IMAGINE User's Manual

IMAGe visualization, analysis and evaluation engINE

11/04/14

Table of Contents

Command Line Interface

This is a repetition of the header of the IMAGINE.m file.

`IMAGINE`
starts the IMAGINE user interface without initial data

`IMAGINE(DATA)`
Starts the IMAGINE user interface with one (DATA is 3D) or multiple panels (DATA is 4D).

`IMAGINE(DATA, PROPERTY1, VALUE1, ...)`
Starts the IMAGINE user interface with data DATA plus supplying some additional information about the dataset/GUI in the usual property/value pair format. Possible combinations are:

PROPERTY	*VALUE*
'Name'	String: A name for the dataset
'Voxelsize'	[3x1] or [1x3] double: The voxel size of the first three dimensions of DATA.
'Units'	String: The physical unit of the voxelsize (e.g. 'mm')
'Window'	[2x1] of [1x2] double: Initial [min max] values for the image brightness.
'Mask'	Logical mask, must have same size as the corresponding image data
'Panels'	[1x2] integer: Determines the initial GUI layout

IMAGINE(DATA1, DATA2, ...)
Starts the IMAGINE user interface with multiple panels, where each input can be either a 3D- or 4D-array. Each dataset can be defined more detailed with the properties above.

Examples:

```
>> load mri % Gives variable D
>> imagine(squeeze(D)); % squeeze because D is in rgb format

>> load mri % Gives variable D
>> imagine(squeeze(D), 'Name', 'Head T1', 'Voxelsize', [1 1 2.7]);
```

Graphical Interface

Menubar

ctrl + o **Open File**[1] Opens a standard file open dialog which can be used to open the following file types:

*.mat	Matlab files. IMAGINE automatically looks for matching variables in the files (at least 2D and numeric or logical). So far cells and structs are not supported. If multiple matching variables are found, a selection dialog is shown.
*.nii	NifTy files. So far, only a maximum of 4 dimensions is supported. The resulting series should be displayed with correct aspect ratio and physical units.
*.gipl	GIPL files. The resulting series should be displayed with correct aspect ratio and physical units.
DICOM	DICOM images. Since I've seen a broad range of possible extensions and cryptic filenames with more dots in it than anything else (thank you Siemens), I decided to check all files which do not have a known file extension for DICOM contents. The files are sorted into 3D sets according to the *SeriesInstance* property. The resulting series should be displayed with correct aspect ratio and physical units.
Image files	Any file type that can be read using MATLAB's `imread` method. Images are sorted into series according to image dimension.

Right-click: Opens a folder open dialog to load all files in a folder.

Cntl + i **Import Workspace Data**[1] Allows importing variables from MATLAB's base workspace. Other workspaces (e.g. workspaces of functions in debug mode) are currently unsupported.

Ctrl + s **Save** Save the contents of the selected panels to image files (those supported by imwrite). Use the fieldnames %SeriesName% and %ImageNumber% in the filename which are automatically replaced with the corresponding variables to obtain different filenames for each image.
Right click: Save the current mask (e.g. ROI or VOI) to files. If the mask is 2D (ROI, livewire), only the mask slice is saved. If the mask is 3D (region growing or isocontour), the whole stack is saved.

Del **Delete** Clears the content of the selected panel(s).

Ctrl + x **Exchange** Exchanges the contents of the two selected panels. Only available if exactly two panels are selected.

Grid Opens a pop-up window that allows choosing the amount of panels and the layout.

Colormap Chose between different predefined or custom colormaps. See below for further information about implementing your own colormaps.

---

[1] Note that adding data to the GUI doesn't change the panel layout, i.e. in order to see the newly added data, one has to increase the amount of panels or scroll through the data using the keyboard.

Ctrl + l **Link Actions** A toggle button that controls the UI's response to user input. When activated (default) all operations (e.g. scrolling, zooming, ...) are applied to all visible panels. When inactive, actions are only applied to the selected panels. If no panel is selected, mouse actions are applied to the panel over which the mouse cursor is located.

Link Windowing A toggle button to couple the brightness/contrast values of the visible data series. If active, the settings of the first visible series are re-used for all remaining series.

Ctrl + 0 **Reset View** Resets the zoom level to original size, windowing to full dynamic range and centers the images in their panels.

Phase Image Shows the argument of complex data. If the data is real, the corresponding panels will show all zeros. Evaluation functions will also work with this data. Click again to return to intensity (real-valued data) or magnitude (complex data) mode.

Maximum Intensity Projection Shows a maximum intensity projection across the current 3rd dimension of the data. Evaluation functions will also work with this data. Scrolling/changing the current image will be disabled. Click again to return to intensity (real-valued data) or magnitude (complex data) mode.

Minimum Intensity Projection Shows a minimum intensity projection across the current 3rd dimension of the data. Evaluation functions will also work with this data. Scrolling/changing the current image will be disabled. Click again to return to intensity (real-valued data) or magnitude (complex data) mode.

Cntl + r **Start Logging Evaldata** This button allows you to start logging the evaluation results (from lines, ROIs and VOIs) to a comma separated spreadsheet file, which can be imported into Excel. On clicking, the desired target file can be selected. Note: You can continue logging into a specific file at all times, even if the function had been deactivated or logging to a different file has been done in between. Just re-open the file and ignore MATLAB's standardly set dialog that the file will be replaced – it will not! *However, IMAGINE will not check whether the information in the first few lines of the csv file (e.g. series names, evaluated measures, etc.) still match the current settings!*

Alt + r **Stop Logging Evaldata** This button stops the logging. Also use this button if you want to start logging into a new target file (the start logging button is only available if the logging has been stopped).

Cntl + z **Undo the last evaluation** Removes the last evaluation dataset from the evaluation file.

Cntl + t **Timeseries Mode** This toggle button controls the behavior of the evaluation tools, the line plots (see below) and the layout of the eval file. Generally speaking, it is useful to turn on this switch if IMAGINE contains lots of series, i.e. if data was acquired over a longer time resulting in multiple 2D or 3D datasets. If switched on, the evaluation will be performed on all series, regardless if currently visible or not (IMAGINE can only display 16 series max simultaneously). If off, only visible series are evaluated. If timeseries mode is on, the evaluation results are written into a new line of the evaluation file for each data series. If off, the data from the visible time series are written next to each other, such that on evaluation operation always results in one new line in the evaluation file. I guess, you probably try this out and see for yourself. See below for the different behavior of the line plots depending on this button.

Cntl + w **Show Lineplots** A toggle button indicating whether line plots are shown after evaluation. If a line plot is shown also depends on the timeseries mode and the selected tool. If timeseries mode is on, then all eval tools will plot the results of the eval functions over the series index for all series (even if not visible). If the timeseries mode is inactive, only the line tool will cause the plot window to

appear. In this case, simply the line profiles, i.e. intensity over location, are shown.

Toolbar

The toolbar lets you select one of five different tools which are described in the following.

m

Pointer Tool There are two types of operations: *clicking*- and *dragging* operations. *Click* the draw panels to **select/deselect** the corresponding panels or use in conjunction with the shift or control key to select multiple draw panels similar to selecting files in a file explorer.
Three types of *dragging* operations can be used: **Move** the images in their draw panels using the left mouse button, **zoom** images using the right mouse button and **window** (adjust brightness/contrast) the images using the middle mouse button.

r

Rotation Tool Left-click and drag the mouse cursor in any direction. After a certain distance the image will rotate around the first MATLAB dimension (if dragging left/right) or the second MATLAB dimension (if dragging up/down). The dragging start point will become the current slice. Right-click and drag to perform an in-plane rotation of the data (rotate along MATLAB's 3rd dimension). With this tool selected you can still perform the **window** function just like using the pointer tool.

l

Line Profile Evaluation Tool Use this tool to draw a profile line. On releasing the mouse button, the underlying image data of all panels will be interpolated along the lines and sent to the line evaluation function. Interpolation is scaled such that the length of the data vector equals the length of the line in the physical unit of the series (e.g. px or mm).
Right-Click: The line evaluation functions can be selected by right-clicking the icon. See the section *Writing Evaluation Functions* to find out how to implement evaluation functions for your needs.

o

ROI Evaluation Tool Use this tool to create a 2D ROI. The V 2.0 version has three different modes:
Rectangle: Press, hold and drag using the *left mouse button* to draw a rectangular ROI.
Circle/Ellipse: Press hold and drag using the *right mouse button* to draw an ellipse, or, if you press the *shift key* before pressing the mouse button, to draw a circle.
Polygon: Click and release the left mouse button to start a polygon or adds points to an existing polygon; right-clicking deletes the last polygon point. Close the polygon by either double-clicking or middle mouse button. Upon closing the polygon, the data will be sent to the ROI evaluation functions.
Right-Click the Icon: The 2D evaluation functions can be selected by right-clicking the icon. See the section *Writing Evaluation Functions* to find out how to implement evaluation functions for your needs.

w

Livewire Tool This tool lets you draw a ROI using the livewire algorithm. It automatically guides the delineation along image gradients and edges. Apart from this, same basic behavior as the polygon ROI tool. This tool may be unavailable if the compilation of the mex files failed. See the mex troubleshooting section.
Right-Click: The 2D evaluation functions can be selected by right-clicking the icon. See the section *Writing Evaluation Functions* to find out how to implement evaluation functions for your needs.

g **3D Region Growing** A region growing algorithm which generally operates on all three dimensions, but can also be applied to 2D data. The region growing tries to automatically determine a reasonable stopping criterion. The tolerance can be adjusted by *right-clicking in the panel*, which brings up a context menu. This tool may be unavailable if the compilation of the mex files failed. See the mex troubleshooting section.
Right-Click: The 3D evaluation functions can be selected by right-clicking the icon. See the section *Writing Evaluation Functions* to find out how to implement evaluation functions for your needs.

i **3D Isocontour Region Growing** A 3D isocontour algorithm which returns the 50 % isocontour volume around a seed point. Also works on 2D data. The region is grown as long as a candidate voxel has at least 50 % the intensity of the brightest voxel in the region. Start by clicking near a high intensity voxel to prevent unwanted/unreasonable behavior. This tool may be unavailable if the compilation of the mex files failed. See the mex troubleshooting section.
Right-Click: The 3D evaluation functions can be selected by right-clicking the icon. See the section *Writing Evaluation Functions* to find out how to implement evaluation functions for your needs.

p **Tag Tool** Use this tool to **rename** panel data, specify the voxel size (aspect ratio) and the physical units.

Misc Keyboard Functions

Besides the shortcuts shown in the tables above, there are some keyboard functions for scrolling images and series and for changing the tools. Use the **up/down arrows** to navigate through the series and the **left/right arrows** to scroll through the image stacks. Those of you who are used to Siemens medical imaging consoles may find it convenient that the same functionality is found on the number block: Use the **4/5 keys** to navigate through the series and the **1/2 keys** to scroll through the image stacks. Use the **spacebar** to cycle through the tools. The cycling order is backwards if the shift key is pressed simultaneously.

Writing Evaluation Functions

With IMAGINE 2.0, the interface for line and ROI evaluation was unified. For convenience, these functions can be written in dedicated m-files and stored in the subdirectory "EvalFunctions". The active evaluation function (i.e. the function that will be executed after having defined a line or ROI in the IMAGINE UI) can be set by right-clicking the corresponding tools. Each tool category (1D, 2D, 3D) have their own set of evaluation functions. This means (as of V 2.0) that the two 2D tools ROI and livewire share such a set.

With version 2.0, I introduced a much more convenient syntax for the evaluation functions. Much more functionality (e.g. plotting) is done by IMAGINE itself and has not to be implemented in the eval functions anymore. Also the loop over the data is handles by IMAGINE such that the eval function only has to carry out the calculations for a single dataset.

The IMAGINE UI now calls a line evaluation function with the following syntax:

```
[dDataOut, sName, sUnitFormat] = LINEEVALFCN(dData);
```

The first input argument `SData` is a vector of type double and contains all samples along the line profile or in the ROI/VOI.

The first output argument `dDataOut` is a double scalar and represents the result of the eval operation. To implement e.g. a maximum operation, simply use `dDataOut =`

`max(dData);`.The other two output arguments supply some additional information for displaying porpose: String `sName` simply describes the function's purpose (e.g. `sName = 'Max'`). The string `sUnitFormat` is a bit more tricky: It is a printf-style format string to represent the unit of the function's result. The `sUnitFormat` string is used by IMAGINE as the first input argument to a sprint call, where the second argument is the physical unit string of the data (e.g. 'px' or 'mm'). If the function's result is unitless, simply return `sUnitFormat = ''`. If it returns a distance, (e.g. in the FWHM evaluation), set `sUnitFormat = '%s'`. If the function calculates the slope steepness of a gradient, set `sUnitFormat = '1/%s'` to print the reciprocal unit.

See the supplied function *fFWHM* for an example evaluation function.

Implementing Custom Colormaps

To use your own customized colormap, simply create a function with the following interface:

```
Function dColormap = COLORMAPFCN(N);
```

This function should return a colormap array dColormap of size Nx3 with values in [0 1] and the column vectors representing red green and blue (just like the built-in colormap functions). Place this file in the sub-folder *colormaps* and and IMAGINE will automatically add it to the colormap options.

Mex Troubleshooting

The livewire, region growing and isocontour tools are implemented as MATLAB-compatible c++ or *mex* files due to their heavily iterative nature. At first start, IMAGINE tries to compile these files automatically. If this fails, a warning will be generated and the affected tools will not be available. If you face this problem, you may not have set up your mex compiler yet. You can try to do so by typing `>> mex -setup` in the MATLAB console upon which MATLAB tries to locate c++ compilers on your system. It should then list the available compilers and let you choose one. To retry compiling the c++ files, delete the imagineSave.mat file in the IMAGINE folder, which resets all the settings and convinces IMAGINE that it is being called for the first time. Thus, it will try to compile the files. If it still fails, you may have to get another c++ compiler - sorry.

Contribute Yay!

Wanna make IMAGINE better? I see two ways of contributing to that. First, send your feedback/whishes/bug reports to christian.wuerslin@med.uni-tuebingen.de. Second, if you wrote an evaluation function that you think is awesome and you would like to share with the world, send it to the same address and I will incorporate it into the next release.

Revision History

V 1.0	02/21/13	Initial Version
V 1.1	02/25/13	- Auto-add eval function subdirectory to MATLAB path - Added tooltips
V 1.2	03/20/13	- Changed GUI architecture (now uses nested functions). - A lot of comments throughout the code. - Fixed a bug that occurred when deleting data. - Image value display automatically switches to exponential representation when showing values < 0.01.

		- Colors of the line/ROI evaluation functions now follow the MATLAB color order for better visual discrimination. - Rotation tool is now more robust. - Line profiles show captions. - New syntax can return the axes handles thus allowing the user to add plots to the axes. - Panel data can be exchanged.
V 1.3	04/03/13	- Added save function - Added App
V 1.4	04/24/13	- Added colormaps - Added colorbar option - Improved zooming operation - Added evalbar option
V 2.0	10/20/14	- New rendering engine for better contrast - Supports mask overlays - Import mat, NifTy, GIPL and DICOM images - Supports anisotropic data and physical units - Export evaluation results to csv files on the fly - Easier eval function concept - New ROI features (boxes, ellipses, circles) - Livewire ROI evaluation - Region growing VOI evaluation