The Image Processing Toolbox extends the MATLAB® computing environment to provide functions and interactive tools for enhancing and analyzing digital images and developing image processing algorithms. In addition, it facilitates the learning and teaching of image processing techniques in both academic and research settings.

Together, MATLAB and the Image Processing Toolbox provide scientists, researchers, and engineers with a diverse, flexible set of tools for solving complex imaging problems in disciplines such as aerospace/defense, astronomy, remote sensing, medical and scientific imaging, and materials science. Most functions are implemented in the open MATLAB language, letting you explore and customize existing toolbox algorithms or develop your own.

You can use the toolbox for the restoration of noisy or degraded images, image enhancement for improved intelligibility, blob analysis, and extraction and analysis of image data with 2-D statistics and transforms, as well as to develop complete solutions to challenging image processing problems that involve multidimensional data sets.

The toolbox’s deblurring algorithms were used to restore these images that were blurred by distortion operators. From left to right: a circuit board, stained tissue, and an image of the sun.

KEY FEATURES

- Linear filtering and filter design
- Image analysis, including pixel, region, and feature statistics and measurement
- Image enhancement
- Binary and grayscale morphology
- Image segmentation
- Spatial transformation
- Image registration
- Image deblurring
- FFT, DCT, and radon transform
- Region-of-interest processing
- Multidimensional image processing
- DICOM import in addition to supported file formats in MATLAB

Stained tissue image—courtesy of J.C. Russ, The Image Processing Handbook
Sun image—courtesy of the SOHO EIT Consortium. SOHO is a mission of international cooperation between ESA and NASA.
Data Import and Export

Getting your image data in and out of MATLAB is easy. MATLAB and the Image Processing Toolbox support many standard data and image formats from areas such as medical imaging, remote sensing, and astronomy. The supported formats include JPEG, TIFF, HDF, DICOM, and others. You can also create AVI movies from sets of images.

MATLAB supports other industry standard file formats, such as Microsoft® Excel. Additional functions perform ASCII and low-level binary I/O, allowing you to develop custom routines for working with any data format.

Image Analysis and Enhancement

MATLAB and the Image Processing Toolbox support a broad range of advanced image processing functions. You can extract and analyze image features, compute feature measurements, and apply filter algorithms. You can use filters for different types of image noise or build customized filters with the filter design tools that are included. Interactive tools allow you to select arbitrary regions of interest, measure distances in images, and obtain pixel values and statistics.

You can use the inverse Radon transform (commonly used in tomography applications) to reconstruct images from projection data. The discrete cosine transform, which is at the heart of the JPEG standard, lets you prototype compression algorithms. The toolbox includes several edge detection algorithms, including the Canny, Sobel, and Roberts methods, to identify object boundaries in an image.

An extensive collection of morphology functions for grayscale and binary images allows you to perform operations that are sensitive to specific shapes in the input image. Using the morphological functions of the toolbox you can quickly perform image processing tasks, such as edge detection, noise removal, skeletonization, and granulometry. Specialized morphology functions include hole filling, peak detection, and watershed segmentation. All morphology functions support multidimensional images.

Image Manipulation

The Image Processing Toolbox lets you use advanced techniques to alter different aspects of your images. The included functions let you easily align, transform, and deblur images. You can also interactively crop and resize your images.

An interactive point-selection tool lets you align two images by picking points in a pair of images that identify the same feature or landmark in each. The two images can then be aligned by performing a spatial transformation such as linear conformal, affine, projective, polynomial, piecewise linear, or local weighted mean. The toolbox also lets you define your own transformations and supports multidimensional transformations.

The toolbox supports several fundamental algorithms such as Lucy-Richardson, Wiener, and regularized filter deconvolution for the deblurring or deconvolution of images. All deblurring functions support multidimensional images.

Visualization

MATLAB visualization tools let you depict the information contained in your image data through histograms, contour plots, montages, pixel profiles, transparent overlays, and images texture-mapped onto surfaces. You can view and measure image attributes, such as the location and value of pixels in an image, or display colorbars to view the mapping of colors to values.
Using the powerful volume visualization in MATLAB you can create graphical representations, including an isosurface display of multidimensional image data sets, helping you to highlight certain characteristics or values. You can also use multiple light sources for colored surfaces and a camera-based viewing and perspective control to enhance graphical representations. Point-and-click tools let you easily edit and annotate graphics.

**Algorithm Development and Application Deployment**

MATLAB lets you easily encapsulate your image processing solution in a customized software application. Helpful features include an interactive GUI builder to rapidly develop custom graphical front ends for your image processing applications. In addition, you can use MATLAB’s built-in programming tools — such as a visual debugger for algorithm development and refinement — and an algorithm performance profiler, to accelerate development.

You can share image processing algorithms created in the MATLAB language across all MATLAB supported platforms. You can also integrate your algorithms with existing C programs or deploy the developed algorithms and GUIs as stand-alone applications. With the MATLAB Compiler, you can automatically convert your image processing algorithms into C and C++ code.

You can solve challenging multidisciplinary problems using the Image Processing Toolbox in combination with other MATLAB Toolboxes such as Signal Processing, Wavelet, Mapping, Neural Network, and Statistics. The MATLAB language facilitates the creative use of computation, visualization, and specialized techniques that can be drawn from any of the toolboxes within the MATLAB product family.
Sample Functions

Some of these functions are native to the MATLAB language.

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

- `colorbar`: Display colorbar
- `immovie`: Make movie from multiframe image
- `imshow`: Display image
- `montage`: Display multiple image frames as rectangular montage
- `movie`: Play recorded movie frames
- `subimage`: Display multiple images in single figure
- `warp`: Display image as texture-mapped surface

### Spatial Transformations
For performing various geometric manipulations on images.

- `imcrop`: Crop image
- `imresize`: Resize image
- `imrotate`: Rotate image
- `imtransform`: Apply spatial transformation to image
- `tformarray`: Apply spatial transformation to multidimensional array

### Image Registration
For selecting control points and aligning two images.

- `cp2tform`: Infer geometric transformation from control point pairs
- `cpcorr`: Tune control point locations using cross-correlation
- `cpselect`: Control point selection tool
- `normxcorr2`: Normalized two-dimensional cross-correlation

### Image File I/O
For loading, saving, and retrieving information about images. (Supported file formats include: AVI, BMP, CDF, DICOM, FITS, HDF, HDF-EOS, JPEG, PCX, PNG, TIFF, XWD)

- `dicominfo`: Read metadata from a DICOM message
- `dicomread`: Read a DICOM image
- `imfinfo`: Return information about image file
- `imread`: Read image file
- `imwrite`: Write image file

### Image Arithmetic
For performing integer and floating-point arithmetic operations on images.

- `imabsdiff`: Compute absolute difference of two images
- `imadd`: Add two images, or add constant to image
- `imcomplement`: Complement image
- `imdivide`: Divide two images, or divide image by constant
- `imlincomb`: Compute linear combination of images
- `immultiply`: Multiply two images, or multiply image by constant
- `imsubtract`: Subtract two images, or subtract constant from image

### Image Enhancements
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
- `stretchlim`: Find limits to contrast stretch an image
- `wiener2`: Perform 2-D adaptive noise-removal filtering

### Linear Filtering
For applying arbitrary as well as predefined filters to images and multidimensional datasets.

- `fspecial`: Create predefined filters
- `imfilter`: Filter 2-D and multidimensional images

### 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

### 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

### Image Deblurring
For deblurring images using different algorithms.

- `deconvlucy`: Deblur image using Lucy-Richardson method
- `deconvreg`: Deblur image using a regularized filter
- `deconvwnr`: Deblur image using Wiener method

Note: The Signal Processing Toolbox is 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.
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 nine functions to perform the analysis.
Sample Functions continued

**Morphological Operations (binary images)**
For extracting information about shapes from binary images.

- `applylut` Perform neighborhood operations using lookup tables
- `bwarea` Compute area of objects in binary image
- `bwareaopen` Binary area open (remove small objects)
- `bwdist` Compute distance transform of binary image
- `bweuler` Compute Euler number of binary image
- `bwhitmiss` Binary hit-miss operation
- `bwlabel` Label connected components in 2-D binary image
- `bwlabeln` Label connected components in multidimensional binary image
- `bwmorph` Perform morphological operations on binary image
- `bwpack` Pack binary image
- `bwperim` Determine perimeter of objects in binary image
- `bwestect` Select objects in binary image
- `bwulterode` Ultimate erosion
- `bwnpack` Unpack 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 by one with fewer colors

**Color Space Conversions**
For converting images from one color space to another.

- `hsv2rgb` Convert HSV values to RGB color space
- `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

**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

**Image Types and Type Conversions**
For converting images from one type to another.

- `dither` Convert image using dithering
- `gray2ind` Convert intensity image to indexed image
- `grayscale` Create indexed image from intensity image by thresholding
- `graythresh` Compute global image threshold using Otsu’s method
- `im2bw` Convert image to binary image by thresholding
- `ind2gray` Convert indexed image to intensity image

**Documentation and Examples**
The Image Processing Toolbox User’s Guide describes toolbox functionality in a tutorial and reference format. It includes many examples that take you step-by-step through the solution of an image processing problem in MATLAB. In addition, GUI-driven demonstrations introduce major features of the product in an easy-to-use format.

For demos, application examples, tutorials, user stories, and pricing:
- Visit [www.mathworks.com](http://www.mathworks.com)
- Contact The MathWorks directly
  - US & Canada 508-647-7000
  - Benelux +31 (0)182 53 76 44
  - France +33 (0)1 41 14 67 14
  - Germany +49 (0)89 995901 0
  - Spain +34 93 362 13 00
  - Switzerland +41 (0)31 954 20 20
  - UK +44 (0)1223 423 200

Visit [www.mathworks.com](http://www.mathworks.com) to obtain contact information for authorized MathWorks representatives in countries throughout Asia Pacific, Latin America, the Middle East, Africa, and the rest of Europe.