C-code generation support for more than 20 functions using MATLAB Coder ........................................ 1-2

gabor and imgaborfilt: New class and function for designing and applying Gabor filter banks for use in edge and texture analysis ................................................................. 1-2

grayconnected and imboxfilt: Segment regions by intensities and apply spatial domain filters ............... 1-3

Performance: Performance improvements for grayscale morphology and image filtering ................... 1-3

dpxread and dpxinfo: Read Digital Picture Exchange files ................................................................. 1-3

imwarp function supports new SmoothEdges parameter ...... 1-3

DICOM functions support new options ......................... 1-3

New example shows use of imaging functions with the Vision HDL Toolbox ........................................... 1-4

C-code generation support for more than 20 functions, including regionprops, watershed, bweuler, bwlabe, bwperim, and multithresh using MATLAB Coder ............. 2-2
App for batch processing sets of images .................... 2-2
Fast geodesic interactive segmentation ...................... 2-3
Optimized function for Gaussian filtering .................. 2-4
Fill entire region including border pixels ................. 2-4
Visualize results of boundary tracing .................... 2-4
Examine contents of DICOM files .......................... 2-4
Live image capture in Color Thresher app ................ 2-4
regionprops function can return results in table .......... 2-5
GPU acceleration for imregionalmax, imregionalmin,    2-5
imgaussfilt, imgaussfilt3, and regionprops functions ...
Functions Being Removed .................................. 2-5

R2014b

Apps for image segmentation and region analysis ........ 3-2
  Image Segmenter app ..................................... 3-2
  Image Region Analyzer app .............................. 3-4

C-code generation support for 16 additional functions using
MATLAB Coder, including bwtraceboundary, imadjust,
imclearborder, and medfilt2 .............................. 3-5

Nonrigid image registration .............................. 3-5

Image warping using displacement fields ................. 3-6

Image segmentation using the Fast Marching Method
algorithm ................................................. 3-6
Image comparison using mean-squared error .......................... 3-6
Image filtering based on object properties ................................. 3-6
Color space conversion functions ........................................... 3-6
activecontour function supports parameter to control
tendency of contour to expand or contract ............................... 3-7
Region-of-Interest (ROI) functions now support deletion from
context menu ........................................................................ 3-7
dicomwrite function now supports the ability to specify the
bitdepth of images written ..................................................... 3-7
GPU acceleration for bwlabel and imregdemons .................... 3-7

R2014a

C-code generation for more than 25 functions, including
dge, imfilter, imwarp, imopen, imclose, imerode, and
imdilate using MATLAB Coder ......................................... 4-2
GPU acceleration for an additional nine functions,
including bwdist, imfill, imreconstruct, iradon, radon, and
stretchlim ................................................................. 4-2
App for color image thresholding ........................................... 4-2
Image quality metrics, including peak signal to noise (psnr)
and structured similarity metric (ssim) .............................. 4-3
Guided filtering for image enhancement ................................. 4-3
Phase correlation and translation-only image registration
functions ........................................................................... 4-4
File names of Image Processing Toolbox examples
changed ............................................................................ 4-4
Location of sample images changed ................................. 4-5

regionprops function uses new algorithm to calculate perimeter .............................................. 4-5

R2013b

GPU acceleration for more than 20 functions, including
  bwmorph, edge, imresize, and medfilt2 ............................... 5-2

Additional 2D geometric transformations: piecewise linear,
  local weighted mean, and polynomial ............................... 5-2

Additional parameter in imregister and imregtform to
  specify initial transformation ........................................... 5-2

fitgeotrans function for fitting geometric transformation to
  control point pairs ......................................................... 5-3

imregister and imregtform Return Different Values ....... 5-3

Functions Being Removed ................................................... 5-3

R2013a

Image segmentation using active contours ...................... 6-2

Classes and functions for representing and applying 2-D and
  3-D geometric transformations .......................................... 6-2

Classes for defining world coordinate system of an image .. 6-2

Code generation for conndef, imcomplement, imfill, imhmax,
  imhmin, imreconstruct, imregionalmax, imregionalmin,
iptcheckconn, and padarray functions (using MATLAB Coder) ........................................ 6-2

GPU acceleration for imrotate, imfilter,imdilate,imerode, imopen,imclose,imtophat,imbothat,imshow,padarray, and bwlookup functions (using Parallel Computing Toolbox) ........................................ 6-3

Unsharp mask filtering ........................................ 6-3

R2012b

Image gradient computation with imgradient and imgradientxy functions ................................ 7-2

Histogram matching with imhistmatch function .......... 7-2

Multilevel thresholding with multithresh and imquantize functions ........................................ 7-2

3-D image registration with imregister function .......... 7-2

Code generation for bwmorph and bwlookup with MATLAB Coder ........................................ 7-2

Added function bwlookup ........................................ 7-2

Writing private metadata when anonymizing DICOM files .......................... 7-3

Expanded color options with imshowpair ..................... 7-3

Performance improvements .................................... 7-3

New Example ........................................ 7-3
Intensity-Based Image Registration .................. 8-2
Two New Functions to Visually Compare Images ....... 8-2
Circle Detection Using the Circular Hough Transform .... 8-2
Performance Improvements ............................... 8-2
New and Updated Demos ................................. 8-2
Data Type Change to Output Variable for bwdist ........ 8-3
iradon Function Updated .................................. 8-3
Functions Being Removed ................................. 8-3

Parallel Block Processing Now Possible with blockproc ... 9-2
New bwdistgeodesic Function Computes Geodesic Distance Transform ........................................ 9-2
New graydist Function Computes Gray-Weighted Distance Transform ........................................ 9-2
New imapplymatrix Function Computes Linear Combination of Color Channels .............................. 9-2
Performance Improvements ............................... 9-2
Faster Functions .......................................... 9-2
hdrread Now Correct for Small Values .................. 9-2
Change in Behavior for dicomwrite ...................... 9-3
Warning and Error ID Changes ........................................ 9-3
Functions and Function Elements Being Removed ............. 9-3

R2011a

New bwconvhull Function Computes Convex Hull Image . 10-2
New dicomwrite Option Writes Multiframe Imagery to Single File ................................................................. 10-2
nitfread Now Reads NITF Files with JPEG-Compressed Images ................................................................. 10-2
Reduced Memory Use for std2 ........................................ 10-2
Reduced Memory Use for watershed .............................. 10-2
iccread and iccwrite Now Warn in Cases of Unrecognized PrimaryPlatform Signatures ........................................ 10-3
Plot Selector Now Includes implay ................................. 10-3
Support for Code Generation from MATLAB ................. 10-3
edge Function No Longer Smooths Image Twice .......... 10-3
Functions and Function Elements Being Removed ........ 10-4

R2010b

New corner Function Detects Corners in Image ............ 11-2
Efficient Display and Navigation of Very Large Images of Arbitrary Format in imtool .............................. 11-2
Now Possible to Control Padding Behavior when Using the blockproc Function ........................................... 11-2

Writing to JPEG2000 File Format Supported by blockproc ................................................................. 11-2

Enhancements to the dicomread Function ..................... 11-2

The ImageMagnification Field of the nitfinfo Function Now Returns a Numeric Value ................................. 11-3

Performance Improvements ...................................................... 11-3
  Faster Functions ................................................................. 11-3

Functions and Function Elements Being Removed ............ 11-3

R2010a

New ImageAdapter Class Supports Custom File Formats for blockproc ...................................................... 12-2

The blockproc Function Now Supports Spatially Varying Operations ...................................................... 12-2

Plot Selector Now Generates Plots for imshow and imtool ................................................................. 12-2

makecform Now Supports White Point Adaptation ............ 12-2

Intel Integrated Performance Primitives Library Support Extended to imdilate, imerode, and medfilt2 ............. 12-2

imreconstruct Now Supports int64 and uint64 ................. 12-3

Performance Improvements ...................................................... 12-3
  Faster Functions ................................................................. 12-3
  Multithreaded Functions .................................................... 12-3

Non-interactive Syntax of improfile Returns Different Output ................................................................. 12-3
New blockproc Function to Process Large Images 13-2

Intel Integrated Performance Primitives Library Upgraded and Support Extended to maci64 13-3

Expanded hough Function Allows Specification of Arbitrary Theta Search Space 13-3

The imfilter Function Now Faster for uint16 and double Inputs 13-4

Improved Speed for Calculating N-D Euclidean Distance Transforms with the bwdist Function 13-4

Modified Behavior for the regionprops ConvexHull Property 13-5

Efficient Display and Navigation of Very Large NITF-File Images in imtool 13-5

Performance Improvements 13-5
  Faster Functions 13-5
  Multithreaded Functions 13-6

Faster, Less Memory-Intensive Workflow for Labeling Regions and Measuring Their Properties 14-2

Multithreaded Implementation of imfilter Function 14-2

Efficient Display and Navigation of Very Large Images in imtool 14-2

New Dialog Box for Setting Toolbox Preferences 14-2
New imcolormaptool Function That Opens Choose Colormap Tool .......................................................... 14-3

End Point and Branch Point Detection Now Possible ................................................................. 14-3

nitfread Now Allows Image Subregion Selection ................................................................. 14-3

Support for Intel IPP on Mac ........................................................................................................ 14-3

ggetColor, getLabelVisible, and setLabelVisible Methods
Added to imdistline ......................................................................................................................... 14-4

Five Functions Moved to MATLAB .......................................................................................... 14-4

Fan-Beam Functions Updated ...................................................................................................... 14-4

R2008b

Performance Improvements ........................................................................................................ 15-2

New cornermetric Function Detects Corners ............................................................................. 15-2

Now Support Absolute Colorimetric Rendering Intent for GrayTRC and MatTRC ............................................ 15-2

New createMask Method Creates Mask for Any ROI ................................................................. 15-2

Interactive Tools Refresh when Target Image Changes ... ....................................................... 15-2

The imscrollpanel 'PreserveView' Parameter Now Works for Images of All Sizes .................. 15-2

Distance Tool and Cropping Tool Now Modes in imtool ... ....................................................... 15-3

In imtool Opening Adjust Contrast Tool No Longer Selects Window/Level Tool ......................................................... 15-3

immovie Command No Longer Shows Preview .............................................................................. 15-4

xii  Contents
Replace Calls to ipttable Function with MATLAB uitable Function ........................................... 15-4
imcontour Second Output Argument Changed .................. 15-5
impixelinfo Tool Disappears when Image Changes .......... 15-5
Some Code Moved into Different Directories ................... 15-6
Functions and Demos Being Removed ......................... 15-6

R2008a

Create High Dynamic Range (HDR) Images and Write Them to Files ........................................... 16-2
Measure Properties of Regions in Grayscale Images ....... 16-2
Display Very Large Images by Subsampling ................. 16-2
Enhancements to ROI Tools ........................................ 16-2
ROI Tools Reimplemented as MATLAB Classes .............. 16-2
ROI Tools Support New wait and resume Methods .......... 16-3
Interactively Add New Vertices to ROI Polygons .......... 16-3
Enhancements to Color Functions ............................... 16-3
makecform Supports Converting Between sRGB and CMYK .................................................. 16-3
icccwrite Creates Smaller ICC Profiles ....................... 16-3
cp2tform Function Supports New Transformations ......... 16-4
hough Function Uses Specified RhoResolution Values .... 16-4
Enhancements to Interactive Tools .............................. 16-4
New and Updated Demos ........................................... 16-4
Enhancements to Other Functions .............................. 16-5
New Interactive Image Sequence and Video Viewer ........ 17-2

Image Tool Includes Cropping, Enhanced Contrast
   Adjustment, and Saving of Modified Images .......... 17-2

New Function for Converting Bayer Pattern Encoded Images to RGB ..................... 17-2

New Function for Creating a Multiresolution Gaussian Pyramid ..................... 17-2

Enhanced ROI Definition Behavior for imcrop, roifill, and roipoly ..................... 17-3

New Modular Interactive GUI-building Tools ..................... 17-3

New Programmable ROI Tools ..................... 17-3

Support for Reading NITF and HDR Images ..................... 17-4

Enhanced Performance ..................... 17-4

DICOM Dictionary Upgrade ..................... 17-4

Changes to Other Functions ..................... 17-4

Enhancements to imresize Function ..................... 18-2

applycform Supports Tetrahedral Interpolation ..................... 18-2

Control Point Selection Tool Enhancements ..................... 18-2

Enhancements to impoint, imline, and imrect Functions ..................... 18-3
Enhancements to montage Function ........................................... 18-3

**makecform** Uses 'icc' Whitepoint for L*a*b*/sRGB Conversions ................................................................. 18-3

**normxcorr2** Might Return Different Results .......................... 18-3

**watershed** Uses New Algorithm ........................................... 18-3

Changes to Other Functions .................................................. 18-4

---

**R2006b**

Enhancements to DICOM Capabilities ................................. 19-2

New Symmetric Option with graycomatrix Function ............ 19-2

Enhancements to ICC Color Capabilities .......................... 19-2

Enhancements to the imdistline Function ......................... 19-2

**setColor Method Accepts Predefined Color Strings** ........ 19-3

---

**R2006a**

Enhanced ICC Profile Capabilities ................................. 20-2

New Pointer Management Functions ................................. 20-2

New Constraint Creation Function .................................. 20-2

Functions **cp2tform, tforminv, imtransform** .................. 20-2

**IPPL Not Used on 64-Bit Systems** ................................. 20-3
Support for Two New Medical Image File Formats .... ... 21-2

New Point, Rectangle, and Line Functions ............... 21-2

Image Tool Enhancements and Improvements .............. 21-2
  New Distance Tool .................................. 21-2
  Adjust Contrast Tool Enhancements and Improvements ... 21-2

New Utility Functions for Use with Profile-Based Color Space Conversion Functions .................. 21-2

New Documentation on Processing Image Sequences .... 21-2

Control Point Selection Tool Now Works on Macintosh Systems ........................................ 21-3

Obsolete and Deleted Functions ............................. 21-3

Image Tool is Not Compilable ............................. 21-3

Major Bug Fixes ........................................... 22-2
  Major Revisions to Fan-Beam Functions .................. 22-2
  Changes to the DICOM Functions ........................ 22-2
  Changes to Image Tool and Modular Interactive Tools ... 22-3
  Changes to the imshow Function ......................... 22-4
  Fixes to Other Functions ................................ 22-4
  Fixes to Image Processing Toolbox Deployment Issues ... 22-5

New Directory Needed .................................... 22-5
R2015b

Version: 9.3

New Features

Bug Fixes
C-code generation support for more than 20 functions using MATLAB Coder

The following table lists the Image Processing Toolbox™ functions that have been enabled for code generation in this release. For all target platforms, these functions generate C code. Some of these functions can also generate C code that uses a precompiled, platform-specific shared library (.dll, .so, or .dylib). Using a shared library preserves performance optimizations in these functions but limits the target to only those platforms that support MATLAB (see system requirements). For a complete list of Image Processing Toolbox that support code generation, see “List of Supported Functions with Usage Notes”.

<table>
<thead>
<tr>
<th>Function</th>
<th>Function</th>
<th>Function</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>bwareaopen</td>
<td>houghpeaks</td>
<td>immse</td>
<td>integralBoxFilter</td>
</tr>
<tr>
<td>grayconnected</td>
<td>imabsdiff</td>
<td>imresize</td>
<td>psnr</td>
</tr>
<tr>
<td>hough</td>
<td>imcrop</td>
<td>imrotate</td>
<td></td>
</tr>
<tr>
<td>houghlines</td>
<td>imgaborfilt</td>
<td>imtranslate</td>
<td></td>
</tr>
</tbody>
</table>

1 If you choose the generic MATLAB Host Computer option in the MATLAB® Coder™ configuration settings, this function generates C code that uses a precompiled, platform-specific shared library.

This release also includes functions, listed in the following table, that previously generated C code that uses a platform-specific shared library but now also generate C code that does not require a shared library. For these functions, if you choose the generic MATLAB Host Computer option in the MATLAB Coder configuration settings, these functions generate C code that uses a precompiled, platform-specific shared library.

<table>
<thead>
<tr>
<th>Function</th>
<th>Function</th>
<th>Function</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>histeq</td>
<td>im2uint16</td>
<td>imlincomb</td>
<td>rgb2ycbcr</td>
</tr>
<tr>
<td>im2int16</td>
<td>imadjust</td>
<td>intlut</td>
<td>ycbcr2rgb</td>
</tr>
</tbody>
</table>

**gabor and imgaborfilt: New class and function for designing and applying Gabor filter banks for use in edge and texture analysis**

The toolbox includes two new functions, imgaborfilt and gabor, that provide an easy-to-use interface for Gabor filtering.
**grayconnected and imboxfilt:** Segment regions by intensities and apply spatial domain filters

The toolbox includes a new function, `grayconnected`, that provides a way to segment regions of similar intensity in a grayscale image. In addition, the toolbox includes several new functions that create an integral image, `integralImage` and `integralImage3`, and that can use the integral image for filtering, `imboxfilt`, `imboxfilt3`, `integralBoxFilter`, and `integralBoxFilter3`.

**Performance: Performance improvements for grayscale morphology and image filtering**

The performance of several morphological processing functions and image filtering functions has been improved.

**dpxread and dpxinfo: Read Digital Picture Exchange files**

The toolbox includes two new functions, `dpxread` and `dpxinfo`, that provide an easy-to-use interface for reading DPX files.

**imwarp function supports new SmoothEdges parameter**

The `imwarp` function supports a new parameter, `SmoothEdges`, that controls whether `imwarp` pads the input image to create a smoother edge in the output image.

**DICOM functions support new options**

The `dicominfo`, `dicomread`, `dicomdisp`, and `ddicomnon` functions supports a new parameter, `UseVRHeuristic`. When set to `true` (the default), this parameter instructs the parser to use a heuristic to help read certain noncompliant files which switch value representation (VR) modes incorrectly. The functions issue a warning if they use the heuristic.

The `dicominfo` function supports a new parameter, `UseDictionaryVR`, which specifies whether the data types of returned metadata should conform to the data dictionary, regardless of what information is present in the file.
New example shows use of imaging functions with the Vision HDL Toolbox

There is a new example, “Generate HDL Code for Image Sharpening,” that showcases use of Image Processing Toolbox functions with the Vision HDL Toolbox™.
R2015a

Version: 9.2

New Features

Bug Fixes

Compatibility Considerations
C-code generation support for more than 20 functions, including `regionprops`, `watershed`, `bweuler`, `bwlabel`, `bwperim`, and `multithresh` using MATLAB Coder

The following table lists the Image Processing Toolbox functions that have been enabled for code generation in this release. For all target platforms, these functions generate C code. If you choose the generic MATLAB Host Computer option in the MATLAB Coder configuration settings, many of these functions also generate C code that uses a precompiled, platform-specific shared library (.dll, .so, or .dylib). Using a shared library preserves performance optimizations in these functions but limits the target to only those platforms that support MATLAB (see system requirements). For a complete list of Image Processing Toolbox that support code generation, see List of Supported Functions with Usage Notes.

<table>
<thead>
<tr>
<th>bweuler¹</th>
<th>bwperim¹</th>
<th>watershed¹</th>
</tr>
</thead>
<tbody>
<tr>
<td>bwlabel</td>
<td>regionprops¹</td>
<td></td>
</tr>
</tbody>
</table>

¹ If you choose the generic MATLAB Host Computer option in the MATLAB Coder configuration settings, these functions generate C code that uses a precompiled, platform-specific shared library.

This release also includes functions, listed in the following table, that previously generated C code that uses a platform-specific shared library but now also generate C code that does not require a shared library. You choose which by specifying the platform in MATLAB Coder. If you choose the generic MATLAB Host Computer option in the MATLAB Coder configuration settings, these functions generate C code that uses a precompiled, platform-specific shared library.

<table>
<thead>
<tr>
<th>bwselect</th>
<th>imclearborder</th>
<th>imfill</th>
<th>imhmin</th>
<th>imregionalmin</th>
<th>multithresh</th>
</tr>
</thead>
<tbody>
<tr>
<td>edge</td>
<td>imextendedmax</td>
<td>imhist</td>
<td>imreconstruct</td>
<td>imwarp</td>
<td>ordfilt2</td>
</tr>
<tr>
<td>im2uint8</td>
<td>imextendedmin</td>
<td>imhmax</td>
<td>imregionalmax</td>
<td>medfilt2</td>
<td>stretchlim</td>
</tr>
</tbody>
</table>

App for batch processing sets of images

The toolbox includes new app that facilitates batch processing called the Batch Processing App, shown below. The Batch Processing app enables you to perform an image processing operation on a selection of files or the entire contents of a folder. Using the
app, you specify the folder and the function that you want executed. For an example, see Batch Processing Using the Image Batch Processor App.

Fast geodesic interactive segmentation

The toolbox includes a new function, imseggeodesic, that provides adaptive, geodesic distance-based binary or trinary segmentation for color images. With this function, you
can specify a few pixels, called scribbles, belonging to the different regions (for example, background and foreground in binary segmentation) in the image, and from them the whole image is automatically segmented.

**Optimized function for Gaussian filtering**

The toolbox includes two new functions, imgaussfilt and imgaussfilt3, that provide an easy-to-use interface for Gaussian filtering for both 2-D and 3-D data. With these functions, you can switch between filtering in the spatial or frequency domains. In addition, these functions support anisotropic Gaussian filters, enabling you to apply Gaussian smoothing with different standard deviations along each data dimension. Use these functions instead of using `fspecial` and `imfilter` to create a Gaussian kernel and then applying it.

**Fill entire region including border pixels**

The toolbox includes a new function, `regionfill`, that provides a way to fill a specified region in an image using inward interpolation so that the region blends in with the surrounding background. `regionfill` replaces the `roifill` function. `regionfill` fill in regions in an image including the border pixels. `roifill` left the pixels on the border of the region unprocessed.

**Visualize results of boundary tracing**

The toolbox includes a new function, `visboundaries`, that provides a way to plot region boundaries on set of axes. You can use this function to visualize the results returned by the `bwboundaries` function.

**Examine contents of DICOM files**

The toolbox includes a new function, `dicomdisp`, that displays the metadata of a DICOM file at the command prompt. `dicomdisp` can be useful when debugging issues with DICOM files.

**Live image capture in Color Thresher app**

You can now do color thresholding on an image acquired from a Webcam using the Color Thresher app. The new Image Capture tab allows you to bring a live image from USB
Webcams into the app. Previously, you had to save your images to disk and manually add them into the app.

The image capture functionality in the Color Thresher app allows you to:

- Capture live images from USB Webcams
- Save an acquired image to a variable
- Integrate between image acquisition and color thresholding
- Control camera properties, such as brightness and contrast

Use the `colorThresholder` function to open the app. Then select **Load Image > Acquire Image From Camera** to open the Image Capture tab. Select your device, set any properties, and preview the image. You can then capture an image.

**regionprops function can return results in table**

The `regionprops` function can now return results in a table. To use this capability, specify the string `'table'` as the first input argument.

**GPU acceleration for `imregionalmax`, `imregionalmin`, `imgaussfilt`, `imgaussfilt3`, and `regionprops` functions**

This release adds GPU acceleration for several toolbox functions: `imgaussfilt`, `imgaussfilt3`, `imregionalmax`, `imregionalmin`, and `regionprops`. GPU acceleration for these functions requires Parallel Computing Toolbox™ software and meeting the GPU computing requirements detailed here.

### Functions Being Removed

<table>
<thead>
<tr>
<th>Functionality</th>
<th>What Happens When You Use This Functionality?</th>
<th>Use This Instead</th>
<th>Compatibility Considerations</th>
</tr>
</thead>
<tbody>
<tr>
<td>roifill</td>
<td>Still works but issues a warning</td>
<td>regionfill</td>
<td>Replace instances of <code>roifill</code> with <code>regionfill</code>.</td>
</tr>
</tbody>
</table>
R2014b

Version: 9.1

New Features

Bug Fixes
Apps for image segmentation and region analysis

The toolbox includes new apps:

- Image Segmenter app
- Image Region Analyzer app

Image Segmenter app

The Image Segmenter app enables you to segment images using the active contours algorithm. In this app, you first initialize the segmentation by specifying a rough segmentation or initial condition. When you click Evolve, the app evolves the initial segmentation, performing the number of iterations you specify, creating a binary mask image. For an example, see Image Segmentation Using the Image Segmenter App.
Image Region Analyzer app

The Image Region Analyzer app enables you to explore binary images and filter images based on the properties of regions in the image. For example, you can filter an image to remove all objects smaller than a particular size. The following figure shows the Image Region Analyzer app. When you select the value of a property in the table, the app highlights the corresponding region in the image. For examples, see Calculate Region Properties Using Image Region Analyzer and Filter Images on Region Properties Using Image Region Analyzer App.
C-code generation support for 16 additional functions using MATLAB Coder, including bwtraceboundary, imadjust, imclearborder, and medfilt2

This release includes 16 additional toolbox functions that support the generation of C code using MATLAB Coder. Some Image Processing Toolbox functions generate C code that depends on a platform-specific shared library (.dll, .so, or .dylib). Using a shared library preserves performance optimizations in these functions but limits the target to only those platforms that support MATLAB (see system requirements).

The following table lists the Image Processing Toolbox functions that have been enabled for code generation in this release. The table identifies functions that use a shared library. For a complete list of Image Processing Toolbox that support code generation, see List of Supported Functions with Usage Notes.

<table>
<thead>
<tr>
<th>bwdist&lt;sup&gt;1&lt;/sup&gt;</th>
<th>imadjust&lt;sup&gt;1&lt;/sup&gt;</th>
<th>intlut&lt;sup&gt;1&lt;/sup&gt;</th>
<th>ordfilt2&lt;sup&gt;1&lt;/sup&gt;</th>
<th>ycbcr2rgb&lt;sup&gt;1&lt;/sup&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>bwtraceboundary</td>
<td>imclearborder&lt;sup&gt;1&lt;/sup&gt;</td>
<td>iptcheckmap</td>
<td>rgb2gray&lt;sup&gt;1&lt;/sup&gt;</td>
<td></td>
</tr>
<tr>
<td>fitgeotrans</td>
<td>imlincomb&lt;sup&gt;1&lt;/sup&gt;</td>
<td>medfilt2&lt;sup&gt;1&lt;/sup&gt;</td>
<td>rgb2ycbcr&lt;sup&gt;1&lt;/sup&gt;</td>
<td></td>
</tr>
<tr>
<td>histeq&lt;sup&gt;1&lt;/sup&gt;</td>
<td>imquantize</td>
<td>multithresh&lt;sup&gt;1&lt;/sup&gt;</td>
<td>stretchlim&lt;sup&gt;1&lt;/sup&gt;</td>
<td></td>
</tr>
</tbody>
</table>

<sup>1</sup> Generated code uses a precompiled, platform-specific shared library.

This release also includes nine functions, listed in the following table, that generate C code or generate C code that uses a platform-specific shared library, depending on which platform you specify in MATLAB Coder. If you choose the generic MATLAB Host Computer option in the MATLAB Coder configuration settings, these functions generate C code that uses a precompiled, platform-specific shared library.

<table>
<thead>
<tr>
<th>bwlookup</th>
<th>imclose</th>
<th>imfilter</th>
</tr>
</thead>
<tbody>
<tr>
<td>bwthresh</td>
<td>imdilate</td>
<td>imopen</td>
</tr>
<tr>
<td>imbothat</td>
<td>imerode</td>
<td>imtophat</td>
</tr>
</tbody>
</table>

Nonrigid image registration

The toolbox includes a new function, imregdemons, that provides a way to perform nonrigid image registration. The toolbox already supports several types of rigid image registration that work at a global level, applying the same mathematical transformation
to every pixel in an image. Now the toolbox supports a function that works at a local level, capable of applying different transformations to every image pixel. The `imregdemons` function returns the warped image and a displacement field that describes how each pixel is transformed.

**Image warping using displacement fields**

The `imwarp` function now accepts the displacement field returned by the `imregdemons` as input. The `imregdemons` function returns a warped image and a displacement field that describes how each pixel is transformed. If you want more detailed control of a nonrigid registration, you can pass this displacement field to `imwarp` to perform the transformation. For example, by using `imwarp`, you can specify the interpolation method used.

**Image segmentation using the Fast Marching Method algorithm**

The toolbox includes a new function, `imsegfmm`, that segments an image using the Fast Marching Method (FMM) algorithm. To segment an image, you must first create a weight image using either the `gradientweight` or `graydiffweight` functions. These functions create an image in which every pixel is a value. You then pass this weight image to `imsegfmm`, specifying where the algorithm should start, the seed location.

**Image comparison using mean-squared error**

The toolbox includes a new function, `immse`, that calculates the mean-squared error.

**Image filtering based on object properties**

The toolbox includes two new functions, `bwareafilt` and `bwpropfilt` that filter images based on the values of image properties. For example, using `bwareafilt`, you can remove objects from an image if their area is less than a specified number of pixels.

**Color space conversion functions**

The toolbox includes several new functions to convert between the RGB, XYZ, and $L^a'b'$ color spaces.

<table>
<thead>
<tr>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>rgb2lab</code></td>
</tr>
<tr>
<td><code>rgb2xyz</code></td>
</tr>
<tr>
<td><code>lab2xyz</code></td>
</tr>
<tr>
<td>lab2rgb</td>
</tr>
<tr>
<td>--------</td>
</tr>
</tbody>
</table>

Use these new conversion functions instead of using `makecform` with these conversion types: `lab2srgb`, `srgb2lab`, `srgb2xyz`, `xyz2srgb`, `lab2xyz`, and `xyz2lab`.

**activecontour function supports parameter to control tendency of contour to expand or contract**

The `activecontour` function supports a new parameter, `ContractionBias`, that influences whether the contour grows outward or shrinks inward during segmentation.

**Region-of-Interest (ROI) functions now support deletion from context menu**

The `imellipse`, `imfreehand`, `imline`, `impoint`, `impoly`, and `imrect` functions, that you use to define regions of interest in images, now support a deletion option from their context menus.

**dicomwrite function now supports the ability to specify the bitdepth of images written**

The `dicomwrite` function now supports the `UseMetadataBitDepths` parameter which you can use to specify the bitdepth of the image written to the DICOM file.

**GPU acceleration for `bwlabel` and `imregdemons`**

This release adds GPU acceleration for two toolbox functions: `bwlabel` and `imregdemons`. GPU acceleration for these functions requires Parallel Computing Toolbox software and meeting the GPU computing requirements detailed here.
R2014a

Version: 9.0

New Features

Bug Fixes

Compatibility Considerations
C-code generation for more than 25 functions, including edge, imfilter, imwarp, imopen, imclose,imerode, and imdilate using MATLAB Coder

You can generate standalone C code for a group of toolbox functions, listed below. Generating code requires MATLAB Coder.

<table>
<thead>
<tr>
<th>affine2d</th>
<th>im2double</th>
<th>imclose</th>
<th>imhist</th>
<th>mean2</th>
</tr>
</thead>
<tbody>
<tr>
<td>bwpack</td>
<td>im2int16</td>
<td>imdilate</td>
<td>imopen</td>
<td>projective2d</td>
</tr>
<tr>
<td>bwselect</td>
<td>im2single</td>
<td>imerode</td>
<td>imref2d</td>
<td>strel</td>
</tr>
<tr>
<td>bwunpack</td>
<td>im2uint16</td>
<td>imextendedmax</td>
<td>imref3d</td>
<td></td>
</tr>
<tr>
<td>edge</td>
<td>im2uint8</td>
<td>imextendedmin</td>
<td>imtophat</td>
<td></td>
</tr>
<tr>
<td>getrangefromc</td>
<td>imbothat</td>
<td>imfilter</td>
<td>imwarp</td>
<td></td>
</tr>
</tbody>
</table>

GPU acceleration for an additional nine functions, including bwdist, imfill, imreconstruct, iradon, radon, and stretchlim

This release adds GPU acceleration for an additional nine toolbox functions, listed below. GPU acceleration for these functions requires Parallel Computing Toolbox software and meeting the GPU computing requirements detailed here.

<table>
<thead>
<tr>
<th>bwdist</th>
<th>imreconstruct</th>
<th>normxcorr2</th>
</tr>
</thead>
<tbody>
<tr>
<td>imcomplement</td>
<td>iradon</td>
<td>radon</td>
</tr>
<tr>
<td>imfill</td>
<td>mean2</td>
<td>stretchlim</td>
</tr>
</tbody>
</table>

App for color image thresholding

The toolbox includes a new Color Thresholder app that enables you to segment color images by manipulating their color components. The app presents the image using several standard color spaces. You choose which color space representation gives the best contrast between foreground and background and then use the component histograms to segment the image.
Image quality metrics, including peak signal to noise (psnr) and structured similarity metric (ssim)

The toolbox includes two new functions, ssim and psnr, that compute image quality metrics.

Guided filtering for image enhancement

The toolbox includes a new function, imguidedfilter, that provides an edge-preserving nonlinear filter for use with images.
Phase correlation and translation-only image registration functions

The toolbox includes a new function, `imregcorr`, that applies an FFT-based transform to register two images with regard to translation, rotation, and scale. The toolbox includes a new function, `imtranslate`, that applies a translation transformation to an input image and returns the transformed image. Using `imtranslate`, you can provide spatial reference information, specify the method used for interpolation, and control other aspects of the translation.

File names of Image Processing Toolbox examples changed

The file names of Image Processing Toolbox examples have changed. The following table lists the example files in alphabetical order by their former names.

<table>
<thead>
<tr>
<th>Old Example File Name</th>
<th>New Example File Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ipexaerial.m</td>
<td>RegisterAerialPhotoExample.m</td>
</tr>
<tr>
<td>ipexangle.m</td>
<td>MeasureAngleExample.m</td>
</tr>
<tr>
<td>ipexautorotate.m</td>
<td>RotationFeatureMatchingExample.m</td>
</tr>
<tr>
<td>ipexbatch.m</td>
<td>BatchProcessImageExample.m</td>
</tr>
<tr>
<td>ipexblind.m</td>
<td>BlindImageDeblurringExample.m</td>
</tr>
<tr>
<td>ipexblockproceedge.m</td>
<td>BlockProcessLargeImageExample.m</td>
</tr>
<tr>
<td>ipexblockprocstats.m</td>
<td>BlockProcessStatisticsExample.m</td>
</tr>
<tr>
<td>ipexcell.m</td>
<td>CellSegmentationExample.m</td>
</tr>
<tr>
<td>ipexcheckerboard.m</td>
<td>GalleryTransformedImagesExample.m</td>
</tr>
<tr>
<td>ipexcircles.m</td>
<td>DetectCirclesExample.m</td>
</tr>
<tr>
<td>ipexconformal.m</td>
<td>ConformalMappingImageExample.m</td>
</tr>
<tr>
<td>ipexcontrast.m</td>
<td>ContrastEnhancementExample.m</td>
</tr>
<tr>
<td>ipexfabric.m</td>
<td>LabColorSegmentationExample.m</td>
</tr>
<tr>
<td>ipexhistology.m</td>
<td>KMeansSegmentationExample.m</td>
</tr>
<tr>
<td>ipexknee.m</td>
<td>RegisterMultimodalImagesExample.m</td>
</tr>
<tr>
<td>ipexlanstretch.m</td>
<td>MultispectralImageEnhancementExample.m</td>
</tr>
<tr>
<td>ipexlucy.m</td>
<td>LucyRichardsonImageDeblurringExample.m</td>
</tr>
<tr>
<td>Old Example File Name</td>
<td>New Example File Name</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>ipexmri.m</td>
<td>MRISliceExample.m</td>
</tr>
<tr>
<td>ipexndvi.m</td>
<td>MultispectralVegetationSegmentationExample.m</td>
</tr>
<tr>
<td>ipexnormxcorr2.m</td>
<td>RegisterNormalizedCrossCorrelationExample.m</td>
</tr>
<tr>
<td>ipexpendulum.m</td>
<td>PendulumLengthExample.m</td>
</tr>
<tr>
<td>ipexprops.m</td>
<td>MeasureGrayscaleRegionsExample.m</td>
</tr>
<tr>
<td>ipexradius.m</td>
<td>MeasureRadiusExample.m</td>
</tr>
<tr>
<td>ipexreconstruct.m</td>
<td>ReconstructImageExample.m</td>
</tr>
<tr>
<td>ipexregularized.m</td>
<td>RegularizedImageDeblurringExample.m</td>
</tr>
<tr>
<td>ipexrice.m</td>
<td>NonuniformIlluminationExample.m</td>
</tr>
<tr>
<td>ipexrotate.m</td>
<td>RotationFitgeotransExample.m</td>
</tr>
<tr>
<td>ipexroundness.m</td>
<td>RoundObjectsExample.m</td>
</tr>
<tr>
<td>ipexshear.m</td>
<td>PadShearImageExample.m</td>
</tr>
<tr>
<td>ipexsnow.m</td>
<td>SnowflakesGranulometryExample.m</td>
</tr>
<tr>
<td>ipextexturefilter.m</td>
<td>TextureSegmentationExample.m</td>
</tr>
<tr>
<td>ipextraffic.m</td>
<td>TrafficSegmentationExample.m</td>
</tr>
<tr>
<td>ipexwatershed.m</td>
<td>WatershedSegmentationExample.m</td>
</tr>
<tr>
<td>ipexwiener.m</td>
<td>WienerImageDeblurringExample.m</td>
</tr>
</tbody>
</table>

Location of sample images changed

The location of Image Processing Toolbox sample images has changed. These images were stored in the `imdemos` folder and are now stored in the `imdata` folder.

regionprops function uses new algorithm to calculate perimeter

The regionprops function uses a new algorithm to calculate a perimeter, when used with the ‘Perimeter’ option. Because of this change, regionprops returns different results for the perimeter calculations than it did in earlier releases.
Compatibility Considerations

While the new algorithm used with `regionprops` returns more accurate perimeter calculation, you can get the same return value as previous releases by specifying the ‘perimeterold’ option.
R2013b

Version: 8.3

New Features

Bug Fixes

Compatibility Considerations
GPU acceleration for more than 20 functions, including bwmorph, edge, imresize, and medfilt2

This release introduces GPU acceleration for a group of toolbox functions, listed below. GPU acceleration for these functions requires Parallel Computing Toolbox software and meeting the GPU computing requirements detailed here.

<table>
<thead>
<tr>
<th>bwmorph</th>
<th>im2single</th>
<th>imgradientxy</th>
<th>medfilt2</th>
</tr>
</thead>
<tbody>
<tr>
<td>corr2</td>
<td>im2uint8</td>
<td>imhist</td>
<td>rgb2gray</td>
</tr>
<tr>
<td>edge</td>
<td>im2uint16</td>
<td>imlincomb</td>
<td>rgb2ycbcr</td>
</tr>
<tr>
<td>histeq</td>
<td>imabsdiff</td>
<td>imnoise</td>
<td>std2</td>
</tr>
<tr>
<td>im2double</td>
<td>imadjust</td>
<td>imresize</td>
<td>stdfilt</td>
</tr>
<tr>
<td>im2int16</td>
<td>imgradient</td>
<td>mat2gray</td>
<td>ycbcr2rgb</td>
</tr>
</tbody>
</table>

Additional 2D geometric transformations: piecewise linear, local weighted mean, and polynomial

The toolbox includes several new classes for representing 2D geometric transformations, listed below.

| images.geotrans.PiecewiseLinearTransformation2d | 2D piecewise linear geometric transformation |
| images.geotrans.PolynomialTransformation2d    | 2D polynomial geometric transformation     |
| images.geotrans.LocalWeightedMeanTransformation2d | 2D local weighted mean geometric transformation |

Additional parameter in imregister and imregtform to specify initial transformation

The imregister and imregtform functions both support a new parameter, InitialTransformation, that enables you to specify the starting point for the transformation. Specifying an initial geometric transformation as a starting point, the functions can return better results.
**fitgeotrans function for fitting geometric transformation to control point pairs**

The toolbox includes a new function, fitgeotrans, that takes pairs of control points and uses them to infer a geometric transformation. The function returns a geometric transformation object that can be used with imwarp.

**imregister and imregtform Return Different Values**

Some changes to the implementation of the imregister and imregtform functions might cause these functions to return different results than they did before R2013b.

**Compatibility Considerations**

You might need to change parameters to the imregister and imregtform functions to achieve similar results to what you had previously.

**Functions Being Removed**

<table>
<thead>
<tr>
<th>Functionality</th>
<th>What Happens When You Use This Functionality?</th>
<th>Use This Instead</th>
<th>Compatibility Considerations</th>
</tr>
</thead>
<tbody>
<tr>
<td>cp2tform</td>
<td>Still works but issues a warning</td>
<td>fitgeotrans</td>
<td>Replace instances of cp2tform with fitgeotrans.</td>
</tr>
<tr>
<td>imtransform</td>
<td>Still works but issues a warning</td>
<td>imwarp</td>
<td>Replace instances of imtransform with imwarp.</td>
</tr>
<tr>
<td>maketform</td>
<td>Still works but issues a warning</td>
<td>fitgeotrans, affine2d, affine3d, or projective2d</td>
<td>Replace instances of maketform with one of the recommended functions.</td>
</tr>
</tbody>
</table>
R2013a

Version: 8.2

New Features

Bug Fixes
Image segmentation using active contours

The toolbox includes a new function, activecontour, for segmenting an image using active contours.

Classes and functions for representing and applying 2-D and 3-D geometric transformations

The toolbox includes several new classes for representing 2-D and 3-D geometric transformations, listed below.

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>affine2d</td>
<td>2-D affine geometric transformation</td>
</tr>
<tr>
<td>affine3d</td>
<td>3-D affine geometric transformation</td>
</tr>
<tr>
<td>projective2d</td>
<td>2-D projective geometric transformation</td>
</tr>
<tr>
<td>imregtform</td>
<td>Estimate the geometric transformation that aligns two images</td>
</tr>
<tr>
<td>imwarp</td>
<td>Apply geometric transformation to image</td>
</tr>
</tbody>
</table>

Classes for defining world coordinate system of an image

The toolbox includes several new classes for representing world coordinate systems associated with an image, listed below.

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>imref2d</td>
<td>Reference 2-D image to world coordinates</td>
</tr>
<tr>
<td>imref3d</td>
<td>Reference 3-D image to world coordinates</td>
</tr>
</tbody>
</table>

Code generation for conndef, imcomplement, imfill, imhmax, imhmin, imreconstruct, imregionalmax, imregionalmin, iptcheckconn, and padarray functions (using MATLAB Coder)

You can generate standalone C code for a group of toolbox functions, listed below. Generating code requires MATLAB Coder.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>conndef</td>
<td>imcomplement</td>
</tr>
<tr>
<td>imfill</td>
<td>imhmax</td>
</tr>
</tbody>
</table>
GPU acceleration for imrotate, imfilter, imdilate, imerode, imopen, imclose, imtophat, imbothat, imshow, padarray, and bwlookup functions (using Parallel Computing Toolbox)

This release introduces GPU acceleration for a group of toolbox functions, listed below. GPU acceleration for these functions requires Parallel Computing Toolbox software and meeting the GPU computing requirements detailed here.

Unsharp mask filtering

The toolbox includes a new function, imsharpen, that does unsharp mask filtering. Use this new function instead of using the 'unsharp' option with the fspecial function.
R2012b

Version: 8.1

New Features

Bug Fixes

Compatibility Considerations
Image gradient computation with `imgradient` and `imgradientxy` functions

The toolbox includes two new functions for computing image gradients: `imgradient` and `imgradientxy`. `imgradient` computes the gradient magnitude and direction. `imgradientxy` computes the X and Y directional gradients.

Histogram matching with `imhistmatch` function

The new function `imhistmatch` adjusts the histogram of an image to match the N-bin histogram of a reference image.

Multilevel thresholding with `multithresh` and `imquantize` functions

The new `imquantize` and `multithresh` functions enable multilevel grayscale thresholding and labeling. `imquantize` labels an image using a fixed range of grayscale levels. `multithresh` computes threshold levels using Otsu's method. The levels computed by `multithresh` can be used as input to `imquantize`.

3-D image registration with `imregister` function

In addition to 2-D images, `imregister` can now align 3-D images using intensity-based registration of the images' voxel data.

Code generation for `bwminopoly` and `bwlookup` with MATLAB Coder

Standalone C code can be generated for `bwminopoly` and `bwlookup`. 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.

Added function `bwlookup`

Compatibility Considerations

Function `applylut` is not recommended; use `bwlookup` instead.
Writing private metadata when anonymizing DICOM files

The dicomanon function now supports writing private metadata fields using the 'WritePrivate' parameter.

Expanded color options with imshowpair

imshowpair now supports additional color options for displaying image differences and stereo imagery when using the new 'ColorChannels' parameter. Using the 'red-cyan' value with this parameter is particularly useful for viewing stereo anaglyphs.

Performance improvements

The performance of the following functions has improved by taking advantage of hardware optimizations and multicore capabilities:

• adapthisteq
• histeq
• imrotate
• intlut

The dicomanon, dicominfo, dicomlookup, dicomread, dicomwrite functions have optimized implementations.

New Example

Detect and measure circular objects in an image (ipexcircles).
R2012a

Version: 8.0

New Features

Bug Fixes

Compatibility Considerations
Intensity-Based Image Registration

The new imregister function lets you automatically align two images using intensity values, even when the images were created by two different devices (multimodal). With intensity-based registration, you do not need to specify control points.

You use the new imregconfig function to create the optimizer and the metric that imregister uses to specify the desired registration parameters.

Two New Functions to Visually Compare Images

The toolbox includes two new functions for visually comparing images: imshowpair and imfuse.

imshowpair creates a composite of two images and displays them in a figure.

imfuse creates a composite of two images and returns a third image that is a numeric matrix containing a fused version of the original images.

Circle Detection Using the Circular Hough Transform

The new imfindcircles function uses the Hough transform to find circular elements in grayscale, RGB, or binary images. To view the circles that have been detected, overlaid on the original image, use the viscircles function.

Performance Improvements

The performance of the imlincomb function has improved by taking advantage of multicore capabilities.

New and Updated Demos

The toolbox includes these new and updated demos.

• Registering Multimodal MRI Images (ipexknee)
• Finding the Rotation and Scale of a Distorted Image (ipexrotate)
• Measuring the Radius of a Roll of Tape (ipexradius) – Updated to use the new imfindcircles function
Data Type Change to Output Variable for bwdist

The bwdist function returns two output variables: the Euclidean distance transform and the closest-pixel map. The data type of the second output variable, the closest-pixel map, has changed.

Compatibility Considerations

Before R2012a, the data type of the second output variable returned by the bwdist function was single. Now the data type of the second output variable returned by bwdist is dynamically chosen.

iradon Function Updated

A small imprecision in the way the dc level was computed in iradon has been corrected.

Compatibility Considerations

Output variables I and H will be slightly different (about 1% to 2%) than in previous versions.

Functions Being Removed

<table>
<thead>
<tr>
<th>Functionality</th>
<th>What Happens When You Use This Functionality?</th>
<th>Use This Instead</th>
<th>Compatibility Considerations</th>
</tr>
</thead>
<tbody>
<tr>
<td>iptcheckinput</td>
<td>Still works but issues a warning</td>
<td>validateattributes</td>
<td>Replace all existing instances of iptcheckinput with validateattributes.</td>
</tr>
<tr>
<td>iptcheckstrs</td>
<td>Still works but issues a warning</td>
<td>validatestring</td>
<td>Replace all existing instances of iptcheckstrs with validatestring.</td>
</tr>
<tr>
<td>iptchecknargin</td>
<td>Still works but issues a warning</td>
<td>narginchk</td>
<td>Replace all existing instances of iptchecknargin with narginchk.</td>
</tr>
</tbody>
</table>
R2011b

Version: 7.3

New Features

Bug Fixes

Compatibility Considerations
Parallel Block Processing Now Possible with blockproc

If you have Parallel Computing Toolbox, you can now use a new option in blockproc to improve performance of block processing tasks. Set the 'UseParallel' argument to true to use this option.

New bwdistgeodesic Function Computes Geodesic Distance Transform

Use bwdistgeodesic to compute the geodesic distance transform for a binary image.

New graydist Function Computes Gray-Weighted Distance Transform

Use graydist to compute the gray-weighted distance transform of a grayscale image.

New imapplymatrix Function Computes Linear Combination of Color Channels

The new imapplymatrix function applies a weighted sum to the color planes of an image for use in color space conversions.

Performance Improvements

Faster Functions

- imhist for large images
- rgb2gray

hdrread Now Correct for Small Values

The hdrread function now returns correct small values. Specifically, the special case value of (0,0,0,0) now maps to (0,0,0).

Compatibility Considerations

Before R2011b, the value (0,0,0) mapped to (0.5740E-41, 0.5740E-41, 0.5740E-41).
Change in Behavior for dicomwrite

If you use the dicomwrite function with the 'CreateMode' option set to 'Create' and pass in a data structure that contains the 'InversionTime' tag, you will always receive 'InversionTime' in your output.

Compatibility Considerations

Before R2011b, if your input structure contained the 'InversionTime' field, the DICOM file may or may not contain the 'InversionTime' field. The inclusion of 'InversionTime' depended on other parameters.

Warning and Error ID Changes

Many warning and error IDs have changed from their previous versions. These warnings or errors typically appear during a function call.

Compatibility Considerations

If using warning or error IDs, you might need to change the strings you use. For example, if you turned off a warning for a certain ID, the warning might now appear under a different ID. If you use a try/catch statement in your code, replace the old identifier with the new identifier. There is no definitive list of the differences, or of the IDs that changed.

Functions and Function Elements Being Removed

<table>
<thead>
<tr>
<th>Functionality</th>
<th>What Happens When You Use This Functionality?</th>
<th>Use This Instead</th>
<th>Compatibility Considerations</th>
</tr>
</thead>
<tbody>
<tr>
<td>edge function — K-direction syntax</td>
<td>Errors</td>
<td>BW = edge(..., direction)</td>
<td>The syntax BW = edge(...,K) has been removed. Use the BW = edge(...,direction) syntax instead.</td>
</tr>
<tr>
<td>Functionality</td>
<td>What Happens When You Use This Functionality?</td>
<td>Use This Instead</td>
<td>Compatibility Considerations</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------</td>
<td>------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>edge function — marr-hildreth syntax</td>
<td>Errors</td>
<td><code>edge(I, 'marr-hildreth', ...)</code></td>
<td>The syntax <code>edge(I, 'marr-hildreth', ...)</code> has been removed. Use the `edge(I, 'log', ...) syntax instead.</td>
</tr>
<tr>
<td><strong>imfeature</strong></td>
<td>Errors</td>
<td><code>regionprops</code></td>
<td><code>imfeature</code> has been removed. Use <code>regionprops</code> instead.</td>
</tr>
<tr>
<td>immovie — immovie(D,size)</td>
<td>Errors</td>
<td><code>immovie(X,map)</code></td>
<td><code>immovie(D,size)</code> is an obsolete syntax and is no longer supported. Use <code>immovie(X,map)</code> instead.</td>
</tr>
<tr>
<td>imrotate function — [R,G,B] output syntax</td>
<td>Errors</td>
<td><code>RGB2 = imrotate(RGB1)</code></td>
<td>The syntax <code>[R,G,B] = imrotate(RGB)</code> has been removed. Use the <code>RGB2 = imrotate(RGB1)</code> syntax instead.</td>
</tr>
<tr>
<td>imrotate — no output argument syntax</td>
<td>Starting in R2011b, no output argument syntax returns result in ans.</td>
<td>Call <code>imshow</code> to display the output of imrotate</td>
<td>Replacement is to call <code>imshow</code> to display the output of <code>imrotate</code>, like this: <code>B = imrotate(A,30); imshow(B)</code></td>
</tr>
<tr>
<td>Functionality</td>
<td>What Happens When You Use This Functionality?</td>
<td>Use This Instead</td>
<td>Compatibility Considerations</td>
</tr>
<tr>
<td>--------------------------------------------------------</td>
<td>-----------------------------------------------</td>
<td>-------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>imshow function — imshow(..., DISPLAY_OPTION)</td>
<td>Errors</td>
<td>imshow(..., 'InitialMagnification', 100)</td>
<td>• The syntax imshow(...,'truesize') has been removed. Use the imshow(...,'InitialMagnification',100) syntax instead.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>or</td>
<td>• The syntax imshow(...,'notruesize') has been removed. Use the imshow(...,'InitialMagnification','fit') syntax instead.</td>
</tr>
<tr>
<td>imshow function — imshow(x,y,...) syntax</td>
<td>Errors</td>
<td>imshow(...,'XData',x,'YData',y)</td>
<td>The syntax imshow(x,y,...) has been removed. Use the imshow(...,'XData',x,'YData',y) syntax instead.</td>
</tr>
<tr>
<td>imshow function — imshow(I,N) syntax</td>
<td>Errors</td>
<td>Not applicable</td>
<td>The syntax imshow(I,N) has been removed. Your grayscale image will be displayed using 256 shades of gray.</td>
</tr>
<tr>
<td>imview</td>
<td>Errors</td>
<td>imtool</td>
<td>imview has been removed. Use imtool instead.</td>
</tr>
<tr>
<td>Functionality</td>
<td>What Happens When You Use This Functionality?</td>
<td>Use This Instead</td>
<td>Compatibility Considerations</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------------</td>
<td>------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>iptsetpref function — 'ImshowTruesize' preference</td>
<td>Errors</td>
<td>'ImshowInitialMagnification' preference</td>
<td>Replace all existing instances of 'ImshowTruesize' with 'ImshowInitialMagnification' preference.</td>
</tr>
<tr>
<td>iptsetpref function — 'ImviewInitialMagnification' preference</td>
<td>Errors</td>
<td>'ImtoolInitialMagnification' preference</td>
<td>Replace all existing instances of 'ImviewInitialMagnification' with 'ImtoolInitialMagnification'.</td>
</tr>
<tr>
<td>Any calls to iptsetpref from within a startup.m file</td>
<td>It will do nothing.</td>
<td>Any calls to iptsetpref in any session are persistent.</td>
<td>You can safely remove calls to iptsetpref from startup.m files.</td>
</tr>
<tr>
<td>isbw</td>
<td>Errors</td>
<td>Not applicable</td>
<td>isbw has been removed. No replacement.</td>
</tr>
<tr>
<td>isgray</td>
<td>Errors</td>
<td>Not applicable</td>
<td>isgray has been removed. No replacement.</td>
</tr>
<tr>
<td>isind</td>
<td>Errors</td>
<td>Not applicable</td>
<td>isind has been removed. No replacement.</td>
</tr>
<tr>
<td>isrgb</td>
<td>Errors</td>
<td>Not applicable</td>
<td>isrgb has been removed. No replacement.</td>
</tr>
<tr>
<td>medfilt2 function — medfilt2(A,[M N],[Mb Nb],...) syntax</td>
<td>Errors</td>
<td>Not applicable</td>
<td>No replacement.</td>
</tr>
<tr>
<td>montage — montage(D,[M N P]) syntax</td>
<td>Errors</td>
<td>Not applicable</td>
<td>Syntax has been removed. No replacement.</td>
</tr>
<tr>
<td>Functionality</td>
<td>What Happens When You Use This Functionality?</td>
<td>Use This Instead</td>
<td>Compatibility Considerations</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------</td>
<td>------------------</td>
<td>----------------------------------------------------------------</td>
</tr>
<tr>
<td>uintlut</td>
<td>Errors</td>
<td>intlut</td>
<td>uintlut has been removed. Use intlut instead.</td>
</tr>
<tr>
<td>wiener2 function — wiener2(I, [m n],[mblock nblock],...) syntax</td>
<td>Errors</td>
<td>wiener2(I,[m n]) or wiener2(I,[m n],noise)</td>
<td>The syntax wiener2(I,[m n],[mblock nblock]) has been removed. Use the wiener2(I, [m n]) syntax instead. The syntax wiener2(I, [m n],[mblock nblock],noise) has been removed. Use the wiener2(I, [m n],noise) syntax instead.</td>
</tr>
</tbody>
</table>
R2011a

Version: 7.2

New Features

Bug Fixes

Compatibility Considerations
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:

```matlab
% 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 \texttt{watershed} was class \texttt{double} in previous releases, and is now an unsigned integer class.

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

\begin{verbatim}
L = watershed(A);
L = double(L);
\end{verbatim}

\textbf{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.

\textbf{Plot Selector Now Includes \texttt{implay}}

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

\textbf{Support for Code Generation from MATLAB}

You can now generate standalone C code for two Image Processing Toolbox functions: \texttt{label2rgb} and \texttt{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.

\textbf{edge Function No Longer Smooths Image Twice}

In previous releases, the implementation of the Canny filter, called with the \texttt{canny} method of the \texttt{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:

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

**Functions and Function Elements Being Removed**

<table>
<thead>
<tr>
<th>Function or Function Element</th>
<th>What Happens When You Use This Function or Element?</th>
<th>Use This Instead</th>
<th>Compatibility Considerations</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ipttable</code></td>
<td>Errors</td>
<td><code>uitable</code></td>
<td>Replace all existing instances of <code>ipttable</code> with <code>uitable</code>.</td>
</tr>
</tbody>
</table>
R2010b

Version: 7.1

New Features

Bug Fixes

Compatibility Considerations
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 Working with Data in Unsupported Formats 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

<table>
<thead>
<tr>
<th>Function or Function Element Name</th>
<th>What Happens When You Use This Function or Element?</th>
<th>Use This Instead</th>
<th>Compatibility Considerations</th>
</tr>
</thead>
<tbody>
<tr>
<td>blkproc</td>
<td>Still runs</td>
<td>blockproc</td>
<td>None</td>
</tr>
</tbody>
</table>
R2010a

Version: 7.0

New Features

Bug Fixes

Compatibility Considerations
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 (\texttt{uint8}, \texttt{uint16}, and \texttt{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 \texttt{int64} and \texttt{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.
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 example below demonstrate how to update your code from blkproc to blockproc.

% 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);

Here’s another example showing how to update your code from blkproc to blockproc.

% 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

<table>
<thead>
<tr>
<th>Function and Syntax</th>
<th>What Happens When You Use the Function or Element?</th>
<th>Use This Instead</th>
<th>Compatibility Considerations</th>
</tr>
</thead>
</table>

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>'ThetaResolution'</td>
<td>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', ( n_{\theta} = 2 \cdot \text{ceil}(90/\text{ThetaResolution}) ). ( \theta ) angle values are in the range ([-90, 90)) degrees. If ( 90/\text{ThetaResolution} ) is not an integer, the actual angle spacing is ( 90/\text{ceil}(90/\text{ThetaResolution}) ).</td>
</tr>
</tbody>
</table>

### 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
• incomplement
• imdilate
• imerode
• imfilter
• improfile
• imrotate
Multithreaded Functions

- applylut
- bwpack
- bwunpack
- imdilate
- imerode
- imreconstruct
R2009a

Version: 6.3

New Features

Bug Fixes

Compatibility Considerations
**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:

```matlab
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:

  ```matlab
  [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.
R2008b

Version: 6.2

New Features

Bug Fixes

Compatibility Considerations
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 imsrotatepanel 'PreserveView' Parameter Now Works for Images of All Sizes

The replaceImage function in the imsrotatepanel 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 Considerations

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 Considerations

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 Considerations

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.

<table>
<thead>
<tr>
<th>R2008a Code</th>
<th>R2008b Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>table =</td>
<td>table =</td>
</tr>
<tr>
<td>ipttable(parent, cell_array_data); uitable(parent, 'Data', cell_array_data);</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>R2008a Code</th>
<th>R2008b Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>table =</td>
<td>field_names = fieldnames(struct_data); values = struct2cell(struct_data);</td>
</tr>
</tbody>
</table>
### R2008a Code

<table>
<thead>
<tr>
<th>R2008b Code</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>for idx = 1:numel(values)</code></td>
</tr>
<tr>
<td><code>val = values{idx};</code></td>
</tr>
<tr>
<td>`if ~ischar(val)</td>
</tr>
<tr>
<td><code>values{idx} = evalc('disp(values{idx})');</code></td>
</tr>
<tr>
<td><code>end</code></td>
</tr>
<tr>
<td><code>end</code></td>
</tr>
<tr>
<td><code>table = uitable(parent,'Data', [field_names values]);</code></td>
</tr>
</tbody>
</table>

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 Considerations

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

```matlab
[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:

```matlab
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/colorspects` 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**

<table>
<thead>
<tr>
<th>Function or Demo Name</th>
<th>What Happens When You Use Function or Demo?</th>
<th>Use This Instead</th>
<th>Compatibility Considerations</th>
</tr>
</thead>
<tbody>
<tr>
<td>pixval</td>
<td>Errors</td>
<td>Use <code>impixelinfo</code> for pixel reporting and <code>imdistline</code> for measuring distance.</td>
<td>Replace all existing instances of <code>pixval</code> with <code>impixelinfo</code> or <code>imdistline</code>.</td>
</tr>
<tr>
<td>dctdemo</td>
<td>Errors</td>
<td>NA</td>
<td>None</td>
</tr>
<tr>
<td>edgedemo</td>
<td>Errors</td>
<td>NA</td>
<td>None</td>
</tr>
<tr>
<td>firdemo</td>
<td>Errors</td>
<td>NA</td>
<td>None</td>
</tr>
<tr>
<td>landsatdemo</td>
<td>Errors</td>
<td>NA</td>
<td>None</td>
</tr>
<tr>
<td>nrfildemo</td>
<td>Errors</td>
<td>NA</td>
<td>None</td>
</tr>
<tr>
<td>qtdemo</td>
<td>Errors</td>
<td>NA</td>
<td>None</td>
</tr>
<tr>
<td>roidemo</td>
<td>Errors</td>
<td>NA</td>
<td>None</td>
</tr>
</tbody>
</table>
R2008a

Version: 6.1

New Features

Bug Fixes

Compatibility Considerations
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 Considerations

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 Considerations**

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 Considerations**

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.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description of Enhancement</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>imageinfo</code></td>
<td>Accepts files of several additional file formats as an input argument, including NITF, Interfile, and Analyze file formats.</td>
</tr>
<tr>
<td><code>imshow</code></td>
<td>Supports a new colormap parameter for specifying a colormap for grayscale images.</td>
</tr>
<tr>
<td><code>imtool</code></td>
<td>Supports a new colormap parameter for specifying a colormap for grayscale images.</td>
</tr>
<tr>
<td><code>truesize</code></td>
<td>Preserves the border preference setting of the figure when adjusting the image display size.</td>
</tr>
</tbody>
</table>
R2007b

Version: 6.0

New Features

Bug Fixes

Compatibility Considerations
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,
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 Considerations**

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 Considerations**

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.
<table>
<thead>
<tr>
<th>Function</th>
<th>Description of Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>imshow</td>
<td>Is not supported when MATLAB is started with the -nojvm option.</td>
</tr>
<tr>
<td>imhist</td>
<td>Can now be embedded in custom GUIs.</td>
</tr>
<tr>
<td>fanbeam, ifanbeam, fan2para, para2fan</td>
<td>The fan-beam functions now return different answers than in previous releases due to a bug fix.</td>
</tr>
<tr>
<td>imadjdemo</td>
<td>This demo has been deleted from the toolbox.</td>
</tr>
</tbody>
</table>
Version: 5.4

New Features

Bug Fixes

Compatibility Considerations
Enhancements to `imresize` Function

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

Compatibility Considerations

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 Considerations

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 Considerations

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.

`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').

Compatibility Considerations

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` 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.
Compatibility Considerations

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.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description of Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>imshow</td>
<td>New 'border' parameter, to control whether imshow includes a border around the image</td>
</tr>
<tr>
<td></td>
<td>displayed, and 'parent' parameter, to specify the axes in which to display the image.</td>
</tr>
<tr>
<td>imscrollpanel</td>
<td>New 'replaceImage' parameter lets you replace the image displayed in the scroll panel with a new image.</td>
</tr>
<tr>
<td>iradon</td>
<td>New 'none' value for the filter parameter returns an unfiltered backprojection; also supports new interpolation types.</td>
</tr>
<tr>
<td>iptsetpref</td>
<td>New UseIPPL preference.</td>
</tr>
</tbody>
</table>
R2006b

Version: 5.3

New Features

Bug Fixes

Compatibility Considerations
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 enhancements 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.
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.

Functions cp2tform, tforminv, imtransform

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.
Compatibility Considerations

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.
R14SP3

Version: 5.1

New Features

Bug Fixes

Compatibility Considerations
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.

Obsolete and Deleted Functions

Compatibility Considerations

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

<table>
<thead>
<tr>
<th>Function</th>
<th>Enhancement</th>
</tr>
</thead>
<tbody>
<tr>
<td>impositionrect</td>
<td>This function is obsolete. Use imrect to perform the same tasks.</td>
</tr>
<tr>
<td>pixval</td>
<td>This function is obsolete. It now issues a warning when used. Use impixelinfo for pixel reporting and use imdistline for measuring distance</td>
</tr>
</tbody>
</table>

Image Tool is Not Compilable

Compatibility Considerations

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

Compatibility 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.
<table>
<thead>
<tr>
<th>Function</th>
<th>Bug Fixes</th>
</tr>
</thead>
<tbody>
<tr>
<td>dicomread</td>
<td>No longer errors when reading files that contain extraneous pixel data; instead, <code>dicomread</code> issues a warning message. However, if the file does not contain enough pixel data, <code>dicomread</code> issues an error.</td>
</tr>
<tr>
<td>dicomwrite</td>
<td>• No longer is case sensitive when parsing input parameters. For example, you can specify either 'CreateMode' or 'createmode'.</td>
</tr>
<tr>
<td></td>
<td>• Preserves the full precision of data converted to decimal string metadata. Previously, <code>dicomwrite</code> limited precision to six digits.</td>
</tr>
<tr>
<td></td>
<td>• 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), <code>dicomwrite</code> 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 <code>dicomwrite</code> writes the file with explicit value representation (VR), the file might have a different VR value, but the data will be the same.</td>
</tr>
<tr>
<td></td>
<td>• Includes the <code>TriggerTime</code> field for additional values of <code>ScanOptions</code>, including 'CT'. Previously, <code>dicomwrite</code> only included the <code>TriggerTime</code> attribute if the <code>ScanOptions</code> field indicated a gated heart MR.</td>
</tr>
<tr>
<td></td>
<td>• No longer issues an <code>Unsupported SOP class</code> error message if, when 'create' mode is specified, semantic verification is not available for an information object. Instead, <code>dicomwrite</code> 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, <code>dicomwrite</code> only performs syntactic checking, not semantic verification. Consequently, <code>dicomwrite</code> can write many more types of DICOM files in 'copy' mode than it can in 'create' mode. See the <code>dicomwrite</code> reference page for important information about data integrity.</td>
</tr>
</tbody>
</table>

### 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.

<table>
<thead>
<tr>
<th>Function</th>
<th>Enhancement</th>
</tr>
</thead>
<tbody>
<tr>
<td>applycform</td>
<td>Now correctly handles profiles that contain a gamut tag.</td>
</tr>
<tr>
<td>cpcorr</td>
<td>Now is more numerically robust. For this release, the subfunction findpeak, which cpcorr calls, has been</td>
</tr>
</tbody>
</table>
### Function | Enhancement
--- | ---
 | improved and is now a private function, rather than a subfunction.
   | No longer causes a docked figure window to become undocked.
   | 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.
   | 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.
   | Now works correctly with binary images.
   | Now returns a correct output image when called with the syntax
     ```matlab
     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.
  ```

### New Directory Needed
The Image Processing Toolbox software now requires the following new directory on the MATLAB path:

```
toolbox\shared\imageslib
```