Image Processing Toolbox™ Release Notes

Contents

Summary by Version 1

Version 7.2 (R2011a) Image Processing Toolbox 5

Version 7.1 (R2010b) Image Processing Toolbox 9

Version 7.0 (R2010a) Image Processing Toolbox 12

Version 6.4 (R2009b) Image Processing Toolbox 16

Version 6.3 (R2009a) Image Processing Toolbox 22

Version 6.2 (R2008b) Image Processing Toolbox 26

Version 6.1 (R2008a) Image Processing Toolbox 33

Version 6.0 (R2007b) Image Processing Toolbox 38

Version 5.4 (R2007a) Image Processing Toolbox 43

Version 5.3 (R2006b) Image Processing Toolbox 47

Version 5.2 (R2006a) Image Processing Toolbox 50

Version 5.1 (R14SP3) Image Processing Toolbox 52

Version 5.0.2 (R14SP2) Image Processing Toolbox 55

Version 5.0.1 (R14SP1) Image Processing Toolbox 60

Version 5.0 (R14) Image Processing Toolbox 61

Version 4.2 (R13SP2) Image Processing Toolbox **85**

Version 4.1 (R13SP1) Image Processing Toolbox **87**

Version 4.0 (R13+) Image Processing Toolbox **90**

Version 3.2 (R13+) Image Processing Toolbox **99**

Version 3.1 (R12.1) Image Processing Toolbox **103**

Version 3.0 (R12+) Image Processing Toolbox **104**

Version 2.2.2 (R12) Image Processing Toolbox **113**

Compatibility Summary for Image Processing Toolbox **116**

Summary by Version

This table provides quick access to what's new in each version. For clarification, see "Using Release Notes" on page 2.

Version (Release)	New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Latest Version V7.2 (R2011a)	Yes Details	Yes Summary	Bug Reports Includes fixes
V7.1 (R2010b)	Yes Details	Yes Summary	Bug Reports Includes fixes
V7.0 (R2010a)	Yes Details	Yes Summary	Bug Reports Includes fixes
V6.4 (R2009b)	Yes Details	Yes Summary	Bug Reports Includes fixes
V6.3 (R2009a)	Yes Details	Yes Summary	Bug Reports Includes fixes
V6.2 (R2008b)	Yes Details	Yes Summary	Bug Reports Includes fixes
V6.1 (R2008a)	Yes Details	Yes Summary	Bug Reports Includes fixes
V6.0 (R2007b)	Yes Details	Yes Summary	Bug Reports Includes fixes
V5.4 (R2007a)	Yes Details	Yes Summary	Bug Reports Includes fixes
V5.3 (R2006b)	Yes Details	Yes Summary	Bug Reports Includes fixes
V5.2 (R2006a)	Yes Details	Yes Summary	Bug Reports Includes fixes

Version (Release)	New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
V5.1 (R14SP3)	Yes Details	Yes Summary	Bug Reports Includes fixes
V5.0.2 (R14SP2)	No	Yes Summary	Fixed bugs Details
V5.0.1 (R14SP1)	No	No	Fixed bugs Details
V5.0 (R14)	Yes Details	Yes Summary	Fixed bugs Details
V4.2 (R13SP2)	Yes Details	No	Fixed bugs Details
V4.1 (R13SP1)	Yes Details	No	Fixed bugs Details
V4.0 (R13+)	Yes Details	Yes	Fixed bugs Details
V3.1 (R12.1)	Yes Details	Yes Summary	Fixed bugs Details
V3.0 (R12+)	Yes Details	No	No
V2.2.2 (R12)	Yes Details	No	Fixed bugs Details

Using Release Notes

Use release notes when upgrading to a newer version to learn about:

- New features
- Changes
- Potential impact on your existing files and practices

Review the release notes for other MathWorks® products required for this product (for example, MATLAB® or Simulink®). Determine if enhancements, bugs, or compatibility considerations in other products impact you.

If you are upgrading from a software version other than the most recent one, review the current release notes and all interim versions. For example, when you upgrade from V1.0 to V1.2, review the release notes for V1.1 and V1.2.

What Is in the Release Notes

New Features and Changes

- New functionality
- Changes to existing functionality

Version Compatibility Considerations

When a new feature or change introduces a reported incompatibility between versions, the **Compatibility Considerations** subsection explains the impact.

Compatibility issues reported after the product release appear under Bug Reports at the MathWorks Web site. Bug fixes can sometimes result in incompatibilities, so review the fixed bugs in Bug Reports for any compatibility impact.

Fixed Bugs and Known Problems

MathWorks offers a user-searchable Bug Reports database so you can view Bug Reports. The development team updates this database at release time and as more information becomes available. Bug Reports include provisions for any known workarounds or file replacements. Information is available for bugs existing in or fixed in Release 14SP2 or later. Information is not available for all bugs in earlier releases.

Access Bug Reports using your MathWorks Account.

Documentation on the MathWorks Web Site

Related documentation is available on `mathworks.com` for the latest release and for previous releases:

- Latest product documentation
- Archived documentation

Version 7.2 (R2011a) Image Processing Toolbox

This table summarizes what's new in Version 7.2 (R2011a).

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	Yes—Details labeled as **Compatibility Considerations**, below. See also Summary.	Bug Reports Includes fixes

The following sections describe new features and changes introduced in this version:

- "New bwconvhull Function Computes Convex Hull Image" on page 5
- "New dicomwrite Option Writes Multiframe Imagery to Single File" on page 6
- "nitfread Now Reads NITF Files with JPEG-Compressed Images" on page 6
- "Reduced Memory Use for std2" on page 6
- "Reduced Memory Use for watershed" on page 6
- "iccread and iccwrite Now Warn in Cases of Unrecognized PrimaryPlatform Signatures" on page 7
- "Plot Selector Now Includes implay" on page 7
- "Support for Code Generation from MATLAB" on page 7
- "edge Function No Longer Smooths Image Twice" on page 7
- "Functions and Function Elements Being Removed" on page 8

New bwconvhull Function Computes Convex Hull Image

Use `bwconvhull` to compute the convex hull image from a binary image.

New dicomwrite Option Writes Multiframe Imagery to Single File

Set the new `'MultiframeSingleFile'` option of the `dicomwrite` function to `true` to write multiframe imagery to one file, regardless of how many frames the input image contains.

nitfread Now Reads NITF Files with JPEG-Compressed Images

If you have NITF files containing JPEG-compressed images, you can now read them using `nitfread`.

Reduced Memory Use for std2

The `std2` function has improved performance for large 1-, 8-, and 16-bit integers.

Compatibility Considerations

Compared to releases before R2011a, the `std2` function now returns slightly different results for some images. To receive the same results as previously, use this code:

```
% For input image im
if ~isa(im,'double')
    im = double(im);
end
std_old = std(im(:));
```

Reduced Memory Use for watershed

The `watershed` function is now more memory efficient than it was in releases before R2011a.

Compatibility Considerations

The watershed regions in the label matrix returned by `watershed` have different indices than they did before R2011a.

Also, the label matrix returned by `watershed` was class `double` in previous releases, and is now an unsigned integer class.

If you want to return a label matrix of class `double`, as you did before, use the `double` function to convert it:

```
L = watershed(A);
L = double(L);
```

iccread and iccwrite Now Warn in Cases of Unrecognized PrimaryPlatform Signatures

If `iccread` or `iccwrite` encounter an unrecognized PrimaryPlatform signature in the profile header, they will warn. In releases before R2011a, these functions would error instead of warn in cases with unrecognized PrimaryPlatform signatures.

Plot Selector Now Includes implay

The `implay` function has been added to the list of functions available in the Plot Selector. You can now display data in `implay` directly from the Plot Selector workspace tool. For details about the Plot Selector, see Enhanced Plot Selector Simplifies Data Display.

Support for Code Generation from MATLAB

You can now generate standalone C code for two Image Processing Toolbox™ functions: `label2rgb` and `fspecial`. The generated C code meets the strict memory and data type requirements of embedded target environments. To generate this code you need a MATLAB Coder license. See the Code Generation for Image Processing Toolbox Functions chapter in the User's Guide for details, including limitations.

edge Function No Longer Smooths Image Twice

In previous releases, the implementation of the Canny filter, called with the `canny` method of the `edge` function, smoothed the image twice while constructing the gradient image. The function smoothed the image once using a Gaussian filter and then used a first derivative of a Gaussian filter to extract a smoothed version of the image gradient. Smoothing an image and then differentiating it is the same as convolving the image with a derivative of the smoothing kernel, so this implementation had the effect of smoothing the image twice. In addition, the original implementation of the Canny filter included an extra morphological thinning step that is not in the published algorithm.

Compatibility Considerations

The `edge` function no longer smooths an image twice. If you are setting the value of `sigma` and want similar results to the previous implementation, increase `sigma` by a factor of `sqrt(2)`.

To achieve the same results produced by the previous implementation, use this syntax:

```
BW = edge(I,'canny_old',...)
```

Functions and Function Elements Being Removed

Function or Function Element	What Happens When You Use This Function or Element?	Use This Instead	Compatibility Considerations
`ipttable`	Errors	`uitable`	Replace all existing instances of `ipttable` with `uitable`.

Version 7.1 (R2010b) Image Processing Toolbox

This table summarizes what's new in Version 7.1 (R2010b).

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	Yes—Details labeled as **Compatibility Considerations**, below. See also Summary.	Bug Reports Includes fixes

The following sections describe new features and changes introduced in this version:

- "New corner Function Detects Corners in Image" on page 9
- "Efficient Display and Navigation of Very Large Images of Arbitrary Format in imtool" on page 10
- "Now Possible to Control Padding Behavior when Using the blockproc Function" on page 10
- "Writing to JPEG2000 File Format Supported by blockproc" on page 10
- "Enhancements to the dicomread Function" on page 10
- "The ImageMagnification Field of the nitfinfo Function Now Returns a Numeric Value" on page 11
- "Performance Improvements" on page 11
- "Functions and Function Elements Being Removed" on page 11

New corner Function Detects Corners in Image

The new `corner` function detects corners in a grayscale or binary image. Corners are a feature you can use to find the correspondence between images.

Compatibility Considerations

In R2008b and later releases, you could find corners by computing a `cornermetric` matrix with the `cornermetric` function and then finding peak values. Now, you can simplify your workflow by using the `corner` function.

Efficient Display and Navigation of Very Large Images of Arbitrary Format in imtool

The `rsetwrite` function allows you to create a multiresolution image pyramid (R-Set) from a large image file. In previous releases, the `rsetwrite` function accepted TIFF or NITF image files. Now, in addition to accepting these image files, `rsetwrite` accepts `ImageAdapter` objects. `ImageAdapter` objects allow you to work with images of arbitrary file format. See "Writing an Image Adapter Class" for guidelines on how to create an `ImageAdapter` object.

Now Possible to Control Padding Behavior when Using the blockproc Function

With the new `'PadMethod'` option, you can now control padding behavior when using the `blockproc` function. In previous releases, the `blockproc` function only supported zero padding along the boundary of the image. Now, the function supports padding the image with a scalar pad value, repeated border elements of `A`, or the mirror reflection of `A`.

Writing to JPEG2000 File Format Supported by blockproc

The `blockproc` function now provides more flexibility in format choices. The function supports writing to JPEG2000 file formats with `*.jp2`, `*j2c`, or `*.j2k` extensions.

Enhancements to the dicomread Function

The `dicomread` function has been enhanced in two ways: the function now reads multiframe imagery faster, and it reads files containing JPEG-2000 encoded imagery.

The ImageMagnification Field of the nitfinfo Function Now Returns a Numeric Value

The `ImageMagnification` field of the `nitfinfo` function has been updated. Previously, if you used the function to return a structure with file-level metadata, the `ImageMagnification` field of the structure contained an incorrect value. (The incorrect value was either an empty image magnification value or the text value for the field.) Now, the `ImageMagnification` field returns the value for the image magnification.

Performance Improvements

Faster Functions

- `blockproc`
- `imresize`
- `iradon`

Functions and Function Elements Being Removed

Function or Function Element Name	What Happens When You Use This Function or Element?	Use This Instead	Compatibility Considerations
`blkproc`	Still runs	`blockproc`	None

Version 7.0 (R2010a) Image Processing Toolbox

This table summarizes what's new in Version 7.0 (R2010a).

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	Yes—Details labeled as **Compatibility Considerations**, below. See also Summary.	Bug Reports Includes fixes

The following sections describe new features and changes introduced in this version:

- "New ImageAdapter Class Supports Custom File Formats for blockproc" on page 12
- "The blockproc Function Now Supports Spatially Varying Operations" on page 13
- "Plot Selector Now Generates Plots for imshow and imtool" on page 13
- "makecform Now Supports White Point Adaptation" on page 13
- "Intel Integrated Performance Primitives Library Support Extended to imdilate, imerode, and medfilt2" on page 13
- "imreconstruct Now Supports int64 and uint64" on page 14
- "Performance Improvements" on page 14
- "Non-interactive Syntax of improfile Returns Different Output" on page 15

New ImageAdapter Class Supports Custom File Formats for blockproc

The `blockproc` function, introduced in R2009b, supported file-based block processing for arbitrarily large images. In R2009b, you could use `blockproc` to read or write TIFF images or to read JPEG2000 images. Now, with the addition of the new `ImageAdapter` class, you can design your own class to use `blockproc` with images of arbitrary file format.

The blockproc Function Now Supports Spatially Varying Operations

Additional fields have been added to the `blockproc` “block struct” that contain spatial information. These new fields facilitate operations that depend on location.

Plot Selector Now Generates Plots for imshow and imtool

The Plot Selector workspace tool creates graphs of workspace variables. The `imshow` and `imtool` functions have been added to the list of possible plotting functions available in the Plot Selector. For more information about the Plot Selector, see Enhanced Plot Selector Simplifies Data Display.

makecform Now Supports White Point Adaptation

`makecform` uses the white point specified by the International Color Consortium (ICC) as the default for the `srgb2lab` and `lab2srgb` transform types. You can now adapt to a white point other than `whitepoint('ICC')`, the default value, by using a new syntax to specify the adapted white point:

```
C = makecform(type, 'AdaptedWhitePoint', WP)
```

You can also create a linear chromatic-adaptation transform:

```
C = makecform('adapt', 'WhiteStart', WPS, 'WhiteEnd', WPE, ...
    'AdaptModel', modelname)
```

This transform allows you to adapt *XYZ* color values from one white point to another.

Intel Integrated Performance Primitives Library Support Extended to imdilate, imerode, and medfilt2

The functions `imdilate` and `imerode` are now hardware optimized for `ones(3)` neighborhoods for `single`, `uint8`, and `uint16` input images.

The `medfilt2` function is now hardware optimized for integer data types (`uint8`, `uint16`, and `int16`) and the `single` data type with kernel size 3 x 3.

imreconstruct Now Supports int64 and uint64

The `imreconstruct` function now supports data types `int64` and `uint64`.

Performance Improvements

Faster Functions

- `edge`
- `imdilate`
- `imerode`
- `imfilter`
- `imresize`
- `iradon`
- `medfilt2`

Multithreaded Functions

- `bwmorph`
- `edge`
- `imabsdiff`
- `imadd`
- `imclose`
- `imdivide`
- `immultiply`
- `imopen`
- `iradon`
- `medfilt2`

Non-interactive Syntax of improfile Returns Different Output

One of the non-interactive syntaxes of `improfile` now returns different output. The output for the syntax

```
C = improfile(I,xi,yi,N)
```

has changed. In the syntax above, `N` specifies the number of points for which to compute intensity values and `xi` and `yi` specify the spatial coordinates of the endpoints of the line segments.

For a given line defined by `xi` and `yi`, `improfile` now returns a profile sampled at both endpoints and all sampling points in between at roughly unit interval spacing. If the distance between `xi` and `yi` is `N` pixels, the profile is evaluated at `N+1` points.

Compatibility Considerations

In previous releases, if you supplied the `xi` and `yi` end points as (1,1) and (10,1), the profile would be evaluated at nine points, the nine unit-length intervals between 1 and 10 in the continuous *x-y* plane. These nine points would be the two end points, plus seven points in between.

Version 6.4 (R2009b) Image Processing Toolbox

This table summarizes what's new in Version 6.4 (R2009b).

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	Yes—Details labeled as **Compatibility Considerations**, below. See also Summary.	Bug Reports Includes fixes

The following sections describe new features and changes introduced in this version:

- "New blockproc Function to Process Large Images" on page 16
- "Intel Integrated Performance Primitives Library Upgraded and Support Extended to maci64" on page 18
- "Expanded hough Function Allows Specification of Arbitrary Theta Search Space" on page 18
- "The imfilter Function Now Faster for uint16 and double Inputs" on page 19
- "Improved Speed for Calculating N-D Euclidean Distance Transforms with the bwdist Function" on page 19
- "Modified Behavior for the regionprops ConvexHull Property" on page 20
- "Efficient Display and Navigation of Very Large NITF-File Images in imtool" on page 20
- "Performance Improvements" on page 20

New blockproc Function to Process Large Images

The new `blockproc` function supports file-based block processing for arbitrarily large TIFF images. The new function supports in-memory operations as well as file-to-file processing of images which are too large to load completely into memory.

Compatibility Considerations

In previous releases, you could use the `blkproc` function for in-memory block-processing of images. The `blkproc` function will be removed in a future release. Replace all instances of `blkproc` with `blockproc`.

When updating your code from `blkproc` to `blockproc`, it is important to note that the user-defined function, `fun`, has a new signature. It now takes a structure, the "block struct," as input instead of simply a matrix of image data.

The examples below demonstrate how to update your code from `blkproc` to `blockproc`.

Example One: DCT2.

```
% BLKPROC code
I = imread('cameraman.tif');
fun = @dct2;
J = blkproc(I,[8 8],fun);

% BLOCKPROC equivalent (using an anonymous function)
fun = @(block_struct) dct2(block_struct.data);
J = blockproc(I,[8 8],fun);
```

Example Two: Filtering.

```
% BLKPROC code
I = imread('concordorthophoto.png');
h = fspecial('gaussian',[11 11],2.5);
fun = @(x) imfilter(x,h,'conv','same');
J = blkproc(I,[500 500],[5 5],fun);

% BLOCKPROC equivalent (using an anonymous function)
fun = @(block_struct) imfilter(block_struct.data,h,'conv','same');
J = blockproc(I,[500 500],fun,'BorderSize',[5 5]);
```

Intel Integrated Performance Primitives Library Upgraded and Support Extended to maci64

The Intel® Integrated Performance Primitives (Intel IPP) Library has been upgraded from Version 5.3.1 to Version 6.0 Update 1. Intel IPP Library support has been extended to 64-bit Intel-based Mac computers.

Expanded hough Function Allows Specification of Arbitrary Theta Search Space

The `hough` function now yields faster results for narrower *theta* ranges due to the addition of a parameter/value pair for specifying *theta* values.

Compatibility Considerations

In previous releases, the `'ThetaResolution'` parameter controlled the *theta* values for the `hough` function. Now `'ThetaResolution'` is being replaced by the new `'Theta'` parameter.

Function Elements Being Removed

Function and Syntax	What Happens When You Use the Function or Element?	Use This Instead	Compatibility Considerations
`hough(BW, 'ThetaResolution',val)`	Still runs	`hough(BW,'Theta', -90:val:(90-val))`	Input parameter no longer recommended. Use new `'Theta'` parameter.

Note that with the introduction of the `'Theta'` parameter, not all abbreviated forms of `'ThetaResolution'` will still work. In previous releases, if you entered the following syntax:

```
hough(BW, 'T', val)
```

`'T'` stood for `'ThetaResolution'`. Now if you enter this same syntax, `'T'` stands for the new `'Theta'` parameter.

If you have old code that uses the `'ThetaResolution'` parameter, please see the definition below:

Parameter	Description
`'ThetaResolution'`	Real scalar value between 0 and 90, exclusive, that specifies the spacing (in degrees) of the Hough transform bins along the *theta* axis. Default: 1.

For `'ThetaResolution'`, `ntheta = 2*ceil(90/ThetaResolution)`. *theta* angle values are in the range [-90, 90) degrees. If `90/ThetaResolution` is not an integer, the actual angle spacing is `90/ceil(90/ThetaResolution)`.

The imfilter Function Now Faster for uint16 and double Inputs

The `imfilter` function now runs faster with `uint16` and `double` inputs than in previous releases. This performance enhancement is due to the use of the Intel IPP Library with inputs of these types.

Compatibility Considerations

Using the Intel IPP library for `uint16` images, poses no compatibility issues. The same is not true, however, for the `double` data type.

If an input image contains NaN values and a filtering kernel contains zero values, the `imfilter` function now gives different results when the Intel IPP library is enabled versus when it is disabled. If you want to preserve the behavior of previous releases, use `iptsetpref('UseIPPL',false)` to disable the Intel IPP library.

Improved Speed for Calculating N-D Euclidean Distance Transforms with the bwdist Function

A new algorithm improves the speed and reduces the memory footprint for the `bwdist` function.

Compatibility Considerations

In previous releases, the `bwdist` function used different algorithms for computing the Euclidean distance transform and the associated label matrix. If you need the same results produced by the previous implementation, use the function `bwdist_old`.

Modified Behavior for the regionprops ConvexHull Property

The `'ConvexHull'` property of `regionprops` depends on the MATLAB `convhull` function. Due to changes in `convhull`, the results returned by `'ConvexHull'` will now be slightly different than in previous releases.

Compatibility Considerations

The order of the vertices returned by the `'ConvexHull'` property of `regionprops` may differ from that returned in releases before R2009b. Also, the returned hull may contain additional collinear points that were omitted in previous releases.

Efficient Display and Navigation of Very Large NITF-File Images in imtool

The `rsetwrite` function allows you to create multi-resolution image pyramids (R-Sets) that you can open in `imtool`. In previous releases, `rsetwrite` worked only with TIFF files. Now it accepts NITF files, as well, as long as they are Version 2.0 or greater, contain an uncompressed image, have integer data (no floating point data), and have three or fewer image bands. Finally, if a NITF file has more than one band of data, the data must be unsigned.

Performance Improvements

The performance of several existing toolbox functions has been improved in this release. In some cases, other toolbox functions call these functions and therefore will also benefit from these speed improvements.

Faster Functions

- `bwdist`

- `imcomplement`
- `imdilate`
- `imerode`
- `imfilter`
- `improfile`
- `imrotate`

Multithreaded Functions

- `applylut`
- `bwpack`
- `bwunpack`
- `imdilate`
- `imerode`
- `imreconstruct`

Version 6.3 (R2009a) Image Processing Toolbox

This table summarizes what's new in Version 6.3 (R2009a).

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	Yes—Details labeled as **Compatibility Considerations**, below. See also Summary.	Bug Reports Includes fixes

The following sections describe new features and changes introduced in this version:

- "Faster, Less Memory-Intensive Workflow for Labeling Regions and Measuring Their Properties" on page 23
- "Multithreaded Implementation of imfilter Function" on page 23
- "Efficient Display and Navigation of Very Large Images in imtool" on page 23
- "New Dialog Box for Setting Toolbox Preferences" on page 23
- "New imcolormaptool Function That Opens Choose Colormap Tool" on page 24
- "End Point and Branch Point Detection Now Possible" on page 24
- "nitfread Now Allows Image Subregion Selection" on page 24
- "Support for Intel IPP on Mac" on page 24
- "getColor, getLabelVisible, and setLabelVisible Methods Added to imdistline" on page 25
- "Five Functions Moved to MATLAB" on page 25
- "Fan-Beam Functions Updated" on page 25

Faster, Less Memory-Intensive Workflow for Labeling Regions and Measuring Their Properties

The `bwconncomp` function computes connected components for binary images. It uses significantly less memory and is sometimes faster than `bwlabel` and `bwlabeln`.

To extract features from a binary image using `regionprops` with default connectivity, just pass `BW` directly into `regionprops` (i.e., `regionprops(BW)`). To compute a label matrix having more memory-efficient data type (e.g., `uint8` versus `double`), use the `labelmatrix` function on the output of `bwconncomp`.

Multithreaded Implementation of imfilter Function

The `imfilter` function is now multithreaded.

Efficient Display and Navigation of Very Large Images in imtool

The new `rsetwrite` function allows you to create a multi-resolution image pyramid (R-Set) from a large TIFF image file. In previous releases, large images would not open in `imtool`, or they did open, but navigation was slow. You can now open your R-Set with `imtool` and explore it as you would a standard image.

New Dialog Box for Setting Toolbox Preferences

A new preferences dialog box allows customization of Image Processing Toolbox preferences. You can access the dialog box via the **File** menu in the MATLAB desktop, the **File** menu in the Image Tool (`imtool`), or directly from the command line by typing `iptprefs`.

A new preference has been added that allows you to specify whether the Overview tool opens automatically when you launch the Image Tool.

Compatibility Considerations

In previous releases, the Overview tool opened automatically with `imtool`. The new default behavior is for the Overview tool to no longer open automatically. If you would like to revert to the previous behavior you can set this preference via the Image Processing Preferences dialog box (`iptprefs`).

In previous releases, if you changed the preferences with the `iptsetpref` command, these changes would revert to the default setting when you finished a MATLAB session. Now, if you change preferences, these changes will remain intact from one MATLAB session to the next.

New imcolormaptool Function That Opens Choose Colormap Tool

The new function `imcolormaptool` opens the Choose Colormap tool. The Choose Colormap tool allows you to interactively change the colormap of a displayed image. You can also access the tool from the **Tools** menu of the Image Tool, as in previous releases.

End Point and Branch Point Detection Now Possible

`bwmorph` now detects end points and branch points in binary images.

nitfread Now Allows Image Subregion Selection

The `nitfread` function now includes a `PixelRegion` parameter that returns a subimage as specified by row and column vectors.

Compatibility Considerations

From R2007a to R2008b, the `nitfread` function returned `uint8` data for images with 1-bit data. Now `nitfread` returns `logical` data for images with 1-bit data. If you want this function to behave as it did in the past, enter the following:

```
  imdata = uint8(nitfread(filename));
```

Support for Intel IPP on Mac

In previous releases, the Image Processing Toolbox leveraged the Intel Integrated Performance Primitives (Intel IPP) Library on 32- and 64-bit Linux® and Windows® platforms. Now Intel IPP-use has been extended to the Mac®.

getColor, getLabelVisible, and setLabelVisible Methods Added to imdistline

imdistline now includes a getColor method that returns the color used to draw a specific ROI object. Also, the new getLabelVisible and setLabelVisible methods make it possible to control the visibility of the Distance tool text label.

Five Functions Moved to MATLAB

The following five functions moved from the Image Processing Toolbox to MATLAB: cmpermute, cmunique, dither, imapprox, and rgb2ind. The behavior of some of the functions has changed slightly, as described in the compatibility considerations listed below.

Compatibility Considerations

- Functions dither and imapprox, when called without output arguments, no longer display their output as an image via a call to imshow. Now, if you want to display the resulting image, assign the output to one or more variables and call imshow. For example, try the following:

  ```
  [Y,newmap] = imapprox(X,map,n)
  imshow(Y,newmap)
  ```

- Function rgb2ind errors when called with the syntax rgb2ind(RGB). You must specify the number of colors, tolerance, or colormap. For example, you can use the syntax rgb2ind(RGB,128), where 128 represents the number of colors.
- Function imapprox errors when called with the syntax imapprox(x,map). As with rgb2ind, you must specify additional parameters.

Fan-Beam Functions Updated

Compatibility Considerations

Due to a bug fix, the fan-beam functions (fanbeam, ifanbeam, fan2para, para2fan) now return different answers than in previous releases.

Version 6.2 (R2008b) Image Processing Toolbox

This table summarizes what's new in Version 6.2 (R2008b).

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	Yes—Details labeled as **Compatibility Considerations**, below. See also Summary.	Bug Reports Includes fixes

The following sections describe new features and changes introduced in this version.

- "Performance Improvements" on page 27
- "New cornermetric Function Detects Corners" on page 27
- "Now Support Absolute Colorimetric Rendering Intent for GrayTRC and MatTRC" on page 27
- "New createMask Method Creates Mask for Any ROI" on page 27
- "Interactive Tools Refresh when Target Image Changes" on page 27
- "The imscrollpanel 'PreserveView' Parameter Now Works for Images of All Sizes" on page 28
- "Distance Tool and Cropping Tool Now Modes in imtool" on page 28
- "In imtool Opening Adjust Contrast Tool No Longer Selects Window/Level Tool" on page 28
- "immovie Command No Longer Shows Preview" on page 29
- "Replace Calls toipttable Function with MATLAB uitable Function" on page 29
- "imcontour Second Output Argument Changed" on page 30
- "impixelinfo Tool Disappears when Image Changes" on page 30
- "Some Code Moved into Different Directories" on page 31

- "Functions and Demos Being Removed" on page 31

Performance Improvements

The performance of several existing toolbox functions has been improved in this release, including:

- Binary erosion and dilation (`imdilate`, `imerode`, `bwhitmiss`, and `rangefilt`)
- `graycomatrix`
- Image arithmetic and filtering now leverage the IPP Library on 32- and 64-bit Windows and Linux platforms.

New cornermetric Function Detects Corners

New `cornermetric` function detects corners.

Now Support Absolute Colorimetric Rendering Intent for GrayTRC and MatTRC

New additions to the `makecform` syntax include rendering intents for the Matrix/Tone Reproduction Curve (`MatTRC`) model and the single-channel Tone Reproduction Curve (`GrayTRC`) model.

New createMask Method Creates Mask for Any ROI

Use the new `createMask` method of the `imroi` base class to return a mask, or binary image, that is the same size as the input image with 1s inside the ROI object and 0s outside. The new method is available in the following classes: `impoint`, `imline`, `imrect`, `imellipse`, `impoly`, and `imfreehand`.

Interactive Tools Refresh when Target Image Changes

The following modular interactive tools now update automatically if you modify the target image: Adjust Contrast, Pixel Region, Pixel Information, Overview, Display Range, and Image Information.

The imscrollpanel 'PreserveView' Parameter Now Works for Images of All Sizes

The `replaceImage` function in the `imscrollpanel` API has been modified. You can now use the `'PreserveView'` parameter even in cases where your replacement image is not the same size as your original image. The new image will appear with the center of view in the same relative position as in the original image.

Compatibility Considerations

In previous releases, the default for different size images was for the new image to appear centered and at 100% magnification.

Distance Tool and Cropping Tool Now Modes in imtool

The Distance tool and Cropping tool have been modified. Now to use the Distance tool, you click one end of the distance to be measured, drag, and release to complete the measurement. With the new version of the Cropping tool, you may click and drag to define the cropping region as many times as you want. If you define one region and then decide to crop a different region instead, simply click and drag the mouse again to define the new region.

Compatibility Considerations

In previous releases, the Distance tool appeared as a horizontal bar of set length. You could drag the ends of the tool to change size and orientation.

In previous releases, if you defined one cropping region, you could move this box or change the size, but you couldn't start with a new box unless you canceled the tool, clicked on the "Crop Image" toolbar button, and then defined the new region.

In imtool Opening Adjust Contrast Tool No Longer Selects Window/Level Tool

When you open the Adjust Contrast tool, the Window/Level tool is no longer turned on automatically. To operate this feature, simply select the Window/Level tool icon from the Image Tool toolbar. (To identify the icon, note that if you move the cursor over the Window/Level tool icon, the words "Adjust contrast/brightness via mouse motion" appear.) Or, you can select "Tools" from the Image Tool menu and then click on "Window/Level."

Compatibility Consideration

In previous releases, when you opened the Adjust Contrast tool, the Window/Level tool automatically turned on at the same time. Note that the Window/Level tool still turns on when you call `imcontrast` from the command line.

immovie Command No Longer Shows Preview

`immovie` no longer opens a figure window to display the movie as it is being created. You can display and explore the output of `immovie` using `implay`.

Compatibility Consideration

If you want to use `movie` to visualize the output but don't know how to set up the figure appropriately, call `imshow` on one of the movie frames first before calling `movie`.

Replace Calls to ipttable Function with MATLAB uitable Function

The `ipttable` function is being deprecated and will be removed in a future release.

Compatibility Consideration

If you used the `ipttable` function to display tabular data, you should replace use of `ipttable` with the MATLAB function `uitable`.

If you used the cell array syntax of `ipttable`, make the following changes to your code to achieve a similar effect using `uitable`. For more information about using `uitable`, see the `uitable` function reference page.

R2008a Code	R2008b Code
`table = ipttable(parent,cell_array_data);`	`table = uitable(parent,'Data',cell_array_data);`

If you used the struct syntax of `ipttable`, make the following changes to your code to achieve a similar effect with `uitable`.

R2008a Code	R2008b Code

`table =`
`ipttable(parent,struct_data);`
	`field_names = fieldnames(struct_data);`
	`values = struct2cell(struct_data);`
	`for idx = 1:numel(values)`
	`   val = values{idx};`
	`   if ~ischar(val) || size(val,1) > 1`
	`      values{idx} =`
	`evalc('disp(values{idx})');`
	`   end`
	`end`
	`table = uitable(parent,'Data',`
	`      [field_names values]);`

Code written in previous releases that depends on `ipttable` will begin to warn and eventually error in later releases.

imcontour Second Output Argument Changed

The second output argument of `imcontour` is now a handle to an hggroup object instead of an array of handles to patch objects.

Compatibility Consideration

If you need to access handles of individual patch objects, use the following code to work around the change.

```
[c, handleToHGGroup] = imcontour(..);
arrayOfHandlesToPatchObjects = get(handleToHGGroup, 'Child');
```

impixelinfo Tool Disappears when Image Changes

If you use `imshow` to display an image, open the `impixelinfo` tool, and use `imshow` to open a new image, the `impixelinfo` tool will disappear along with the first image.

Compatibility Considerations

In previous versions, if you entered the following code:

```
imshow pout.tif
impixelinfo
imshow peppers.png
```

the `impixelinfo` tool would update to reflect changes to the image. Now you must call the `impixelinfo` tool again after opening the second image.

Some Code Moved into Different Directories

- Colorspace functionality moved into the new `toolbox/images/colorspaces` directory.
- Medical file formats moved into the `toolbox/images/iptformats` directory with other file formats, and the `toolbox/images/medformats` directory was removed.

Functions and Demos Being Removed

Function or Demo Name	What Happens When You Use Function or Demo?	Use This Instead	Compatibility Considerations
`pixval`	Errors	Use `impixelinfo` for pixel reporting and `imdistline` for measuring distance.	Replace all existing instances of `pixval` with `impixelinfo` or `imdistline`.
`dctdemo`	Errors	NA	None
`edgedemo`	Errors	NA	None
`firdemo`	Errors	NA	None
`landsatdemo`	Errors	NA	None
`nrfiltdemo`	Errors	NA	None

Function or Demo Name	What Happens When You Use Function or Demo?	Use This Instead	Compatibility Considerations
`qtdemo`	Errors	NA	None
`roidemo`	Errors	NA	None

Version 6.1 (R2008a) Image Processing Toolbox

This table summarizes what's new in Version 6.1 (R2008a).

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	Yes—Details labeled as **Compatibility Considerations**, below. See also Summary.	Bug Reports Includes fixes

The following sections describe new features and changes introduced in this version.

- "Create High Dynamic Range (HDR) Images and Write Them to Files" on page 33
- "Measure Properties of Regions in Grayscale Images" on page 34
- "Display Very Large Images by Subsampling" on page 34
- "Enhancements to ROI Tools" on page 34
- "Enhancements to Color Functions" on page 35
- "cp2tform Function Supports New Transformations" on page 36
- "hough Function Uses Specified RhoResolution Values" on page 36
- "Enhancements to Interactive Tools" on page 36
- "New and Updated Demos" on page 36
- "Enhancements to Other Functions" on page 37

Create High Dynamic Range (HDR) Images and Write Them to Files

Create a high dynamic range image from a group of low dynamic range images using the new `makehdr` function. The low dynamic range images must be spatially registered. You can write the HDR image to a file using the `hdrwrite` function. These functions complement the `hdrread` and `tonemap` functions introduced in R2007b.

Measure Properties of Regions in Grayscale Images

The `regionprops` function now accepts grayscale images as an input parameter, returning measurements based on the values of pixels in specified regions. Using `regionprops`, you can obtain measurements of regions in the image such as the maximum, minimum, and mean intensities in the region, and the weighted centroid.

Display Very Large Images by Subsampling

You can now display very large images from TIFF files by using the `imshow` function's new `'Reduce'` parameter. When you specify this parameter, `imshow` displays a subsampled version of the image. `imshow` determines the subsampling factor by considering the size of the image and the reduction required to fit the image on your screen. The `'Reduce'` parameter makes it possible to view very large images in their entirety that could not previously be displayed. Note, however, that the image subsampling that is performed reduces that amount of image data displayed.

Enhancements to ROI Tools

The toolbox includes several functions that enable the definition of regions of interest of various shapes: `impoint`, `imline`, `impoly`, `imrect`, and `imfreehand`. These ROI tools have several enhancements:

ROI Tools Reimplemented as MATLAB Classes

The ROI tools have been reimplemented as MATLAB classes. This change does not affect how the ROI tools function; they function identically to their previous implementation. The documentation uses the MATLAB functional syntax descriptions rather than the dot notation. That is, the documentation shows how to call the class methods specifying a handle to the object as the first argument, `method(h,...)`. Note, however, that you can still use the dot notation when calling the methods, `obj.method(...)`. In addition, the `iptgetapi` function now returns an object of the new class which means that code similar to the following will continue to work:

```
api = iptgetapi(h)
api.method()
```

Compatibility Consideration. The class of the data returned by the ROI tools is now a handle to an ROI class, such as `imline` or `impoly`. In addition, several undocumented methods supported by the ROI tools have been removed: `getContextMenu`, `setContextMenu`, `getDrawAPI`, `addCallback`, and `removeCallback`.

ROI Tools Support New wait and resume Methods

The ROI tools now support `wait` and `resume` methods so that they can be used in scripts. By using the `wait` method, you can enable users of your script to make the initial placement of the ROI, adjust the ROI and accept it, and then use the position in the script. For example, using the `wait` method with an ROI tool, you could write a script that creates a mask.

The `resume` method is a programmatic way to return control to the command line. When called after `wait`, `resume` causes `wait` to return the accepted position of the ROI.

Interactively Add New Vertices to ROI Polygons

You can now add vertices interactively to polygonal ROIs that you define using the `impoly` function. To create the new vertex, position the pointer over an edge of the polygon and press the A key. The pointer changes shape. Click the mouse to add a new vertex. The `roifill` and `roipoly` functions, which use `impoly` to implement ROIs, also support this new capability.

Enhancements to Color Functions

The following color functions have been enhanced.

makecform Supports Converting Between sRGB and CMYK

The `makecform` function now supports two new color space conversion types for converting between sRGB and CMYK: `'srgb2cmyk'` and `'cmyk2srgb'`.

iccwrite Creates Smaller ICC Profiles

The `iccwrite` function now uses certain optimizations to reduce the size of the International Color Consortium (ICC) color profiles that it creates. `iccwrite` uses aliasing to avoid writing tag data multiple times when it is included in more than one profile table.

cp2tform Function Supports New Transformations

The `cp2tform` function supports two new transformation types: `'similarity'` and `'nonreflective similarity'`.

Compatibility Consideration

The `'linear conformal'` transformation type supported by the `cp2tform` function has been renamed to `'nonreflective similarity'`.

hough Function Uses Specified RhoResolution Values

The `hough` function now uses the value you specify for the `'RhoResolution'` parameter. In previous releases, the function did not use the value specified.

Compatibility Consideration

The Hough matrix, `H`, and the `Rho` outputs returned by the `hough` function have different results than those obtained from the same function in previous releases.

Enhancements to Interactive Tools

The following modular interactive tools have been enhanced.

- Adjust Contrast tool (`imcontrast`) — The **Adjust Data** button in the Adjust Contrast tool is disabled until you make a change to image contrast.
- Pixel Region tool (`impixelregion`) — To improve the visibility of the image pixels being examined, the Pixel Region tool stops including grid lines in the display at low magnifications.

New and Updated Demos

The toolbox includes the following new and changed demos.

- *Batch Processing Image Files in Parallel* is an existing demo that has been updated, and simplified, through use of the `parfor` function.
- *Detecting Cars in a Video of Traffic* is a new demo that shows how to use the toolbox to visualize and analyze videos or image sequences.
- *Measuring Regions in Grayscale Images* is a new demo that shows how to use the `regionprops` function with grayscale images.

Enhancements to Other Functions

This release includes changes to the following functions.

Function	Description of Enhancement
`imageinfo`	Accepts files of several additional file formats as an input argument, including NITF, Interfile, and Analyze file formats.
`imshow`	Supports a new colormap parameter for specifying a colormap for grayscale images.
`imtool`	Supports a new colormap parameter for specifying a colormap for grayscale images.
`truesize`	Preserves the border preference setting of the figure when adjusting the image display size.

Version 6.0 (R2007b) Image Processing Toolbox

This table summarizes what's new in Version 6.0 (R2007b).

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	Yes—Details labeled as **Compatibility Considerations**, below. See also Summary.	Bug Reports Includes fixes

New features and changes introduced in this version are

- "New Interactive Image Sequence and Video Viewer" on page 39
- "Image Tool Includes Cropping, Enhanced Contrast Adjustment, and Saving of Modified Images" on page 39
- "New Function for Converting Bayer Pattern Encoded Images to RGB" on page 39
- "New Function for Creating a Multiresolution Gaussian Pyramid" on page 40
- "Enhanced ROI Definition Behavior for `imcrop`, `roifill`, and `roipoly`" on page 40
- "New Modular Interactive GUI-building Tools" on page 40
- "New Programmable ROI Tools" on page 40
- "Support for Reading NITF and HDR Images" on page 41
- "Enhanced Performance" on page 41
- "DICOM Dictionary Upgrade" on page 41
- "Changes to Other Functions" on page 42

New Interactive Image Sequence and Video Viewer

The toolbox now supports a new interactive image sequence viewer, called the Movie Player (`implay`). Using the Movie Player you can:

- Play a MATLAB movie, AVI file, or multidimensional array.
- Step through a movie or sequence of images, frame-by-frame, or jump to the beginning or end of the sequence.
- Examine a frame using the Pixel Region tool or export the frame to the Image Tool.

Image Tool Includes Cropping, Enhanced Contrast Adjustment, and Saving of Modified Images

The Image Tool (`imtool`) supports several enhancements:

- You can now modify the image data after performing a contrast adjustment operation. Previously, contrast adjustment only affected the display of the image, not the actual image data. To modify image data, click **Adjust Data** in the Adjust Contrast tool.
- You can now interactively crop an image displayed in the Image Tool using the Crop Image button in the toolbar (or select **Crop Image** from the Tools menu).
- You can now save the image displayed in the image tool in any of several common image file formats. Select **Save As** from the Image Tool File menu.

New Function for Converting Bayer Pattern Encoded Images to RGB

The toolbox now supports a function, `demosaic`, that can convert a Bayer pattern encoded image into an RGB image.

A Bayer filter mosaic, or color filter array, refers to the arrangement of color filters that let each sensor in a single-sensor digital camera record only red, green, or blue data. The patterns emphasize the number of green sensors to mimic the human eye's greater sensitivity to green light. The `demosaic` function uses interpolation to convert the two-dimensional Bayer-encoded image into a truecolor image, in the RGB color space.

New Function for Creating a Multiresolution Gaussian Pyramid

The toolbox now supports a function, `impyramid`, that you can use to create a multiresolution Gaussian pyramid. If you specify the `'reduce'` parameter, `impyramid` returns a low-pass filtered version of the image, half the size of the original image. If you specify the 'expand' parameter, `impyramid` returns a filtered image twice the size of the original image. `impyramid` uses convolution with a Gaussian filter kernel to produce the images.

Enhanced ROI Definition Behavior for `imcrop`, `roifill`, and `roipoly`

The `imcrop`, `roifill`, and `roipoly` functions now let you define an ROI and then adjust the size and position of the ROI interactively using the mouse. In previous releases, these functions supported the interactive definition of ROIs, but only gave you one chance at the definition. Now, when you are satisfied with the size and shape of the ROI, double-click to perform the cropping, filling, or mask creation operation.

Compatibility Consideration

In previous releases, when defining a polygonal ROI using `roipoly`, pressing **Backspace** deleted the most recent vertex you had defined in the polygon. With this release, pressing **Backspace** deletes the entire polygon. To delete an individual vertex, move the pointer over the vertex, right-click to view the vertex context menu, and then choose **Delete Vertex**.

New Modular Interactive GUI-building Tools

The set of modular interactive tools now includes functions to display a file chooser dialog box and write data to a file (`imsave`). The toolbox also includes a function (`imputfile`) that displays the file chooser dialog box and returns the user's selections.

New Programmable ROI Tools

The set of programmable ROI creation functions provided by the toolbox now includes three additional shapes:

- Polygons (`impoly`)
- Ellipses (`imellipse`)
- Freehand shapes (`imfreehand`)

The toolbox already includes ROI creation functions to create points, lines, and rectangles.

Each of the ROI creation functions supports an API that you can use to control aspects of its behavior and appearance. For example, you can use API functions to specify the position of the ROI or retrieve the coordinates of its current position.

Support for Reading NITF and HDR Images

- Read metadata from a National Imagery Transmission Format (NITF) file using `nitfinfo`.
- Read an image from a NITF file using `nitfread`.
- Read high dynamic range (HDR) images using `hdrread`.
- Convert high dynamic range images into a format that can be displayed using the `tonemap` function.

Enhanced Performance

- Enhanced performance for thinning and skeletonization using `bwmorph`.
- Enhanced performance for filtering RGB images using `imfilter`.

DICOM Dictionary Upgrade

The default DICOM dictionary has been upgraded to the 2007 version released by NEMA. A text version of this dictionary is included in the product, `dicom-dict.txt`. This upgrade fixes a problem with the earlier version of the dictionary which contained two instances of the same tag, which caused warnings.

Compatibility Consideration

If your DICOM code depends on hard-coded old attribute names, you may see failures. In addition, some DICOM files may no longer parse. Customers who require attribute settings from the 2005 version can use the `dicomdict` function to access the old data dictionary, which we are shipping in R2007b. That is, `dicom-dict.txt` will have 2007 values and `dicom-dict-2005.txt` is the version of `dicom-dict.txt` found in R2006a and R2007a.

Changes to Other Functions

This release includes changes to the following functions.

Function	Description of Change
`imshow`	Is not supported when MATLAB is started with the `-nojvm` option.
`imhist`	Can now be embedded in custom GUIs.
`fanbeam`,`ifanbeam`,`fan2para`,`para2fan`	The fan-beam functions now return different answers than in previous releases due to a bug fix.
`imadjdemo`	This demo has been deleted from the toolbox.

Version 5.4 (R2007a) Image Processing Toolbox

This table summarizes what's new in Version 5.4 (R2007a).

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	Yes—Details labeled as **Compatibility Considerations**, below. See also Summary.	Bug Reports Includes fixes

New features and changes introduced in this version are

- "Enhancements to imresize Function" on page 43
- "applycform Supports Tetrahedral Interpolation" on page 44
- "Control Point Selection Tool Enhancements" on page 44
- "Enhancements to impoint, imline, and imrect Functions" on page 44
- "Enhancements to montage Function" on page 45
- "Compatibility Considerations" on page 45
- "Changes to Other Functions" on page 45

Enhancements to imresize Function

`imresize` now runs faster, uses less memory, supports new interpolation methods, and supports new options for specifying output size.

Compatibility Consideration

The `imresize` function has been completely rewritten with new algorithms, new options, and new syntaxes. If you need the results produced by the version of `imresize` in previous releases, use the `imresize_old` function.

applycform Supports Tetrahedral Interpolation

The `applycform` function now uses tetrahedral interpolation for profiles containing multidimensional lookup tables, and returns more accurate results.

Compatibility Consideration

The results returned by `applycform` are more accurate but they are different than results returned in previous releases, for profiles containing multidimensional lookup tables.

Control Point Selection Tool Enhancements

The Control Point Selection Tool has enhanced visual appearance and usability. For example, points are now numbered for easier identification of matched pairs.

In addition, the tool now supports a `'wait'` option which enables `cpselect` to be used in scripts. When you specify this option, `cpselect` blocks the MATLAB command line until point selection is completed. For information about using the Control Point Selection Tool, see "Image Registration" and the reference page for the `cpselect` function.

Compatibility Consideration

The Control Point Selection Tool no longer includes the **Redo** or **Undo** options on the Edit menu.

Enhancements to impoint, imline, and imrect Functions

The `impoint`, `imline`, and `imrect` function now support an interactive placement capability. Using the mouse, you can specify the initial position of the point, line, or rectangle. In addition, the `imrect` function now supports interactive resizing using the mouse. See the reference pages for these functions for more information and examples.

Enhancements to montage Function

The `montage` function now supports parameters that control the arrangement and appearance of the images displayed. See the `montage` reference page for these functions for more information and examples

Compatibility Considerations

makecform Uses 'icc' Whitepoint for L*a*b*/sRGB Conversions

The `makecform` function now only uses the white point type `'icc'` for color space conversions from *L*a*b** to *srgb* (type = `'lab2srgb'`) and from *srgb* to *L*a*b** (type = `'srgb2lab'`). In previous releases, you could specify other white point values for these conversions, using the optional `'Whitepoint'` parameter. This syntax now issues a warning when any other white point besides `'icc'` is specified.

normxcorr2 Might Return Different Results

The `normxcorr2` function now returns values in the range [-1,1] for all inputs. In previous releases, `normxcorr2` returned values outside this range for certain inputs.

watershed Function Uses New Algorithm

The watershed transform algorithm used by the `watershed` function has changed. The previous algorithm occasionally produced labeled watershed basins that were not contiguous. If you need to obtain the same results as the previous algorithm, use the function `watershed_old`.

Changes to Other Functions

This release includes changes to the following functions.

Function	Description of Change
`imshow`	New `'border'` parameter, to control whether imshow includes a border around the image displayed, and `'parent'` parameter, to specify the axes in which to display the image.
`imscrollpanel`	New `'replaceImage'` parameter lets you replace the image displayed in the scroll panel with a new image.
`iradon`	New `'none'` value for the `filter` parameter returns an unfiltered backprojection; also supports new interpolation types.
`iptsetpref`	New `UseIPPL` preference.

Version 5.3 (R2006b) Image Processing Toolbox

This table summarizes what's new in Version 5.3 (R2006b).

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	Yes—Details labeled as **Compatibility Considerations**, below. See also Summary.	Bug Reports Includes fixes

New features and changes introduced in this version are

- "Enhancements to DICOM Capabilities" on page 47
- "New Symmetric Option with graycomatrix Function" on page 47
- "Enhancements to ICC Color Capabilities" on page 48
- "Enhancements to the imdistline Function" on page 48
- "setColor Method Accepts Predefined Color Strings" on page 49

Enhancements to DICOM Capabilities

This release includes the following new features and enhancements to the DICOM capabilities of the Image Processing Toolbox:

- The toolbox includes a new function, `dicomlookup`, that provides a way to find the name of an attribute in a DICOM data dictionary by specifying its group and element tags, or find the group and element tags for an attribute by specifying its name.
- The performance of the `dicominfo` function has been significantly improved

New Symmetric Option with graycomatrix Function

The `graycomatrix` function now supports a new option: `'symmetric'`. With this option, you can create a gray-level co-occurrence matrix (GLCM) that is symmetric about its diagonal. This is consistent with the GLCM definition given by Haralick in his 1973 article. For more information, see `graycomatrix`.

Enhancements to ICC Color Capabilities

The toolbox includes the following enchancements to the ICC color capabilities:

- The `applycform` function can now transform colors using profiles that contain parametric curve types.
- The `iccread` function now supports named colors in ICC profiles.

Compatibility Considerations

The `whitepoint` function, when used with the `'d50'` argument, returns different results in R2006b than it did in R2006a. The previously returned *XYZ* color values were incorrect according to the current interpretation of standards. If your algorithm depended on the old values, you might see subtly different results.

Enhancements to the imdistline Function

This release includes the following enhancements to the `imdistline` function:

- The `imdistline` function now uses a different cursor shape at its endpoints to highlight that these endpoints can be grabbed to change the length or direction of the line. The function uses a hand cursor over endpoints and a fleur cursor over the body of the line.
- The `imdistline` function reference page now includes an example that shows how to use the `XData` and `YData` properties of the associated image to express distance in non-pixel units.

Compatibility Considerations

The Distance Tool's `getAngleFromHorizontal` method now returns a value between 0 and 180 degrees. Previously, this function incorrectly returned a value between 0 and 90. For an explanation of how `getAngleFromHorizontal` calculates this angle, see the `imdistline` function.

setColor Method Accepts Predefined Color Strings

The `setColor` method of the `imdistline`, `imline`, `impoint`, and `imrect` functions accepts an RGB triplet or the short- or long-name version of the MATLAB predefined color names.

Version 5.2 (R2006a) Image Processing Toolbox

This table summarizes what’s new in V5.2 (R2006a).

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	Yes—Details labeled as **Compatibility Considerations**, below. See also Summary.	Bug Reports Includes fixes

New features and changes introduced in this version are

- “Enhanced ICC Profile Capabilities” on page 50
- “New Pointer Management Functions” on page 51
- “New Constraint Creation Function” on page 51
- “Compability Considerations” on page 51
- “IPPL Not Used on 64-Bit Systems” on page 51

Enhanced ICC Profile Capabilities

The `iccread` and `iccwrite` functions have been updated to support recent changes to the ICC specification.

In addition, `iccread` can now read and process the following additional profile types:

- DeviceLink profiles — Provide transformation from one device space to another.
- ColorSpace profiles — Provide transformation between a non-device color space and the profile connection space (PCS).
- Abstract profiles — Enable color transformations to be defined that provide specific color effects.

- Grayscale profiles — Specify the relationship between device values and the PCS for specific colors.

In addition, `iccread` can now read parametric curve types.

New Pointer Management Functions

The toolbox includes three new utility functions, `iptPointerManager`, `iptGetPointerBehavior`, and `iptSetPointerBehavior`, that you can use to manage changes to the pointer in GUIs. For example, you can use the pointer management functions to change the appearance of the pointer when it moves over objects in a figure. These functions can be useful when building GUIs with the toolbox modular GUI tools.

New Constraint Creation Function

The toolbox includes a new utility function, `makeConstrainToRectFcn`, that you can use to specify drag constraints for the `imdistline`, `imline`, `impoint`, and `imrect` functions. You specify the constraints as arguments to the `makeConstrainToRectFcn` and this function returns a handle to a constraint function. To use this constraint with an object, set the value of the `setConstraintFcn` API for the object to this function handle.

Compability Considerations

When using the `cp2tform`, `tforminv`, or `imtransform` functions with the transform type `'piecewise linear'` you might get different answers from previous versions due to a bug fix. If you have a transformation structure (`TFORM`) saved from an older version, you may want to regenerate it from control points to get improved performance.

IPPL Not Used on 64-Bit Systems

Certain functions in the Image Processing Toolbox, such as the image arithmetic functions, use the Intel Performance Primitives Library (IPPL), if it's available. (See `ippl` for more information.) Note that these functions do not use the IPPL on 64-bit systems.

Version 5.1 (R14SP3) Image Processing Toolbox

This table summarizes what's new in Version 5.1 (R14SP3).

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	Yes—Details labeled as **Compatibility Considerations**, below. See also Summary.	Bug Reports Includes fixes

New features and changes introduced in this version are

- "Support for Two New Medical Image File Formats" on page 52
- "New Point, Rectangle, and Line Functions" on page 52
- "Image Tool Enhancements and Improvements" on page 53
- "New Utility Functions for Use with Profile-Based Color Space Conversion Functions" on page 53
- "New Documentation on Processing Image Sequences" on page 53
- "Control Point Selection Tool Now Works on Macintosh Systems" on page 53
- "Compatibility Considerations" on page 54

Support for Two New Medical Image File Formats

The toolbox now includes functions for reading metadata and image data from two additional medical image file formats. Analyze 7.5 and Interfile. For more information, see Reading and Writing Data in Medical Formats.

New Point, Rectangle, and Line Functions

The toolbox includes three functions, `impoint`, `imline`, and `imrect`, that you can use to create draggable points, lines, and rectangles in a figure window. These functions can be used as building blocks for other GUI tools.

Image Tool Enhancements and Improvements

New Distance Tool

The Image Tool now includes a new Distance tool that you can use to determine the distance between any two points in an image. This tool is also available in the toolbox's suite of modular interactive GUI tools. Using the `imdistline` function you can add the Distance tool to GUIs of your own creation. For more information, see Measuring Features in an Image

Adjust Contrast Tool Enhancements and Improvements

The Adjust Contrast tool has been redesigned to provide better usability. For examples, the Adjust Contrast tool Window/Level capability is now a separate mode with its own activation button.

New Utility Functions for Use with Profile-Based Color Space Conversion Functions

The toolbox has two new utility functions, `iccroot` and `iccfind`, for use with the profile-based color conversion functions. For more information, see Performing Profile-based Color Space Conversions.

New Documentation on Processing Image Sequences

The Image Processing Toolbox User's Guide includes a new section, Working with Image Sequences, that describes which toolbox functions can be used with sequences of image, also known as image stacks

Control Point Selection Tool Now Works on Macintosh Systems

The Control Point Selection Tool now works on Macintosh® systems.

Compatibility Considerations

Obsolete and Deleted Functions

The following table lists toolbox functions that have been made obsolete or removed in this version.

Function	Enhancement
`impositionrect`	This function is obsolete. Use `imrect` to perform the same tasks.
`pixval`	This function is obsolete. It now issues a warning when used. Use `impixelinfo` for pixel reporting and use `imdistline` for measuring distance

Image Tool Is Not Compilable

The `imtool` function is not compilable with the MATLAB Compiler.

Version 5.0.2 (R14SP2) Image Processing Toolbox

This table summarizes what's new in Version 5.0.2 (R14SP2).

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
No	Yes—Details labeled as **Compatibility Considerations**, below. See also Summary.	Bug fixes Details

New features and changes introduced in this version are

- "Major Bug Fixes" on page 55
- "Compatibility Considerations" on page 59

Major Bug Fixes

This release contains the following bug fixes.

Major Revisions to Fan-Beam Functions

This release includes numerous updates and improvements to the fan-beam functions: `fanbeam`, `ifanbeam`, `fan2para`, and `para2fan`. The fixes include improved calculations, improved documentation, and examples.

For example, `fanbeam` now returns the correct sensor locations when the geometry is `'line'`. The `ifanbeam` and `fan2para` now consistently use the correct default value for the `'FanSensorSpacing'` parameter. If you tried the fan-beam functions in a previous release, you might try them again to take advantage of these improvements.

In addition to the functional changes, many improvements to the documentation of the fan-beam functions have been made.

`fanbeam` help now includes

- An example that shows how to extract projection data at a specific rotation angle from the fan-beam data returned
- An explanation of how `fanbeam` calculates the number of rows and columns in `F`, the fan-beam data returned
- The default value for the `'FanSensorSpacing'` parameter for both `'line'` and `'arc'` geometries
- Guidelines for setting the value of the `D` parameter

The help for the `ifanbeam` function now includes an example that shows how to use the `'minimal'` coverage parameter.

Compatilbility Considerations. Results computed with earlier versions of the fan-beam functions cannot be used with the new versions of these functions.

Changes to the DICOM Functions

The following fixes have been made to the `dicomread` and `dicomwrite` functions.

Function	Bug Fixes
`dicomread`	No longer errors when reading files that contain extraneous pixel data; instead, `dicomread` issues a warning message. However, if the file does not contain enough pixel data, `dicomread` issues an error.
`dicomwrite`	• No longer is case sensitive when parsing input parameters. For example, you can specify either `'CreateMode'` or `'createmode'`. • Preserves the full precision of data converted to decimal string metadata. Previously, `dicomwrite` limited precision to six digits. • No longer errors when writing files with metadata values that must be stored as a decimal string or integer string. Now, when writing private data attributes (attributes that are not listed in the DICOM data dictionary), `dicomwrite` assigns the attributes the type `UN` (for unknown) and writes the data to the file as a byte-for-byte copy of its in-memory representation. Because `dicomwrite` writes the file with explicit value representation (VR), the file might have a different VR value, but the data will be the same.

Function	Bug Fixes
	• Includes the `TriggerTime` field for additional values of `ScanOptions`, including `'CT'`. Previously, `dicomwrite` only included the `TriggerTime` attribute if the `ScanOptions` field indicated a gated heart MR. • No longer issues an `Unsupported SOP class` error message if, when `'create'` mode is specified, semantic verification is not available for an information object. Instead, `dicomwrite` issues a more helpful message indicating that it might be able to write the data if the mode was `'copy`', rather than `'create'`. In `'copy'` mode, `dicomwrite` only performs syntactic checking, not semantic verification. Consequently, `dicomwrite` can write many more types of DICOM files in `'copy'` mode than it can in `'create'` mode. See the `dicomwrite` reference page for important information about data integrity.

Changes to Image Tool and Modular Interactive Tools

The following fixes have been made to the Image Tool and other modular interactive tools:

- The Image Tool now always makes the **Open** and **Import from Workspace** options available on its **File** menu. Previously, the Image Tool disabled these options if the tool contained an image. If the Image Tool contains an image, the newly imported image is displayed in a new Image Tool using the default preferences.
- The Image Tool zoom buttons can now be used on an image that has superimposed vector data.
- The Image Tool toolbar buttons no longer create multiple versions of the modular interactive tools when clicked rapidly in quick succession.
- The Image Information tool now displays correctly on Linux systems. Previously, it displayed as a blank window.
- The Overview tool can now be resized from any corner. Previously, resizing the tool using a corner other than the lower left caused the image to become progressively smaller until it disappeared.

- The Overview tool zoom buttons now provide an affordance that informs users when they cannot use these buttons to zoom in or out on the image displayed in the associated scroll panel.
- The Pixel Region tool now displays floating-point values correctly. Previously, the pixel value text strings displayed spilled over into adjacent pixels for some floating-point images.
- The Pixel Region tool now works correctly with images displayed in subplots.
- The Pixel Region tool no longer causes the target image to become tiny and move to a different position in the figure.

Changes to the imshow Function

- The `imshow` function no longer overwrites nondefault axes in a figure.
- The `imshow` function ignores any initial magnification value you specify when used to display an image in a figure that is docked (the figure's `WindowStyle` property is set to `'docked'`). In these cases, `imshow` displays the image at the largest magnification that fits the window (`'fit`' magnification) and issues a warning.

Fixes to Other Functions

The following tables lists fixes that have been done to other toolbox functions.

Function	Enhancement
`applycform`	Now correctly handles profiles that contain a `gamut` tag.
`cpcorr`	Now is more numerically robust. For this release, the subfunction `findpeak`, which `cpcorr` calls, has been improved and is now a private function, rather than a subfunction.
`imhist`	No longer causes a docked figure window to become undocked.

Function	Enhancement
`imrotate`	Now correctly rotates N-dimensional arrays, where N is greater than 3. In previous releases, `imrotate` would accept N-D arrays but only return a 3-D array.
`normxcorr2`	Now always returns real values. In previous releases, due to roundoff error, some sets of input data caused the `normxcorr2` function to return a complex valued matrix of correlation coefficients.
`pixval`	Now works correctly with binary images.
`rgb2ind`	Now returns a correct output image when called with the syntax `rgb2ind(rgb,n,'nodither')` where `n` is greater than 256.

Fixes to Image Processing Toolbox Deployment Issues

- Performance issues that occurred when deploying compiled image processing toolbox functions that call IPPL routines have been fixed.
- Running compiled versions of `imtool` and some of the other modular interactive tools no longer generates the following warning messages about classes not being cleared:

```
Warning: Objects of graphics.linkprop class exist - not clearing
this class or any of its super-classes.
Warning: An object instance still exists. Use the objectdirectory
command to see a count of existing instances.
```

Compatibility Considerations

The Image Processing Toolbox software now requires the following new directory on the MATLAB path:

```
toolbox\shared\imageslib
```

Version 5.0.1 (R14SP1) Image Processing Toolbox

This table summarizes what's new in Version 5.0.1 (R14SP1).

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
No	No	Bug fixes Details

Major Bug Fixes

The following are major bug fixes in this release.

DCT Image Compression Demo Now Calculates Mean Squared Error Correctly

The 2-D DCT Image Compression demo (`dctdemo`) previously calculated the mean squared error incorrectly.

Choose Colormap Tool Now Works with Compiled Applications

The Choose Colormap tool now saves the colormap selection when running in a compiled version of the Image Tool.

Image in Pixel Region Tool Now Updates When Using the Adjust Contrast Tool

Previously, if you used `imtool` to display a grayscale image, launched the Pixel Region tool, and then used the Adjust Contrast tool to adjust the contrast of the image, the pixel colors in the Pixel Region tool did not update properly. This has been fixed.

Version 5.0 (R14) Image Processing Toolbox

This table summarizes what's new in Version 5.0 (R14SP2):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	Yes—Details labeled as **Compatibility Considerations**, below. See also Summary.	Bug Fixes Details

New features and changes introduced in this version are:

- "New Image Exploration and Enhancement Tool" on page 62
- "New Modular Interactive Tools" on page 62
- "New Modular Tool Utility Functions" on page 65
- "Hough Transform" on page 66
- "Texture Analysis" on page 67
- "New ICC Color Profile Export Function" on page 68
- "New Demos" on page 69
- "New DICOM Anonymizer Function" on page 69
- "New Integer Lookup Table Function" on page 69
- "New Toolbox Utility Functions" on page 69
- "Updates to the imshow Function" on page 70
- "Changes to Toolbox Preferences" on page 74
- "Changes to Other Toolbox Functions" on page 75
- "Performance Improvements" on page 77
- "Improved Memory Usage" on page 77
- "Major Bug Fixes" on page 78

- “Compatibility Considerations” on page 79
- “General Issues” on page 80
- “Issues Specific to the Linux Platform” on page 83
- “Issues Specific to the Macintosh Platform” on page 83

New Image Exploration and Enhancement Tool

The Image Processing Toolbox includes a new, open-architecture, image exploration and enhancement tool, called the Image Tool.

The Image Tool replaces the Image Viewer, providing all the display and exploration capabilities of its predecessor. For example, you can use the Image Tool to display an image, view general information about the image, get information about individual pixels or regions of pixels in the image, and navigate large images using the Overview navigation window, scroll bars, and magnification tools.

In addition, the Image Tool introduces several new tools, such as the Adjust Contrast tool and the Choose Colormap tool. You can use the Adjust Contrast tool to adjust the brightness and contrast of an image interactively. You can use the Choose Colormap tool to change the colormap for indexed and intensity images to any of MATLAB colormaps or to a user-defined colormap.

Unlike the Image Viewer, the Image Tool is built using standard features of MATLAB Handle Graphics®. This enables the Image Tool to provide access to the image being displayed using standard Handle Graphics techniques. For example, you can use annotations and overlay vector graphics on images displayed in the Image Tool. You can use `imtool` as an example to follow or as a base to use to create your own application.

To start the Image Tool, use the `imtool` function:

```
imtool('moon.tif')
```

New Modular Interactive Tools

The toolbox includes several new modular interactive tools that you can activate from the command line and use with images displayed in a MATLAB figure window, called the *target image* in this documentation. The tools are modular because they can be used independently or in combination to create custom graphical interfaces for image processing applications. The Image Tool uses these modular tools.

The following table lists these modular tools along with the functions you use to create them.

Modular Tool	Description	Function
Adjust Contrast tool	Display histogram of image pixel values in the target image and enable interactive adjustment of contrast and brightness by manipulating the range.	Use `imcontrast` to create the tool in a separate figure window and associate it with an image.
Display Range tool	Display a text string identifying the display range values of the target image.	Use `imdisplayrange` to create the tool, associate it with an image, and embed it in a figure or uipanel.
Image Information tool	Display basic information about an image, along with any metadata the image might contain.	Use `imageinfo` to create the tool. To collect image information, use `imattributes`, `imfinfo`, and `dicominfo`.

Modular Tool	Description	Function
Magnification box	Create a text edit box containing the current magnification of the target image. Users can type in the desired magnification. **Note** The target image must be contained in a scroll panel.	Use `immagbox` to create the tool, associate it with an image, and embed it in a figure or uipanel.
Overview tool	Display the target image in its entirety with the portion currently visible in the scroll panel outlined by a rectangle superimposed on the image. Moving the rectangle changes the portion of the target image that is currently visible in the scroll panel. **Note** The target image must be contained in a scroll panel.	Use `imoverview` to create the tool in a separate figure window and associate it with an image. Use `imoverviewpanel` to create the tool in a uipanel that can be embedded within another figure uipanel.

Modular Tool	Description	Function
Pixel Information tool	Display information about the pixel the mouse is over in the target image.	Use `impixelinfo` to create the tool, associate it with an image, and display it in a figure or uipanel. If you want to display only the pixel values, without the text label, use `impixelinfoval`.
Pixel Region tool	Display pixel values for a specified region in the target image.	Use `impixelregion` to create the tool in a separate figure window and associate it with an image. Use `impixelregionpanel` to create the tool as a uipanel that can be embedded within another figure or uipanel.
Scroll Panel	Display target image in a uipanel with scroll bars.	Use `imscrollpanel` to add a scroll panel to an image displayed in a figure window.

New Modular Tool Utility Functions

In addition to the modular tools listed above, the toolbox includes a number of new utility functions that make GUI building easier. The following table lists these utility functions in alphabetical order. The tools reside in the `$MATLAB/toolbox/images/imuitools` directory, where `$MATLAB` represents your MATLAB installation directory.

Function	Description
`getimagemodel`	Retrieve `imagemodel` objects from image handles
`imattributes`	Return information about image attributes
`imgca`	Get handle to current image axes
`imgcf`	Get handle to current image figure
`imgetfile`	Image Open File dialog box
`imhandles`	Get all image handles
`impositionrect`	Create position rectangle
`iptaddcallback`	Add function handle to callback list
`iptcheckhandle`	Check validity of handle
`iptgetapi`	Get Application Programmer Interface from a handle
`ipticondir`	Directories containing IPT and MATLAB icons
`iptremovecallback`	Delete function handle from callback list
`iptwindowalign`	Align figure windows

Hough Transform

The toolbox now includes three new functions that provide support for the Hough transform.

- `hough`
- `houghpeaks`
- `houghlines`

The `hough` function implements the Standard Hough Transform (SHT). The Hough transform is designed to detect lines, using the parametric representation of a line:

```
rho = x*cos(theta) + y*sin(theta).
```

The variable `rho` is the distance from the origin to the line along a vector perpendicular to the line. `theta` is the angle between the *x*-axis and this vector. The `hough` function generates a parameter space matrix whose rows and columns correspond to these `rho` and `theta` values, respectively.

The `houghpeaks` functions finds peak values in this space, which represent potential lines in the input image.

The `houghlines` function finds the endpoints of the line segments corresponding to peaks in the Hough transform, and it automatically fills in small gaps.

Texture Analysis

The toolbox now supports a set of functions that you can use for texture analysis. These functions include

- `entropy` — Calculates the entropy of an intensity image
- `entropyfilt` — Calculates the local entropy of an intensity image
- `graycomatrix` — Computes the gray-level co-occurrence matrix from an image
- `graycoprops` — Extracts properties from a gray-level co-occurrence matrix
- `rangefilt` — Calculates the local range of an image
- `stdfilt` — Calculates the standard deviation of an image

Texture analysis refers to the characterization of regions in an image by their texture content. Texture analysis attempts to quantify intuitive qualities described by terms such as rough, silky, or bumpy in the context of an image. In this case, the roughness or bumpiness refers to variations in the brightness values or gray levels.

Some of the most commonly used texture measures are derived from the Grey Level Co-occurrence Matrix (GLCM). The GLCM is a tabulation of how often different combinations of pixel brightness values (gray levels) occur in a pixel pair in an image. You can use the `graycomatrix` function to create a GLCM and then use `graycoprops` to extract feature information (e.g., contrast, correlation, energy, and homogeneity) from the GLCM.

The texture analysis support also includes several new functions that filter using standard statistical measures, such as range, standard deviation, and entropy. (Entropy is a statistical measure of randomness.) To see an example of using these filtering functions, view the "Texture Segmentation Using Texture Filters" demo. Use the `iptdemos` function to access toolbox demos.

New ICC Color Profile Export Function

The toolbox includes a new function, `iccwrite`, that you can use to write International Color Consortium (ICC) color profile data to a file. ICC profiles provide color management systems with the information necessary to convert color data between native device color spaces and device-independent color spaces, called Profile Connection Space (PCS).

The toolbox also includes a function, `isicc`, that can verify whether the data is a valid ICC profile. The `iccwrite` function can output profile data in accordance with both Version 2 (ICC.1:2001-04) and Version 4 (ICC.1:2001-12) of the ICC specification. For more information about the changes between Version 2 and Version 4 of the specification, go to the ICC Web site, `www.color.org`.

In addition, `iccread`, `makecform`, and `applycform` all now work with Version 4 profiles as well as Version 2 profiles, and with color profiles with more than four channels.

Changes to iccread

To add color profile export support, and to accommodate Version 4 of the specification, there are some differences in the way data is returned by `iccread`. In the structure returned by `iccread` some of the fields that contained text strings in previous releases are now structures. Accessing the text string in the fields requires an additional level of dereferencing. For example, the value of the Description field was a text string.

```
P.Description

ans =

sRGB IEC61966-2.1 991203
```

Now, the value of this field is a structure with two fields: `String` and `Optional`. To access the text string, you must access the `String` field in this structure.

```
P.Description.String

ans =

sRGB IEC61966-2.1 991203
```

New Demos

The toolbox includes two new demos:

- Texture Segmentation Using the Text Filters
- Analyzing a Multispectral LANDSAT Image

In addition, some of the existing demos have been reorganized into new categories. The Color Segmentation and Morphological Segmentation demos have been moved to the new Image Segmentation category.

New DICOM Anonymizer Function

The toolbox now includes a new function, `dicomanon`, that can remove all confidential data from a DICOM file.

New Integer Lookup Table Function

The toolbox now includes a new function, `intlut`, that can convert arrays of `uint8`, `uint16`, and `int16` integer values using a lookup table.

New Toolbox Utility Functions

The toolbox includes several new utility functions that can help with input argument parsing. These functions check the validity of arguments and issue standard error messages, if the argument is invalid. The toolbox includes other utility functions to get the dynamic range of an image and convert a positive integer to an ordinal string.

The following table lists these functions in alphabetical order. The functions reside in the `$MATLAB/toolbox/images/iptutils` directory, where `$MATLAB` represents your MATLAB installation directory.

Function	Description
`getrangefromclass`	Check dynamic range of image
`iptcheckconn`	Check validity of connectivity argument
`iptcheckinput`	Check validity of input arguments
`iptcheckmap`	Check validity of colormap argument
`iptchecknargin`	Check number of arguments
`iptcheckstrs`	Check validity of string arguments
`iptnum2ordinal`	Convert positive integer to ordinal string

Updates to the imshow Function

There have been several changes to the behavior and syntaxes supported by the `imshow` function. The following sections describe these changes.

- "Filenames No Longer Displayed as a Title in Figure Window Border" on page 70
- "Nondefault Spatial Coordinate Syntax Changed" on page 71
- "Initial Image Magnification DISPLAY_OPTION Syntax Changed" on page 71
- "New Image Scaling Algorithm Determines Image Display" on page 72
- "New Image Display Range Syntax" on page 73
- "Number-of-Gray-Levels Syntax Obsoleted" on page 74

Filenames No Longer Displayed as a Title in Figure Window Border

The `imshow` function when called with a filename

```
imshow(filename)
```

no longer automatically displays the filename as a title in the figure window border. In previous releases, the gray space was proportional to image size. Often, this left too little space for small images and too much for big images, making positioning the title in the figure window problematic.

Changes to the algorithm `imshow` uses to calculate the border size when the `'ImshowBorder'` preference is set to `'loose'` should allow plenty of room for title, axes tick marks, and axes labels in a way that is independent of the image size. (See "New Image Display Range Syntax" on page 73 for more information.)

Nondefault Spatial Coordinate Syntax Changed

The `imshow` syntax for specifying nondefault spatial coordinates has changed to use parameter/value pairs. The old syntax

```
imshow(x,y,...)
```

is now

```
imshow(...,'XData',x,'YData',y)
```

`imshow` still accepts the old syntax, automatically translating it to the new syntax and issuing the following warning.

```
IMSHOW(x,y,...) is an obsolete syntax. Use
IMSHOW(...,'XData',x,'YData',y) instead.
```

Initial Image Magnification DISPLAY_OPTION Syntax Changed

The `imshow` `display_option` syntax is obsolete. The old syntax,

```
imshow(...,display_option)
```

where `display_option` was either `'truesize'` or `'notruesize'`, has been replaced by the following parameter/value pair syntax:

```
imshow(...,'InitialMagnification',initial_mag)
```

As the value, you can specify a numeric magnification percentage value or the text string `'fit'`.

`imshow` still accepts the old syntax, automatically translating the old `display_option` values to the new syntax, as shown in the following table. The table also includes the text of the warning message issued by `imshow`.

Old Syntax	Automatic Translation to New Syntax

`imshow(...,'truesize')`	`imshow(...,'InitialMagnification',100)`
	where 100% magnification specifies the same one-image-pixel to one-screen-pixel magnification achieved by the `'truesize'` option. `imshow` also issues the following warning message:
	`Warning: IMSHOW(...,'truesize') is an obsolete syntax.`
	`Use IMSHOW(...,'InitialMagnification',100) instead.`

`imshow(...,'notruesize')`	`imshow(...,'InitialMagnification','fit')`
	where the behavior is similar to the `'notruesize'` behavior. `imshow` issues the following warning message:
	`Warning: IMSHOW(...,'notruesize') is an obsolete syntax.`
	`Use IMSHOW(...,'InitialMagnification','fit') instead.`

New Image Scaling Algorithm Determines Image Display

When you specify a numeric magnification percentage value, `imshow` performs the following processing to determine how to display the image:

1. Calculate the gutter dimensions (gray space around image) and figure window decoration dimensions.

Note `imshow` only has to calculate these dimensions once per MATLAB session because they are independent of the image being displayed and depend on the system running the display function.

---

2. Determine the screen dimensions for the image at the requested display magnification.
3. Add the results of step 1 and step 2.
4. Determine if the figure fits on the screen at the specified magnification.

   If the image in the figure window fits on the screen, display it.

   If the image in the figure window does not fit on the screen, `imshow` reduces the magnification until the image fits on the screen and displays it. `imshow` issues a warning message that the image has been scaled, including the magnification value in the message.

Note `imshow` now uses the same magnification increments as the zoom tool (100%, 67%, 50%, 33%,...).

New Image Display Range Syntax

The `imshow` syntax in which you specify the display range of the image

```
imshow(I,[LOW HIGH])
```

has been augmented to use a parameter/value pair syntax

```
imshow(...,'DisplayRange',[LOW HIGH])
```

`imshow` still accepts the old syntax.

Note, however, that with the new parameter/value syntax, you can specify the target image as a filename, as in the following example.

```
imshow(filename,'DisplayRange'[LOW HIGH])
```

If you want to specify the display range for an intensity image specified by a filename, you must use the `'DisplayRange'` parameter.

Number-of-Gray-Levels Syntax Obsoleted

The `imshow` syntax in which you specified the number of gray levels used to display the image

```
imshow(I,N)
```

has been obsoleted.

Compatibility Considerations. If you use this syntax, `imshow` outputs the following warning message:

```
IMSHOW(I,N) is an obsolete syntax. Your intensity image will be
displayed using 256 shades of gray.
```

Changes to Toolbox Preferences

The names of several Image Processing Toolbox preferences have changed in Version 5. The following table lists these preferences with the new name, where available.

Obsolete Preference	New Preference
`'ImshowTruesize'`	`'ImshowInitialMagnification'`
`'ImviewInitialMagnification'`	`'ImtoolInitialMagnification'`

`'TruesizeWarning'`	No replacement.
	Use the MATLAB `warning` function to control whether you see the warning that appears when displaying a large image with `imshow`.
	`warning off Images:initSize:adjustingMag`
	`warning on Images:initSize:adjustingMag`

Changes to Other Toolbox Functions

The following table lists toolbox functions that have been changed in Version 5 of the Image Processing Toolbox.

Function	Enhancement
`blkproc`	The `fun` argument must be specified as a function handle; it can no longer be specified as an `inline` function or text string. The syntax `blkprc(...,fun,P1,P2...)` is no longer supported. Use an anonymous function instead.
`colfilt`	See the entry for `blkprc` in this table. It describes the change made to this function.
`deconvblind`	See the entry for `blkprc` in this table. It describes the change made to this function.
`dicomread`	No longer supports the `'Dictionary'` and `'Raw'` parameters.
`edge`	Supports new syntaxes that return the gradient components when gradient-based methods are used (Sobel, Prewitt, Roberts).
`getimage`	No longer accepts a handle to a texture-mapped surface as an input argument or returns a texture-mapped surface as an image. `getimage` now only returns data for image objects. `getimage` also returns a new flag identifying a binary image.
`graythresh`	Now implements Otsu's class separability metric, which measures the effectiveness of a threshold computation. For this metric, the lower bound of 0 represents a monotone image and the upper bound of 1 represents a two-valued image.
`ifanbeam`	See the entry for `iradon` in this table.

Function	Enhancement
`im2bw`	Might produce different results when used in conjunction with the `graythresh` function. The `graythresh` function uses Otsu's method which, by definition, splits the pixels in an image into two classes based on the calculated threshold. All the pixels up to and including the pixels equal to the threshold belong to the first class and the remaining pixels belong to the other class. To match this algorithm, the `im2bw` function now uses the greater-than operator (>) instead of the greater-than-or-equal operator (>=). Thus `im2bw` might produce different results from previous releases.
`imfilter`	Now automatically detects and exploits filter separability to speed up the filter computation. This change in the computational algorithm can result in small differences in the output values because of a combination of floating-point roundoff differences and integer rounding effects.
`iradon`	Now supports all interpolation types supported by `interp1`. Previously, only `'linear'`, `'nearest'`, and `'spline'` were supported.
`makelut`	See the entry for `blkprc` in this table. It describes the change made to this function.
`nlfilter`	See the entry for `blkprc` in this table. It describes the change made to this function.
`qtdecomp`	See the entry for `blkprc` in this table. It describes the change made to this function.
`regionprops`	Now calculates the perimeter of each labeled region in a label matrix.
`roifilt2`	The `fun` argument must be specified as a function handle; it can no longer be specified as an `inline` function or text string. The syntax `roifilt(...,fun,P1,P2...)` is no longer supported. Use anonymous function instead.

Performance Improvements

The performance of several existing toolbox functions has been improved in this release, including:

- `regionprops` (six times faster than previous version)
- `imfilter` (faster for separable filters)

The `dicomread` function has a five to 10 times performance improvement over the previous version.

The performance of the following morphology functions has been improved when used with large rectangular structuring elements.

- `imdilate`
- `imerode`

Functions that call `imdilate` and `imerode`, specifying large rectangular structuring elements, might also see a speed improvement, i.e., `imopen`, `imclose`, `imtophat`, and `imbothat`.

The performance of the following image type conversion functions and color space conversion functions has been improved by using the `intlut` function with data of classes `uint8` and `uint16`.

- `imadjust`
- `imcomplement`
- `ind2gray`
- `rgb2gray`
- `rgb2ntsc`
- `rgb2ycbcr`
- `rcbcr2rgb`

Improved Memory Usage

The memory usage of the following deblurring functions has been improved by clearing temporary variables as the algorithms proceed.

- `deconvblind`
- `deconvlucy`
- `deconvreg`
- `deconvwnr`

edgetaper

The `edgetaper` function now uses single precision in its calculations for images with an integer data type in order to reduce memory usage. This means that `edgetaper` returns an answer that is slightly different from the answer returned by previous versions of the toolbox. If you want `edgetaper` to use double precision for integer data types, convert your image to `double` before calling `edgetaper`.

improfile

The `improfile` function used to cast input images to `double` if the image was of class logical or if the image was of a nondouble class that uses an interpolation method other than nearest-neighbor.

The function now casts them to single to reduce memory overhead. In these cases, the output of `improfile` might differ slightly from previous versions of the toolbox.

Major Bug Fixes

The following are important bug fixes in Version 5 of the Image Processing Toolbox software.

Canny Edge Detector Handles Constant-Valued (Flat) Images

The Canny edge detector now accepts single-valued images, also called monotone images, constant-value images, or flat images. Instead of issuing an error when an input image is single-valued, the `edge` function used with Canny edge detection now returns an output image containing all zeros, indicating that it found no edges.

imcomplement Returns Correct Answer for Signed Integer Input

`imcomplement` was incorrectly calculating the complement of an image with signed integer data type. This problem has been fixed.

imlincomb Correctly Handles int16 Data Combined with Scalar Having a 0.5 Fractional Part

`imlincomb` was giving an incorrect answer when combining `int16` data with a scalar that had 0.5 as a fractional part. This problem has been fixed.

iradon No Longer Introduces a Vertical Shift of One Pixel

The `iradon` function now correctly calculates the vertical origin of the input projections. Previously the calculated origin was off by one for inputs with an even number of projection samples. The effect of this problem could be observed by computing the Radon transform (using radon) of a test pattern containing horizontal edges, followed by computing the inverse Radon transform (using `iradon`). Careful comparison of the test pattern with the output of `iradon` showed a vertical misregistration of one pixel.

imshow Correctly Renders Indexed Images with Colormaps Having More Than 256 Colors

On the Windows platform, `imshow` now sets the figure's `'Renderer'` property to `'zbuffer'` for indexed images with associated colormaps having more than 256 entries, so they render correctly. Previously these images would appear black.

Compatibility Considerations

The issues mentioned here are all described in more detail in previous sections.

- Changes to `imshow` syntax and to toolbox preferences. Old syntaxes and preferences will still work as expected, but they will now warn.
- The functions `edgetaper`, `im2bw`, `imfilter`, and `improfile` may give slightly different answers from previous releases in certain cases.

Obsolete and Deleted Functions

The following tables lists toolbox functions that have been made obsolete or removed in Version 5.

Function	Status
`dctmtx2`	This function, which was obsoleted in previous release, has been removed from the toolbox.
`im2mis`	This function, which was obsoleted in previous release, has been removed from the toolbox.
`imview`	This function is obsolete. It now warns and passes its arguments to the `imtool` function.
`imzoom`	This function, which was obsoleted in previous release, has been removed from the toolbox.
`uintlut`	This function, which only accepted `uint8` and `uint16` data, now warns and passes arguments to the `intlut` function, which also handles `int16` data.

General Issues

The following are known issues.

- "imoverview, imoverviewpanel, imscrollpanel and imtool Performance with Large Intensity Images" on page 81
- "Image Tool Cursor Interactions" on page 81
- "Adjust Contrast Tool Does Not React to Changes in CLim, CData, or CDataMapping" on page 81
- "impixelregionpanel Might Interfere with ButtonDown events in Parent Figure" on page 82
- "imagemodel, imageinfo, and imattributes Return Incorrect Data Type" on page 82
- "cpselect Function Is Not Compilable" on page 82
- "Example in warp Function Reference Page Displays Incorrectly" on page 82

imoverview, imoverviewpanel, imscrollpanel and imtool Performance with Large Intensity Images

There is a performance problem with the Image Tool and its related navigation tools when used with large intensity images.

Possible workarounds:

- For images of any class, the image can be treated as an RGB image via `repmat`.

  ```
  I = imread('concordorthophoto.png');
  imtool(repmat(I,[1 1 3]))
  ```

- For images of class `uint8` or `uint16`, the image can be treated as an indexed image.

  ```
  I = imread('concordorthophoto.png');
  n = double(intmax(class(I)));
  map = gray(n);
  imtool(I,map)
  ```

Both workarounds will make `imcontrast` unusable and pixel reporting will be for the wrong image type. The first workaround uses more memory than the second workaround.

Image Tool Cursor Interactions

If you run the Image Tool and activate one of the navigational tools (zoom in, zoom out, pan) prior to turning on the Adjust Contrast tool, the navigational tool will be turned off and will be replaced by the mouse behavior of `imcontrast`. Conversely, if you run `imtool`, start the Adjust Contrast tool, and then start a navigation tool, the mouse behavior of `imcontrast` is turned off.

Adjust Contrast Tool Does Not React to Changes in CLim, CData, or CDataMapping

If you turn on the Adjust Contrast tool using `imcontrast` or `imtool`, and then set the axes `CLim` property or the image `CData` or `CDataMapping` properties, the Adjust Contrast tool does respond to these changes in the image or axes. Once you click on the tool, it updates based on changes to the `CLim` values.

impixelregionpanel Might Interfere with ButtonDown events in Parent Figure

If you create an `impixelregionpanel` and another `imscrollpanel` in the same figure, the `impixelregionpanel` can interfere with `ButtonDown` events throughout the figure. This can, for example, prevent the user from clicking and dragging a position rectangle located in an `imscrollpanel` elsewhere in the figure.

Workaround: After creating all of the panels needed for your GUI, execute this code:

```
uistack(hPixelRegionPanel, 'bottom')
```

where `hPixelRegionPanel` is the handle returned by `impixelregionpanel`.

imagemodel, imageinfo, and imattributes Return Incorrect Data Type

The `imagemodel`, `imageinfo`, and `imattributes` functions return class `double` for `int16` or `single` images. These functions determine the data type by querying the image object's `CData`. For `int16` and `single` images, the image object converts its `CData` to class `double`.

For example,

```
h = imshow(int16(ones(10)));
class(get(h,'CData'));
```

returns `'double'`. Consequently, `imageinfo` and `imattributes` would return a class type of `double` and calling the image model object's `getClassType` method returns double.

cpselect Function Is Not Compilable

You cannot compile MATLAB applications that call the `cpselect` function.

Example in warp Function Reference Page Displays Incorrectly

On some Windows XP systems, the example on the `warp` function reference page does not display correctly. When `warp` is called, the axes is empty. To work around this problem, switch the renderer to the `Zbuffer` renderer, as follows.

```
set(gcf,'renderer','ZBuffer')
```

Issues Specific to the Linux Platform

The following are known issues with Image Processing Toolbox 5 on Linux systems.

Alt+Click Zoom Behavior on Linux Systems

On Linux systems, if the **Alt+Click** combination is defined for any other Window Manager behavior, when you choose the zoom in or zoom out tools in the Image Tool, **Alt+click** will not zoom in the opposite direction from the currently selected tool.

Issues Specific to the Macintosh Platform

The following issues are unique to Macintosh systems.

- "Image Tool Limitations" on page 83
- "Image Information Tool Not Supported" on page 83
- "Resize Behavior of impixelinfo and imdisplayrange" on page 84
- "Difference in Scroll Bar Behavior" on page 84
- "Pixel Region Tool Slow on Macintosh Systems" on page 84
- "Alt+Click Does Not Work for Opposite Zoom in imtool" on page 84

Image Tool Limitations

On Macintosh systems, the Image Tool (`imtool`) has the following limitations:

- The Magnification tool is not supported.
- The Image Information toolbar button is not functional.

Image Information Tool Not Supported

On Macintosh systems, the `imageinfo` function is not supported because it require Java™ figures, which are not available on this platform.

Resize Behavior of impixelinfo and imdisplayrange

On Macintosh systems, the `impixelinfo` and `imdisplayrange` functions do not resize correctly because they require Java figures, which are not available on this platform.

Difference in Scroll Bar Behavior

On Macintosh systems, if you create a scroll panel (`imscrollpanel`) and drag the scroll bars, the image does not update until you release the mouse from the drag. On other platforms, the image updates continuously during the drag.

Pixel Region Tool Slow on Macintosh Systems

On Macintosh systems, the `impixelregion` and `impixelregionpanel` are slower than on other platforms.

Alt+Click Does Not Work for Opposite Zoom in imtool

On Macintosh systems, if you are using the Image Tool and choose the zoom in or zoom out tool, **Alt+click** does not zoom in the opposite direction from the currently selected tool.

Version 4.2 (R13SP2) Image Processing Toolbox

This table summarizes what's new in Version 4.2 (R13SP2).

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	Yes—Details labeled as **Compatibility Considerations**, below. See also Summary.	Bug Fixes Details

New features and changes introduced in this version are

- "Enhanced DICOM Support" on page 85
- "Major Bug Fixes" on page 86

Enhanced DICOM Support

The toolbox includes the following enhancements to the DICOM functions:

- The `dicomwrite` function can now write data in any modality that can be read using `dicomread`. Note, however, that when writing data in these modalities, `dicomwrite` does not verify the data or check to see if the correct amount of data is written.
- The toolbox can now read and write private metadata, even if they are not defined in the DICOM data dictionary. When private metadata is not in the data dictionary, `dicomread` uses generic names for the metadata fields, rather than the descriptive names available to attributes defined in the data dictionary.
- You can create a custom data dictionary that contains definitions of your private metadata, using the `dicomdict` function.

Compatibility Considerations

The DICOM data dictionary changed so that some words are different now than in previous releases. This is due to updates in the data dictionary as defined by the DICOM standards committee.

Major Bug Fixes

The Image Processing Toolbox 4.2 includes the following bug fixes:

- The `bwdist` function now produces correct results for very large images (such as 10000-by-5000).
- The `dicomread` function no longer produces the "Error using reshape" message when reading a DICOM file containing multiframe data and overlays stored in both the pixel data and in the metadata.
- The `dicomwrite` function can now correctly handle data attributes with multiple VMs (value multiplicity).
- The `dicomwrite` function now produces correct results for the RLE (run-length encoding) compression type when the number of bits per pixel is greater than 8.
- The `im2col` function, when called with the syntax `im2col(1:N, [1 N])`, now returns a column vector.
- The `imview` function, when called with the syntax `imview(I,[])`, where `I` is a constant image, no longer produces a warning message about a badly conditioned polynomial.
- The `montage` function no longer displays an extra blank row when displaying certain multiples of images.
- The `poly2mask` function no longer errors when trying to close certain polygons, specifically polygons where the last element of vector `X` doesn't match the first element, `x(1) ~= x(end)`, and the last element of vector `Y` matches the first, `y(1) == y(end)`, or vice versa.
- The `stretchlim` function now uses more bins to achieve better results for input images of class `uint16` or `double`.

Version 4.1 (R13SP1) Image Processing Toolbox

This table summarizes what's new in Version 4.1 (R13SP1).

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	No	Bug Fixes Details

New features and changes introduced in this version are

- "Reading and Writing Data with JPEG Lossless Compression" on page 87
- "Reading ICC Profiles Embedded in TIFF Files" on page 88
- "Reading and Writing L*a*b* Color Data" on page 88
- "Major Bug Fixes" on page 88

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 version of the toolbox 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.

Version 4.0 (R13+) Image Processing Toolbox

This table summarizes what's new in Version 4.0 (R13+).

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	Yes—Details labeled as **Compatibility Considerations**, below. See also Summary.	Bug fixes Details

New features and changes introduced in this version are

- "New Image Viewer" on page 90
- "Enhanced Color Space Functions" on page 92
- "New Image Enhancement Methods" on page 92
- "Enhanced DICOM Support" on page 92
- "Fan Beam Projection Transforms" on page 93
- "Boundary Tracing Functions" on page 93
- "Unsigned Integer Lookup Tables" on page 93
- "Optimized Image Arithmetic Functions" on page 93
- "Performance Improvements" on page 94
- "Changes to Existing Functions" on page 94
- "Major Bug Fixes" on page 96
- "Compatibility Considerations" on page 97

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.

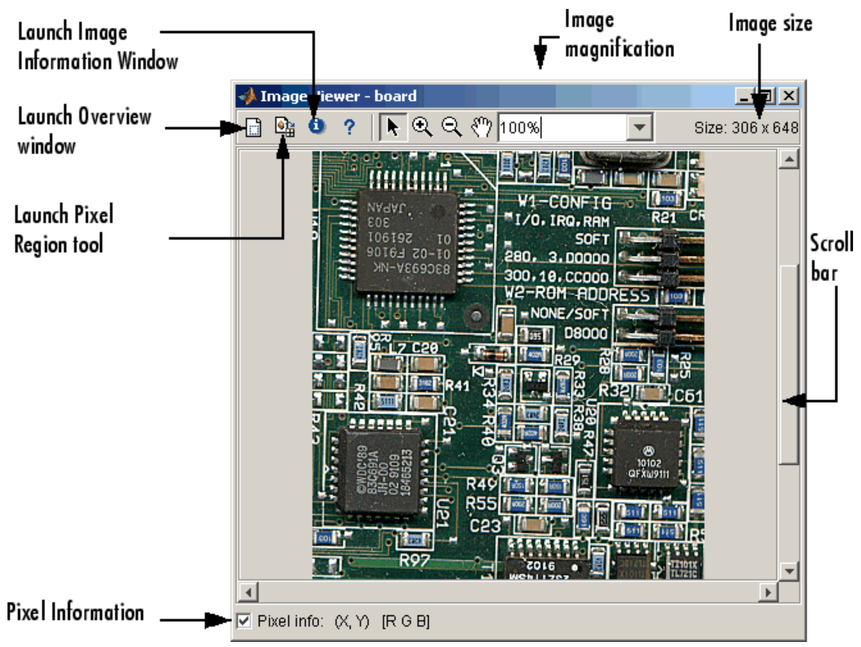
To start the Image Viewer, use the `imview` function.

```
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 `ippl` 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`

Changes to Existing Functions

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

Function	Enhancement
`cp2tform`	The algorithms have been modified to make them more robust numerically when the input coordinates have a very large offset from the origin.
`cpselect`	The Control Point Selection Tool now displays a single legend window even if multiple instances of the tool are being displayed.
`imadjust`	Supports a simplified, single-input syntax, in which it uses `stretchlim` to compute contrast-stretch parameters automatically.
`immovie`	No longer flickers while a movie is being generated.

Function	Enhancement
`medfilt2`	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.
`ordfilt2`	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.
`regionprops`	Uses an improved method for computing the convex hull of a labeled object, resulting in more accurate `ConvexHull`, `Convexity`, and `ConvexImage` measurements.
`roipoly`	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 `roipolyold`.
`tformfwd` and `tforminv`	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`	$sRGB$ profile
`swopcmyk.icm`	CMYK input profile

Major Bug Fixes

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

Function	Fix
`bwlabel`	No longer produces a segmentation violation when called with an empty matrix.
`cpselect`	• The Control Point Selection Tool no longer causes MATLAB to hang when invoked from within a script or GUI with the `input` or `uiwait` functions, which wait for users to complete the selection. • Point prediction in the Control-Point Selection Tool no longer fails with the following error message: `Attempt to reference field on non-structure array 'pickPair'.`
`dicomread`	Now reads DICOM files that contain 1-bit overlay data.

Function	Fix
`imabsdiff`, `imadd`, `imdivide`, `imlincomb`, `imsubtract`	The image arithmetic functions no longer error when called with logical inputs
`imfill`	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.
`imresize`	The anti-aliasing filtering, applied by `imresize` when shrinking an image, no longer causes a narrow strip of dark pixels to appear around the edge of the image.
`label2rgb`	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.
`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 `strel('ball', r, h, 0)` returns an empty `strel` object instead of a degenerate ball structuring element with radius 0.

Compatibility Considerations

Obsolete and Deleted Functions

The following functions have been obsoleted or deleted.

Function	Status
`isrgb`	This function has been obsoleted and will issue a warning when called
`isind`	This function has been obsoleted and will issue a warning when called.
`isbw`	This function has been obsoleted and will issue a warning when called.
`isgray`	This function has been obsoleted and will issue a warning when called

Version 3.2 (R13+) Image Processing Toolbox

This table summarizes what's new in Version 3.2 (R13+).

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	Yes—Details labeled as **Compatibility Considerations**, below. See also Summary.	No

New features and changes introduced in this version are

- "Writing DICOM Files" on page 99
- "Representing Binary Images" on page 99
- "Better Input Image Type Checking" on page 101
- "Changes to Existing Functions" on page 101

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.

Compatibility Considerations

The change in the representation of binary images has the following compatibility considerations.

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
```

Better Input Image Type Checking

The toolbox now performs more error checking of input images, specifically input classes, attributes and option string processing, with clearer error messages

Changes to Existing Functions

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

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

Version 3.1 (R12.1) Image Processing Toolbox

This table summarizes what's new in Version 3.1 (R12.1).

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	No	Fixed bugs Details

New features and changes introduced in this version are

- "deconvblind Added to Deblurring Functions" on page 103
- "label2rgb Added to Toolbox" on page 103
- "Major Bug Fixes" on page 103

deconvblind Added to Deblurring Functions

The toolbox now includes the `deconvblind` function which deblurs an image using the blind deconvolution algorithm.

label2rgb Added to Toolbox

The toolbox now includes a new utility function, `label2rgb`, that converts a label matrix into an RGB color image.

Major 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 `NaN`s 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.

Version 3.0 (R12+) Image Processing Toolbox

This table summarizes what's new in Version 3.0 (R12+).

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	No	No

New features and changes introduced in this version are

- "Morphology" on page 104
- "Spatial Transformations" on page 106
- "Image Registration" on page 107
- "Integer Image Arithmetic" on page 107
- "Integer Image Filtering" on page 108
- "Deconvolution/Deblurring" on page 108
- "Support for DICOM Files" on page 109
- "Miscellaneous New Functions" on page 109
- "New Demos" on page 110

Morphology

This release 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 (morphological opening and closing) 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
`bwareaopen`	Binary area open (remove small objects)
`bwdist`	Distance transform
`bwhitmiss`	Binary hit-miss operation
`bwlabeln`	Label-connected components in N-D binary image
`bwpack`	Pack binary image
`bwulterode`	Ultimate erosion
`bwunpack`	Unpack binary image
`conndef`	Default connectivity array
`imbothat`	Perform bottom-hat filtering
`imclearborder`	Suppress light structures connected to image border
`imclose`	Close image
`imdilate`	Dilate image
`imerode`	Erode image
`imextendedmax`	Extended-maxima transform
`imextendedmin`	Extended-minima transform
`imfill`	Fill image regions and holes
`imhmax`	H-maxima transform
`imhmin`	H-minima transform
`imimposemin`	Impose minima
`imopen`	Open image
`imreconstruct`	Morphological reconstruction
`imregionalmax`	Regional maxima
`imregionalmin`	Regional minima
`imtophat`	Tophat filtering
`strel`	Create morphological structuring element

Function	Description
`strel/getheight`	Get structuring element height
`strel/getnhood`	Get structuring element neighborhood
`strel/getsequence`	Get sequence of decomposed structuring elements
`strel/isflat`	Return true for flat structuring element
`strel/reflect`	Reflect structuring element about its center
`strel/translate`	Translate structuring element
`watershed`	Find image watershed regions

Spatial Transformations

This release includes 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.

Function	Description
`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`)

Function	Description
`tformarray`	Geometric transformation of a multidimensional array
`tformfwd`	Apply inverse geometric transformation
`tforminv`	Apply forward geometric transformation

Image Registration

The toolbox includes several new 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.

Function	Description
`cp2tform`	Infer spatial transformation from control-point pairs
`cpcorr`	Tune control-point locations using cross-correlation
`cpselect`	Control-point selection tool (graphical user interface)
`cpstruct2pairs`	Convert `CPSTRUCT` to valid pairs of control points
`normxcorr2`	Normalized two-dimensional cross-correlation

Integer Image Arithmetic

The toolbox 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.

Function	Description
`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

The toolbox includes 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
`imfilter`	Filter 2-D and N-D images

Deconvolution/Deblurring

The toolbox adds support for several fundamental algorithms for the deconvolution (deblurring) of images. All of the functions support multidimensional problems.

Function	Description
`deconvblind`	Deblur image using blind deconvolution algorithm [New with Version 3.1]
`deconvlucy`	Deblur image using Lucy-Richardson algorithm
`deconvreg`	Regularized deconvolution
`deconvwnr`	Wiener deconvolution

Function	Description
`edgetaper`	Taper image edges according to PSF
`fspecial`	Existing function; added `'disk'` and `'motion'` options
`otf2psf`	Convert optical transfer function to point-spread function
`psf2otf`	Convert point-spread function to optical transfer function

Support for DICOM Files

The toolbox 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
`dicomread`	Read image data from DICOM file
`dicominfo`	Read metadata from DICOM file

Miscellaneous New Functions

This release includes 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
`circshift`	Shift array circularly Note: This function was moved into MATLAB in release 3.2 of the Image Processing Toolbox software.
`graythresh`	Compute global image threshold using Otsu's method (image enhancement)

Function	Description
`im2mis`	Convert image to Java MemoryImageSource Note: This function was renamed to `im2java` and moved into MATLAB in release 3.2 of the Image Processing Toolbox software.
`imnoise`	Added support for new noise types: `'poisson'` and `'localvar'`
`label2rgb`	Convert label matrix to RGB image [New for Version 3.1]
`padarray`	Pad array
`regionprops`	Renamed from existing function `imfeature`; extended to N-D
`stretchlim`	Find limits to contrast stretch an image

New Demos

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

Demo Name	Brief Description
`ipexconformal`	Explore a Conformal Mapping: illustrates how to use spatial- and image-transformation functions to perform a conformal mapping.
`ipexdeconvblind`	Deblurring Images Using the Blind Deconvolution Algorithm: illustrates use of the `deconvlucy` function. [New with Version 3.1]
`ipexdeconvlucy`	Deblurring Images Using the Lucy-Richardson Algorithm: illustrates use of the `deconvlucy` function.
`ipexdeconvreg`	Deblurring Images Using a Regularized Filter: illustrates use of the `deconvreg` function.
`ipexdeconvwnr`	Deblurring Images Using the Wiener Filter: illustrates use of the `deconvwnr` function.

Demo Name	Brief Description
`ipexgranulometry`	Finding the Granulometry of Stars in an Image: illustrates how to use morphology functions to perform granulometry.
`ipexmri`	Exploring 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.
`ipexnormxcorr2`	Registering an Image Using Normalized Cross-correlation: illustrates how to use translation to align two images.
`ipexregaerial`	Registering an Aerial Photo to an Orthophoto: illustrates how to use the Control Point Selection Tool to align two images.
`ipexrotate`	Finding the Rotation and Scale of a Distorted Image: illustrates how to use the `cp2tform` function to get the rotation angle and scale factor of a distorted image.
`ipexsegcell`	Detecting a Cell Using Image Segmentation: illustrates how to use dilation and erosion to perform edge detection.
`ipexsegmicro`	Detecting Microstructures Using Image Segmentation: illustrates how to use morphological opening and closing to extract large objects from an image.
`ipexsegwatershed`	Detecting Touching Objects Using Watershed Segmentation: illustrates use of morphology functions to perform marker-control watershed segmentation.

Demo Name	Brief Description
`ipexshear`	Padding and Shearing an Image Simultaneously: illustrates how to use the padding options of the image transformation functions.
`ipextform`	Creating a Gallery of Transformed Images: illustrates how to use the `imtransform` function to perform many types of image transformations.

Version 2.2.2 (R12) Image Processing Toolbox

This table summarizes what's new in Version 2.2.2 (R12).

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	No	Fixed bugs Details

New features and changes introduced in this version are:

- "New Demo" on page 113
- "Support For Function Handles" on page 113
- "Documentation Enhanced" on page 114
- "Major Bug Fixes" on page 114

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 documentation 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

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.

Compatibility Summary for Image Processing Toolbox

This table summarizes new features and changes that might cause incompatibilities when you upgrade from an earlier version, or when you use files on multiple versions. Details are provided in the description of the new feature or change.

Version (Release)	New Features and Changes with Version Compatibility Impact
Latest Version V7.2 (R2011a)	See the **Compatibility Considerations** subheading for each of these new features and changes: • "Reduced Memory Use for std2" on page 6 • "Reduced Memory Use for watershed" on page 6 • "edge Function No Longer Smooths Image Twice" on page 7 • "Functions and Function Elements Being Removed" on page 8
V7.1 (R2010b)	See the **Compatibility Considerations** subheading for this new feature: • "New corner Function Detects Corners in Image" on page 9
V7.0 (R2010a)	See the **Compatibility Considerations** subheading for this change: • "Non-interactive Syntax of improfile Returns Different Output" on page 15

Version (Release)	New Features and Changes with Version Compatibility Impact
V6.4 (R2009b)	See the **Compatibility Considerations** subheading for each of these new features and changes: • "New blockproc Function to Process Large Images" on page 16 • "Expanded hough Function Allows Specification of Arbitrary Theta Search Space" on page 18 • "The imfilter Function Now Faster for uint16 and double Inputs" on page 19 • "Improved Speed for Calculating N-D Euclidean Distance Transforms with the bwdist Function" on page 19 • "Modified Behavior for the regionprops ConvexHull Property" on page 20
V6.3 (R2009a)	See the **Compatibility Considerations** subheading for each of these new features and changes: • "New Dialog Box for Setting Toolbox Preferences" on page 23 • "nitfread Now Allows Image Subregion Selection" on page 24 • "Five Functions Moved to MATLAB" on page 25 • "Fan-Beam Functions Updated" on page 25

Version (Release)	New Features and Changes with Version Compatibility Impact
V6.2 (R2008b)	See the **Compatibility Considerations** subheading for each of these new features and changes: • "The imscrollpanel 'PreserveView' Parameter Now Works for Images of All Sizes" on page 28 • "Distance Tool and Cropping Tool Now Modes in imtool" on page 28 • "immovie Command No Longer Shows Preview" on page 29 • "Replace Calls to ipttable Function with MATLAB uitable Function" on page 29 • "imcontour Second Output Argument Changed" on page 30 • "impixelinfo Tool Disappears when Image Changes" on page 30 • "Functions and Demos Being Removed" on page 31
V6.1 (R2008a)	See the **Compatibility Considerations** subheading for each of these new features and changes: • "ROI Tools Reimplemented as MATLAB Classes" on page 34 • "cp2tform Function Supports New Transformations" on page 36

Version (Release)	New Features and Changes with Version Compatibility Impact
V6.0 (R2007b)	See the **Compatibility Considerations** subheading for each of these new features and changes: • "Enhanced ROI Definition Behavior for `imcrop`, `roifill`, and `roipoly`" on page 40 • "DICOM Dictionary Upgrade" on page 41
V5.4 (R2007a)	See the **Compatibility Considerations** subheading for each of these new features and changes: • "Enhancements to imresize Function" on page 43 • "applycform Supports Tetrahedral Interpolation" on page 44 • "Control Point Selection Tool Enhancements" on page 44 • "Compatibility Considerations" on page 45
V5.3 (R2006b)	See the **Compatibility Considerations** subheading for each of these new features and changes: • "Enhancements to the imdistline Function" on page 48 • "Enhancements to ICC Color Capabilities" on page 48
V5.2 (R2006a)	See the **Compatibility Considerations** subheading for each of these new features: • "Compability Considerations" on page 51

Version (Release)	New Features and Changes with Version Compatibility Impact
V5.1 (R14SP3)	See the **Compatibility Considerations** subheading for this release: • "Compatibility Considerations" on page 54
V5.0.2 (R14SP2)	See the **Compatibility Considerations** subheading for this release: • "Compatibility Considerations" on page 59
V5.0.1 (R14SP1)	None
V5.0 (R14)	See the **Compatibility Considerations** subheading for this release: • "Compatibility Considerations" on page 79
V4.2 (R13SP2)	See the **Compatibility Considerations** subheading for this new feature: • "Enhanced DICOM Support" on page 85
V4.1 (R13SP1)	None
V4.0 (R13+)	See the **Compatibility Considerations** subheading for this release: • "Compatibility Considerations" on page 97
V3.2 (R13+)	See the **Compatibility Considerations** subheading for this new feature: • "Representing Binary Images" on page 99
V3.1 (R12.1)	None
V3.0 (R12+)	None
V2.2.2 (R12)	None