

IMAQ VISION HIGH-LEVEL FUNCTION EXAMPLES

This guide describes IMAQ Vision high-level functions for use in infrared imaging applications.

Contents

Introduction	1
Blob Detection	2
Light Meter (Area)	4
Light Meter (Line)	5
Light Meter (Point)	6
Draw Result (Area).....	7
Draw Result (Line)	8
Draw Result (Point)	9
IMAQ VI Error Clusters	10

Introduction

The following VIs are examples of high-level functions that you can write using IMAQ Vision. Examples include blob detection, light meters, and drawing results on an image. You are free to use them and distribute them in your applications.

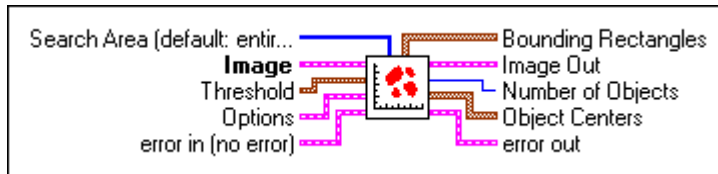
These examples are ideal for the analysis of infrared images. After you acquire an image with your infrared camera, use the software that comes with your camera to transform the image the raw pixel intensity values into temperature values. You can then analyze the temperature values with the VIs in this guide. Use the blob detection example to identify objects based on a temperature range. Use the light meter examples to analyze the temperature values in an area, along a line, or at a point in the image. Use the draw result examples to add an overlay to an image display. The overlay can indicate the area, line, or point analyzed in the image.



Note In all the VIs in this guide, **error in (no error)** is a cluster that describes the error status before a VI executes. **error out** is a cluster that describes the error status after a VI executes. For more information about these clusters, see the *IMAQ VI Error Clusters* section in later in this guide.

Blob Detection

Detects blobs or binary objects in an image based on pixel intensity. The VI returns information about the number of blobs found in the image and the location of each blob. A threshold performed on the image separates the objects from the background. Morphology operations filter the particles, and blob analysis determines the number and location of objects.



Search Area (default: entire image) defines a four-element array that contains the coordinates [Left/Top/Right/Bottom] of the area to search for blobs. The entire image is searched if the input is empty or not connected.



Image is the reference to the image structure to search for blobs.



Threshold is a cluster specifying the threshold range.



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.



Options lets you select advanced options, as follows:



Reject Border, if TRUE, eliminates objects that touch the edge of the image.



Fill Holes, if TRUE, fills the holes found in the blobs.



Min Object Size, if TRUE, removes all blobs that have an area less than **Min Size** pixels.



Min Size is the minimum size of a blob in pixels. Any blob with an area less than **Min Size** pixels is removed. This filtering occurs only if **Min Object Size** is TRUE.



Max Object Size, if TRUE, removes all blobs that have an area greater than **Max Size** pixels.



Max Size is the maximum size of a blob in pixels. Any blob with an area greater than **Max Size** pixels is removed. This filtering occurs only if **Max Object Size** is TRUE.

[00]

Bounding Rectangles is an array containing the coordinates of the rectangle that bounds each blob detected in the image. Each rectangle is composed of the following elements:

I32

X Left indicates the x coordinate of the top-left corner of the rectangle.

I32

Y Top indicates the y coordinate of the top-left corner of the rectangle.

I32

X Right indicates the x coordinate of the bottom-right corner of the rectangle.

I32

Y Bottom indicates the y coordinate of the bottom-right corner of the rectangle.

57

Image Out is the reference to **Image**.

I32

Number of Objects is the number of blobs detected in the image after any filtering indicated in **Options** is performed.

[00]

Object Centers is an array containing the center of each blob detected in the image. Each center is composed of the following elements:

5GL

X is the x coordinate of the center of mass of the blob.

5GL

Y is the y coordinate of the center of mass of the blob.

Light Meter (Area)

Analyzes the intensity values in an image. You can use this VI to calculate the histogram of the area and intensity statistics.

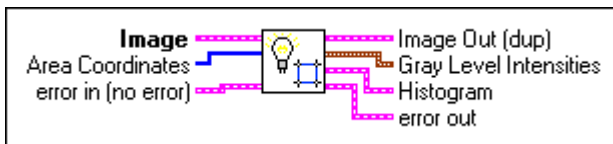


Image is the reference to the image structure to be analyzed.



Area Coordinates defines a four-element array that contains the coordinates [Left/Top/Right/Bottom] of the area to analyze.



Image Out (dup) is the reference to **Image**.



Gray Level Intensities returns information about the pixel intensity values in the area. This cluster contains the following elements:



Mean Intensity returns the mean pixel intensity values in the area.



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.



Minimum Intensity returns the minimum pixel intensity in the area.



Maximum Intensity returns the maximum pixel intensity in the area.



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 $[(Starting\ Value + (n - 1) \times Interval\ Width), (Starting\ Value + n \times (Interval\ Width - 1))]$. This histogram is computed using 256 classes.

Light Meter (Line)

Analyzes the intensity values along a line in an image. Calculates the line profile and intensity statistics.

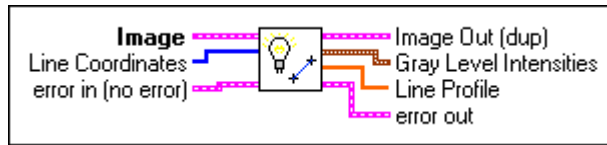


Image is the reference to the image structure to be analyzed.



Line Coordinates is a four-element array which contains the endpoints of the line. The array contains [start X, start Y, end X, end Y].



Image Out (dup) is the reference to **Image**.



Gray Level Intensities returns information about the pixel intensity values in the area. This cluster contains the following elements:



Mean Intensity returns the mean pixel intensity values in the line.



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.



Minimum Intensity returns the minimum pixel intensity in the line.



Maximum Intensity returns the maximum pixel intensity in the line.



Line Profile returns the pixel intensity values along the specified line.

Light Meter (Point)

Analyzes the intensity value at a point in an image.

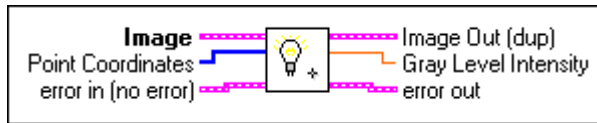


Image is the reference to the image structure to be analyzed.

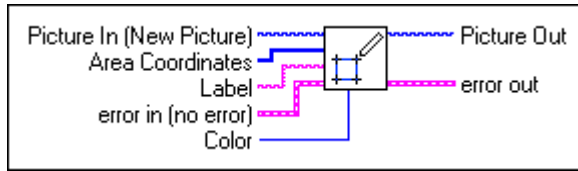
Point Coordinates is a two-element array which contains the x coordinate and y coordinate of the point to analyze.

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

Gray Level Intensity is the mean pixel intensity value of the 3×3 matrix of pixels centered at **Point Coordinates**.

Draw Result (Area)

Draws a rectangle on a picture. You can then overlay the rectangle onto an IMAQ Vision window using the IMAQ AddPictToWindow VI. You can use the Draw Result (Area) VI to indicate the location of an analyzed area on an image.



Picture In (New Picture) is the picture to which to add the rectangle. If not wired, this parameter defaults to an empty picture.



Area Coordinates defines a four-element array that contains the coordinates [Left/Top/Right/Bottom] of the area to add to the picture.



Label is text that you can add above the rectangle.



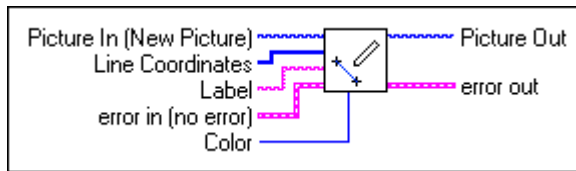
Color is a color numeric that specifies the color of the pixel. The default is red.



Picture Out is the modified picture.

Draw Result (Line)

Draws a line on a picture. You can then overlay the line onto an IMAQ Vision window using the IMAQ AddPictToWindow VI. You can use the Draw Result (Line) VI to indicate the location of an analyzed line on an image.



Picture In (New Picture) is the picture to which to add the line. If not wired, this parameter defaults to an empty picture.



Line Coordinates is a four-element array which contains the endpoints of the line. The array contains [start X, start Y, end X, end Y].



Label is text that you can add above the line.



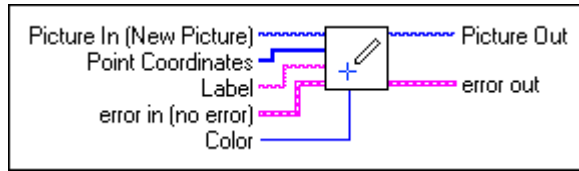
Color is a color numeric that specifies the color of the pixel. The default is red.



Picture Out is the modified picture.

Draw Result (Point)

Draws a cross on a picture. You can then overlay the cross onto an IMAQ Vision window using the IMAQ AddPictToWindow VI. You can use the Draw Result (Point) VI to indicate the location of an analyzed point on an image.



Picture In (New Picture) is the picture to which to add the point. If not wired, this parameter defaults to an empty picture.



Point Coordinates is a two-element array which contains the x coordinate and y coordinate of the point.



Label is text that you can add above the cross.



Color is a color numeric that specifies the color of the pixel. The default is red.



Picture Out is the modified picture.

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.