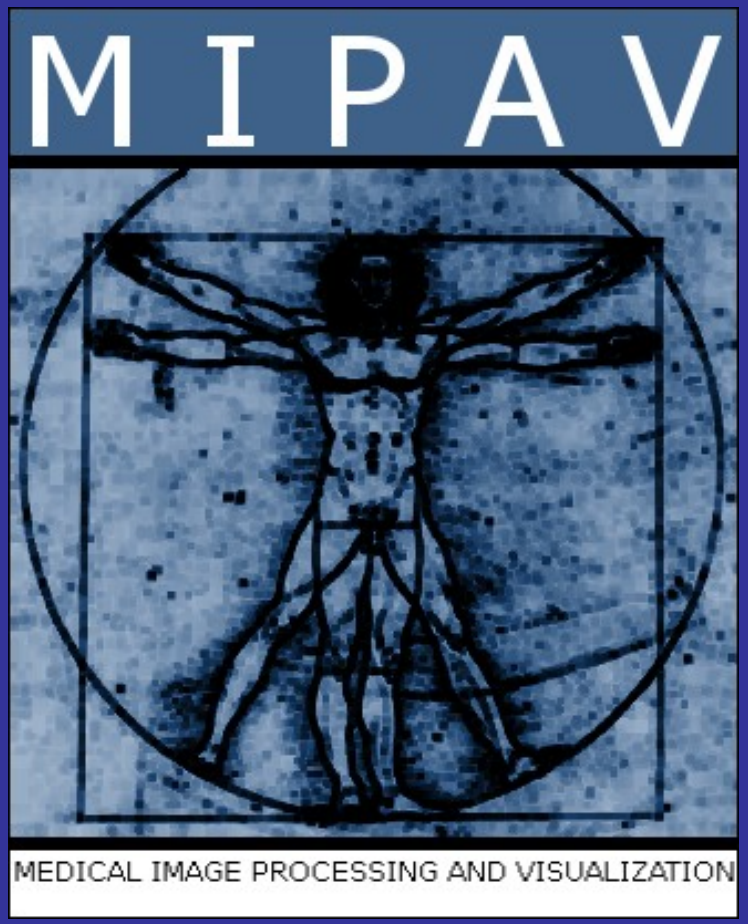


http://mipav.cit.nih.gov

Scripting MIPAV and the Java Image Science Toolkit

Evan McCreedy

email: mccreedy@mail.nih.gov

(301) 496-3323

Biomedical Image Processing Research Services Section (BIRSS)

http://mipav.cit.nih.gov

MIPAV Team

Employees

Ruida Cheng

William Gandler

Matthew McAuliffe

Evan McCreedy

Justin Senseney

Fellows

Sara Shen

Contractors

Alexandra Bokinsky, Geometric Tools Inc. (Visualization)

Olga Vovk, SRA International Inc. (Technical Writing)

Alumni

Paul Hemler, Agatha Munzon, Nishith Pandya,

Beth Tyriee, Hailong Wang

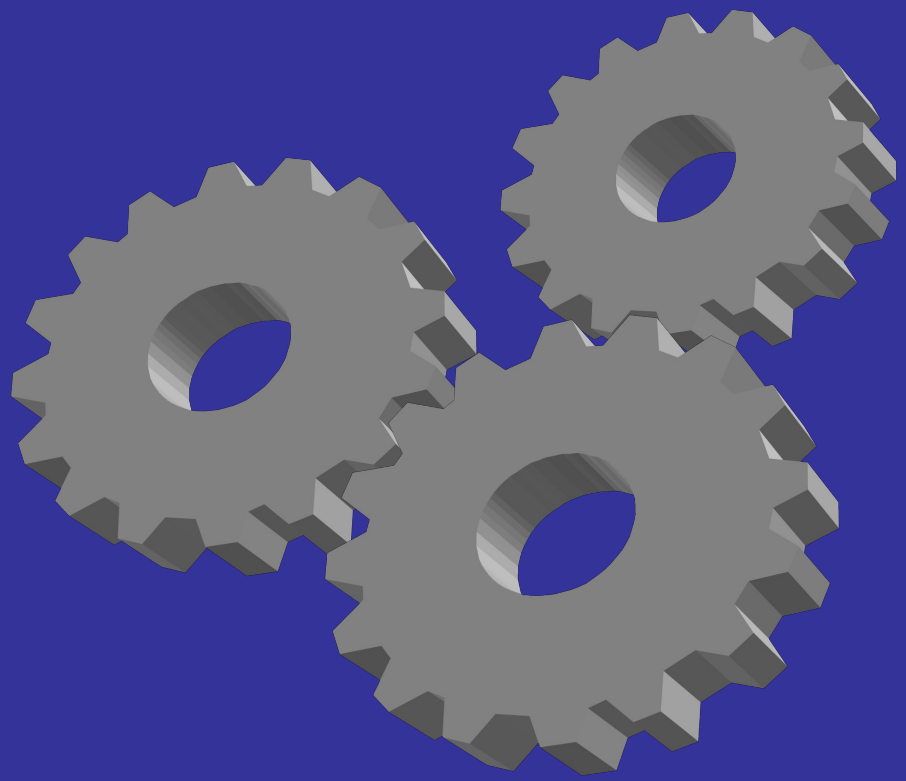
Extensibility
Plugins and Scripts

- **Plugins**
  - Written in Java using the MIPAV API.
- **Scripting**
  - Use MIPAV to record and save function(s) applied to image dataset(s)
  - Apply the script to any number of image datasets using the ‘Run script’ dialog.
  - Apply the script to any number of image datasets from the command line.

Usability

- **Most NIH intramural researchers not interested in software-development:**
  - **they want a GUI for interactive work**
- MIPAV provides:
  - a **GUI driven application** for user-interaction;
  - **scripts** to automate repetitive functions;
  - **plugins** for customized functionality;
  - **command line input** for batch processing.

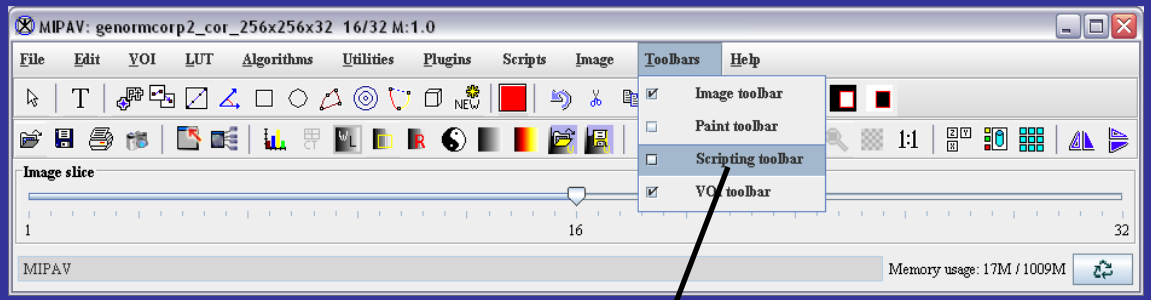
Scripting

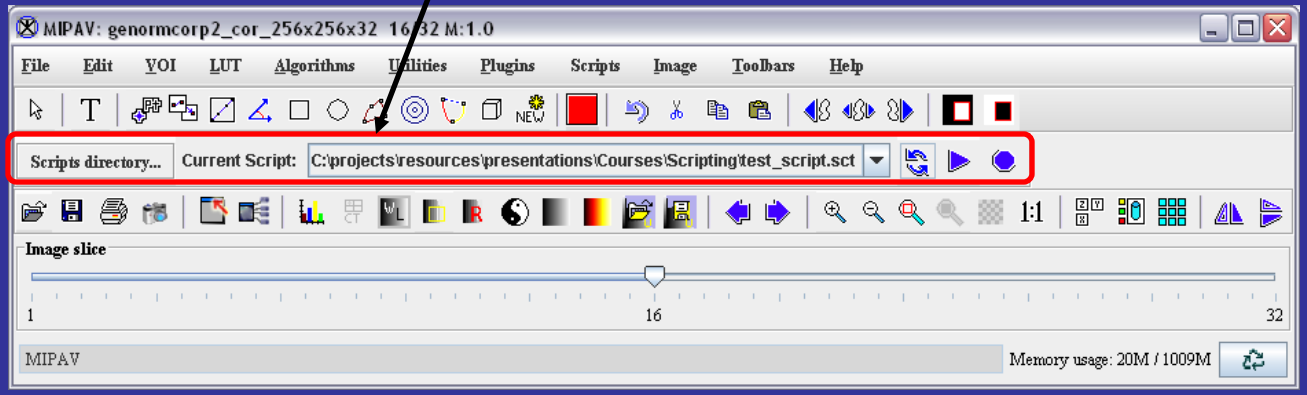


Scripting (Macros)

- Automation of functions applied to a group of datasets
  - Increase efficiency and productivity
- MIPAV does NOT require learning a new scripting language.
  - Record and execute scripts via MIPAV's GUI.
  - Scripts can also be executed from the command line to process images from another program or script.

Scripting


MIPAV: genormcorp2_cor_256x256x32 16/32 M:1.0
File Edit VOI LUT Algorithms Utilities Plugins Scripts Image Toolbars Help
Image toolbar
Paint toolbar
Scripting toolbar
VOI toolbar
Image slice
1
16
32
MIPAV
Memory usage: 17M / 1009M

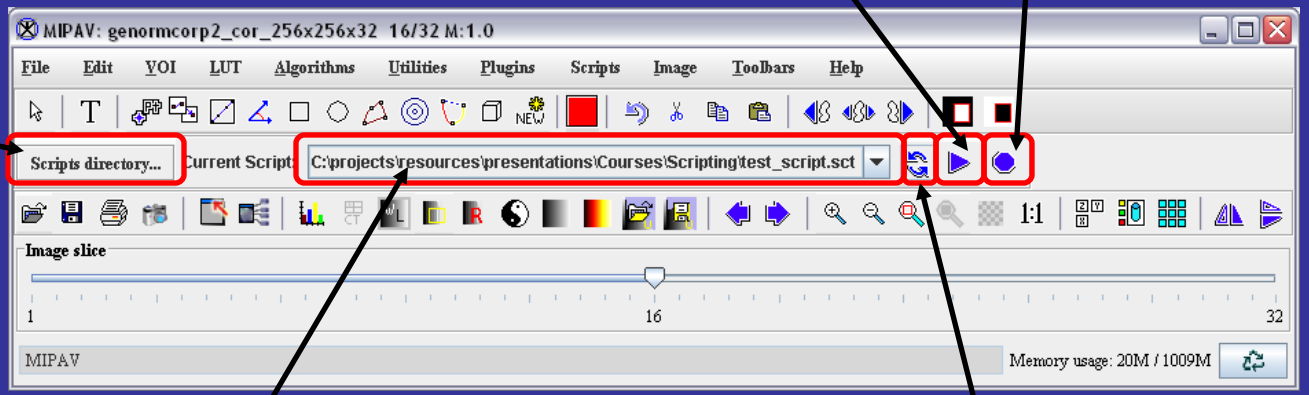

MIPAV: genormcorp2_cor_256x256x32 16/32 M:1.0
File Edit VOI LUT Algorithms Utilities Plugins Scripts Image Toolbars Help
Scripts directory...
Current Script:
C:\projects\resources\presentations\Courses\Scripting\test_script.sct
Image slice
1
16
32
MIPAV
Memory usage: 20M / 1009M

Scripting

Run script

Record script

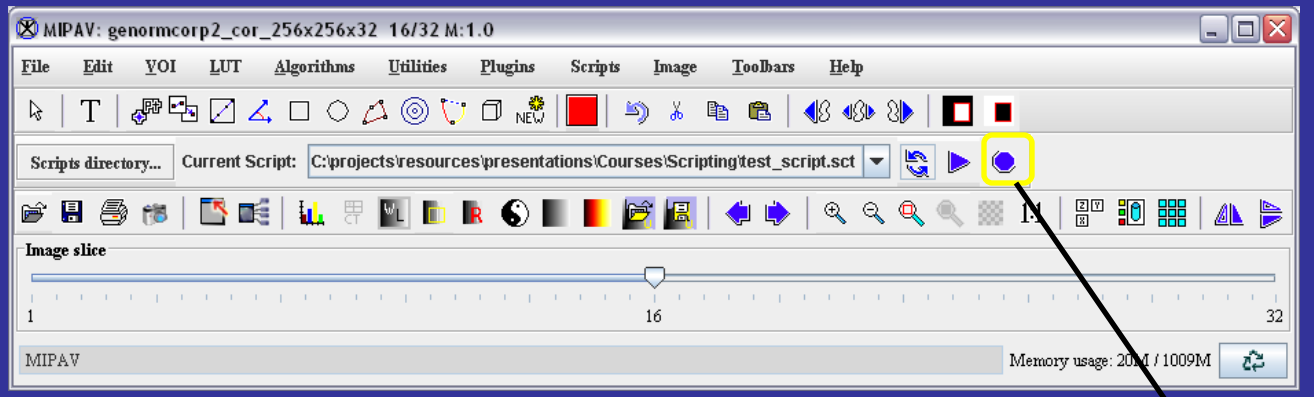
Allows the user to select the directory used for scripts

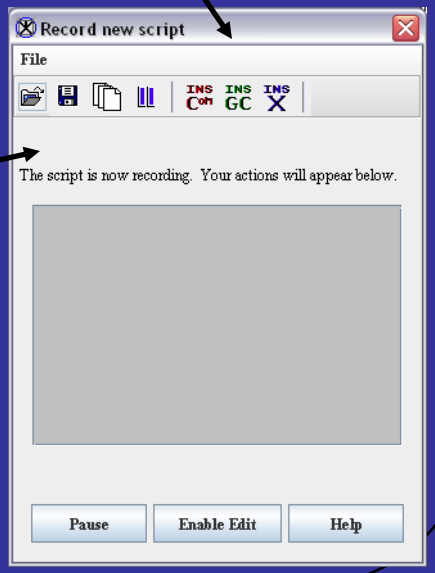


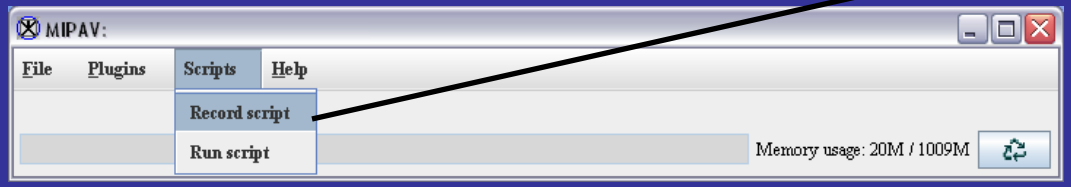
List of scripts found in the script's home directory. The usual suffix is ".sct".

Refresh the list of scripts.

Script Recording







Recording A Script

Inserts a comment into the script

Inserts a command to recover memory

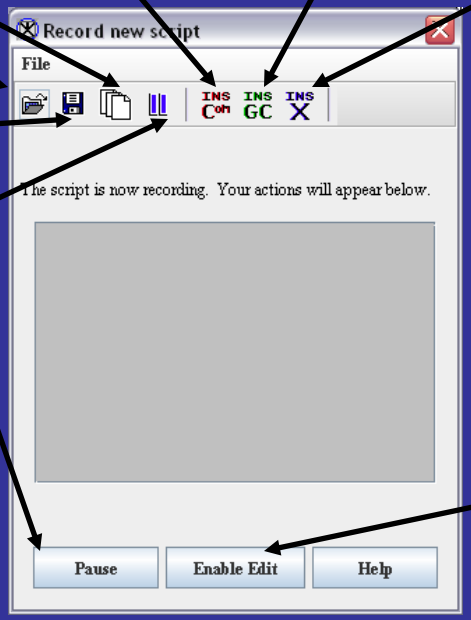
Resets the script recorder

Inserts an exit command to quit MIPAV. Used in scripts that will be run from the command line.

Opens script file

Saves script file

Pauses script recording



Allows manual editing of the script. For advanced users.

Scripting Features

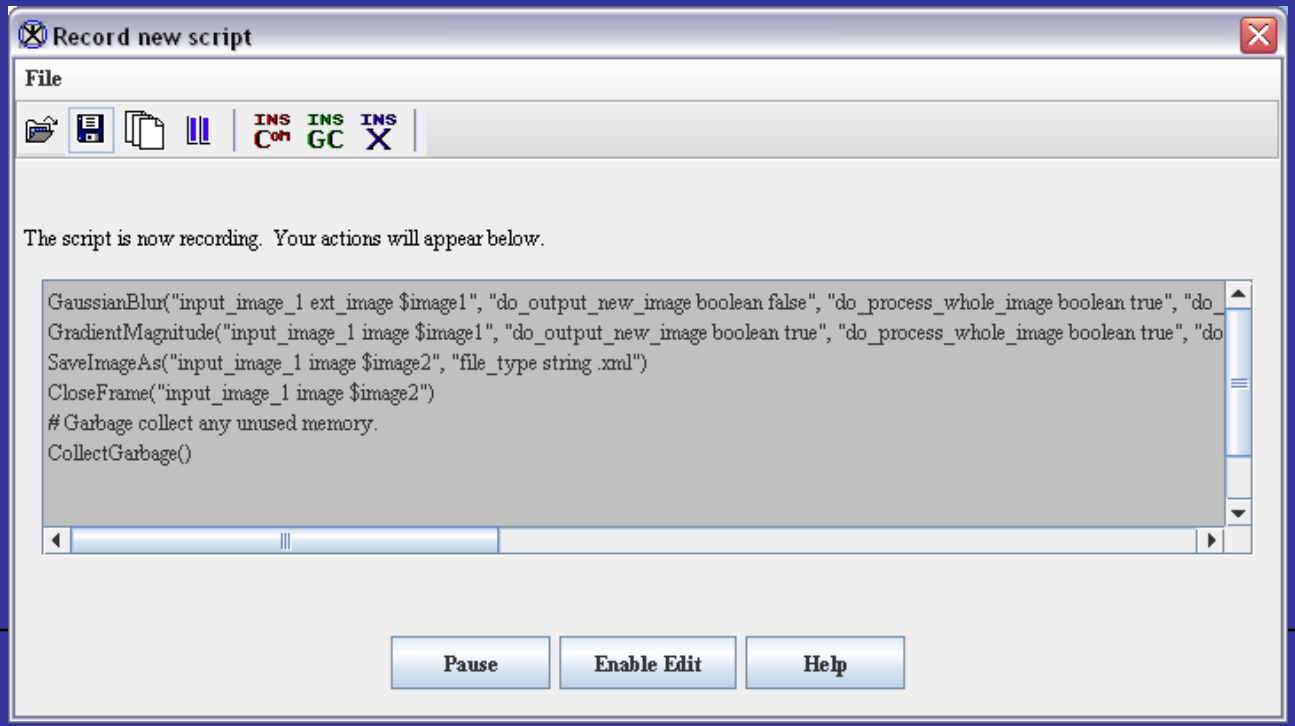
- Arbitrary ordering of parameters allowed
- Parameter labels to facilitate parameter editing
- Script execution dialog
- More robust error messages

Script Example

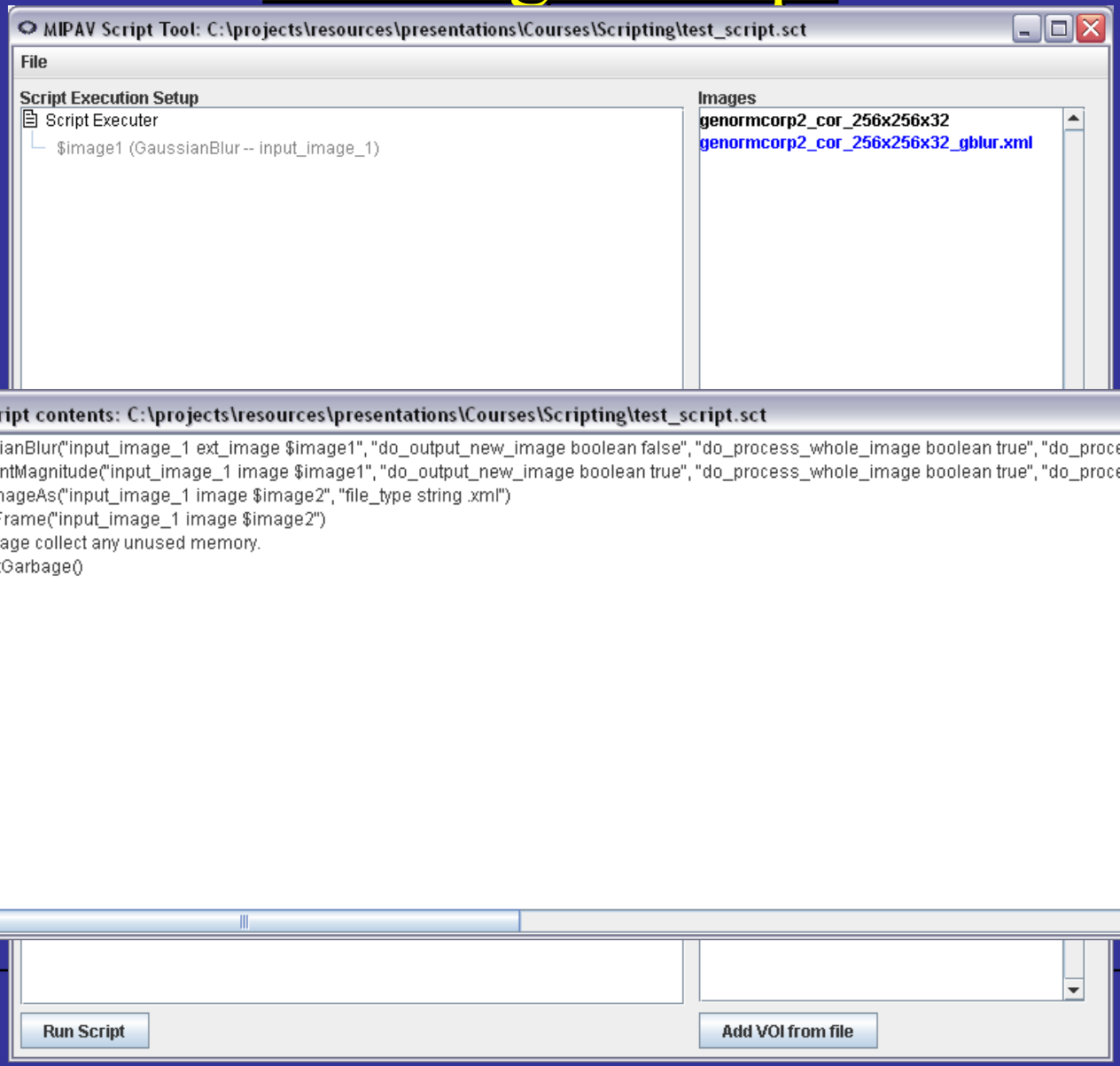
- GaussianBlur(“input_image_1 ext_image $image1”,
  “do_output_new_image boolean true”,
  “do_process_whole_image boolean true”, …)

Script Lab

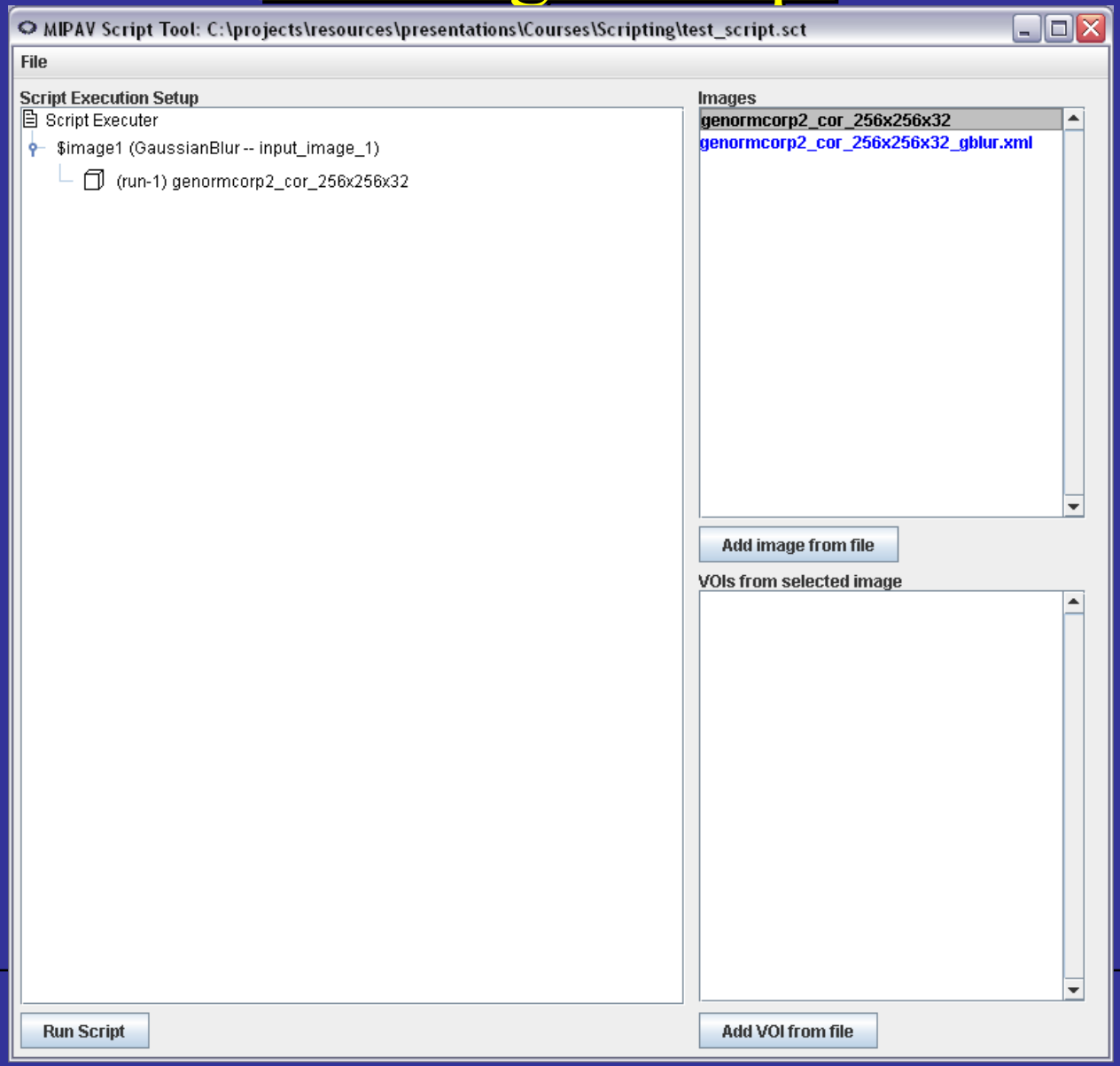
1. Open an image
2. If an algorithm is to operate using a VOI, the VOI must be loaded from disk using VOI > Open VOI > Open VOI
3. Start script recording
4. Run the algorithms in the order specified
5. Save the script when done



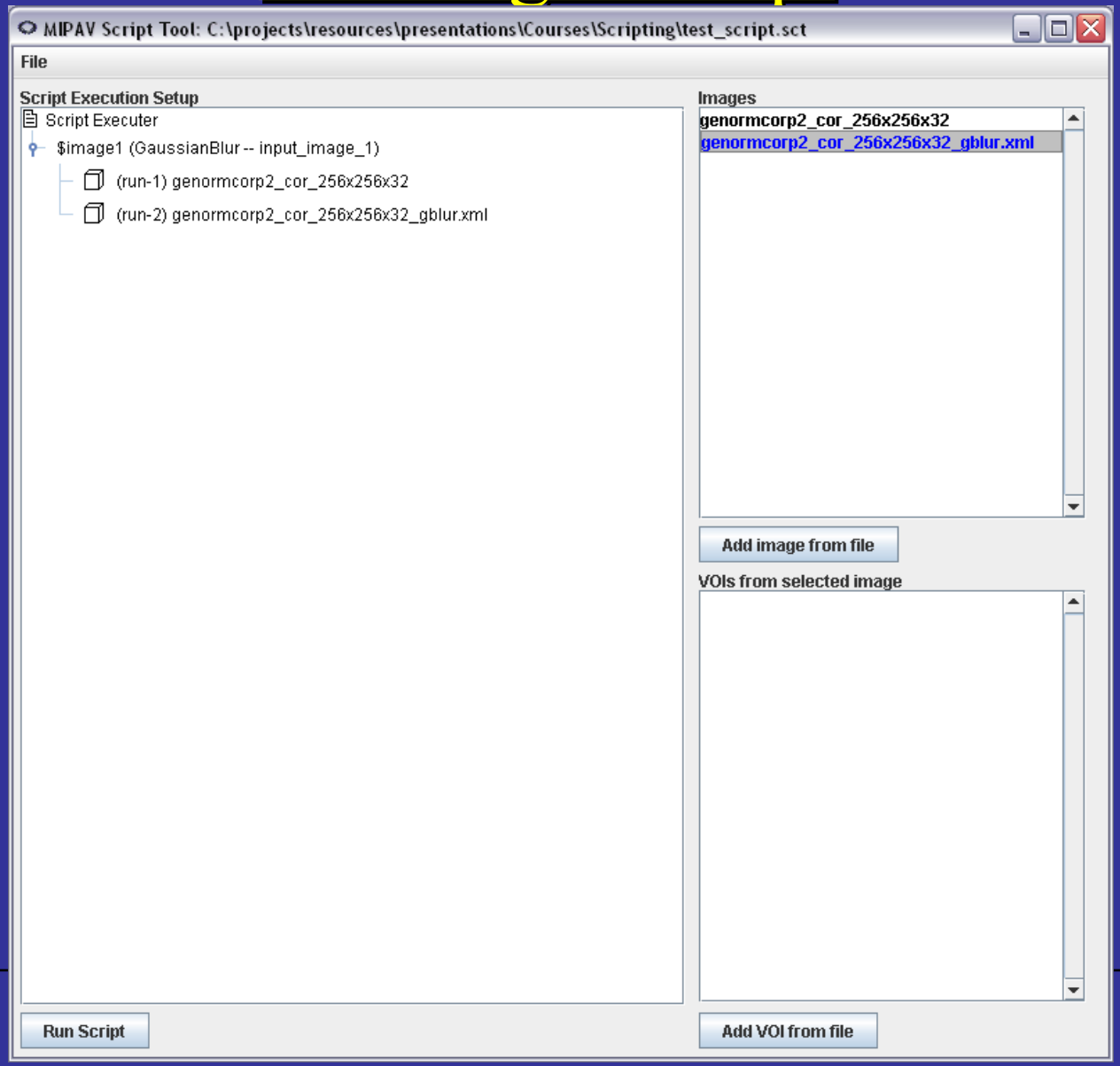
Running a Script



Running a Script



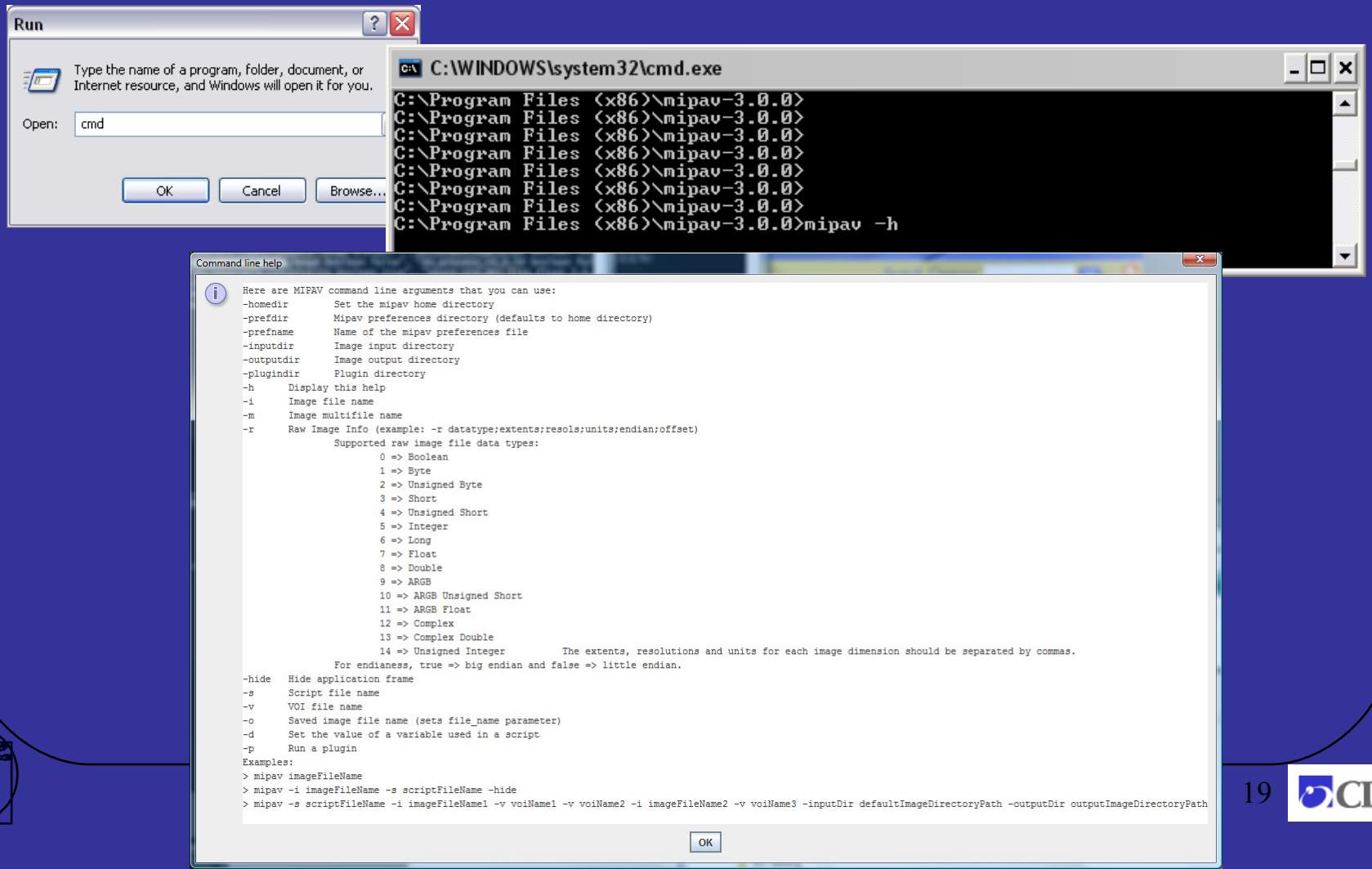
Running a Script



Scripting from the Command Line

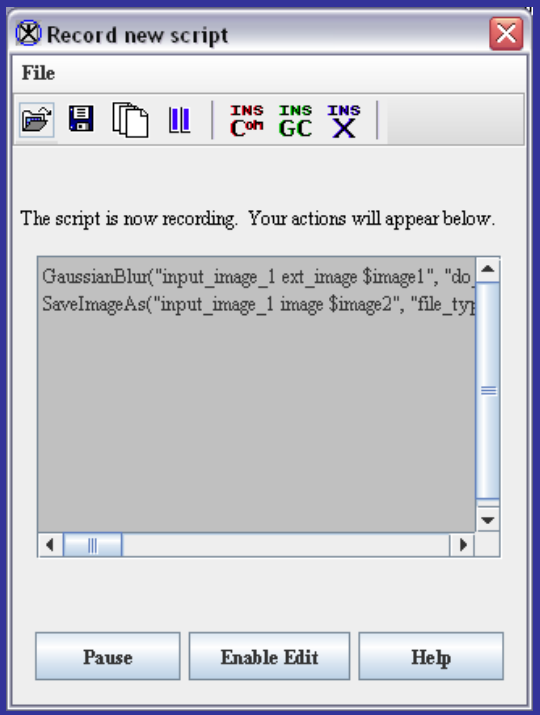
- MIPAV can be called from scripts (Perl, batch, shell scripts, etc.)

Calling a Script from the DOS command line



Writing a Script for Batch Processing

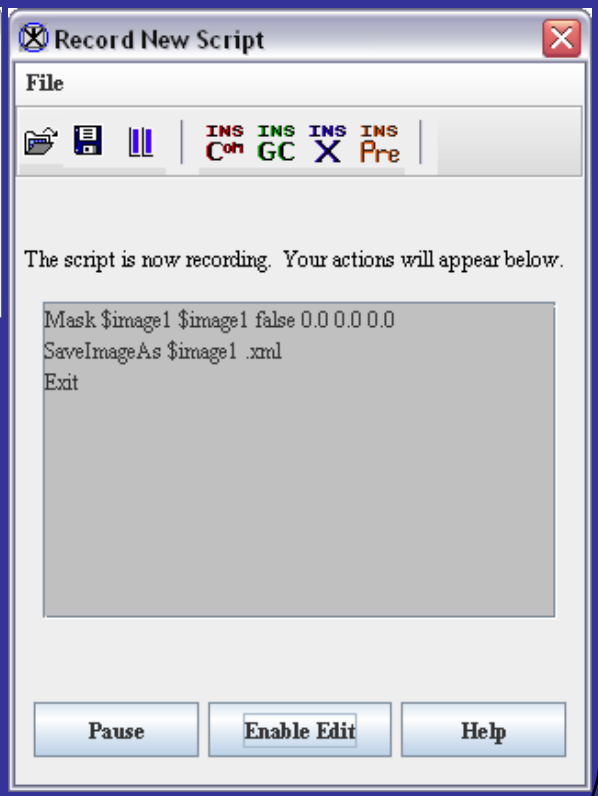
- Do not record loading VOIs into the script – we load these from the command line with the -v option
- Don't forget to "**Save As…**" before finishing
  - -o options saves the result image using the user provide string.
    - E.g. "-o outputFileName"
- Optionally, add an **Exit()** command to the end of the script
  - An **Exit()** will be automatically added to the end of a script executed with the -hide command



Writing a Script for Batch Processing – Lab

Classroom Lab

1. **Open an image**
2. **Open a VOI**
3. **Start script recording**
4. **Mask the image**
5. **Save result image as…**
6. **Save as new script when done** – call it *example.sct*



Batch Processing with MIPAV

- From the command line:
  - We can call MIPAV to run the script we have recorded
  - We can call a batch script or shell script to run a MIPAV active script
  - Windows:
    - Batch scripts (.bat files)
    - Cygwin (http://www.cygwin.com/)
    - ActivePerl
  - Linux/UNIX/Mac OS X
    - Shell scripting (bash, tcsh, zsh, etc.)
    - Perl
    - Python

Lab for Batch Processing with MIPAV

1. Open the command shell
2. At the command prompt:

   mipav –s example.sct –i image.xml –v voi.voi

Note: Runs the script just once on the given image

Batch Processing with MIPAV

Multiple images can also be processed through the use of scripting:

```
@echo off

rem usage: loop_xml.bat image_directory script voi_file
echo processes xml files in a dir with a mipav script, using a voi

for %%f in (%1\*.xml) do "C:\Program Files (x86)\mipav\mipav" -i %%f -s %2 -v %3
```

Similar scripts are even easier to write in shell scripts and Perl programs.

Command Line Notes

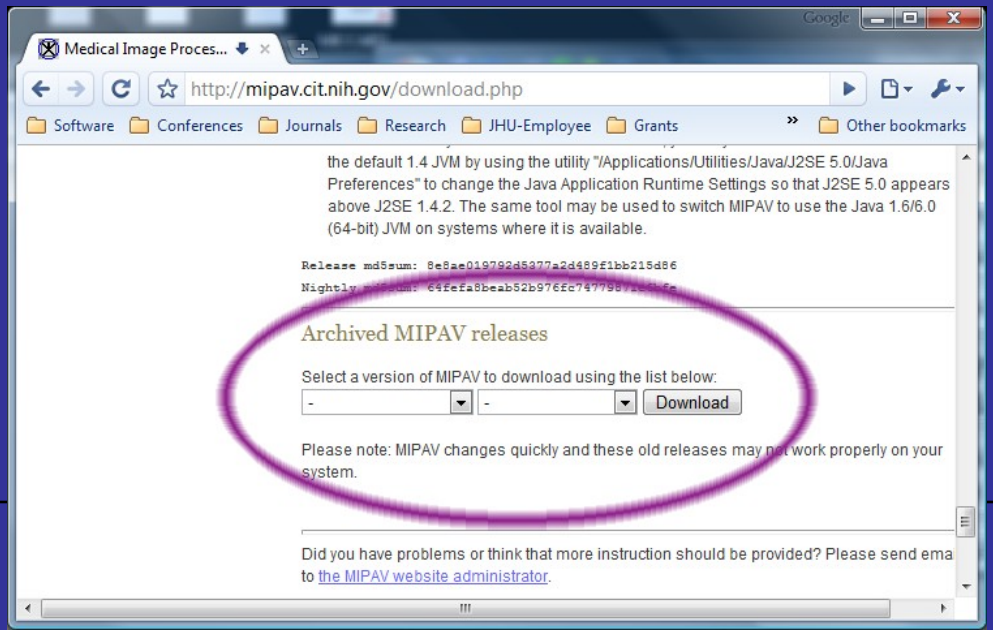
- Use the –hide option to prevent the MIPAV GUI from showing up.
- The –m option should be used to open multi-file images (such as DICOM volumes).
- Use the –help option to learn about other command line switches

Java Image Science Toolkit

- Originally developed at Johns Hopkins, now also supported by Vanderbilt (LGPL license)
- Created as a set of MIPAV extensions & plugins
- Main features
  - Automated GUI generation for application plugins
  - Graphical image processing workflow layout tools
  - Command line interfaces
- http://www.nitrc.org/projects/jist

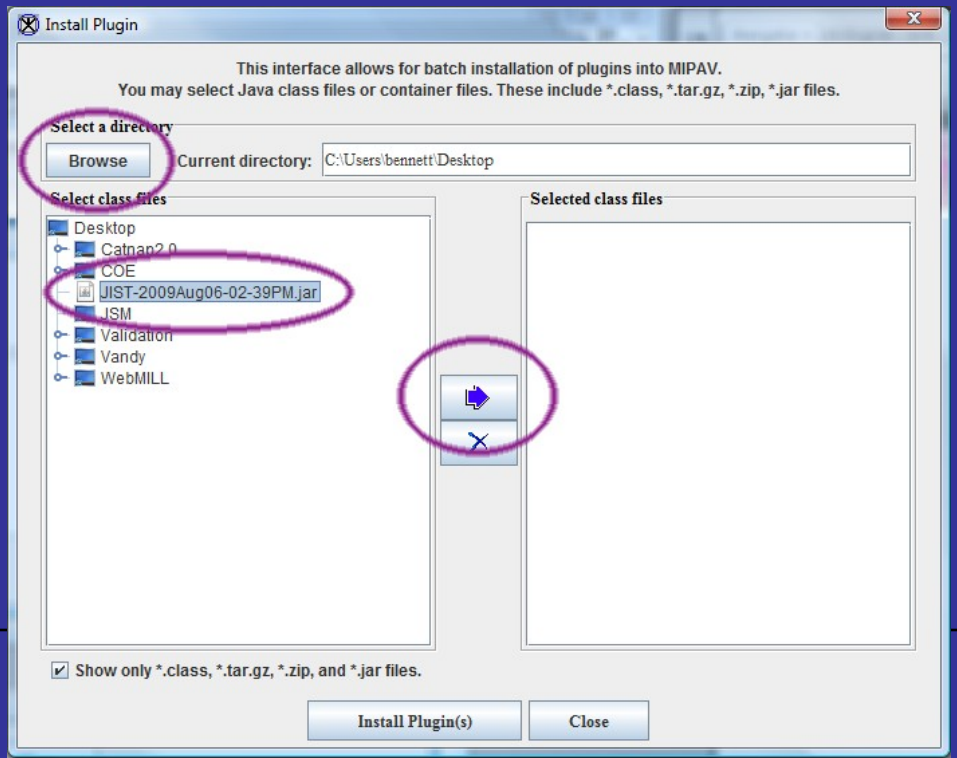
Installing JIST

- Install MIPAV
  - Download & install the MIPAV version compatible with the release of JIST you will be using
  - Latest release of JIST works with MIPAV 5.3.1



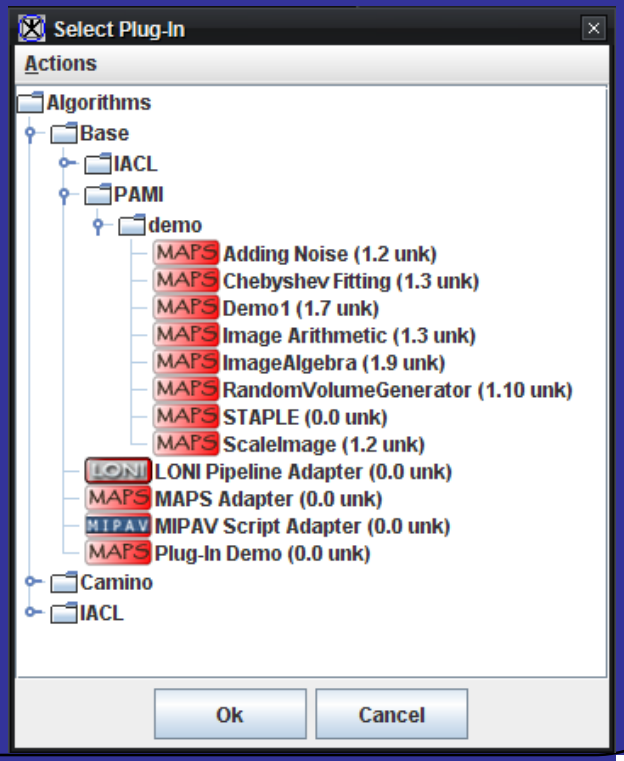
Installing JIST (cont.)

- Download the JIST jar file
- Use MIPAV's Plugins -> Install plugin dialog



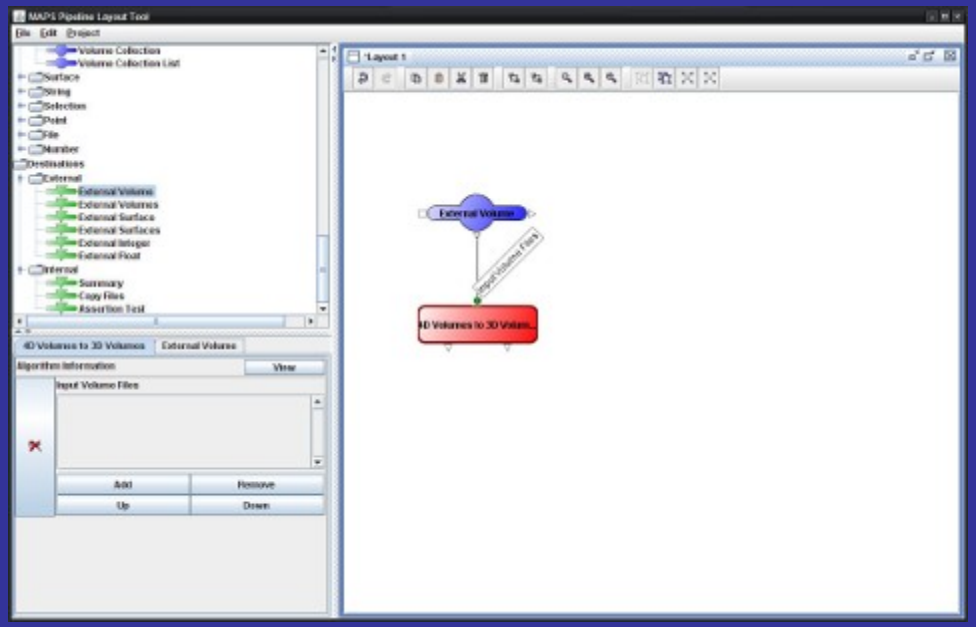
JIST Plugin Selector

- Useful for running or debugging a single JIST module



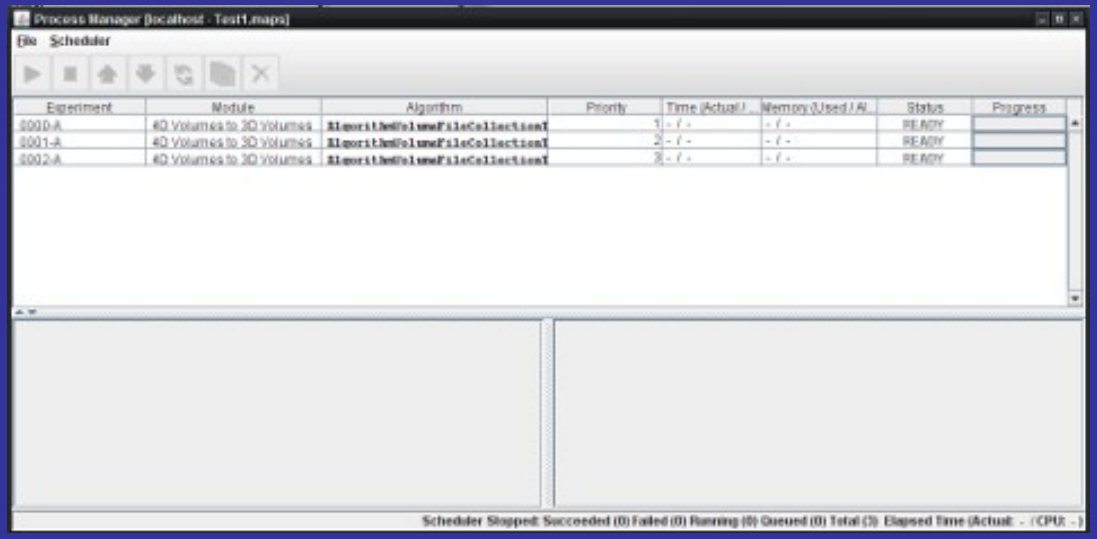
JIST Layout Tool

- Connects JIST modules together to create image processing workflows
- GUI interface



JIST Process Manager

- Run from the Layout Tool or standalone
- Executes layouts and displays status



JIST Lab

- Get JIST from http://www.nitrc.org/projects/jist
- Install JIST
- Create a simple workflow
  - Choose a MIPAV algorithm (e.g., Median filter)
  - Select the input image for the module
  - Output the result image to an External file

MIPAV

Medical Image Processing, Analysis, & Visualization

Thank you!