MatrixUser v2.0 User Guide

Fang Liu
(leoliuf@gmail.com)

October 2, 2013

Contents

1 What is MatrixUser

The MatrixUser is a matrix analysis software package developed under Matlab Graphical User Interface Developing Environment (GUIDE). It features functions that are designed and optimized for working with multi-dimensional matrix under Matlab. These functions typically include functions for multi-dimensional matrix display, matrix (image stack) analysis and matrix processing.

MatrixUser v2.0 is made available and can be downloaded from SourceForge website. MatrixUser is released as free software. This means that you are free to use and modify this software as your needs, as long as you acknowledge the original author in any future work. If you find MatrixUser useful for the publication of any scientific results, including a line in your acknowledgments section referencing to MatrixUser and this belowing address is requested.

MatrixUser downloading address:

`https://sourceforge.net/projects/matrixuser/`

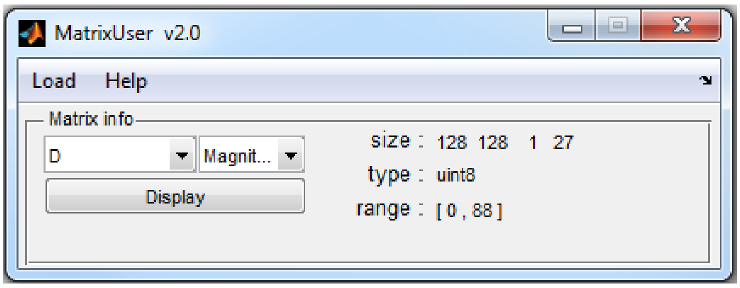
2 Installing and Running MatrixUser

To run MatrixUser, the Matlab is the only dependency package that is required. Note that MatrixUser v2.0 uses certain functions from Matlab imaging processing toolbox, therefore this toolbox is also required for properly running some image processing functions in MatrixUser. The current version of MatrixUser has been tested under the following Matlab versions:

- Matlab R2011a 64-bit Windows
- Matlab R2012b 64-bit Unix

Installing and running MatrixUser is easy, you only need to download MatrixUser source code in a compressed file, then uncompress the MatrixUser root folder, put the folder to any location in you computer. To run MatrixUser, you need open Matlab, then simple run the 'Main.m' script that is under the MatrixUser root folder.

The MatrixUser main window (Figure 1) opens up as a matrix manager for loaded matrices.



size : 128 128 1 27
type : uint8
range : [0 , 88]

Figure 1: MatrixUser Main Window

3 Data Import

MatrixUser supports any valid Matlab matrix. By default, MatrixUser reads the Matlab base workspace, and scans any matrices that are currently existing in the Matlab session, then creates a matrix list for tracking their content. Once these matrices are updated, MatrixUser will also update the matrix list. In addition, there are several different approaches to import data from outside Matlab into MatrixUser. The imported matrices will be saved into Matlab base workspace. The import functions are located under 'Load' menu, including:

- Load MAT file

  The default Matlab .mat file is natively supported by MatrixUser.

- Load System Clipboard

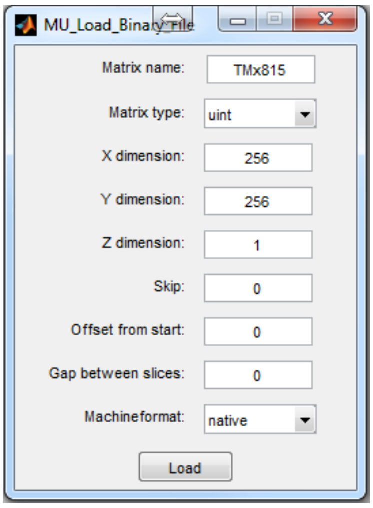
  If image content exists in the system clipboard, it can be converted into a RGB image that contains a three slice matrix with each slice corresponds to the individual Red, Green and Blue channel.

- Load ScreenShot

  MatrixUser takes a full screenshot for current monitor and saves it into a RGB image as described above.

- Load from Binary file

  Binary data file is supported by MatrixUser. The user needs to properly configure loading parameters (Figure 2) according to the matrix size and data type information.



MU_Load_Binary File

Matrix name: TMx815

Matrix type: uint

X dimension: 256

Y dimension: 256

Z dimension: 1

Skip: 0

Offset from start: 0

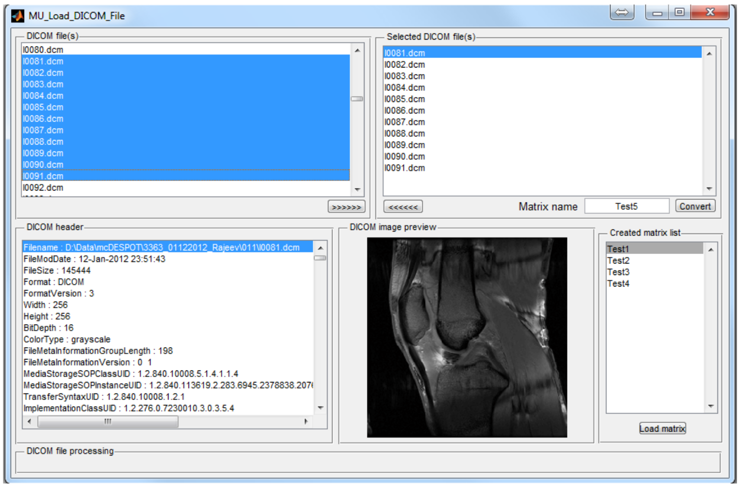
Gap between slices: 0

Machineformat: native

Figure 2: MatrixUser Binary File Loading Interface

- Load DICOM file(s)

  MatrixUser supports loading multiple DICOM files by using a file filter interface (Figure 3). The user needs to load DICOM files into the loading interface by selecting wanted DICOM files (multiple selection supported) from a data folder. The selected files are listed in the DICOM file list. The user can click a single file to read its DICOM header and image preview. To manually create a matrix using DICOM files, the user needs to choose files from the left list, then press '>>>>>>' to push the files into selected DICOM file list. A matrix name must also be provided. Then pressing 'Convert' button will create a matrix based on chosen DICOM files. The user can load these created matrices into workspace by pressing 'Load matrix' button.



Matrix name Test5

```
Filename : D:\Data\mcDESPOT\3363_01122012_Rajeev\011\I0081.dcm
FileModDate : 12-Jan-2012 23:51:43
FileSize : 145444
Format : DICOM
FormatVersion : 3
Width : 256
Height : 256
BitDepth : 16
ColorType : grayscale
FileMetaInformationGroupLength : 198
FileMetaInformationVersion : 0  1
MediaStorageSOPClassUID : 1.2.840.10008.5.1.4.1.1.4
MediaStorageSOPInstanceUID : 1.2.840.113619.2.283.6945.2378838.207
TransferSyntaxUID : 1.2.840.10008.1.2.1
ImplementationClassUID : 1.2.276.0.7230010.3.0.3.5.4
```

Figure 3: MatrixUser DICOM File Loading Interface

- Load DICOM file(s) in Batch

  MatrixUser also supports loading DICOM files in a batch mode from a data folder. This function requires the folder path is provided. MatrixUser will try to create separate matrices for DICOM files that may come from different image series. A matrix selection interface will provides converted matrices that can be loaded into Matlab workspace.

- Load from NIfTI file

  NIfTI file with .nii suffix is supported by MatrixUser.

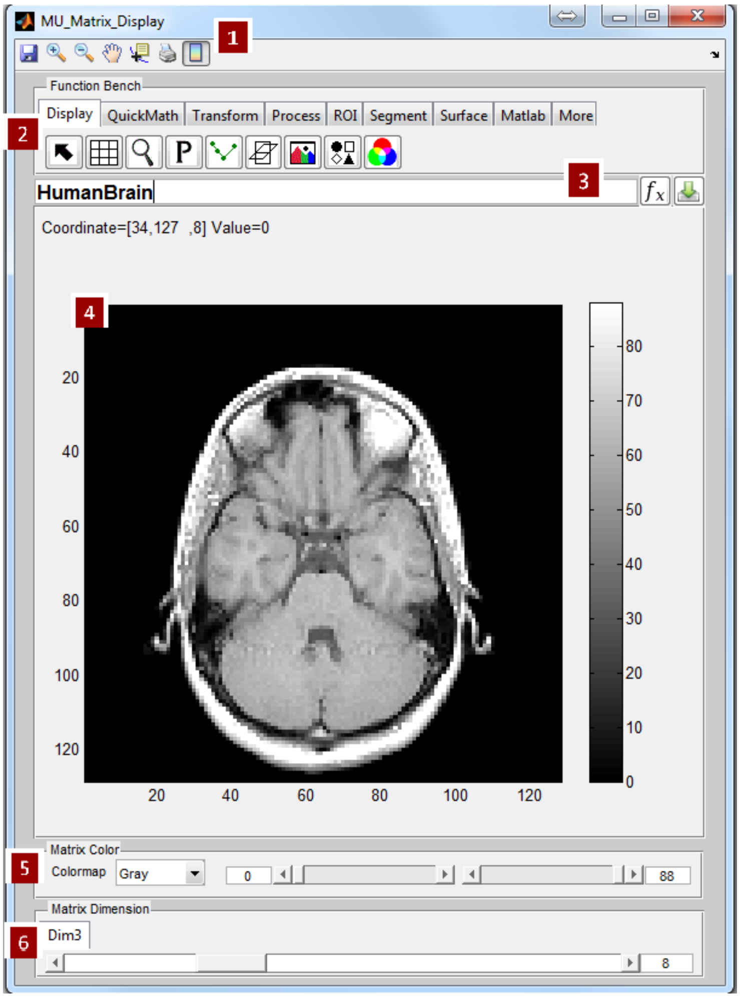
4 Window Layout

The user can choose to display selected matrix by using the popup menu on the main window. The matrix size, type and value range are calculated and provided on the right side. The user can press 'Display' button to activate MatrixUser display window. If the selected matrix contains complex value, four options are available for displaying magnitude, phase, real and imaginary of the matrix. Figure 4 demonstrates an overview of the window layout of the MatrixUser display window. The window consists of

1. Matlab Default Toolbar

   Matlab toolbar provides basic functions for analyzing displayed matrix. These functions include:

   - 💾 : Save current image axes into an image file



HumanBrain

Coordinate=[34,127 ,8] Value=0

Figure 4: MatrixUser Display Window

- 🔍 : Zoom in matrix area
- 🔍 : Zoom out matrix area
- ✋ : Manually move matrix position
- ✚ : Check individual voxel value and index
- 🖨 : Print current figure
- ▮ : Turn on/off color bar

2. MatrixUser Function Library

   Most of the matrix analysis functions are represented under function bench panel. MatrixUser groups these functions into categories and dynamically loads them according to the size and compatibility of current displayed matrix. A multi-tab is used to contain individual function button that corresponds to each function. The tabs under the multi-tab are used to switch between function categories, which include

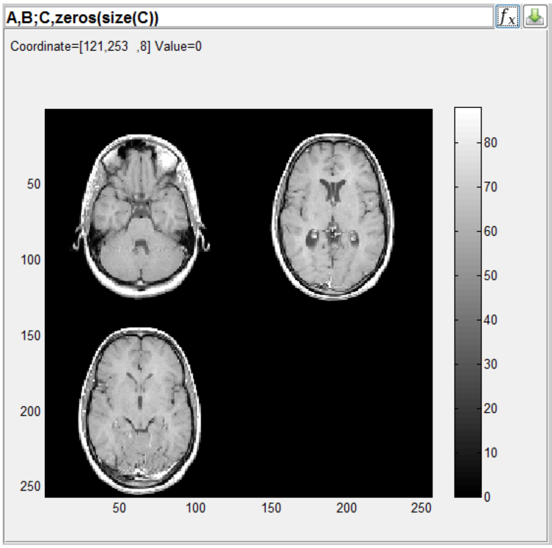
   - Display : Multi-dimensional matrix display relevant functions
   - QuickMath : Perform math calculation for current matrix
   - Transform : Transform current matrix
   - Process : Basic matrix processing functions
   - ROI : Region-of-Interest relevant functions
   - Segment : Segmentation relevant functions
   - Surface : Generate surface or mesh plot for current image
   - Matlab : Matlab default image tools
   - More : Uncategorized functions

3. Matrix Calculator

   The matrix calculator consists of three control units, including a matrix expression editbox, an execution button (f_x) and a matrix saving button (⬇). Any valid matrix calculation expression can be edited and executed in the calculator and updated in the display window, which serves as a convenient way for analyzing matrix calculation result. Matrix concatenation and recombination can also be done in the calculator, for example, to side-by-side compare multiple 3D matrices (Figure 5). Some valid calculation examples are, but not limited to:

   - A,B Combines A and B in one row

- A,B;C,D Combines A and B in the first row, C and D in the second row
- A,B;C,zeros(size(C)) Combines A and B in the first row, C in the second row, pad with zero value
- sin(A),cos(B) Calculates voxel-wise sin for A and cos for B, then combine them in one row
- A(:,1:10,:) Extracts submatrix from A and display it



A,B;C,zeros(size(C))

Coordinate=[121,253 ,8] Value=0

Figure 5: MatrixUser Concatenation Example

Where A, B, C and D are multi-dimensional matrices with proper matrix size in order to maintain valid Matlab matrix calculation rule as in above examples. Also note that all the source matrices have to exist in the workspace. Pressing the execution button will perform the calculation and save the result as a temporary matrix. The user can also save the temporary matrix into workspace by pressing matrix saving button. The saved temporary matrix will have a ‘_tmp’ suffix by default.

4. Matrix Display Axes

   One slice of current matrix is displayed in the display axes. The user can use mouse cursor to inspect the coordinate and value of any voxel. Moving mouse wheel back and forth moves the slice location in the current dimension and updates the display axes.

5. Matrix Color Control Group

The matrix color control group provides a set of sliders, editboxes and popup menu that help control image color map and contrast. This group consists of

- Colormap Popup Menu: Choose different colormap scheme
- Upper Color Bound Slider (right slider): Slide to control the upper bound of color limits
- Upper Color Bound EditText (right editbox): Edit to control the upper bound of color limits
- Lower Color Bound Slider (left slider): Slide to control the lower bound of color limits
- Lower Color Bound EditText (left editbox): Edit to control the lower bound of color limits

6. Matrix Dimension Control Group

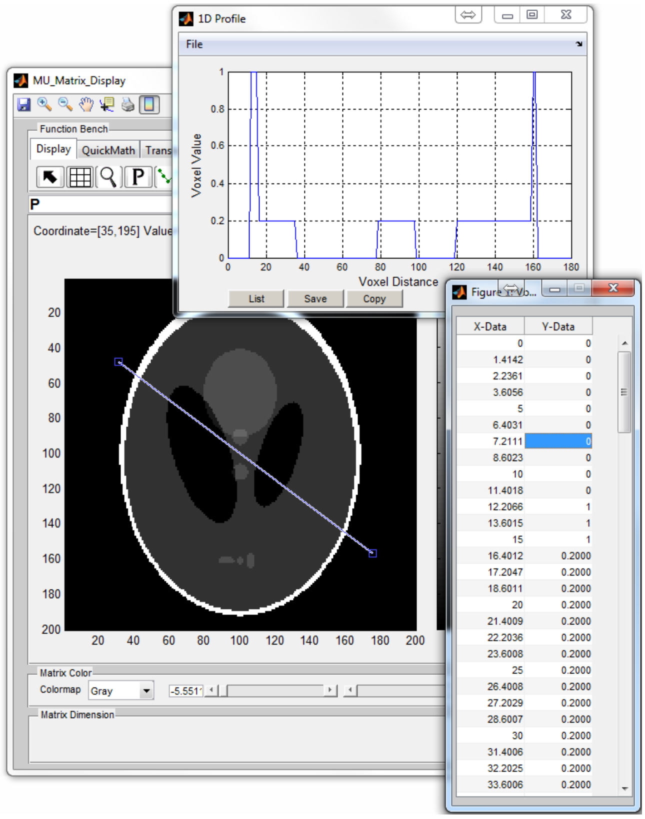
   MatrixUser measures the size of the displayed matrix and assigns one slider and editbox for each dimension that is above 2 (x and y). These control units are located in individual dimension tab and can be used to switch among slices in current active dimension.

5 Function Library

1. Display

   Matrix display relevant functions are listed under this tab.

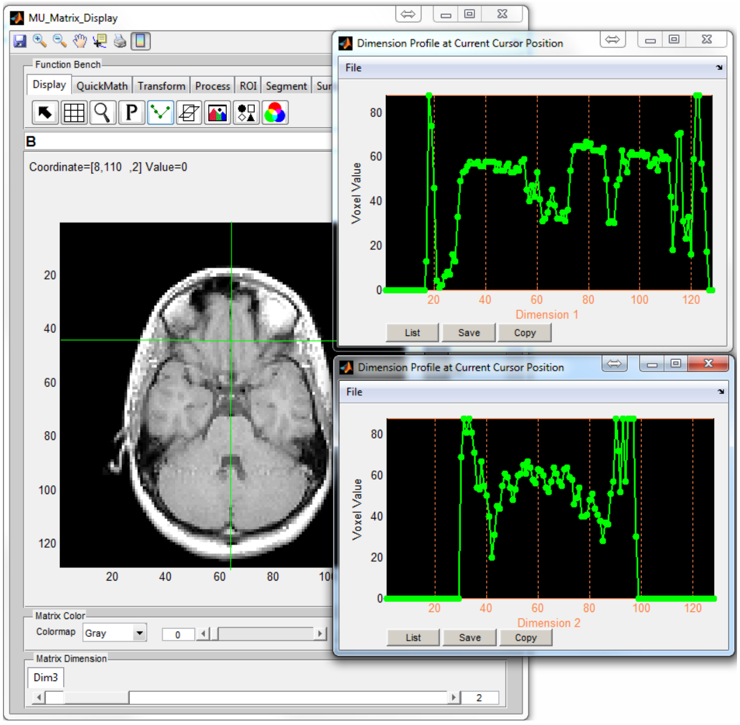
   - ↖ : Reset matrix display and erase additional display effect
   - ⊞ : Turn on and off black grid line on the image display axes
   - 🔍 : Open an instant magnifier which zooms in specific image area
   - P : Plot and update a profile curve along a resizable checking line (Figure 6)
   - ✓ : Plot and update a profile curve along one given matrix dimension (Figure 7)
   - ▱ : Open a separate window for another two orthogonal matrix display axes (Figure 8). The operation buttons on the second window consists of
     - 🟥 : Activate or deactivate localizer line on main display
     - 🟨 : Activate or deactivate localizer line on main display



Coordinate=[35,195] Value

X-Data	Y-Data
0	0
1.4142	0
2.2361	0
3.6056	0
5	0
6.4031	0
7.2111	0
8.6023	0
10	0
11.4018	0
12.2066	1
13.6015	1
15	1
16.4012	0.2000
17.2047	0.2000
18.6011	0.2000
20	0.2000
21.4009	0.2000
22.2036	0.2000
23.6008	0.2000
25	0.2000
26.4008	0.2000
27.2029	0.2000
28.6007	0.2000
30	0.2000
31.4006	0.2000
32.2025	0.2000
33.6006	0.2000

Figure 6: An Example of 1D Profile. The user can relocate and resize the profile checking line on the image for inspecting live 1D profile. The buttons on the profile window can list profile data, save the data as ‘data_plt’ array into workspace or copy the data into clipboard.



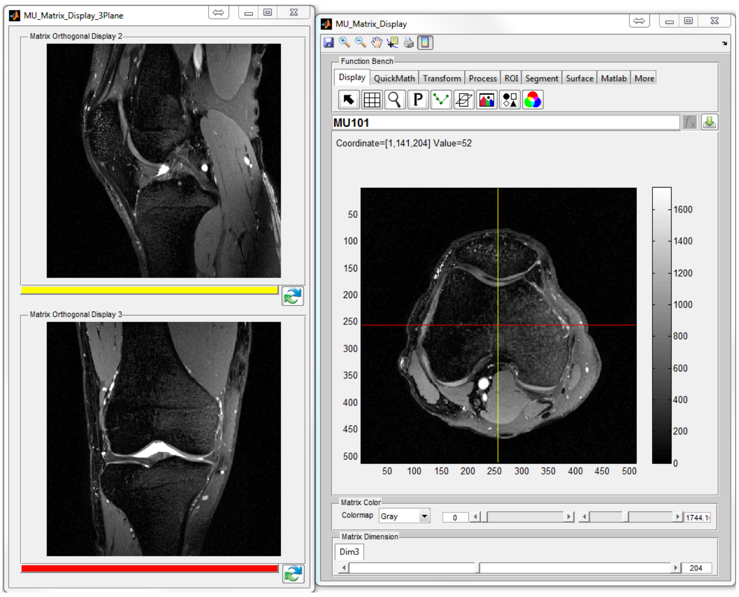
```
Coordinate=[8,110 ,2] Value=0
```

Colormap Gray 0

2

Figure 7: An Example of Plotting the First (Column) and Second (Row) Matrix Dimension Profile. The user needs to specify which matrix dimension to plot. The profile curve will update according to current mouse cursor position. The user can pause or resume live updating by right click on main display window.

– 🔄 : Switch matrices between main display and second display. This operation simply permutes current matrix into its orthogonal version. The user can save the transformed matrix using (⬇).



Coordinate=[1,141,204] Value=52

Figure 8: 3D Slicer. To change view slice, the user needs to activate localizer line and then click on wanted coordinate on the main display window.

- 🖼 : Create a RGB Image, assign current slice and following two slices to Red, Green and Blue channel, respectively
- ●□ : Create a montage image using multiple slices (Figure 9)
- ● : Overlap two matrices that have the same matrix size (Figure 10), notice that the user can press ↖ to remove foreground matrix from overlapping with background matrix.

2. QuickMath

This function category performs quick math calculation for current matrix. A few commonly used math calculation are provided under this tab. Instead, complex calculation can be performed using matrix calculator as mentioned above.

- |X| : Calculate absolute value
- -X : Calculate negative value

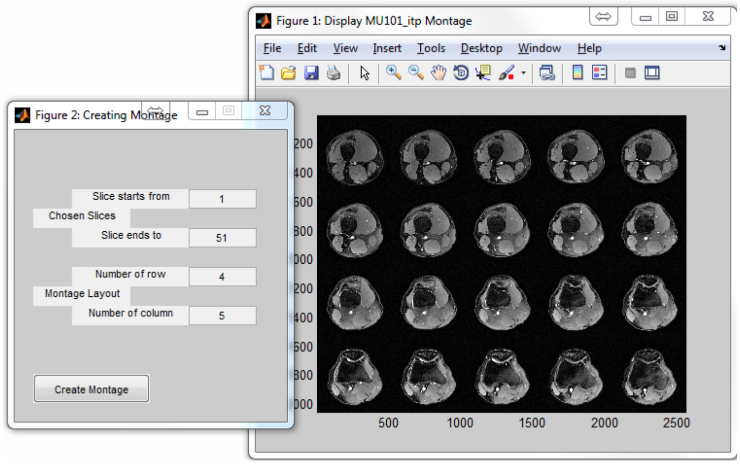


Figure 1: Display MU101_itp Montage
File Edit View Insert Tools Desktop Window Help
200
400
600
800
000
200
400
600
800
000
500
1000
1500
2000
2500
Figure 2: Creating Montage
Create Montage

Chosen Slices
Slice starts from 1
Slice ends to 51

Montage Layout
Number of row 4
Number of column 5

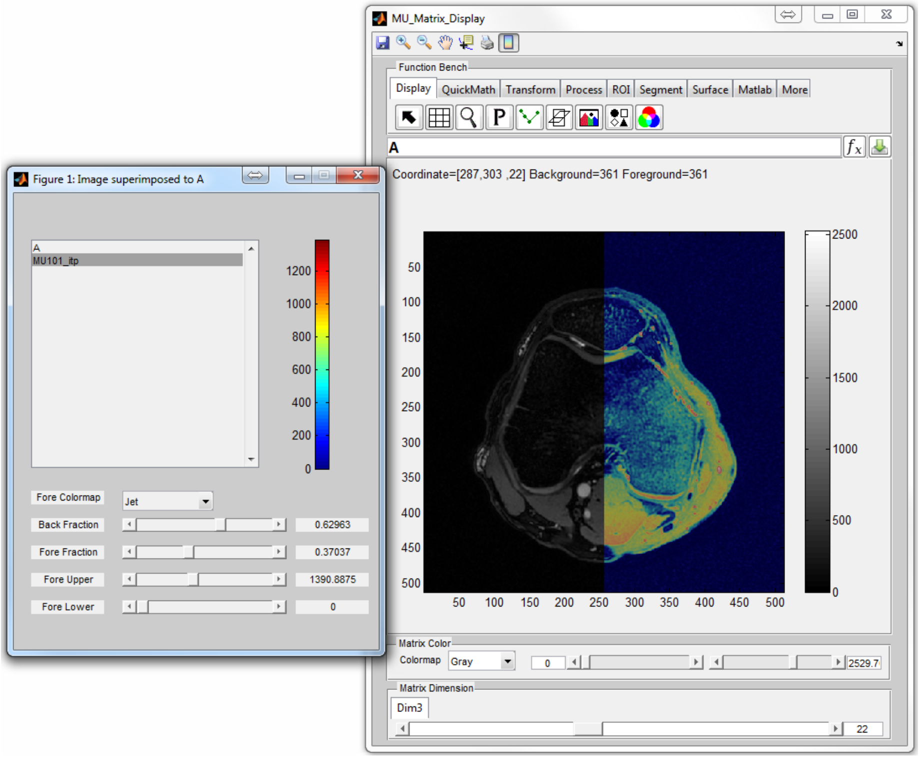
Figure 9: Montage Image. An example of creating a 4-by-5 montage image using multiple matrix slices starting from the first slice.

- ln : Calculate natural logarithm
- log : Calculate common (base 10) logarithm
- e^x : Calculate exponential
- 10^x : Calculate the power of 10
- sin : Calculate sine
- cos : Calculate cosine
- tan : Calculate tangent
- $\sin^{-1}$: Calculate inverse sine
- $\cos^{-1}$: Calculate inverse cosine
- $\tan^{-1}$: Calculate inverse tangent

3. Transform

   This function category performs spatial transformation or fast Fourier transform (FFT) to current matrix.

- ◭ : Flip matrix horizontally (along the first dimension)
- ◀ : Flip matrix vertically (along the second dimension)
- ◭ : Flip matrix along slice direction (the third dimension)



Coordinate=[287,303 ,22] Background=361 Foreground=361

Fore Colormap Jet

Back Fraction 0.62963

Fore Fraction 0.37037

Fore Upper 1390.8875

Fore Lower 0

Colormap Gray 0 2529.7

Dim3 22

Figure 10: An Example of Overlapping Two Matrices. A second window is provided for adjusting overlapping effect. The user can switch to overlap any size compatible matrices from the matrix list. There are also control units to change foreground colormap scheme, to modify image intensity fraction, or to change foreground color limits. The matrix values at current cursor for both background and foreground matrices are updated simultaneously at main display.

- : Rotate matrix 90 degree in the counter clockwise direction
- : Rotate matrix 90 degree in the clockwise direction
- : Rotate matrix certain degree along an axis in the 3D space, which is specified using the rotation axis origin and direction.
- : Translate matrix along certain direction
- FFT : Perform multi-dimensional FFT for current matrix, the user needs to specify up to which dimension to perform FFT.

4. Process

This function category performs basic matrix processing functions.

- : Create binary mask according to given threshold
- : Perform smoothing operation to current matrix
- : Perform sharpening operation to current matrix
- : Perform edge detection to current matrix
- : Provides various image filters (Figure 11)

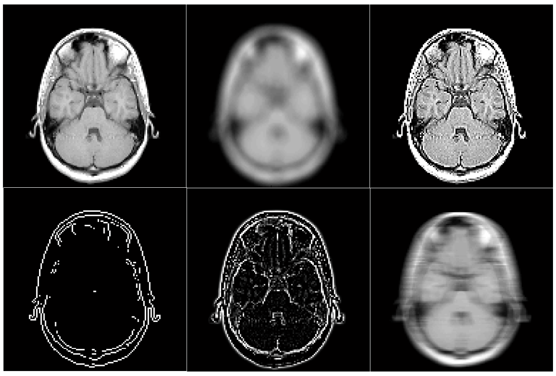


Figure 11: Image Filters. An example is shown for applying various image filters.

- : Add noise to matrix (Figure 12)
- : Replace voxel value for the voxels within certain value range and outside a polygon area. The user needs to draw a polygon first (double click to confirm the polygon).

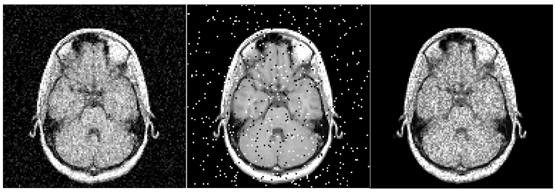


Figure 12: Image Noise. An example is shown for applying various image noise.

- : Replace voxel value for the voxels within certain value range and inside a polygon area. The user needs to draw a polygon first (double click to confirm the polygon).
- : Replace voxel value for the voxels inside a polygon area. The user needs to draw a polygon first (double click to confirm the polygon).
- : Replace matrix area with a selected matrix source region. The user needs to draw a free-hand area (source region), drag the free-hand region to target matrix area, then double click to confirm the operation.
- : Crop current matrix using a rectangle box (double click to confirm the cropping)
- : Extract parts of current matrix using irregular shape (double click to confirm the extracting)
- : Resample current matrix by using chosen interpolation method (Figure 13). The user can specify sampling factors in x, y and z direction for 3D matrix.

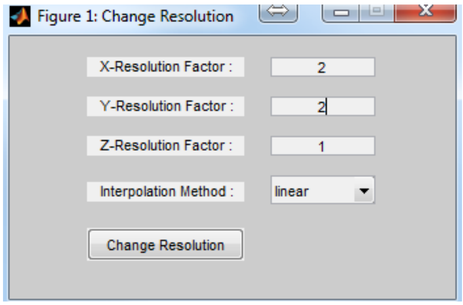


Figure 1: Change Resolution

X-Resolution Factor : 2

Y-Resolution Factor : 2

Z-Resolution Factor : 1

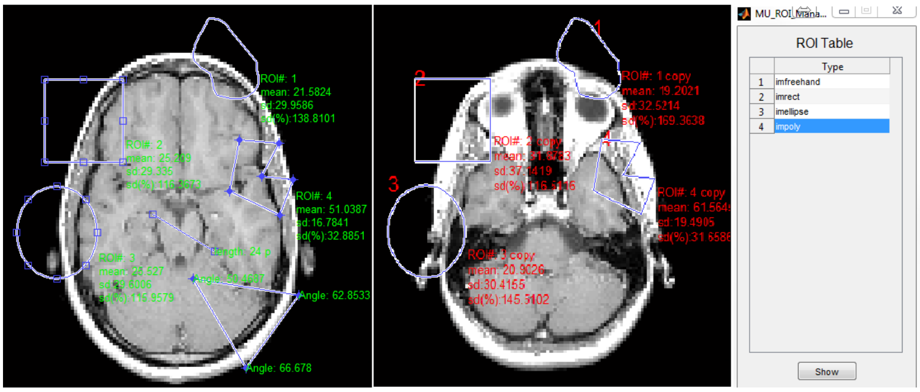
Interpolation Method : linear

Figure 13: Interpolation Window. An example is shown for doubling resolution in x and y direction by using linear interpolation.

5. ROI

MatrixUser provides a set of function buttons for performing Region-of-Interest (ROI) analysis (Figure 14). To create a ROI, the user needs to click ROI button first, then draw a ROI on the image axes. The statistical measures (i.e. mean, standard deviation and relative standard deviation) for voxels in delineated ROI is calculated and updated with moving ROI position or changing ROI shape. The ROI function buttons consists of

- : Draw a free hand ROI
- : Draw a rectangle or square ROI
- : Draw a circle or ellipse ROI
- : Draw a polygon ROI
- : Draw a straight line for measuring distance in units of pixels
- : Draw a polygon for measuring the interior angles in degrees
- : Record existing ROI shape and location into a ROI list, allow to redraw selected ROI in multiple image axes. To redraw a existing ROI, the user needs to select a ROI in the list, then press 'Show' button. Note that the copied ROI is no longer resizable and movable.
- : Plot histogram for current slice (Figure 15); if ROIs exist, plot histogram for latest activated ROI



ROI Table

	Type
1	imfreehand
2	imrect
3	imellipse
4	impoly

Figure 14: Draw Multiple ROIs and Redraw on A Second Image. The source ROIs are in green, copied ROIs are in red.

6. Segment

MatrixUser supports functions for performing manual segmentation, editing, saving and loading segmented regions for multiple slices. To create a segmentation, the user needs to click segmentation button first, then draw

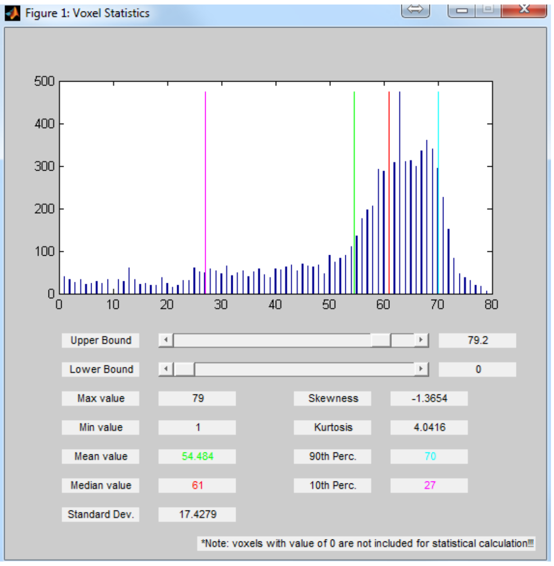


Figure 1: Voxel Statistics

Upper Bound 79.2

Lower Bound 0

Max value 79

Min value 1

Mean value 54.484

Median value 61

Standard Dev. 17.4279

Skewness -1.3654

Kurtosis 4.0416

90th Perc. 70

10th Perc. 27

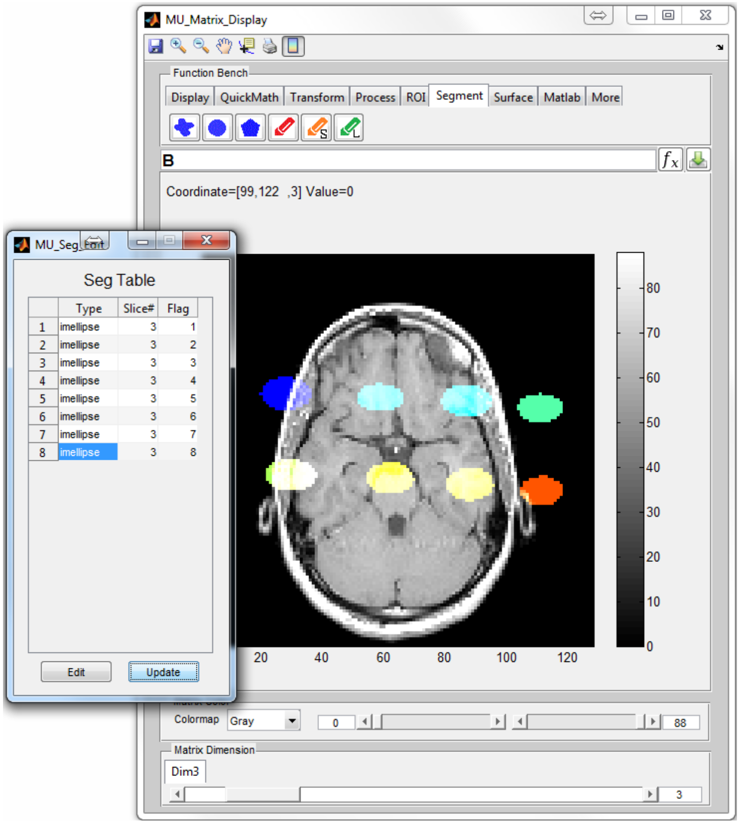
*Note: voxels with value of 0 are not included for statistical calculation!!!

Figure 15: Histogram for One Image Slice

a region on the image axes. After finishing drawing, the user can modify the region location and shape before double clicking the region to confirm segmentation. The segmentation buttons consists of

- ✚ : Do a free-hand segmentation
- ● : Do a circle or ellipse segmentation
- ⬟ : Do a polygon segmentation
- ✎ : Edit segmentation
- ✎S : Save segmentation into a MAT file
- ✎L : Load segmentation from a MAT file

To edit segmented region (Figure 16), the user needs to press ✎ to open segmentation manager. The manager will automatically detect the type and location of segmented regions. The user can click any region item to inspect the location of the region. To edit chosen region, the user must click 'Edit' button to activate region outline. Both the shape and mask flag are editable for segmented region. After editing, the user must click 'Update' to conform modification. The user can press ✎S to save current segmentation into a MAT file which contains a mask matrix with the same size of the original matrix and a cell array storing segmentation location information. If only the mask matrix is desired, the user can



B

Coordinate=[99,122 ,3] Value=0

Colormap Gray 0 88

3

Seg Table

	Type	Slice#	Flag
1	imellipse	3	1
2	imellipse	3	2
3	imellipse	3	3
4	imellipse	3	4
5	imellipse	3	5
6	imellipse	3	6
7	imellipse	3	7
8	imellipse	3	8

Figure 16: Editing Segmentation

simply press to save it into workspace. The user can press to load previous segmented regions from saved MAT file. Notice that the user can press to remove segmentation from overlapping with background matrix.

7. Surface

   This function category generates surface or mesh plot for current image.

   - : Create contour plot of current matrix
   - : Create 3D contour plot of current matrix
   - : Create filled 2D contour plot
   - : Create 3D shaded surface plot
   - : Create surface plot and contour
   - : Create surface plot with colormap based lighting
   - : Create mesh plot
   - : Create mesh plot and contour
   - : Create a curtain around a mesh plot
   - : Create a ribbon plot
   - : Create a waterfall plot
   - : Create pcolor (checkerboard) plot

8. Matlab

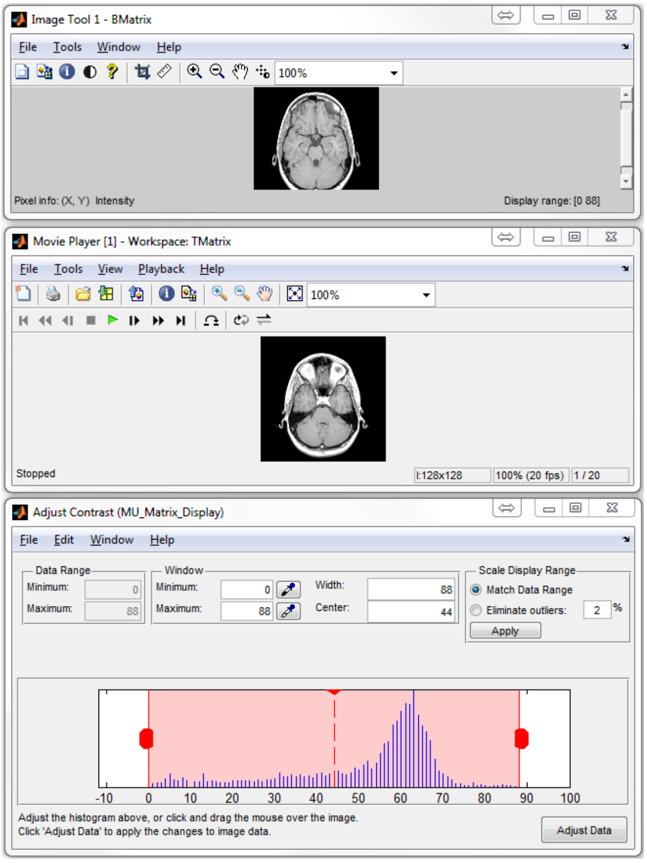
   Matlab default image tools (Figure 17) are tailored for MatrixUser and included in this category.

   - : Perform imtool for current image
   - : Perform immovie for playback current 3D matrix
   - : Perform imcontrast for adjusting image contrast

9. More

   Uncategorized functions are categorized under this tab.

   - : Create a 3D graph for rendering current matrix (Figure 18)



Pixel info: (X, Y) Intensity

Display range: [0 88]

Stopped

I:128x128 | 100% (20 fps) | 1 / 20

Data Range

Minimum: 0

Maximum: 88

Window

Minimum: 0

Maximum: 88

Width: 88

Center: 44

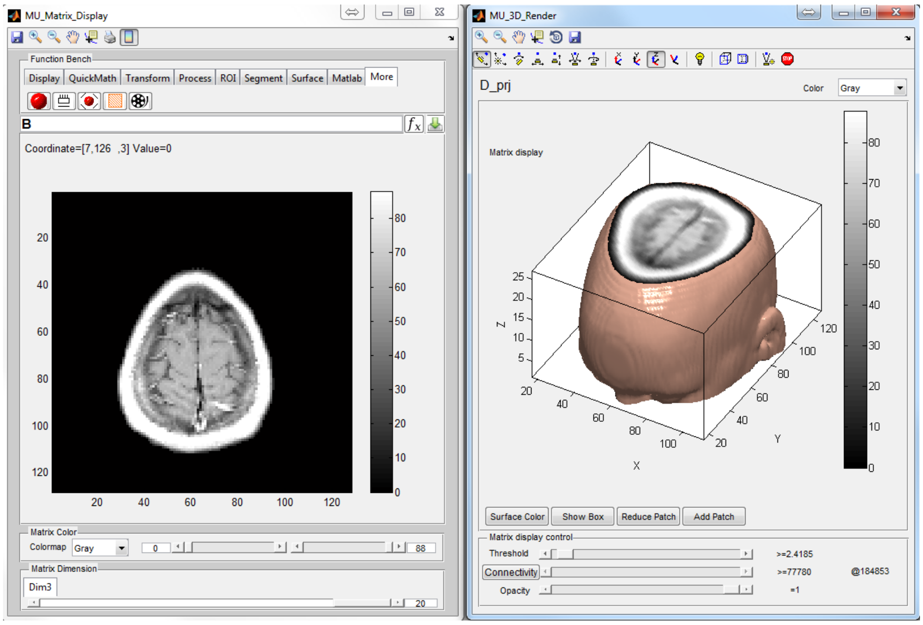
Scale Display Range

● Match Data Range

○ Eliminate outliers: 2 %

Adjust the histogram above, or click and drag the mouse over the image.
Click 'Adjust Data' to apply the changes to image data.

Figure 17: Matlab Tools



B

Coordinate=[7,126 ,3] Value=0

Matrix Color

Colormap Gray 0 88

Matrix Dimension

Dim3 20

D_prj

Color Gray

Matrix display control

Threshold >=2.4185

Connectivity >=77780 @184853

Opacity =1

Figure 18: An Example of 3D Human Brain Rendering. Control units on the rendering window are provided for fine tuning the renderer. The user can select isosurface threshold, cutoff connectivity threshold (i.e. object with total voxels less than the threshold will be removed from the rendering. '@' is followed by the voxel number of current largest object) and object opacity. A set of pushbuttons are also available for changing surface color, display box and patches. The default Matlab camera toolbar are provided for adjusting the lighting effect.

- : Perform projection along any given matrix dimension. Support multi-dimensional matrix projection
- : Perform 3D projection along x, y or z axis with certain angle increment (Figure 19)

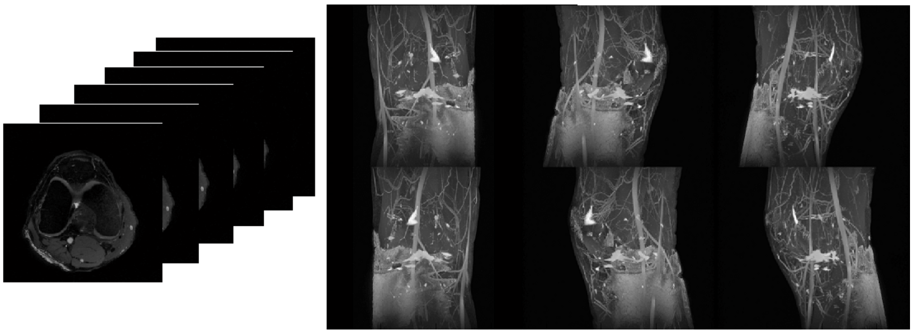
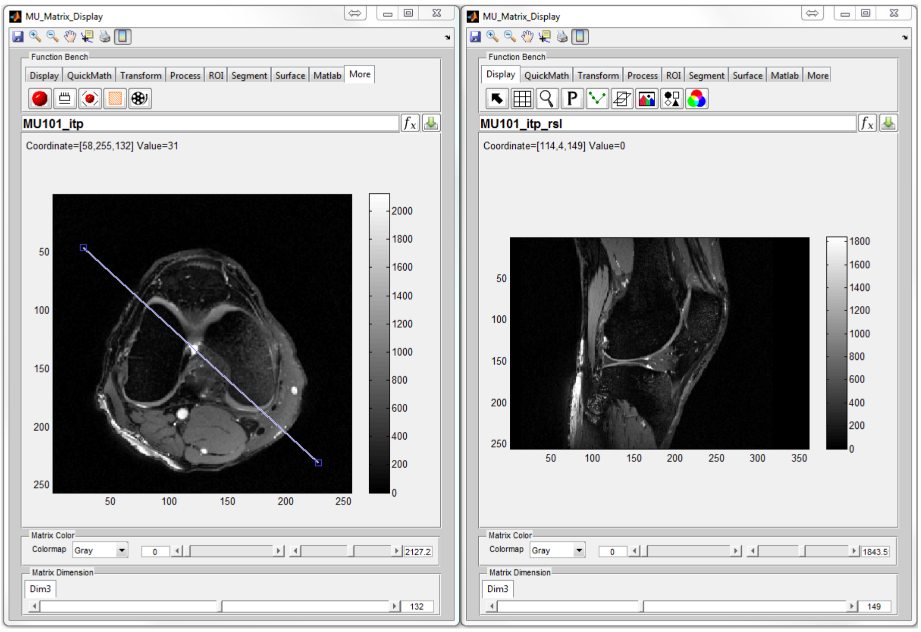


Figure 19: 3D Maximum Intensity Projection (MIP). An example of 3D MIP around axial axis of a human knee MRI image stack shows the vascular system of the knee joint.

- : Reslice 3D matrix at given direction. The user needs to draw a line and double click to confirm for indicating the reslicing direction (Figure 20)
- : Create a movie using current matrix display. Support making movie for overlapped matrices.



MU101_itp

Coordinate=[58,255,132] Value=31

Colormap Gray 0 2127.2

132

MU101_itp_rsl

Coordinate=[114,4,149] Value=0

Colormap Gray 0 1843.5

149

Figure 20: Reslice 3D Matrix. An example of 3D reslicing generates a new stack of images in the oblique plane from an axial human knee MRI image stack. Note that the resliced images are extracted from the plane perpendicular to the indicating line on the left window.