Matlab Utility Pack 1
http://www.DIR-lab.com
Available for download: 7/6/09

Included software provides for the following:

- read raw images into Matlab workspace
- write image variables in current workspace to file in raw image format
- image viewer for scrolling 3D image volumes in axial, sagittal, or coronal views.

This software package includes basic utility functions for manipulating 3D raw image volumes in Matlab. The included functions *require* the Matlab Image Processing Toolbox, and should be added to the Matlab path prior to use. This can be done from the Matlab interface through the "File" menu, under the "Set Path..." option. The included applications may or may not require support functions to operate properly (indicated below). The support functions should not be called directly, but must accompany the parent function in the Matlab path. All included function files are saved in the locked Matlab *.p format. Thus, the actual code may not be edited or reviewed.

A brief description of the included contents is provided below. As always, please direct any questions, comments, or bug reports to Inquiries@DIR-lab.com.

Content:

- ImageImprt.p
- ImageImprtInterface.p
- WriteImage.p
- ImageView3d.p
- ImageView3dExt.p
- MatlabUtilityPack1.pdf

The specific functions are described below:

ImageImprt.p: General interface function for importing raw image files into Matlab workspace. The interface may be used for any data type, size, or byte ordering, all of which are specified by the user. No header or image information files are required.

function call: `>>  ImageImprt;`
required support functions: ImageImprtInterface.p

WriteImage.p: Write Matlab variable to file in current directory. Image variables are written in raw image format, with *.img extension.

ImageVariable: Image variable to write to file
PrecisionString: 'uint16', 'int16', or 'float32'
FileName: String indicating name of output file, no extension

function call: `>>  WriteImage(ImageVariable,'PrecisionString','FileName');`

required support functions: none

ImageView3d.p: General interface function for viewing 3D image volumes in Matlab. The images to view must exist as variables in the current Matlab workspace. The interface allows for 2D scrolling through the 3D volume in either axial, sagittal, or coronal view, with basic zoom and pan functionality. Edit boxes allow the user to specify voxel dimensions, in order to affect the aspect ratio of the image display.

There is also the option to import a formatted list of landmark coordinate positions for overlay of the landmarks on the image display. Landmark positions must be contained in a text file, with one XYZ position on each row (single tab separating columns). Once imported, landmark positions appear as green crosshairs in the image display.

The display window and leveling is controlled via the built-in Matlab function `imcontrast`. It is called through the interface via a toggle button. Activate the toggle to bring up the image contrast interface. Select the contrast toggle again to deactivate the contrast interface; do not close it directly. Also, there is no need to hit the 'Adjust Data' button in the contrast interface, the display range will change automatically as it is manually adjusted. Do to issues regarding backwards compatibility, the contrast tool is only available if the ImageView3d interface is running in Matlab releases: 2006b, 2007a, or 2008b. For all other releases, the contrast toggle button will be disabled.

A pixel information tool is also provided, which actively displays the current 2D coordinates of the cursor when it is positioned within the image display. Note that the XY position is relative to the current display orientation. As with the contrast tool, the pixel information tool is only activated when running Matlab releases: 2006b, 2007a, or 2008b.

Only one instance of the ImageView3d interface may be run at a time. However, a menu option is provided to open a second display window so that two image volumes may be manipulated and viewed simultaneously. A selection box prompts the user to indicate which variable in the current workspace to visualize in the second display. All functions, including landmark visualization, are available for both display windows. The second display may be closed and re-opened with a new variable at any time. However, closing the main interface display (Figure A) will terminate the application.

There is a 'Link Landmarks' menu option which allows corresponding landmark positions to be viewed in the two display figures. This feature requires that the imported landmark lists are the same size.

ImageVariable: Image variable in current workspace to view

function call: `>>  ImageView3d(ImageVariable);`
required support functions: ImageView3dExt.p