

# Image Processing Toolbox Release Notes

---

The “Image Processing Toolbox 4.1 Release Notes” on page 1-1 summarize the changes introduced in the latest version of the Image Processing Toolbox. The following topics are discussed in these Release Notes:

- “New Features” on page 1-2
- “Major Bug Fixes” on page 1-4

---

**Note** Image Processing Toolbox 4.1 is an update to Version 4.0, which was released as a Web-download release after Release 13.

---

If you are upgrading from a release earlier than Release 13, you should also see these sections:

- “Image Processing Toolbox 4.0 Release Notes” on page 2-1
- “Image Processing Toolbox 3.2 Release Notes” on page 3-1
- “Image Processing Toolbox 3.1 Release Notes” on page 4-1
- “Image Processing Toolbox 2.2.2 Release Notes” on page 5-1

## Printing the Release Notes

If you would like to print the Release Notes, you can link to a PDF version.



## Image Processing Toolbox 4.1 Release Notes

1

<b>New Features</b> .....	1-2
Reading and Writing Data with JPEG Lossless Compression .	1-2
Reading ICC Profiles Embedded in TIFF Files .....	1-2
Reading and Writing L*a*b* Color Data .....	1-2
<b>Major Bug Fixes</b> .....	1-4
applycform Fixes .....	1-4
Compiling Spatial Transformation Functions .....	1-4

## Image Processing Toolbox 4.0 Release Notes

2

<b>New Features</b> .....	2-2
New Image Viewer .....	2-2
Enhanced Color Space Functions .....	2-3
New Image Enhancement Methods .....	2-4
Enhanced DICOM Support .....	2-4
Fan Beam Projection Transforms .....	2-4
Boundary Tracing Functions .....	2-5
Unsigned Integer Lookup Tables .....	2-5
Optimized Image Arithmetic Functions .....	2-5
Performance Improvements .....	2-5
<b>Minor Enhancements</b> .....	2-6
Changes to Sample Images Included .....	2-7
Obsolete Functions .....	2-7
<b>Major Bug Fixes</b> .....	2-8

## Image Processing Toolbox 3.2 Release Notes

---

### 3

<b>New Features</b> .....	<b>3-2</b>
Writing DICOM Files .....	<b>3-2</b>
Representing Binary Images .....	<b>3-2</b>
Changes to Existing Functions .....	<b>3-3</b>
<b>Major Bug Fixes</b> .....	<b>3-4</b>
<b>Upgrading from an Earlier Release</b> .....	<b>3-5</b>
Change to Data Type of Output Binary Images .....	<b>3-5</b>
Change to Interpretation of Input Images .....	<b>3-5</b>
Converting Binary Images to an Integer Data Type .....	<b>3-5</b>

## Image Processing Toolbox 3.1 Release Notes

---

### 4

<b>New Features</b> .....	<b>4-2</b>
Morphology .....	<b>4-3</b>
Spatial Transformations .....	<b>4-4</b>
Image Registration .....	<b>4-5</b>
Integer Image Arithmetic .....	<b>4-6</b>
Integer Image Filtering .....	<b>4-7</b>
Deconvolution/Deblurring .....	<b>4-7</b>
Support for DICOM Files .....	<b>4-8</b>
Miscellaneous New Functions .....	<b>4-8</b>
New Demos .....	<b>4-9</b>
<b>Major Bug Fixes</b> .....	<b>4-11</b>
Version 3.1 Bug Fixes .....	<b>4-11</b>

## Image Processing Toolbox 2.2.2 Release Notes

---

# 5

<b>New Features</b> .....	<b>5-2</b>
New Demo .....	<b>5-2</b>
Support For Function Handles .....	<b>5-2</b>
Documentation Enhanced .....	<b>5-2</b>
<b>Major Bug Fixes</b> .....	<b>5-3</b>
imshow Fixes .....	<b>5-3</b>
bwlabel Segmentation Violation Eliminated .....	<b>5-3</b>
dilate And erode Return Correct Answers .....	<b>5-3</b>
freqz2 Fixes .....	<b>5-3</b>
fspecial Function's 'LoG' Option .....	<b>5-3</b>
Improved Display for imcrop, improfile, and roipoly .....	<b>5-4</b>



# Image Processing Toolbox

## 4.1 Release Notes

---

<b>New Features</b> . . . . .	1-2
Reading and Writing Data with JPEG Lossless Compression . . . . .	1-2
Reading ICC Profiles Embedded in TIFF Files . . . . .	1-2
Reading and Writing L*a*b* Color Data . . . . .	1-2
<b>Major Bug Fixes</b> . . . . .	1-4
applycform Fixes . . . . .	1-4
Compiling Spatial Transformation Functions . . . . .	1-4

## New Features

This section introduces the new features and enhancements added in the Image Processing Toolbox 4.1 since Version 4.0 (which was released as a Web-download release after Release 13).

### Reading and Writing Data with JPEG Lossless Compression

The toolbox now supports reading and writing data that has been compressed using JPEG lossless compression. With lossless compression, you can recover the original image from its compressed form. Lossless compression, however, achieves lower compression ratios than its counterpart, lossy compression.

Using either the `imread` function or the `dicomread` function, you can read data that has been compressed using JPEG lossless compression.

Using either the `imwrite` or the `dicomwrite` function, you can write data to a JPEG file using lossless compression. For the `imwrite` function, you specify the `Mode` parameter with the `'lossless'` value. For the `dicomwrite` function, you specify the `CompressionMode` parameter with the `'JPEG lossless'` value.

### Reading ICC Profiles Embedded in TIFF Files

`iccread` can now read ICC profiles that are embedded in a TIFF file, if the TIFF file contains one. ICC profiles contain information that color management systems need to translate color data between devices.

To determine if a TIFF file contains an ICC profile, use the `imfinfo` function to retrieve information about the file. If the returned data contains the `ICCProfileOffset` field, the file contains an embedded ICC profile.

### Reading and Writing L\*a\*b\* Color Data

The `imread` function can now read color data that uses the  $L^*a^*b^*$  color space from TIFF files. The TIFF files can contain  $L^*a^*b^*$  values that are in 8-bit or 16-bit CIELAB encodings or in 8-bit or 16-bit ICCLAB encodings.

If a file contains 8-bit or 16-bit CIELAB data, `imread` automatically converts the data into 8-bit or 16-bit ICCLAB encoding. The 8-bit or 16-bit CIELAB data cannot be represented as a MATLAB array because it contains a combination of signed and unsigned values.



The `imwrite` function can write  $L^*a^*b^*$  data to a file using either the 8-bit or 16-bit CIELAB encoding or the 8-bit or 16-bit ICCLAB encoding. You select the encoding by specifying the value of the `ColorSpace` parameter.

## Major Bug Fixes

The Image Processing Toolbox, Version 4.1, includes the following bug fixes

### **applycform Fixes**

The `applycform` function includes two bug fixes.

- The `applycform` function did not apply some profiles correctly when the input color was in the *XYZ* color space. Specifically, profiles containing an 8-bit or 16-bit lookup table containing a non-identity "E" matrix were not processed correctly by `applycform`. For details about the E matrix, see ICC Specification ICC.1:2001-04, sections 6.5.7 and 6.5.8.
- The `applycform` function now handles correctly Matrix/TRC profiles that contain a single gamma correction factor. Previously, the forward and inverse conversions were reversed.

### **Compiling Spatial Transformation Functions**

Applications that call the `imresize`, `imrotate`, `imtransform`, `tformarray`, `tformfwd`, and `tforminv` functions can now be compiled using the MATLAB Compiler.

# Image Processing Toolbox 4.0 Release Notes

---

<b>New Features</b> . . . . .	2-2
New Image Viewer . . . . .	2-2
Enhanced Color Space Functions . . . . .	2-3
New Image Enhancement Methods . . . . .	2-4
Enhanced DICOM Support . . . . .	2-4
Fan Beam Projection Transforms . . . . .	2-4
Boundary Tracing Functions . . . . .	2-5
Unsigned Integer Lookup Tables . . . . .	2-5
Optimized Image Arithmetic Functions . . . . .	2-5
Performance Improvements . . . . .	2-5
<b>Minor Enhancements</b> . . . . .	2-6
Changes to Sample Images Included . . . . .	2-7
Obsolete Functions . . . . .	2-7
<b>Major Bug Fixes</b> . . . . .	2-8

## New Features

This section introduces the new features and enhancements added in the Image Processing Toolbox 3.2 since Version 3.2 (Release 13).

### New Image Viewer

The toolbox includes a new tool for displaying images, called the Image Viewer. This tool supports zooming, scrolling, and overview navigation with large images. The Image Viewer automatically displays the pixel value at the mouse location but you can also use a special zoom tool, called the Pixel Region tool, to perform simultaneous color and quantitative inspection of individual pixels. You can also view metadata for the image file or MATLAB variable.

To start the Image Viewer, use the `imview` function.

```
imview('board.tif')
```

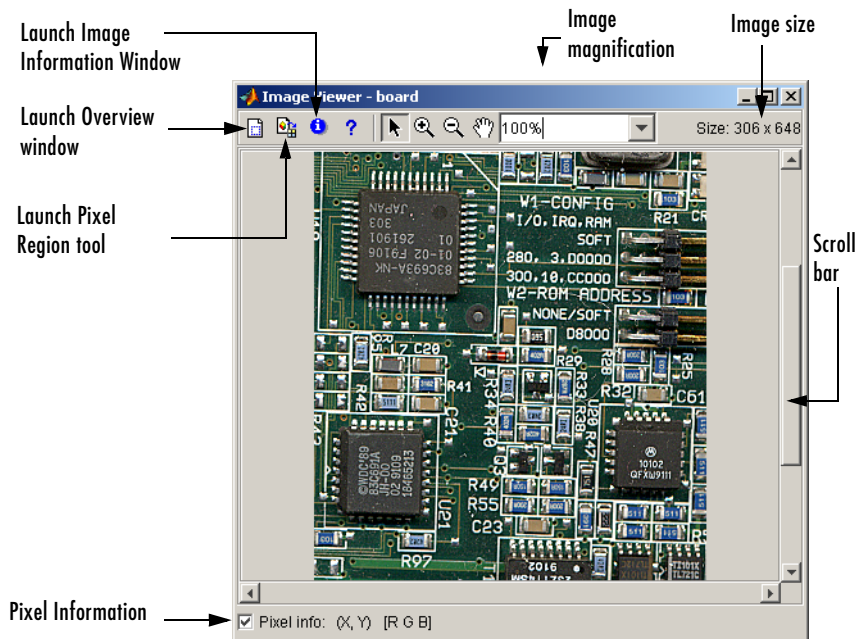
The following figure illustrates the Image Viewer and its capabilities.

---

**Note** On platforms that don't support JAVA, have an older version of JAVA, and on Macintosh systems, calls to `imview` invoke the `imshow` function. The toolbox issues this warning when `imview` is invoked:

```
'IMVIEW is not available on this platform.', ...  
'Calling IMSHOW instead.');
```

---



## Enhanced Color Space Functions

The toolbox includes a pair of new functions, `makecform` and `applycform`, for converting to and from a family of standard, device-independent color spaces. The functions support conversions between members of the family of color spaces defined by the CIE *Commission Internationale de l'Éclairage* (International Commission on Illumination), including the  $XYZ$ ,  $xyY$ ,  $uvL$ ,  $u'v'L$ ,  $L^*a^*b^*$ , and  $L^*ch$  color spaces. The functions also support conversion to and from the industry standard  $sRGB$  color space.

The toolbox also includes new function, `iccread`, for reading in ICC color profiles and using them to transform color data.

In addition, the toolbox also includes functions for converting the class representation of converted color spaces: `lab2uint8`, `lab2uint16`, `lab2double`, `xyz2uint8`, and `xyz2double` functions.

### **New Image Enhancement Methods**

The toolbox includes two new image enhancement functions: `adapthisteq` and `decorrstretch`.

The `adapthisteq` function performs contrast-limited adaptive histogram equalization (CLAHE). This function uses a contrast-enhancement method that works significantly better than regular histogram equalization for most images.

The `decorrstretch` function performs a decorrelation stretch on truecolor images, or images with multiple color or spectral bands. Decorrelation stretch is a technique used to enhance, or stretch, the color differences in an image. This function can be used, for example, to aid visual interpretation when two or more bands are significantly correlated.

### **Enhanced DICOM Support**

The `dicomwrite` function now supports exporting to DICOM files using the MR (magnetic resonance) and CT (computed tomography) modalities.

The `dicominfo` and `dicomread` functions can now read some files that are marginally noncompliant with the DICOM specification. Some commonly-used medical imaging devices produce such files. In addition, these functions can now read some files produced by GE devices that use certain private transfer syntaxes.

The toolbox includes a new function, `dicomuid`, that generates DICOM unique identifiers. This is the same method used by the `dicomwrite` function.

### **Fan Beam Projection Transforms**

The toolbox includes two new functions, `fanbeam` and `ifanbeam`, for computing an alternative mathematical representation of an image using fan beam projections. Using the `ifanbeam` function, you can reconstruct an image from fan beam projection data.

The toolbox also includes functions, `fan2para` and `para2fan`, for converting projection data between fan-beam and parallel-beam geometries. (You use the `radon` function to create parallel beam projection data.)

## Boundary Tracing Functions

The toolbox includes a new function, `bwboundaries`, to trace the boundaries of all objects in a binary image. The new `bwtraceboundary` function traces a single boundary from a given starting point.

## Unsigned Integer Lookup Tables

The toolbox includes a new function, `uintlut`, that changes element values in a `uint8` or `uint16` array by passing them through a 256-element or 65,536-element lookup table. This low-level utility function has been used to speed up other toolbox functions such as `imadjust`.

## Optimized Image Arithmetic Functions

The image arithmetic functions have been optimized in two ways:

- Portable code improvements have been made to speed up the arithmetic functions on all platforms.
- Pentium- and MMX-specific code improvements have been made to provide additional speed improvements on the Windows and Linux platforms. The changes are based on the Intel Performance Primitives Library.

Functions affected by these improvements include the `imabsdiff`, `imadd`, `imcomplement`, `imdivide`, `imlincomb`, and the `immultiply` functions. To determine if the Intel Performance Primitives Library is being used, call the `ipp1` function.

## Performance Improvements

A variety of existing toolbox functions have been optimized to run faster and use less memory.

- Image type conversion functions
- Certain image enhancement functions: `imadjust` and `imhist`
- Certain color space conversion functions: `rgb2gray`, `rgb2ntsc`, `rgb2ycbcr`, and `ycbcr2rgb`
- Deblurring functions: `deconvblind`, `deconvlucy`, `deconvreg`, and `deconvwnr`

## Minor Enhancements

In addition to the major new features, the toolbox includes several additional enhancements.

<b>Function</b>	<b>Enhancement</b>
<code>cp2tform</code>	The algorithms have been modified to make them more robust numerically when the input coordinates have a very large offset from the origin.
<code>cpselect</code>	The Control Point Selection Tool now displays a single legend window even if multiple instances of the tool are being displayed.
<code>imadjust</code>	Supports a simplified, single-input syntax, in which it uses <code>stretchlim</code> to compute contrast-stretch parameters automatically.
<code>immovie</code>	No longer flickers while a movie is being generated.
<code>medfilt2</code>	Class support has been extended to all numeric types and it uses a new algorithm for large window sizes that is significantly faster than the old one.
<code>ordfilt2</code>	Class support has been extended to all numeric types and it uses a new algorithm for large rectangular domains that is significantly faster than the old one.
<code>regionprops</code>	Uses an improved method for computing the convex hull of a labeled object, resulting in more accurate <code>ConvexHull</code> , <code>Convexity</code> , and <code>ConvexImage</code> measurements.
<code>roipoly</code>	Uses a new algorithm that produces more intuitive results and has better performance. Users who have code that requires the same output as the previous version can use the function <code>roipolyold</code> .
<code>tformfwd</code> and <code>tforminv</code>	Supports additional syntaxes that make these functions easier to use for common operations.



## Changes to Sample Images Included

The sample images listed below have been removed from the toolbox.

afmsurf	bonemarr	enamel	ngc4024l	rice	testpat2
alumgrns	circles	flowers	ngc4024m	saturn	text
bacteria	circlesm	ic	ngc4024s	shot1	tissue1
blood1	debye1	lily	pearlite	testpat1	

The following new sample images are included with the toolbox.

blobs.png	peppers.png	testpat1.png
circles.png	rice.png	text.png
coins.png	saturn.png	tissue.png
glass.png	solarspectra.fts	

## New ICC Profiles

The toolbox includes sample ICC profiles that can be used with the color space conversion functions.

File	Description
lab8.icm	8-bit $L^*a^*b^*$ profile
monitor.icm	Typical monitor profile
sRGB.icm	<i>sRGB</i> profile
swopcmk.icm	CMYK input profile

## Obsolete Functions

Because of fundamental ambiguities in their behavior and definition, the `isrgb`, `isind`, `isbw`, and `isgray` functions have been obsoleted and will issue a warning when called.

## Major Bug Fixes

The Image Processing Toolbox, Version 4.0, includes the following bug fixes.

<b>Function</b>	<b>Fix</b>
<code>bwlabel</code>	No longer produces a segmentation violation when called with an empty matrix.
<code>cpselect</code>	<ul style="list-style-type: none"><li>• The Control Point Selection Tool no longer causes MATLAB to hang when invoked from within a script or GUI with the <code>input</code> or <code>uiwait</code> functions, which wait for users to complete the selection.</li><li>• Point prediction in the Control-Point Selection Tool no longer fails with the error message “Attempt to reference field on non-structure array 'pickPair'.”</li></ul>
<code>dicomread</code>	Now reads DICOM files that contain 1-bit overlay data.
<code>imabsdiff</code> , <code>imadd</code> , <code>imdivide</code> , <code>imlincomb</code> , <code>imsubtract</code>	The image arithmetic functions no longer error when called with logical inputs
<code>imfill</code>	Now fills hole pixels on the outer edge of an image that are not connected to the background because a non-default connectivity was specified.
<code>imresize</code>	The anti-aliasing filtering, applied by <code>imresize</code> when shrinking an image, no longer causes a narrow strip of dark pixels to appear around the edge of the image.
<code>label2rgb</code>	Now uses a new default colormap. The zero-color in the previous default colormap was the same as the last color in the colormap, with the result that the object labeled with the highest number could not be distinguished from the background.

<b>Function</b>	<b>Fix</b>
radon	An off-by-one error in the calculation of the center pixel location for images with even dimensions has been fixed.
rgb2ind	No longer errors when passed an input image that contain only a single color.
strel	The syntax <code>strel('ball', r, h, 0)</code> returns an empty <code>strel</code> object instead of a degenerate ball structuring element with radius 0.

If you are upgrading from a release earlier than Release 13, then you should also see “Major Bug Fixes” on page 3-4 of the Image Processing Toolbox 3.2 Release Notes.



# Image Processing Toolbox

## 3.2 Release Notes

---

<b>New Features</b> . . . . .	3-2
<b>Major Bug Fixes</b> . . . . .	3-3
<b>Upgrading from an Earlier Release</b> . . . . .	3-5

## New Features

This section introduces the new features and enhancements added in the Image Processing Toolbox 3.2 since Version 3.1 (Release 12.1).

The new features introduced in the Image Processing Toolbox, Version 3.2, include:

- More error checking of input images, specifically input classes, attributes and option string processing, with clearer error messages
- Support for writing DICOM files
- Changes to how binary images are represented. (Support for the new MATLAB `logical` data type.)
- Enhancements to several existing functions

If you are upgrading from a release earlier than Release 12.1, then you should also see “New Features” on page 4-2.

### Writing DICOM Files

The Image Processing Toolbox now supports writing files in Digital Imaging and Communications in Medicine (DICOM) format, using the `dicomwrite` function. Previous releases of the toolbox supported reading DICOM files with the `dicomread` function and reading metadata from a DICOM file using the `dicominfo` function.

### Representing Binary Images

In previous releases, toolbox functions that returned binary images returned them as `uint8` logical arrays. The toolbox used the presence of the logical flag to signify that the data range in the file was `[0,1]`.

With this release, the toolbox returns binary images as logical arrays, using the new MATLAB `logical` data type. For more information about the new logical class, see the MATLAB 6.5 Release Notes.

## Changes to Existing Functions

The Image Processing Toolbox, Version 3.2, includes changes to these existing functions.

Function	Description of Change
<code>circshift</code>	Moved into MATLAB
<code>freqz2</code>	Checks for insignificant real part in addition to insignificant imaginary part
<code>getnhood</code>	Returns a logical array
<code>gray2ind</code>	More efficient memory usage
<code>imfill</code>	New syntax for grayscale images does not require 'holes' argument. This option is selected automatically.
<code>imlincomb</code>	Accepts more than two images as input and you can specify the output class
<code>immovie</code>	Flicker during movie creation eliminated
<code>imtransform</code>	Linear and bicubic interpolation are faster
<code>ordfilt2</code>	Uses a different algorithm for binary images that improves processing speed for these images
<code>roifilt2</code>	More efficient. Operation is performed only on the region of interest, not the entire image.

### Major Bug Fixes

The Image Processing Toolbox 3.2 includes several bug fixes made since Version 3.1. You can see a list of the particularly important Version 3.2 bug fixes.

If you are viewing these Release Notes in PDF form, please refer to the HTML form of the Release Notes, using either the Help browser or the MathWorks Web site and use the link provided.

If you are upgrading from a release earlier than Release 12.1, then you should also see “Major Bug Fixes” on page 4-11.



## Upgrading from an Earlier Release

This section describes several upgrade issues involved in moving from the Image Processing Toolbox Version 3.1 to Version 3.2.

### Change to Data Type of Output Binary Images

All the Image Processing Toolbox functions that return a binary image now return a binary image of class `logical`. In previous releases, these functions returned binary images of a numeric class with the logical flag set. The Image Processing Toolbox used the existence of the logical flag to identify a binary image.

If your application checks the data type of the binary images returned by Image Processing Toolbox functions, you will need to change your code.

---

**Note** The logical class is not one of the numeric classes in MATLAB.

---

### Change to Interpretation of Input Images

Image Processing Toolbox functions that accept different types of images, such as grayscale and binary, no longer attempt to determine if an input image of a numeric class is intended to be a binary image.

In previous releases, toolbox functions that accepted different types of images checked the contents of an image to determine how to interpret it. For example, if an image was of class `double` and contained only 0s and 1s, the toolbox function would interpret it as a binary image. With Version 3.2, the toolbox only interprets images of class `logical` as binary images.

In the Image Processing Toolbox, the names of functions that accept both grayscale and binary images typically start with the characters “im”, such as `imdilate`.

### Converting Binary Images to an Integer Data Type

With this release, if you convert a binary image to a numeric type, the image ceases to be a binary image.

In previous releases, the Image Processing Toolbox conversion functions `im2uint8` and `im2double` preserved the binary attribute of the converted image. For example, if you converted a binary image of class `double`, which had the logical flag set, the output image returned by the `im2uint8` function would also be a logical image of class `uint8`, with the logical flag set.

For example, create a simple logical array

```
bw = logical([1 0; 0 1])
```

```
bw =
```

```
     1     0
     0     1
```

```
whos
```

Name	Size	Bytes	Class
bw	2x2	4	logical array

When you convert this array to a `uint8` data type, notice that it is no longer of class `logical`.

```
new_image = im2uint8(bw)
```

```
new_image =
```

```
    255     0
     0    255
```

```
whos
```

Name	Size	Bytes	Class
bw	2x2	4	logical array
new_image	2x2	4	uint8 array

# Image Processing Toolbox

## 3.1 Release Notes

---

<b>New Features</b> . . . . .	4-2
Morphology . . . . .	4-3
Spatial Transformations . . . . .	4-4
Image Registration . . . . .	4-5
Integer Image Arithmetic . . . . .	4-6
Integer Image Filtering . . . . .	4-7
Deconvolution/Deblurring . . . . .	4-7
Support for DICOM Files . . . . .	4-8
Miscellaneous New Functions . . . . .	4-8
New Demos . . . . .	4-9
<b>Major Bug Fixes</b> . . . . .	4-11
Version 3.1 Bug Fixes . . . . .	4-11

### New Features

This section describes the new features and enhancements of the Image Processing Toolbox, Version 3.0, and the Version 3.1 update, both introduced since the Image Processing Toolbox 2.2.2 (Release 12.0).

---

**Note** The Image Processing Toolbox 3.0 was made available in Web-downloadable form after Release 12.0. The Image Processing Toolbox 3.1 was part of Release 12.1.

---

The new features introduced in the Image Processing Toolbox, Version 3.0, include:

- Many new morphology functions
- New spatial transformation functions
- New image registration functions, with a new graphical user interface
- New integer image arithmetic functions
- New integer image filtering function
- New image deblurring (deconvolution) functions
- Support for DICOM files
- Miscellaneous new functions
- New image processing demos

The new features introduced in the Image Processing Toolbox, Version 3.1, include:

- New deblurring function, `deconvblind`, that implements the blind deconvolution algorithm
- New utility function, `label2rgb`, that converts a label matrix into an RGB color image

For information about Image Processing Toolbox features that are incorporated from Version 2.2.2, see “New Features” on page 5-2.

## Morphology

Version 3.0 adds a broad suite of new mathematical morphology tools open up broad new classes of applications in segmentation and image enhancement.

The existing dilation and erosion operators have been extended to work with grayscale images. New functions range from additional basic operators (opening, closing, tophat) to advanced tools useful for segmentation (distance transforms, reconstruction-based operators, and the watershed transform). The functions use advanced techniques for high performance, including automatic-structuring element decomposition, 32-bit binary image packing, and queue-based algorithms.

Function	Description
<code>bwareaopen</code>	Binary area open (remove small objects)
<code>bwdist</code>	Distance transform
<code>bwhitmiss</code>	Binary hit-miss operation
<code>bwlabeln</code>	Label-connected components in N-D binary image
<code>bwpack</code>	Pack binary image
<code>bwulterode</code>	Ultimate erosion
<code>bwunpack</code>	Unpack binary image
<code>conndef</code>	Default connectivity array
<code>imbothat</code>	Perform bottom-hat filtering
<code>imclearborder</code>	Suppress light structures connected to image border
<code>imclose</code>	Close image
<code>imdilate</code>	Dilate image
<code>imerode</code>	Erode image
<code>imextendedmax</code>	Extended-maxima transform
<code>imextendedmin</code>	Extended-minima transform

<b>Function</b>	<b>Description</b>
<code>imfill</code>	Fill image regions and holes
<code>imhmax</code>	H-maxima transform
<code>imhmin</code>	H-minima transform
<code>imimposemin</code>	Impose minima
<code>imopen</code>	Open image
<code>imreconstruct</code>	Morphological reconstruction
<code>imregionalmax</code>	Regional maxima
<code>imregionalmin</code>	Regional minima
<code>imtophat</code>	Tophat filtering
<code>strel</code>	Create morphological structuring element
<code>strel/getheight</code>	Get structuring element height
<code>strel/getnhood</code>	Get structuring element neighborhood
<code>strel/getsequence</code>	Get sequence of decomposed structuring elements
<code>strel/isflat</code>	Return true for flat structuring element
<code>strel/reflect</code>	Reflect structuring element about its center
<code>strel/translate</code>	Translate structuring element
<code>watershed</code>	Find image watershed regions

## **Spatial Transformations**

Version 3.0 adds functions for applying a variety of spatial transformations to images and to points. This is a core computational capability. Supported transform types include affine, projective, and user-defined custom transformations. Multidimensional transformations are supported, where you can control which dimensions are the transform dimensions. For example, you can apply a two-dimensional transform to an RGB image, and each color plane is automatically transformed the same way. You can even control the type of

interpolation independently along each dimension, and specify interpolants that you define.

<b>Function</b>	<b>Description</b>
checkerboard	Create checkerboard image
findbounds	Find output bounds for geometric transformation
fliptform	Flip the input and output roles of a TFORM struct
imtransform	Apply geometric transformation to image
makeresampler	Create resampler structure
maketform	Create geometric transformation structure (TFORM)
tformarray	Geometric transformation of a multidimensional array
tformfwd	Apply inverse geometric transformation
tforminv	Apply forward geometric transformation

## Image Registration

Version 3.0 adds several functions useful for registering (aligning) two images. This is critical in remote sensing and medical imaging, for example. There are functions for inferring various spatial transformations from control-point pairs, for the subpixel adjustment of control-point pair locations, and for normalized cross-correlation. There is also a graphical user interface (GUI) for selecting control-point pairs in a pair of images.

<b>Function</b>	<b>Description</b>
cp2tform	Infer spatial transformation from control-point pairs
cpcorr	Tune control-point locations using cross-correlation
cpselect	Control-point selection tool (graphical user interface)

<b>Function</b>	<b>Description</b>
cpstruct2pairs	Convert CPSTRUCT to valid pairs of control points
normxcorr2	Normalized two-dimensional cross-correlation

## **Integer Image Arithmetic**

The Image Processing Toolbox 3.1 includes new functions for performing arithmetic on image arrays without converting them to double-precision. In addition to the basic operations (add, subtract, multiply, and divide), there are several key functions (absolute difference, linear combination, and complementation) that cannot readily be implemented in terms of the basic operations.

<b>Function</b>	<b>Description</b>
imabsdiff	Absolute difference of two images
imadd	Add two images, or add constant to image
imcomplement	Complement image
imdivide	Divide two images, or divide image by constant
imlincomb	Linear combination of images
immultiply	Multiply two images, or multiply image by constant
imsubtract	Subtract two images, or subtract constant from image



## Integer Image Filtering

Version 3.0 added a function for performing filtering on image arrays without converting them to double precision, a significant memory savings in a common operation. You can specify several different boundary padding options. You can also perform higher dimensional filtering.

Function	Description
<code>imfilter</code>	Filter 2-D and N-D images

## Deconvolution/Deblurring

Version 3.0 added support for several fundamental algorithms for the deconvolution (deblurring) of images. All of the functions support multidimensional problems.

Function	Description
<code>deconvblind</code>	Deblur image using blind deconvolution algorithm [New with Version 3.1]
<code>deconvlucy</code>	Deblur image using Lucy-Richardson algorithm
<code>deconvreg</code>	Regularized deconvolution
<code>deconvwnr</code>	Wiener deconvolution
<code>edgetaper</code>	Taper image edges according to PSF
<code>fspecial</code>	Existing function; added 'disk' and 'motion' options
<code>otf2psf</code>	Convert optical transfer function to point-spread function
<code>psf2otf</code>	Convert point-spread function to optical transfer function

## Support for DICOM Files

Version 3.0 adds functions for reading image data and metadata from DICOM files. DICOM is an important file and network interchange standard in the area of medical imaging.

Function	Description
<code>dicomread</code>	Read image data from DICOM file
<code>dicominfo</code>	Read metadata from DICOM file

## Miscellaneous New Functions

Version 3.1 included several new utility functions or previously undocumented utility functions. Most of these were created to support functions in the key feature categories, such as deconvolution.

Function	Description
<code>circshift</code>	Shift array circularly  Note: This function was moved into MATLAB in release 3.2 of the Image Processing Toolbox.
<code>graythresh</code>	Compute global image threshold using Otsu's method (image enhancement)
<code>im2mis</code>	Convert image to Java <code>MemoryImageSource</code>  Note: This function was renamed to <code>im2java</code> and moved into MATLAB in release 3.2 of the Image Processing Toolbox.
<code>imnoise</code>	Added support for new noise types: 'poisson' and 'localvar'

<b>Function</b>	<b>Description</b>
label2rgb	Convert label matrix to RGB image [New for Version 3.1]
padarray	Pad array
regionprops	Renamed from existing function <code>imfeature</code> ; extended to N-D
stretchlim	Find limits to contrast stretch an image

## New Demos

The Image Processing Toolbox 3.1 includes the 15 new extended example demos, presented in HTML form.

<b>Demo Name</b>	<b>Brief Description</b>
ipexconformal	Explore a Conformal Mapping: illustrates how to use spatial- and image-transformation functions to perform a conformal mapping.
ipexdeconvblind	Deblurring Images Using the Lucy-Richardson algorithm: illustrates use of the <code>deconvlucy</code> function. [New with Version 3.1]
ipexdeconvlucy	Deblurring Images Using the Lucy-Richardson algorithm: illustrates use of the <code>deconvlucy</code> function.
ipexdeconvreg	Deblurring Images Using a Regularized Filter: illustrates use of the <code>deconvreg</code> function.
ipexdeconvwnr	Deblurring Images Using the Wiener Filter: illustrates use of the <code>deconvwnr</code> function.
ipexgranulometry	Finding the Granulometry of Stars in an Image: illustrates how to use morphology functions to perform granulometry.

<code>ipexmri</code>	Extracting Slices from a 3-Dimensional MRI Data Set: illustrates how to use the image transformation functions to interpolate and reslice a three-dimensional MRI data set, providing a convenient way to view a volume of data.
<code>ipexnormxcorr2</code>	Registering an Image Using Normalized Cross-correlation: illustrates how to use translation to align two images.
<code>ipexregaerial</code>	Registering an Aerial Photo to an Orthophoto: illustrates how to use the Control Point Selection Tool to align two images.
<code>ipexrotate</code>	Finding the Rotation and Scale of a Distorted Image: illustrates how to use the <code>cp2tform</code> function to get the rotation angle and scale factor of a distorted image.
<code>ipexsegcell</code>	Detecting a Cell Using Image Segmentation: illustrates how to use dilation and erosion to perform edge detection.
<code>ipexsegmicro</code>	Detecting Microstructures Using Image Segmentation: illustrates how to use morphological opening and closing to extract large objects from an image.
<code>ipexsegwatershed</code>	Detecting Touching Objects Using Watershed Segmentation: illustrates use of morphology functions to perform marker-control watershed segmentation.
<code>ipexshear</code>	Padding and Shearing an Image Simultaneously: illustrates how to use the padding options of the image transformation functions.
<code>ipextform</code>	Creating a Gallery of Transformed Images: illustrates how to use the <code>imtransform</code> function to perform many types of image transformations.

## Major Bug Fixes

This section describes major bug fixes included in the Image Processing Toolbox, Version 3.0 and Version 3.1.

For information about additional bug fixes that are incorporated from Version 2.2.2, see “Major Bug Fixes” on page 5-3.

### Version 3.1 Bug Fixes

- `fspecial` — Fixed incorrect normalization for the Gaussian filter option.
- `improfile` — Fixed an occasional indexing problem caused by round-off error.
- `rgb2ind` — Fixed a problem that caused `rgb2ind` to produce bad results for very large images.
- Functions that operate on binary input images now treat NaNs in a consistent manner. When an input array that is expected to be a binary image contains NaN values, the NaN value is always treated as 1.



# Image Processing Toolbox

## 2.2.2 Release Notes

---

<b>New Features</b> . . . . .	5-2
New Demo . . . . .	5-2
Support For Function Handles . . . . .	5-2
Documentation Enhanced . . . . .	5-2
<b>Major Bug Fixes</b> . . . . .	5-3
imshow Fixes . . . . .	5-3
bwlabel Segmentation Violation Eliminated . . . . .	5-3
dilate And erode Return Correct Answers . . . . .	5-3
freqz2 Fixes . . . . .	5-3
fspecial Function's 'LoG' Option . . . . .	5-3
Improved Display for imcrop, improfile, and roipoly . . . . .	5-4

## New Features

The focus of the Image Processing Toolbox 2.2.2 is on bug fixes (see “Major Bug Fixes” below).

A number of important new features will be released in the Image Processing Toolbox 3.0, which will be made available in a Web downloadable version after Release 12.0 is released.

### New Demo

The Image Processing Toolbox 2.2.2 includes the new `landsatdemo` function, which is a demo that illustrates how to construct color composite images from multispectral Landsat data.

### Support For Function Handles

The following functions have been updated to support function handles, a new MATLAB 6.0 language feature:

- `blkproc`
- `colfilt`
- `nlfilter`
- `qtdecomp`
- `roifilt2`

The MATLAB language has a new data type called the function handle. The function handle captures all the information about a function that MATLAB needs to evaluate it. You can pass a function handle in an argument list to other functions.

### Documentation Enhanced

The online *Image Processing Toolbox User's Guide* was enhanced for Release 12 by adding a “Getting Started” section, and by adding glossaries of relevant terms at the beginning of several chapters.



## Major Bug Fixes

The Image Processing Toolbox 2.2.2 includes several important bug fixes that were made in the Image Processing Toolbox 2.2.1 (Release 11.1). This section describes the bugs and how they have been fixed.

### **imshow Fixes**

You can now display the same image twice using `imshow`, without the previous problem of having the images appear to move slightly the second time.

Also, you can now use the syntax `imshow(I, [])` when all the elements of `I` are the same. Now `imshow` displays `I` using an intermediate shade of gray. Previously, `imshow` would generate an error for this case. (This fix was introduced in the Image Processing Toolbox 2.2.1 (Release 11.1).)

### **bwlabel Segmentation Violation Eliminated**

You can now pass a matrix to `bwlabel` that contains values other than 0 or 1. `bwlabel` treats any nonzero element as an object element. Previously, `bwlabel` would cause a segmentation violation for this case. (This fix was introduced in the Image Processing Toolbox 2.2.1 (Release 11.1).)

### **dilate And erode Return Correct Answers**

The `dilate` and `erode` functions now return the correct answer in all cases. In prior versions of the Image Processing Toolbox, in some cases these functions returned the incorrect answer if you specified the frequency-domain option with a structuring element that contained more than 255 elements.

### **freqz2 Fixes**

The `freqz2` function now returns correct values for the frequency scaling. Also, `freqz2` no longer uses an excessive amount of memory.

### **fspecial Function's 'LoG' Option**

The `Log` option of the `fspecial` function now returns correctly scaled values.

## **Improved Display for `imcrop`, `improfile`, and `roipoly`**

The animated lines that the `imcrop`, `improfile`, and `roipoly` functions display on top of images are now displayed clearly.