleton L.

Skeleton L uses this type structuring element:

0	?	1
0	1	1
0	?	1

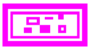
Skeleton M uses this type structuring element:

?	?	1
0	1	1
?	?	1

Skiz is an inverse skeleton (Skeleton L on an inverse image).

Image Src is the reference to the source (input) image.

Image Dst is the reference to the destination image. If it is connected, it must be the same type as the **Image Src**.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

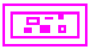
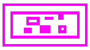


Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.



error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

11

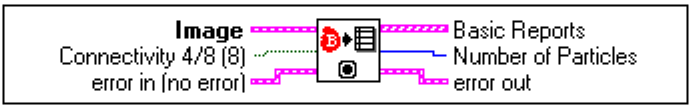
Analysis VIs

This chapter describes the Analysis VIs in IMAQ Vision. These VIs return information about the contents of the image and particles in the image.

IMAQ BasicParticle

Detects and measures particles. This VI returns the area and position of particles in a binary image.

8



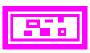


Image is the input source image used for calculating the matrices. The image must be binary. A particle consists of pixels that do not contain a 0 value. The source image must have been created with a border size of at least 2.

Connectivity 4/8 (8) specifies the type of connectivity used by the algorithm for particle detection. The connectivity mode directly determines whether an adjacent pixel belongs to the same particle or a different particle. The default is Connectivity 8. The following values are possible:

TRUE	**Connectivity 8**	(Default) Particle detection is performed in connectivity mode 8.
FALSE	**Connectivity 4**	Particle detection is performed in connectivity mode 4.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Basic Reports is an array that returns a set of measurements from the detected particles. This cluster contains the following elements:

- **Area (pixels)** indicates the surface area of a particle in number of pixels.
- **Area (calibrated)** indicates the surface area of a particle in user-defined units.
- **Global Rectangle** is a cluster that contains the coordinates of a bounding rectangle for the object detected in the image. This cluster includes the following parameters:
  - **x1Left** indicates the x coordinate of the top-left corner of the rectangle.
  - **y1Top** indicates the y coordinate of the top-left corner of the rectangle.
  - **x2Right** indicates the x coordinate of the bottom-right corner of the rectangle.
  - **y2Bottom** indicates the y coordinate of the bottom-right corner of the rectangle.

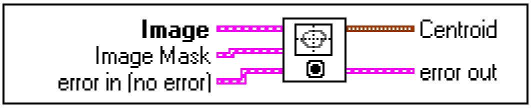
Number of Particles returns the number of pixels detected in a particle.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ Centroid

Computes the center of the image.

8 16 F



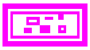


Image is the reference to the image whose centroid has to be calculated.

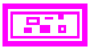
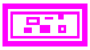
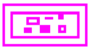


Image Mask is an 8-bit image specifying the region in the image to use for calculating a *centroid*. Only those pixels in the original image that correspond to an equivalent pixel in the mask different than 0 are used for calculating the centroid. A centroid on the complete image is computed if the **Image Mask** is not connected.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Centroid is a cluster containing the X and Y coordinates of the centroid of the image.

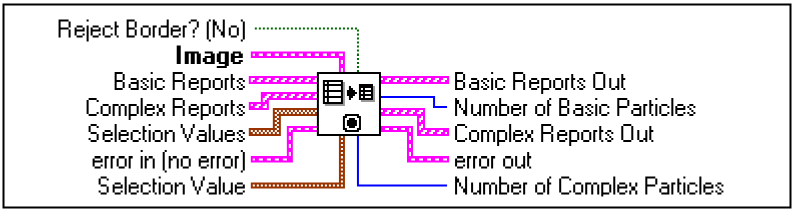


error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ ChooseMeasurements

Returns a selection of particle measurements obtained from IMAQ BasicParticle or IMAQ ComplexParticle based on a minimum and maximum criteria. With this VI, you choose which measurements you want to obtain from a particle-detection process.

8



Reject Border? (No) determines whether particles touching the border should be measured. If set to TRUE, the measurements for particles touching the border are rejected. In this case the input image source must be connected to the input **Image**. The default is FALSE.

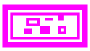
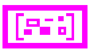
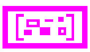


Image is the same input source image that is used to measure the particle coefficients by IMAQ BasicParticle or IMAQ ComplexParticle. This input is used only in a case in which particles touching the border are discarded for measurement calculations (**Reject Border?** is set to TRUE).



Basic Reports is the output array of measurements from IMAQ BasicParticle. Refer to the *IMAQ BasicParticle* description for information on the measurements stored in each element of this array.



Complex Reports is the output array of measurements from IMAQ ComplexParticle. Refer to the *IMAQ ComplexParticle* description for information on the measurements stored in each element of this array.

Selection Values is an array of selection criteria. Each criteria is composed of the following elements:

Parameter is an indicator that determines the coefficient (measurement) to be selected. **Parameter** can have values compatible to those described in IMAQ ComplexMeasure. The validity of these values depends on the type of measurements passed as input (for example, through **Basic Reports** or **Complex Reports**).

Note Only the particle measurements that respond to the selection criteria are selected. The coefficient values must be contained in the interval between **Lower Value** and **Upper Value**.

The following values are possible for selecting basic measurements from **Basic Reports**:

Area (pixels)	Surface area of particle in pixels
Area (calibrated)	Surface area of particle in user units
Left column (X)	Left X coordinate of bounding rectangle
Upper row (Y)	Top Y coordinate of bounding rectangle
Right column (X)	Right X coordinate of bounding rectangle
Lower row (Y)	Bottom Y coordinate of bounding rectangle

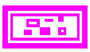
The following values are possible for selecting complex measurements from **Complex Reports**:

Area (pixels)	Surface area of particle in pixels
Area (calibrated)	Surface area of particle in user units
Number of holes	Number of holes
Hole's area (pixels)	Surface area of the holes in pixels
Left column (X)	Left X coordinate of bounding rectangle
Upper row (Y)	Top Y coordinate of bounding rectangle
Right column (X)	Right X coordinate of bounding rectangle
Lower row (Y)	Bottom Y coordinate of bounding rectangle
Longest segment length	Length of longest horizontal line segment
Longest segment left column (X)	Left-most X coordinate of longest horizontal line

Longest segment top row (Y)	Y coordinate of longest horizontal line segment
Perimeter	Length of outer contour of particle
Hole's Perimeter	Perimeter of all holes
SumX	Sum of the X-axis for each pixel of the particle
SumY	Sum of the Y-axis for each pixel of the particle
SumXX	Sum of the X-axis squared for each pixel of the particle
SumYY	Sum of the Y-axis squared for each pixel of the particle
SumXY	Sum of the X-axis and Y-axis for each pixel of the particle
Corrected projection x	Projection corrected in x
Corrected projection y	Projection corrected in y

Lower Value is the minimum value (boundary) for the values to be selected.

Upper Value is the maximum value (boundary) for the values to be selected.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Selection Value is a selection criteria. This value is used only if the array of selection criteria is not connected to **Selection Values**. The selection criteria possess the same structure as each element in the array **Selection Values**. The default value for **Parameter** is –1, which specifies that all measurements are made (no selection).

Basic Reports Out is an output containing an array of the basic measurements selected.

Number of Basic Particles is an output containing the number of basic measurements selected.

Complex Reports Out is an output containing an array of the complex measurements selected.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Number of Complex Particles is an output containing the number of complex measurements selected.

IMAQ ComplexMeasure

Calculates the coefficients of all detected particles. This VI returns an array of coefficients whose measurements are based on the results sent from IMAQ ComplexParticle.

8

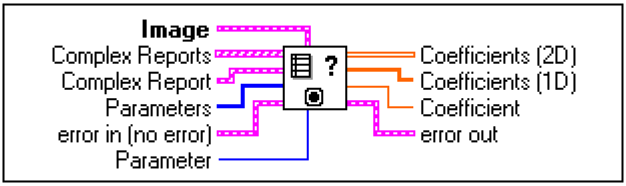


Image is the same input source image that is used to measure the particle coefficients by IMAQ ComplexParticle.

Complex Reports is the output array of measurements from IMAQ ComplexParticle. Refer to the *IMAQ ComplexParticle* description for information on the measurements stored in each element of this array.

Complex Report is an extraction of the output array of measurements from IMAQ ComplexParticle. Refer to the *IMAQ ComplexParticle* description for information on the measurements stored in each element of this array. This input is used only in a case in which **Complex Reports** is not connected and specifies that the measurements are to be made on a single particle.

Parameters is an array specifying a descriptor list of the coefficients that you want to calculate. You can calculate one or more coefficients for one or more particles. The descriptor list is described in the table for the **Parameter** control.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Parameter is an array specifying a descriptor list of the coefficients that you want to calculate. You can calculate one or more coefficients for one or more particles. This input is used only in a situation in which the input **Parameters** is not connected. You can choose from the following descriptor values:

Area (pixels)	Surface area of particle in pixels
Area (calibrated)	Surface area of particle in user units
Number of Holes	Number of holes
Hole's Area	Surface area of the holes in user units
Total Area	Total surface area (holes and particles) in user units
Scanned Area	Surface area of the entire image in user units
Ratio Area/Scanned Area %	Percentage of the surface area of a particle in relation to the scanned area
Ratio Area/Total Area %	Percentage of a particle's surface area in relation to the total area
Center of Mass (X)	X coordinate of the center of gravity
Center of Mass (Y)	Y coordinate of the center of gravity
Left Column (X)	Left X coordinate of bounding rectangle
Upper Row (Y)	Top Y coordinate of bounding rectangle
Right Column (X)	Right X coordinate of bounding rectangle

Lower Row (Y)	Bottom Y coordinate of bounding rectangle
Width	Width of bounding rectangle in user units
Height	Height of bounding rectangle in user units
Longest Segment Length	Length of longest horizontal line segment
Longest Segment Left Column (X)	Left-most X coordinate of longest horizontal line segment
Longest Segment Row (Y)	Y coordinate of longest horizontal line segment
Perimeter	Length of outer contour of particle in user units
Hole's Perimeter	Perimeter of all holes in user units
SumX	Sum of the X-axis for each pixel of the particle
SumY	Sum of the Y-axis for each pixel of the particle
SumXX	Sum of the X-axis squared for each pixel of the particle
SumYY	Sum of the Y-axis squared for each pixel of the particle
SumXY	Sum of the X-axis and Y-axis for each pixel of the particle
Corrected projection X	Projection corrected in *x*
Corrected projection Y	Projection corrected in *y*
Moment of inertia Ixx	Inertia matrix coefficient in *xx*
Moment of inertia Iyy	Inertia matrix coefficient in *yy*
Moment of inertia Ixy	Inertia matrix coefficient in *xy*

Mean chord X	Mean length of horizontal segments
Mean chord Y	Mean length of vertical segments
Max intercept	Length of longest segment
Mean intercept perpendicular	Mean length of the chords in an object perpendicular to its max intercept
Particle orientation	Direction of the longest segment
Equivalent ellipse minor axis	Total length of the axis of the ellipse having the same area as the particle and a major axis equal to half the max intercept
Ellipse major axis	Total length of major axis having the same area and perimeter as the particle in user units
Ellipse minor axis	Total length of minor axis having the same area and perimeter as the particle in user units
Ratio of equivalent ellipse axis	Fraction of major axis to minor axis
Rectangle big side	Length of the large side of a rectangle having the same area and perimeter as the particle in user units
Rectangle small side	Length of the small side of a rectangle having the same area and perimeter as the particle in user units
Ratio of equivalent rectangle sides	Ratio of rectangle big side to rectangle small side
Elongation factor	Max intercept/mean perpendicular intercept
Compactness factor	Particle area/(breadth × width)

Heywood circularity factor	Particle perimeter/perimeter of circle having same area as particle
Type Factor	A complex factor relating the surface area to the moment of inertia
Hydraulic Radius	Particle area/particle perimeter
Waddel disk diameter	Diameter of the disk having the same area as the particle in user units
Diagonal	Diagonal of an equivalent rectangle in user units

Coefficients (2D) is a 2D array containing the specified measurements. This array is used only when you have specified multiple coefficients (measurements) for each particle. The data is stored by particle followed by the coefficients.

Coefficients (1D) is a 1D array containing the specified measurements. This array is used only when you have specified either multiple coefficients (measurements) for a single particle or a single coefficient for multiple particles.

Coefficient is the measurement specified for a single particle.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

The output from this VI can be in one of three forms: **Coefficients (2D)**, **Coefficients (1D)**, or **Coefficient**. The final type of output depends on the connected inputs, as shown in Table 11-1.

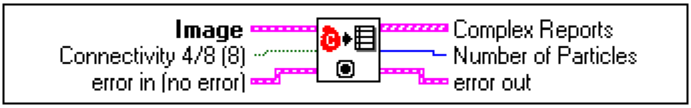
Table 11-1. IMAQ ComplexMeasure Outputs

Possible Inputs	Resulting Type of Output
Complex Reports and Parameters	Coefficients (2D)
Complex Reports and Parameter	Coefficients (1D)
Complex Report and Parameters	Coefficients (1D)
Complex Report and Parameter	Coefficient

IMAQ ComplexParticle

Detects and measures particles. This VI returns a set of measurements made from particles in a binary image.

8



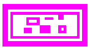
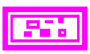


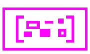
Image is the input source image used for calculating the matrices. The image must be binary. A particle consists of pixels that do not contain a 0 value. The source image must have been created with a border size of at least 2.

Connectivity 4/8 (8) specifies the type of connectivity used by the algorithm for particle detection. The connectivity mode directly determines whether an adjacent pixel belongs to the same particle or a different particle. The default is Connectivity 8. The following values are possible:

TRUE	**Connectivity 8**	(Default) Particle detection is performed in connectivity mode 8.
FALSE	**Connectivity 4**	Particle detection is performed in connectivity mode 4.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.



Complex Reports is an array that returns a set of measurements from the detected particles. This cluster contains the following elements:

Area (pixels) indicates the surface area of a particle in number of pixels.

Area (calibrated) indicates the surface area of a particle in user-defined units.

Perimeter is the perimeter size in user units.

Number of Holes is the number of holes in the particle.

Hole's Area (pixels) is the total surface area of all the holes in a particle, in pixels.

Hole's Perimeter is the total perimeter size calculated from all the holes in a particle, in user-defined units.

Global Rectangle is a cluster that contains the coordinates of a bounding rectangle for the object detected in the image. This cluster includes the following parameters:

- **x1Left** indicates the x coordinate of the top-left corner of the rectangle.
- **y1Top** indicates the y coordinate of the top-left corner of the rectangle.
- **x2Right** indicates the x coordinate of the bottom-right corner of the rectangle.
- **y2Bottom** indicates the y coordinate of the bottom-right corner of the rectangle.

Σ x is the sum of the X-axis for each pixel of the particle.

Σ y is the sum of the Y-axis for each pixel of the particle.

Σ xx is the sum of the X-axis squared for each pixel of the particle.

Σ xy is the sum of the X-axis and Y-axis for each pixel of the particle.

Σ yy is the sum of the Y-axis squared for each pixel of the particle.

Longest Segment Length is the longest segment length of the particle.

Longest Segment Coordinates are the coordinates of the left-most pixel in the **Longest Segment Length** of the particle. The top-most segment coordinates are used in a case in which more than one **Longest Segment Length** exist. This cluster contains the following parameters:

x is the X-axis coordinate of the pixel the farthest left in the **Longest Segment Length** in the particle.

y is the Y-axis coordinate of the pixel the farthest left in the **Longest Segment Length** in the particle.

Projection x is half the sum of the horizontal segments in a particle that do not overlap another adjacent horizontal segment.

Projection y is half the sum of the vertical segments in a particle that do not overlap another adjacent vertical segment.

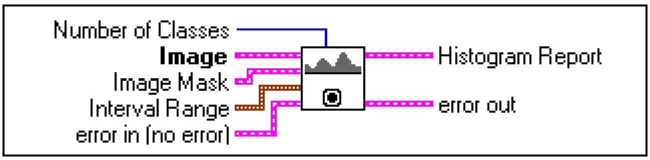
Number of Particles returns the number of detected particles.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ Histogram

Calculates the histogram of an image.

8 16 F



Number of Classes specifies the number of classes used to classify the pixels. The number of obtained classes differs from the specified amount in a case in which the minimum and maximum boundaries are overshot in the **Interval Range.** It is advised to specify an even number of classes (for example, 2, 4, or 8) for 8-bit or 16-bit images. The default value is 256, which is designed for 8-bit images. This value gives a uniform class distribution or one class for each pixel in an 8-bit image.

Image is the input source image used for calculating the histogram.

Image Mask is an 8-bit image specifying the region in the image to use for calculating a histogram. Only those pixels in the original image that correspond to an equivalent pixel in the mask different than 0 are used for calculating the histogram. A histogram on the complete image occurs if the **Image Mask** is not connected.

Interval Range is a cluster specifying the minimum and maximum boundaries for the histogram calculation. Only those pixels having a value that falls in this range are taken into account by the histogram calculation. This cluster is composed of the following elements:

Minimum is the minimum interval value. The default value of (0, 0) ensures that the real minimum value is determined by the source image, as described in the following table.

Image Type	Minimum Value Used
8	(0, 0)
16	Minimum pixel value found in the image
F	Minimum pixel value found in the image

Maximum is the maximum interval value. The default value of (0, 0) ensures that the real maximum value is determined by the source image, as described in the following table.

Image Type	Maximum Value Used
8	255
16	Maximum pixel value found in the image
F	Maximum pixel value found in the image

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Histogram Report is a cluster that returns the histogram values. This cluster contains the following elements:

Histogram returns the histogram values in an array. The elements found in this array are the number of pixels per class. The *n*th class contains all pixel values belonging to the interval $[(\textit{Starting Value} + (n - 1) \times \textit{Interval Width}),$ $(\textit{Starting Value} + n \times (\textit{Interval Width} - 1))]$.

Minimal Value returns the smallest pixel value used in calculating the histogram.

Maximal Value returns the largest pixel value used in calculating the histogram.

Starting Value returns the smallest pixel value from the first class calculated in the histogram. It can be equal to the **Minimum** value from the **Interval Range** or the smallest value found for the image type connected.

Interval Width returns the length of each class.

Mean Value returns the mean value of the pixels used in calculating the histogram.

Standard Deviation returns the standard deviation from the histogram. A higher value corresponds to a better distribution of the values in the histogram and the image.

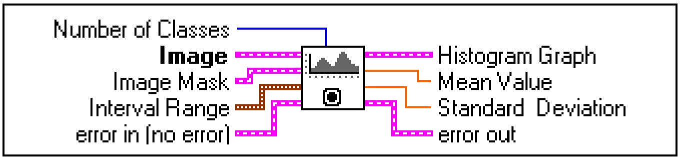
Area (pixels) returns the number of pixels used in the histogram calculation. This is influenced by the values specified in **Interval Range** and the contents of **Image Mask**.

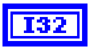
error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ Histograph

Calculates the histogram from an image. This VI returns a data type (cluster) compatible with a LabVIEW or BridgeVIEW graph.

8 **16** **F**





Number of Classes specifies the number of classes used to classify the pixels. The number of obtained classes differs from the specified amount in a case in which the minimum and maximum boundaries are overshot in the **Interval Range**. You are advised to specify an even number of classes (for example, 2, 4, or 8) for 8-bit or 16-bit images. The default value is 256, which is designed for 8-bit images. This value gives a uniform class distribution or one class for each pixel in an 8-bit image.

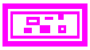


Image is the input source image used for calculating the histogram.

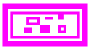


Image Mask is an 8-bit image specifying the region in the image to use for calculating a histogram. Only those pixels in the original image that correspond to an equivalent pixel in the mask different than 0 are used for calculating the histogram. A histogram on the complete image occurs if the **Image Mask** is not connected.

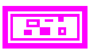
Interval Range is a cluster specifying the minimum and maximum boundaries for the histogram calculation. Only those pixels having a value that falls in this range are taken into account by the histogram calculation. This cluster is composed of the following elements:

Minimum is the minimum interval value. The default value of (0, 0) ensures that the real minimum value is determined by the source image, as described in the following table.

Image Type	Minimum Value Used
8	(0)
16	Minimum pixel value found in the image
F	Minimum pixel value found in the image

Maximum is the maximum interval value. The default value of (0, 0) ensures that the real maximum value is determined by the source image, as described in the following table.

Image Type	Maximum Value Used
8	255
16	Maximum pixel value found in the image
F	Maximum pixel value found in the image



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Histogram Graph is a cluster that returns the histogram values. This cluster contains the following elements:

Starting Value returns the smallest pixel value from the first class calculated in the histogram. It can be equal to the **Minimum** value from the **Interval Range** or the smallest value found for the image type connected.

Incremental Value returns the incrementing value that specifies how much to add to **Starting Value** in calculating the median value of each class from the histogram. The median value x_n from the *n*th class is expressed as follows: $x_n = \textit{Starting Value} + n \times \textit{Incremental Value}$.

Histogram returns the histogram values in an array. The elements found in this array are the number of pixels per class. The *n*th class contains all pixel values belonging to the interval $[(\textit{Starting Value} + (n - 1) \times \textit{Interval Width}), (\textit{Starting Value} + n \times (\textit{Interval Width} - 1))]$.

Mean Value returns the mean value of the pixels used in calculating the histogram.

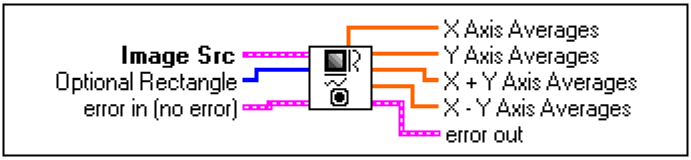
Standard Deviation returns the standard deviation from the histogram. The higher this value, the better the distribution of the values in the histogram and the image.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ LinearAverages

Computes the average pixel intensity (mean line profile) on the whole or part of the image.

8 16 F



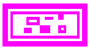
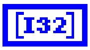
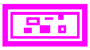


Image Src is the reference to the source (input) image.



Optional Rectangle defines an array (four elements) containing the coordinates (Left/Top/Right/Bottom) of the region to extract. The operation is applied to the entire image if the input is empty or not connected.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

X Axis Averages is the linear average along each column in the image.

Y Axis Averages is the linear average along each row in the image.

X + Y Axis Averages is the linear average along each diagonal running from bottom-left to top-right.

X - Y Axis Averages is the linear average along each diagonal running from top-left to bottom-right.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ LineProfile

Calculates the profile of a line of pixels. This VI returns a data type (cluster) compatible with a LabVIEW or BridgeVIEW graph. The relevant pixel information is taken from the specified vector (line).

8 16 F

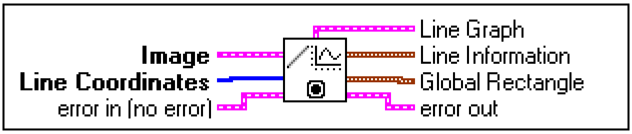


Image is the input source image used for calculating the line profile.

Line Coordinates is an array specifying the pixel coordinates that form the end points of the line.

Note A line with the coordinates [0, 0, 0, 255] is formed from 256 pixels. Any pixels designated by **Line Coordinates** found outside the actual image are set to 0 in **Line Graph**.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Line Graph is a cluster that contains the line profile with an X origin at 0 and an increment of 1. The cluster contains the following elements:

x0 always returns 0.

dx always returns 1.

Pixels Line returns the line profile calculated in an array in which elements represent the pixel values belonging to the specified vector.

Line Information is a cluster containing relevant information about the pixels found in the specified vector. This cluster contains the following elements:

Min returns the smallest pixel value found in the line profile.

Max returns the largest pixel value found in the line profile.

Mean returns the mean value of the pixels found in the line profile.

Var returns the standard deviation from the line profile.

Count found in the line profile.

Global Rectangle is a cluster that contains the coordinates of a bounding rectangle for the line in the image. This cluster includes the following parameters:

x1Left indicates the x coordinate of the top-left corner of the rectangle.

y1Top indicates the y coordinate of the top-left corner of the rectangle.

x2Right indicates the x coordinate of the bottom-right corner of the rectangle.

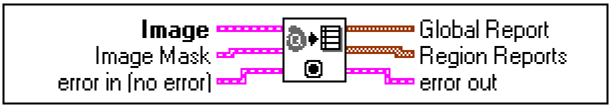
y2Bottom indicates the y coordinate of the bottom-right corner of the rectangle.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ Quantify

Quantifies the contents of an image or the regions within an image. The region definition is performed with a labeled image mask. Each mask has a single unique value.

8 16 F



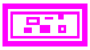


Image is the input source image.

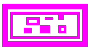
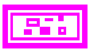


Image Mask is an 8-bit image specifying the regions to quantify in the image. Only those pixels in the original image that correspond to an equivalent pixel in the mask are used for the quantification. Each pixel in this image (mask) indicates, by its value, which region belongs to the corresponding pixel in **Image**. Up to 255 different regions can be quantified directly from the **Image**. A quantification is performed on the complete image if the **Image Mask** is not connected.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Global Report is a cluster containing the quantification data relative to all the regions within an image, or to the entire image if the **Image Mask** is not connected. This cluster contains the following elements:

Mean Value of the pixels.

Standard Deviation of the pixel values indicates the distribution of the values in relation to the average. The higher this value, the better the distribution of the pixel values.

Minimal Value returns the smallest pixel value.

Maximal Value returns the largest pixel value.

Area (calibrated) returns the analyzed surface area in user-units.

Area (pixels) returns the analyzed surface area in pixels.

% returns the percentage of the analyzed surface in relation to the complete image.

Region Reports is a cluster containing the quantification data relative to each region within an image, or the entire image if the **Image Mask** is not connected. The *n*th element in this array contains the data regarding the *n*th region. The size of this array is equal to the largest pixel value in **Image Mask**. The returned data is identical to the data in **Global Report**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

12

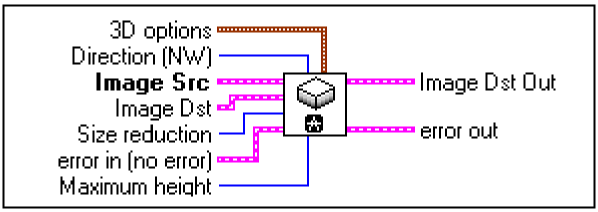
Geometry VIs

This chapter describes the Geometry VIs in IMAQ Vision. You can use these VIs to perform geometrical transformations on an image, such as rotating or transposing an image.

IMAQ 3DView

Displays an image using an isometric view. Each pixel from the image source is represented as a column of pixels in the 3D view. The pixel value corresponds to the altitude.

8 C



3D options is a cluster containing the following elements:

SGL **alpha** defines the angle between the horizontal and the baseline. The value can be between 0° and 45°. The default value is 30°.

SGL **beta** defines the angle between the horizontal and the second baseline. The value can be between 0° and 45°. The default value is 30°.

I32 **border** defines the border size in the 3D view. The default value is 20.

background defines the background color for the 3D view. The default is 85.

plane specifies the view to display if the image is complex. The following four possible planes can be visualized from a complex image. For complex images, the default is the magnitude.

real

imaginary

(Default) magnitude

phase

Direction (NW) defines the viewing orientation shown for the 3D view. The following four viewing angles are possible. The default is North West.

(Default) North West

South West

South East

North East

Image Src is the reference to the source (input) image.

Image Dst must be an 8-bit image.

Size reduction is a factor applied to the source image to calculate the final dimensions of the 3D view image. This factor is a divisor that is applied to the source image when determining the final height and width of the 3D view image. A factor of 1 uses all of the pixels of the source image when determining the 3D view image. A factor of 2 uses every other line and every other column of the pixels of the source image to determine the 3D view image. The default is 2.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Maximum height defines the maximum height of a pixel from the image source that is drawn in 3D. This value is mapped from a maximum of 255 (from the source image) in relation to the baseline in the 3D view. A value of 255, therefore, gives a one-to-one correspondence between the intensity value in the source image and the display in 3D view. The default value of

64 results in a four-fold reduction between the original intensity value of the pixel in the source image and the final displayed 3D image.

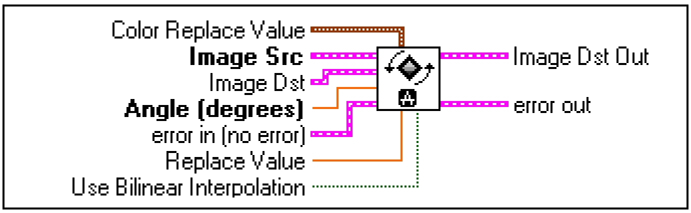
Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ Rotate

Rotates an image.

8 16 F RGB HSL



Color Replace Value is a cluster containing the alpha, red, green, and blue channel values used for filling a color image. The default is 0.

Image Src is the reference to the source (input) image.

Image Dst is the reference of the image destination. If it is connected, it must be the same type as the **Image Src**.

Angle (degrees) defines the angle to rotate. The default is 0°.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Use Bilinear Interpolation specifies the type of interpolation to use to compute the values of the rotated pixels. If set to TRUE (default), IMAQ Rotate uses bilinear interpolation. If set to FALSE, zero-order interpolation is used.

Replace Value defines the filling value created by the rotation. The default is 0.

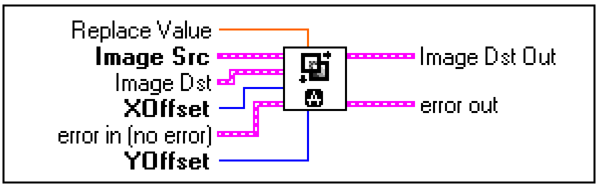
Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ Shift

Translates an image based on a horizontal and vertical offset.

8 16 F RGB HSL



Replace Value defines the filling value created by the shift. The default is 0.

Image Src is the reference to the source (input) image.

Image Dst is the reference of the image destination. If it is connected, it must be the same type as the **Image Src**.

XOffset is the horizontal offset added to the image. The default is 0.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

YOffset is the vertical offset added to an image. The default is 0.

Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ Symmetry

Transforms an image through its symmetry.

8 16 F RGB HSL

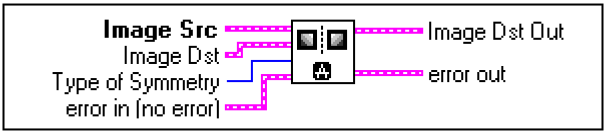


Image Src is the reference to the source (input) image.

Image Dst is the reference of the image destination. If it is connected, it must be the same type as the **Image Src**.

U32

Type of Symmetry specifies the symmetry used. The default is horizontal symmetry. You can choose from the following values:

Horizontal	(Default) Based on the horizontal axis of the image
Vertical	Based on the vertical axis of the image
Central	Based on the center of the image
1st Diagonal	Based on the first diagonal of the image (the image must be square)
2nd Diagonal	Based on the second diagonal of the image (the image must be square)

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

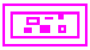
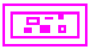


Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.



error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

13

Complex VIs

This chapter describes the Complex VIs.

Frequency processing is another technique for extracting information from an image. Instead of using the location and direction of light-intensity variations, you can use frequency processing to manipulate the frequency of the occurrence of these variations in the spatial domain. This new component is called the *spatial frequency*, which is the frequency with which the light intensity in an image varies as a function of spatial coordinates.

Spatial frequencies of an image are computed with the Fast Fourier transform (FFT). The FFT is calculated in two steps: a 1D transform of the rows, followed by a 1D transform of the columns of the previous results. The complex numbers that compose the FFT plane are encoded in a 64-bit floating-point image called a complex image: 32 bits for the real part and 32 bits for the imaginary part. IMAQ Vision can read and write complex images through IMAQ ReadFile and IMAQ WriteFile using the AIPD file type.

In an image, details and sharp edges are associated with high spatial frequencies because they introduce significant gray-level variations over short distances. Gradually varying patterns are associated with low spatial frequencies. By filtering spatial frequencies, you can remove, attenuate, or highlight the spatial components to which they relate.

You can use a lowpass frequency filter to attenuate or remove (truncate) high frequencies present in the FFT plane. This filter suppresses information related to rapid variations of light intensities in the spatial image. An inverse FFT after a lowpass frequency filter produces an image in which noise, details, texture, and sharp edges are smoothed (IMAQ ComplexAttenuate or IMAQ ComplexTruncate).

A highpass frequency filter attenuates or removes (truncates) low frequencies present in the FFT plane. This filter suppresses information related to slow variations of light intensities in the spatial image. In this case, an inverse FFT after a highpass frequency filter produces an image in which overall patterns are sharpened and details are emphasized (IMAQ ComplexAttenuate or IMAQ ComplexTruncate).

A *mask frequency filter* removes frequencies contained in a mask specified by the user (IMAQ Mask).

The display of complex images is handled by IMAQ WindDraw. This VI displays an image by inverting the high and low frequencies and then dividing their values by a size factor.

The following formula calculates this size factor m:

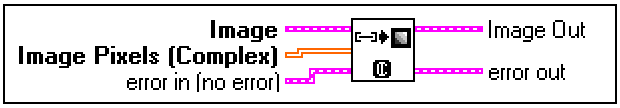
$$m = f(w + h) = f(32.2n) = 2.4n$$

where w is the width of the image and h is the height.

IMAQ ArrayToComplexImage

Creates a complex image from a complex 2D array.

C



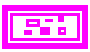


Image is the reference to the complex image to be created.

Image Pixels (Complex) is the complex 2D array (Line, Column) containing all the pixel values that form the image. The first index corresponds to the vertical axis and the second to the horizontal index. The final size of the image is equal to the size of the array. The image passed in the input **Image** is forced to the same size as the complex 2D array encoded by **Input Pixels**.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

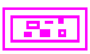


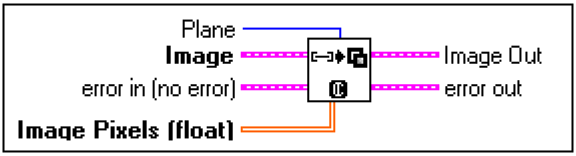
Image Out is the reference to the destination (output) image.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ ArrayToComplexPlane

Replaces the real part or the imaginary part of a complex image, starting from a 2D array of floating-point values.

C



Plane specifies which component of the complex image is replaced with the values encoded in the array of floating points **Image Pixels.** The following values are valid:

(Default) Real

Imaginary

Image is the reference to the input complex image.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Image Pixels (float) is a 2D floating-point array (Line, Column) containing all the pixel values that form the image. The first index corresponds to the vertical axis and the second to the horizontal index. The final size of the image is equal to the size of the array. The image passed in the input **Image** is forced to the same size as the array encoded by **Input Pixels**.

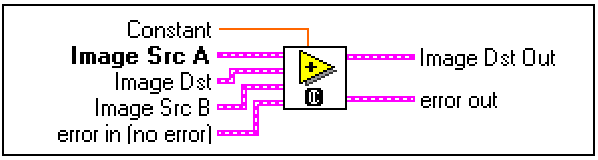
Image Out is the reference to the destination (output) image.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ ComplexAdd

Adds two images where the first is a complex image or adds a complex image and a complex constant.

C



Constant is the complex constant added to the input **Image Src A** for image-constant operations. The default is 0.

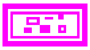


Image Src A is the handle of the first source image and must be a complex image.

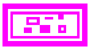


Image Dst is the handle of the complex image that contains the resulting FFT image. This input can accept only a complex image.

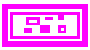
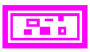
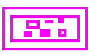


Image Src B is the handle of the second source image. This input can accept an 8-bit, 16-bit, 32-bit floating-point, or complex image. If the image is not a complex image, the imaginary part of the **Image Dst** is equal to **Image Src A**.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src A**.



error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

An operation between an image and a constant occurs when the input **Image Src B** is not connected. The two possibilities are distinguished in the following equations:

$$Dst(x, y) = SrcA(x, y) + SrcB(x, y)$$

or

$$Dst(x, y) = SrcA(x, y) + Constant$$

Table 13-1 describes the different image type combinations that work with this VI, where *I* is the resulting image that is connected to the output **Image Dst**.

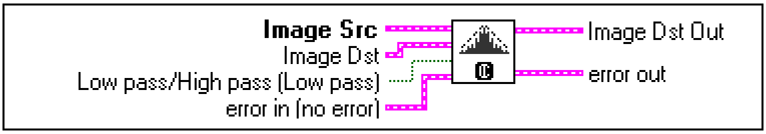
Table 13-1. Image Type Combinations That Work with IMAQ ComplexAdd

Image Connected to Image Src A	Image Connected to Image Src B	Equations
A complex image: I_c	An 8-bit, 16-bit, or 32-bit floating-point image: $I_{8\text{-bit}}$, $I_{16\text{-bit}}$, or $I_{32\text{-bit}}$	Real(I) = Real(I_c) + ($I_{8\text{-bit}}$, $I_{16\text{-bit}}$, or $I_{32\text{-bit}}$) Imaginary(I) = Imaginary(I_c)
A complex image: I_{c1}	Another complex image: I_{c2}	Real(I) = Real(I_{c1}) + Real(I_{c2}) Imaginary(I) = Imaginary(I_{c1}) + Imaginary(I_{c2})

IMAQ ComplexAttenuate

Attenuates the frequencies of a complex image.

C



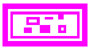


Image Src is the image reference source.

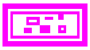
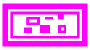


Image Dst is the reference of the image destination.

Low pass/High pass (Low pass) determines which frequencies are attenuated. Choose **Low pass (F)** to attenuate the high frequencies or **High pass (T)** to attenuate the low frequencies. The default is FALSE, which specifies lowpass.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ ComplexConjugate

Computes the conjugate of a complex image. This VI converts the complex pixel data $z = a + ib$ of an FFT image into $z' = a - ib$.

C

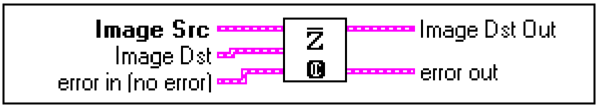


Image Src is the handle of the source image for the image that is used to measure the conjugate. This input can accept only a complex image.

Image Dst is the handle of the complex image that contains the resulting FFT image. This input can accept only a complex image.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

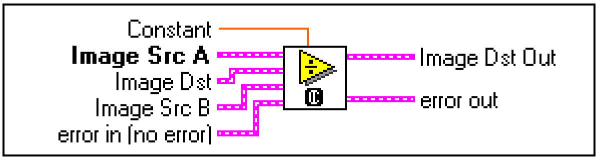
Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ ComplexDivide

Divides one image by another where the first is a complex image or divides a complex image by a complex constant.

C



Constant. The input **Image Src A** is divided by this complex constant for image-constant operations. The default is 0.

Note Division by 0 is not allowed. A constant of 0 is automatically replaced by 1. If one of the two source images is empty, the result is a copy of the other.

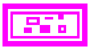


Image Src A is the handle of the first source image and must be a complex image.

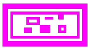


Image Dst is the handle of the complex image that contains the resulting FFT image. This input can accept only a complex image.

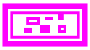
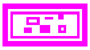


Image Src B is the handle of the second source image. This input can accept an 8-bit, 16-bit, 32-bit floating-point, or complex image.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

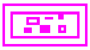
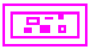


Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src A**.



error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

An operation between an image and a constant occurs when the input **Image Src B** is not connected. The two possibilities are distinguished in the following equations:

$$Dst(x, y) = SrcA(x, y) \div SrcB(x, y)$$
or
$$Dst(x, y) = SrcA(x, y) \div Constant$$

The following describes the different image type combinations that work with this VI, where I is the resulting image that is connected to the output **Image Dst**.

- Image Connected to **Image Src A**: A complex image: I_c

  Image Connected to **Image Src B**: An 8-bit, 16-bit, or 32-bit floating point image: $I_{8\text{-bit}}$, $I_{16\text{-bit}}$, or $I_{32\text{-bit}}$

  Equations:

$$\mathrm{Real}(I) = \frac{\mathrm{Real}(I_c)}{(I_{8\text{-bit},}\ I_{16\text{-bit},}\ \text{or}\ I_{32\text{-bit}})}$$

$$\mathrm{Imaginary}(I) = \frac{\mathrm{Imaginary}(I_c)}{(I_{8\text{-bit},}\ I_{16\text{-bit},}\ \text{or}\ I_{32\text{-bit}})}$$

- Image Connected to **Image Src A**: A complex image: I_{c1}

  Image Connected to **Image Src B**: Another complex image: I_{c2}

  Equations:

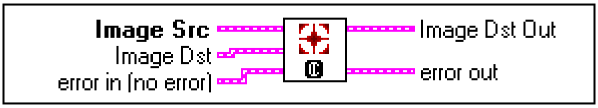
$$\mathrm{Real}(I) = \frac{\mathrm{Real}(I_{c1}) \times \mathrm{Real}(I_{c2}) + \mathrm{Imaginary}(I_{c1}) \times \mathrm{Imaginary}(I_{c2})}{\mathrm{Real}(I_{c2})^2 + \mathrm{Imaginary}(I_{c2})^2}$$

$$\mathrm{Imaginary}(I) = \frac{\mathrm{Imaginary}(I_{c1}) \times \mathrm{Real}(I_{c2}) + \mathrm{Real}(I_{c1}) \times \mathrm{Imaginary}(I_{c2})}{\mathrm{Real}(I_{c2})^2 + \mathrm{Imaginary}(I_{c2})^2}$$

IMAQ ComplexFlipFrequency

Transposes the complex components of an FFT image of a complex image. The high and low frequency components of an FFT image are inverted to produce a central symmetric representation of the spatial frequencies.

C



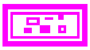


Image Src is the handle of the source image for the image to be transposed. This input can accept only a complex image.

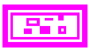
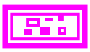


Image Dst is the handle of the complex image that contains the resulting FFT image. This input can accept only a complex image.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

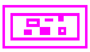
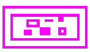


Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.

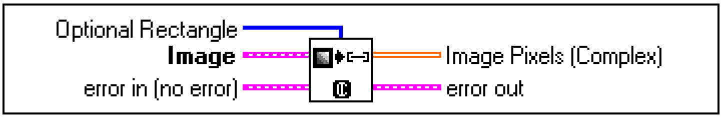


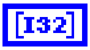
error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ ComplexImageToArray

Extracts the pixels from a complex image into a 2D complex array.

C





Optional Rectangle specifies a rectangular region of the complex image to be extracted. The operation is applied to the entire image if the input is empty or not connected.

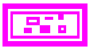
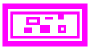


Image is the reference to the complex image.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

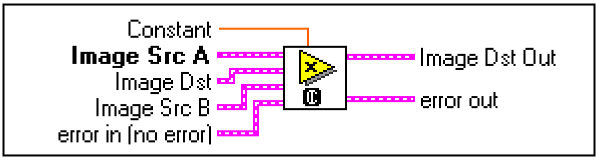
Image Pixels (Complex) is a 2D array containing all the pixel values that comprise the image. The first index corresponds to the vertical axis and the second to the horizontal index. The final size of the array is equal to the size of the image or to the size of the optional rectangle.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ ComplexMultiply

Multiplies two images where the first is a complex image or multiplies a complex image and a complex constant.

C



Constant. The input **Image Src A** is multiplied by this complex constant for image-constant operations. The default is 0.

Image Src A is the handle of the first source image and must be a complex image.

Image Dst is the handle of the complex image that contains the resulting FFT image. This input can accept only a complex image.

Image Src B is the handle of the second source image. This input can accept an 8-bit, 16-bit, 32-bit floating-point, or complex image.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src A**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

An operation between an image and a constant occurs when the input **Image Src B** is not connected. The two possibilities are distinguished in the following equations:

$$Dst(x, y) = SrcA(x, y) \times SrcB(x, y)$$

or

$$Dst(x, y) = SrcA(x, y) \times Constant$$

Table 13-2 describes the different image type combinations that work with this VI, where *I* is the resulting image that is connected to the output **Image Dst**.

Table 13-2. Image Type Combinations That Work with IMAQ ComplexMultiply

Image Connected to Image Src A	Image Connected to Image Src B	Equations
A complex image: I_c	An 8-bit, 16-bit, or 32-bit floating-point image: $I_{8\text{-bit}}$, $I_{16\text{-bit}}$, or $I_{32\text{-bit}}$	$\text{Real}(I) = \text{Real}(I_c) \times (I_{8\text{-bit}}, I_{16\text{-bit}}, \text{ or } I_{32\text{-bit}})$ $\text{Imaginary}(I) = \text{Imaginary}(I_c) \times (I_{8\text{-bit}}, I_{16\text{-bit}}, \text{ or } I_{32\text{-bit}})$
A complex image: I_{c1}	Another complex image: I_{c2}	$\text{Real}(I) = \text{Real}(I_{c1}) \times \text{Real}(I_{c2}) - \text{Imaginary}(I_{c1}) \times \text{Imaginary}(I_{c2})$ $\text{Imaginary}(I) = \text{Imaginary}(I_{c1}) \times \text{Real}(I_{c2}) + \text{Real}(I_{c1}) \times \text{Imaginary}(I_{c2})$

IMAQ ComplexPlaneToArray

Extracts the pixels from the real part, imaginary part, magnitude, or phase from a complex image into a floating-point 2D array.

C

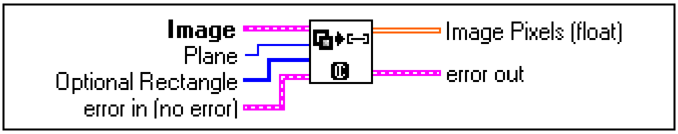
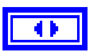


Image is the reference to the input complex image.



Plane indicates which component of the complex image is extracted into an array. The following values are valid:

(Default) Real

Imaginary

Magnitude

Phase

Optional Rectangle specifies a rectangular region of the complex image to be extracted. The operation is applied to the entire image if the input is empty or not connected.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Image Pixels (float) is a 2D floating-point array containing all the pixel values that comprise the image. The first index corresponds to the vertical axis and the second to the horizontal index. The final size of the array is equal to the size of the image or to the size of the optional rectangle.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ ComplexPlaneToImage

Extracts the pixels from the real part, imaginary part, magnitude, or phase from a complex image into an 8-bit, 16-bit, or 32-bit floating-point image.

C

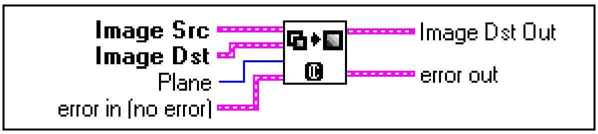


Image Src must be a complex image.

Image Dst must be an 8-bit, 16-bit, or 32-bit floating-point image.

Plane indicates which component of the complex image is extracted. The following values are valid:

(Default) Real

Imaginary

Magnitude

Phase

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

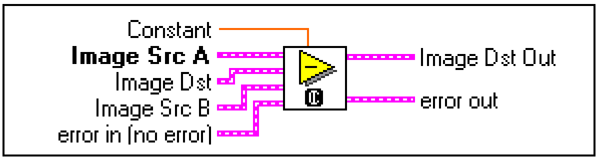
Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. It is the same as **Image Dst**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ ComplexSubtract

Subtracts two images where the first is a complex image or subtracts a complex constant from a complex image.

C



Constant is the complex constant subtracted from the input **Image Src A** for image-constant operations. The default is 0.

Image Src A is the handle of the first source image and must be a complex image.

Image Dst is the handle of the complex image that contains the resulting FFT image. This input can accept only a complex image.

Image Src B is the handle of the second source image. This input can accept an 8-bit, 16-bit, 32-bit floating-point, or complex image. If the image is not a complex image, the imaginary part of the **Image Dst** is equal to **Image Src A**.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src A**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

An operation between an image and a constant occurs when the input **Image Src B** is not connected. The two possibilities are distinguished in the following equations:

$$Dst(x, y) = SrcA(x, y) - SrcB(x, y)$$

or

$$Dst(x, y) = SrcA(x, y) - Constant$$

Table 13-3 describes the different image type combinations that work with this VI, where *I* is the resulting image that is connected to the output **Image Dst**.

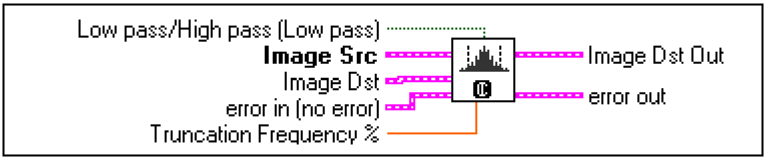
Table 13-3. Image Type Combinations That Work with IMAQ ComplexSubtract

Image Connected to Image Src A	Image Connected to Image Src B	Equations
A complex image: I_c	An 8-bit, 16-bit, or 32-bit floating-point image: $I_{8\text{-bit}}$, $I_{16\text{-bit}}$, or $I_{32\text{-bit}}$	$\text{Real}(I) = \text{Real}(I_c) - (I_{8\text{-bit}}, I_{16\text{-bit}}, \text{ or } I_{32\text{-bit}})$ $\text{Imaginary}(I) = \text{Imaginary}(I_c)$
A complex image: I_{c1}	Another complex image: I_{c2}	$\text{Real}(I) = \text{Real}(I_{c1}) - \text{Real}(I_{c2})$ $\text{Imaginary}(I) = \text{Imaginary}(I_{c1}) - \text{Imaginary}(I_{c2})$

IMAQ ComplexTruncate

Truncates the frequencies of a complex image.

C



Low pass/High pass (Low pass) determines which frequencies are truncated. Choose **Low pass (F)** to remove the high frequencies or **High pass (T)** to remove the low frequencies. The default is FALSE, which specifies lowpass.

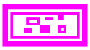


Image Src is the image reference source. It must be a complex image.

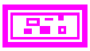
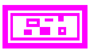


Image Dst is the reference of the image destination. If it is connected, it must be the same type as the **Image Src**.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Truncation Frequency % is the percentage of the frequencies that are retained within a Fourier-transformed image. This percentage is expressed with respect to the length of the diagonal of the FFT image and the Boolean **Low pass/High pass (Low pass)**. The default value is 10.

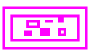


Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

For example, the defaults **Low pass (F)** and **10** result in retaining 10 percent of the frequencies starting from the center (low frequencies). Selecting **High pass (T)** and **10** results in retaining 10 percent of the frequencies starting from the outer periphery.

IMAQ FFT

Computes the FFT of an image. The FFT is a complex image in which high frequencies are grouped at the center, while low frequencies are located at the edges.

Note The calculated FFT is not normalized. Use IMAQ ComplexDivide to normalize the complex image.

8 16 F

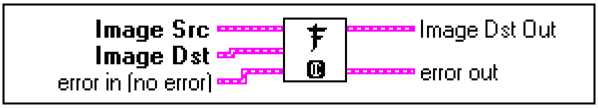


Image Src is the handle of the source image.

Image Dst is the handle of the complex image that contains the resulting FFT image. This input can accept only a complex image. The complex image is resized to the **Image Src**.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

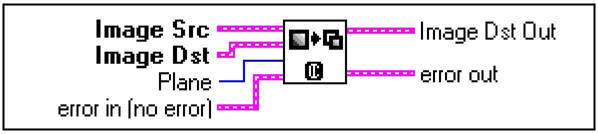
Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ ImageToComplexPlane

Extracts the pixels from an 8-bit, 16-bit, or 32-bit floating-point image into the real part or imaginary part of a complex image.

8 16 C



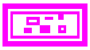


Image Src must be an 8-bit, 16-bit, or 32-bit floating-point image.

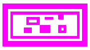
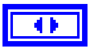


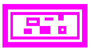
Image Dst must be a complex image.



Plane specifies which component of the complex image is replaced. The following values are valid:

(Default) Real

Imaginary



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

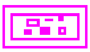


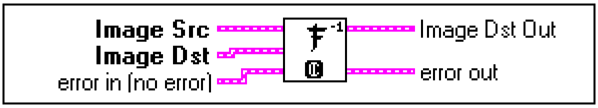
Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ InverseFFT

Computes the inverse FFT of a complex image (2 × 32-bit floating point).

C



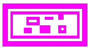


Image Src is the handle of the source image. This input can accept only a complex image.

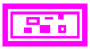
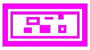


Image Dst is the handle of the 8-bit, 16-bit, or 32-bit floating-point image that contains the resulting spatial image.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

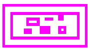
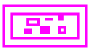


Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI.



error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Note This VI uses a buffer equal to the size of the complex image. An 8-bit image with a resolution of 256 × 256 pixels uses 64 KB of memory. The FFT associated with this image requires eight times the memory, or 64 × 8 = 512 KB. The calculation of the inverse FFT also requires a temporary buffer of 512 KB. Therefore, the total memory necessary for this operation is 1,080 KB.

14

Color VIs

This chapter describes the Color VIs in IMAQ Vision.

Introduction

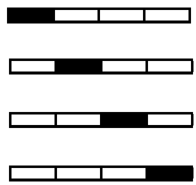
An RGB or HSL image is a color image coded in three parts. For RGB, these parts are red, green, and blue. For HSL, these parts are hue, saturation, and luminance. A pixel encoded in 32 bits is actually four channels:

alpha channel (not used)

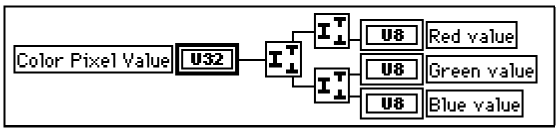
red or hue channel

green or saturation channel

blue or luminance channel



A color pixel encoded as an unsigned 32-bit integer control can be decomposed as shown in the following graphic.



A color image always is encoded in memory in either the form (R, G, B) or the form (H, S, L). However, there are a number of other coding models such as (H, S, I) and (H, S, V). The (H, S, I) model is composed as hue, saturation, and intensity, and the (H, S, V) model as hue, saturation, and value.

To determine the values for hue, saturation, intensity, or value, a measurement is made from the red, green, and blue components. These extractions are approximate. In effect, a color converted between two of the different color models (for instance, RGB to HSL) and then reconverted back to the original color model, does not have exactly the same values as the original image. This difference is because of the 8-bit encoding of the image planes, which causes some loss of data.

The following formulas show the method to convert an RGB color value to an HSL color value:

$$H = \frac{256\tan^{-1}\left(\frac{\sqrt{3}(G-B)}{2R-G-B}\right)}{2\pi}$$

$$S = 256\left(1 - \frac{3\min(R, G, B)}{R+G+B}\right)$$

$$L = 0.299R + 0.587G + 0.114B$$

In the HSV and HSI modes, H and S are computed as approximations to the model shown above. V and I are computed as follows:

$$V = \frac{\max(R, G, B) - \min(R, G, B)}{2}$$

$$I = \max(R, G, B)$$

The principal operations that can be performed on color images are:

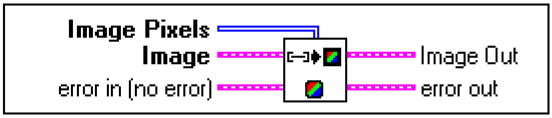
- Extracting or replacing a color image plane (R, G, B, H, S, L, V, I)
- Applying a threshold to a color image based on one of the four color models (RGB, HSL, HSV, or HSI)
- Performing a histogram on a color image based on one of the four color models (RGB, HSL, HSV, or HSI)
- Matching colors in an image region to some previously known colors

Other Color VIs are auxiliary VIs you can use for the following purposes:

- Extracting or replacing a pixel, a line, or a part of an image
- Converting the image from one color model to another
- Converting the image to and from an array of data

IMAQ ArrayToColorImage

Creates a color image from a 2D array. This VI receives the values as a 2D array of unsigned 32-bit integer controls. You can convert a 2D array of clusters coding the three color values as either (R, G, B), (H, S, L), (H, S, V), or (H, S, I) into a 2D array of pixels (unsigned 32-bit integer controls) using the IMAQ ColorValueToInteger VI.



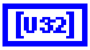


Image Pixels contains the pixel values as a 2D array of unsigned 32-bit integer controls.

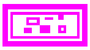
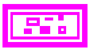


Image must be an RGB or an HSL image.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

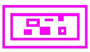
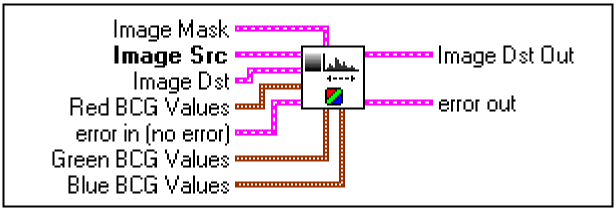


Image Out is the reference to the destination (output) image.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ ColorBCGLookup

Applies a brightness, contrast, and gamma correction to each color plane separately.



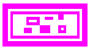


Image Mask is an 8-bit image that specifies the region in the image to modify. Only those pixels in the original image that correspond to an equivalent pixel in the mask different than 0 are processed. All other pixels keep their original value. The complete image is modified if **Image Mask** is not connected.

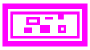


Image Src is the image reference source. It must be an RGB image.

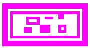


Image Dst (RGB) is the reference of the image destination. If it is connected, it must be an RGB image.

Red BCG Values is a cluster of the following three values to adjust for the red plane:

Brightness (default = 128) sets the brightness of the red plane of the image. This value is used as the X intercept of the transfer function in the look-up table. The neutral value is 128 (no change in the plane). A higher value returns a brighter plane. A value less than 128 decreases the overall brightness of the plane.

Contrast (default = 45.0) sets the contrast of the red plane of the image. This control is used as the slope of the transfer function in the look-up table and is expressed in degrees. A slope of 45° is neutral. A higher value returns a more contrasted red plane. A value smaller than 45 decreases the contrast of the red plane.

Gamma (default = 1.0) sets the gamma correction applied to the red plane of the image. The neutral value is 1. A value greater than 1 gives extended contrast for small pixel values and less contrast for large pixel values. A value smaller than 1 returns a plane with less contrast for small pixel values and extended contrast for large pixel values.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Green BCG Values is a cluster of the following three values to adjust for the Green plane:

Brightness (default = 128) sets the brightness of the green plane of the image.

Contrast (default = 45.0) sets the contrast of the green plane of the image.

Gamma (default = 1.0) sets the gamma correction applied to the green plane of the image.

Blue BCG Values is a cluster of the following three values to adjust for the Blue plane:

Brightness (default = 128) sets the brightness of the blue plane of the image.

Contrast (default = 45.0) sets the contrast of the blue plane of the image.

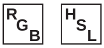
Gamma (default = 1.0) sets the gamma correction applied to the blue plane of the image.

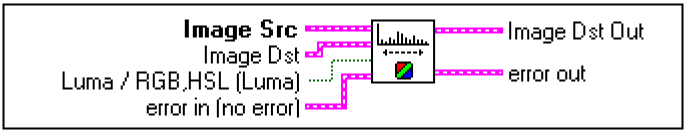
Image Dst Out is the reference to the output RGB image that is obtained by applying the color LUT to the source image.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ ColorEqualize

Equalizes a color image. This VI equalizes either the luminance plane (default) or all three planes.





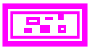


Image Src is the reference to the source image. It must be an RGB or an HSL image.

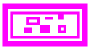
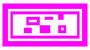


Image Dst is the reference to the destination image. If connected, it must be an RGB or an HSL image. It must have the same type as **Image Src**.

Luma / R,G,B (Luma) specifies whether to perform the operation on the luminance plane or on all three planes. An equalization on the luminance plane conserves the hue and saturation from the color image. An equalization of the three planes gives a stronger contrast but changes the hue and saturation of the color image. The default is **Luma**.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

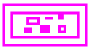
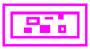


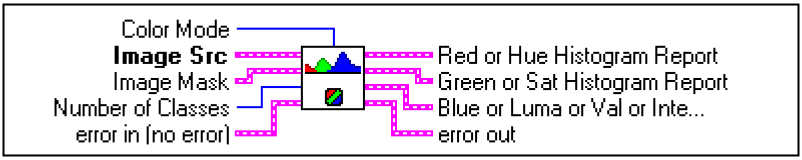
Image Dst Out is the reference to the output image that is obtained after equalization of the source color image.

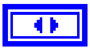


error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ ColorHistogram

Calculates the histograms extracted from the three planes of an image. This VI can function in one of four modes corresponding to the four color models (RGB, HSL, HSV, or HSI). IMAQ ColorHistograph, a variant of the IMAQ ColorHistogram VI, has the advantage that its output data is directly compatible with a LabVIEW or BridgeVIEW graph.



Color Mode defines the image color format to use for the operation. The default is RGB. You can choose from the following values:

(Default) RGB

HSL

HSV

HSI

Image Src is the input source image used for calculating the histogram. It must be an RGB or an HSL image.

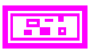


Image Mask, if connected, must be an 8-bit image. **Image Mask** specifies the region of the image to take into account for the histogram computation. Only those pixels in the original image that correspond to an equivalent pixel in the mask different than 0 are processed. All other pixels are ignored.

Number of Classes specifies the number of classes used to classify the pixels. The default is 256.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Red or Hue Histogram Report is a cluster that returns the detailed results from a histogram calculated on a red or hue plane, depending on the **Color Mode**. This cluster is the same as the cluster used by IMAQ Histogram. It contains the following elements:

Histogram returns the histogram values in an array. The elements found in this array are the number of pixels per class. The *n*th class contains all pixel values belonging to the interval $[(\mathit{Starting\ Value} + (n - 1) \times \mathit{Interval\ Width}), (\mathit{Starting\ Value} + n \times \mathit{Interval\ Width} - 1)]$.

Minimal Value returns the smallest pixel value used in calculating the histogram.

Maximal Value returns the largest pixel value used in calculating the histogram.

Starting Value is always equal to 0 here. It returns the smallest pixel value from the first class calculated in the histogram. It can be equal to the **Minimum** value from the **Interval Range** or the smallest value found for the image type connected.

Interval Width returns the length of each class.

Mean Value returns the mean value of the pixels used in calculating the histogram.

Standard Variation returns the standard deviation from the histogram. A higher value corresponds to a better distribution of the values in the histogram and the image.

Area (pixels) returns the number of pixels used in the histogram calculation. This is influenced by the contents of **Image Mask**.

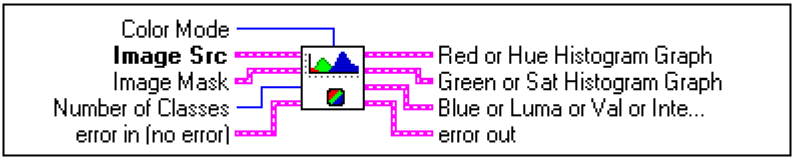
Green or Sat Histogram Report is a cluster that returns the detailed results from a histogram calculated on the green or saturation plane, depending on the **Color Mode**. It has the same elements as found in **Red or Hue Histogram Report**.

Blue or Luma or Val or Inten Histogram Report is a cluster that returns the detailed results from a histogram calculated on the blue, luminance, value, or intensity planes, depending on the **Color Mode**. It has the same elements as found in **Red or Hue Histogram Report**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ ColorHistograph

Calculates the histograms extracted from the three planes of an image. This VI can function in one of four modes corresponding to the four color models (RGB, HSL, HSV, or HSI). The output from this VI is directly compatible with a LabVIEW or BridgeVIEW graph.



Color Mode defines the image color format to use for the operation. The default is RGB. You can choose from the following values:

(Default) RGB

HSL

HSV

HSI

Image Src is the input source image used for calculating the histogram. It must be an RGB or an HSL image.

Image Mask, if connected, must be an 8-bit image. **Image Mask** specifies the region of the image to take into account for the histogram computation. Only those pixels in the original image that correspond to an equivalent pixel in the mask different than 0 are processed. All other pixels are ignored.

Number of Classes specifies the number of classes used to classify the pixels. The default is 256.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Red or Hue Histogram Graph is a cluster that returns the detailed results from a histogram calculated on a red or hue plane, depending on the **Color Mode**. This cluster is the same as the cluster used by IMAQ Histograph. It contains the following elements:

Starting Value is always equal to 0 here. This parameter is returned in the type Histogram Report, as in the IMAQ Histograph VI.

Incremental Value returns the incrementing value that specifies how much to add to **Starting Value** in calculating the median value of each class from the histogram. The median value x_n from the nth class is $x_n = \textit{Starting Value} + n \times \textit{Incremental Value}$.

Histogram returns the histogram values in an array. The elements found in this array are the number of pixels per class. The nth class contains all pixel values belonging to the interval $[(\textit{Starting Value} + (n - 1) \times \textit{Interval Width}), (\textit{Starting Value} + n \times \textit{Interval Width} - 1)]$.

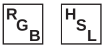
Green or Sat Histogram Graph is a cluster that returns the detailed results from a histogram calculated on the green or saturation plane, depending on the **Color Mode**. It has the same elements as found in **Red or Hue Histogram Graph**.

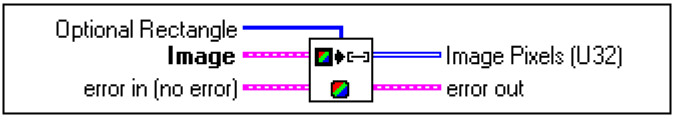
Blue or Luma or Val or Inten Histogram Graph is a cluster that returns the detailed results from a histogram calculated on the blue, luminance, value, or intensity planes, depending on the **Color Mode**. It has the same elements as found in **Red or Hue Histogram Graph**.

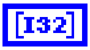
error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ ColorImageToArray

Extracts the pixels from a color image or from part of a color image into a 2D array. This VI returns the values as a 2D array of unsigned 32-bit integer indicators. You can convert this 2D array into a 2D array of clusters coding the three color values as either (R, G, B), (H, S, L), (H, S, V), or (H, S, I) using the IMAQ IntegerToColorValue VI.







Optional Rectangle designates a rectangular region (Left/Top/Right/Bottom) within an image in which the pixels are to be changed. If this array is empty the entire image is changed.

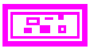
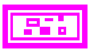


Image must be an RGB or an HSL image.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

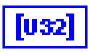
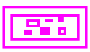


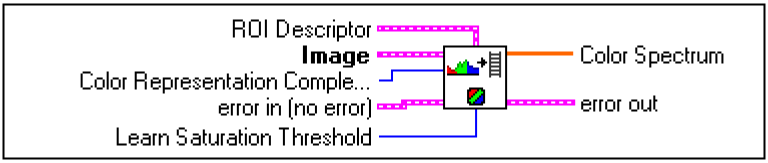
Image Pixels (U32) returns the pixel values as a 2D array of unsigned 32-bit integer indicators.



error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ ColorLearn

Extracts the color features of an image, which can be used for color matching or other applications related with color information, such as color identification and color image segmentation.



ROI Descriptor describes a region(s) in the image that contains the color to be learned. The operation is applied to the entire image if the input is empty or not connected. If the ROI descriptor contains multiple regions, the color information in all these regions is accumulated before learning.

Image Src is the reference to the color image from which you want to learn a color(s).

Color Representation Complexity specifies the level of color complexity in the image (low, medium, or high) that will affect the number of features in the color feature array. The default is **Low**. Set this option to **High** when you need to distinguish colors with very close hue values.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

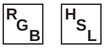
Learn Saturation Threshold specifies the threshold value to distinguish two colors with the same hue value. The default is 80.

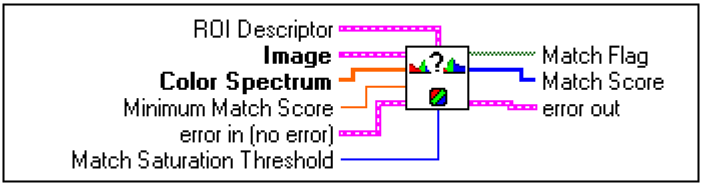
Color Spectrum returns the color feature found in the image region. These features represent the color information in the image region in a compact form.

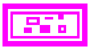
error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ ColorMatch

Finds the match between multiple colors in an image and those defined by the input **Color Spectrum**, which was output from the IMAQ ColorLearn VI.







ROI Descriptor contains information about the regions in the image whose color information has to be compared with the information in the **Color Spectrum** color feature array. If the **ROI Descriptor** contains multiple regions, the color information in each region is compared with the input color information individually and the match results are reported for each region. If the **ROI Descriptor** is not connected, the entire image is considered as a region.

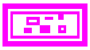
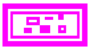


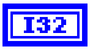
Image is the reference to the color image from which you want to extract the color feature to find the match with the color defined by the input **Color Spectrum** color feature array.

Color Spectrum contains information about the color to be matched. During the matching phase, the color information in each specified image region is compared with the information contained in this array.

Minimum Match Score specifies the threshold for color matching scores. The data range is between 0 and 1000. A match is found if the **Match Score** is higher than this value.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.



Match Saturation Threshold specifies the threshold value to distinguish two colors with the same hue value. The default is 80. The value used here should be the same as that used in the IMAQ ColorLearn VI.

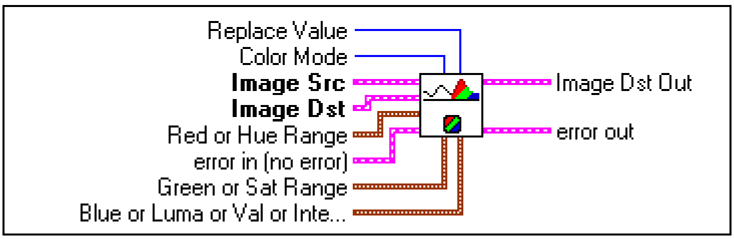
Match Flag is an array containing Boolean information that indicates whether there is a match between the input color information and the color in each specified region in the image. For each specified region in the image, the match flag is set to TRUE if its color information matches the input color information.

Match Score is an array containing the match scores for each region in the image. The match scores are represented on a scale of 0 to 1000 where 0 indicates no match and 1000 indicates a perfect match.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ ColorThreshold

Applies a threshold to the three planes of an RGB or HSL image and places the result into an 8-bit image. A test is performed with each range **(Red or Hue Range**, **Green or Sat Range**, and **Blue or Luma or Val or Inten Range)** to determine whether the corresponding pixel from the **Image Src** is set to the value specified in **Replace Value**. If a pixel from the **Image Src** does not have corresponding pixel values specified in all three ranges, the corresponding pixel in **Image Dst Out** is set to 0.



Note If a range is not specified for a given plane, the **Replace Value** is used for that plane. Therefore you easily can apply a threshold to one of the three ranges without having to set the values of the other two ranges.

Replace Value specifies the value applied to the destination image when the corresponding pixel from the **Image Src** is found in all three ranges. The default is 1.

Color Mode defines the image color format to use for the operation. The default is RGB. You can choose from the following values:

(Default) RGB

HSL

HSV

HSI

Image Src is the reference to the image to threshold. It must be an RGB or an HSL image.

Image Dst must be connected and must be an 8-bit image.

Red or Hue Range is a cluster used to determine the thresholding range for the red or hue plane, depending on the **Color Mode**. Any pixel values not included in this range are reset to zero in the destination image. The pixel values included in this range are altered depending on the status of the **Replace** input. By default, all pixel values are included (0, 255). **Red or Hue Range** includes the following elements:

> **Lower Value** is the minimal pixel value in the red or hue plane that is used for the threshold. The default is 0.
>
> **Upper Value** is the maximal pixel value in the red or hue plane that is used for the threshold. The default is 255.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Green or Sat Range is a cluster used to determine the thresholding range for the green or saturation plane, depending on the **Color Mode**. Any pixel values not included in this range are reset to zero in the destination image. The pixel values included in this range are altered depending on the status of the **Replace** input. By default, all pixel values are included (0, 255). **Green or Sat Range** has the same elements as found in **Red or Hue Range**.

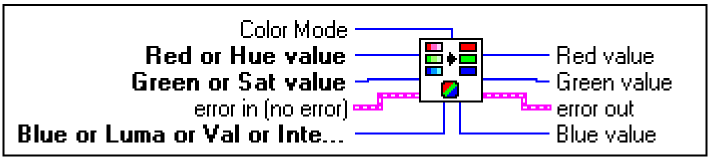
Blue or Luma or Val or Inten Range is a cluster used to determine the thresholding range for the blue, luminance, or value plane, depending on the **Color Mode**. Any pixel values not included in this range are reset to zero in the destination image. The pixel values included in this range are altered depending on the status of the **Replace** input. By default, all pixel values are included (0, 255). **Blue or Luma or Val or Inten Range** has the same elements as found in **Red or Hue Range**.

Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. **Image Dst Out** is the same as **Image Dst**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ ColorToRGB

Converts an HSL, HSV, or HSI color value into an RGB color value.



Color Mode defines the image color format conversion to perform. The default is no change. You can choose from the following values:

RGB	(Default) no change
HSL	Convert from HSL
HSV	Convert from HSV
HSI	Convert from HSI

Red or Hue value is the input value for the first color plane, depending on the **Color Mode** chosen.

Green or Sat value is the input value for the second color plane, depending on the **Color Mode** chosen.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Blue or Luma or Val or Inten value is the input value for the third color plane, depending on the **Color Mode** chosen.

Red value is the output red value.

Green value is the output green value.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Blue value is the output blue value.

IMAQ ColorUserLookup

Applies a look-up table (LUT) to each color plane.

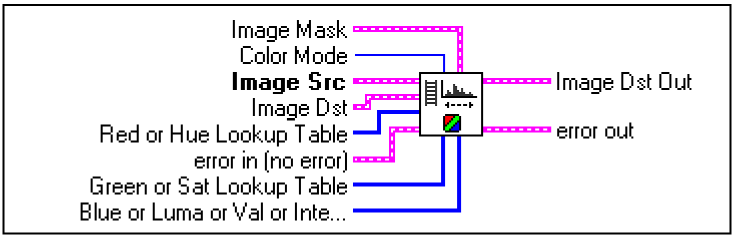


Image Mask is an 8-bit image that specifies the region in the image to modify. Only those pixels in the original image that correspond to an equivalent pixel in the mask different than 0 are processed. All other pixels keep their original value. The complete image is modified if **Image Mask** is not connected.

Color Mode defines the image color format to use for the operation. The default is RGB. You can choose from the following values:

(Default) RGB

HSL

HSV

HSI

Image Src is the reference to the source image. It must be an RGB or an HSL image.

Image Dst is the reference to the destination image. If connected, it must be an RGB or an HSL image. It must have the same type as **Image Src**.

Red or Hue Lookup Table is the LUT applied to the first color plane, depending on the **Color Mode**. This array can contain a maximum of 256 elements. The array is filled automatically when you specify fewer than 256 elements. This procedure does not change pixel values that are not explicitly specified from the values of the LUT you input. By default this array is empty, and no replacement occurs on this plane.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Green or Sat Lookup Table is the LUT applied to the second color plane, depending on the **Color Mode**. This array can contain a maximum of 256 elements. The array is filled automatically when you specify fewer than 256 elements. This procedure does not change pixel values that are not explicitly specified from the values of the LUT you input. By default this array is empty, and no replacement occurs on this plane.

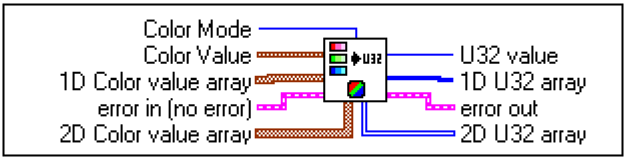
Blue or Luma or Val or Inten Lookup Table is the LUT applied to the third color plane, depending on the **Color Mode**. This array can contain a maximum of 256 elements. The array is filled automatically when you specify fewer than 256 elements. This procedure does not change pixel values that are not explicitly specified from the values of the LUT you input. By default this array is empty, and no replacement occurs on this plane.

Image Dst Out is the reference to the output color image that is obtained by applying the color LUT to the source image.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ ColorValueToInteger

Converts clusters composed of three colors in mode (R, G, B), (H, S, L), (H, S, V), or (H, S, I) into colors encoded in the form of an unsigned 32-bit integer control. The elements of these clusters can contain single values, 1D arrays, 2D arrays, or a combination of the above.



Color Mode defines the image color format to use for the output. The default is RGB, which specifies that the input and output values are the same. You can choose from the following values:

RGB	(Default) no change
HSL	Convert to HSL
HSV	Convert to HSV
HSI	Convert to HSI

Color Value is a cluster containing a color in (R, G, B), (H, S, L), (H, S, V), or (H, S, I), depending on the **Color Mode**.

Red or Hue Value is the first color plane value, depending on the **Color Mode**.

Green or Sat Value is the second color plane value, depending on the **Color Mode**.

Blue or Luma or Val or Inten Value is the third color plane value, depending on the **Color Mode**.

1D Color value array is a 1D array of clusters containing the color values. The values are in (R, G, B), (H, S, L), (H, S, V), or (H, S, I), depending on the status of the set **Color Mode**. These clusters are the same type as **Color Value**.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

2D Color value array is a 2D array of clusters containing the color values. The values are in (R, G, B), (H, S, L), (H, S, V), or (H, S, I), depending on the status of the set **Color Mode**. These clusters are the same type as **Color Value**.

U32 value receives the color value resulting from the input **Color Value**, and it is encoded as an unsigned 32-bit integer control.

1D U32 array receives the color value resulting from the input **1D Color value array**, and it is encoded as a 1D array of unsigned 32-bit integer controls.

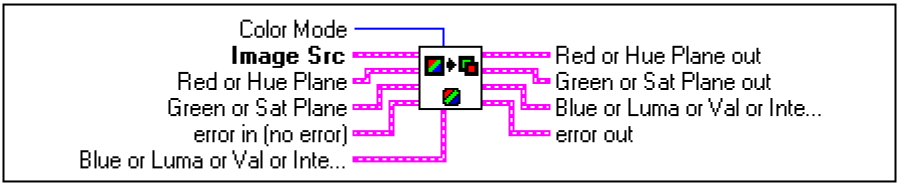
error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

2D U32 array receives the color value resulting from the input **2D Color value array**, and it is encoded as a 2D array of unsigned 32-bit integer controls.

IMAQ ExtractColorPlanes

Extracts the three planes (RGB, HSL, HSV, or HSI) from an image.

RGB HSL



Color Mode defines the image color format to use for the operation. The default is RGB. You can choose from the following values:

(Default) RGB

HSL

HSV

HSI

Image Src is the reference to an image that has its three planes extracted: RGB, HSL, HSV, or HSI. It must be an RGB or an HSL image.

Red or Hue Plane is the reference to the destination image. It contains the first color plane. This plane can be either the red plane (**Color Mode** RGB) or the hue plane (**Color Mode** HSL, HSV, or HSI). It must be an 8-bit image. The color plane is not extracted if the input is not connected.

Green or Sat Plane is the reference to the destination image. It contains the second color plane. This plane can be either the green plane (**Color Mode** RGB) or the saturation plane (**Color Mode** HSL, HSV, or HSI). It must be an 8-bit image. The color plane is not extracted if the input is not connected.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Blue or Luma or Val or Inten Plane is the reference to the destination image. It contains the third color plane. This plane can be either the blue plane (**Color Mode** RGB), the luminance plane (**Color Mode** HSL), the value plane (**Color Mode** HSV) or the intensity plane (**Color Mode** HSI). It must be an 8-bit image. The color plane is not extracted if the input is not connected.

Red or Hue Plane out is the reference to the image containing the red (or hue) plane of the source (input) image.

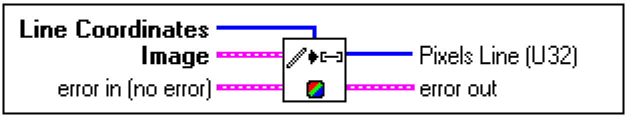
Green or Sat Plane out is the reference to the image containing the green (or saturation) plane of the source (input) image.

Blue or Luma or Val or Inten Plane out is the reference to the image containing the blue (or luminance, value, or intensity) plane of the source (input) image.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ GetColorPixelLine

Extracts a line of pixels from a color image. This VI returns an array of unsigned 32-bit integer indicators. You can convert this array into an array of clusters coding the three color values as either (R, G, B), (H, S, L), (H, S, V), or (H, S, I) using the IMAQ IntegerToColorValue VI.



Line Coordinates is an array specifying the two end points of the line to extract.

Note A line designated by the coordinates [0, 0, 0, 255] consists of 256 pixels. The output **Pixels Line** contains the values specified by this line. Any pixel values outside the image automatically are set to 0 in **Pixels Line**.

Image must be an RGB or an HSL image.

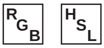
error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Pixels Line (U32) returns the pixel values as a 1D array of unsigned 32-bit integer indicators.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ GetColorPixelValue

Reads the pixel values from a color image. This VI returns the pixel value as an unsigned 32-bit integer indicator. You can convert this indicator into a cluster containing three elements possessing either (R, G, B), (H, S, L), (H, S, V), or (H, S, I) using the IMAQ IntegerToColorValue VI.



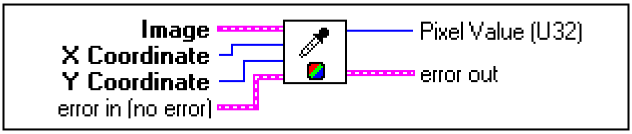


Image must be an RGB or an HSL image.

X Coordinate is the horizontal position of the pixel.

Y Coordinate is the vertical position of the pixel.

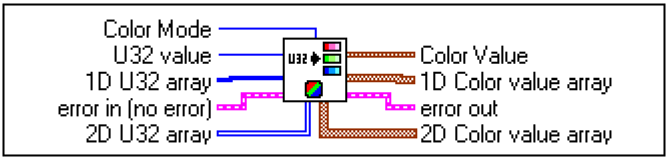
error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

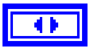
Pixel Value (U32) returns the pixel value as an unsigned 32-bit integer indicator.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ IntegerToColorValue

Converts colors in the form of an unsigned 32-bit integer control into a cluster composed of the three colors in mode (R, G, B), (H, S, L), (H, S, V), or (H, S, I). You can enter these colors as a single value, a 1D array, a 2D array, or a combination of the above.

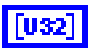




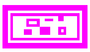
Color Mode defines the image color format to use for the output. The default is RGB, which specifies that the input and output values are the same. You can choose from the following values:

RGB	(Default) no change
HSL	Convert to HSL
HSV	Convert to HSV
HSI	Convert to HSI

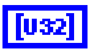
U32 value is a color value encoded as an unsigned 32-bit integer control.



1D U32 array is a set of color values encoded as a 1D array of unsigned 32-bit integer controls.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.



2D U32 array is a set of color values encoded as a 2D array of unsigned 32-bit integer controls.

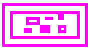
Color Value is a cluster containing the color value resulting from the input **U32 value**. This cluster can contain the values (R, G, B), (H, S, L), (H, S, V), or (H, S, I), depending on the status of the set **Color Mode**. The cluster is composed of the following elements:

Red or Hue Value is the first color plane value, depending on the **Color Mode**.

Green or Sat Value is the second color plane value, depending on the **Color Mode**.

Blue or Light or Val Value is the third color plane value, depending on the **Color Mode**.

1D Color value array is a 1D array containing the color value resulting from the input **1D U32 Array**. This array can contain the values (R, G, B), (H, S, L), (H, S, V), or (H, S, I), depending on the status of the set **Color Mode**.

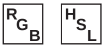


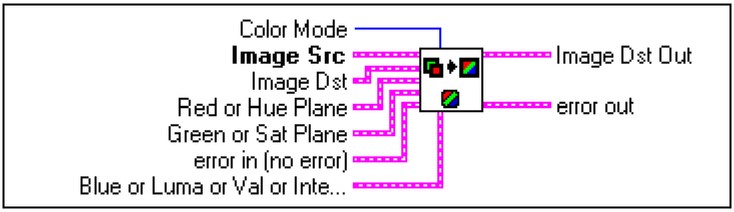
error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

2D Color value array is a 2D array containing the color value resulting from the input **2D U32 Array**. This array can contain the values (R, G, B), (H, S, L), (H, S, V), or (H, S, I), depending on the status of the set **Color Mode**.

IMAQ ReplaceColorPlane

Replaces one or more image planes from a color image (RGB, HSL, HSV, or HSI). Only the planes connected at the input are replaced. If all three planes are connected, the input **Image Src** is not necessary, and only the **Image Dst** is used. The image is resized to the dimensions of the planes passed on input. Therefore their sizes must be identical. If one or two planes are connected, the planes must have the same dimension as the source image.





Color Mode defines the image color format to use for the operation. The default is RGB. You can choose from the following values:

(Default) RGB

HSL

HSV

HSI

Image Src is the reference to an image that has its three color planes replaced. It must be an RGB or an HSL image. This image is not necessary if the destination image and the three color planes are connected.

Image Dst is the reference to the destination image. It must be an RGB or an HSL image. It must have the same type as **Image Src**.

Red or Hue Plane is the reference to the first color plane. This plane can be either the red plane (**Color Mode** RGB) or the hue plane (**Color Mode** HSL, HSV, or HSI). It must be an 8-bit image. The color plane is not replaced if the input is not connected.

Green or Sat Plane is the reference to the second color plane. This plane can be either the green plane (**Color Mode** RGB) or the saturation plane (**Color Mode** HSL, HSV, or HSI). It must be an 8-bit image. The color plane is not replaced if the input is not connected.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

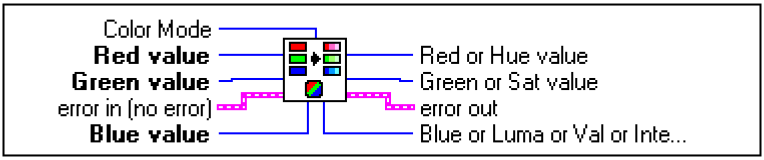
Blue or Luma or Val or Inten Plane is the reference to the third color plane. This plane can be either the blue plane (**Color Mode** RGB), the luminance plane (**Color Mode** HSL), the value plane (**Color Mode** HSV), or the intensity plane (**Color Mode** HSI). It must be an 8-bit image. The color plane is not replaced if the input is not connected.

Image Dst Out is the reference to the output RGB or HSL image that is obtained by replacing one or more planes of the source color image.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ RGBToColor

Converts an RGB color value into another format (HSL, HSV, or HSI).



Color Mode defines the image color format conversion to perform. The default is no change. You can choose from the following values:

RGB	(Default) no change
HSL	Convert to HSL
HSV	Convert to HSV
HSI	Convert to HSI

Red value is the input red value.

Green value is the input green value.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Blue value is the input blue value.

Red or Hue value is the output value for the first color plane, depending on the **Color Mode** chosen.

Green or Sat value is the output value for the second color plane, depending on the **Color Mode** chosen.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Blue or Luma or Val or Inten value is the output value for the third color plane, depending on the **Color Mode** chosen.

IMAQ SetColorPixelLine

Changes a line of pixels from a color image. This VI receives an array of unsigned 32-bit integer controls. You can convert an array of clusters coding the color three values (R, G, B), (H, S, L), (H, S, V), or (H, S, I) into an array of pixels (unsigned 32-bit integer controls) using the IMAQ IntegerToColorValue VI.

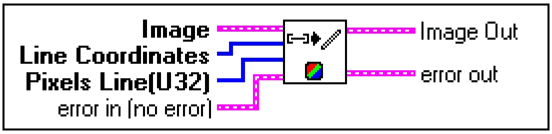


Image must be an RGB or an HSL image.

Line Coordinates is an array specifying the two end points of the line to modify. Any pixels designated by the **Line Coordinates** found outside the actual image are not replaced.

Pixels Line(U32) contains the pixel values as a 1D array of unsigned 32-bit integer controls.

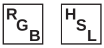
error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

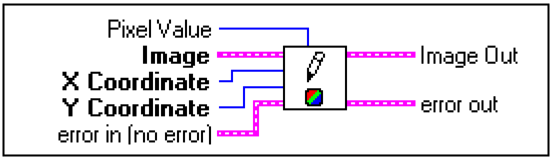
Image Out is the reference to the destination (output) image.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ SetColorPixelValue

Changes the pixel value for a color image. This VI receives the pixel value as an unsigned 32-bit integer control. You can convert the values (R, G, B), (H, S, L), (H, S, V), or (H, S, I) into an unsigned 32-bit integer control using the IMAQ ColorValueToInteger VI.





Pixel Value contains the pixel value as an unsigned 32-bit integer control.

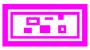
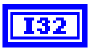
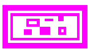


Image must be an RGB or an HSL image.



X Coordinate is the horizontal position of the pixel.

Y Coordinate is the vertical position of the pixel.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

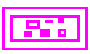
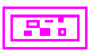


Image Out is the reference to the destination (output) image.



error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

15

External Library Support VIs

This chapter describes the External Library Support VIs in IMAQ Vision. G programmers who have a good understanding of DLLs can use this set of VIs to write their own image acquisition device VIs.

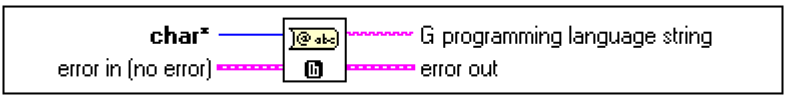
These VIs give you additional functionalities that are not provided by LabVIEW or BridgeVIEW when using an external library. You can use these VIs to do the following actions:

- Get a pointer in the pixel space of an image
- Copy the data of a char* type pointer to a G programming language string
- Copy a memory block addressed by a pointer to a G programming language string
- Change the border size of an image
- Modify the pixel values at the border of an image

Caution These functions are intended for advanced users. Improper use may result in crashing your computer.

IMAQ CharPtrToString

Copies a C character string to a G programming language string. This VI can accept a char* pointer to get the associated string.

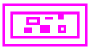


char* is the C character string pointer. The end of the character string is marked with a 0 (`\00`) value.

The copied string size is limited to 65,536 bytes.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

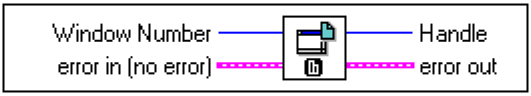
G programming language string is a G programming language string containing all characters before `\00` (end of string mark in C).

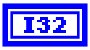


error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

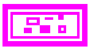
IMAQ Get Window Handle

Returns the operating system handle to an IMAQ Vision window.



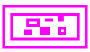


Window Number specifies the window in memory.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Handle returns the requested window handle.

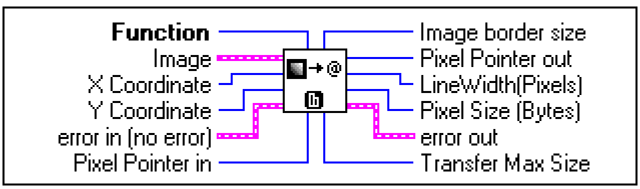


error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ GetImagePixelPtr

Obtains a pointer on the pixels of an image. This VI also returns information on the organization of the image pixels in memory.

8 16 F C RGB HSL



Function has three modes:

0	**Map Pixel Pointer**	Obtains the pointer on a pixel of an image and obtains information related to the organization of the pixels of this image in memory.
1	**Unmap Pixel Pointer**	Frees the pointer and related information previously obtained using **Map Pixel Pointer** (mode 0).
2	**Get Pixel Infos**	Obtains information related to the organization of the pixels of an image in memory without mapping a pointer.

Image is the reference to the image on which the pointer is obtained.

X Coordinate allows you to select the X coordinate of the pixel in the image on which the pointer is required. This parameter is not used in the **Unmap Pixel Pointer** mode. The default is 0.

Y Coordinate allows you to select the Y coordinate of the pixel in the image on which the pointer is required. This parameter is not used in the **Unmap Pixel Pointer** mode. The default is 0.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Pixel Pointer in is used only in the **Unmap Pixel Pointer** mode (see the **Function** description). When the VI is executed to obtain a pointer using the **Map Pixel Pointer** function, some information regarding the pointer that is required to unmap the pixel pointer is recorded.

Note You need to give this pointer to the VI to retrieve this information when executing the **Unmap Pixel Pointer** function.

Image border size is the border size of the image.

Pixel Pointer out is the pointer on the pixels of the image. This pointer is obtained only in the **Map Pixel Pointer** mode.

LineWidth (Pixels) returns the total number of pixels in a horizontal line in the image. This is the sum of the X size of the image, the borders of the image, and the left and right alignments of the image, as shown in Figure 15-1. This number may not match the horizontal size of the image.

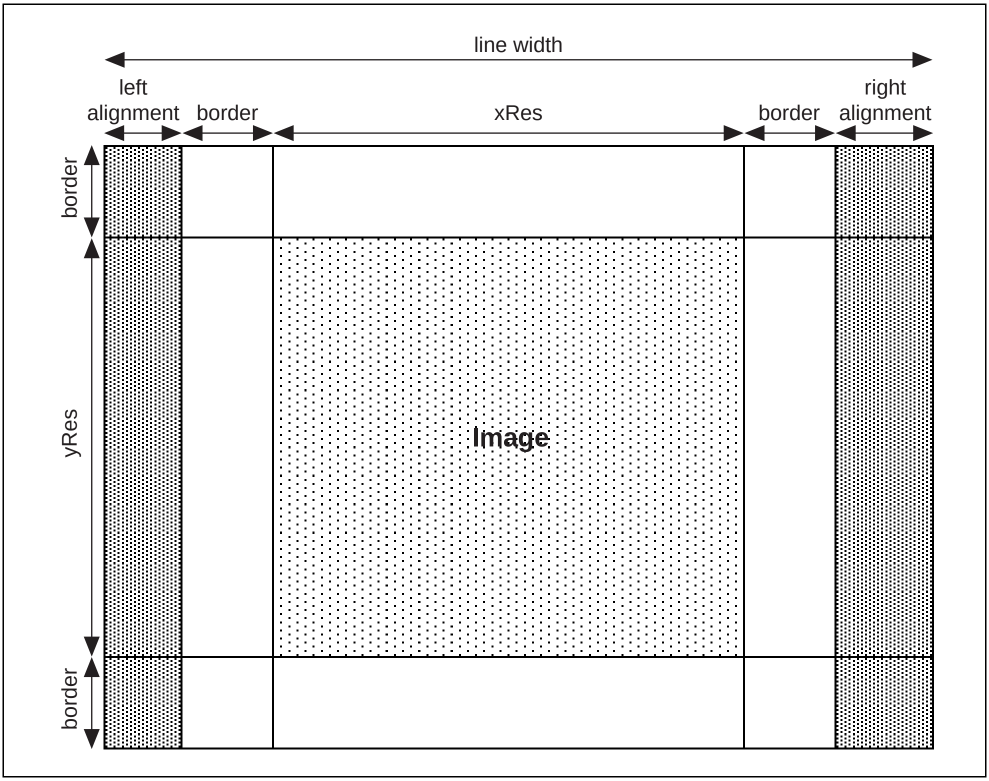


Figure 15-1. Computing the Total Number of Pixels in a Horizontal Line

Pixel Size (Bytes) returns the size in bytes of each pixel in the image. This value multiplied with the **LineWidth** gives the number of bytes occupied by a line of the image in memory.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Transfer Max Size returns the number of bytes from the pixel pointer to the end of the image. This size represents the maximum size of bytes that can be transferred. For example, for an 8-bit image of size 256 × 256 and border 1, the line width is 272, and the maximum transfer size from pixel (0, 0) is 69,632 bytes.

Example

Figure 15-2 illustrates a typical implementation scheme for IMAQ GetImagePixelPtr.

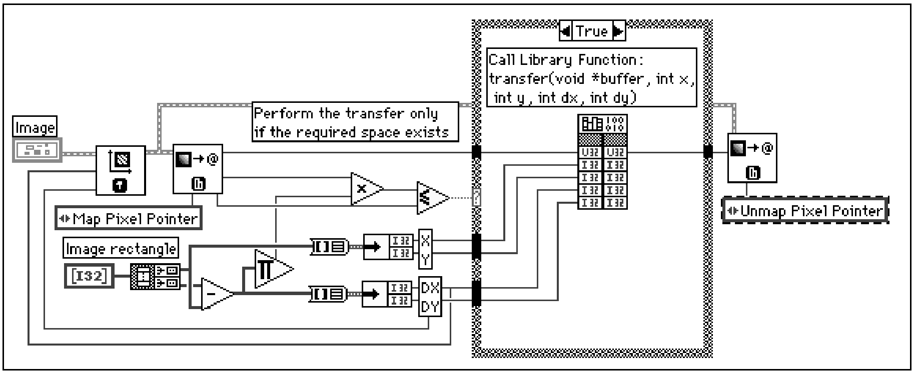


Figure 15-2. IMAQ GetImagePixelPtr Example

This VI receives an image and a rectangle. The transfer call needs five parameters: destination address, X and Y start coordinates, and the X and Y size of the transfer. This VI uses the following steps:

1. From the image rectangle, computes the image size.
2. Resizes the image and obtains a pixel pointer on the coordinates [0, 0] of the image.
3. Verifies that the maximum transfer size is compatible with the parameters needed by the called library.
4. If everything is correct, begins transferring.
5. Unmaps the pixel pointer.

The following code uses IMAQ GetImagePixelPtr to apply a function *f* on the pixels of a floating-point image. The pointer on the pixel (0, 0) of the image (`FirstPixelPtr`) has been retrieved from the VI. In the following C code, `xSize`, `ySize`, and `LineWidth` have been obtained from other VIs.

```
int xSize; // the x size of the image
int ySize; // the y size of the image (Given by IMAQ GetImageSize
           // or IMAQ GetImageInfo)
int LineWidth; // the line width of the image (Given by
               // IMAQ GetImagePixelPtr)
float *FirstPixelPtr; // Given by IMAQ GetImagePixelPtr
float *TempPixelPtr;
int i, j;
for (j = 0; j < ySize; j++) // for each line of the image
    {
    TempPixelPtr = FirstPixelPtr;
```

```
for (i = 0; i < xSize; i++) // for each pixel of the line
    {
    *TempPixelPtr = f (*TempPixelPtr);// apply the function
    TempPixelPtr++;// pixel increment
    }
FirstPixelPtr += LineWidth; // line increment
}
```

IMAQ ImageBorderOperation

Fills the border of an image.

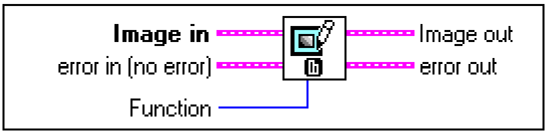


Image in is the reference to the image that has to be modified.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Function indicates the method used to fill the border of the image. This parameter has three possible values:

Border Mirror Repeats the pixel values of the image near the border into the border by symmetry.

Border Copy Sets the value of the border pixels to the value of the image pixel near the border.

Border Clear Sets all border pixels to 0.

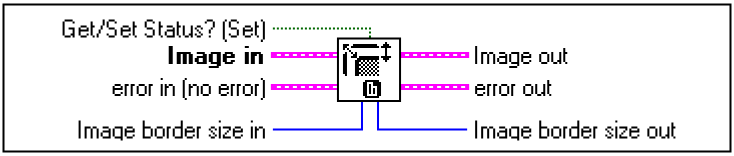
Image Out is the reference to the destination (output) image.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ ImageBorderSize

Sets the border size of the image and determines the current border size of the image.

8 16 F C RGB HSL



Get/Set Status? (Set) determines whether the image border size is changed to the Image border size value (Set) or the current image border size value is retrieved (Get).

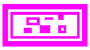
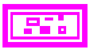


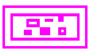
Image in is the reference to the image that has to be modified.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Image border size in determines the new border size of the image.

Image out is the reference to the destination (output) image.

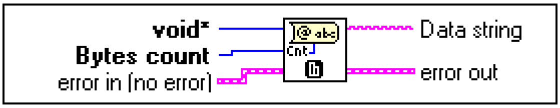


error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

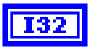
Image border size out is the border size of the image.

IMAQ MemPeek

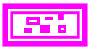
Copies memory into a string.



void* is the pointer of the memory to be copied.

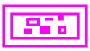


Bytes count is the number of bytes to be copied into the string.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Data string is the G programming language string containing the bytes of the specified memory zone.



error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

16

Browser VIs

This chapter describes VIs for creating and using an image browser. You can use these VIs to accomplish the following tasks:

- Configure a browser
- Configure the selection of one or many images
- Add an image
- Remove an image
- Replace an image
- Get or set the selection status of an image

Each thumbnail in the browser is made up of or defined using the elements as shown in Figure 16-1.

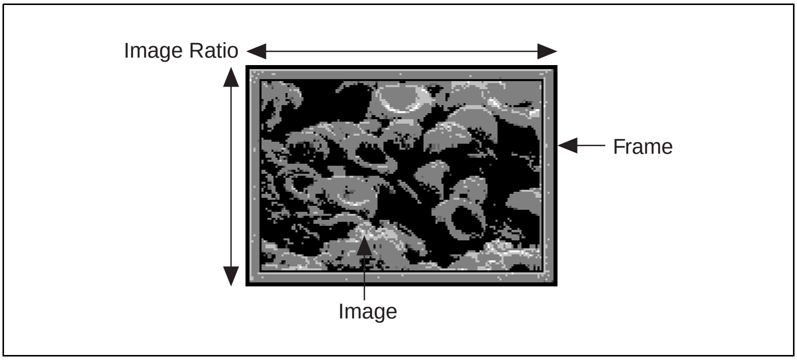
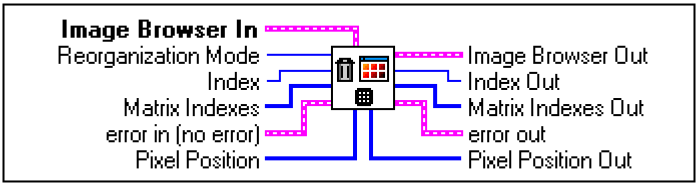


Figure 16-1. Browser Elements

The browser is based on a standard IMAQ RGB image window, which enables the display of any type of image, including 8-bit images with color palettes. The Browser VIs are found only in the Advanced version of IMAQ Vision.

IMAQ Browser Delete

Deletes an image from the browser. Set only one of the following controls: **Index**, **Matrix Indexes**, or **Pixel Position**. After deleting an image, the browser reorganizes the remaining images by filling the free space or leaving it vacant.



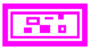
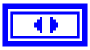
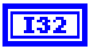


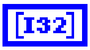
Image Browser In is the RGB image used by the browser.



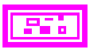
Reorganization Mode is the remapping mode of the browser. The free positions are filled or are left vacant.



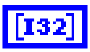
Index is the position of the image to delete.



Matrix Indexes is the (Column, Line) position of the deleted thumbnail.



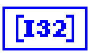
error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.



Pixel Position defines the pixel coordinates of the area where the thumbnail is removed.

Image Browser Out is the resulting browser image.

Index Out is the index of the position where the image is removed from the browser.



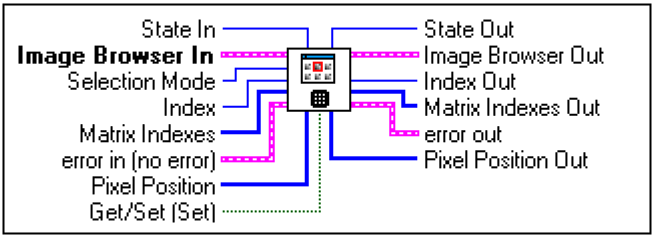
Matrix Indexes Out is the (Column, Line) position of the removed thumbnail.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Pixel Position Out returns the pixel coordinates (X, Y top-left, X, Y right-bottom) defining the area where the image is removed.

IMAQ Browser Focus

Gets or sets the status of images on the browser. You can choose either of the possible status values: selected or not selected. Selected images are framed using the **Focus Color** (see the *IMAQ Browser Focus* description).



State In is the image status value.

Image Browser In is the RGB image the browser uses.

Selection Mode has two possible modes: multiple selection or single selection.

Index is the position of the image to select.

Matrix Indexes is the (Column, Line) position of the selected thumbnail.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Pixel Position defines the pixel coordinates of the area where the thumbnail is selected.

Get/Set (Set) is the VI action.

State Out is the image status.

Image Browser Out is the resulting browser image.

Index Out is the index of the position of the selected image.

Matrix Indexes Out is the (Column, Line) position of the selected thumbnail.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Pixel Position Out returns the pixel coordinates (X, Y top-left, X, Y right-bottom) defining the area where the image is selected.

IMAQ Browser Focus Setup

Configures the appearance of a thumbnail that is selected on the browser. A selected image is framed using the **Focus Color**.

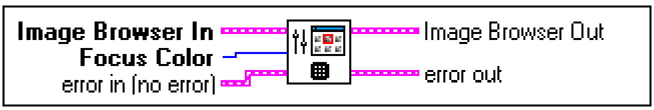


Image Browser In is the RGB image used by the browser.

Focus Color is the color used to draw the frame around a selected thumbnail image.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Image Browser Out is the resulting browser image.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ Browser Insert

Adds or inserts an image on the browser. The image is resampled to fit the size of the thumbnail size. The resampling value applied to the image is the same on both axes to maintain the image aspect.

This VI adds the thumbnail at either the last position or the first vacant position, depending on the selected mode. If all positions are occupied, an additional line of thumbnail images is added to the browser.

8 16 F RGB

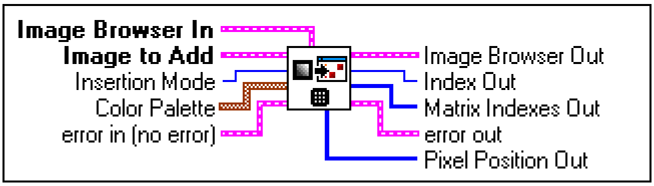


Image Browser In is the RGB image used by the browser.

Image to Add is the image to add or insert in the browser. This image can be an 8-bit, 16-bit, floating-point, RGB, or HSL image.

Insertion Mode is the mode used for inserting the thumbnail. Two modes are available:

First vacant position

Last position

Color Palette is the palette used to show an 8-bit image. If this control is not set, a gray-level palette is used.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Image Browser Out is the resulting browser image.

Index Out is the index of the position where the image is placed on the browser.

Matrix Indexes Out is the (Column, Line) position of the added thumbnail.

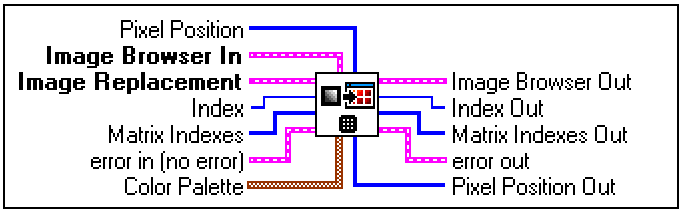
error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Pixel Position Out returns the pixel coordinates (X, Y top-left, X, Y right-bottom) defining the area where the image is inserted.

IMAQ Browser Replace

Replaces an image in the browser. The resampling ratio is computed to be as small as possible and the same on both axes. Set only one of the following controls: **Index**, **Matrix Indexes**, or **Pixel Position**.

8 16 F RGB



Pixel Position chooses to replace the image containing that position.

Image Browser In is the RGB image used by the browser.

Image Replacement is the image to replace in the browser. This image can be an 8-bit, 16-bit, floating-point, RGB, or HSL image.

Index is the position of the image to replace.

Matrix Indexes is the (Column, Line) position of the image to replace.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Color Palette is the palette used to show an 8-bit image. If this control is not set, a gray-level palette is used.

Image Browser Out is the resulting browser image.

Index Out is the index of the position where the image is placed on the browser.

Matrix Indexes Out is the (Column, Line) position of the replaced thumbnail.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Pixel Position Out returns the pixel coordinates (X, Y top-left, X, Y right-bottom) defining the area where the image is replaced.

IMAQ Browser Setup

Configures the browser. This VI sets all the required parameters that determine the aspect and the thumbnail organization on the browser.

You create and configure a browser using this VI. These tasks are mandatory because all the functions in this library act on browsers and not on standard images.

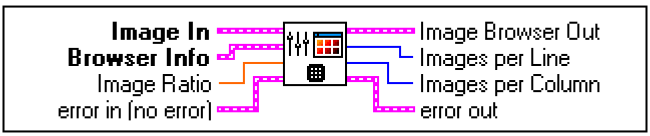


Image In is the RGB image used by the browser. This image forms the base of the browser. All thumbnails placed in the browser are pasted onto this image.

Browser Info is the cluster configuring the initial aspect of the browser. This cluster contains the following parameters:

- **Browser Size (X, Y)** is the browser size.
- **Images per Line** is the number of thumbnail images per line on the browser.
- **Background Color** is the color filling the background of the browser.
- **Frame Size** is the width of the frame that contains each thumbnail.
- **Frame Style** defines the frame style of the browser. You can choose a raised, bevelled, outline, hidden, step, or raised-outline frame style (for a width of five pixels).

Image Ratio is the width–height ratio of the thumbnail images of the browser. This value is 3/2 for PAL images and 4/3 for NTSC images.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Image Browser Out is the configured browser image.

Images per Line is the number of thumbnail images per line on the browser.

Images per Column is the number of thumbnail images per column on the browser.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

17

Inspection Tool VIs

This chapter describes the group of VIs designed for gauging, measurement, and inspection applications. You can use these VIs to accomplish the following tasks:

- Find features such as edges in an image
- Measure the distance between edges
- Compute angles between different points in an image
- Perform subpixel analysis on the image to improve the accuracy of measurements made in the image
- Establish a coordinate system for an object under inspection to track features during the translation and rotation of the object
- Specify and maintain regions of interest for typical inspection applications

The Inspection Tool VIs are found only in the Advanced version of IMAQ Vision. IMAQ Vision groups these VIs into separate palettes as follows:

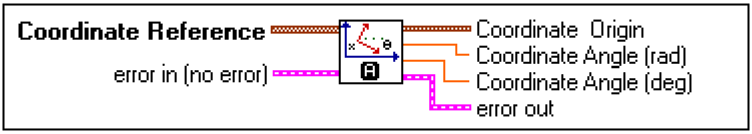
- Alignment and ROI VIs
- Caliper VIs
- LCD VIs
- Meter VIs
- Barcode VIs

Alignment and ROI Tools

These VIs are designed to handle region of interest (ROI) descriptors and establish reference coordinate systems for objects under inspection.

IMAQ Coordinate Reference

Builds a reference for any arbitrary coordinate system with respect to the image plane. The reference of the coordinate system is specified as the position of the origin of the coordinate system and the orientation of its X-axis with respect to that of the image plane.



Coordinate Reference is an array of point clusters that define the coordinate system. If two points are specified, these points are assumed to lie along the X-axis of the coordinate system and the first point is used as the origin of the coordinate axis, as shown in Figure 17-1.

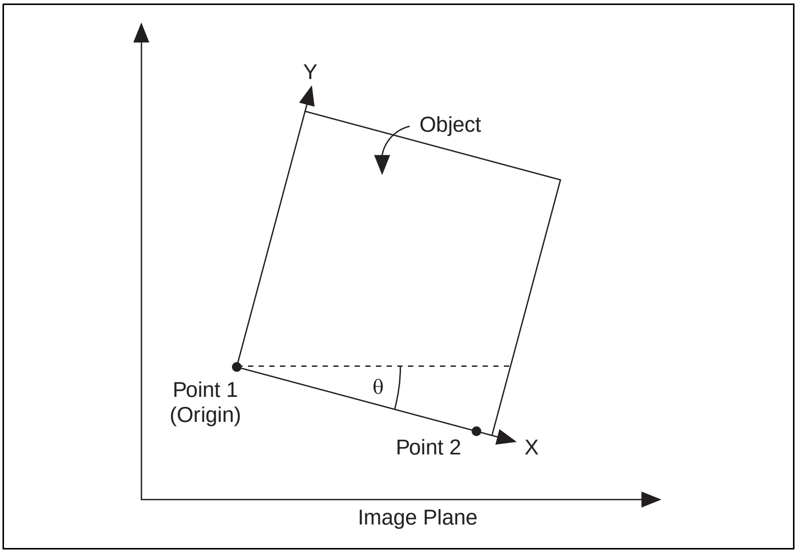


Figure 17-1. Coordinate Reference with Two Points Specified

If three points are specified, the first two points are assumed to be along the X-axis, and the third point is assumed to be on the Y-axis of the coordinate system, as shown in Figure 17-2.

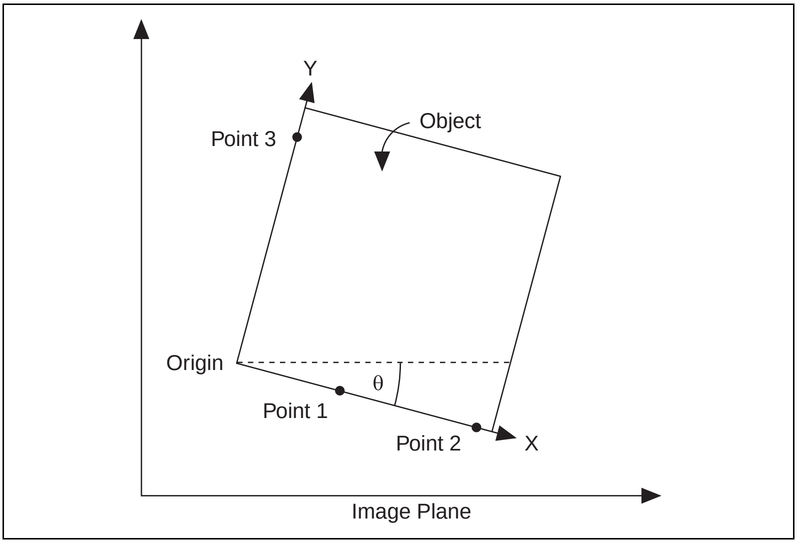


Figure 17-2. Coordinate Reference with Three Points Specified

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Coordinate Origin is a point cluster that specifies the location of the origin of the coordinate system within an image plane.

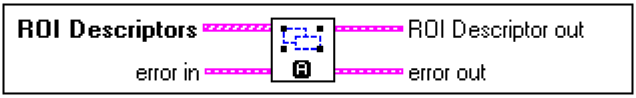
Coordinate Angle (rad) is the angle in radians formed by the X-axis of the coordinate system and the image plane.

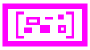
Coordinate Angle (deg) is the angle in degrees formed by the X-axis of the coordinate system and the image plane.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

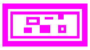
IMAQ Group ROIs

Builds a single ROI descriptor from an of array ROI descriptors.

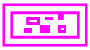




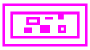
ROI Descriptors is an array of ROI descriptors that are to be merged.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.



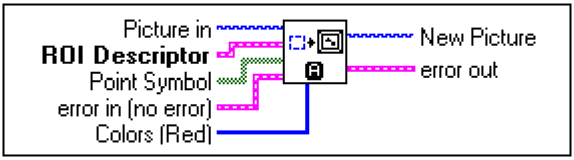
ROI Descriptor out is an ROI descriptor containing all the contours in the array of ROIs.



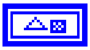
error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ ROI to Picture

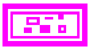
Draws the contours of a ROI descriptor in a LabVIEW or BridgeVIEW picture.



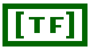
Note To be executable, this function requires the LabVIEW/BridgeVIEW Picture Functions, which are available only in the Full or Professional Development Systems.



Picture in is the picture to which you want to add the contours of the ROI descriptor. If not wired, it defaults to an empty picture.



ROI Descriptor is the descriptor that defines the region of interest in the image.



Point Symbol is the bitmap used to depict a point. The default bitmap is a 7 × 7 crosshair.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Colors (Red) is an array of numerical values coding the color of each contour of the ROI descriptor. If you require the contours drawn in the same color, enter only one color element. By default, the first color element is set to red so all the contours are drawn in red.

New Picture is the returned LabVIEW or BridgeVIEW picture.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ ROIProfile

Calculates the profile of the pixels along the boundary of an ROI descriptor. This VI returns a data type (cluster) that is compatible with a LabVIEW or BridgeVIEW graph. This VI also returns other information such as pixel statistics and the true coordinates of the ROI boundary.

8 16 F

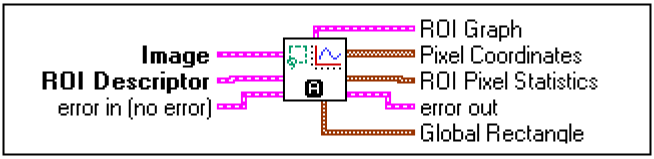


Image is the input source image used for calculating the ROI profile.

ROI Descriptor is the descriptor that defines the region of interest.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

ROI Graph is a cluster that contains the ROI profile with an X origin at 0 and an increment of 1. The cluster contains the following elements:

x0 always returns 0.

dx always returns 1.

Pixels Line returns the ROI profile calculated in an array in which elements represent the pixel values belonging to the specified vector.

Pixel Coordinates is an array consisting of the spatial coordinates of each pixel along the ROI boundary.

ROI Pixel Statistics is a cluster containing relevant information about the pixels found along the ROI boundary. This cluster contains the following elements:

Min returns the smallest pixel value found in the ROI profile.

Max returns the largest pixel value found in the ROI profile.

Mean returns the mean value of the pixels found in the ROI profile.

Var returns the standard deviation from the ROI profile.

Count returns the count of pixels found in the ROI profile.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Global Rectangle is a cluster that contains the coordinates of a bounding rectangle for the ROI in the image. This cluster includes the following parameters:

x1Left indicates the x coordinate of the top-left corner of the rectangle.

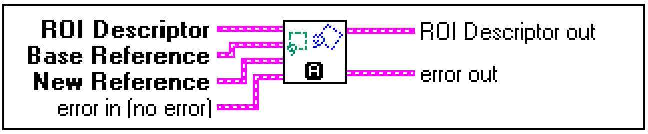
y1Top indicates the y coordinate of the top-left corner of the rectangle.

x2Right indicates the x coordinate of the bottom-right corner of the rectangle.

y2Bottom indicates the y coordinate of the bottom-right corner of the rectangle.

IMAQ Transform ROI

Rotates and translates an ROI in an image to transform the ROI from one coordinate system (**Base Reference**) to another (**New Reference**).



ROI Descriptor is the descriptor that defines the region of interest.

Base Reference is a cluster containing the following elements:

Origin is a point cluster that specifies the origin of the base-reference coordinate system.

Angle is the angle the base-reference coordinate system makes with the image coordinate system.

New Reference is a cluster containing the following elements:

Origin is a point cluster that specifies the origin of the new-reference coordinate system.

Angle is the angle the new-reference coordinate system makes with the image coordinate system.

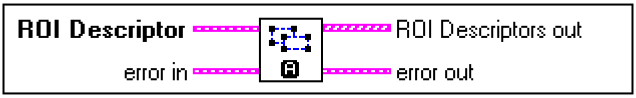
error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

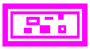
ROI Descriptor out contains the new descriptor for the transformed ROI.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

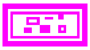
IMAQ Ungroup ROIs

Separates an ROI descriptor (describing many contours) into an array of simple ROI descriptors. Each of the ROI descriptors returned contains a single contour.

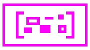




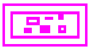
ROI Descriptor is the input ROI containing multiple contours.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.



ROI Descriptors out is the returned array of ROI descriptors. Each ROI descriptor contains a single contour.



error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

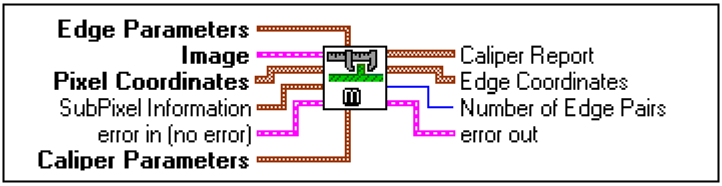
Caliper Tools

The Caliper Tool VIs detect specific features in an image. The features you can detect include edges, peaks, and rotational shifts. All features are detected along regions of interest that you specify.

IMAQ Caliper Tool

Finds edge pairs along a specified path in the image. This VI performs an edge extraction similar to IMAQ Edge Tool then finds edge pairs based on specified criteria such as the distance between the leading and trailing edges, edge contrasts, and so on.

8 16 F



Edge Parameters is a cluster defining the characteristics of the filter used to detect edges. This cluster consists of the following parameters:

contrast specifies the threshold for the contrast of the edge. Only edges with a contrast greater than this value are used in the detection process. Contrast is defined as the difference between the average pixel intensity before the edge and the average pixel intensity after the edge.

filter width specifies the number of pixels that are averaged to find the contrast at either side of the edge.

Steepness specifies the slope of the edge. This value represents the number of pixels that correspond to the transition area of the edge.

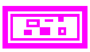


Image is the input source image. 8-bit, 16-bit, and floating-point images are supported.

Pixel Coordinates is an array consisting of the spatial coordinates of pixels in the image.

SubPixel Information is a cluster containing the following parameters for subpixel analysis:

Interpolation Type specifies the method used to perform the interpolation. The following two values are possible:

Quadratic

Cubic Spline

SubPixel Accuracy specifies the number of samples to obtain from a pixel. A subpixel accuracy of one-fourth specifies that each pixel is split into four subpixels.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Caliper Parameters specifies the following parameters for finding edge pairs:

Polarity specifies the polarity of the leading and trailing edge in the search process. The following values are possible:

None	Polarity of the edges is ignored.
Dark-Light	The leading edge has a dark-to-light polarity, and the trailing edge has a light-to-dark polarity.
Light-Dark	The leading edge has a light-to-dark polarity, and the trailing edge has a dark-to-light polarity.
Dark-Dark	The leading edge has a dark-to-light polarity, and the trailing edge has a dark-to-light polarity.
Light-Light	The leading edge has a light-to-dark polarity, and the trailing edge has a light-to-dark polarity.

Separation specifies the desired separation between the edge pairs. Edge pairs with separation greater or less than this value within some tolerance are ignored. If this parameter is set to 0, all edge pairs are found.

Separation Deviation specifies a tolerance value for the separation between the edges. This value influences the score of the detected edge pairs.

Caliper Report is an array of clusters that contain the following information about the detected edge pairs:

- **Separation** is the computed distance in pixels between the edges in the edge pair.
- **Edge1 Position** is the location of the leading edge.
- **Edge1 Contrast** is the contrast of the leading edge.
- **Edge2 Position** is the location of the trailing edge.
- **Edge2 Contrast** is the contrast of the trailing edge.
- **Score** is unused.

Edge Coordinates is an array of point clusters consisting of the spatial coordinates of the detected edges.

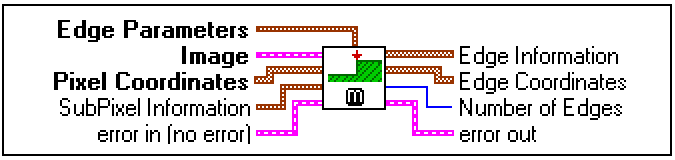
Number of Edge Pairs returns the number of detected edge pairs.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ Edge Tool

Finds edges along a path defined in the image. Edges are determined based on their contrast, width, and steepness.

8 16 F



Edge Parameters is a cluster defining the characteristics of the filter used to detect the edges. This cluster consists of the following parameters:

Contrast specifies the threshold for the contrast of the edge. Only edges with a contrast greater than this value are used in the detection process. Contrast is defined as the difference between the average pixel intensity before the edge and the average pixel intensity after the edge.

Filter width specifies the number of pixels that are averaged to find the contrast at either side of the edge.

Steepness specifies the slope of the edge. This value represents the number of pixels that correspond to the transition area of the edge.

Image is the input source image. 8-bit, 16-bit, and floating-point images are supported.

Pixel Coordinates is an array consisting of the spatial coordinates of pixels in the image.

SubPixel Information is a cluster containing the following parameters for subpixel analysis:

Interpolation Type specifies the method used to perform the interpolation. The following two values are possible:

Quadratic

Cubic Spline

SubPixel Accuracy specifies the number of samples to obtain from a pixel. A subpixel accuracy of one-fourth specifies that each pixel is split into four subpixels.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Edge Information is an array that contains the following information about each detected edge:

Position indicates the position of the edge from the first point in the **Pixel Coordinates** array.

Contrast specifies the intensity contrast at the edge.

Polarity indicates the polarity of the edge. The following two values are possible:

Rising edge (going from dark to light)

Falling edge (going from light to dark)

Score is unused.

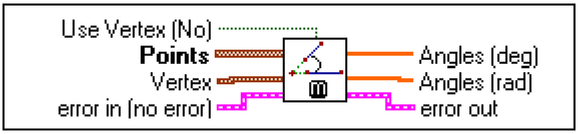
Edge Coordinates is an array of point clusters consisting of the spatial coordinates of the detected edges.

Number of Edges specifies the number of edges found.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ Get Angles

Computes the angles formed by sets of four points in an image or between sets of two points and a common vertex.



Use Vertex (No) specifies whether a vertex point is used while computing the angle.

Points is an array of point clusters. If the vertex point is not used, four points at a time are considered while computing the angle, as shown in the left-hand side of Figure 17-3. If **Use Vertex** is TRUE, the angle made by two consecutive points in the array and the vertex is computed, as shown in the right-hand side of Figure 17-3.

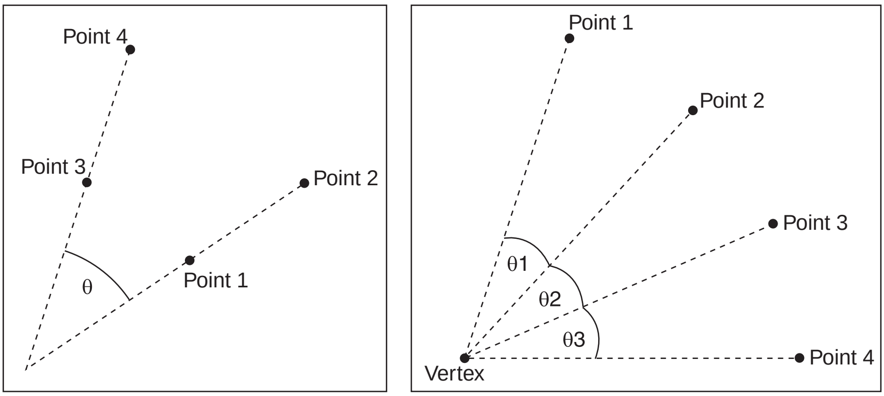
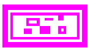


Figure 17-3. Effect of Vertex Point on IMAQ Get Angles

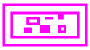
Vertex specifies the vertex point to use if the **Use Vertex** option is TRUE.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Angles (deg) is an array containing the computed angles in degrees.

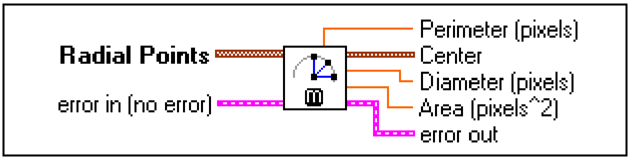
Angles (rad) is an array containing the computed angles in radians.



error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

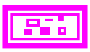
IMAQ Get Circle

Computes the center of the circle described by three or more points located on its circumference. It also returns the diameter of the circle in pixels, the area of the disk, and its perimeter.



Radial Points is an array of point-coordinate clusters. This array must contain three or more point-coordinate clusters corresponding to three or more points on the circumference of the circle.

Three points on the circumference of the circle determine a unique center. When you provide more than three points, the function computes the barycenter of a set of centers, where each center is given by a group of three points.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Perimeter (pixels) is the perimeter of the circle in pixels.

Center indicates the coordinates of the center of the circle.

Diameter (pixels) is the diameter of the circle.

Area (pixels^2) is the area of the disk in square pixels.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ Interpolate 1D

Resamples an array of pixels from an image using multiple interpolation functions. Use this VI to perform subpixel analysis of pixel profiles in the image.

8 16 F

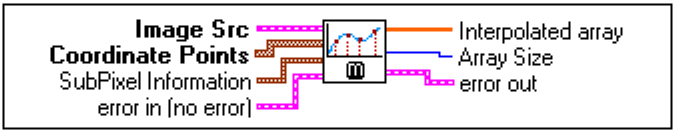


Image Src is the input source image used for calculating the ROI profile.

Coordinate Points is an array consisting of the spatial coordinates of pixels in the image.

SubPixel Information is a cluster containing the following parameters for subpixel analysis:

Interpolation Type specifies the method used to perform the interpolation. The following three values are possible:

Bi-Linear

Quadratic

Cubic Spline

SubPixel Accuracy specifies the number of samples to obtain from a pixel. A subpixel accuracy of one-fourth specifies that each pixel is split into four subpixels.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Interpolated array contains the resampled pixel data.

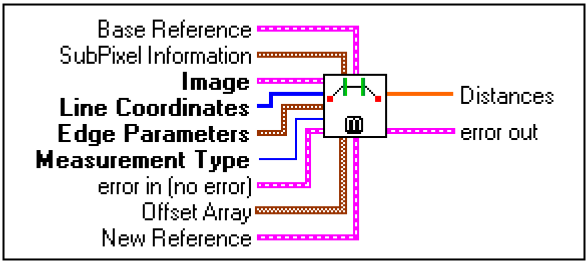
Array Size specifies the size of the interpolated array.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ Line Gauge Tool

Measures the distance between selected edges with high-precision subpixel accuracy. You can use different measurement modes with this VI, such as measuring distances between points and edges and vice versa. This VI also can step and repeat its measurements across the image. Depending on the type of the measurement, this VI first determines the relevant edges in the image before computing the distances.

8 16 F



Base Reference is a cluster containing the following elements:

Origin is a point cluster that specifies the origin of the base-reference coordinate system.

Angle is the angle the base-reference coordinate system makes with the image coordinate system.

SubPixel Information is a cluster containing the following parameters for subpixel analysis:

Interpolation Type specifies the method used to perform the interpolation. The following two values are possible:

Quadratic

Cubic Spline

SubPixel Accuracy specifies the number of samples to obtain from a pixel. A subpixel accuracy of one-fourth specifies that each pixel is split into four subpixels.

Image is the input source image. 8-bit, 16-bit, and floating-point images are supported.

Line Coordinates is an array specifying the pixel coordinates that form the end points of the line.

Edge Parameters is a cluster defining the characteristics of the filter used to detect edges. This cluster consists of the following parameters:

contrast specifies the threshold for the contrast of the edge. Only edges with a contrast greater than this value are considered in the detection process. Contrast is defined as the difference between the average pixel intensity before the edge and the average pixel intensity after the edge.

filter width specifies the number of pixels that are averaged to find the contrast at either side of the edge.

Steepness specifies the slope of the edge. This value represents the number of pixels that correspond to the transition area of the edge.

Measurement Type specifies the type of measurement you want to perform. The following values are possible:

Edge To Edge	Measures the distance between two edge locations in an image.
Edge To Point	Measures the distance between an edge in the image and another point in the image.
Point To Edge	Measures the distance between a point and an edge in the image.
Point To Point	Measures the distance between two points in the image.

Note Subpixel information is not used when measuring distances between points.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Offset Array is an array of point clusters that specifies the shift in the start and end points of the pixel coordinates. The **Offset Array** is used for repeating the measurement across the image.

New Reference is a cluster containing the following elements:

Origin is a point cluster that specifies the origin of the new-reference coordinate system.

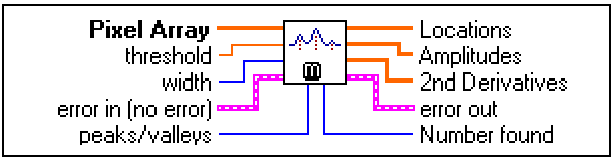
Angle is the angle the new-reference coordinate system makes with the image coordinate system.

Distances is an array containing the measured distances.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ Peak-Valley Detector

Finds the location, amplitude, and second derivative of peaks or valleys in the input array. This VI is based on an algorithm that fits a quadratic polynomial to sequential groups of data points. The number of data points used in the fit is specified by **width**. For each peak or valley, the quadratic fit is tested against the threshold level. Peaks with heights lower than the threshold or valleys with troughs higher than the threshold are ignored.



Pixel Array contains the pixel data to be processed.

threshold rejects peaks or valleys that are too small. Any peak found with a fitted amplitude that is less than **threshold** is ignored. Valleys are ignored if the fitted trough is greater than **threshold**.

width specifies the number of consecutive data points to use in the quadratic least-squares fit. The value should be no more than approximately half of the half-width of the peaks or valleys and can be much smaller for noise-free data. Large widths can reduce the apparent amplitude of peaks and shift the apparent location.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

peaks/valleys specifies whether to look for peaks (positive-going bumps) or valleys (negative-going bumps). The settings for this control are 0 (peaks) and 1 (valleys).

Locations is an array containing the locations of peaks or valleys found in the current block of data. Locations are reported in indices from the beginning of processing.

Amplitudes is an array containing the amplitudes of peaks or valleys found in the current block of data.

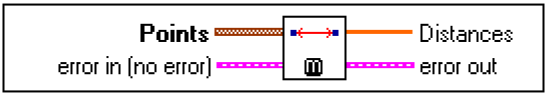
2nd Derivatives is an array containing the second derivatives of peaks or valleys found in the current block of data.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Number found is the number of peaks or valleys found in the current block of data. **Number found** is the size of the **Locations**, **Amplitudes**, and **2nd Derivatives** arrays.

IMAQ PointDistances

Computes the distance, in pixels, between consecutive pairs of points.



Points is an array of point-coordinate clusters.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Distances is an array containing the computed distances. Distance *i* is computed between the points *i* and (*i*+1).

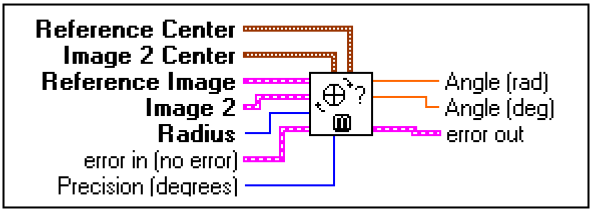
error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ Rotation Detect

Detects the rotational shift between two images, generally between a reference image, with the part being inspected at a known orientation, and another image that has the part in an unknown position.

This VI extracts pixel values around a circular region in the reference image. These values then are compared to the same region in **Image 2**. The algorithm looks for the rotational shift between those two samples. To speed up the process, adjust the requested precision of the result.

8 16 F



Reference Center contains coordinates of the center of a circular region in the reference image. Pixel values along this region in the reference image are used as features for computing the rotational shift.

Image 2 Center contains coordinates of the center of a circular region in **Image 2**. The pixel values along this region in **Image 2** are used to compute the rotational shift.

Reference Image is the reference to the image containing the part to be inspected at a known position.

Image 2 is the reference to the image that contains the part to be inspected at an unknown rotational shift.

Radius is the radius of the circular regions in both images. Pixel values along this region in the two images are used to detect the rotation angle between the two images.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Precision (degrees) is the sampling period of the pixel values extracted from the circular region. The speed of this VI is affected directly by the sampling period. If the sampling period is high (the number of samples along the circular region are few), the processing speed of the VI increases at the cost of reduced accuracy in the computed rotational shift. In many cases, a precision higher than five degrees is not needed to position the regions of inspection of a part. The default is 5.

Angle (rad) returns the angle in radians corresponding to the shift giving the best match between the two samples.

Angle (deg) returns the angle in degrees corresponding to the shift giving the best match between the two samples.

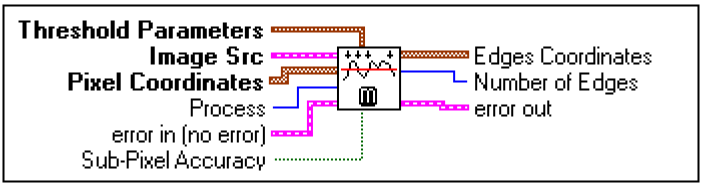
error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ Simple Edge

Finds step edges along an array of pixel coordinates. This VI can return the first, both the first and the last, or all the edges found.

Use the **Threshold Parameters** to determine whether a change in the pixel values is considered as an edge. The main criterion is a threshold level on the pixel values. This threshold value can be either relative or absolute. Relative threshold level is specified as a percentage of the pixel range found in the pixel path. Absolute value is based directly on the pixel values. You can compute the location of the edges with subpixel accuracy.

8 16 F



Threshold Parameters is a cluster containing information used to determine whether a change in pixel value is considered as an edge. This cluster contains the following parameters:

Level Type is an enumerated type control determining whether the **Threshold Level** is expressed in absolute or relative values. The default is absolute value.

Threshold Level can be either absolute or relative. Absolute threshold is based on the pixel values. Relative threshold is expressed as a percentage of the pixel-value range found along the path defined by the pixel coordinates.

Hysteresis can be either absolute or relative. This parameter determines the difference in threshold level between a rising and a falling edge, enabling accurate detection in noisy images.

Image Src is the reference to the image that you need to process.

Pixel Coordinates is an array containing the coordinates of the pixels to be inspected. You can obtain the pixel coordinates from an ROI descriptor using the IMAQ ROIProfile VI.

Process determines the type of search. The VI can return the first edge, both the first and the last edge, or all edges found along the ROI path.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Sub-Pixel Accuracy determines the accuracy required for the location of the edge coordinates. Setting this control to FALSE enables fast edge detection. A subpixel localization of the edges is obtained when this control is TRUE. The subpixel result is computed using a local quadratic interpolation. The default is FALSE.

Edges Coordinates returns the coordinates of the detected edges. If the **Sub-Pixel Accuracy** control is set to TRUE, the result is given with subpixel accuracy.

Number Of Edges returns the number of edges found.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

LCD

The VIs in this library simplify and accelerate the development of applications that require reading values from seven-segment displays. Use these VIs to extract seven-segment digit information from an image.

The reading process consists of two phases:

- A learning phase during which the user specifies an area of interest in the image to locate the seven-segment display
- A reading phase during which the area specified by the user is analyzed to read the seven-segment digit

This library provides the high-level vision processes required for recognizing and reading seven-segment digit indicators. The VIs in this library are designed for seven-segment displays that use either LCDs (liquid crystal displays) or LEDs (seven segments composed of light-emitting diodes or electroluminescent indicators).

The tools in this library can perform the following tasks:

- Detect the area around each seven-segment digit from a rectangular area that contains multiple digits
- Read the value of a single digit
- Read the value, the sign, and the decimal separator of the displayed number

LCD Algorithm Limits

This section explains the limit conditions of the algorithm used for the LCD VIs. Four factors can cause a bad detection:

- Very high horizontal or vertical light drift
- Very low contrast between the background and the segments
- Very high level of noise
- Very low resolution of the image

Each of these factors is quantified to indicate when the algorithm presented here might not give accurate results.

Light drift is quantified by the difference between the average pixel values at the top left and the bottom right of the background of the LCD screen. Detection results might be inaccurate if light drift is greater than 90 in 8-bit images.

Contrast is measured as the difference between the average pixel values in a rectangular region in the background and a rectangular region in a segment. This difference must be more than 30 in 8-bit images (256 gray levels) to obtain accurate results.

Noise is defined as the standard deviation of the pixel values contained in a rectangular region in the background. It must be less than 15 in 8-bit images (256 gray levels) to obtain accurate results.

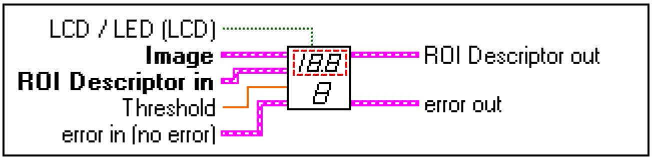
Each digit must be larger than 18 × 12 pixels to obtain accurate results.

IMAQ Get LCD ROI

Calculates the area of interest of each digit from a rectangular area of interest around the whole indicator, which can contain multiple digits. Use this VI first in a calibration phase. To find the area of each digit, all the segments of the indicator must be activated.

This VI is designed for LCD and electroluminescent indicators. It is resistant to light drift. It returns the area of interest containing the global rectangle for each digit. Its type is ROI Descriptor, a standard type in IMAQ Vision for G.

8 16 F

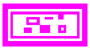


LCD/LED (LCD) is a Boolean that controls the indicator type. FALSE corresponds to LCD indicators, and TRUE corresponds to LED indicators. The default is FALSE.

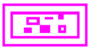
Image is the reference to the image containing the seven-segment display.

ROI Descriptor in is the descriptor of the area of interest drawn by the user that corresponds to the global rectangle of the indicator. It is compatible with the type ROI Descriptor of IMAQ Vision for G.

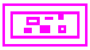
Threshold is the threshold that determines whether a segment is ON or OFF. A segment is considered ON if the standard deviation of the pixels along a line profile across the segment is greater than this threshold. Increase this value when using images with high contrast. Lower this value when using images with low contrast. The default is 8.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.



ROI Descriptor out contains the global rectangles of each digit in the display.

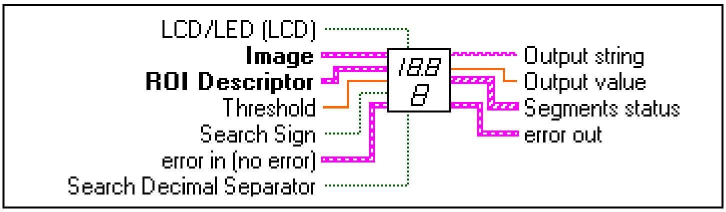


error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ Read LCD

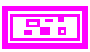
Reads an LCD display containing as many as four digits in an input image using the descriptor of the ROI calculated by the IMAQ Get LCD ROI VI. IMAQ Read LCD returns the number as a double and as a string. The **Segments status** output indicates whether the number is recognized properly or any of the digits are not recognized fully. This output returns the status of each of the digits in the display. This VI reads LCD or LED indicators and tolerates light drift.

8 16 F



LCD/LED (LCD) is a Boolean that controls the indicator type. FALSE corresponds to LCD indicators, and TRUE corresponds to LED indicators. The default is FALSE.

Image is the reference to the image that contains the seven-segment display.



ROI Descriptor is the descriptor of the area of interest calculated by the IMAQ Get LCD ROI VI that contains the quadrilaterals surrounding each digit.

Threshold is the threshold that determines whether a segment is ON or OFF. A segment is considered ON if the standard deviation of the pixels along a line profile across the segment is greater than this threshold. Increase this value when using images with high contrast. Lower this value when using images with low contrast. The default is 8.

Search Sign indicates whether the algorithm must read the sign of the indicator.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Search Decimal Separator determines whether to look for a decimal separator after each digit. The default is FALSE.

Output string contains the number that has been read from the display. If the decimal separator is ON and the control **Search Decimal Separator** is TRUE, it is represented in the string by a dot (`.`) symbol.

Output value contains the value of the number represented by the display.

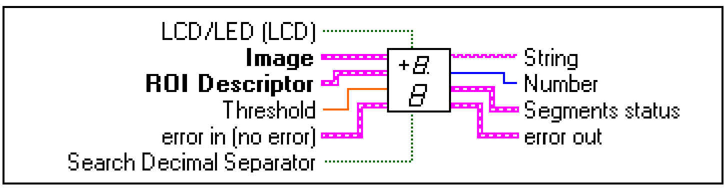
Segments status is an array of clusters containing the state of the segments of each digit.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ Read Single Digit

Reads a single seven-segment digit from a rectangular area of interest drawn around this digit. This VI reads LCD and LED indicators. It returns a value as a string and a cluster containing the state of each segment (ON or OFF).

8 16 F



LCD/LED (LCD) is a Boolean that controls the indicator type. FALSE corresponds to LCD indicators, and TRUE corresponds to LED indicators. The default is FALSE.

Image is the reference to the image containing the seven-segment display.

ROI Descriptor is the descriptor of the area of interest drawn by the user that corresponds to a quadrilateral surrounding the digit. The type of ROI must be rectangular or polygonal.

Threshold is the threshold that determines whether a segment is ON or OFF. A segment is considered ON if the standard deviation of the pixels along a line profile across the segment is greater than this threshold. Increase this value when using images with high contrast. Lower this value when using images with low contrast. The default is 8.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Search Decimal Separator determines whether to look for a decimal separator after each digit. The default is FALSE.

String contains the value of the digit and decimal separator. If the decimal separator is ON and the control **Search Decimal Separator** is TRUE, it is represented in the string by a dot (**.**) symbol.

Number contains the value represented by the digit.

Segments status is a cluster of Booleans containing the state of each segment of the digit. The order of the cluster is illustrated below in Figure 17-4.

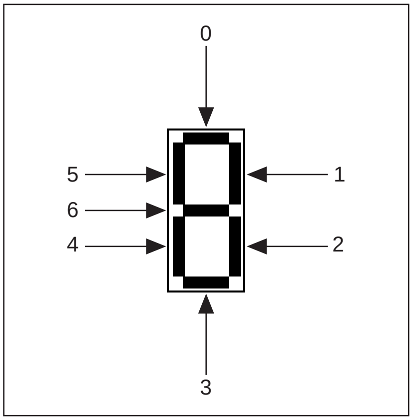


Figure 17-4. Ordering of the Segments Status Cluster in IMAQ Read Single Digit

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Meter

The VIs in this library simplify and accelerate the development of applications that require reading values from meters or gauges. These VIs provide high-level vision processes to extract the position of a meter or gauge needle. You can use this information to build different applications such as the calibration of a gauge.

The recognition process consists of two phases:

- A learning phase during which the user must specify the extremities of the needle
- An analysis phase during which the current position of the needle is determined

The Meter VIs are designed to work with meters or gauges that have either a dark needle on a light background or a light needle on a dark background.

You can use the VIs in this library to compute the base of the needle and its extremities from an area of interest indicating the initial and the full-scale position of the needle. You then can use these VIs to read the position of the needle using parameters computed earlier.

Meter Algorithm Limits

This section explains the limit conditions of the algorithm used for the Meter VIs. The algorithm is fairly insensitive to light variations.

The position of the base of the needle is very important in the detection process. Carefully draw the lines that indicate the initial and the full-scale position of the needle. The coordinates of the base and of the points of the arc curved by the tip of the needle are computed by IMAQ Get Meter and IMAQ Get Meter 2. These coordinates are not computed again in IMAQ Read Meter, so do not move the meter after the calibration phase.

IMAQ Get Meter

Calibrates the meter or gauge using the initial and the full-scale position of the needle. It calculates the position of the base of the needle and the arc traced by the tip of the needle.

Use this VI with VU-meters and gauges that have either a dark needle on a light background or a light needle on a dark background.

8 16 F

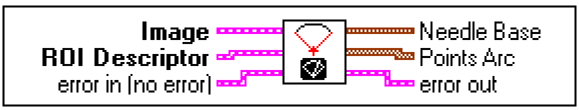


Image is the reference to the image containing the gauge or meter.

ROI Descriptor is the descriptor drawn by the user to identify the initial position of the needle and the full-scale position of the needle. The origin of each line must be the tip of the needle.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Needle Base is a cluster containing the coordinates of the base of the needle.

Points Arc is an array of clusters containing the coordinates of the points of the arc traced by the tip of the needle.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ Get Meter 2

Calibrates the meter or gauge using three points on the meter: the base of the needle, the tip of the needle at its initial position, and the tip of the needle at its full-scale position. It calculates the position of the points along the arc covered by the tip of the needle.

Use this VI with VU-meters or gauges equipped with a dark needle on a light background or with a light needle on a dark background.

8 16 F

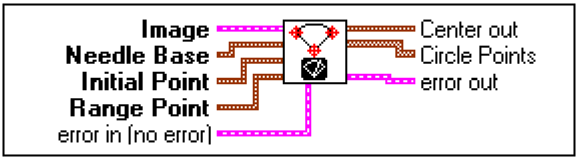


Image is the reference to the image containing the meter.

Needle Base is a cluster containing the coordinates of the base of the needle.

Initial Point is a point cluster containing the coordinates of the tip of the needle in its initial position.

Range Point is a point cluster containing the coordinates of the tip of the needle in its full-scale position.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

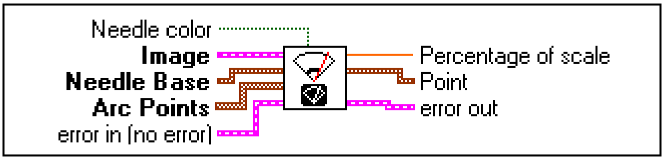
Center out is a point cluster containing the coordinates of the rotation center of the needle.

Circle Points is an array of point clusters containing the coordinates of the points along the arc traced by the tip of the needle.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ Read Meter

Reads the position of the needle using the base of the needle and the array of the points of the arc traced by the tip of the needle. This VI reads VU-meters with a dark needle on a light background or with a light needle on a dark background. It returns the location of the tip of the needle as a percentage of the full range of the meter.



Needle color is a Boolean that controls the color of the needle. FALSE corresponds to a dark needle on a light background, and TRUE corresponds to a light needle on a dark background. The default is FALSE.

Image is the reference to the image containing the meter.

Needle Base is a cluster containing the coordinates of the base of the needle.

Arc Points is an array of clusters containing the coordinates of the points of the arc traced by the tip of the needle.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Percentage of scale contains the angle of the needle as a percentage of the full-scale angle.

Point is a cluster containing the coordinates of the tip of the needle.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Barcodes

The VIs in this library simplify and accelerate the development of applications that require reading of values from 1D barcodes.

You can use these VIs to decode characters encoded into 1D barcodes. IMAQ Vision for G currently supports the following barcodes: Code 25, Code 39, Code 93, Code 128, EAN 8, EAN 13, Codabar, MSI, and UPC A.

The process used to recognize the barcodes consists of two phases:

- A learning phase in which the user specifies an area of interest in the image that helps to localize the region occupied by the barcode.
- The recognition phase during which the region specified by the user is analyzed to decode the barcode.

Barcode Algorithm Limits

This section explains the limits of the algorithm used to decode a barcode. The following factors can cause errors in the decoding process:

- Very low resolution of the image
- Very high horizontal or vertical light drift
- The contrast along the bars of the image
- High level of noise

Each of these factors is quantified to give the limit conditions of the VIs. The limit conditions are different for barcodes that have two different widths of bars and spaces (Code 39, Codabar, Code 25, and MSI code) and for barcodes that have four different widths of bars and spaces (Code 93, Code 128, EAN 13, EAN 8, and UPC A).

The resolution of an image is determined by the width of the smallest bar and space. These widths must be at least 3 pixels for all barcodes.

Light drift is quantified by the difference between the average of the gray level of the left (upper) line and the right (bottom) line of the background of the barcode. Decoding inaccuracies can occur if the light drift is greater than 120 for barcodes with two different widths of bars and spaces and greater than 100 for barcodes with four different widths of bars and spaces.

In overexposed images, the gray levels of the wide and narrow bars in the barcode tend to differ. Decoding results might not be accurate if the difference in gray levels between bars of two different widths is less than 80 for barcodes with two different widths of bars and spaces and less than 100 for barcodes with four different widths of bars and spaces.

Consider the difference in gray levels between the narrow bars and the wide bars. Notice that the narrow bars are scarcely visible. If this difference of gray level exceeds 115 on 8-bit images (256 gray levels) for barcodes with two different widths of bars and spaces and 100 for barcodes with four different widths of bars and spaces, the results of reading might be corrupted.

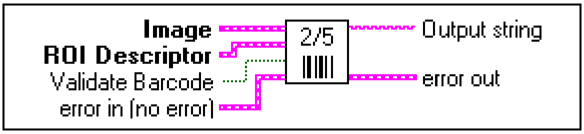
Noise is defined as the standard deviation of a rectangular region of interest drawn in the background. It must be less than 57 for barcodes with two different widths of bars and spaces and less than 27 for barcodes with four different widths of bars and spaces.

Notice that reflections on the barcode can introduce errors in the value read from the barcode. Bars and spaces masked by the reflection produce errors.

IMAQ Read Cod25

Reads a barcode and decodes the numeric characters encoded with the interleaved 2 of 5 standard barcode. It returns the results as a string.

8 16 F



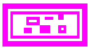
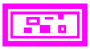
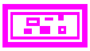


Image is the reference to the image that contains the barcode.



ROI Descriptor is the descriptor of the area of interest drawn by the user. This area of interest must be rectangular and must cross the barcode.

Validate Barcode (False) determines whether to check for decoding errors using the check character in the barcode. The default is FALSE.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Output string contains the decoded value of the barcode.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ Read Cod39

Reads a barcode and decodes the alphanumeric characters encoded with the standard Code 39 barcode. It returns the results as a string.

8 16 F

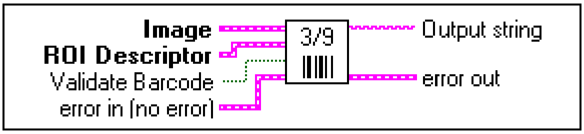


Image is the reference to the image that contains the barcode.

ROI Descriptor is the descriptor of the area of interest drawn by the user. This area of interest must be rectangular and must cross the barcode.

Validate Barcode (False) determines whether to check for decoding errors using the check character in the barcode. The default is FALSE.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

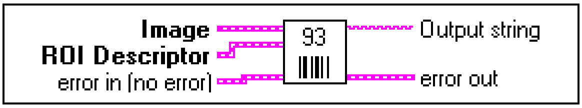
Output string contains the decoded value of the barcode.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ Read Cod93

Reads a barcode and decodes the alphanumeric characters encoded with the standard Code 93 barcode. It returns the results as a string.

8 16 F



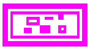
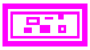
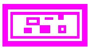


Image is the reference to the image that contains the barcode.

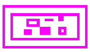


ROI Descriptor is the descriptor of the area of interest drawn by the user. This area of interest must be rectangular and must cross the barcode.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Output string contains the decoded value of the barcode.



error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ Read Cod128

Reads a barcode and decodes the alphanumeric characters encoded with the standard Code 128 barcode. It returns the results as a string.

8 16 F

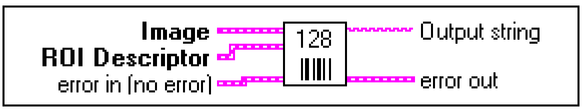


Image is the reference to the image that contains the barcode.

ROI Descriptor is the descriptor of the area of interest drawn by the user. This area of interest must be rectangular and must cross the barcode.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Output string contains the decoded value of the barcode.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ Read Codabar

Reads a barcode and decodes the alphanumeric characters encoded with the standard Codabar barcode. It returns the results as a string.

8 16 F

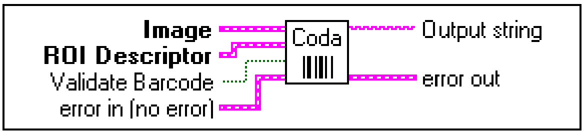


Image is the reference to the image that contains the barcode.

ROI Descriptor is the descriptor of the area of interest drawn by the user. This area of interest must be rectangular and must cross the barcode.

Validate Barcode (False) determines whether to check for decoding errors using the check character in the barcode. The default is FALSE.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Output string contains the decoded value of the barcode.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ Read EAN8

Reads a barcode and decodes the numeric characters encoded with the standard EAN 8 barcode. It returns the results as a string.

8 16 F

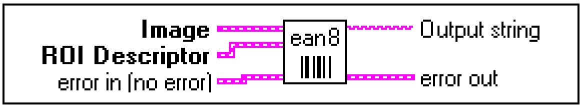


Image is the reference to the image that contains the barcode.

ROI Descriptor is the descriptor of the area of interest drawn by the user. This area of interest must be rectangular and must cross the barcode.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Output string contains the decoded value of the barcode.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ Read EAN13

Reads a barcode and decodes the numeric characters encoded with the EAN 13 barcode. It returns the results as a string.

8 **16** **F**

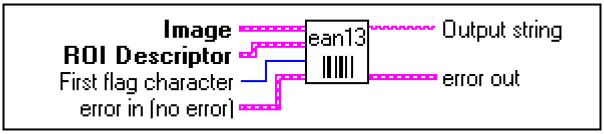


Image is the reference to the image that contains the barcode.

ROI Descriptor is the descriptor of the area of interest drawn by the user. This area of interest must be rectangular and must cross the barcode.

First flag character is the first character of the code. This user-specified character determines how the other characters are encoded. The default is 3.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

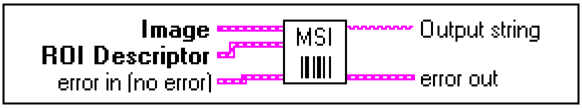
Output string contains the decoded value of the barcode.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ Read MSI

Reads a barcode and decodes the numeric characters encoded with the standard MSI barcode. It returns the results as a string.

8 16 F



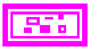
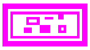
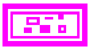


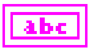
Image is the reference to the image that contains the barcode.



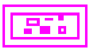
ROI Descriptor is the descriptor of the area of interest drawn by the user. This area of interest must be rectangular and must cross the barcode.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.



Output string contains the decoded value of the barcode.

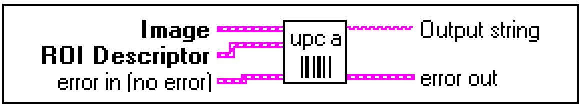


error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ Read UPC A

Reads a barcode and decodes the numeric characters encoded with the standard UPC A barcode. It returns the results as a string.

8 **16** **F**



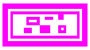
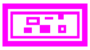
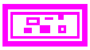


Image is the reference to the image that contains the barcode.

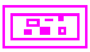


ROI Descriptor is the descriptor of the area of interest drawn by the user. This area of interest must be rectangular and must cross the barcode.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Output string contains the decoded value of the barcode.



error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

18

Searching and Matching VIs

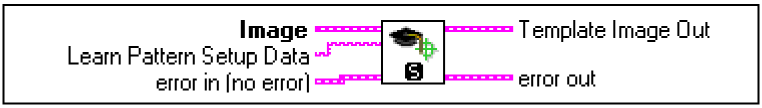
This chapter describes the searching and matching VIs in IMAQ Vision. These VIs are found only in the Advanced version of IMAQ Vision.

The first step in performing a search or match operation is creating a template or model image. This template image represents the pattern or sub-image that you want to locate in other images during the match process. After you create a template image, you need to add extra information that succinctly describes the template for pattern matching to the image. This is done during the learn phase using the IMAQ Learn Pattern VI. After IMAQ Vision learns the template, the IMAQ Match Pattern VI can use it during the match phase to locate its presence and position in other images.

IMAQ Learn Pattern

Creates a description of the template image that you are going to look for during the matching stage. This description data is appended to the input template image. During the matching step, the template descriptor is extracted from the template image and used to search for the template in the match image.

8



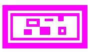
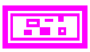


Image is a reference to the template image that you want to search for in the image.

Learn Pattern Setup Data is a string that contains information obtained from the IMAQ Setup Learn VI. If this input is not connected, default parameters are used during the learning stage.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

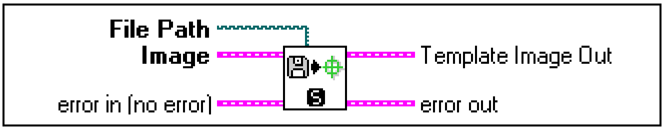
Template Image Out is a reference to the template image. This image contains the data defining the template pattern for the matching stage.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ Load Template Image

Restores a template image stored in a file via the IMAQ Save Template Image VI.

8



File Path is the complete pathname of a file that contains the template image.

Image is the reference to the image to which the data from the template image file is applied.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

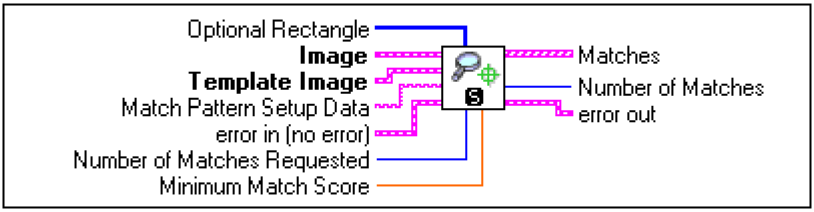
Template Image Out is the reference to the image structure containing the data read from the template image file.

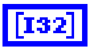
error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ Match Pattern

Searches for a pattern, or template image, in the input **Image**. Use the IMAQ Learn Pattern VI prior to using this VI to ensure that the template image has been configured for the match stage.

8





Optional Rectangle designates a rectangular region (Left/Top/Right/Bottom) within an image in which to search for the template pattern. If this array is empty the entire image is examined.

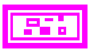
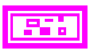


Image is a reference to the image in which the template image is to be located.

Template Image is a reference to the image to be located during the match process. The template image is the image obtained from the output of the IMAQ Learn Pattern VI.

Match Pattern Setup Data is a string that contains information obtained from the IMAQ Setup Match VI. If this input is not connected, default parameters are used during the learning stage.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Number of Matches Requested is the number of valid matches expected.

Minimum Match Score is the smallest score a match must have to be considered valid.

Matches is a cluster that returns the following elements:

Position (x, y) is a cluster corresponding to the center of the template.

Angle (degrees) is the rotation angle of the template in the source image at the current match location.

Scale (%) is the change in percentage size of the matched template from the original template.

Score (0-1000) ranks the match results on a scale of 0 to 1000, where

0	no match
1000	perfect match

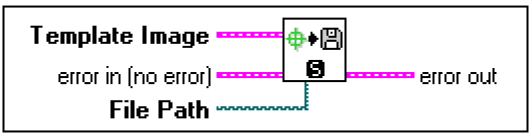
Bounding Box is an array of five points that define the boundary of the template pattern in the image.

Number of Matches is the number of instances of the template pattern found in the image based on the input conditions.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ Save Template Image

Saves the template image to a file in PNG format. Only the image portion of the template image is accessible to the user. The extra information that is stored in this file and is used to locate the template is not accessible to the user. You can use the IMAQ Load Template Image VI to recover the original template image from the saved file.



Template Image is the reference to the image structure to be written as a template image file.

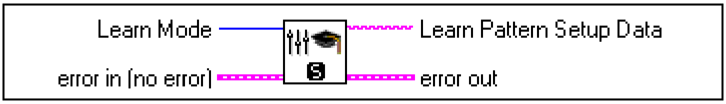
error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

File Path is the complete pathname, including the drive, directory, and filename of the file to be written.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ Setup Learn Pattern

Sets parameters that are used internally during the learning stage. Run this VI before using the IMAQ Learn Pattern VI.



Learn Mode specifies the mode to use when setting up a learn pattern. You can choose from the following values:

All extracts information for shift and rotation invariant matching.

Shift Information (Default) extracts information for shift invariant matching.

Rotation Information extracts information for rotation invariant matching.

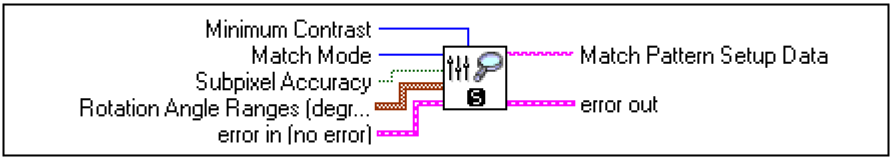
error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

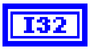
Learn Pattern Setup Data is a string that contains information pertaining to the setup parameters selected for the learning stage. Connect this output to the IMAQ Learn Pattern VI.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

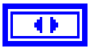
IMAQ Setup Match Pattern

Sets parameters that are used internally during the matching process. Run this VI before using the IMAQ Match Pattern VI.





Minimum Contrast specifies the minimum contrast expected in the image. Contrast is defined as the difference between the maximum pixel value and the minimum pixel value in the image.

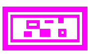


Match Mode specifies the technique to use when looking for the template pattern in the image. You can choose from the following values:

Shift Invariant (default)	Searches for the template pattern in the image, assuming it is not rotated more than ± 4 degrees
Rotation Invariant	Searches for the template in the image with no restriction on the rotation of the template

Subpixel Accuracy (T/F) determines whether the match results should be returned with subpixel accuracy. The default is FALSE.

Rotation Angle Ranges (degrees) is an array of angle ranges, where each range specifies how much you expect the template pattern to rotate in the image. Each range is specified by a lower angle and an upper angle, both of which are expressed in degrees.



error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

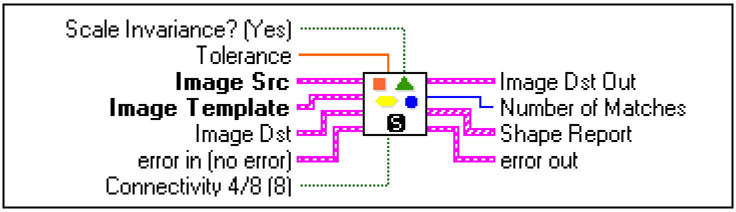
Match Pattern Setup Data is a string that contains information pertaining to the setup parameters selected for the matching stage. Connect this output to the IMAQ Match Pattern VI.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

IMAQ Shape Match Tool

Finds objects in an image whose shape matches the shape of the object specified by a template. The matching process is invariant to rotation and can be set to be invariant to the scale of the objects. This VI requires that the objects have been separated from the background (the input image is binary). This VI labels the image to give each object in the image a unique ID before performing the match operation.

8



Scale Invariance? (Yes) specifies whether to use size or scale invariance when matching objects in the source image to the template object.

Tolerance indicates the percentage difference allowable between the template shape and similar shapes in the image.

Image Src is an 8-bit binary image that contains objects of different shapes.

Image Template is an 8-bit binary image that contains the object to match.

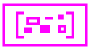
Image Dst is a reference to an 8-bit binary image that will contain the objects from the **Image Src** that match the object in the **Image Template**.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

Connectivity 4/8 (8) specifies how the algorithm determines whether an adjacent pixel is the same or different particle. The default is Connectivity 8.

Image Dst Out contains the objects that match the template object.

Number of Matches specifies the number of objects in the source image that match the template object within the specified tolerance.



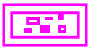
Shape Report is an array of clusters in which each cluster contains the following information on the matched objects:

Global Rectangle contains the bounding rectangle of the object.

Centroid is a point-coordinate cluster containing the location of the centroid of the object.

Object Size is the size in pixels of the object.

Score (between 1 and 1000) specifies how similar the object in the image is to the template. A score of 1000 implies a perfect match.



error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see the section *IMAQ VI Error Clusters* in Chapter 1, *VI Overview and Programming Concepts*.

A

Technical Support Resources

This appendix describes the comprehensive resources available to you in the Technical Support section of the National Instruments Web site and provides technical support telephone numbers for you to use if you have trouble connecting to our Web site or if you do not have internet access.

NI Web Support

To provide you with immediate answers and solutions 24 hours a day, 365 days a year, National Instruments maintains extensive online technical support resources. They are available to you at no cost, are updated daily, and can be found in the Technical Support section of our Web site at `www.natinst.com/support`.

Online Problem-Solving and Diagnostic Resources

- **KnowledgeBase**—A searchable database containing thousands of frequently asked questions (FAQs) and their corresponding answers or solutions, including special sections devoted to our newest products. The database is updated daily in response to new customer experiences and feedback.
- **Troubleshooting Wizards**—Step-by-step guides lead you through common problems and answer questions about our entire product line. Wizards include screen shots that illustrate the steps being described and provide detailed information ranging from simple getting started instructions to advanced topics.
- **Product Manuals**—A comprehensive, searchable library of the latest editions of National Instruments hardware and software product manuals.
- **Hardware Reference Database**—A searchable database containing brief hardware descriptions, mechanical drawings, and helpful images of jumper settings and connector pinouts.
- **Application Notes**—A library with more than 100 short papers addressing specific topics such as creating and calling DLLs, developing your own instrument driver software, and porting applications between platforms and operating systems.

Software-Related Resources

- **Instrument Driver Network**—A library with hundreds of instrument drivers for control of standalone instruments via GPIB, VXI, or serial interfaces. You also can submit a request for a particular instrument driver if it does not already appear in the library.
- **Example Programs Database**—A database with numerous, non-shipping example programs for National Instruments programming environments. You can use them to complement the example programs that are already included with National Instruments products.
- **Software Library**—A library with updates and patches to application software, links to the latest versions of driver software for National Instruments hardware products, and utility routines.

Worldwide Support

National Instruments has offices located around the globe. Many branch offices maintain a Web site to provide information on local services. You can access these Web sites from `www.natinst.com/worldwide`.

If you have trouble connecting to our Web site, please contact your local National Instruments office or the source from which you purchased your National Instruments product(s) to obtain support.

For telephone support in the United States, dial 512 795 8248. For telephone support outside the United States, contact your local branch office:

Australia 03 9879 5166, Austria 0662 45 79 90 0, Belgium 02 757 00 20, Brazil 011 284 5011, Canada (Ontario) 905 785 0085, Canada (Québec) 514 694 8521, China 0755 3904939, Denmark 45 76 26 00, Finland 09 725 725 11, France 01 48 14 24 24, Germany 089 741 31 30, Hong Kong 2645 3186, India 91805275406, Israel 03 6120092, Italy 02 413091, Japan 03 5472 2970, Korea 02 596 7456, Mexico (D.F.) 5 280 7625, Mexico (Monterrey) 8 357 7695, Netherlands 0348 433466, Norway 32 27 73 00, Singapore 2265886, Spain (Madrid) 91 640 0085, Spain (Barcelona) 93 582 0251, Sweden 08 587 895 00, Switzerland 056 200 51 51, Taiwan 02 2377 1200, United Kingdom 01635 523545

Glossary

Prefix	Meanings	Value
p-	pico-	10^{-12}
n-	nano-	10^{-9}
μ-	micro-	10^{-6}
m-	milli-	10^{-3}
k-	kilo-	10^{3}
M-	mega-	10^{6}
G-	giga-	10^{9}
t-	tera-	10^{12}

Numbers

1D — One-dimensional.

2D — Two-dimensional.

3D — Three-dimensional.

3D view — Displays the light intensity of an image in a three-dimensional coordinate system, where the spatial coordinates of the image form two dimensions and the light intensity forms the third dimension.

A

AIPD — The National Instruments internal image file format used for saving calibration information associated with an image and for saving complex images.

alignment — The process by which a machine vision application determines the location, orientation, and scale of a part being inspected.

area threshold — Detects objects based on their size, which can fall within a user-specified range.

arithmetic operators	The image operations multiply, divide, add, subtract, and remainder.
asynchronous	Property of a function or operation that begins an operation and returns control to the program before the completion or termination of the operation.
auto-median function	A function that uses dual combinations of opening and closing operations to smooth the boundaries of objects.

B

barycenter	The barycenter of a range of an image's grayscale values is the grayscale value representing the centroid of that range in the image histogram.
binary image	An image containing objects usually represented with a pixel intensity of 1 (or 255) and the background of 0.
binary morphology	Functions that perform morphological operations on a binary image.
blob	Binary large object. A particle, or object, present in a binary image.
blurring	Reduces the amount of detail in an image. Blurring commonly occurs because the camera is out of focus. You can blur an image intentionally by applying a lowpass frequency filter.
BMP	Bitmap. Image file format commonly used for 8-bit and color images.
border function	Removes objects (or particles) in a binary image that touch the image border.

C

caliper	Finds edge pairs along a specified path in the image. This function performs an edge extraction and then finds edge pairs based on specified criteria such as the distance between the leading and trailing edges, edge contrasts, and so forth.
circle function	Detects circular objects in a binary image.
closing	A dilation followed by an erosion. A closing fills small holes in objects and smooths the boundaries of objects.

CLUT	Color look-up table. Table for converting the value of a pixel in an image into a red, green, and blue (RGB) intensity.
color images	Images containing color information, usually encoded in the RGB form.
complex images	Save information obtained from the FFT of an image. The complex numbers that compose the FFT plane are encoded in 64-bit floating-point values: 32 bits for the real part and 32 bits for the imaginary part.
connectivity	Defines which of the surrounding pixels of a given pixel constitute its neighborhood.
connectivity-4	Only pixels adjacent in the horizontal and vertical directions are considered neighbors.
connectivity-8	All adjacent pixels are considered as neighbors.
convex function	Computes the convex regions of objects in a binary image.
convolution	*See* linear filter.
convolution kernel	Simple 3 × 3, 5 × 5, or 7 × 7 matrices (or templates) used to represent the filter in the filtering process. The contents of these kernels are a discrete two-dimensional representation of the impulse response of the filter that they represent.

D

Danielsson function	Similar to the distance functions, but with more accurate results.
densitometry	Determination of optical or photographic density.
density function	For each gray level in a linear histogram, it gives the number of pixels in the image that have the same gray level.
device	Plug-in data acquisition board that can contain multiple channels and conversion devices.
differentiation filter	Extracts the contours (edge detection) in gray level.
digital image	An image $f(x, y)$ that has been converted into a discrete number of pixels. Both spatial coordinates and brightness are specified.

dilation	Increases the size of an object along its boundary and removes tiny holes in the object.
distance calibration	Determination of the physical dimensions of a pixel by defining the physical dimensions of a line in the image.
distance function	Assigns to each pixel in an object a gray-level value equal to its shortest Euclidean distance from the border of the object.

E

edge	Defined by a sharp change (transition) in the pixel intensities in an image or along an array of pixels.
edge contrast	The difference between the average pixel intensity before and the average pixel intensity after the edge.
edge hysteresis	The difference in threshold levels between a rising and a falling edge.
edge steepness	The number of pixels that correspond to the slope or transition area of an edge.
entropy	A measure of the randomness in an image. An image with high entropy contains more pixel value variation than an image with low entropy.
equalize function	*See* histogram equalization.
erosion	Reduces the size of an object along its boundary and eliminates isolated points in the image.
exponential and gamma corrections	Expand the high gray-level information in an image while suppressing low gray-level information.
exponential function	Decreases the brightness and increases the contrast in bright regions of an image and decreases contrast in dark regions.

F

FFT	Fast Fourier Transform. A method used to compute the Fourier Transform of an image.
fiducial	A reference pattern on a part that helps a machine vision application find the part’s location and orientation in an image.

Fourier spectrum	The magnitude information of the Fourier Transform of an image.
Fourier Transform	Transforms an image from the spatial domain to the frequency domain.
frequency filters	Counterparts of spatial filters in the frequency domain. For images, frequency information is in the form of spatial frequency.

G

gauging	Measurement of an object or distances between objects.
Gaussian filter	A filter similar to the smoothing filter, but using a Gaussian kernel in the filter operation. The blurring in a Gaussian filter is more gentle than a smoothing filter.
gradient convolution filter	*See* gradient filter.
gradient filter	Extracts the contours (edge detection) in gray-level values. Gradient filters include the Prewitt and Sobel filters.
gray level	The brightness of a point (pixel) in an image.
gray-level dilation	Increases the brightness of pixels in an image that are surrounded by other pixels with a higher intensity.
gray-level erosion	Reduces the brightness of pixels in an image that are surrounded by other pixels with a lower intensity.
gray-level images	Images with monochrome information.
gray-level morphology	Functions that perform morphological operations on a gray-level image.

H

highpass attenuation	Inverse of lowpass attenuation.
highpass FFT filter	Removes or attenuates low frequencies present in the FFT domain of an image.
highpass filter	Emphasizes the intensity variations in an image, detects edges (or object boundaries), and enhances fine details in an image.

highpass frequency filter	Attenuates or removes (truncates) low frequencies present in the frequency domain of the image. A highpass frequency filter suppresses information related to slow variations of light intensities in the spatial image.
highpass truncation	Inverse of lowpass truncations.
histogram	Indicates the quantitative distribution of the pixels of an image per gray-level value.
histogram equalization	Transforms the gray-level values of the pixels of an image to occupy the entire range (0 to 255 in an 8-bit image) of the histogram, increasing the contrast of the image.
histogram inversion	Finds the photometric negative of an image. The histogram of a reversed image is equal to the original histogram flipped horizontally around the center of the histogram.
hit-miss function	Locates objects in the image similar to the pattern defined in the structuring element.
hole filling function	Fills all holes in objects that are present in a binary image.
HSI	Color encoding scheme in Hue, Saturation, and, Intensity.
HSL	Color encoding scheme using Hue, Saturation, and Luminance information where each image in the pixel is encoded using 32-bits: 8 bits for hue, 8 bits for saturation, 8 bits for luminance, and 8 unused bits.
HSV	Color encoding scheme in Hue, Saturation, and Value.

I

image	A two-dimensional light intensity function $f(x, y)$, where, x and y denote spatial coordinates and the value f at any point (x, y) is proportional to the brightness at that point.
image file	A file containing image information and data.
image processing	Encompasses various processes and analysis functions that you can apply to an image.
image understanding	A technique that interprets the content of the image at a symbolic level rather than a pixel level.

image visualization	The presentation (display) of an image (image data) to the user.
inner gradient	Finds the inner boundary of objects.
inspection	The process by which parts are tested for simple defects such as missing parts or cracks on part surfaces.
inspection functions	Detects specific features in an image, including edges, peaks, and rotational shifts.
intensity calibration	Assigning user-defined quantities such as optical densities or concentrations to the gray-level values in an image.
intensity profile	The gray-level distribution of the pixels along the ROI in an image.
intensity range	Defines the range of gray-level values in an object of an image.
intensity threshold	Characterizes an object based on the range of gray-level values in the object. If the intensity range of the object falls within the user-specified range, it is considered an object; otherwise it is considered part of the background.
interpolation	The technique used to find values between known values when resampling an image or array of pixels.

J

JPEG	Joint Photographic Experts Group. Image file format for storing 8-bit and color images with lossy compression.

L

labeling	The process by which each object in a binary image is assigned a unique value. This process is useful for identifying the number of objects in the image and giving each object a unique identity.
Laplacian filter	Extracts the contours of objects in the image by highlighting the variation of light intensity surrounding a pixel.
line gauge	Measures the distance between selected edges with high-precision subpixel accuracy along a line in an image. For example, this function can be used to measure distances between points and edges and vice versa. This function also can step and repeat its measurements across the image.

line profile	Represents the gray-level distribution along a line of pixels in an image.
linear filter	A special algorithm that calculates the value of a pixel based on its own pixel value as well as the pixel values of its neighbors. The sum of this calculation is divided by the sum of the elements in the matrix to obtain a new pixel value.
logarithmic and inverse gamma corrections	Expand low gray-level information in an image while compressing information from the high gray-level ranges.
Logarithmic function	Increases the brightness and contrast in dark regions of an image and decreases the contrast in bright regions of the image.
logic operators	The image operations AND, NAND, OR, XOR, NOR, difference, mask, mean, max, and min.
lossless compression	Compression in which the decompressed image is identical to the original image.
lossy compression	Compression in which the decompressed image is visually similar but not identical to the original image.
lowpass attenuation	Applies a linear attenuation to the frequencies in an image, with no attenuation at the lowest frequency and full attenuation at the highest frequency.
lowpass FFT filter	Removes or attenuates high frequencies present in the FFT domain of an image.
lowpass filter	Attenuates intensity variations in an image. You can use these filters to smooth an image by eliminating fine details and blurring edges.
lowpass frequency filter	Attenuates high frequencies present in the frequency domain of the image. A lowpass frequency filter suppresses information related to fast variations of light intensities in the spatial image.
lowpass truncation	Removes all frequency information above a certain frequency.
L-skeleton function	Uses an L-shaped structuring element in the Skeleton function.
LUT	Look-up table. Table containing values used to transform the gray-level values of an image. For each gray-level value in the image, the corresponding new value is obtained from the look-up table.

M

machine vision application	An inspection or measurement application that uses images acquired from a 2D sensor (typically a CCD camera) to help with inspection or measurement.
mask	Isolates parts of an image for further processing.
mask FFT filter	Removes frequencies contained in a mask (range) specified by the user.
mask image	An image containing a value of 1 and values of 0. Pixels in the source image with a corresponding mask image value of 1 are processed, while the others are left unchanged.
match score	A number ranging from 0 to 1000 that indicates how closely an acquired image matches the template image. A match score of 1000 indicates a perfect match. A match score of 0 indicates no match.
median filter	A lowpass filter that assigns to each pixel the median value of its neighbors. This filter effectively removes isolated pixels without blurring the contours of objects.
MMX	Multimedia Extensions. Intel chip-based technology that allows parallel operations on integers, which results in accelerated processing of 8-bit images.
morphological transformations	Extract and alter the structure of objects in an image. You can use these transformations for expanding (dilating) or reducing (eroding) objects, filling holes, closing inclusions, or smoothing borders. They mainly are used to delineate objects and prepare them for quantitative inspection analysis.
M-skeleton	Uses an M-shaped structuring element in the skeleton function.

N

neighbor	A pixel whose value affects nearby pixels' values when an image is processed. The neighbors of a pixel are usually defined by a kernel.
neighborhood operations	Operations on a point in an image that take into consideration the values of the pixels neighboring that point.

nonlinear filter	Replaces each pixel value with a nonlinear function of its surrounding pixels.
nonlinear gradient filter	A highpass edge-extraction filter that favors vertical edges.
nonlinear Prewitt filter	A highpass edge-extraction filter that favors horizontal and vertical edges in an image.
nonlinear Sobel filter	A highpass edge-extraction filter that favors horizontal and vertical edges in an image.
Nth order filter	Filters an image using a nonlinear filter. This filter orders (or classifies) the pixel values surrounding the pixel being processed. The pixel being processed is set to the *N*th pixel value, where *N* is the order of the filter.

O

opening	An erosion followed by a dilation. An opening removes small objects and smoothes boundaries of objects in the image.
operators	Allow masking, combination, and comparison of images. You can use arithmetic and logic operators in IMAQ Vision.
optical character verification	A machine vision application that inspects the quality of printed characters.
optical representation	Contains the low-frequency information at the center and the high-frequency information at the corners of an FFT-transformed image.
outer gradient	Finds the outer boundary of objects.

P

palette	The gradation of colors used to display an image on screen, usually defined by a color look-up table.
pattern matching	The technique used to locate quickly known reference patterns or fiducials in an image.
picture element	An element of a digital image.
pixel	Picture element.

pixel calibration	Directly calibrating the physical dimensions of a pixel in an image.
pixel depth	The number of bits used to represent the gray level of a pixel.
PNG	Portable Network Graphic. Image file format for storing 8-bit, 16-bit, and color images with lossless compression.
Power 1/Y function	Similar to a logarithmic function but with a weaker effect.
Power Y function	*See* exponential function.
Prewitt filter	Extracts the contours (edge detection) in gray-level values using a 3 × 3 filter kernel.
probability function	Defines the probability that a pixel in an image has a certain gray-level value.
proper-closing	A finite combination of successive closing and opening operations that you can use to fill small holes and smooth the boundaries of objects.
proper-opening	A finite combination of successive opening and closing operations that you can use to remove small particles and smooth the boundaries of objects.
pyramidal matching	A technique used to increase the speed of a pattern matching algorithm by matching subsampled versions of the image and the reference pattern.

Q

quantitative analysis	Obtaining various measurements of objects in an image.

R

Reverse function	Inverts the pixel values in an image, producing a photometric negative of the image.
RGB	Color encoding scheme using red, green and blue (RGB) color information where each pixel in the color image is encoded using 32 bits: 8 bits for red, 8 bits for green, 8 bits for blue, and 8 bits for the alpha value (unused).
Roberts filter	Extracts the contours (edge detection) in gray level, favoring diagonal edges.

ROI	Region of interest. An area of the image that is graphically selected from a window displaying the image. This area can be used to focus further processing.
rotation-invariant matching	A pattern matching technique in which the reference pattern can be at any orientation in the test image.
rotational shift	The amount by which one image is rotated with respect to a reference image. This rotation is computed with respect to the center of the image.

S

scale-invariant matching	A pattern matching technique in which the reference pattern can be any size in the test image.
segmentation function	Fully partitions a labeled binary image into non-overlapping segments, with each segment containing a unique object.
separation function	Separates objects that touch each other by narrow isthmuses.
shape matching	Finds objects in an image whose shape matches the shape of the object specified by a template. The matching process is invariant to rotation and can be set to be invariant to the scale of the objects.
shift-invariant matching	A pattern matching technique in which the reference pattern can be located anywhere in the test image but cannot be rotated or scaled.
Sigma filter	A highpass filter that outlines edges.
skeleton function	Applies a succession of thinning operations to an object until its width becomes one pixel.
skiz	Obtains lines in an image that separate each object from the others and are equidistant from the objects that they separate.
smoothing filter	Blurs an image by attenuating variations of light intensity in the neighborhood of a pixel.
Sobel filter	Extracts the contours (edge detection) in gray-level values using a 3×3 filter kernel.
spatial calibration	Assigning physical dimensions to the area of a pixel in an image.

spatial filters	Alter the intensity of a pixel with respect to variations in intensities of its neighboring pixels. You can use these filters for edge detection, image enhancement, noise reduction, smoothing, and so forth.
spatial resolution	The number of pixels in an image, in terms of the number of rows and columns in the image.
Square function	*See* exponential function.
Square Root function	*See* logarithmic function.
standard representation	Contains the low-frequency information at the corners and high-frequency information at the center of an FFT-transformed image.
structuring element	A binary mask used in most morphological operations. A structuring element is used to determine which neighboring pixels contribute in the operation.
sub-pixel analysis	Used to find the location of the edge coordinates in terms of fractions of a pixel.
synchronous	Property or operation that begins an operation and returns control to the program only when the operation is complete.
syntax	Set of rules to which statements must conform in a particular programming language.

T

thickening	Alters the shape of objects by adding parts to the object that match the pattern specified in the structuring element.
thinning	Alters the shape of objects by eliminating parts of the object that match the pattern specified in the structuring element.
threshold	Separates objects from the background by assigning all pixels with intensities within a specified range to the object and the rest of the pixels to the background. In the resulting binary image, objects are represented with a pixel intensity of 255 and the background is set to 0.
threshold interval	Two parameters, the lower threshold gray-level value and the upper threshold gray-level value.

TIFF	Tagged Image File Format. Image format commonly used for encoding 8-bit and color images.
truth table	A table associated with a logic operator that describes the rules used for that operation.

Index

D

E

F

G

H

I

L

M

T

V

W