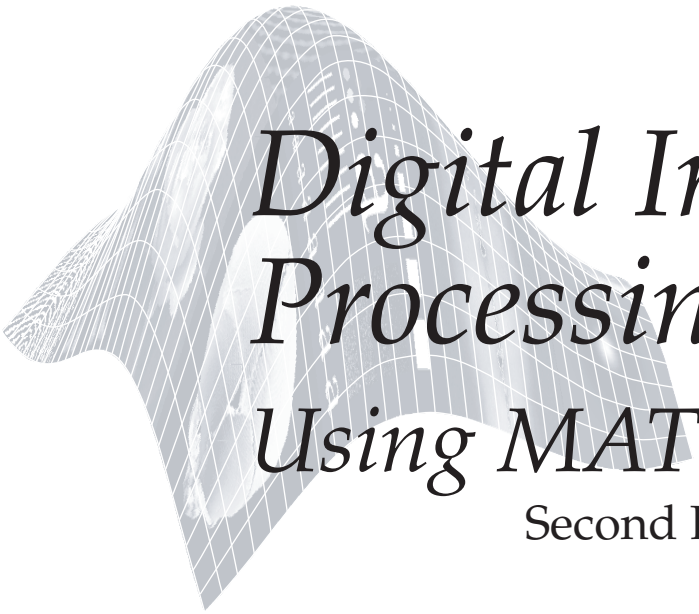

Digital Image Processing Using MATLAB®

Second Edition

Rafael C. Gonzalez
University of Tennessee

Richard E. Woods
MedData Interactive

Steven L. Eddins
The MathWorks, Inc.

Gatesmark Publishing®
A Division of Gatesmark,® LLC
www.gatesmark.com

Library of Congress Cataloging-in-Publication Data on File

Library of Congress Control Number: 2009902793

Gatesmark Publishing
A Division of Gatesmark, LLC
www.gatesmark.com

© 2009 by Gatesmark, LLC

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, without written permission from the publisher.

Gatesmark Publishing® is a registered trademark of Gatesmark, LLC, www.gatesmark.com.

Gatesmark® is a registered trademark of Gatesmark, LLC, www.gatesmark.com.

MATLAB® is a registered trademark of The MathWorks, Inc., 3 Apple Hill Drive, Natick, MA 01760-2098

The authors and publisher of this book have used their best efforts in preparing this book. These efforts include the development, research, and testing of the theories and programs to determine their effectiveness. The authors and publisher shall not be liable in any event for incidental or consequential damages with, or arising out of, the furnishing, performance, or use of these programs.

Printed in the United States of America
10 9 8 7 6 5 4 3 2 1

ISBN 978-0-9820854-0-0

APPENDIX A

M-Function Summary

Preview

Section A.1 of this appendix contains a listing by name of all the functions in the Image Processing Toolbox, and all the new (custom) functions developed in the preceding chapters. The latter functions are referred to as *DIPUM* functions, a term derived from the first letter of the words in the title of the book. Section A.2 lists the MATLAB functions used throughout the book. All page numbers listed refer to pages in the book, indicating where a function is first used and illustrated. In some instances, more than one location is given, indicating that the function is explained in different ways, depending on the application. Use of a gray dash "—" in the page reference indicates a toolbox function not used in the book; information about them can be obtained in the product documentation. All MATLAB functions listed in Section A.2 are used in the book.

Each page number in that section identifies the first use of the MATLAB function indicated.
The following functions are grouped loosely in categories similar to those found in Image Processing Toolbox documentation. A new category (e.g., wavelets) was created in cases for which no toolbox category exists (e.g., wavelets).

A.1 Image Processing Toolbox and DIPUM Functions

The following functions are grouped loosely into categories similar to those found in Image Processing Toolbox documentation.

Function category and Name	Description	Pages
Image display and exploration		
`ice` (DIPUM)	Interactive Color Editor.	352, 727
`immovie`	Make movie from multiframe image.	—
`implay`	Play movies, videos, or image sequences.	407, 474
`imshow`	Display image in Handle Graphics figure.	18

`imtool`	Display image in the Image Tool.	19
`montage`	Display multiple image frames as rectangular montage.	474
`rgbcube` (DIPUM)	Displays an RGB cube on the MATLAB desktop.	319
`subimage`	Display multiple images in single figure.	—
`warp`	Display image as texture-mapped surface	—

Image file I/O

`analyze75info`	Read metadata from header file of Mayo Analyze 7.5 data set.	—
`analyze75read`	Read image file of Mayo Analyze 7.5 data set.	—
`dicomanon`	Anonymize DICOM file.	—
`dicomdict`	Get or set active DICOM data dictionary.	—
`dicominfo`	Read metadata from DICOM message.	—
`dicomlookup`	Find attribute in DICOM data dictionary.	—
`dicomread`	Read DICOM image.	—
`dicomuid`	Generate DICOM Unique Identifier.	—
`dicomwrite`	Write images as DICOM files.	—
`hdrread`	Read Radiance HDR image.	—
`hdrwrite`	Write Radiance HDR image.	—
`makehdr`	Create high dynamic range image.	—
`interfileinfo`	Read metadata from Interfile files.	—
`interfileread`	Read images from Interfile files.	—
`isnitf`	Check if file is NITF.	—
`movie2tifs` (DIPUM)	Creates a multiframe TIFF file from a MATLAB movie.	475
`nitfinfo`	Read metadata from NITF file.	—
`nitfread`	Read NITF image.	—
`seq2tifs` (DIPUM)	Creates a multi-frame TIFF file from a MATLAB sequence.	475
`tifs2movie` (DIPUM)	Create a MATLAB movie from a multiframe TIFF file.	475
`tifs2seq` (DIPUM)	Create a MATLAB sequence from a multi-frame TIFF file.	475

Image arithmetic

`imabsdiff`	Absolute difference of two images.	—
`imcomplement`	Complement image.	83, 331
`imlincomb`	Linear combination of images.	50
`ippl`	Check for presence of Intel Performance Primitives Library (IPPL).	—

Geometric transformations

`checkerboard`	Create checkerboard image.	238
`findbounds`	Find output bounds for spatial transformation.	—
`fliptform`	Flip input and output roles of TFORM structure.	—
`imcrop`	Crop image.	—
`impyramid`	Image pyramid reduction and expansion.	—
`imresize`	Resize image.	—
`imrotate`	Rotate image.	291, 659
`imtransform`	Apply 2-D spatial transformation to image.	289
`imtransform2` (DIPUM)	2-D image transformation with fixed output location.	298
`makeresampler`	Create resampling structure.	—
`maketform`	Create spatial transformation structure (TFORM).	279, 309
`pixeldup` (DIPUM)	Duplicates pixels of an image in both directions.	238
`pointgrid` (DIPUM)	Points arranged on a grid.	282

`reprotate` (DIPUM)	Rotate image repeatedly.	303
`tformarray`	Apply spatial transformation to N-D array.	—
`tformfwd`	Apply forward spatial transformation.	281
`tforminv`	Apply inverse spatial transformation.	281
`vistform` (DIPUM)	Visualization transformation effect on set of points.	283

Image registration

`cpstruct2pairs`	Convert CPSTRUCT to control point pairs.	—
`cp2tform`	Infer spatial transformation from control point pairs.	307
`cpcorr`	Tune control point locations using cross-correlation.	—
`cpselect`	Control Point Selection Tool.	306
`normxcorr2`	Normalized two-dimensional cross-correlation.	313, 683
`visreg` (DIPUM)	Visualize registered images.	308

Pixel values and statistics

`corr2`	2-D correlation coefficient.	—
`imcontour`	Create contour plot of image data.	—
`imhist`	Display histogram of image data.	94
`impixel`	Pixel color values.	—
`improfile`	Pixel-value cross-sections along line segments.	—
`localmean` (DIPUM)	Computes an array of local means.	572
`mean2`	Average or mean of matrix elements.	76, 92
`regionprops`	Measure properties of image regions (blob analysis).	642
`statmoments` (DIPUM)	Computes statistical central moments of image histogram.	225
`std2`	Standard deviation of matrix elements.	—

Image analysis

`bayesgauss` (DIPUM)	Bayes classifier for Gaussian patterns.	685
`bound2eight` (DIPUM)	Convert 4-connected boundary to 8-connected boundary.	605
`bound2four` (DIPUM)	Convert 8-connected boundary to 4-connected boundary.	605
`bound2im` (DIPUM)	Converts a boundary to an image.	600
`bsubsamp` (DIPUM)	Subsample a boundary.	605
`bwboundaries` (DIPUM)	Trace region boundaries in binary image.	599
`bwtraceboundary`	Trace object in binary image.	—
`colorgrad` (DIPUM)	Computes the vector gradient of an RGB image.	369
`colorseg` (DIPUM)	Performs segmentation of a color image.	373
`connectpoly` (DIPUM)	Connects vertices of a polygon.	605
`cornermetric`	Create corner metric matrix from image.	638
`cornerprocess` (DIPUM)	Processes the output of function `cornermetric`.	638
`diameter` (DIPUM)	Measure diameter and related properties of image regions.	626
`edge`	Find edges in intensity image.	542
`fchcode` (DIPUM)	Computes the Freeman chain code of a boundary.	607
`frdescp` (DIPUM)	Computes Fourier descriptors.	629
`ifrdescp` (DIPUM)	Computes inverse Fourier descriptors.	629
`im2minperpoly` (DIPUM)	Minimum perimeter polygon.	617
`imstack2vectors` (DIPUM)	Extracts vectors from an image stack.	663
`invmoments` (DIPUM)	Compute invariant moments of image.	658
`hough`	Hough transform.	553
`houghlines`	Extract line segments based on Hough transform.	555

`houghpeaks`	Identify peaks in Hough transform.	555
`localthresh` (DIPUM)	Local thresholding.	573
`mahalanobis` (DIPUM)	Computes the Mahalanobis distance.	678
`movingthresh` (DIPUM)	Image segmentation using a moving average threshold.	576
`otsuthresh` (DIPUM)	Otsu's optimum threshold given a histogram.	564
`polyangles` (DIPUM)	Computes internal polygon angles.	704
`principalcomps` (DIPUM)	Principal-component vectors and related quantities.	664
`qtdecomp`	Quadtree decomposition.	584
`qtgetblk`	Get block values in quadtree decomposition.	584
`qtsetblk`	Set block values in quadtree decomposition.	—
`randvertex` (DIPUM)	Adds random noise to the vertices of a polygon.	704
`regiongrow` (DIPUM)	Perform segmentation by region growing.	580
`signature` (DIPUM)	Computes the signature of a boundary.	620
`specxture` (DIPUM)	Computes spectral texture of an image.	655
`splitmerge` (DIPUM)	Segment an image using a split-and-merge algorithm.	585
`statxture` (DIPUM)	Computes statistical measures of texture in an image.	645
`strsimilarity` (DIPUM)	Computes a similarity measure between two strings.	701
`x2majoraxis` (DIPUM)	Aligns coordinate x with the major axis of a region.	628

Image compression

`compare` (DIPUM)	Computes and displays the error between two matrices.	423
`cv2tifs` (DIPUM)	Decodes a `TIFS2CV` compressed image sequence.	483
`huff2mat` (DIPUM)	Decodes a Huffman encoded matrix.	440
`huffman` (DIPUM)	Builds a variable-length Huffman code for a symbol source.	429
`im2jpeg` (DIPUM)	Compresses an image using a JPEG approximation.	457
`im2jpeg2k` (DIPUM)	Compresses an image using a JPEG 2000 approximation.	466
`imratio` (DIPUM)	Computes the ratio of the bytes in two images/variables.	421
`jpeg2im` (DIPUM)	Decodes an IM2JPEG compressed image.	461
`jpeg2k2im` (DIPUM)	Decodes an IM2JPEG2K compressed image.	469
`lpc2mat` (DIPUM)	Decompresses a 1-D lossless predictive encoded matrix.	451
`mat2huff` (DIPUM)	Huffman encodes a matrix.	436
`mat2lpc` (DIPUM)	Compresses a matrix using 1-D lossles predictive coding.	450
`ntrop` (DIPUM)	Computes a first-order estimate of the entropy of a matrix.	426
`quantize` (DIPUM)	Quantizes the elements of a UINT8 matrix.	454
`showmo` (DIPUM)	Displays the motion vectors of a compressed image sequence.	483
`tifs2cv` (DIPUM)	Compresses a multi-frame TIFF image sequence.	480
`unravel` (DIPUM)	Decodes a variable-length bit stream.	442

Image deblurring

`deconvblind`	Deblur image using blind deconvolution.	250
`deconvlucy`	Deblur image using Lucy-Richardson method.	248
`deconvreg`	Deblur image using regularized filter.	245
`deconvwnr`	Deblur image using Wiener filter.	241
`edgetaper`	Taper edges using point-spread function.	242
`otf2psf`	Convert optical transfer function to point-spread function.	—
`psf2otf`	Convert point-spread function to optical transfer function.	—

Image enhancement

`adapthisteq`	Contrast-limited Adaptive Histogram Equalization (CLAHE).	107
`adpmedian` (DIPUM)	Perform adaptive median filtering.	235

`decorrstretch`	Apply decorrelation stretch to multichannel image.	—
`gscale` (DIPUM)	Scales the intensity of the input image.	92
`histeq`	Enhance contrast using histogram equalization.	100
`imadjust`	Adjust image intensity values or color map.	82
`medfilt2`	2-D median filtering.	—
`ordfilt2`	2-D order-statistic filtering.	125
`stretchlim`	Find limits to contrast stretch an image.	84
`intlut`	Convert integer values using lookup table.	—
`intrans` (DIPUM)	Performs intensity (gray-level) transformations.	89
`wiener2`	2-D adaptive noise-removal filtering.	—

Image noise

`imnoise`	Add noise to image.	126, 211
`imnoise2` (DIPUM)	Generates an array of random numbers with specified PDF.	216
`imnoise3` (DIPUM)	Generates periodic noise.	221

Linear filtering

`convmtx2`	2-D convolution matrix.	—
`dftfilt` (DIPUM)	Performs frequency domain filtering.	179
`fspecial`	Create predefined 2-D filters.	120
`imfilter`	N-D filtering of multidimensional images.	114
`spfilt` (DIPUM)	Performs linear and nonlinear spatial filtering.	229

Linear 2-D filter design

`bandfilter` (DIPUM)	Computes frequency domain band filters.	199
`cnotch` (DIPUM)	Generates circularly symmetric notch filters.	203
`freqz2`	2-D frequency response.	181
`fsamp2`	2-D FIR filter using frequency sampling.	—
`ftrans2`	2-D FIR filter using frequency transformation.	—
`fwind1`	2-D FIR filter using 1-D window method.	—
`fwind2`	2-D FIR filter using 2-D window method.	—
`hpfilter` (DIPUM)	Computes frequency domain highpass filters.	195
`lpfilter` (DIPUM)	Computes frequency domain lowpass filters.	175,189
`recnotch` (DIPUM)	Generates rectangular notch (axes) filters.	205

Fuzzy logic

`aggfcn` (DIPUM)	Aggregation function for a fuzzy system.	149
`approxfcn` (DIPUM)	Approximation function.	152
`bellmf` (DIPUM)	Bell-shaped membership function.	145
`defuzzify` (DIPUM)	Output of fuzzy system.	149
`fuzzyfilt` (DIPUM)	Fuzzy edge detector.	162
`fuzzysysfcn` (DIPUM)	Fuzzy system function.	150
`implfcns` (DIPUM)	Implication functions for a fuzzy system.	147
`lambdafcns` (DIPUM)	Lambda functions for a set of fuzzy rules.	146
`makefuzzyedgesys` (DIPUM)	Script to make MAT-file used by `FUZZYFILT`.	161
`onemf` (DIPUM)	Constant membership function (one).	145
`sigmamf` (DIPUM)	Sigma membership function.	144
`smf` (DIPUM)	S-shaped membership function.	144
`trapezmf` (DIPUM)	Trapezoidal membership function.	143

`triangmf` (DIPUM)	Triangular membership function.	143
`truncgaussmf` (DIPUM)	Truncated Gaussian membership function.	145
`zeromf` (DIPUM)	Constant membership function (zero).	145

Image transforms

`dct2`	2-D discrete cosine transform.	—
`dctmtx`	Discrete cosine transform matrix.	
`fan2para`	Convert fan-beam projections to parallel-beam.	274
`fanbeam`	Fan-beam transform.	269
`idct2`	2-D inverse discrete cosine transform.	—
`ifanbeam`	Inverse fan-beam transform.	271
`iradon`	Inverse Radon transform.	263
`para2fan`	Convert parallel-beam projections to fan-beam.	275
`phantom`	Create head phantom image.	261
`radon`	Radon transform.	260

Neighborhood and block processing

`bestblk`	Optimal block size for block processing.	—
`blkproc`	Distinct block processing for image.	459
`col2im`	Rearrange matrix columns into blocks.	460
`colfilt`	Columnwise neighborhood operations.	118
`im2col`	Rearrange image blocks into columns.	460
`nlfilter`	General sliding-neighborhood operations.	—

Morphological operations (gray scale and binary images)

`conndef`	Default connectivity array.	—
`imbothat`	Bottom-hat filtering.	529
`imclearborder`	Suppress light structures connected to image border.	521
`imclose`	Morphologically close image.	501
`imdilate`	Dilate image.	492
`imerode`	Erode image.	500
`imextendedmax`	Extended-maxima transform.	—
`imextendedmin`	Extended-minima transform.	595
`imfill`	Fill image regions and holes.	521, 603
`imhmax`	H-maxima transform.	—
`imhmin`	H-minima transform.	531
`imimposemin`	Impose minima.	596
`imopen`	Morphologically open image.	501
`imreconstruct`	Morphological reconstruction.	518
`imregionalmax`	Regional maxima.	—
`imregionalmin`	Regional minima.	593
`imtophat`	Top-hat filtering.	529
`watershed`	Watershed transform.	590

Morphological operations (binary images)

`applylut`	Neighborhood operations using lookup tables.	507
`bwarea`	Area of objects in binary image.	—
`bwareaopen`	Morphologically open binary image (remove small objects).	—
`bwdist`	Distance transform of binary image.	589

`bweuler`	Euler number of binary image.	—
`bwhitmiss`	Binary hit-miss operation.	505
`bwlabel`	Label connected components in 2-D binary image.	515
`bwlabeln`	Label connected components in N-D binary image.	—
`bwmorph`	Morphological operations on binary image.	511
`bwpack`	Pack binary image.	
`bwperim`	Find perimeter of objects in binary image.	598
`bwselect`	Select objects in binary image.	—
`bwulterode`	Ultimate erosion.	—
`bwunpack`	Unpack binary image.	—
`endpoints` (DIPUM)	Computes end points of a binary image.	507
`makelut`	Create lookup table for use with `APPLYLUT`.	507

Structuring element (STREL) creation and manipulation

`getheight`	Get STREL height.	—
`getneighbors`	Get offset location and height of STREL neighbors.	—
`getnhood`	Get STREL neighborhood.	—
`getsequence`	Get sequence of decomposed STRELs.	497
`isflat`	True for flat STRELs.	—
`reflect`	Reflect STREL about its center.	492
`strel`	Create morphological structuring element (STREL).	494
`translate`	Translate STREL.	—

Texture analysis

`entropy`	Entropy of intensity image.	—
`entropyfilt`	Local entropy of intensity image.	—
`graycomatrix`	Create gray-level co-occurrence matrix.	648
`graycoprops`	Properties of gray-level co-occurrence matrix.	649
`rangefilt`	Local range of image.	—
`specxture` (DIPUM)	Computes spectral texture of an image.	655
`statxture` (DIPUM)	Computes statistical measures of texture in an image.	645
`stdfilt`	Local standard deviation of image.	572

Region-based processing

`histroi` (DIPUM)	Computes the histogram of an ROI in an image.	227
`poly2mask`	Convert region-of-interest polygon to mask.	—
`roicolor`	Select region of interest based on color.	—
`roifill`	Fill in specified polygon in grayscale image.	—
`roifilt2`	Filter region of interest.	—
`roipoly`	Select polygonal region of interest.	225

Wavelets

`appcoef2`	Extract 2-D approximation coefficients.	398
`detcoef2`	Extract 2-D detail coefficients.	398
`dwtmode`	Discrete wavelet transform extension mode.	387
`waveback` (DIPUM)	Computes inverse FWTs for multi-level decomposition.	409
`wavecopy` (DIPUM)	Fetches coefficients of a wavelet decomposition structure.	402
`wavecut` (DIPUM)	Zeroes coefficients in a wavelet decomposition structure.	401
`wavedec2`	Multilevel 2-D wavelet decomposition.	385

`wavedisplay` (DIPUM)	Display wavelet decomposition coefficients.	404
`wavefast` (DIPUM)	Computes the FWT of a '3-D extended' 2-D array.	391
`wavefilter` (DIPUM)	Create wavelet decomposition and reconstruction filters.	388
`wavefun`	Wavelet and scaling functions 1-D.	382
`waveinfo`	Information on wavelets.	382
`waverec2`	Multilevel 2-D wavelet reconstruction.	409
`wavework` (DIPUM)	is used to edit wavelet decomposition structures.	399
`wavezero` (DIPUM)	Zeroes wavelet transform detail coefficients.	415
`wfilters`	Wavelet filters.	381
`wthcoef2`	Wavelet coefficient thresholding 2-D.	398

Colormap manipulation

`cmpermute`	Rearrange colors in color map.	—
`cmunique`	Eliminate unneeded colors in color map of indexed image.	—
`imapprox`	Approximate indexed image by one with fewer colors.	321

Color space conversions

`applycform`	Apply device-independent color space transformation.	344
`hsi2rgb` (DIPUM)	Converts an HSI image to RGB.	338
`iccfind`	Search for ICC profiles by description.	—
`iccread`	Read ICC color profile.	347
`iccroot`	Find system ICC profile repository.	—
`iccwrite`	Write ICC color profile.	—
`isicc`	True for complete profile structure.	—
`lab2double`	Convert L*a*b* color values to double.	—
`lab2uint16`	Convert L*a*b* color values to uint16.	—
`lab2uint8`	Convert L*a*b* color values to uint8.	—
`makecform`	Create device-independent color space transformation structure (CFORM).	344
`ntsc2rgb`	Convert NTSC color values to RGB color space.	329
`rgb2hsi` (DIPUM)	Converts an RGB image to HSI.	337
`rgb2ntsc`	Convert RGB color values to NTSC color space.	328
`rgb2ycbcr`	Convert RGB color values to YCbCr color space.	329
`whitepoint`	XYZ color values of standard illuminants.	—
`xyz2double`	Convert XYZ color values to double.	—
`xyz2uint16`	Convert XYZ color values to uint16.	—
`ycbcr2rgb`	Convert YCbCr color values to RGB color space.	329

Array operations

`dftuv` (DIPUM)	Computes meshgrid frequency matrices.	186
`padarray`	Pad array.	118
`paddedsize` (DIPUM)	Computes padded sizes useful for FFT-based filtering.	174

Image types and type conversions

`demosaic`	Convert Bayer pattern encoded image to a true color image.	—
`dither`	Convert image using dithering.	324
`gray2ind`	Convert intensity image to indexed image.	325
`grayslice`	Create indexed image from intensity image by thresholding.	325
`graythresh`	Global image threshold using Otsu's method.	562

`im2bw`	Convert image to binary image by thresholding.	30
`im2double`	Convert image to double precision.	29
`im2int16`	Convert image to 16-bit signed integers.	—
`im2java2d`	Convert image to Java Buffered Image.	—
`im2single`	Convert image to single precision.	29
`im2uint8`	Convert image to 8-bit unsigned integers.	29
`im2uint16`	Convert image to 16-bit unsigned integers.	29
`ind2gray`	Convert indexed image to intensity image.	325
`label2rgb`	Convert label matrix to RGB image.	—
`mat2gray`	Convert matrix to intensity image.	30
`rgb2gray`	Convert RGB image or color map to grayscale.	326
`rgb2ind`	Convert RGB image to indexed image.	325
`tofloat` (DIPUM)	Convert image to floating point.	32
`tonemap`	Render high dynamic range image for viewing.	—

Toolbox preferences

`iptgetpref`	Get value of Image Processing Toolbox preference.	—
`iptsetpref`	Set value of Image Processing Toolbox preference.	291

Toolbox utility functions

`getrangefromclass`	Get dynamic range of image based on its class.	—
`intline`	Integer-coordinate line drawing.	606
`iptcheckconn`	Check validity of connectivity argument.	—
`iptcheckinput`	Check validity of array.	—
`iptcheckmap`	Check validity of color map.	—
`iptchecknargin`	Check number of input arguments.	—
`iptcheckstrs`	Check validity of text string.	—
`iptnum2ordinal`	Convert positive integer to ordinal string.	—

Modular interactive tools

`imageinfo`	Image Information tool.	—
`imcontrast`	Adjust Contrast tool.	—
`imdisplayrange`	Display Range tool.	—
`imdistline`	Draggable Distance tool.	—
`imgetfile`	Open Image dialog box.	—
`impixelinfo`	Pixel Information tool.	—
`impixelinfoval`	Pixel Information tool without text label.	—
`impixelregion`	Pixel Region tool.	—
`impixelregionpanel`	Pixel Region tool panel.	—
`imputfile`	Save Image dialog box.	—
`imsave`	Save Image tool.	—

Navigational tools for image scroll panel

`imscrollpanel`	Scroll panel for interactive image navigation.	—
`immagbox`	Magnification box for scroll panel.	—
`imoverview`	Overview tool for image displayed in scroll panel.	—
`imoverviewpanel`	Overview tool panel for image displayed in scroll panel.	—

Utility functions for interactive tools

`axes2pix`	Convert axes coordinate to pixel coordinate.	—
`getimage`	Get image data from axes.	—
`getimagemodel`	Get image model object from image object.	—
`imagemodel`	Image model object.	—
`imattributes`	Information about image attributes.	—
`imhandles`	Get all image handles.	—
`imgca`	Get handle to current axes containing image.	—
`imgcf`	Get handle to current figure containing image.	—
`imellipse`	Create draggable, resizable ellipse.	—
`imfreehand`	Create draggable freehand region.	—
`imline`	Create draggable, resizable line.	—
`impoint`	Create draggable point.	—
`impoly`	Create draggable, resizable polygon.	—
`imrect`	Create draggable, resizable rectangle.	—
`iptaddcallback`	Add function handle to callback list.	—
`iptcheckhandle`	Check validity of handle.	—
`iptgetapi`	Get Application Programmer Interface (API) for handle.	—
`iptGetPointerBehavior`	Retrieve pointer behavior from HG object.	—
`ipticondir`	Directories containing IPT and MATLAB icons.	—
`iptPointerManager`	Install mouse pointer manager in figure.	—
`iptremovecallback`	Delete function handle from callback list.	—
`iptSetPointerBehavior`	Store pointer behavior in HG object.	—
`iptwindowalign`	Align figure windows.	—
`makeConstrainToRectFcn`	Create rectangularly bounded position constraint function.	—
`truesize`	Adjust display size of image.	—

Interactive mouse utility functions

`getline`	Select polyline with mouse.	—
`getpts`	Select points with mouse.	—
`getrect`	Select rectangle with mouse.	—

Miscellaneous functions

`conwaylaws` (DIPUM)	Applies Conway's genetic laws to a single pixel.	509
`i2percentile` (DIPUM)	Computes a percentile given an intensity value.	567
`iseven` (DIPUM)	Determines which elements of an array are even numbers.	203
`isodd` (DIPUM)	Determines which elements of an array are odd numbers.	203
`manualhist` (DIPUM)	Generates a two-mode histogram interactively.	105
`timeit` (DIPUM)	Measure time required to run function.	66
`percentile2i` (DIPUM)	Computes an intensity value given a percentile.	567
`tofloat` (DIPUM)	Converts input to single-precision floating point.	*32*
`twomodegauss` (DIPUM)	Generates a two-mode Gaussian function.	104

A.2 MATLAB Functions

The following MATLAB functions, listed alphabetically, are used in the book.

MATLAB Function	Description	Pages
A		
`abs`	Absolute value.	168
`all`	True if all elements of a vector are nonzero.	53
`angle`	Phase angle.	171
`annotation`	Creates an annotation object.	102
`ans`	Most recent answer.	55
`any`	True if any element of a vector is nonzero.	53
`atan2`	Four quadrant inverse tangent.	170
`autumn`	Shades of red and yellow color map.	324
`axis`	Control axis scaling and appearance.	96
`axis`	Control axis scaling and appearance.	191
B		
`bar`	Bar graph.	95
`base2dec`	Convert base B string to decimal integer.	693
`bin2dec`	Convert binary string to decimal integer.	438
`bin2dec`	Convert binary string to decimal integer.	693
`blanks`	String of blanks.	692
`bone`	Gray-scale with a tinge of blue color map.	324
`break`	Terminate execution of WHILE or FOR loop.	61
`bsxfun`	Binary singleton expansion function.	676
C		
`cart2pol`	Transform Cartesian to polar coordinates.	621
`cat`	Concatenate arrays.	319
`catch`	Begin CATCH block.	58
`ceil`	Round towards plus infinity.	171
`cell`	Create cell array.	431
`celldisp`	Display cell array contents.	75, 431
`cellfun`	Apply a function to each cell of a cell array.	75
`cellplot`	Display graphical depiction of cell array.	431
`cellstr`	Create cell array of strings from character array.	692
`char`	Create character array (string).	26, 73, 693
`circshift`	Shift array circularly.	605
`colon`	Colon operator (:) for forming vectors and indexing.	33
`colorcube`	Enhanced color-cube color map.	324
`colormap`	Color look-up table.	191, 323
`computer`	Computer type.	55
`continue`	Pass control to the next iteration of FOR or WHILE loop.	62
`conv2`	Two dimensional convolution.	394

`cool`	Shades of cyan and magenta color map.	324
`copper`	Linear copper-tone color map.	323
`cumsum`	Cumulative sum of elements.	101

D

`deblank`	Remove trailing blanks.	693
`dec2base`	Convert decimal integer to base B string.	700
`dec2bin`	Convert decimal integer to a binary string.	436
`dec2hex`	Convert decimal integer to hexadecimal string.	693
`diag`	Diagonal matrices and diagonals of a matrix.	374
`diff`	Difference and approximate derivative.	529
`disp`	Display array.	71
`dither`	Convert image using dithering.	323
`double`	Convert to double precision.	26

E

`edit`	Edit M-file.	46
`eig`	Eigenvalues and eigenvectors.	665
`else`	Used with IF.	58
`elseif`	IF statement condition.	58
`end`	Terminate scope of FOR, WHILE, SWITCH, TRY, and IF statements.	34
`eps`	Spacing of floating point numbers.	55
`error`	Display message and abort function.	59
`eval`	Execute string with MATLAB expression.	694
`eye`	Identity matrix.	44

F

`false`	False array.	44, 587
`fft2`	Two-dimensional discrete Fourier Transform.	168
`fftshift`	Shift zero-frequency component to center of spectrum.	169
`figure`	Create figure window.	19
`filter`	One-dimensional digital filter.	575
`find`	Find indices of nonzero elements.	215
`fix`	Round towards zero.	152
`flag`	Alternating red, white, blue, and black color map.	324
`fliplr`	Flip matrix in left/right direction.	262
`flipud`	Flip matrix in up/down direction.	262
`floor`	Round towards minus infinity.	171
`for`	Repeat statements a specific number of times.	59
`format`	Set output format.	56
`fplot`	Plot function.	98
`full`	Convert sparse matrix to full matrix.	43

G

`gca`	Get handle to current axis.	96
`gcf`	Get handle to current figure.	737
`get`	Get object properties.	56, 353

H

I

`isinf`	True for infinite elements.	54
`isinteger`	True for arrays of integer data type.	54
`isletter`	True for letters of the alphabet.	54, 693
`islogical`	True for logical array.	27
`islogical`	True for logical array.	54
`ismember`	True for set member.	54
`isnan`	True for Not-a-Number.	54
`isnumeric`	True for numeric arrays.	54
`ispc`	True for the PC (Windows) version of MATLAB.	728
`isprime`	True for prime numbers.	54
`isreal`	True for real array.	54
`isscalar`	True if array is a scalar.	54
`isspace`	True for white space characters.	54
`isspace`	True for white space characters.	693
`issparse`	True for sparse matrix.	54
`isstruct`	True for structures.	54
`isvector`	True if array is a vector.	54

J

`j`	Imaginary unit.	55
`jet`	Variant of HSV.	324

L

`length`	Length of vector.	59
`lines`	Color map with the line colors.	324
`linspace`	Linearly spaced vector.	34
`log`	Natural logarithm.	84
`log10`	Common (base 10) logarithm.	84
`log2`	Base 2 logarithm and dissect floating point number.	84
`logical`	Convert numeric values to logical.	27
`lookfor`	Search all M-files for keyword.	46
`lower`	Convert string to lowercase.	201, 693

M

`magic`	Magic square.	44
`makecounter`	Used by NESTEDDEMO.	141
`mat2str`	Convert a 2-D matrix to a string in MATLAB syntax.	699
`max`	Largest component.	48
`mean`	Average or mean value.	76, 517
`median`	Median value.	126
`mesh`	3-D mesh surface.	190
`meshgrid`	X and Y arrays for 3-D plots.	69
`mfilename`	Name of currently executing M-file.	730
`min`	Smallest component.	48
`movie2avi`	Create AVI movie from MATLAB movie.	475

N

`NaN`	Not-a-Number.	55
`nargchk`	Validate number of input arguments.	88
`nargin`	Number of function input arguments.	87
`nargout`	Number of function output arguments.	87
`ndims`	Number of dimensions.	42
`nextpow2`	Next higher power of 2.	175
`norm`	Matrix or vector norm.	675
`num2str`	Convert numbers to a string.	693
`numel`	Number of elements in an array or subscripted array expression.	59

O

`ones`	Ones array.	44

P

`permute`	Permute array dimensions.	677
`persistent`	Define persistent variable.	507
`pi`	3.1415926535897....	55
`pink`	Pastel shades of pink color map.	324
`plot`	Linear plot.	41, 98
`pol2cart`	Transform polar to Cartesian coordinates.	621
`pow2`	Base 2 power and scale floating point number.	438
`print`	Print figure or model. Save to disk as image or M-file.	25
`prism`	Prism color map.	324
`prod`	Product of elements.	119

Q

`quad`	Numerical integration based on quadratures.	56

R

`rand`	Uniformly distributed pseudorandom numbers.	44, 215
`randn`	Normally distributed pseudorandom numbers.	44, 215
`real`	Complex real part.	170
`realmax`	Largest positive floating point number.	55
`realmin`	Smallest positive normalized floating point number.	55
`regexp`	Match regular expression.	695
`regexpi`	Match regular expression, ignoring case.	696
`regexprep`	Replace string using regular expression.	696
`rem`	Remainder after division.	152
`reshape`	Change size.	401, 438
`return`	Return to invoking function.	58
`rexexpi`	NOT FOUND.	693
`rgb2hsv`	Convert red-green-blue colors to hue-saturation-value.	330
`round`	Round towards nearest integer.	25
`rot90`	Rotate matrix 90 degrees.	115

S

`set`	Set object properties.	96
`setfield`	Set structure field contents.	743
`shading`	Color shading mode.	194
`single`	Convert to single precision.	26
`size`	Size of array.	16
`sort`	Sort in ascending or descending order.	431
`sortrows`	Sort rows in ascending order.	604
`sparse`	Create sparse matrix.	42
`spline`	Cubic spline data interpolation.	352
`spring`	Shades of magenta and yellow color map.	324
`sprintf`	Write formatted data to string.	60, 693
`sscanf`	Read string under format control.	693
`stem`	Discrete sequence or "stem" plot.	96
`str2double`	Convert string to double precision value.	693
`str2num`	Convert string matrix to numeric array.	693
`strcat`	Concatenate strings.	696
`strcmp`	Compare strings.	73, 697
`strcmpi`	Compare strings ignoring case.	74, 454, 697
`strfind`	Find one string within another.	698
`strjust`	Justify character array.	698
`strmatch`	Find possible matches for string.	693
`strncmp`	Compare first N characters of strings.	697
`strncmpi`	Compare first N characters of strings ignoring case.	698
`strread`	Read formatted data from string.	73
`strread`	Read formatted data from string.	693
`strrep`	Replace string with another.	698
`strtok`	Find token in string.	699
`strvcat`	Vertically concatenate strings.	697
`sub2ind`	Linear index from multiple subscripts.	40
`subplot`	Create axes in tiled positions.	384
`sum`	Sum of elements.	37
`summer`	Shades of green and yellow color map.	324
`surf`	3-D colored surface.	193
`switch`	Switch among several cases based on expression.	62

T

`text`	Text annotation.	96
`tic`	Start a stopwatch timer.	65
`title`	Graph title.	96
`toc`	Read the stopwatch timer.	65
`transpose`	Transpose.	33
`true`	True array.	44, 587
`try`	Begin TRY block.	58

U

`uicontrol`	Create user interface control.	731
`uint16`	Convert to unsigned 16-bit integer.	26

`uint32`	Convert to unsigned 32-bit integer.	26
`uint8`	Convert to unsigned 8-bit integer.	26
`uiresume`	Resume execution of blocked M-file.	737
`uiwait`	Block execution and wait for resume.	737
`unique`	Set unique.	604
`upper`	Convert string to uppercase.	201, 693

V

`varargin`	Variable length input argument list.	88
`varargout`	Variable length output argument list.	88
`ver`	MATLAB, Simulink and toolbox version information.	55
`version`	MATLAB version number.	55
`view`	3-D graph viewpoint specification.	191

W

`waitbar`	Display wait bar.	151
`while`	Repeat statements an indefinite number of times.	61
`white`	All white color map.	324
`whitebg`	Change axes background color.	322
`whos`	List current variables, long form.	17
`winter`	Shades of blue and green color map.	324

X

`xlabel`	X-axis label.	96
`xlim`	X limits.	98
`xor`	Logical EXCLUSIVE OR.	53

Y

`ylabel`	Y-axis label.	96
`ylim`	Y limits.	98

Z

`zeros`	Zeros array.	44