

# Image Processing Toolbox 2

for use with MATLAB® 5



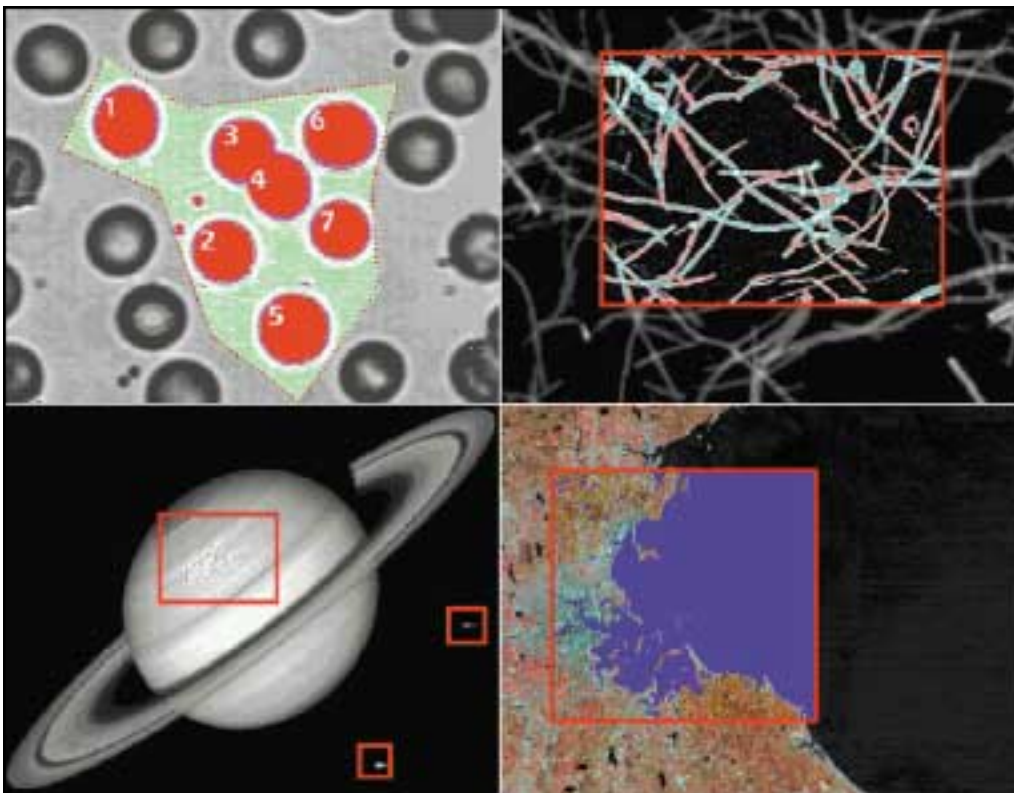
The Image Processing Toolbox is a comprehensive set of image processing and analysis tools designed to support a broad range of applications requiring image enhancement, data reduction, and analysis of digital images. The Image Processing Toolbox builds upon the MATLAB foundation to provide a unique solution that combines high-level image processing and analysis functions with a powerful and extensible technical computing environment.

## OVERVIEW

MATLAB and the Image Processing Toolbox provide a completely interactive, feature-rich environment that helps you speed the development of applications involving image processing and analysis. The Image Processing Toolbox features a robust set of image processing algorithms and provides direct access to MATLAB's comprehensive foundation of numeric computation, data analysis, algorithm development, and visualization capabilities. Together, MATLAB and the Image Processing Toolbox deliver a broad, flexible set of tools that can be used to address a wide range of multidisciplinary problems.

## KEY FEATURES

- Geometric operations
- Pixel values and statistics
- Image analysis
- Image enhancement and noise removal
- Linear filtering
- Linear 2-D filter design
- Image transforms
- Neighborhood and block processing
- Region-of-Interest (ROI) processing
- Binary image operations
- Colormap manipulation
- Color space conversions
- Image type conversions



*This group of images illustrates the arbitrary region-of-interest (ROI) processing capability in the Image Processing Toolbox. Clockwise from upper left: blood cells, nerve fibers, Boston Harbor, and the planet Saturn. Boston Harbor courtesy of USGS/EROS.*

Version 2 of the Image Processing Toolbox takes advantage of new MATLAB 5 features, such as support for multidimensional arrays, expanded image file I/O capabilities, enhanced graphics performance, and new application development features. Other improvements in version 2 include significantly increased processing performance, a broadened image processing and analysis suite, and expanded support for 8- and 16-bit data types.

## BACKGROUND

Image processing plays an integral role in a wide variety of scientific and engineering applications. Enhancing and extracting information from image data is frequently used in combination with other analysis techniques—such as signal processing, fuzzy logic, neural networks, or statistics—to solve multidisciplinary problems in applications such as aerospace/defense, astronomy, remote sensing, medical imaging, bioscientific imaging, manufacturing automation, material science, and genetics research.

Many image processing solutions currently available to engineers and scientists lack the breadth and depth of capabilities required for these multidisciplinary problems. Image processing software programs such as GUI-based application packages suffer from fixed functionality and cross-platform incompatibility, while C/C++ libraries require a substantial programming effort on the part of the user.

The Image Processing Toolbox, in contrast, combines high-level, easy-to-use functions for image processing and analysis with MATLAB's open and extensible technical computing environment. This solution eliminates the functional limitations of GUI-based packages and



*The toolbox's unsharp filter and contrast enhancement operations were used to enhance this GOES-8 satellite image showing snow cover areas across Northeastern U.S. from the April 1, 1997 blizzard. Courtesy of NOAA/NCDC.*

the time-consuming coding and debugging start-up work associated with C/C++ libraries—while offering the convenience and flexibility of both. With the Image Processing Toolbox you can focus on the “big picture”—extracting reliable answers from your image data.

## THE MATLAB FOUNDATION

MATLAB is a highly expressive, intuitive, and interactive matrix-based language that allows you to efficiently and compactly express image processing operations. MATLAB's matrix-oriented language provides a rich set of analysis and visualization capabilities with which you can quickly and easily manipulate, process, and analyze image data. You can interactively generate 2-D or 3-D graphical plots, visualize interim data on demand, and annotate results for technical publications. In addition, MATLAB gives you immediate and flexible access to image data, down to individual pixels, at any point in the algorithmic process.

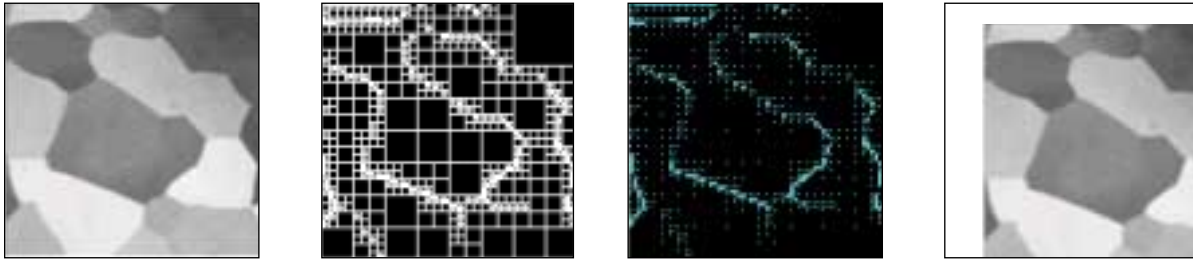
## A FLEXIBLE APPLICATION DEVELOPMENT ENVIRONMENT

MATLAB 5 provides a number of new application development features to facilitate the development of customized image processing applications directly within the MATLAB environment. Programming tools in MATLAB 5 include an interactive visual debugger for algorithm development and refinement, a performance profiler to optimize runtime performance of algorithms in applications, and an interactive graphical user interface builder to rapidly develop custom graphical front ends for your applications.

## IMAGE PROCESSING TOOLBOX APPLICATIONS

The Image Processing Toolbox is currently used at thousands of companies and universities worldwide to solve challenging image processing problems such as:

- Restoration of noisy or degraded images
- Enhancement of images and 2-D sensor data to improve intelligibility and highlight information content
- Research and algorithm development for the next generation of image processing systems and products
- Preprocessing of image data prior to pattern recognition operations (e.g., performed with the Neural Network Toolbox)
- Extraction and analysis of image data features through the use of 2-D statistics and transforms



This series of images illustrates the quadtree decomposition algorithm which is useful in regionally adaptive image compression and regional homogeneity analysis. From left to right: original image of aluminum grains, quadtree decomposition, sparse representation, reconstructed image.

Used in combination with other MATLAB Toolboxes, the Image Processing Toolbox can also solve challenging multidisciplinary problems involving signal processing, wavelet analysis, geographic mapping, neural networks, fuzzy logic, and statistical analysis.

## LIST OF FUNCTIONS

### Image Display

For controlling the display of individual or multiple images and animations.

colorbar	Display colorbar*
getimage	Get image data from axes
image	Create and display image object*
imagesc	Scale data and display as image*
immovie	Make movie from multiframe indexed image
imshow	Display image
montage	Display multiple image frames as rectangular montage
subimage	Display multiple images in single figure
truesize	Adjust display size of image
warp	Display image as texture-mapped surface
zoom	Zoom in and out of image or 2-D plot*

### Image File I/O

For loading, saving, and retrieving information about images from their image headers. Supported file formats include BMP, HDF, JPEG, PCX, TIFF, PNG, and XWD.

imfinfo	Return information about image file*
imread	Read image file*
imwrite	Write image file*

### Geometric Operations

For performing various geometric manipulations on images.

imcrop	Crop image
imresize	Resize image
imrotate	Rotate image
interp2	2-D data interpolation*

### Pixel Values and Statistics

For extracting grayscale statistics and other information about pixel values in images.

corr2	Compute 2-D correlation coefficient
imcontour	Create contour plot of image data
imhist	Display histogram of image data
improfile	Compute pixel-value cross-sections along line segments
mean2	Compute mean of matrix elements
pixval	Interactive pixel-value display
std2	Compute standard deviation of matrix elements

### Image Analysis

For analyzing images to extract information about their structure.

edge	Find edges in intensity image
qtdecomp	Perform quadtree decomposition
qtgetblk	Get block values in quadtree decomposition
qtsetblk	Set block values in quadtree decomposition

\* native to the MATLAB language

## Image Enhancement

For enhancing images to make certain features easier to see or to reduce noise.

histeq	Enhance contrast using histogram equalization
imadjust	Adjust image intensity values or colormap
imnoise	Add noise to an image
medfilt2	Perform 2-D median filtering
ordfilt2	Perform 2-D order-statistic filtering
wiener2	Perform 2-D adaptive noise-removal filtering

## Linear Filtering

For applying arbitrary as well as predefined filters to images and N-dimensional datasets.

conv2	Perform 2-D convolution*
convmtx2	Compute 2-D convolution matrix
convn	Perform N-D convolution*
filter2	Perform 2-D linear filtering*
fspecial	Create predefined filters

## Linear 2-D Filter Design

For designing 2-D FIR filters to meet given frequency-domain specifications.

freqspace	Determine 2-D frequency response spacing*
freqz2	Compute 2-D frequency response
fsamp2	Design 2-D FIR filter using frequency sampling
ftrans2	Design 2-D FIR filter using frequency transformation
fwind1	Design 2-D FIR filter using 1-D window method
fwind2	Design 2-D FIR filter using 2-D window method

**Note:** The Signal Processing Toolbox is strongly recommended for 2-D FIR filter design to generate the inputs (1-D windows and 1-D filter prototypes) to the 2-D FIR design functions.

## Image Transforms

For alternative mathematical representations of images for advanced analysis.

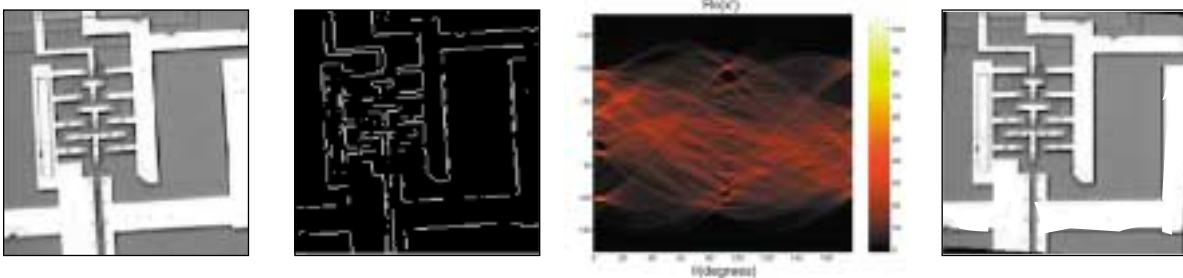
dct2	Compute 2-D discrete cosine transform
dctmtx	Compute discrete cosine transform matrix
fft2	Compute 2-D fast Fourier transform*
fftn	Compute N-D fast Fourier transform*
fftshift	Reverse quadrants of output of FFT*
idct2	Compute 2-D inverse discrete cosine transform
ifft2	Compute 2-D inverse fast Fourier transform*
ifftn	Compute N-D inverse fast Fourier transform*
iradon	Inverse Radon transform
radon	Compute Radon transform

## Neighborhood & Block Processing

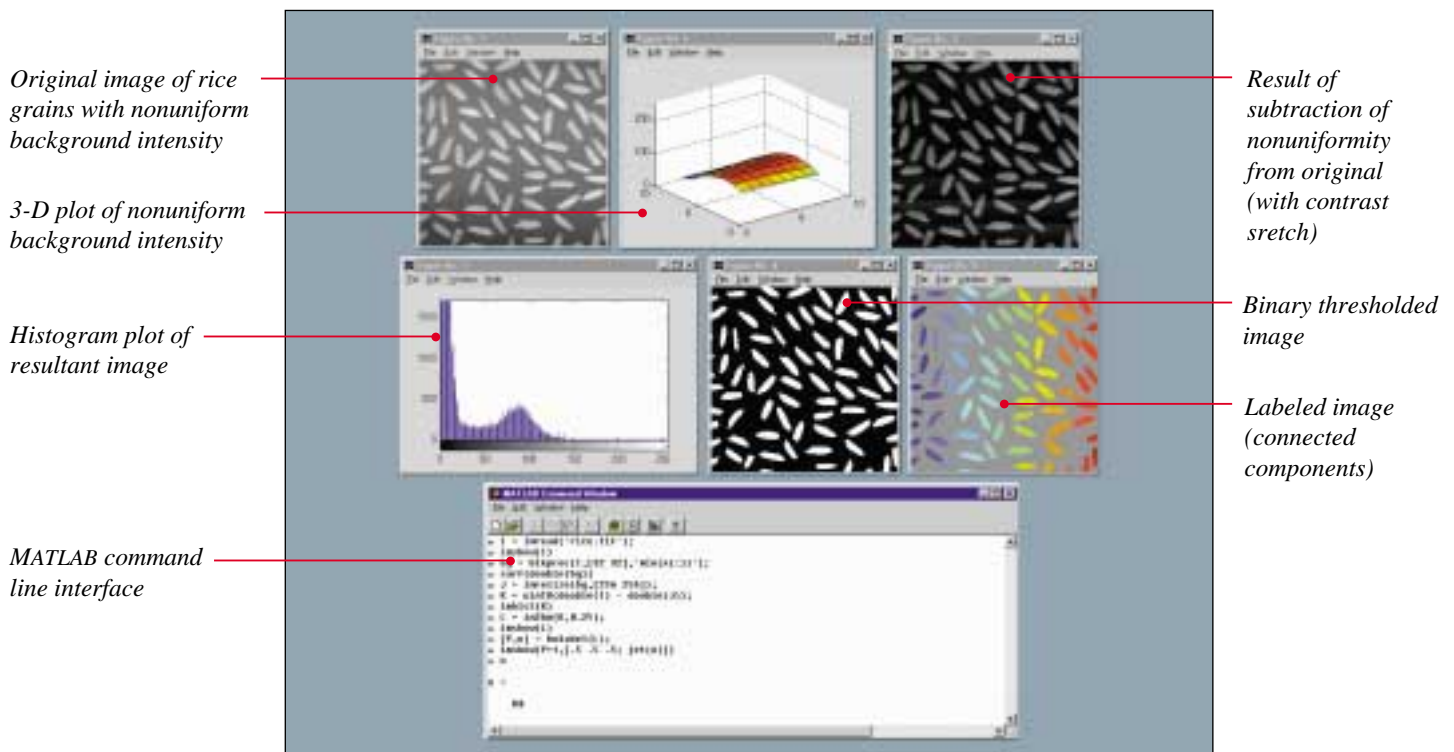
For performing neighborhood operations on images.

bestblk	Choose block size for block processing
blkproc	Implement distinct block processing for image
col2im	Rearrange matrix columns into blocks
colfilt	Perform neighborhood operations using columnwise functions
im2col	Rearrange image blocks into columns
nlfilter	Perform general sliding-neighborhood operations

\* native to the MATLAB language



In this series of images, a radon transform is used to determine the orientation of a scene. From left to right: original image of a microcircuit, binary edge detection, graph of the output of the radon transform showing  $\theta = 94^\circ$ , image rotated by  $-4^\circ$ .



A typical interactive session using MATLAB and the Image Processing Toolbox to perform connected components analysis—also known as blob analysis—on an image with nonuniform background intensity. Note that it took only twelve functions to perform the analysis.

### Region-based Processing

For performing image processing and analysis operations on arbitrary regions of images.

- roicolor Select region of interest, based on color
- roifill Smoothly interpolate within arbitrary region
- roifilt2 Filter a region of interest
- roipoly Select polygonal region of interest

### Binary Image Operations

For computing a variety of morphological and object labeling operations for image enhancement and feature extraction.

- applylut Perform neighborhood operations using lookup tables
- bwarea Compute area of objects in binary image
- bweuler Compute Euler number of binary image
- bwfill Fill regions or holes in binary image
- bwlabel Label connected components in binary image
- bwmorph Perform morphological operations on binary image
- bwperim Determine perimeter of objects in binary image

- bwselect Select objects in binary image
- dilate Perform dilation on binary image
- erode Perform erosion on binary image
- makelut Construct lookup table for use with applylut

### Colormap Manipulation

For manipulating colormaps and approximating indexed images with fewer colors or with alternate colormaps.

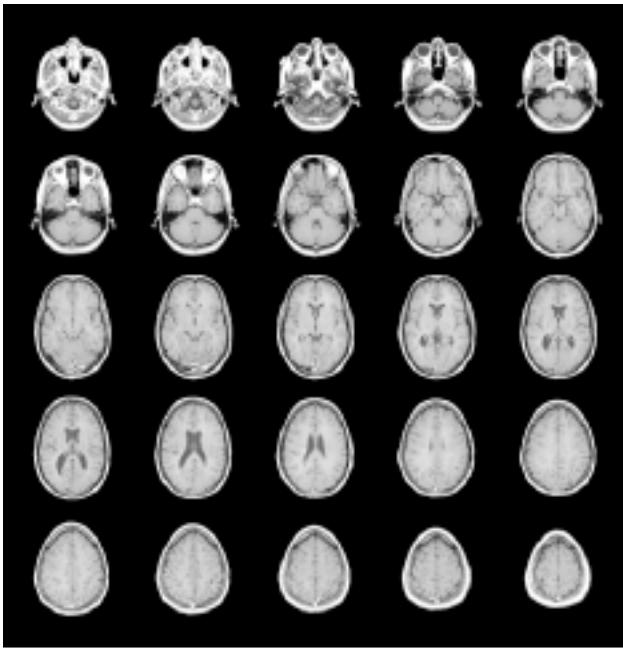
- brighten Brighten or darken colormap\*
- cmpermute Rearrange colors in colormap
- cmunique Find unique colormap colors and corresponding image
- colormap Set or get color lookup table\*
- imapprox Approximate indexed image with one containing fewer colors
- rgbplot Plot RGB colormap components\*

### Color Space Conversions

For converting images from one color space to another.

- hsv2rgb Convert HSV values to RGB color space\*

\* native to the MATLAB language



This series of MRI slices of the human skull illustrates the ability to concatenate a series of images into one image for easier viewing.

ntsc2rgb	Convert NTSC values to RGB color space
rgb2hsv	Convert RGB values to HSV color space*
rgb2ntsc	Convert RGB values to NTSC color space
rgb2ycbcr	Convert RGB values to YCbCr color space
ycbcr2rgb	Convert YCbCr values to RGB color space

### Image Type Conversions

For converting images from one data type to another.

dither	Convert image using dithering
gray2ind	Convert intensity image to indexed image
grayscale	Create indexed image from intensity image by thresholding
im2bw	Convert image to binary image by thresholding
ind2gray	Convert indexed image to intensity image

ind2rgb	Convert indexed image to RGB image*
isbw	Return true for binary image
isgray	Return true for intensity image
isind	Return true for indexed image
isrgb	Return true for RGB image
mat2gray	Convert matrix to intensity image
rgb2gray	Convert RGB image or colormap to grayscale
rgb2ind	Convert RGB image to indexed image

### Toolbox Preferences

For setting properties such as border, axes box, tick labels, and size of image displays.

iptgetpref	Get value of Image Processing Toolbox preference
iptsetpref	Set value of Image Processing Toolbox preference

## DOCUMENTATION AND EXAMPLES


The *Image Processing Toolbox User's Guide* describes toolbox functionality in a thorough tutorial and reference format. GUI-driven demonstration facilities introduce the major features of the product in an easy-to-use format.

## OPEN AND EXTENSIBLE

MATLAB and the Image Processing Toolbox provide an open and extensible environment that enables users to interactively explore their ideas, prototype new algorithms, and develop complete solutions to their most challenging problems. The MATLAB language facilitates the creative use of computation, visualization, and specialized techniques drawn from any toolboxes within the extensive MATLAB product family.

Finally, the cross-platform nature of the MATLAB language allows you to share image processing algorithms and programs across a variety of platforms including PC, UNIX, and Macintosh.

\* native to the MATLAB language

	The MathWorks, Inc. 24 Prime Park Way Natick, MA 01760-1500 USA Tel: 508 647-7000 Fax: 508 647-7101 E-mail: info@mathworks.com www.mathworks.com
--	--