SliceOmatic 5.0

Table of Contents

Introduction 3
- History 5
- Interface tools 6
- File extensions 14

Installation 16
- System Requirements 19
- The Licenses 20
- The sliceO_ini.scp file 23
- The startup script file 25
- The sliceO_5_crash.scp file 26

Program Overview 27
- Modularity 28
- The Files 30
- The sliceOmatic screen 33
- Current and Selected 35
- Differences with Version 4.3 36

Main Menu 38
- The File Menu 39
- Undo / Redo 44
- The Tools 46
- The Modes 48

The Alberta Protocolo 50
- The simplified menus 51

The Display Area 53
- 2D and 3D windows 54
- Frame Selection 57

The Databases 58
- The 1D Database and tree 59
- The 2D Database and tree 60
- The 3D database and tree 61

The 2D Classes 62
- Study 64
- Series 65
- DICOM Tree 66
- Path 68
- Suffix 70
- MPR Oblique 72
- MPR Ortho 74
- ROI 76
- Multiplex 78
- Convolution Filter 81

The 3D Classes 85
- Group 86
- Matrix 87

The 2D Modes 89
- DB Class Management 2D 90
- DB File Management 2D 94
- 2D Measurements 99

The TAG Modes ... 102
Edit ... 104
Geometrical Masks ... 106
Morpho ... 110
Region Growing ... 114
Snakes ... 118
Thresholding ... 123
The 3D Modes ... 126
Display 3D ... 128
Contour Lines ... 129
Shell TomoVision ... 131
Shell Lorensen ... 134
The 2&3D Tools ... 137
Memory Manager ... 138
Point ... 140
Selection ... 143
Snapshot ... 144
Units ... 146
Windows ... 147
The 2D Tools ... 148
Blow-Up ... 149
Color Map ... 150
Color Scheme ... 151
Frame Selection ... 154
Gamma ... 156
Overlay ... 157
Pixel Grid ... 159
Pixel Info ... 161
The TAG Tools ... 162
Tag Color ... 163
Tag Delete ... 164
Tag Lock ... 165
Tag Surface/Volume ... 166
The 3D Tools ... 175
Background ... 176
Lights ... 177
Measures ... 179
Overlay ... 180
Shading ... 181
3D Slices ... 184
Surface Smoothing ... 186
Transform ... 188
Additional Modules ... 190
Class: RR2D ... 191
Class: RR3D ... 195
Tool: 3D Cleanup ... 198
Tool: 3D Displacement Burn-In ... 200
Tool: 3D Geometry Selection ... 201
Tool: 3D Relaxation ... 203
Mode TAG: Histogram Segmnetation ... 206
Mode TAG: TAG Interpolation ... 209

Mode 1D: DB Class management 1D 211
Mode 1D: DB File Management 1D 212
Mode 3D: DB Class management 3D 213
Mode 3D: DB File management 3D 214
Configuration Menu 215
Basic 217
File 219
TAG 221
3D 222
Drivers 224
The Browsers 226
The DICOM Browser 227
The Medi Browser 232
The Script Browser 235
Appendices
A: Command Line Syntax 236
B: Variables and Commands 241
B: The System DLLs 243
B: The Class DLLs 254
B: The Tool DLLs 255
B: The Mode DLLs 266
C: Keyboard Shortcuts 273
D: TAG File Format 276
E: The MOVIE.BYU file format 278

Presentation

sliceOmatic
User's Manual

Version 5.0
Rev: 19 November 2017

TomoVision
3280 Ch. Milletta, Magog, Qc
Canada, J1X 0R4
division of VIRTUAL MAGIC INC.

DISCLAIMER:

Good data processing procedure dictates that any program be thoroughly tested with non-critical data before relying on it. The user must assume the entire risk of using the program.

Virtual Magic Inc. makes no representation or warranty with respect to the adequacy of sliceOmatic for any particular purpose or with respect to its adequacy to produce any particular results. In no events shall Virtual Magic Inc. or its employees be liable for special, direct, indirect or consequential damages, losses, costs, changes, demands, or claims for lost profits, fees, or expenses of any nature.

For questions regarding sliceOmatic, call (819) 843-1999 or FAX (819) 843-1999 during Eastern standard time business hours, or e-mail (support@tomovision.com)

Information in this document is subject to change without notice and does not represent a commitment on the part of TomoVision, Virtual Magic Inc. It is against the law to copy this software in any fashion or any medium.

Introduction:
History

SliceOmatic is a medical imaging software designed to assist in the computation of anatomical volumes from CT and MRI scans.

SliceOmatic was developed as an in-house research tool. It was considered such a powerful tool that its fame grew by word of mouth, and soon it was used extensively by researchers around the world. Those who had the opportunity to use it were unanimous: sliceOmatic is the best tool for both Body Composition research and 3D modeling of anatomical structures. After many years of continual improvement, it is now available to the public.

The first version of sliceOmatic was developed for SGI's UNIX computers. But the advent of Pentium processors and powerful OpenGL graphic cards made it possible to port the application on PCs. Starting with version 4.1, sliceOmatic became a Windows application.

Introduction:
Interface tools

SliceOmatic uses its own set of graphical interface tools. Here is a brief description of the different tools you will encounter.

The Basic tools

The basic tools, defined in the TomoVision_Tools.dll library are:

Buttons

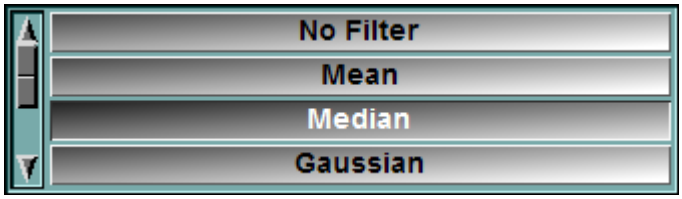
Buttons can be activated by clicking the left mouse button. A button will change color when the mouse pointer is over it. Once activated, a button will look depressed and its text will be white.

If a button appears as only a red outline, it cannot be activated.

Some buttons have yellow indicator light to indicate their state. The state can be toggled ON or OFF by clicking the button.

Button Lists

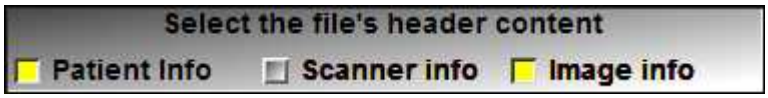
Button lists are used to group many buttons together. If there are more buttons than can be displayed at one time, a slider will appear on the left side of the list. This slider can be used to scroll the list.



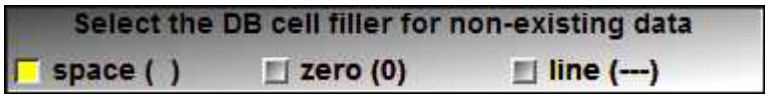
You can also use the mouse wheel when the cursor is over the tool to scroll through the list.

Check Boxs

This tool are used to check item in a list. There's 2 variation of this tool, in the first one you can check as many items as you which in the list. Each selected item will have a yellow indicator.

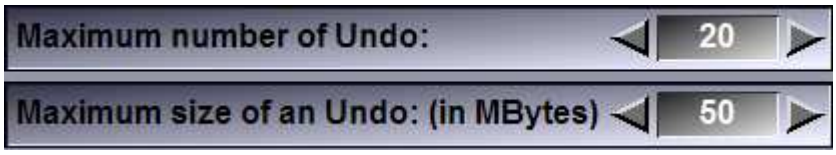


In the second variation, you can only select one item at a time. Selecting an item will automatically deselect the other ones. the selected item will have a yellow indicator.



Selection

This tool enables you to chose a value from a list of possible values. You can use the arrow keys on both side of the current value to scroll through the possible values.



You can also use the mouse wheel when the cursor is over the tool to scroll through the possible values.

Sliders

Parameters controlled by a slider can be modified either by clicking the left mouse button when the mouse pointer is over the desired portion of the slider, or by clicking and dragging the cursor of the slider with the mouse pointer. A slider will change color when the mouse pointer is over it.

Some sliders will also display their current values numerically.

When the mouse pointer is over the slider, you can also increase or decrease the slider's value with the keyboard.

Keys	*Function*
Page Down	+1 to the current value
Home	-1 to the current value
End	+10 to the current value
Page Up	-10 to the current value
Insert	+100 to the current value
Delete	-100 to the current value

In many instances, the width of the cursor is proportional to the range of values represented by the slider. Thus, for the Slice Selection slider, if the total number of slices is four, the cursor will be one fourth the length of the slider.

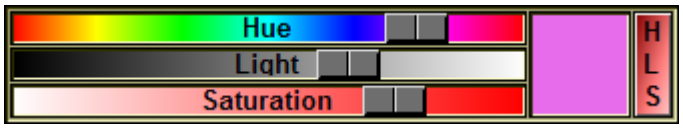
You can also use the mouse wheel when the cursor is over the tool to change the slider's value.


In some cases, the slider can be locked either from the command line, or by pressing the "Scroll Lock" key while the cursor is over the tool. If the slider is locked, its cursor will display a padlock.

Color

The Color editor tools enable you to select a color either by changing the "RGB" (Red, Grn, Blue) values with 3 sliders. A box on the right hand side show the resulting color.

Alternatively, by clicking the right hand side button, you can place the tool in "HLS" (Hue, Light Saturation) mode and specify the color with these 3 values.

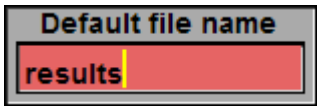


Hue

The Hue tool let you specify a color by changing the its Hue parameter. The box on the right hand side show the resulting color.

Text Input

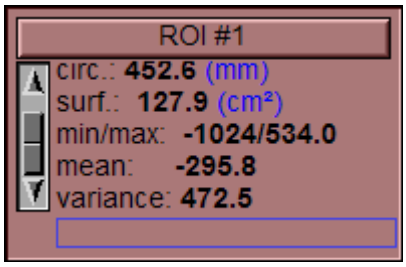
These windows are used to prompt the user for textual information.



You must place the mouse pointer over the window and type the information on the keyboard. Pressing the "Enter" key will signal the window that the text has been entered. You can use the arrow keys to move inside the text and the backspace and delete keys to edit the text. The vertical arrows are used to recall previously entered text strings.

Text Windows

These windows are used to display textual information. If a slider appears to the left or the bottom of the window, it can be used to scroll through the text.



Command Line

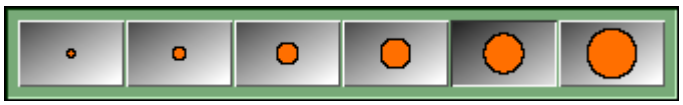
At the bottom of the main text window, there is a text input line where you can enter command lines for the program.

SliecOmatic's Tools

More complex interface tools are also used in some of the sliceOmatic modules. These are defined in the sliceO_Structure.dll library. They are:

Brush Selection

This tool is used to select a brush size when in a mode that use a brush to interact with the images.



You have the choice between 6 brushes, the smallest is a single pixel, the second is 3x3 pixels... up to a 20 pixel wide circle. The brushed in the tool will be of the color of the currently selected TAG value.

You can use the F5 to F10 keys as keyboard shortcuts to select the brushes:

Keys	*Function*
F5	Select the smallest brush (1x1)
...	...
F10	Select the biggest brush (20x20)

You can also control the brushes through the command interface, see ? for more information

TAG Selection

This tool is used to select the current TAG value. Each of the available TAG values has a button.

If a TAG value is locked (see TAG Lock Tool), its button will be unavailable (ex: TAG 2 in the demo image).

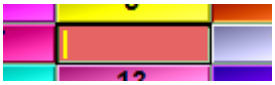
The currently selected TAG value will have its button depressed (ex: TAG 6 in the demo image)

You can use the 0 to 9 keys as keyboard shortcuts to select the first 10 TAGs:

Keys	*Function*
0	Select the TAG 0 ("none")
...	...
9	Select the TAG-9
.	Select the TAG under the cursor if the cursor is in the image area

You can assign labels to the TAGs (ex: TAG 8 = "Fat" in the demo image). To assign a label, you can either do it with a command in the command line, or by clicking with the right mouse button on the TAG's button.

Right click on the button

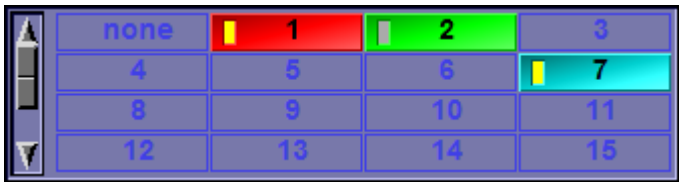


the button change to a text input tool

type in the label and press "Enter"

The TAG now has a new label.

In the 3D modes, the TAG Selection window is used to enable TAGs for 3D operations. A yellow indicator mark the selected TAGs. TAGs that can not be selected will be unavailable.



You can also control the TAGs through the command interface, see ? for more information.

Propagate

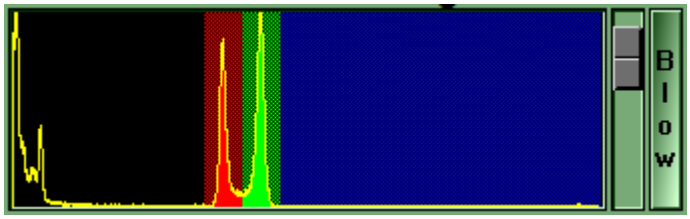
In some modes, you can use the values from one frame as a starting point for the values of another frame. The propagate tool let you select the "source" frame and propagate the values from this frame to a "target" frame (either "Up" or "Down").

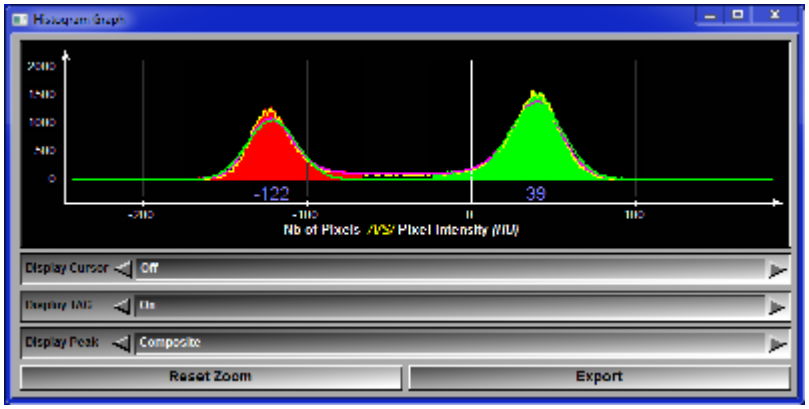
You can only propagate among frames that the children of the same class. So for example, if the Class tree has 3 levels: "Root", "Study" and "Frames", you will only be able to propagate among the frames of the same study.

The Propagate tool offer you to propagate "Down" (next child) or "Up" (previous child) from the "source" frame selected in the middle portion of the interface. You can change the "source" frame either from the image area (selecting the current frame), or with the arrow buttons. Changing the frame selected in the middle section does not propagate anything, only the "Down" and "Up" buttons do.

Histogram

Some tool or modes display an histogram of the image. This is a graph showing how many pixels are present for each of the possible pixel's values. The number of pixels is the vertical axis and the range of pixel values is the horizontal axis.





This is useful to fix threshold values, since the peaks in the graph represent the different tissues in the image. A slider on the right hand side can be used to change the vertical scaling of the histogram. When you move the cursor over the image in the display area, a vertical white line show the value of the pixel under the cursor. In modes where you adjust threshold values, the pixel values covered by the threshold will be highlighted with the color associated with the threshold.

When the cursor is over the tool, the mouse wheel can be used to change the vertical scaling. Clicking on the "Blow" button will open a new window displaying the histogram. From that window, you can zoom on a section of the histogram and export the histogram data. If the module producing the histogram has computed it, you can also display the Gaussian peaks that best fit the histogram.

Introduction:
File extensions

In Windows, there is practically no restriction to the length of a file name or its extension. However, since it is sometimes desirable to transfer files to DOS, we recommend that you limit your file names to 8 characters and your file extensions to 3 characters. File extensions are not necessary but are useful for identifying files.

The file extensions for the files created by **sliceOmatic** are:

Extension	*Identification*
.**scp**	A script file. Script files are used to group a sequence of GLI files and their "z" values in one read operation. They are also used to specify a number of operating parameters for the program.
.tag	A TAG file. These files are the tagged images produced by the program. The program will produce one TAG file for each GLI file.
Body composition result files:	
.txt	A surface area and volume results file in ASCII format. These files contain the results of the surface and volume computations done by the program.
.csv	A surface results file in database compatible format. These files contain the surface area results in a format that can easily be imported into a database program.
3D polygon files:	
.mov	Movie.BYU format. This old polygon format, developed by Brigham Young University for their MOVIE.BYU program in the 80's, is very simple which make it ideal to store simple polygonal objects such as those created in sliceOmatic. The format is described in more detail in appendix E. SliceOmatic can create ".dxf" files, but it can not read them.
.dxf	Autocad file format. The geometries will be saved using the POLYLINE primitives. A definition of the DXF file format can be found in Wikipedia. SliceOmatic can create ".dxf" files, but it can not read them.
.stl	Stereolithography file format (binary version and ASCII versions). A definition of the STL file format can be found in Wikipedia.

.obj	Wavefront OBJ file format. A definition of the OBJ file format can be found in Wikipedia. SliceOmatic can create ".obj" files, but it can not read them.
.wrl	A VRML polygon file. A definition of the VRML file format can be found in Wikipedia. SliceOmatic can create ".wrl" files, but it can not read them.

Installing the software

Installing TomoVision's sliceOmatic from the CD-ROM

What to have on hand:
- TomoVision's sliceOmatic CD
- License Dongle

1. Log in as an administrator.
2. Insert TomoVision's sliceOmatic CD into the computer's CD-ROM drive. If your Autorun is not enabled, click on the icon representing your CD-ROM drive under "My Computer". This will bring up a listing of the files on the sliceOmatic disk. Find SliceO_5.exe and double click to launch the Install application.
3. Follow the on-screen steps.

Installing TomoVision's sliceOmatic from the Web

What to have on hand:
- License Dongle (If you have one. If not, the program will work in "Demo" mode).

Installing the Dongle
- Plug to dongle in the USB port. When the dongle is firmly in place, its indicator light will turn on. For more information on what is a dongle and why it is needed by the software, please refer to "The License" section.

Installing TomoVision's sliceOmatic
1. Log in as an administrator.
2. With your WEB browser, go to the sliceOmatic download page: (http://www.tomovision.com/download/download_sliceo.htm) and click on the download button.
3. From the pop-up menu select "Run" and follow the on-screen steps.

Installing the dongle's device drivers
1. The license dongle's drivers are automatically installed by the same script that installed the sliceOmatic program.

Uninstalling the Software

From the Windows Start Button, select Control Panel in the Settings menu. In the Control Panel, select Add/Remove Programs, left-click on SliceOmatic, and click on the Add/Remove button. The software and all its components are then removed from your hard drive.

Moving sliceOmatic to a new computer

If you wish to transfer the software from one computer to another, you can do it. The licenses are contained in the dongle and are not tied to the computer. Just install the software and the dongle's drivers on the new computer and transfer the dongle to that

computer.

Updating the software

If you have a valid update license, you can download and install the latest version of sliceOmatic from the Web.

To upgrade the software, simply install the latest version over your old version.

Starting the Program

You can find all the installed sliceOmatic programs on the Windows **Start** menu under **Program / Tomovision**. You can also start the program by selecting image files with the mouse, drag them to the program icon and drop them there, or you can start the program without arguments, simply by clicking on its icon. By default the program resides in C:\Program Files\TomoVision.

When sliceOmatic starts, it first parses its command line arguments for special instructions, then it look in its ".ini" files for its default parameters.

SliceOmatic can accept the following command line arguments:

sliceOmatic [-d *path*] [-p *name*] [-t level] [-v] [-z] [-x *pos dim*] [-y *pos dim*]

General operation arguments (shared by all TomoVision's programs)

-t *level*	Trace level (0 <= level <= 5)
-v	Display the program's version and creation date
-x *pos dim*	Horizontal position and dimension of the program's window (in pixels)
-y *pos dim*	Vertical position and dimension of the program's window (in pixels)

SliceOmatic specific arguments

-d path	Starting directory. All File Open and Save dialogs will be rooted in this directory. (By default: path = the current directory)
-p name	Pipe name. The program communicates with other programs through this file. If you wish to open more than 1 instance of the program, you have to specify a different pipe for each instance. (By default: C:\temp\sliceO_5.dat).

Note:

If you want to start more than 1 instance of sliceOmatic at the same time, you will need to assign a different pipe to each instance of sliceOmatic.

Program Crash

In the event of a program crash there will be an error log file called *TomoVision.err* in the Scratch Directory ("C:\Temp" by default). SliceOmatic also saves a list of all the images opened during your last session in "sliceO_5.dat" in the Scratch Directory. Upon startup,

sliceOmatic will ask you if you want to automatically attempt to reload all these files.

Installation:
System Requirements

The CPU & Memory
SliceOmatic is mostly a graphic interface enabling you to control the segmentation of your images. From time to time it does a few computer intensive tasks, but most of the time it just controls the interface. So you do not need the most powerful computer around to run the program.

However, sliceOmatic is a memory hog, so you need a lot of RAM memory. If you plan on segmenting 3D datasets, you will need at least 2Gb of RAM. If you are using a 32 bit operating system, then more than 4GB is overkill since the operating system cannot access it. But if you are using a 64 bit operating system, then the more memory the merrier.

The Graphic Card
SliceOmatic needs a good OpenGL driver and unfortunately most graphic card manufacturers do not have these. We recommend a graphic card with the Nvidia chipset since, to our knowledge, they have the best OpenGL drivers.

SliceOmatic does not need a lot of graphic memory, and unless you do 3D reconstructions, you also do not need the most powerful cards.

The Operating System
SliceOmatic runs on Windows, so you need to use a Microsoft Windows operating system. Any of the Windows (starting with 95) should do. If you are using a 64 bit version (starting with Vista) make sure you install the 64 bit version of the program to take full advantage of it.

Installation:
The Licenses

TomoVision products use 2 layers of protections:

- a hardware layer, provided by the HASP (Hardware Against Software Piracy) dongles from Aladdin
- a software layer consisting of encrypted license activation codes stored in the dongle's memory.

The HASP Dongle

A dongle is a hardware device connected to the USB port. The dongle used by TomoVision has its own timer and memory. All the information needed to license your products is included in the dongle, so it's easy to move your application to a new computer.

USB "HASP" Dongle

If the dongle is not present on your computer, the software will run in "demo" mode and some functions will not be available.

The 30 days Licenses

Each module of each product can be issued a 30 days license. These licenses will unlock the desired module for a period of 30 days starting from the moment the licenses are written in the dongle at our office. The 30 days are controlled by an internal clock in the dongle and are independent of your computer's clock. While the 30 days licenses are active, you can download the latest version of the software from our Web site and it will be covered by the licenses.

The dongles are always shipped with 30 days licenses. The reasons for this are twofold:

- As soon as we get a P.O., we ship the dongle with the 30 days licenses. This enables you to start working with the product immediately while the payment is handled by the accounting department of your institution.
- If the dongle gets lost in transit, we can replace it without worrying about multiple licenses being issued, since the licenses in the lost dongle will expire in 30 days anyway.

The Permanent Licenses

Each module of each product can be issued a permanent license. These licenses will unlock the desired modules. As the name implies, the permanent licenses do not expire. The permanent licenses are written to the dongle with the help of the

TomoVision_License.

The Update License

The upgrade license is issued by product. Each license contains 2 informations: the modules that are covered by the license and the last allowed update date. Until that date, you can download the latest version of the software from our Web pages and it will work with your license dongle. And since your permanent licenses are permanent, it will keep on working for... well, forever!

But, after that date you can no longer download a new version of the software. Any version of the software that is younger than the update expiration date will not be recognized by the licenses. In this case you can either:

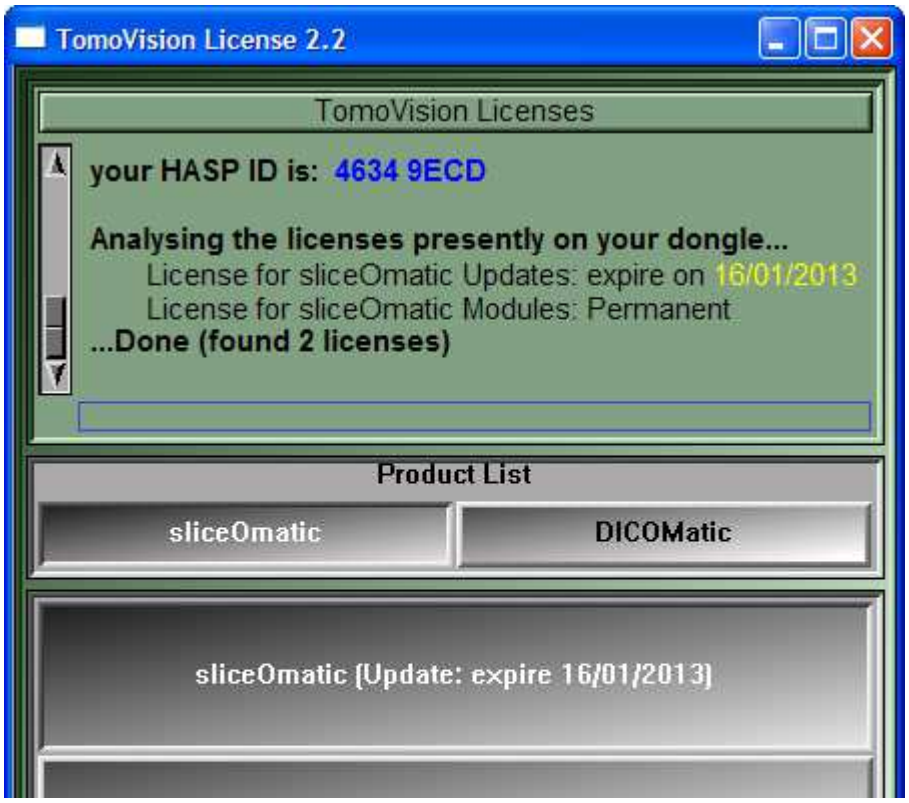
- Renew your update contract. Update contracts are available from TomoVision for a yearly fee.
- Re-install a previous version of the product. Previous versions of the software are available on the download page of sliceOmatic.

If you wish to keep on upgrading your software, you can purchase an additional year of support & upgrade from our Web site at **www.TomoVision.com**.

Please note that update contracts are uninterrupted. This means that the starting date of your next contract is the finishing date of this one, regardless of the date of purchase. If you fail to renew your contract for more than 6 months, it will be terminated. Following this, if you wish to get the latest version of the product, you will have to purchase a new license.

The TomoVision_License Program

The TomoVision_License program is used to read and write license activation codes on the HASP dongle and to generate a registration form if needed. Each dongle has an unique ID number. This unique ID is used by TomoVision when the license codes are generated, so the licenses are only valid for a specific dongle.



The TomoVision_License program is installed automatically along

with your TomoVision application. But you can also download it directly from the "Tools and drivers for the license dongle" page in the Download section of TomoVision's Web site.

Start the TomoVision_License program from the **start\program\TomoVision** menu, or by selecting the "Register" option in the "File" menu of sliceOmatic.

Activating the Licenses

The program "TomoVision_License" is used to transfer the license activation codes to the dongle.

Just drag & drop the activation codes file, sent to you by TomoVision, on the License program. All the activation codes present in the file will be entered. You should now see 2 buttons in the license list for the product sliceOmatic: one for the update license, and one for the temporary or the permanent license.

Generating the Registration Form

Unless asked for by TomoVision, you do not need to generate a registration form.

If TomoVision asks for it, you should:

- Make sure the HASP dongle is connected to your computer.
- Start the TomoVision_License program.
- Click on the "Generate Registration Form" button. This will create a registration file in the c:\temp directory. This file will be named: "TomoVision_xxxxxxxx_ddmmmyy_reg.dat", where xxxxxxxx is the dongle's ID and ddmmmyy is the current date in the form day/month/year.
- Fill out this form (you can use Windows "Notepad" or any other editor to edit the form).
- Email the form to "sales@TomoVision.com"

Installation:
The sliceO_ini.scp file

There are 2 copies of SliceO_ini.scp: the first is in the directory of the sliceOmatic program, the second in the user's private directory (usually on XP it is C:\Documents and Settings\user name\Application Data\TomoVision\, on Windows 7 it is C:\users\user name\AppData\Roaming\TomoVision).

At startup, the program reads both files, starting with the one with the sliceOmatic program. This enables you to have preferences defined for all users (from the copy in the program's directory) and preferences tailored for each user (from the copy in the user's directory).

The copy in the user's directory is generated and modified with the preference menu of the program. If you want to apply the same preferences to all users, overwrite the copy in the program's directory with the one generated by the sliceOconfig program.

The "SliceO_ini.scp" file contains a series of command statements using the script file syntax. A detailed description of these commands can be found in Appendix C of the User's manual. The commands found in the ".ini" file are:

Basic Module:	
`undo: number val`	(val= 0 to 20. Def=20)
`undo: size val`	(val= ? to 100. Def=50)
`overlay: 2D val`	(val is a bit field with the first 4 bits used. Def=00000002)
`interface: size val`	(val=small,medium or big. Def=big)
`main: floating val`	(val="on" or "off, Def="off")
`read: scratch path_name`	(Def="c:\temp")
`read: startup file_name`	(Def="")

File Module:	
`read: path path_name`	(path_name is the default path)
`regional: fraction val`	(val= "." or",". Def=".")
`regional: separator val`	(val= "tab"| or " ". Def="tab")
`regional: filler val`	(val= " ", "0" or"---". Def=" ")
`write: header patient val`	(val= on or off. Def=on)
`write: header scanner val`	(val= on or off. Def=on)
`write: header image val`	(val= on or off. Def=on)

TAG Module:	
`Tag: number val`	(val=16,32,64,128 or 256. Def=128)
`Tag: autosave val`	(val= time in min. Def=0)
`Tag: suffix val`	(val="off" or suffix. Def="off")
`Tag: path val`	(val="off" or path name. Def="off")

3D Module:	
`overlay: 3D val`	(val is a bit field with the first 4 bits used. Def=00000005)
`OpenGL: speed val`	(val=1 to 100, def=1)
`Transform: mode val`	(val= Track, Classic or Patient. Def=Track)

Drivers Module:

`OpenGL: context` *`val`*

(val is the current context. def=fixed by Windows)

Installation:
The startup script file

From the preference menu, you can activate this optional script file.

At startup, sliceOmatic will read the SliceO_ini.scp file. Depending on the preference setups, this file can contains a pointer to another file, the Startup script file. This is a script file that can contains additional commands to configure the program.

For example, you could use this file to rename a few of the tags you will use:

- With Notepad, create a file named "rename.scp"
- In this file place the lines:
  tag: 1 label "Fat"
  tag: 2 label "Muscle"
- In the preference menu, Set the "Startup Script File" tp "On" and browse the file tree until you find your "rename.scp" file.
- Apply & Save the preferences

The next time you start sliceOmatic, the "rename.scp" script will be executed and the first 2 tag buttons will now have "Fat" and "Muscle" as labels instead of "1" and "2".

Installation:
The sliceO_5_crash.scp file

When you open a file in sliceOmatic, the program create a "sliceO_5_crash.scp" file in the user's directory. This file is kept up to date and contain script commands to re-load the files that are currently opened in the program. When you exit the program, this file is deleted. If the program crash, the file is not deleted. The next time you start sliceOmatic, the program detect the file and ask you if you want to reload the same files.

Program Overview

This software is designed to help in the visualization and computation of anatomical volumes from tomographic scanners. To achieve this goal, this software helps the user to perform the following steps:

- Load and visualize the scanner images
- Reorder the images (usually according to their "depth" values)
- Segment the images
- Edit and tag the segmented images
- Compute the surface area and volume of the tagged pixels
- Visualize the segmented volumes
- Extract the surface and volume information of the segmented images

---

Program Overview: Modularity

SliceOmatic 5.0 is designed to be modular. Every class, tool and mode in the program is contained in an individual DLL file. Removing or adding DLL files in the sub-directories planned for this will remove or add the corresponding module from the program.

The DLLs

Depending on their function, the DLL libraries associated with the program will be stored in one of 9 folders:

- The same folder as the executable

  Here we will find libraries that are necessary to the program and all the other DLLs.

  These are:
  - **Lib_Util.dll**. This library contains the basic TomoVision functions such as string manipulation, memory allocation and error reporting functions.
  - **Lib_Tools.dll**. This library contains the TomoVision's tools (buttons, sliders...)
  - **Lib_DICOM.dll**. This library contains all the DICOM manipulation functions.
  - **Lib_Convert.dll**. This library contains all the format converters that enable TomoVision's product to understand the image formats from more than 40 different scanners.
  - **Module_Tools.dll**. This is the library containing the more advanced tools used by sliceOmatic (brush selection, TAG selection...)
  - **Module_Structure.dll**. This library contains all the functions that are used by the other DLLs of sliceOmatic.
  - **Module_Browser.dll**. This library is used to create the DICOM, Medi and script browsers.
  - **Module_Config.dll**. This library is used to manipulate the configuration menu.
  - **Module_Protected.dll**. This library is protected with the HASP key and will only work if you have a valid license. It contains the functions to save sliceOmatic's data.
  - **hasp_windows.dll**. This is a library provided by the HASP key manufacturer. SliceOmatic needs it to access the license key.

- The **System, Menu** and **Config** folders

  Here we place all the "system" DLLs. Those are DLLs that perform basic operations and that sliceOmatic could difficultly live without, such as reading files, the basic TAG operations...
- The **Class_1D, Class_2D** and **Class_3D** folders

  In these folder we have one DLL for each class that can be used in the database tree (except, Root and Frame that are contained in sliceO_Structure.dll).
- The **Tool** folder

  In this folder we have one DLL per tool used in sliceOmatic. Note that not all tools have a graphic interface, so there are more DLLs than what you see in the Tools menu.
- The **Mode_1D, Mode_2D Mode_3D** and **Mode_TAG** folders

  In these folder you have one DLL per Mode.
- The **User_Class_1D, User_Class_2D, User_Class_3D, User_Mode_1D,**

User_Mode_2D, User_Mode_3D, User_mode_TAG and **User_tool** folders
These folders are used to place additional classes/modes/tools.

User Modules

It is very easy to add new modules to the program. Just create a DLL for it (see the developer's manual) and place it in the appropriate sub-directory ("User_Class_*x*D", "User_Mode_*x*D", "User_Mode_TAG" or "User_Tools"). The next time sliceOmatic is started, the new module will be available in the different menus.

License:

User modules will only be loaded in the program if the **sliceOmatic Module** license is present. In demo mode they will not be available.

User modules can be protected by sliceOmatic's license system if so desired. If a developer does not want his module to be freely available, he can ask TomoVision for a license code. He will be provided with a code specific to his module and a license generating program to create user licenses for his module. The generated user licenses can then be read by the TomoVision_License program and incorporated in the license dongle.

Program Overview:
The Files

There are 4 different file types you will encounter in this manual: the GLI files, the TAG files, the GEOM files and the Script files.

The GLI Files

GLI stands for **Grey Level Image**. It is a generic term used to describe the input images you read in sliceOmatic. The pixel values of these images represent the physical properties of the scanned tissue (CT number, photon emission...) expressed in a numerical form by an 8, 16 or even 32 bit integer. They have a range of values that is usually in the thousands (potentially from -32,768 to 32,767 for a 16 bit image). These images will be displayed in shades of grey, hence the term Grey Level Image or GLI.

Note:

Since version 4.0, sliceOmatic can also read and manipulate color images, the term GLI (although not really appropriate) will still be used for these images.

This program can work with any type of images (CT, MRI, US, SPECT, PET...). But since there are a variety of file formats, some image formats may not be recognized by sliceOmatic.

Presently, sliceOmatic is able to read the following standard formats:

DICOM, **NEMA-1, NEMA-2**, **TIFF**, **JPEG**, **Interfile 3.0**.

Also, sliceOmatic is able to read a number of proprietary file formats. A detailed list is available at: http://www.tomovision.com/products/format_image.htm

The TAG Files

The program will help you create a new set of images called **TAG** images. There will be one TAG image for each GLI image. They contains pixels whose values can range from 0 to 255. These values are tags that are used to identify the anatomical structures of the corresponding pixel in the GLI images. (ex: all pixels whose tag is 1 are bone, 2 are fat, 3 are muscle ... Or, 1 are kidney, 2 are liver, 3 are bone...).

The program will display these TAG images as color images superimposed on the GLI images. Depending on the color scheme selected:

- only the TAG images will be displayed
- only the GLI images will be displayed
- the TAG images will be opaque and displayed over the GLI images
- the TAG images will be translucent and displayed over the GLI images.

These new TAG images can be saved in files. The created files will have the same name as the corresponding GLI file but with a ".tag" extension. Optionally, a suffix can be appended at the end of the file. This can be useful if you want multiple users to segment the same GLI images. Each user can have his own suffix and thus have different TAG

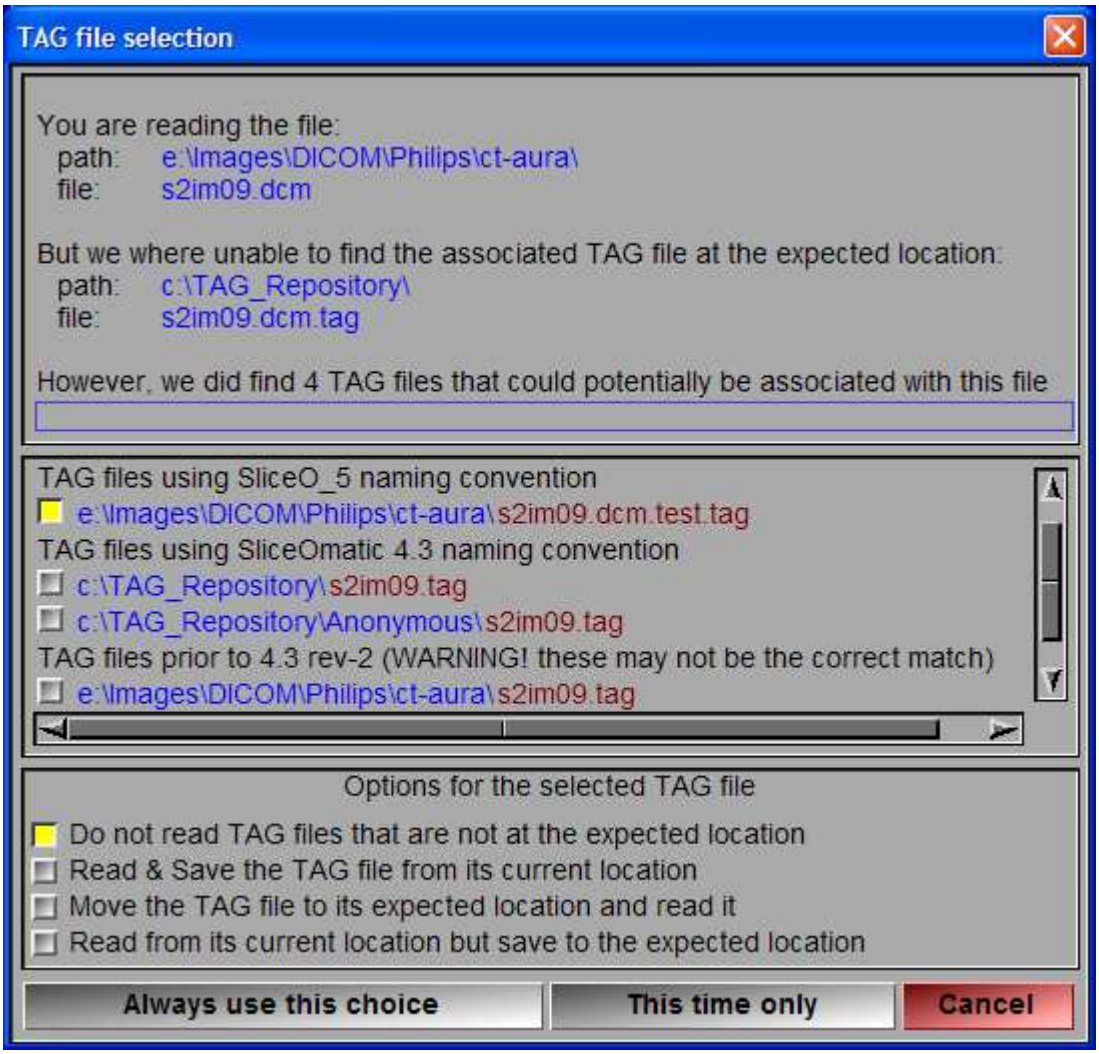
files while sharing the same GLI files.

The format of these files is given in **Appendix B: The TAG File Format**.

These files will be placed either with the GLI images or in a different directory (see the TAG section of **The Configuration Menu**).

When sliceOmatic loads a GLI file, it will automatically search for an associated TAG file and load it if present. Do not attempt to load a TAG file directly, this will only confuse sliceOmatic.

When loading the GLI file the program looks for the associated TAG file in the directory specified in the preferences (by default, the same dir as the GLI file) using the name syntax specified in the preferences. If the program does not find the associated TAG file, it will look in other directories using other syntax. If it does find a TAG file that matches the GLI file, it will pop a window with a warning message and ask for directions.



The Geom Files

The 3D module of sliceOmatic is used to create and manipulate 3D polygonal surfaces. These surfaces can be exported in a number of 3D formats: STL, MOVIE.BYU, DXF, OBJ and VRML. Files in STL or MOVIE.BYU format can also be imported in the

program.

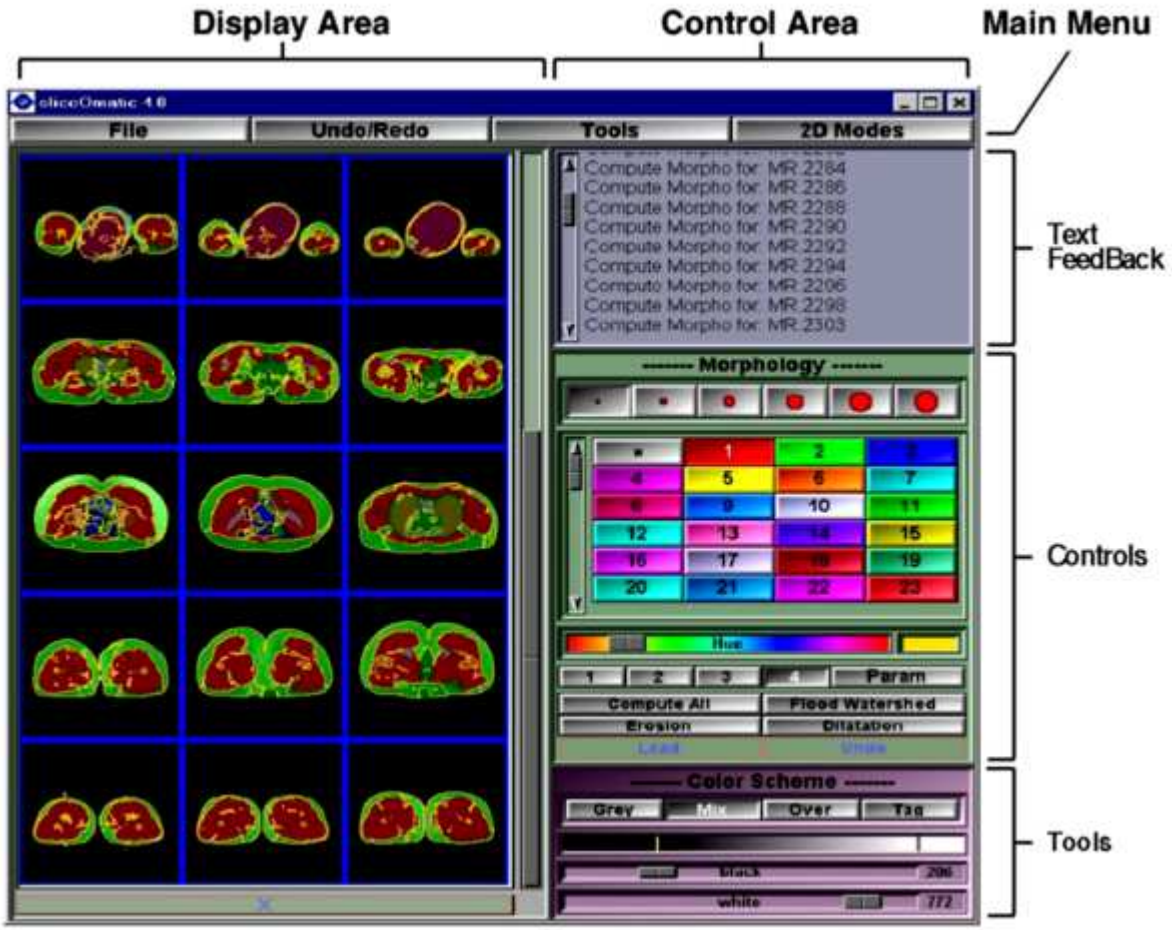
The Script Files

Reading, sorting and changing the orientation of all the slices of a data set can be a long task. SliceOmatic can help you automate this task by generating a script file. SliceOmatic can generate for you a script file that contains all the information necessary to reread all the files currently in sliceOmatic. Alternatively, you can write your own scripts and use them to read and manipulate your data sets. The syntax of these files is given in **Appendix A: The Command line Syntax**.

Program Overview:
The sliceOmatic screen

The screen is divided in 3 main areas:

- the **Main Menu**
- the **Display Area**
- the **Control Area**.



The Main Menu

The Main Menu has 4 buttons: "File" "Undo/Redo", "Tools" and "Modes". These will be discussed in more detail in the "The Main Menu" section.

The Display Area

The display area is where the actual image windows are shown, You can have up to 4 windows at the same time, each of these can display the frames in 2D or 3D and is associated with a Mode. The display area is explained in more detail further down.

The Control Area

The Control Area is itself divided in 3 areas:

- the **Text Feedback window** and the **Command Line**. This window provides you with textual feedback. If an error occurs while the program is doing some operations, it will be displayed here. You can also enter commands to the program by typing them when the cursor is over this window and pressing “Enter”.

- the **Mode Dependent Controls**. The controls in this area will change according to the mode selected in the Main Menu. Each window in the display area is associated with its own mode, changing the current window will change the current mode and (if needed) the active tools.

- the **Mode Dependent Tools**. The tools can be enabled or disabled from the Main Menu. The availability of some tools may also be affected by the currently selected mode (for example, 3D tools are only available if the current mode is a 3D mode).

Note:

The 3 windows of the Control Area (Text, Controls and Tools) can be made "floating windows" from the **configuration menu**. Floating windows can be placed anywhere. If you work with 2 screens, you can placed these windows in one screen and the display area window in another.

Program Overview:
Current and Selected

Here's a brief overview of some of the concepts used in sliceOmatic. They will be viewed in more detail further down.

The Current Window

If multiple windows are present in the display area, only one of them is the current window. The current window's border will be highlighted. Clicking in a window will make it the current window. If only one window is visible, then it is the current window.

The Current Mode

Selecting a Mode from the Main Menu "Modes" section makes it the current mode for the current window. Each window in the display area is associated with its own mode, changing the current window will change the current mode and (if needed) the active tools.

The Current Frame

If multiple frames are visible, only one of these is the current frame. The current frame will have a yellow outline. Clicking on a frame will make it the current frame. If multiple windows are visible, only the current window will have the current frame. If only one frame is visible in the current window (Mode ONE) then it is the current frame.

The Selected Frames

On top of the idea of the current frame, sliceOmatic uses the concept of the selected frames. SliceOmatic offers you a simple and intuitive way to select these frames, either with the mouse (Frame Selection in **The Display Area**) or from list (the Selection tool in **0D Tools**).

By default,

- in Mode One, only the visible frame is the current frame and it is the only selected frame,
- in Mode All, all the frames are selected, and the frame having the yellow border is the current frame.

Changing from one mode to another will reset the frame selection to these default values.

Program Overview:
Differences with Version 4.3

Even though internally sliceO-5 is a completely different beast than sliceOmatic 4.3, from the user point of view is does look a lot like it did before.

Here's an list of the main differences.

The TAG file names

In the previous sliceOmatics, the GLI file name extension was replaced by ".tag". This would create problems if all the files had the same names but with different extensions (ex: toto.1, toto.2...).
In sliceO-5 the extension ".tag" is added AFTER the GLI complete name (including its extension). So the TAG file for "toto.1" will now be "toto.1.tag"

The concept of the Tag repository has been eliminated. You can still save your tags in a designated directory, but the program will not create sub-directories automatically for you.

A new suffix can now be added to the tag file name. This suffix comes after the GLI file's name and before the ".tag" extension. So if you add the suffix "test" and save the TAG file for "toto.1" it will be named: "toto.1.test.tag". The suffix are used when you want to segment the same GLI data multiple times (for example by multiple users, each would have his own suffix).

The Classes

The classes are also a new concept in sliceO-5. By default, the "Study" and "Series" classes are active, giving the same kind of tree that was present in previous sliceOmatics. But you now have full control over the composition of the database. This opens some new possibilities, like the Multiplex class to segment slices that have multiple acquisitions.

Multi Windows and Multi Modes

You can have up to 4 active windows (like in 4.3) and each of these can be associated with a different mode (that's new!). So, for example, you could have one window showing the frames in Morpho, another window showing them in Edit, a third showing them in Regin Growing and the fourth showing the frames in 3D. You can easily edit the frames in Morpho or Region Growing, then touch-it-up in edit all the while viewing the results in 3D. Just clicking in a window will activate the associated mode.

The Surface/Volume Tool

The Surface/Volume tool has been redesigned to be more flexible and to make it easier to create the result files you want. It now has its own configuration menu.

The Modularity

The biggest changes to the program are beneath the surface. It has been completely redesigned to be modular. Each class, tool and mode is now a separate DLL library. You can add a new module simply by placing its DLL in the appropriate directory before the program is loaded.

The User Classes, Tools and Modes

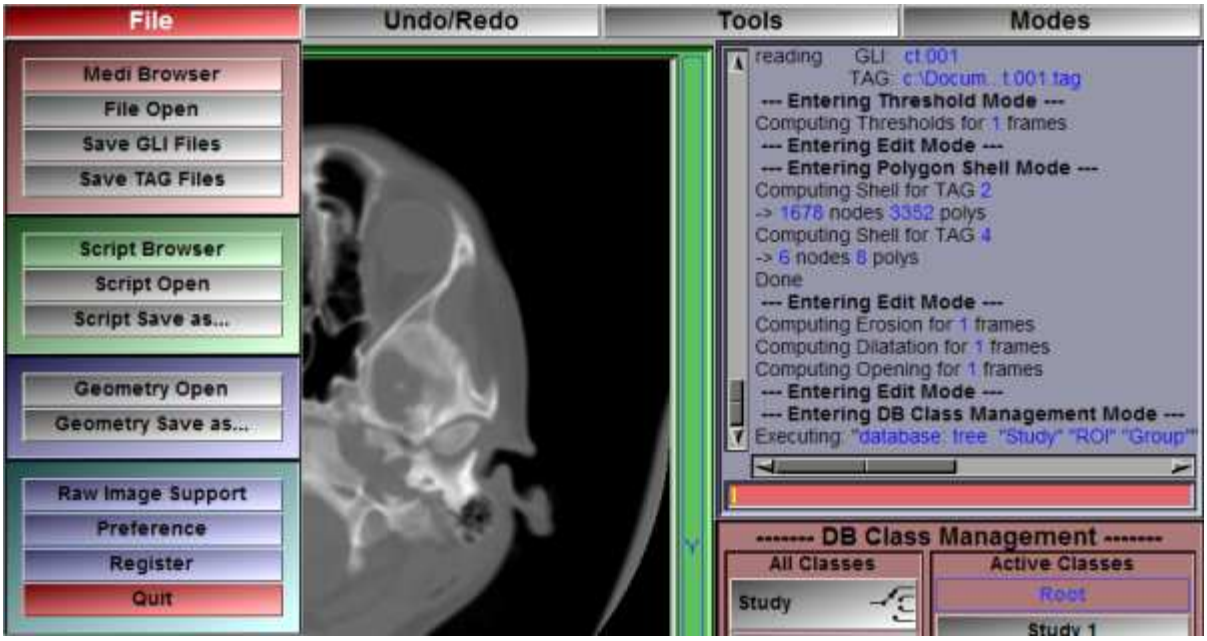
On top of its modularity, the structure of the program is now open. This means that anybody who wants to can create a new module for the program. The license system has also been redesigned to enable the user modules to be protected with their own license within our dongle. So if somebody would like to create a new segmentation module, but does not want to give it away for free, he can protect it with a license (we will provide the license generation program) and sell the license for his module.

The Main Menu

The Main Menu is located at the top of the screen. Its 4 buttons will activate 4 sub-menus: **File**, **Undo/Redo**, **Tools** and **Modes**.

Main Menu:
The File Menu

The File menu contains 13 buttons presented in 4 groups:



The buttons of the first group are used to manipulate image files. With these, you can input images in sliceOmatic, either with the **"DICOM Browser"** or **"Medi Browser"** interfaces or the **"File Open"** window. You can also save the results of the manipulation of the grey-level images with the "**Save GLI Files**" and of the segmentation with the **"Save TAG Files"** window.

The buttons of the second group are used to manipulate script files. The script files are used to read multiple images and do some basic manipulations on these. You can read a script file either with the **"Script Browser"** interface or the **"Script Open"** window. You can also create a script that will reopen all the files currently opened in sliceOmatic with the **"Script Save as..."** window.

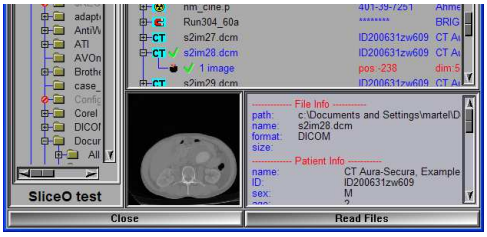
The buttons of the third group are used to manipulate polygonal surface files. You can read a surface file with the **"Read Geometry"** window and save the active surfaces with the **"Save Geometry as ..."** window.

The buttons of the last group allow you to access the **Raw_Header** program in order to read images from unsupported format, the **Configuration Menu** to customize sliceOmatic's interface, the **Tomovision_License** program to register your sliceOmatic modules, and finally to **Quit** the program.

The "DICOM Browser" and "Medi Browser" button

These buttons will start the **DICOM Browser** or the **Medi Browser**. These Explorer-like browsers enables you to preview and select the images you

want to read in sliceOmatic.

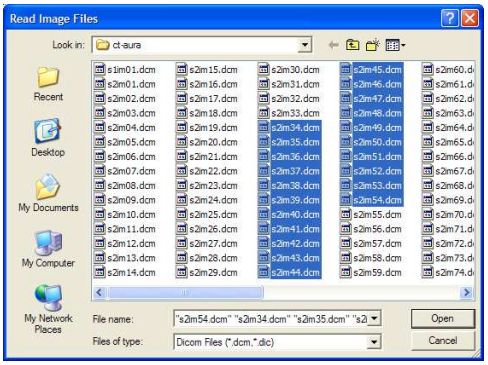


For a detailed description of the **DICOM Browser**, please refer to The DICOM Browser section.

For a detailed description of the **Medi Browser**, please refer to The Medi Browser section.

The "File Open" button

If you already know the name of the image file you want to read, this button will activate a Windows file browser. From this window, you can select multiple file names. All the selected files will be read into sliceOmatic when you click the "OK" button.



The "Save GLI Files" button

Pressing this button will cause sliceOmatic to save all selected GLI files if these files are not in their original orientation.

Note:

Only the files that are in a different orientation than the original images will be saved. In other words, this option will only save images created in the re-slice mode.

License:

If you do not have the **sliceOmatic Module** license, this button will be disabled.

The "Save TAG Files" button

Pressing this button will cause sliceOmatic to save all modified TAG files. If the file does not already exist, it will be created either in the same directory as the GLI file, (with the same name and the ".tag" extension) or in a specified directory, depending on the selected configurations (See the TAG section of **The Configuration Menu**).

Note:

Only the files that have been modified will be saved.

License:

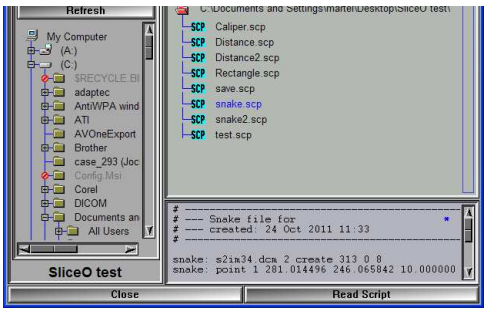
If you do not have the **sliceOmatic Module** license, this button will be disabled.

The "Script Browser" button

This button will start the **Script_Browser**. This

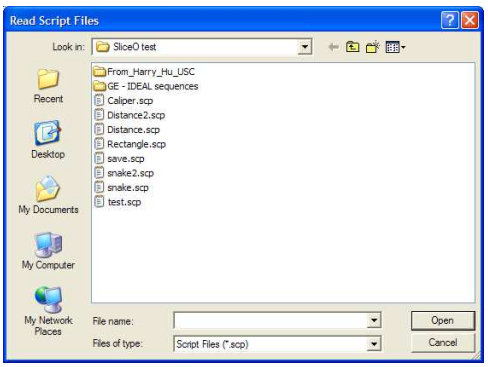
Explorer-like program enables you to preview and select the scripts you want to read in sliceOmatic.

For a detailed description of the **Script_Browser**, please refer to The Script_Browser **section.**



The "Script Open" button

If you already know the name of the script file you want to read, this button will activate a Windows file browser. From this window, you can select the desired script file. The selected file will be read into sliceOmatic when you click the "Open" button.



The "Save Script as ..." button

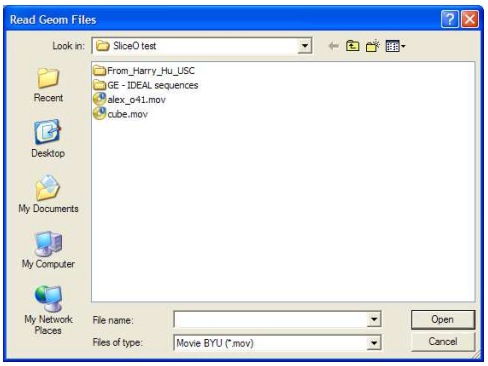
This button will activate a Windows file browser. From this window, you can enter the desired name and location for the script file. When you click the "Save" button, a script file containing the name, orientation and order of all the currently opened files will be written at the desired location. You can use this script to reload the images you are currently working on.

License:

If you do not have the **sliceOmatic Module** license, this button will be disabled.

The "Geometry Open" button

This button will activate a Windows file browser. From this window, you can select one geometry file. The selected file will be read into sliceOmatic when you click the "Open" button. The program can read geometry in MOVIE.BYU and STL format.



The "Save Geometry as ..." button

This button will activate a Windows file browser. From this window, you can enter the desired name and location for the geometry file.

You can save your geometries in one of the following formats:

mov Movie.BYU format. This old polygon format, developed by Brigham Young University for their MOVIE.BYU program in the 80's, is very simple which make it ideal to store simple polygonal objects such as those created in sliceOmatic. The format is described in more detail in appendix E. SliceOmatic can create ".dxf" files, but it cannot read them.

dxf Autocad file format. The geometries will be saved using the POLYLINE primitives. A definition of the DXF file format can be found in Wikipedia. SliceOmatic can create ".dxf" files, but it cannot read them.

stl & stl_ascii Stereolithography file format (binary version and ASCII versions). A definition of the STL file format can be found in Wikipedia.

Note:

STL format can only describe 1 object. If multiple objects are selected, sliceOmatic will create 1 file per object.

obj Wavefront OBJ file format. A definition of the OBJ file format can be found in Wikipedia. SliceOmatic can create ".obj" files, but it cannot read them.

wrl A VRML polygon file. A definition of the VRML file format can be found in Wikipedia. SliceOmatic can create ".wrl" files, but it cannot read them.

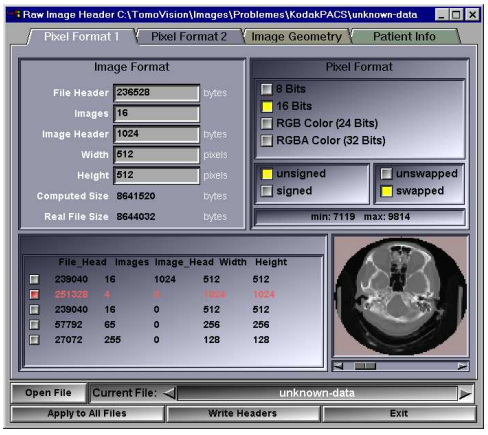
License:

If you do not have the **sliceOmatic Module** license, this button will be disabled.

The "Raw Image Support" button

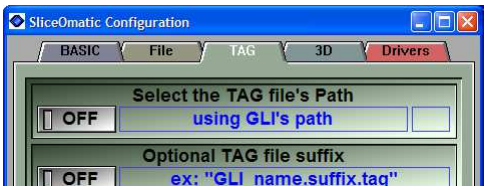
This button will start the **Raw Header** program.

For a detailed description of the **Raw Header** program, please refer to **The Raw Header Program.**



The "Config" button

This button will open the configuration menu to help you setup your startup preferences. Any changes you make to the preferences will be visible the next time you start sliceOmatic.

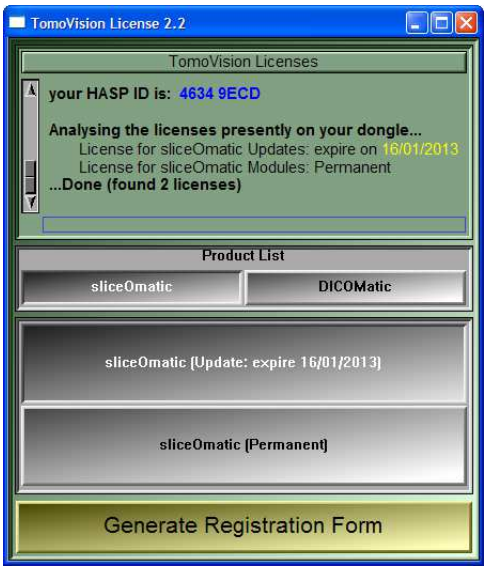


A detailed description of this menu is provided further down in The Configuration Menu section.

The “Register” button

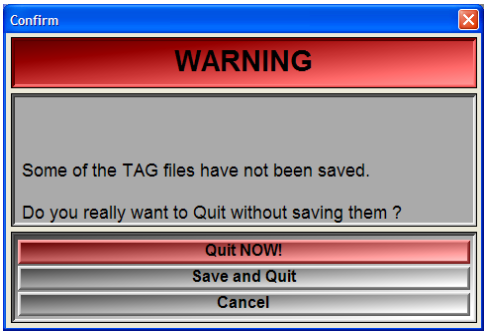
This button will start the **Tomovision_License** program in order to register your sliceOmatic modules.

A detailed description of the Tomovision_License program is provided in the "The License" section of the **Installation Guide**.



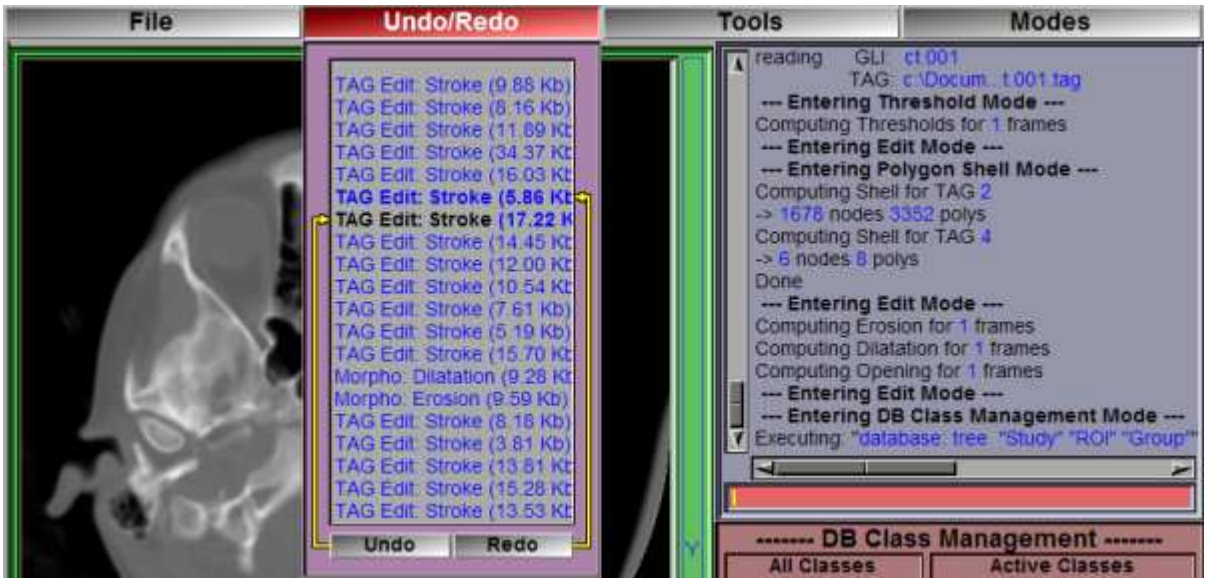
The “Quit” button

This button causes you to exit from the program. If the program detects any modifications to the TAG images since you last saved them with the **“Save TAG Files**” button, a warning pop-up will prompt you to save the changes or lose them.



Main Menu:
Undo / Redo

Most (but not all) of the actions that change pixel values (either GLI or TAG) can be undone. By default, the Undo buffer keeps track of your 20 last actions. These will be displayed in this menu. All “undone” actions can be “redone” until you perform a new action.



Clicking on the **Undo** button will undo the action associated with the button by the arrow on the left side of the menu. This action will be displayed in **Bold Black** font.

Clicking on the **Redo** button will redo the action associated with the button by the arrow on the right side of the menu. This action will be displayed in **Bold Blue** font.

Some of the mode windows will also have an Undo button; clicking on it is equivalent to the Undo button in this menu.

The Undo and the Memory

Undo operations can take a lot of memory. To try to limit this, sliceOmatic lets you configure two parameters:

- The maximum number of Undo,
- The maximum size of each Undo.

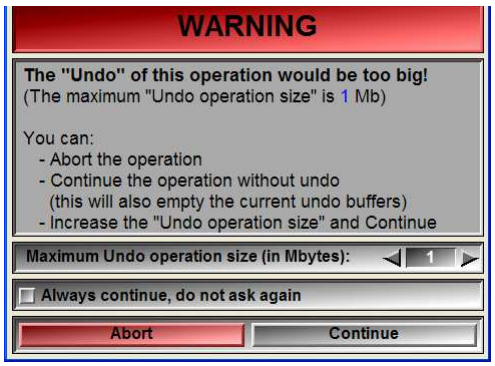
Both parameters can be changed with the configuration menu or with the commands found in the "SliceO_System\Undo.dll" section of **Appendix B**.

Setting the number of Undo to 0 will disable all Undo operations.

Also, if you are running low on memory, one of the options of the Memory Manager tool is to clear the Undo memory. This will erase all Undo operations currently in memory.

If an operation would take more Undo memory than 

the maximum allowed for one operation, a warning pop-up will appear. You will be given the choice of aborting the current operation, continue the operation but lose the Undo for it or increase the maximum size of an Undo operation.



This warning pop-up will tend to appear if you operate on a large number of images at once. You will also have the occasion to check the “always continue” box so that the pop-up does not appear anymore. Instead a warning message will be displayed in the text window each time the Undo limit is exceeded and the Undo possibilities are lost.

The Undo/Redo and the Load/Restore operations

It is easy to confuse the Undo/Redo and the Load/Restore operations. The Undo/Redo enables you to backtrack your last brush strokes. The Load/Restore gives you more control on what you do.

Most of the tag modes in sliceOmatic will give you the possibility to affect the pixels with the left mouse button and to “Restore” or bring back the previous values of the pixels with the right mouse button. It is different from the Undo in the sense that you “unpaint” with the mouse, choosing which pixels will be restored. These “Restore” values come from a saved copy of the tag image: the “Restore buffer”. You can load the current tag image in the Restore buffer at any time with the “Load” command. By default, the tag image is automatically loaded in the Restore buffer the first time you modify its tag values.

Please note that restoring the image with the right mouse button is a brush stroke and this operation can be undone with the Undo/Redo operations.

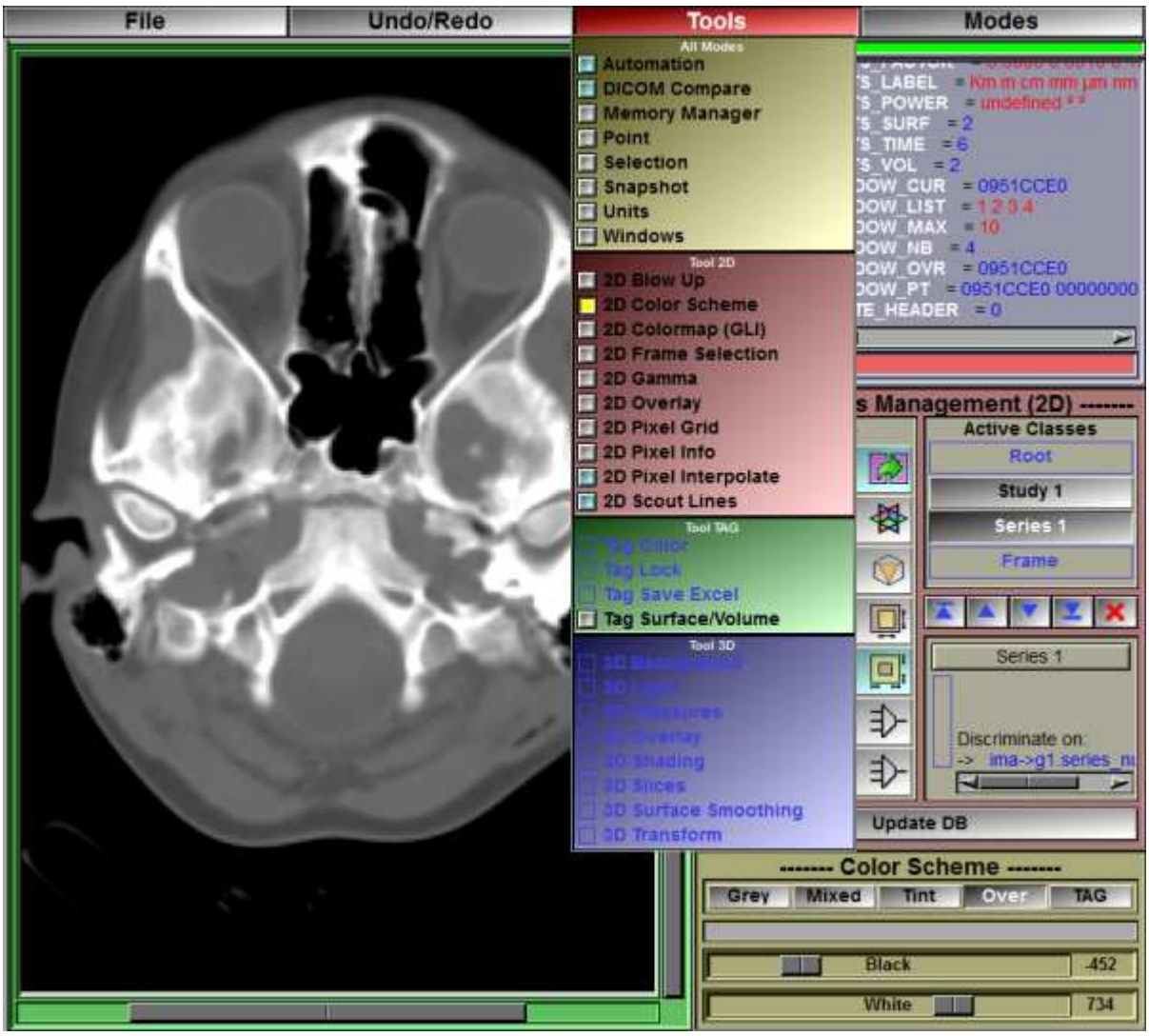
The following key shortcuts can also be used:

Key map	*Action*
CTRL_Z	Undo the last operation
CTRL_Y	Redo the last Undo

The commands related to the Undo/Redo and Load/Restore operations can be found in the "SliceO_System\Undo.dll" section of **Appendix B**.

Main Menu:
The Tools

This menu will enable you to select which of the tools you want to be visible in the control area. The available tools are grouped in 5 categories: "All Modes", "2D, "TAG", "3D" and "Other". Depending on the current mode, certain categories of tools may be unavailable (for example, TAG tools such as "Tag Lock" are only available in one of the segmentation modes). The unavailable tools will have their buttons in blue outline. For the other ones, just clicking on their buttons will activate the tool. Active tools will have their button yellow. Tools from sliceOmatic basic package have a grey button, while tools imported from the "Additional Modules" section of the Web download pages have a cyan button.



The All Modes Tools: These are the tools that are independent of the 2D/3D information, they are available in all Modes.

The 2D Tools: These are tools that are only available if the current Mode is one of the 2D Modes.

The 3D Tools: These are tools that are only available if the current Mode is one of the 3D Modes.

The TAG Tools: These are tools that are only available if the current Mode is one of the Segmentation Modes.

The Other Tools: By default this section is empty. It is only present in the interface if one of the tools can not be placed in its correct location.

Note:

The actual order of the tools in the list is based on the either the "Sort" name provided by the "Sort" function of the DLL or the name of the associated DLL files if the "Sort" function is not present. By convention the sort name for "All Modes" tools start with "0D", 2D with "2D", 3D with "3D ", TAG with "TAG". If a DLL does not respect this convention, its position in the list may not be correct. This has no adverse effect other than aesthetics.

Warning:

If the program is not full screen or if the screen resolution is not sufficient, some of the tools may not have enough room to be displayed in the tool control area. If this happens, you will see the following warning in the text window:

WARNING: Unable to display tool! increase window size, screen resolution or decrease interface size in the preferences menu.

To fix this, you can either increase the size of sliceOmatic's window or decrease the size of the tools through the Configuration Menu.

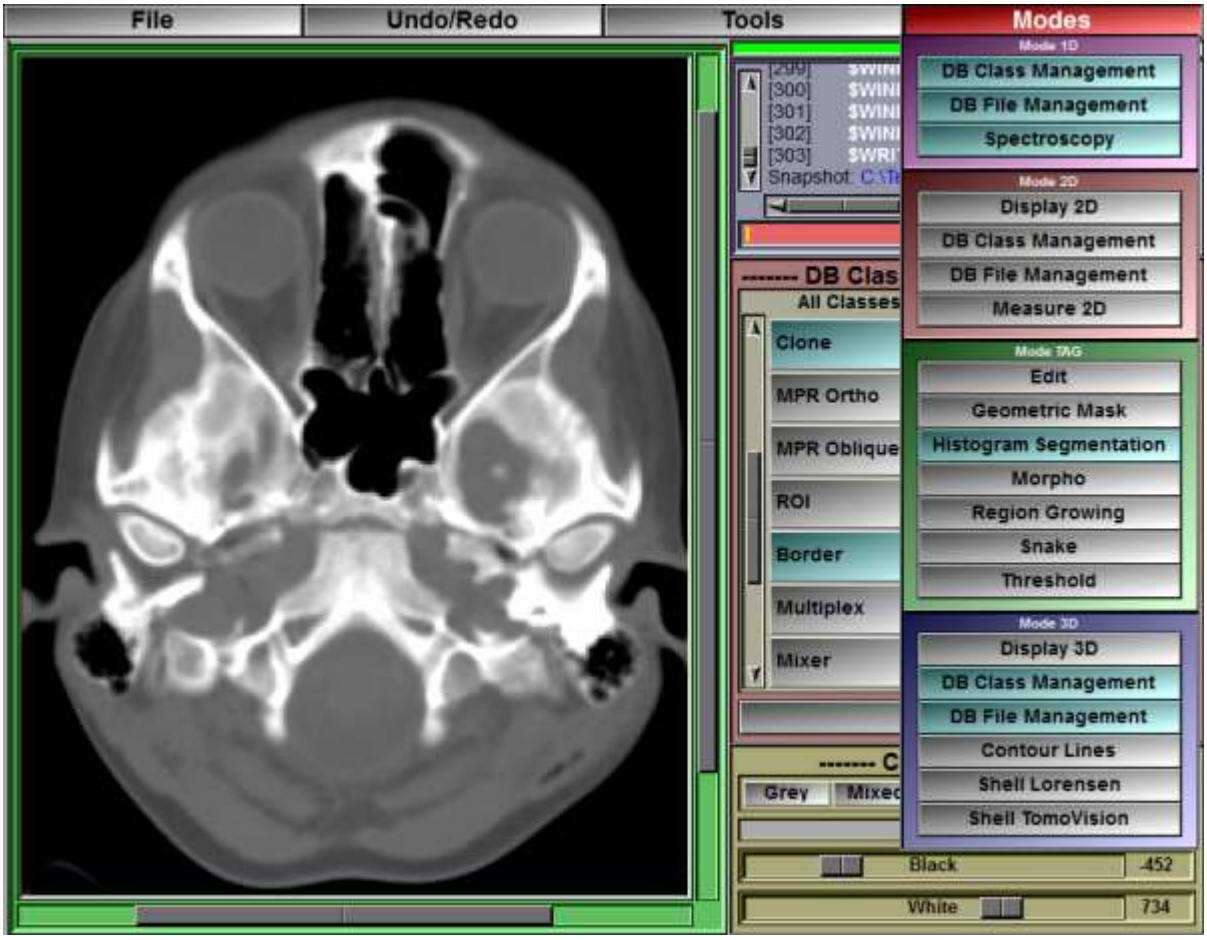
Each of the available tools are described in more detail in the following sections.

If you select more tools than the program can fit in the tool control area, the latest selections will be given priority.

Main Menu:
The Modes

This menu is used to select the operation mode of the program. The File menu contains a number of buttons presented in 3 to 7 groups: "Mode 1D", "Class 1D", "Mode 2D", "Class 2D", ModeTAG", "Mode 3D" and "Class 3D". If a section does not have any item, it will not be displayed. For example, if the 2D database tree does not contain any class that has an interface, then there will not be any items in the "Class 2D" section and it will not be displayed. Modes from sliceOmatic basic package have a grey button, while modes imported from the "Additional Modules" section of the Web download pages have a cyan button.

Clicking on a mode button opens this mode's interface window and associates this mode to the current window. In this way, you can associate different modes to different windows (see "2D and 3D windows" in the Display Area).

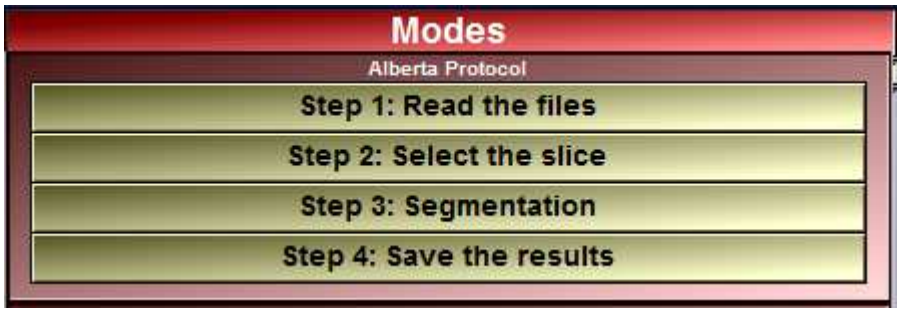


The Simplified Modes: This optional group of buttons is used to simplify the processing of slices using predefined scripts. For example, if the program is invoked with the "Alberta Protocol" shortcut, this group will be named "Alberta Protocol" and will contain 4 buttons named "Step-1" to "Step-4".

The 1D Modes:	This group of buttons is for operations on the 1D database or for 1D manipulation of the images. A detailed description of each of these modes is given in the "The Additional Modules" section.
The 1D Classes:	This group is for interfaces that are associated with classes present in the 2D database. If none of the database classes have an interface, then this group will not be present.
The 2D Modes:	This group of buttons is for operations on the 2D database or for 2D manipulation of the images. A detailed description of each of these modes is given in the "The 2D Modules" section.
The 2D Classes:	This group is for interfaces that are associated with classes present in the 2D database. If none of the database classes have an interface, then this group will not be present.
The TAG Modes:	This group is for segmentation modes.
The 3D Modes:	This group of buttons is for operations on the 3D database or for 3D manipulation of the images and geometries.
The 3D Classes:	This group is for interfaces that are associated with classes present in the 2D database. If none of the database classes have an interface, then this group will not be present.

The Alberta Protocolo

If the program is started with the "Alberta Protocol" shortcut from the start menu, a new group of buttons will be available in the Mode menu.



Once activated, either by starting the program through the "Alberta Protocol" shortcut, or with the configuration interface, the "Modes" menu will have a new group of 4 buttons. Each of these buttons will be associated with one of the protocol's steps.

Clicking on one of these buttons will open a the matching help page to guide you through the actions required for that step. It will also set the modes, tools and windows of the sliceOmatic interface to facilitate these actions.

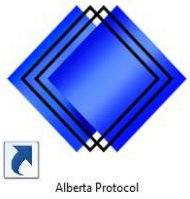
Once you are familiar enough with the protocol, you can disable the opening of the help pages with the configuration interface.

You can also access the help pages associated with the new buttons from here: **Alberta Protocol man pages**.

The Alberta Protocolo:
The simplified menus

Starting the "Alberta Protocol" form the shortcut:

The "Alberta Prortocol" simplified menu can be activated by starting the program with the "Alberta Protocl" shortcut. If the shortcut is not on your desktop, it can be found in sliceOmatic's installation directory (by default: c:\Program Files\TomoVision).



Using the "Alberta Protocol" shortcut will start sliceOmatic with 2 command line arguments:
-s "Sarcopenia"
-n "Alberta Protocol"

Having the "**-s**" command line argument set the following system variables:

- $INTERFACE_FLAG will be set to 3, meaning the simplified menus and the associated help pages are on.
- $INTERFACE_ROOT will be set to "Sarcopenia", meaning that the simplified menus will be associated with the script files found in the Sarcopenia sub-directory of the installation directory.

Having the "**-n**" command line argument set the following system variables:

- $INTERFACE_NAME will be set to "Alberta Protocol", meaning the name of the new group of buttons in the Mode menu will be "Alberta Protocol".

Starting the "Alberta Protocol" from the configuration interface:

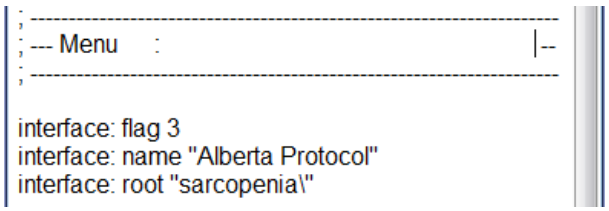
You can also enable the protocol by setting the values of the "Menu" page in the configuratior done, if you click "Apply & Save", not only will you enable the simplified menus, but it will be e re-start the program, even if you do not do it from the ""Alberta Protocol" shortcut.

- You can enable/disable the new menu group with the "Show simplified menus?" choice.
- The sub-directory containing the protocol's script is specified with the "Menu's dir" line.
- The name of the new menu group is given by the "Menu's name" line.

- You can enable/disable the new menu group with the "Display help page?" choice.

If you save the configuration, then the "sliceOmatic_ini.scp" will contain the matching commands in the "Menu" section.

The "Alberta Protocol" configuration file:



The configuration file has 3 commands controlling the simplified menu interface:

interface: flag *value*
Where value is a bit flag, with bit 0x01 controlling the presence of the simplified menu, and bit 0x02 the opening of the help page associated with each buttons.

interface: name *menu_name*
Where "menu name" is the name associated with the new mode group. This name will be copied to the $INTERFACE_NAME variable

interface: root *dir_name*
Where "dir_name" is the directory containing the scripts associated with the new menu buttons. This directory's name will be copied to the $INTERFACE_ROOT variable.

The "step_x" scripts:

The $INTERFACE_ROOT directory contain the script files that will be used in the simplified interface. These scripts are named "step_***x***.scp", with ***x*** going from 1 to n. There will be 1 menu button per consecutive step script in the directory, starting with step 1.

Each of these scripts will pre-set the program's interface to help the user in the task specific to that step.

There are 2 commands in these scripts that are unique only to the simplified menu interface: The "**name**" and the "**html**" commands.

- The "**name**" command give the name of the button (this will be appended to the "step_x" name). for example, if step_1.scp contain the command "name: "Read the files"", then the first button of the new mode group will be called "step 1: Read the files".

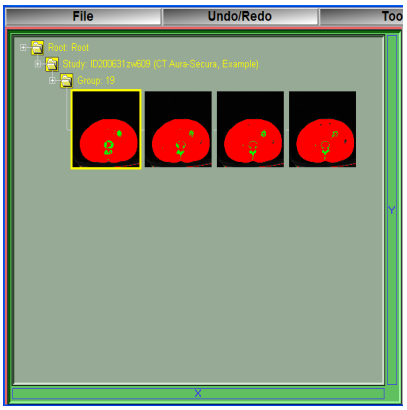
- The "**html**" command give the address of a web page associated with the button. If the configuration parameter "Display help page?" has not been set to off, the associated web page will be open once when each menu button is pressed.

The Step_0.scp script is a special case. It is not associated with a button in the Modes menu, instead it is executed at startup.

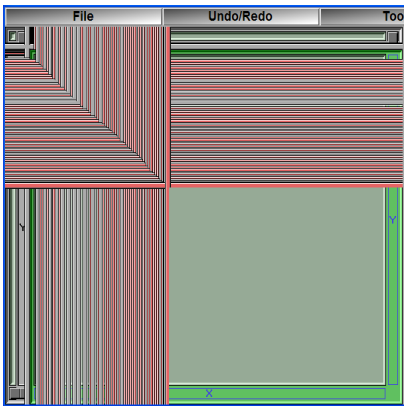
The Display Area

In this screen area, you can see the images that have been loaded in the program.

This area can be split in up to 4 display windows. Each of these windows can show the 2D images or 3D geometries depending on the current mode associated with the window. You can split the display area in multiple windows either by dragging the separator tools at the top and left hand side of the area, or with the "Windows" tool.



Grab both separators



Drag them

You now have 4 windows

When showing 2D frames, the window can be in either the **Mode One** or the **Mode All** modes. In Mode One, only one 2D frame is shown at a time. In Mode All, all the frames can be viewed simultaneously. You can toggle between these modes either by pressing the "**space-bar**" while the cursor is within the window, or with the "**ALL**" and "**ONE**" buttons in the Frame Selection Tool.

In the 3D modes, the window is used to display the 3D surfaces or volumes. The orientation, position and size of the 3D geometries can be interactively modified with the mouse. You have a choice of 3 interface styles. See the Transform tool section in **The 3D Tools** for more information on the possible interfaces.

The Display Area:
2D and 3D windows

A window is either 2D or 3D depending on the current mode that is associated with it. For example if you select "Edit" from the Mode menu, then the current window will be a 2D window and it will be associated with the Edit Mode. If you select "Contour Lines" from the Mode menu, then the current window will be a 3D window and it will be associated with the Contour Lines Mode.

Each window has its associated mode. If multiple windows are present in the display area, clicking on one window will make it the current window. The associated mode will automatically be presented in the control area, and the tools that are selected and are compatible with this Mode will be also be displayed.

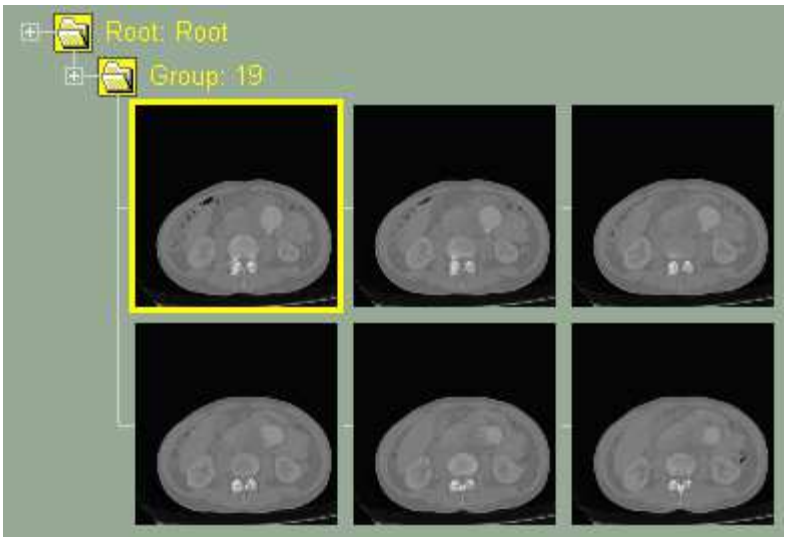
2D Windows: Mode All and Mode One

The 2D windows can be used to display either all your frames (Mode All) at once or just one frame at a time (Mode One).

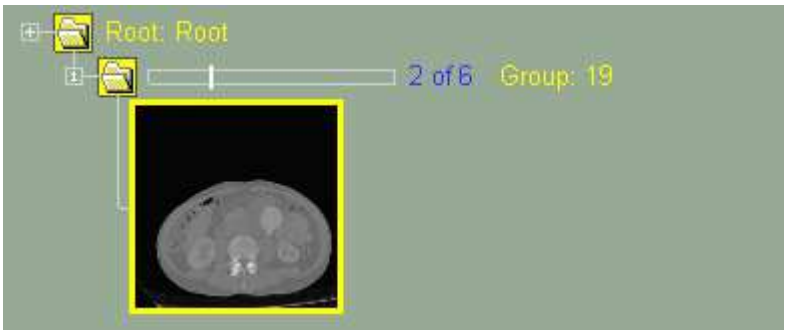
Mode All The database tree with all the frames is displayed. If all the frames cannot fit on the screen, you can use the vertical slider to pan through them.

The complete tree, anchored at the root node, is shown.. Each branch of the tree that has more than 1 child can be in one of 4 states:

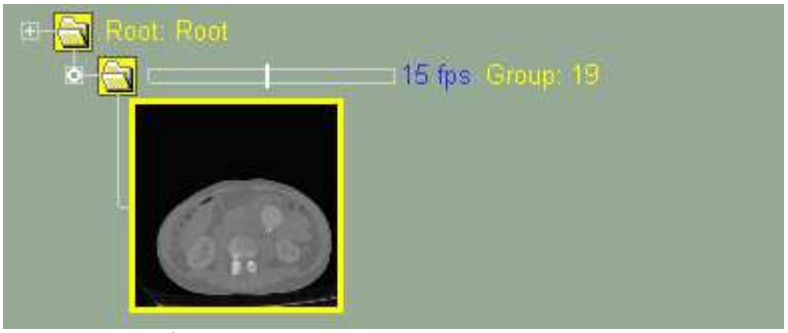
Expand All the children of this branch are displayed.



Single Only one of the children is shown at a time. The slider can be used to select which child is visible.



Loop Only one of the children is shown at a time. The children are cycled automatically after a short delay. The slider can be used to adjust this delay from 1 sec to $^{1}/_{30}$ sec.



Collapsed None of the children are visible

Branches that have only 1 child can only be in "Expand" or "Collapse" states.

The state of a branch is changed by clicking on the symbol at the left hand side of the branch

The current frame will be highlighted by a yellow border, and all the branches higher in the tree that contains the current frame will have their icons highlighted in yellow.
To help distinguish between selected and unselected images, unselected images will be drawn without their red component.

Mode One

Only the current image is displayed. If it cannot fit on the screen, you can use the horizontal and vertical sliders to pan through the image.

Pressing the “Ctrl” key can also be used to drag the image with the mouse.

The displayed image is selected either with the **Slice Selection Tool**, from the keyboard shortcuts, or by clicking on the desired frame in Mode All and switching to Mode One.

In Mode One, only the current frame is selected. This means that most operations (for example, thresholding the images in the Threshold Mode) will only be done on the current frame.

Switching from Mode One to Mode All will also select all frames. This means that most operations will be done on all the frames.

3D Windows

3D windows show objects in 3D space. By default, all 3D polygonal objects loaded or created in sliceO are visible (you can use the "3D Shading" tool to change their appearance and visibility). In certain modes ("Contour Lines", "Shell") you will also see the contours of the TAG surfaces from each selected frames. And if you activate the "3D Slices" tool, you can see the selected frames in 3D space.

The Display Area:
Frame Selection

Most of the Actions in sliceOmatic will be performed on all the selected frames. For example, in the Tag Surface/Volume tool, the computations are performed on all the selected frames.

By default, in Mode One, the displayed slice is the only one selected, and in Mode All, all the slices are selected. This selection however can be changed, either from the "Selection" tool, from the command line, or with the mouse in Mode All. Unselected frames will be displayed without their red component.

From the Display Area in Mode ALL

- Clicking outside the frames will unselect all frames.
- Clicking outside the frames and dragging the mouse will select all the frames inside the selection box.
- Clicking on the icon of the db tree will select all the frames in this branch.
- Clicking on an frame will select this frame.
- Clicking on the icon of the db tree while pressing the CTRL key will toggle the selection of all the frames in this branch.
- Clicking on a frame while pressing the CTRL key will toggle the selection of this frame. (Note: If the frame is already selected and it is not the current one, the first click will be used to make it the current one, the second click will do the selection toggling).
- Clicking on an frame while pressing the SHIFT key will select all frames from the last selected frame to the pointed frame.

From the Keyboard

The following key shortcuts can also be used:

Key map	*Action*
Space-bar	Toggle between MODE ONE and MODE ALL. When going to MODE ONE, only the current frame is selected. When going to MODE ALL, all the frames are selected.

Note:

Switching from Mode ALL to Mode ONE will change the frame selection. In Mode ONE only the current frame is selected. When going to MODE ALL, all the frames are selected.

The Databases

When you read a file in sliceOmatic, the program will place them in a database of files. This database can be examined and manipulated through the "DB File Management" modules.

Then the data contained in the files is placed in a new database arranged in a tree structure. This tree structure can be examined and manipulated through the "DB Class management" modules.

The program will keep 3 database of files and 3 trees. One each for 1D, 2D and 3D structures.

Each tree is composed of "Nodes". Some nodes are always present (such as the "Root" and "Frames" for the 2D tree), and others are created by inserting "Classes" in the architecture of the tree. This is done with the "DB Class management" module. Once the architecture is defined, the program use this information along with the files in the database to create the nodes of the trees. (This is done by reading a new file, or by clicking on the "Update DB" button.)

Each of the 3 trees (1D, 2D and 3D) have their own classes that can be used in their architecture.

The Databases:
The 1D Database and tree

The 1D structures are used for datasets such as the spectroscopy data.

The 1D classes available in the basic package are:
- Study
- Series
- Position

The Databases:
The 2D Database and tree

The 2D structures are used for images.

The 2D classes available in the basic package are:
- Study
- Series
- DICOM Tree
- Convolution Filter
- Mixer
- MPR Ortho
- MPR Oblique
- Multiplex
- ROI
- Unsorted

You also have access to a number of additional classes through the additional modules:
- Border
- Clone
- Gradient Amplification
- Path
- RR2D
- RR3D
- Suffix

The Databases:
The 3D database and tree

The 3D structures are used for 3D surface models, either read from files or generated by the program.

The 3D classes available in the basic package are:

- Group
- Matrix

The 2D Classes

The 2D tree starts at the "Root" node, and finishes with "Frames" nodes. Each file that you read in sliceOmatic contains one or, sometimes, multiple images. Each images is a "frame". The root and the frames are both a "class" of nodes.

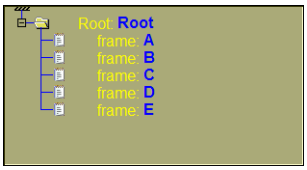
The root and the frames are always present in the database tree. But in between, you can add other classes.

Some of the classes, such as "Study", "Series" and "DICOM Tree" are used to sort the tree in patient/study/series/image hierarchy.

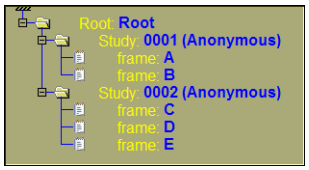
For example if we load 5 files in sliceOmatic, each having 1 image and with the following parameters:

Image files	Study ID	Series number
A	0001	1
B	0001	2
C	0002	1
D	0002	1
E	0002	2

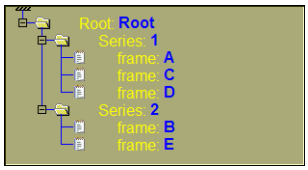
If we create different combinations of trees using the Study and Series classes on top of the default Root and Frames, we will have the following trees:



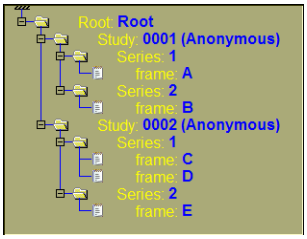
Root & Frame



+ Study



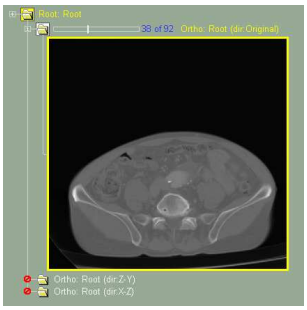
+ Series



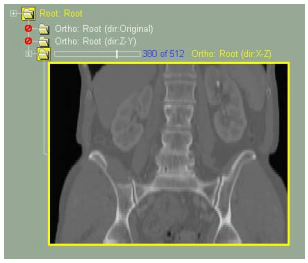
+ Study & Series

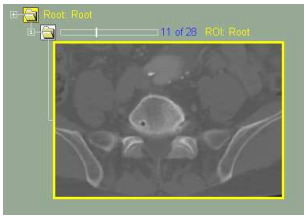
By default, the tree contains the root, study, series and the frame classes. This is to create a tree that is similar to the one present in sliceOmatic 4.3.

Other classes, such as "Filter", "MPR" and "ROI", are used to modify the GLI images.


Original Frame


Filter


MPR


ROI

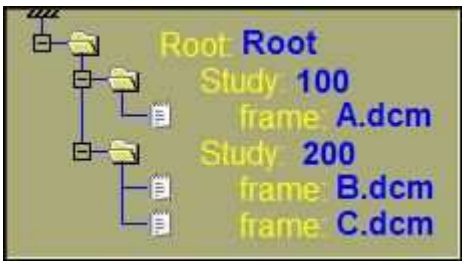
And other classes, such as "Multiplex" or "Mixer", are used to achieve specific results that were impossible in the previous versions of sliceOmatic.

The 2D Classes:
Study

If this class is placed in the DB tree, there will be one instance of this class per study.

It is usually placed immediately after the root class and is used to differentiate between the studies. It is more general than the DICOM Tree class with the tag (0020,0010) (study ID) since it can differentiate between studies of non-DICOM files.

For example, let us say that we read 3 files "A.dcm", "B.dcm, and "C.dcm", file A is part of study #100, and files B and C are part of study #200. If the class tree contains 3 classes: "Root", "Study" and "Frame", the resulting database tree will be:



From the Graphic Interface

This class does not have a graphic interface.

From the Display area

There is no display area interaction specific to this class.

From the Keyboard

There is no keyboard interface specific to this class.

From the Command Line

There is no command line for this class.

The 2D Classes:
Series

If this class is placed in the DB tree, there will be one instance of this class per series.

It is usually placed immediately after the study class and is used to differentiate between the series. It is more general than the DICOM Tree class with the tag (0020,0011) (series number) since it can differentiate between series of non-DICOM files.

From the Graphic Interface

This class does not have a graphic interface.

From the Display area

There is no display area interaction specific to this class.

From the Keyboard

There is no keyboard interface specific to this class.

From the Command Line

There is no command line for this class.

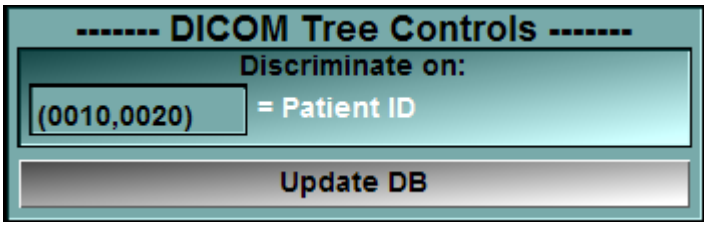
The 2D Classes:
DICOM Tree

There will be one instance of this class for each instance of the "discriminate" DICOM tag.

For example, if the discriminate tag is set to (0010,0020) (= Patient ID), and the frames contained in the parent class come from 2 patients, then 2 instances will be created, one for each patient. The parent class will have 2 children.

You can place this class at multiple levels in the database tree. It can have a different "discriminate" parameter at each level. This parameter can be set with a command line, or from the class graphic interface (see the example below).

From the Graphic Interface



DICOM tag Field Enter the DICOM TAG that will be used to discriminate between the files. The class will have one child per instance of this tag.

By default, if you create multiple levels of this class, they will each have a different discrimination value:

1rst level: (0010,0020) = Patient ID
2nd level: (0020,0010) = Study ID
3rd level: (0020,0011) = Series Number
4th level: (0020,0013) = Instance Number

Once the tag is entered, the program will display the tag's description (in white).

Update DB Recompute the DB tree using the new parameters.

From the Display area

There is no display area interaction specific to this class.

From the Keyboard

There is no keyboard interface specific to this class.

From the Command Line

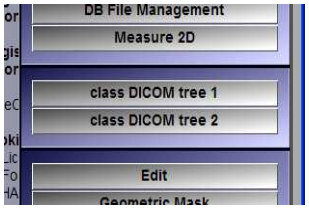
A list of the commands available while in this class can be found in the "SliceO_Class\DICOM_Tree.dll" section of **Appendix B**.

Example: 2 levels of DICOM Tree classes

From the "DB Class Management" mode:

- delete the default "Study" and "Series" classes
- insert the "DICOM Tree" class 2 times
- press on "Update DB".

The database will now have the root, DICOM Tree 1, DICOM Tree 2, and frames classes.



Each of the DICOM Tree classes level will now have a button in the Mode menu.

Selecting each of these in turn enables you to change their "discriminate" parameters.
However please note that by default, the first 4 levels of this class already have different "discriminate parameters" (1=Patient ID, 2=Study ID, 3=Series Number and 4=Instance Number).

The 2D Classes:
Path

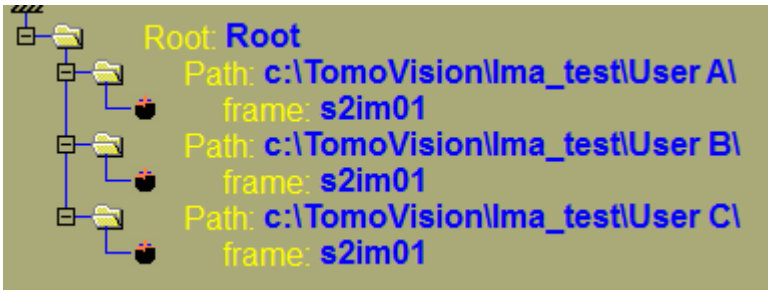
This class enable you to add sort the DB tree according to the path of the TAG files.

It is used if you have the same GLI image segmented multiple times and want to compare them. For example in repeatability studies. You can read the same GLI image in sliceOmatic but only if the associated TAG either have different suffix, or are from different directories. Since the GLI files are the same, the normal sorting classes ("Study", "Series" and "DICOM Tree") are unable to differentiate them. You can use "Path" and "Suffix" for this.

The Path class is enabled through the 2D Mode: "DB Class management" interface. If this class is placed in the DB tree, there will be one instance of this class per TAG file directory.

It is usually placed immediately after the root class and is used to differentiate between the studies.

For example, let us say that we to have 3 different users segment the same slice. We can create 3 directory ("User A", "User B" and "User C") and copy the GLI in each of these, then have each user segment the GLI from one of the directories. You can then read the 3 GLI and use the "path" class to differentiate each. The resulting database tree will be:



From the Graphic Interface

This class does not have a graphic interface.

From the Display area

There is no display area interaction specific to this class.

From the Keyboard

There is no keyboard interface specific to this class.

From the Command Line

There is no command line for this class.

The 2D Classes:
Suffix

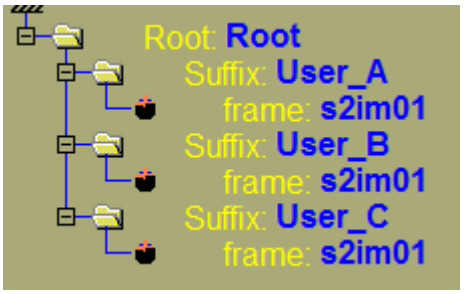
This class enable you to add sort the DB tree according to the path of the TAG files.

It is used if you have the same GLI image segmented multiple times and want to compare them. For example in repeatability studies. You can read the same GLI image in sliceOmatic but only if the associated TAG either have different suffix, or are from different directories. Since the GLI files are the same, the normal sorting classes ("Study", "Series" and "DICOM Tree") are unable to differentiate them. You can use "Path" and "Suffix" for this.

The Path class is enabled through the 2D Mode: "DB Class management" interface. If this class is placed in the DB tree, there will be one instance of this class per TAG file directory.

It is usually placed immediately after the root class and is used to differentiate between the studies.

For example, let us say that we to have 3 different users segment the same slice. We can have each user segment the GLI and save the TAG results using a different suffix ("User_A", "User_B" and "User_C"). You can then read the GLI 3 time changing the desired suffix each time from the configuration menu and use the "suffix" class to differentiate each. The resulting database tree will be:



From the Graphic Interface

This class does not have a graphic interface.

From the Display area

There is no display area interaction specific to this class.

From the Keyboard

There is no keyboard interface specific to this class.

From the Command Line

There is no command line for this class.

The 2D Classes:
MPR Oblique

Each instance of this class will have 2 children:
- The non-modified frames in their original direction,
- A set of new frames in the oblique direction.

Each of these children can be turned on or off.

From the Graphic Interface

Instance selection Select which of the instances of the class you want to work on.

On / Off You can turn on or off one of the MPR branches. If for example, you only want the oblique frames and will not use the original frames, you can set the "Dir. Original" to off and the "Dir. Oblique" to on.

Dir. Original These are the original non-modified frames. The yellow indicator signals that the frames are to be present in the database tree.

Dir. Oblique These are the new oblique frames. The yellow indicator signals that the frames are to be present in the database tree.

Rot. Alpha sliders The oblique plane used to compute the new oblique frames is defined by 3 rotations: Alpha, Beta and Theta. Alpha and Beta can be set with the horizontal and vertical sliders.

Rot. Beta sliders Theta can only be changed through the commands.

Slice preview You can see a preview of each frame. The slider lets you select the frames that are previewed.

Update DB Recompute the database taking into account the settings you selected.

From the Display Area

There is no display area interaction specific to this class.

From the Keyboard

There is no keyboard interface specific to this class.

From the Command Line

There is no command line for this class.

The 2D Classes:
MPR Ortho

Each instance of this class will have 3 children:
- The frames in their original direction,
- A set of new frames in the Z-Y plane direction,
- A set of new frames in the X-Y plane direction.

Each of these children can be turned on or off.

From the Graphic Interface

Instance selection Select which of the instances of the class you want to work on.

On / Off You can turn on or off one of the MPR branches. If for example, you only want the Z-Y plane frames and will not use the original frames, you can set the "X-Y" and "X-Z" to off and the "Z-Y" to on.

Dir. X-Y Enable/Disable the frames that are in the original plane. The yellow indicator signals that the frames are to be present in the database tree. By default, these are "On".

Dir. Z-Y	These are frames that are in a plane perpendicular to the original frames plane. This new plane is created by rotating the old plane 90 degrees around the "Y" axis. The yellow indicator signals that the frames are to be present in the database tree. By default, these are "Off".
Dir. X-Z	These are frames that are in a plane perpendicular to the original frames plane. This new plane is created by rotating the old plane 90 degrees around the "X" axis. The yellow indicator signals that the frames are to be present in the database tree. By default, these are "Off".
Restore	Remove any rotations and flips
Rotations	Rotate the frames + or - 90 degrees in the select plane (around an axis perpendicular to the current plane).
Flips	Rotate the frames 180 degrees around the vertical or horizontal axis of the current plane.
Slice preview	You can see a preview of the region of interest applied to each frame. The slider lets you select the frames in this class.
Update DB	Recompute the database taking into account the settings you selected.

From the Display Area

There is no display area interaction specific to this class.

From the Keyboard

There is no keyboard interface specific to this class.

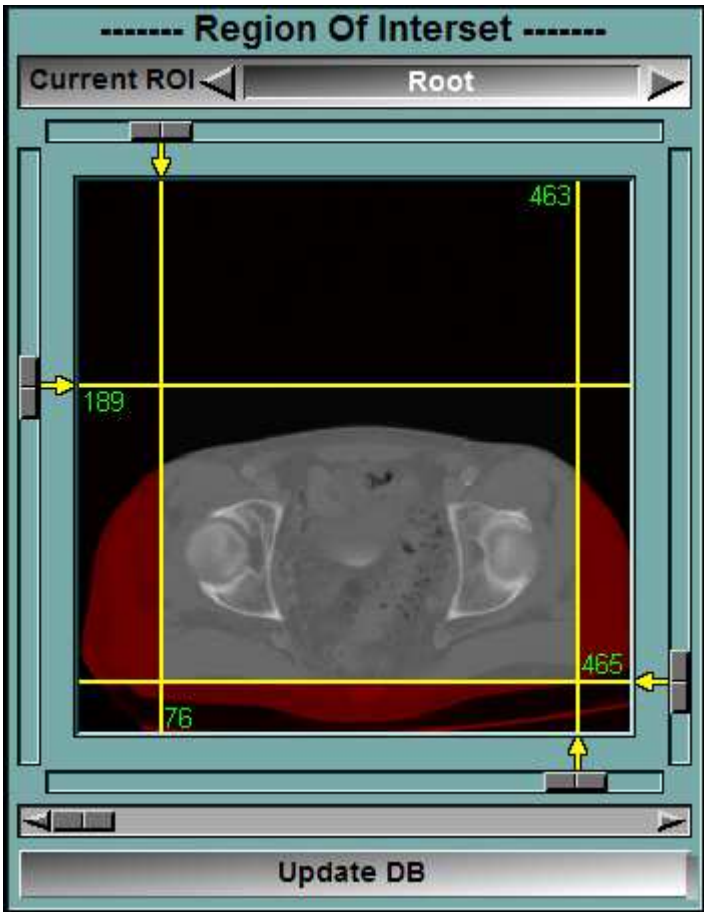
From the Command Line

There is no command line for this class.

The 2D Classes:
ROI

This class enables you to select a portion of the frames to work on.

From the Graphic Interface



Instance selection Select which of the instances of the class you want to work on.

Left/Right/ Top/Bottom sliders Move these sliders to select the region of interest. In the preview image, portions of the frames that are outside the ROI will be displayed with only their red component.

Slice preview You can see a preview of the region of interest applied to each frame. The slider lets you select the frames in this class.

Update DB Recompute the database taking into account the settings you selected.

From the Display Area

There is no display area interaction specific to this class.

From the Keyboard

There is no keyboard interface specific to this class.

From the Command Line

There is no command line for this class.

The 2D Classes:
Multiplex

This class enables multiple GLI frames to be associated to the same TAG frames.

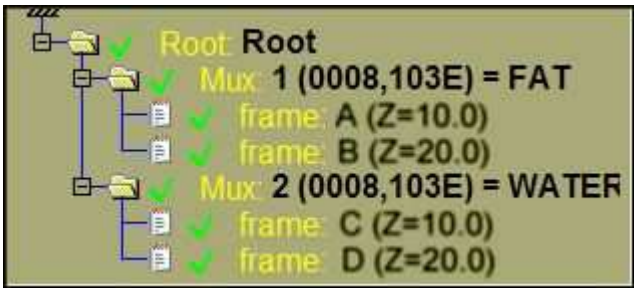
Multiple instances of the class will be created depending on the "discriminate" DICOM tag. Each of these classes will be a "channel". In each of these channels, the frames that share the same value for the "associate" DICOM tag will share the same TAG file.

The actual TAG files will be associated with the selected channel.

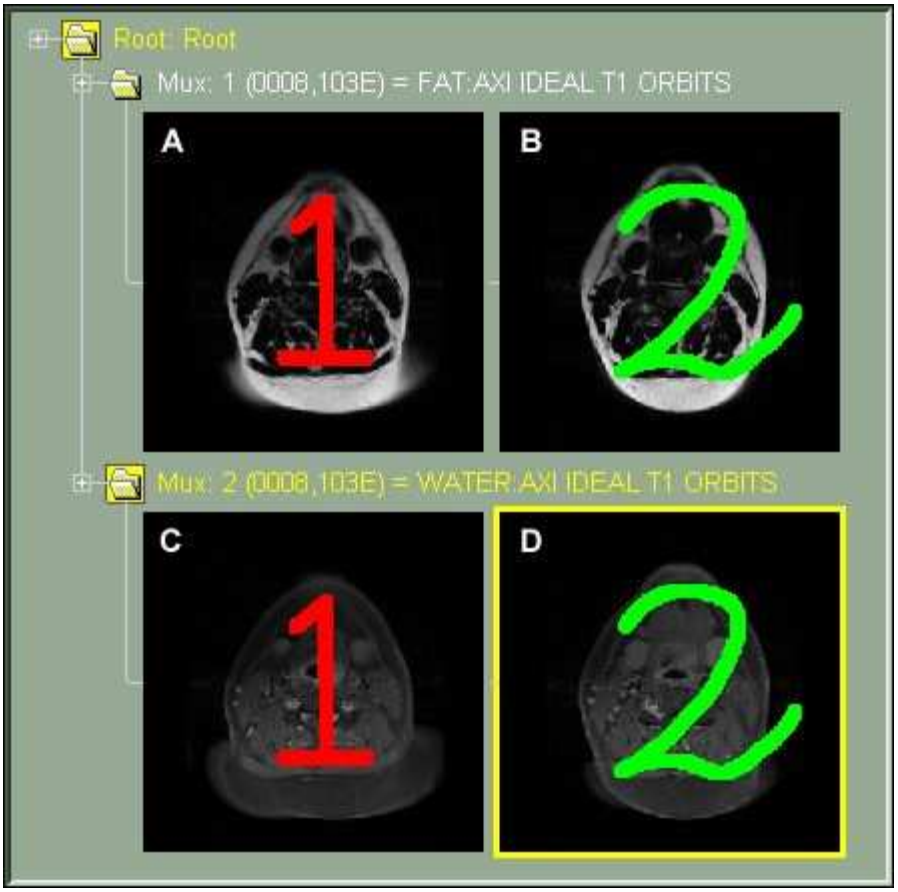
For example: We have 4 files, "A.dcm", "B.dcm", C.dcm" and "D.dcm", each containing 1 image. The first 2 files "A" and "B" have the parameter (008,103E) "Series Description" = "FAT" and the 2 last files "C" and "D" have "Water". File "A" and "C" are located at z=10.0 (the DICOM parameter (0020,1041) "Slice Location" = 10.0), and "B" and "D" have z=20.0.

File Names	(008,103E) "Series Description"	(0020,1041) "Slice Location"
A.dcm	FAT	10.0
B.dcm	FAT	20.0
C.dcm	Water	10.0
D.dcm	Water	20.0

If we only add the "Multiplex" to the default "Root" and "Frame" classes in the "DB Class Management" mode, then the database tree will look like this:



Frames A and C will share the same TAG file, as do frames B and D.



If the TAG are associated with the channel 1, then the TAG files will be "A.dcm.tag" and "B.dcm.tag".

From the Graphic Interface

Discriminate The value of this tag decides in which channel the frames from each file will go.

Associate The value of this tag will be used to match frames across channels.

TAG to channel This parameter decides the file name for the TAG files. Only 1 channel has actual TAG files, all the other channels are associated with these TAG files.

Update DB Recompute the database tree using the new parameters.

From the Display Area

Any modifications done to the TAG values of a frame will immediately be visible in all the associated frames.

From the Keyboard

There is no keyboard interface specific to this class.

From the Command Line

A list of the commands available while in this class can be found in the "SliceO_Class\Multiplex.dll" section of **Appendix B**.

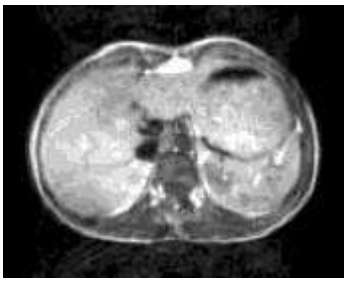
The 2D Classes:
Convolution Filter

This class modifies the content of the GLI frames. All the frames that "pass" through this class will be filtered using a convolution filter.

The available filters are:

No Filter
The image is not affected by the class

Median
For each pixel i,j in the original image, the resulting pixel will be computed by taking the median value of the pixel and its neighbors.



Mean
For each pixel i,j in the original image, the resulting pixel will be computed by taking the mean value of the pixel and its neighbors

Gaussian
The resulting image is produced by convolving the source image with a Gaussian kernel:

$$e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Laplacian of a Gaussian (LoG)
The resulting image is produced by convolving the source image with a kernel formed by the Laplacian of a Gaussian :

$$\nabla^2 e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Sharpen

The resulting image is produced by convolving the source image with the kernel of profile:

Laplace

The resulting image is produced by convolving the source image with the kernel:

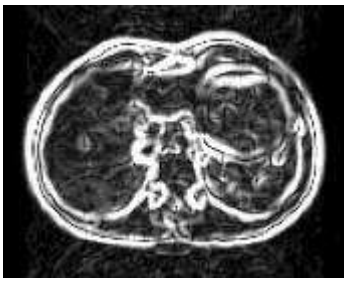
-1	-1	-1
-1	8	-1
-1	-1	-1

Roberts

The resulting image is produced by convolving the source image with the two kernels A and B and adding the results: res = ABS[A] + ABS[B]

A = | | |
|---|---|
| 0 | 1 |
| -1 | 0 |

B = | | |
|---|---|
| 1 | 0 |
| 0 | -1 |



Sobel

The resulting image is produced by convolving the source image with the two kernels A and B and adding the results: res = ABS[A] + ABS[B]

A = | | | |
|---|---|---|
| -1 | 0 | 1 |
| -1 | 0 | 1 |
| -1 | 0 | 1 |

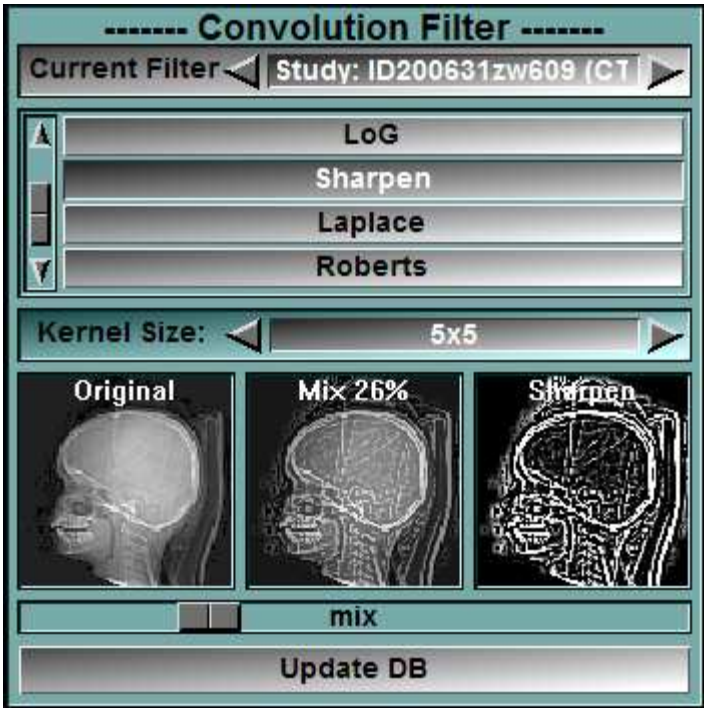
B = | | | |
|---|---|---|
| -1 | -1 | -1 |
| 0 | 0 | 0 |
| 1 | 1 | 1 |

The Median, Mean and Gaussian filters are used to reduce the noise in the images. The Mean and Gaussian filters will blur the edges while the Median filter will maintain clean edges.

The Sharpen and Laplace filters are contrast enhancement filters. These are useful when you would prefer to see anatomical structures more clearly.

The Roberts and Sobel filters are edge detection filters designed to increase the visibility of the edges in the images.

From the Graphic Interface



Instance selection Select which of the instances of the class you want to work on.

Filter list Clicking on one of the filter buttons will select the corresponding filter. The available filters are: Median, Mean, Gaussian, LoG, Sharpen, Laplace, Roberts and Sobel. A brief description of each filter, with some image examples, is located above.

Kernel Size You can adjust the Kernel Size of the Mean, Median, Gaussian and LoG filters.

Demo images Three demo images are provided to help in the filter and mix selection. The image on the left is unfiltered, the image on the right is fully filtered and the center image is a mix of these. The mix parameter is controlled by the mix slider.

Mix slider Filtering operations are controlled by the "mix" slider. The filtered image is mixed with the original image in a proportion that is controlled by the mix slider.
If the slider is completely to the left, the filtering operation will not affect the image.
If the slider is completely to the right, the image will be replaced by its filtered version.

Update DB Clicking on this button will apply this filter to the selected images.

From the Display Area

There is no display area interaction specific to this class.

From the Keyboard

There is no keyboard interface specific to this class.

From the Command Line

There is no command line for this class.

Technical Note

Contrast, Laplace, Roberts and Sobel filters are very sensitive to the noise in the image. You may want to precede these operations with a noise reducing filter (Median, Mean, Gaussian).

The 3D Classes

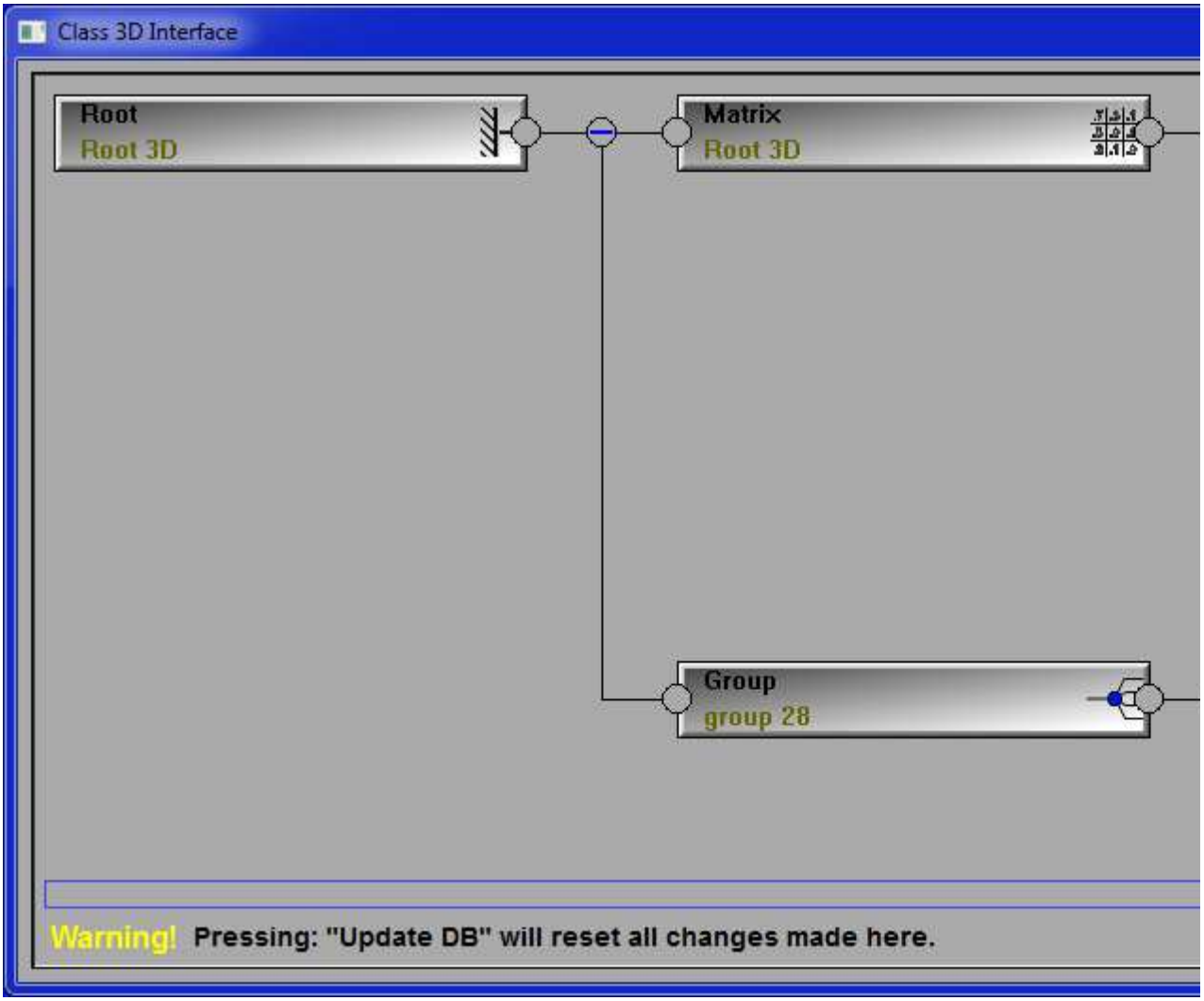
The 3D tree starts at the "Root" node, and finishes with "Geom" nodes. Each 3D geometry file (STL, OBJ...) that you read in sliceOmatic contains one or, sometimes, multiple geometries. Each of these is a "Geom". The root and the geom are both a "class" of nodes.

The root and the geom are always present in the 3D database tree. But in between, you can add other classes.

Note:

To add new classes to the 3D tree, you need to install the additional module: "DB Class management 3D".

The "Matrix" classes are used to selectively transform some (but not all) the geometries. The "Group" class can be used to make logical groups of multiple geometries.



Manual edition of the "Group" and "Matrix" classes in the "DB Cla

The 3D Classes:
Group

This class can be used to create logical group of geometries.

From the Graphic Interface

This class does not have a graphic interface.

From the Display area

There is no display area interaction specific to this class.

From the Keyboard

There is no keyboard interface specific to this class.

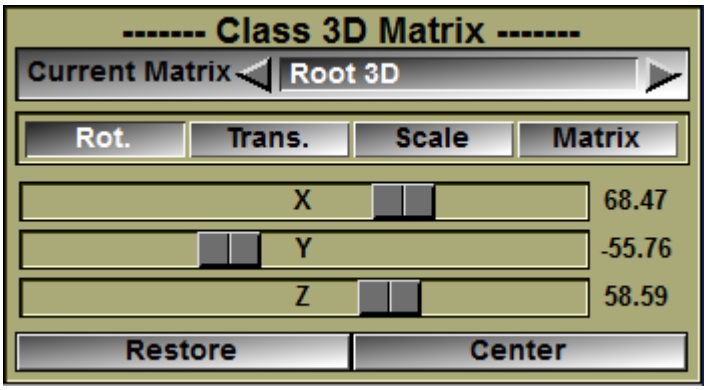
From the Command Line

There is no command line for this class.

The 3D Classes:
Matrix

This class can be used to assign a transformation matrix to specific geometries.

From the Graphic Interface



Instance selection Select which of the instances of the class you want to work on.

Rotation The X/Y/Z sliders control the surfaces' rotation around their axis. The rotations are applied in the following order: X then Y then Z.

Translation The X/Y/Z sliders control the surfaces' translation along their axis.

Scaling The X slider is used to control the surfaces' scaling factor.

Matrix Display the 4x4 numerical values of the matrix. These can not be directly edited.

X/Y/Z Sliders Control the rotation/translation or scaling applied to the transformation matrix.

Restore Reinitialize all transformation matrices and center all currently visible surfaces in the 3D window. (A surface is visible if its shading is either line, flat or smooth.)

Center Center the visible surfaces in the 3D window.

From the Display area

There is no display area interaction specific to this class.

From the Keyboard

There is no keyboard interface specific to this class.

From the Command Line

There is no command line for this class.

The 2D Modes

These modes either affect the database or are limited to the GLI images. When one of these modes is active the 3D and TAG tools will not be available. Any window associated with a TAG Mode will be in 2D mode (either Mode ONE or Mode ALL). The available modes are:

- **DB Class Management 2D**
  This mode enables you to structure the database tree with the classes you want.

- **DB File Management 2D**
  This mode enables you to remove files from the database, change the GLI/TAG file associations, add an offset to the slice positions and change the image's pixel dimensions (useful if you use image formats like JPEG that do not have pixel metrics).

- **2D Measurements**
  This mode is used to make measurements on the 2D frames. A number of measurement instruments are available (rulers, calipers, protractors, Region of Interest...).

From the Keyboard

There are a few keyboard keys that have been mapped to commands that can be used in all the 2D modules

Key	*Function*
Space	Toggle the display mode for the current 2D window between Mode ONE and Mode ALL
Page UP / Page Down	Change the current frame within its parent
Home / End	Change the current frame by changing the parent
PAD_PLUS **PAD_MINUS**	Increase / decrease scaling for all selected frames
Arrow keys	Move cursor 1 pixel in any direction
Print_Screen	Create a snapshot of the display area

The 2D Modes:
DB Class Management 2D

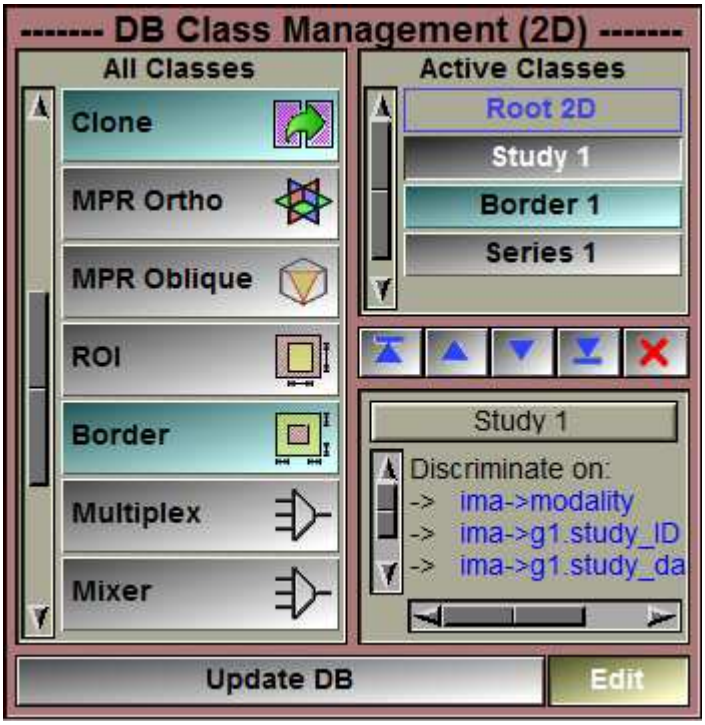
In this mode you can change the composition of the database tree. There are 3 database trees. One each for 1D, 2D and 3D objects. The 1D objects are spectroscopy data, the 2D are images and the 3D are polygon geometries.

This mode enable you to manipulate the composition of the 2D database tree by selecting the 2D classes that will be used in the generation of the tree.

In the "All Classes" list, you have buttons for the available classes that you can use to compose the database tree. Classes from sliceOmatic basic package have a grey button, while classes imported from the "Additional Modules" section of the Web download pages have a cyan button.

For example, the default 2D tree is composed of the Root, Study, Series and Frame classes. When you read an image in the program, it will automatically be inserted in the tree. If the image is part of a new study, then a study class will be created and inserted as a child of the root. If the image is also part of a new series, then a series class will be created and inserted as a child of the study class. And finally a frame class will be created for the image and inserted as a child of the series class. Pressing the "Edit" button will show you the complete tree as it exist in the program's memory.

From the Graphic Interface



All Classes list

This list will present all the 2D classes loaded by the program. The buttons fo have a cyan background. The available classes from the basic package are:

This button will insert a "Study" class in the tree. This cla usually placed right under the root of the tree. Also see "Stu

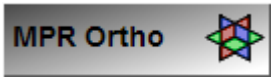
This button will insert a "Series" class in the tree. This cl
usually placed right after the study class in the tree. Also se

This button will insert a "DICOM Tree" class in the tree. Yo
instance will be used to discriminate the frames with a diff
branch for each value of the tag (0010,0020) "Patient ID",
"Study ID", the third will use the tag (0020,0011) "Series Nu
Number". Also see "DICOM Tree" in The Classes section.

Note:

This class will only work if the files are DICOM. Using
"DICOM Tree" class is redundant.

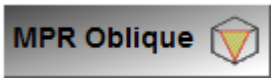
This class will insert an "Unsorted" class in the tress. This
are inserted in the order they where read in the progra
"Unsorted" in the class section.



Each instance of this class will have 3 children, one is the
created by re-slicing the volume in orthogonal directions. Als

Note:

The orthogonal plane children will only be created if the cl
same dimensions.)



Each instance of this class will have 2 children, one is the u
by re-slicing the volume in an oblique direction. Also see "M

Note:

The oblique plane children will only be created if the clas
same dimensions.)

Each instance of this class will enable you to select a Reg
class section.

Note:

This class only makes sense if all the children of each in
different image resolutions in your dataset, use a class hig
or "DICOM Tree").

This class enables you to associate different GLI frames to
representations of the same slice. For example, some M
Also see "Multiplex" in the class section.

This class will insert a "Mixer" class in the tree. This class i
Also see "Mixer" in the class section.

This class will apply a convolution filter to all its children. Als

Active Classes list You can click on one of the classes of this list to select it. The selected class can

Note:

The "root" and "Frame" classes are always present in the tree and cannot be m

Top/Up/Down/Bottom These buttons are used to change the order of the classes in the tree. The sele the top (just after the root class), up one level, down one level or to the bottom (jus

Delete Pressing this button will delete the selected class from the "Active Classes" list.

Information Text This window displays the information about the selected class in the "Active Clas

Update DB The database tree is recomputed using the classes defined in the "Active Classe

Edit The "Edit" button open a class edition window. In this window you can see the co can remove classes and change their connections.

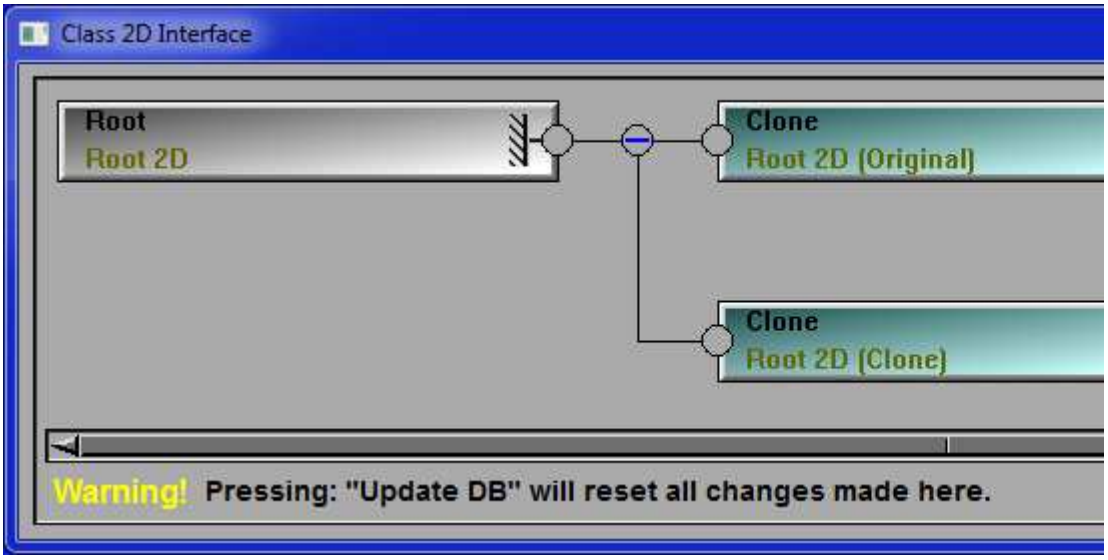
Warning:

Changing classes connections can be problematic in some cases. For exa unpredictable consequences. Make sure you have a good idea of what you ar

Warning:

Pressing the "Update DB" option will re-generate the database based on window will be lost.

The Edit Interface



In this window you can see the complete class database tree as it is in the progra

From this window you can remove classes from the tree. Pressing the "Delet instance from the tree. The children of that class (if there are any) will automatica

You can also detach and re-attach classes by selecting their "in" circles (the one circle (on the right side) of another class. The re-attach children will be added to children. So this technique can also be used to re-sort the children of a class. of all the children of the class.

From the Display Area

There is no display area interaction specific to this mode.

From the Keyboard

There is no keyboard interface specific to this mode.

From the Command Line

There are no command lines specific to this mode.

The 2D Modes:
DB File Management 2D

In this mode you can close files, change the default position of frames, assign new pixel dimensions for image formats that do not have this information, and associate TAG files to the original GLI files.

From the Graphic Interface

Sub-Modes

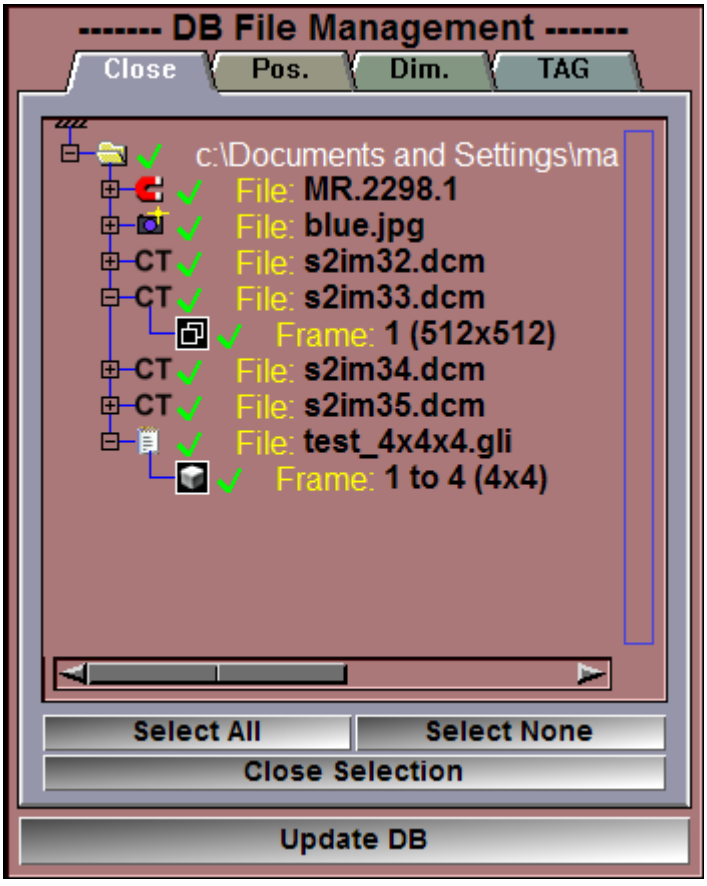
The File Management mode has 4 sub modes: **Close**, **Position**, **Dimensions** and **TAG files**.

Close Tab

In this sub-mode you can close files you do not want to use anymore.

Pressing the "Delete" key while the cursor is over an item will automatically mark it as close. The item will then be removed from the database tree, but the file is still in memory. You can still bring it back to the database tree by re-selecting the file and updating the database.

The files will only be removed from memory when they are marked for closing and you click on the "Update DB" button.



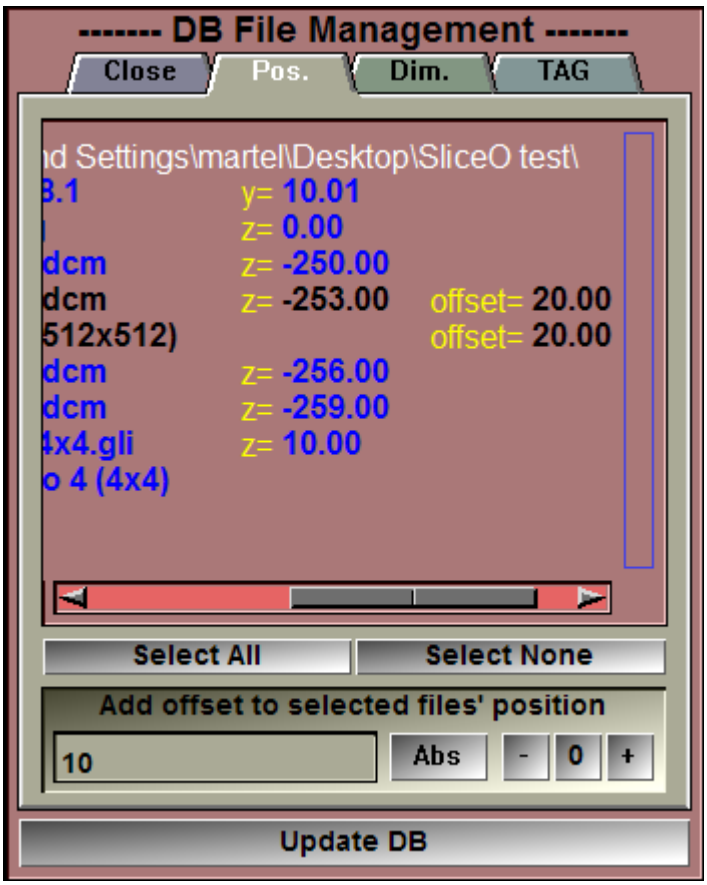
Select All

Select all the files

Select None	De-select all the files
Close Selection	Close all the selected files

Pos. Tab

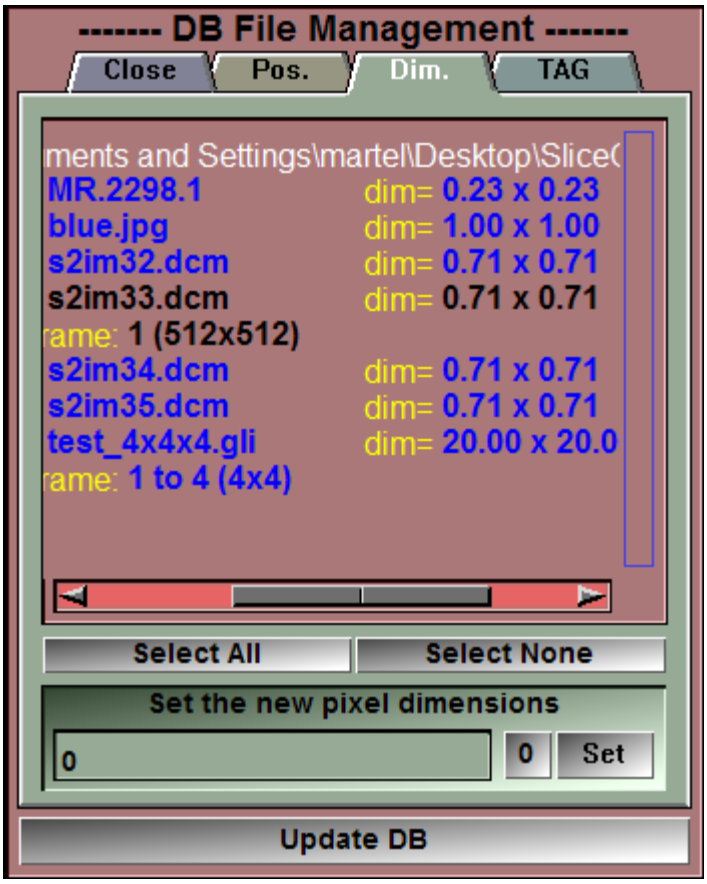
In this sub-mode you can change the "depth" of a frame.



Select All	Select all the files
Select None	De-select all the files
Position value	Enter a numerical value to be used as offset for the frames
Abs or Inc	Clicking this button will toggle the mode between "Abs" and "Inc". In "Abs" mode, the position offset value will be added "as is" to all selected files. In "Inc" mode, the offset value is incremented before being added to each selected files. For example, with 3 selected files and a value of "10", for the "Abs" mode, clicking "+" will produce offsets of "10", "10" and "10". in The "Inc" mode the offsets will be "10", "20" and "30".
+ (Add)	Add the value to the current position for all selected frames
0 (Reset)	Reset the position of the frames to their initial values
- (Subtract)	Subtract the value from the current position for all selected frames

Dim Tab

In this sub-mode you can change the pixel dimensions of a frame (both horizontal and vertical dimensions will be equal though).



Select All Select all the files

Select None De-select all the files

Dimension Enter a new value for the pixel's dimension. This value will be applied to both horizontal and vertical dimensions.

0 (Reset) Reset the dimensions of the pixels to their initial values

Set Set the pixel (horiz and vert) dimensions of the selected frames to the specified value.

TAG file

In this sub-mode you can specify the TAG file associated with each GLI files.

TAG Path If set to "ON", you can specify the path of the associated TAG files for all selected files. By default, when "OFF" the TAG files are in the same directory as the GLI files, but with this option you can place the TAG files in a different directory.

TAG Suffix If set to "ON" you can add a suffix value to the TAG file's names. This is useful if you want to have different TAG files for the same GLI images (ex: you do a multi user study, each user can have his own suffix).

Update DB The database tree is recomputed using the new file information.

From the Display Area

You can change the file's selection by selecting classes and frames from the display area. Pressing the "Delete" key while the cursor is over a frame will remove the frame from the database tree and mark the associated file for deletion.

From the Keyboard

The following keyboard key can be used while in this module:

Key	*Function*

Delete	If the cursor is over a frame in the display area or an item in the file list, then pressing the "Delete" key will remove the frame from the database tree and mark the associated file for deletion.

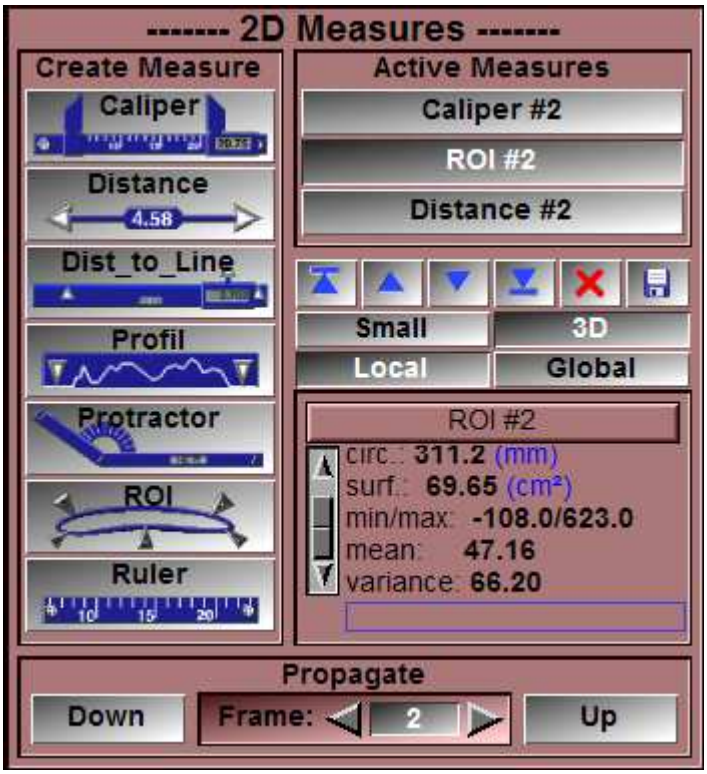
From the Command Line

There are no command lines specific to this mode.

The 2D Modes:
2D Measurements

In this mode, you can create and manipulate 2D measurement instruments.

From the Graphic Interface



Create Measure list Clicking on one of the tools buttons will put the corresponding instrument at the b of the current image.

The available instruments are:

This is another instrument to measure distances on the image

This instrument gives the distance between 2 points.

This instrument measures the distance between a point a line.

The Profile instrument gives a curve of the pixel values alon edge.

This instrument gives you the angle between 3 points.

This instrument enables you to compute some data insi
Region Of Interest (ROI). The ROI is formed by a Cardinal sp
You can move the control points simply by dragging them wit
mouse. If you want more precision, you can add new co
points by clicking on the ROI's perimeter. You can also remo
control point by pressing the "Delete" key while the cursor is
it.

The computed values are:

Nb Pixels:	The number of pixels in the region
Min/max:	The minimum and maximum values of the GLI p in the ROI
Mean:	The mean value of the GLI pixels in the ROI
Variance:	The variance of the GLI pixel values inside the I
Circonf.:	The length of the ROI perimeter
Surface:	The surface covered by the ROI.

This instrument is a simple ruler, graduated in the units o
image.

Active Measure list

Selecting an instrument from this single selection list will change the color of
instrument to red and its information will be displayed in the **Information box.**

Top/Up/Down/Bottom

The instruments are stacked in the order of their creation. You can change that o
with these buttons. The order is important: if you have the control points of mu
instruments overlap one another. The control point that will be selected for drag
belongs to the instrument on top.

Delete

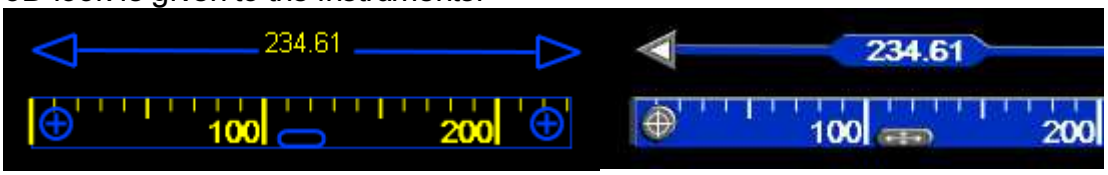
This button will cause the instrument selected by the Active Measure list to
deleted. If no instrument is selected, this button will be disabled.

Save

This button will write to a file the values associated with the instrument selecte
the Active Measure list. If no instrument is selected, this button will be disabled.
syntax used is the same as the "Save to script" command below.

Small / 3D

This setting affects the appearance of all instruments. In the "Small" mode, a si
line drawing is used to represent the instruments. In the 3D mode, a more deta
3D look is given to the instruments.



Small 3D

Local / Global

This setting will be applied to future instruments. In the local setting, when a
instrument is created, it will only be active in the current image. In the "Gl
setting, when a new instrument is created, it will be active in all the images.

Information Text

The numerical values from the selected instrument will be displayed here.

Propagate All the measurement tools present on the starting frame will be copied on the ta frame. This tool is also described in the "Interface tools" section of the introductio

From the Display Area

When the tools are created, they will be at the base of the current frame.

The tools can be directly manipulated in the Display Area by clicking and dragging one of their control points. There are two kinds of control points:

The anchor point

The anchor points icons are either an arrow or a circle with a cros Moving these points will drag the part of the tool that is connected to i

The drag point

The drag point icon is an oval with arrows. Dragging this point v move the complete tool.

Mouse button	*Function*
Left **Right** **Middle**	Drag a measurement tool control point

From the Keyboard

The following keyboard key can be used while in this module:

Key	*Function*
Delete	If the cursor is over an instrument's control point, then pressing the "Delete" key will delete the instrument.

From the Command Line

A list of the commands available while in this mode can be found in the "SliceO_2D\Measure.dll" section of appendix B.

The TAG Modes

These modes are used to segment the images and create the TAG data. Each segmentation technique has its advantages and its inconvenients. When one of these modes is active the 3D tools will not be available. Any window associated with a TAG Mode will be in 2D mode (either Mode ONE or Mode ALL). The available modes are:

- **Edit**
  This is a simple "paint box" that enables to edit the pixels of the TAG images. It is useful to re-touch any small errors that can be left after having used the other segmentation modes.

- **Geometrical Masks**
  The geometrical masks are useful as post processing. You can use the masks to increase the number of tags present in an image. If, for example, you want to differentiate between left and right organs, you can segment the organs using the same TAG values for left and right and after that create a rectangle mask over the right part of the body and "add" the TAG value associate with the mask to the TAG image to create a different TAG value for the right hand side organs.

- **Morpho**
  The Mathematical Morphology mode is very useful for segmenting some type of images. Any tissue that has a big enough surface and a well defined gradient will be easy to segment with this Module. The main advantage of Morpho is that the segmentation technique is based on the variation of the pixel values (the gradients), not the values themselves. So this makes it ideal for MR images where the pixel values are not uniform over the image. A good example would be sub-cutaneous fat in MR images. It is not very good for segmenting very small regions however, so it would not work well with the visceral fat. If you work with a 3D dataset, Morpho propagates very well from one frame to the next.

- **Region Growing**
  The Region Growing Mode is based on thresholding. So it works directly with the pixel values. It has 3 sub-modes: "Paint", "Grow 2D" and "Grow 3D". "Paint" is very useful when you want to threshold only a small region in the image. For example, it is the ideal tool to segment visceral fat. Just select a big brush, fix a threshold that will capture the fat (you can adjust threshold values with the mouse wheel) and "paint" your fat in the abdominal cavity with the brush. "Grow 2D" grow a region starting from the cursor position. All adjacent pixels that fall within the threshold range, and respect the brush constraint, are tagged. "Grow 3D" is similar to the 2D version, but also propagate through all the frames that belong to the same parent in the database tree. Unfortunately, even though the option is present, the Region Growing mode does not propagate well from one frame to the next.

- **Snakes**
  Snakes, also called "Active Contours", work well with tissues that have a smooth well defined contour such as bones in CT. It is also the ideal mode to compute the circumference of the body. Just place a few points outside the body and minimize the curve. You can then transform the snake in a geometry and get the length of the curve from the 3D measurement tool. If you work with a 3D dataset, the Snakes propagate very well from one frame to the next.

- **Thresholding**
  The Thresholding Mode applies a global threshold to the complete image. Apart from a few cases in CT images, global thresholding is not the ideal segmentation tool. Since it is global, it tends to tag tissues that should not have been tagged if you work with images containing multiple organs. For simple images, such as CT of the members, it can be very useful. The threshold are based directly on the pixel values, so in CT you can fix your threshold directly on the Houdsfield values. However, for more complex images with multiple organs, you have more control with the Region Growing tool.

From the Keyboard

There are a few keyboard keys that have been mapped to commands that can be used in all the TAG modules.

Key	*Function*
F1 to **F4**	Select display modes "Grey", "Mixed", "Over" or "TAG"
"**+**" / "**-**"	Increase / decrease TAG opacity in Display Mode "Mixed"
F5 to **F10**	Select brush **#**1 to **#**6
"**0**" to "**9**" and "**Pad 0**" to "**Pad 9**"	Set the current TAG value to TAG-0 through TAG-9
Pad Period and **Pad Delete**	Select the TAG value under the cursor as the current TAG value
Enter	Flood-fill the region under the cursor with the current TAG value
Page UP / Page Down	Change the current frame within its parent
Home / End	Change the current frame by changing the parent
Pad Plus / Pad Minus	Increase / decrease scaling for all selected frames
Arrow keys	Move cursor 1 pixel in any direction
Print Screen	Create a snapshot of the display area

Technical Note:

The "Flood-fill" operation: All the pixels adjacent to the pixel under the cursor and having the same TAG value will be filled by the current TAG value.

From the Command Line

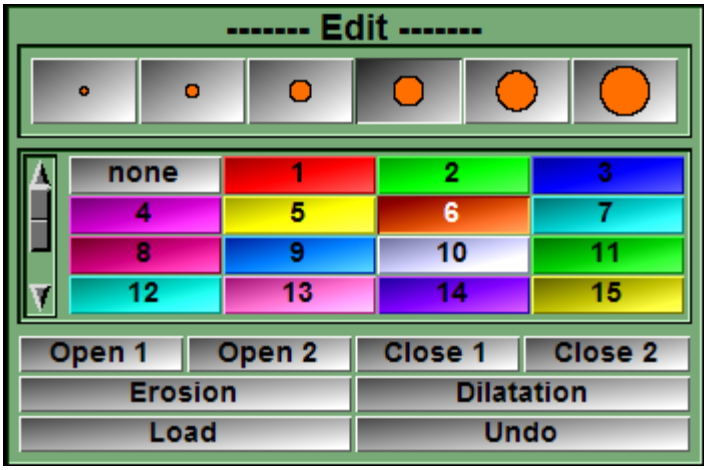
A number of commands affect the tags. These can be found in:
"SliceO_System\Brush.dll" section of **Appendix B**.
"SliceO_System\Tag.dll" section of **Appendix B**.
"SliceO_System\Undo.dll" section of **Appendix B**.

The TAG Modes:
Edit

In this mode you can edit the TAG images.

To edit the image you must choose a current Tag value and a brush. Then you use the mouse to paint the current Tag value on the image in the Display Area. The first modification you do on an image will automatically load the image in a "save" buffer which is used when you "erase".

From the Graphic Interface



Brush — Select the brush used to paint in the Display Area. This tool is also described in the "Interface tools" section of the introduction

TAG Value — Select the current Tag value. This tool is also described in the "Interface tools" section of the introduction

Open 1 / Open 2 — This operation is useful for reducing noise in the image. Any small region (2 pixels across for open 1 and 4 pixels across for open 2) will be erased

Close 1 / Close 2 — This operation is useful for eliminating small holes in the image. Any small regions (2 pixels across for close 1 and 4 pixels across for close 2) will be filled.

Erosion — Shrink the TAG image one pixel in all directions.

Dilatation — Expand the TAG image one pixel in all directions.

Load — Manually load the image in the restore buffer. See Undo/Redo in the **Main Menu** section for more information on the Load and Erase functions

Undo — Undo last brush stroke. It is a shortcut to the “Undo” button of the Uno/Redo Menu. It can only undo “edit” operations.

From the Display Area

The mouse controls associated with this mode are:

Mouse button	*Function*
Left	Apply paint.
Right	Erase your modifications.
Middle	Increase the brush to a big square, allowing to paint or erase a block at a time.

From the Keyboard

There is no keyboard interface specific to this mode.

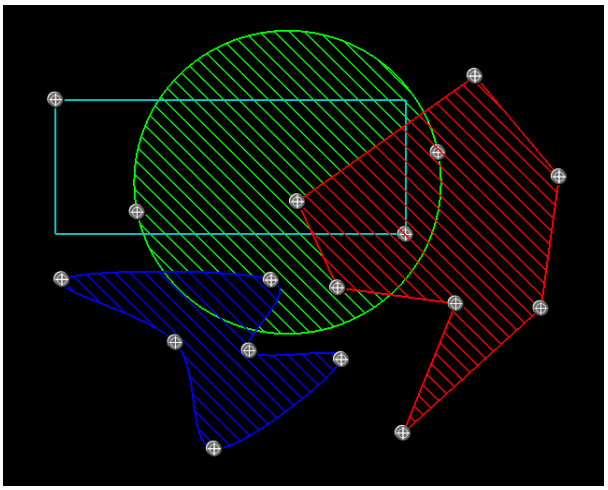
From the Command Line

There are no command lines specific to this mode.

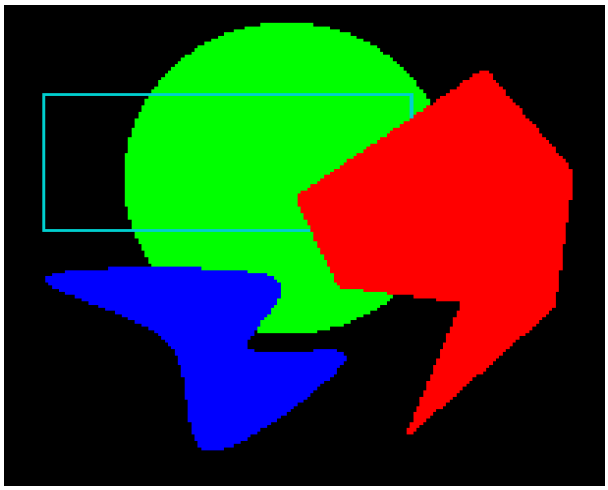
The TAG Modes: Geometrical Masks

In this mode you can stamp geometrical shapes in the TAG images.

Each of these shapes has an associated TAG value. You can "Stamp" these shapes in the TAG image, either replacing the TAG value under them with the value associated with the shape, or adding (or subtracting) their associated value to the ones already present in the TAG image.

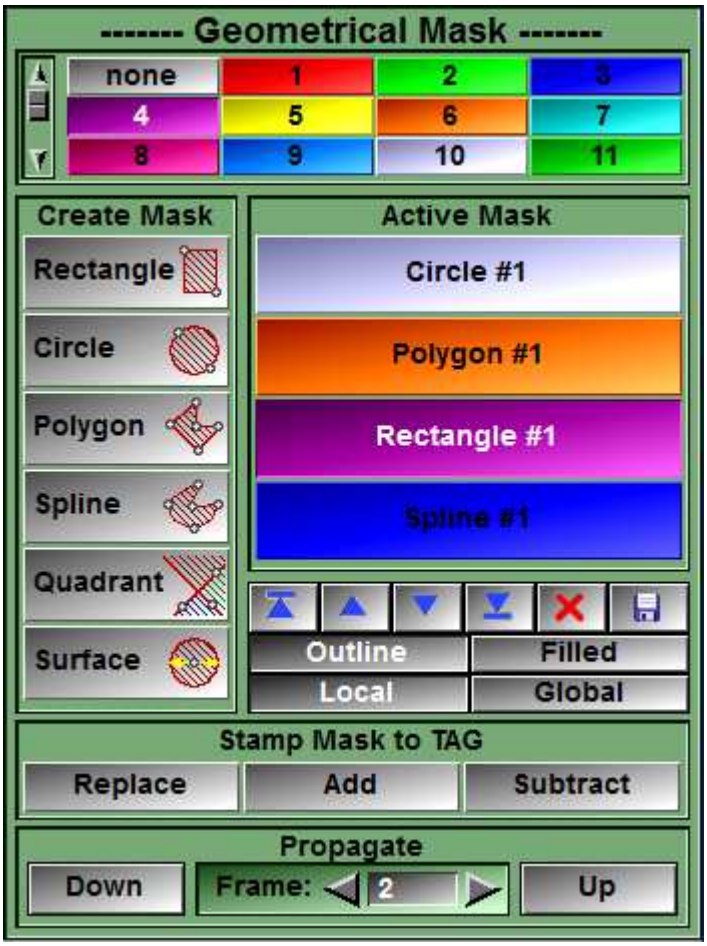


Geometrical Masks



Result of the "Stamp: Replace" operation

From the Graphic Interface



TAG Value

Each mask has a tag value associated with it. You can select this tag valu appropriate button before creating the mask, or you can select a mask from the click on the tag value's button you want to associate with it. This tool is a "Interface tools" section of the introduction

Create Mask List

Select the shape of the mask you want to create. Clicking on one of the buttons the mask in the current image. You can then edit this mask with the mouse.

You have a choice of 5 basic mask shapes:

The "rectangle" shape is defined by its 2 corner points.

The "circle" shape is defined by 2 points on the circumference.

The "polygon" shape is defined by its vertex. You can add a ver clicking in an edge of the shape. You can remove a vertex by p the cursor over it and pressing the "del" key .

The “spline” used is a Catmull-Rom spline. It has the characteris going through every one of its control points. You can add c points by clicking on the border of the shape at the location whe want to insert the new point. You can remove a control point by placing the curso it and pressing the “del” key .

The “Quadrant” split the image in 4 quadrants. These are defin controlled by 2 points, and another line, perpendicular to the fi through a third point. The upper left corner of the image always r first quadrant. In the “filled” mode, the first quadrant will be of the mask’s ta second quadrant of tag+1, etc.

The "Surface" shape has a fixed radius. The radius can be chang command "mask: radius value" where "value" is the new radius.

Active Mask List All the masks that are active for the current image will be displayed in this list. these masks by clicking on its button. A number of operations can be done on th

 **Delete** Delete the selected mask

 **Save** Write the information necessary to re-create the selected mask in a script file.

Top/Up/Down/Bottom The masks are stacked in the order of their creation. You can change that order with these buttons. The order is important: if you stamp the mask to the TAG images with the Replace option, the mask on top will overwrite all other values.

Outline/Filled The mask can be either just an outline or a filled shape. An outline shape will be an uninterrupted 1 pixel wide line around the shape.

Local/Global The mask can be present either only on the image where it was created or on group where it was created.

Stamp Mask to TAG Use the geometrical shape of the masks to “stamp” the tag values. There are stamping:

Replace The tag values under the mask are replaced by the tag value assoc

Add The tag values associated with the masks are added to the ta image.

Subtract The tag values associated with the masks are subtracted from the image.

The tag values are limited to the range 0 to TAG_MAX. The Add and Subtr cause the tag values to go beyond this range.

Propagate Copy all local masks from one slice to the next. This operation will not affect glo are already present on all slices. This tool is also described in the "Interface introduction

From the Display Area

When a new mask is created, it will be placed in its default position on the current image. You can edit this shape by dragging its control points.

You can also add control points to the polygon and spline shape by clicking on the border of the shape at the location where you want to insert the new point. You can remove a control point by placing the cursor over it and pressing the “del” key .

From the Keyboard

The following keyboard key can be used while in this module:

Key	*Function*
Delete	If the cursor is over an instrument's control point, this pressing the "Delete" key will delete the instrument.

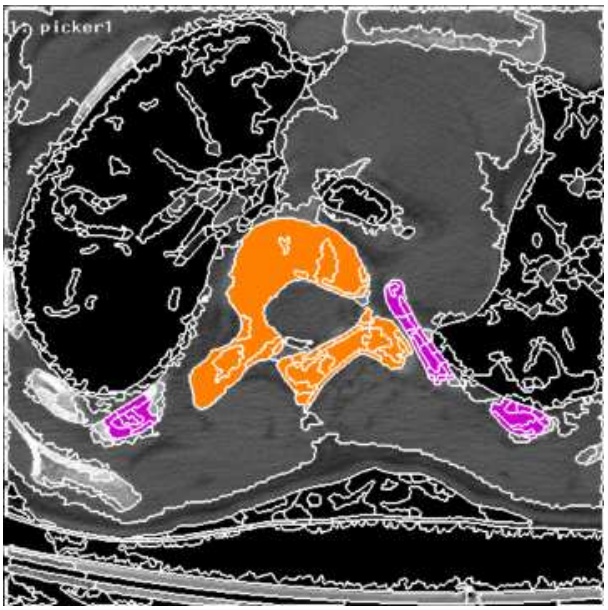
From the command line

A list of the commands available while in this mode can be found in the "SliceO_TAG\Mask.dll" section of appendix B.

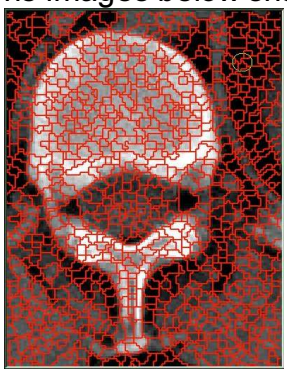
The TAG Modes:
Morpho

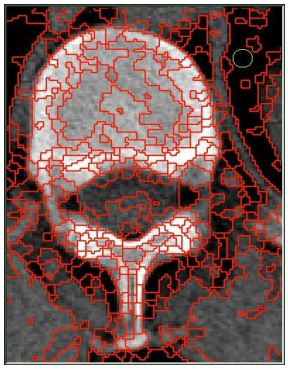
In this mode you can use Mathematical Morphology to segment and edit the TAG images

Mathematical Morphology segmentation is done by computing the Watershed of the gradient of the image. This will give you a kind of mosaic of the image. Each region of this mosaic can then be filled with the appropriate Tag value. Each region should correspond to no more than one tissue type, filling these regions will be faster than editing the image one pixel at a time.

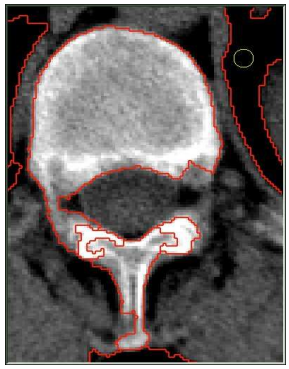


Computing the Watershed yields a great number of regions; we then proceed to merge these regions together. SliceOmatic enables you to compute four (4) different merges. The images below show the effect of the four defaults merges.


Merge 1

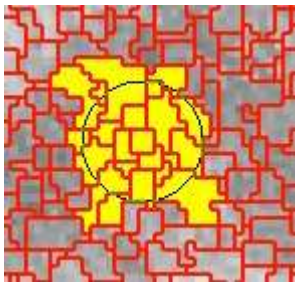

Merge 2


Merge 3


Merge 4

The first step to perform is the computation of the **Watershed** for a GLI image. This can be done by pressing one of the 4 Merge buttons, or the *Compute All* button. Once this is done, the **Water Parting** mesh will be superimposed on the image. The Water Parting is the set of lines that define the different Watersheds. The color and thickness of these lines can be modified with the **Hue** slider and the Line Thickness button (see below). You can then **Flood-Fill** the Watershed with a first approximation of the segmentation obtained through a threshold (you may want to do one or two erosions of the threshold image before doing the flood-fill). This step will fill every pixel of the TAG Image

corresponding to a region of the watershed with the highest Tag value of these pixels.

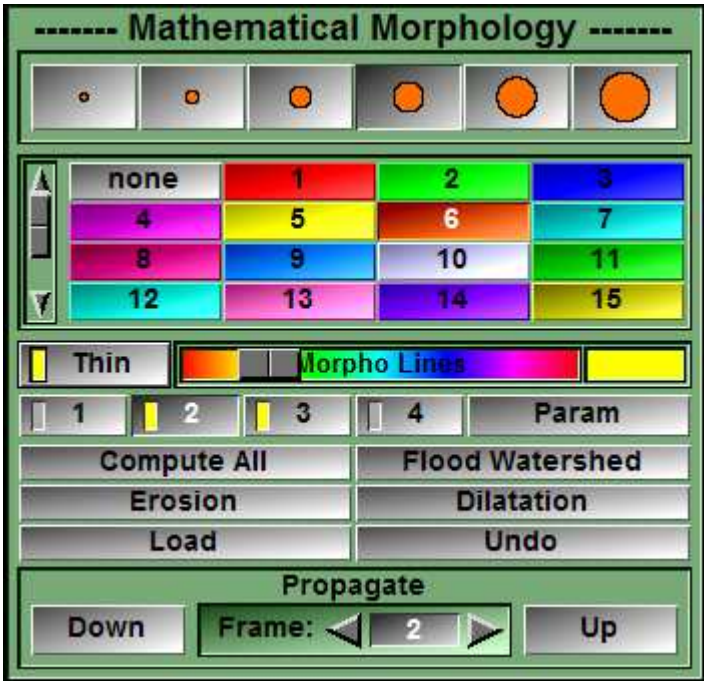


You can then fill individual watershed regions with the mouse. The editing of the TAG image in the **Morpho** mode is similar to the **Edit** mode. You use the left mouse button to add the current Tag value, and the right mouse button to bring back the Tag value from the save buffer. The difference is that instead of modifying only the pixels under the brush, you change the Tag value of all the pixels in the regions touched by the brush.

Pressing the middle mouse button will increase the brush to a big square, allowing you to paint or erase a block at a time.

The first modification you do to an image will automatically load the image in a "save" buffer which is used when you "erase".

From the Graphic Interface

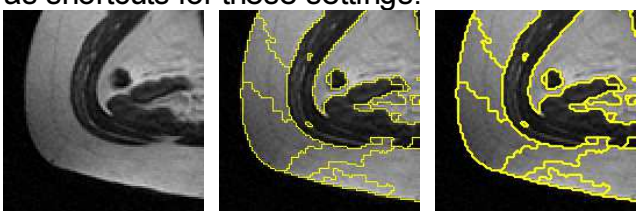


Brush Select the brush used to paint in the Display Area. This tool is also described in the "Interface tools" section of the introduction

TAG Value Select the current Tag value. This tool is also described in the "Interface tools" section of the introduction

Line Thickness

Change the Water parting lines' thickness. This button will sequentially take the values "Off", "Thin" and "Thick". The keys "a", "s" and "d" can also be used as shortcuts for these settings.



Hue slider

Change the Water parting lines' color to make them more visible. An example of the current color is showed in the box at the right of the slider.

1 to 4

SliceOmatic enables you to compute 4 different merges. The merging of regions is controlled by 2 parameters: the surface of each region (the program will attempt to merge regions with a surface smaller than the threshold value) and the difference between regions' mean values (the program will merge regions together only if this difference is smaller than the threshold value). By default, the 4 Merge buttons are assigned increasing merge parameter values. These values can be modified with the **Param** button.

These 4 buttons are used to select the associated Watershed merge. If this merge has not yet been computed or if the parameters of this Merge button have been changed, the Watershed merge is computed.These buttons can also be activated with the keys "q" , "w", "e" and "r" for 1 to 4 respectively. Depending on the Display Mode selected, the program will compute the Watershed of the current image (Display Mode One), or of all the selected images (Display Mode All). The yellow indicator on each of the merge buttons indicates if this merge exists for the current image.

Param

Display a new area in the menu (see below). These new buttons enable you to modify the merge parameters (surface and mean difference) associated with the 4 merge buttons. Surface is expressed in pixels and the mean difference is expressed in % of the image's dynamic range.

1	2	3	4	Param
0.20	1.00	2.50	7.50	mean diff.
5	25	40	50	surface

SliceOmatic will attempt to merge with its neighbor all Watershed regions smaller than "surface". The merge will be done if the difference between the mean value of the pixels in the 2 regions is smaller than the "mean diff." value.

Compute All

Compute the Watershed of all the selected GLI image(s) for all their merge values.

Flood Watershed

The highest Tag value inside each region of the Watershed is used to fill the region. This function is usually used in conjunction with the Threshold segmentation as a first step in the Morphological segmentation: compute a segmentation with the Thresholds, then flood the regions with these results.

Erosion

Shrink the TAG image one pixel in all directions.

Dilatation	Expand the TAG image one pixel in all directions.
Load	Manually load the image in the restore buffer. See Undo/Redo in the **Main Menu** section for more information on the Load and Erase functions
Undo	Undo last brush stroke. It is a shortcut to the “Undo” button of the Uno/Redo Menu. It can only undo “Morpho” operations.
Propagate	Use the segmentation of the image selected by “slice” as a seed for the segmentation of an adjacent image. The propagation can be done either on the image immediately preceding (“Up” button) or following (“Down” button) the selected image. This tool is also described in the "Interface tools" section of the introduction

From the Display Area

The mouse controls associated with this mode are:

Mouse button	*Function*
Left	Flood all the regions under the cursor.
Right	Restore the “erase” color in all the regions under the cursor.
Middle	Increase the brush to a big square, allowing to paint or erase a block at a time.

From the Keyboard

In addition to the keyboard shortcuts seen in section 5.2.0.3, the following keyboard keys , Specific to Morpho, can also be used as a shortcut:

Key map	*Action*
“**q**”, “**w**”, “**e**”, “**r**”	Select Merge 1 to 4
“**a**”,“**s**”, “**d**”	Set the Water parting lines visibility and thickness

From the Command Line

A list of the commands available while in this mode can be found in the "SliceO_TAG\Morpho.dll" section of appendix B.

Technical Note:

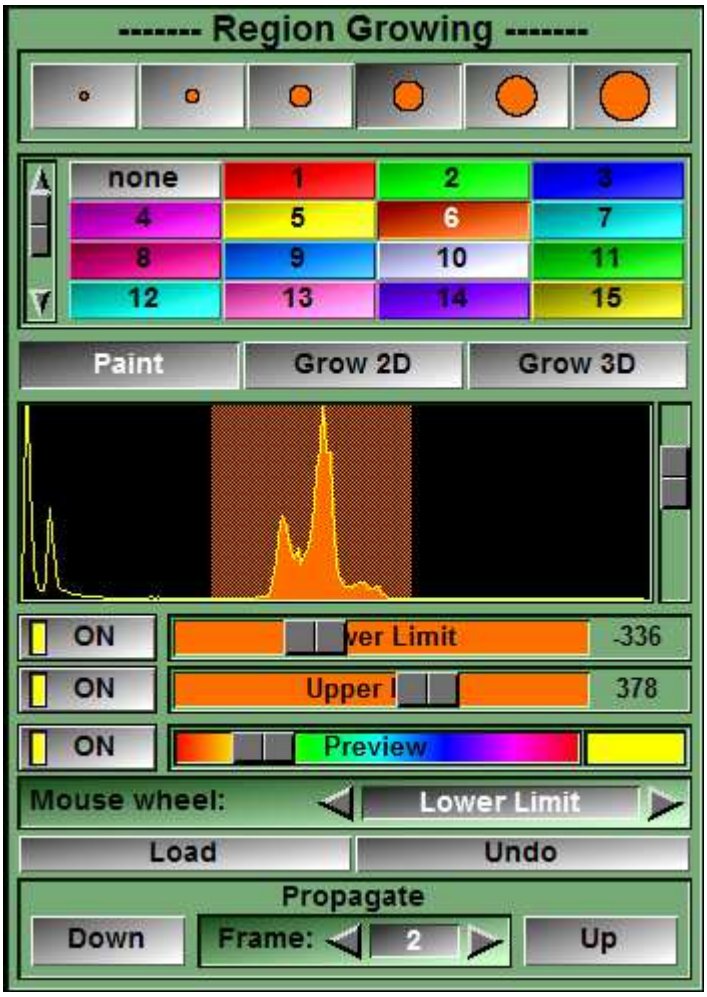
For more information on the Mathematical Morphology operators (erosion, dilatation, opening, closing), please refer to the article: *A New Set of Fast Algorithms for Mathematical Morphology* I and II, Andre Bleau, Jacques De Guise, and A.-Robert LeBlanc. CVGIP: Image Understanding, Vol. 56, No. 2, September 1992, pp. 178-229.

The TAG Modes:
Region Growing

In this mode, you can threshold a specific region of the image.

You must first select a threshold range with the Upper and Lower limit tools. You can then precisely Tag all the pixels that fall within this range with a paint brush (Paint mode) or grow a 2D or 3D region using the pixels under the brush as a seed.

From the Graphic Interface



Brush Select the brush used to paint in the Display Area. This tool is also described in the "Interface tools" section of the introduction

TAG Value Select the current Tag value. This tool is also described in the "Interface tools" section of the introduction

Paint button A left click on the mouse will activate the brush. All the pixels under the brush whose GLI values are within the threshold range will be tagged. The right mouse button is used to “erase” to modification and bring back the Tag values from the last “load” operation.

Grow 2D button

A left click on the mouse will seed a region under the cursor. In order to grow, the region must meet the following criteria:

- The GLI values of the pixels must be within the threshold range.
- The region will not grow through a region smaller than the radius of the brush.
- Depending on the value of the $REGION_OVERWRITE variable, The region will not overwrite a a pixel whose TAG value is:
    - ($REGION_OVERWRITE=4) Smaller than the current Tag value.
    - ($REGION_OVERWRITE=5) Smaller or equal than the current Tag value.
    - ($REGION_OVERWRITE=1) equal to the current Tag value.
    - ($REGION_OVERWRITE=3) Superior or equal to the current Tag value (Default value).
    - ($REGION_OVERWRITE=2) Superiorl to the current Tag value.

The right mouse button can be used to erase the latest grow operation.

Grow 3D button

A left click on the mouse will seed a 3D region under the cursor. The flood will propagate through all the frames that belong to the same parent. In order to grow, the region must meet the following criteria:

- The GLI values of the pixels must be within the threshold range.
- The region will not grow through a region smaller than the radius of the brush.
- Depending on the value of the $REGION_OVERWRITE variable, The region will not overwrite a a pixel whose TAG value is:
    - ($REGION_OVERWRITE=4) Smaller than the current Tag value.
    - ($REGION_OVERWRITE=5) Smaller or equal than the current Tag value.
    - ($REGION_OVERWRITE=1) equal to the current Tag value.
    - ($REGION_OVERWRITE=3) Superior or equal to the current Tag value (Default value).
    - ($REGION_OVERWRITE=2) Superiorl to the current Tag value.

The right mouse button can be used to erase the latest grow operation.

Histogram box

Displays the histogram of the grey values of the GLI image(s). The vertical axis represents the number of pixels having a particular value, while the horizontal axis represents the possible values of the pixels (from Minimum at left to Maximum at right). The threshold range fixed by the lower and upper limits will be superimposed on this histogram. When the cursor is over an image, the GLI value of the pixel under the cursor is indicated by a white vertical bar on the histogram. This tool is also described in the "Interface tools" section of the introduction

Lower limit button

Activates and control the lower limit slider.
“Off”: the upper limit is disabled.
"On": the Upper limit slider directly fixes the minimum GLI value that a pixel can have in order to be within the threshold range for painting or growing.

Lower limit slider

Give the value that is used to form the lower limit of the threshold range. This slider can be locked in position by pressing the “Scroll Lock” key while the cursor is over it.

Upper limit button Activates and control the upper limit slider.
“Off”: the upper limit is disabled.
"On": the Upper limit slider directly fixes the maximum GLI value that a pixel can have in order to be within the threshold range for painting or growing.

Upper limit slider Gives the value that is used to form the upper limit of the threshold range. This slider can be locked in position by pressing the “Scroll Lock” key while the cursor is over it.

Preview If ON, a preview of the segmentation will be displayed in Paint and Grow 2D modes.

Preview color This slider enables you to change the preview’s color to make it more visible. An example of the current color is showed in the box at the right of the slider.
Note:
The preview is unavailable in the Grow_3D sub-mode

Mouse wheel You can use the mouse wheel to change the threshold values. You can select which of the thresholds will be affected by the mouse wheel, or disable the mouse wheel control.

Load Manually load the image in the restore buffer. See Undo/Redo in the **Main Menu** section for more information on the Load and Erase functions

Undo Undo last brush stroke. It is a shortcut to the “Undo” button of the Uno/Redo Menu. It can only undo “Region Growing” operations.

Propagate Propagate the Region Growing to an adjacent frame. This tool is also described in the "Interface tools" section of the introduction

From the Display Area

The mouse controls associated with this mode are:

For the “Paint” sub-mode

Mouse button	***Function***
Left	Threshold the regions under the cursor.
Right	Restore the tag values in all the regions under the cursor.
Middle	Increase the brush to a big square, allowing to threshold or restore a larger surface.
Wheel	Increase/decrease the select threshold value

For the “Grow 2D” and “Grow 3D” sub-mode

Mouse button	***Function***
Left	Grow a region starting at the current cursor position.
Right	Grow a region starting at the current cursor position.

Middle	Increase the brush to a big square
Wheel	Increase/decrease the select threshold value

From the Keyboard

There is no keyboard interface specific to this mode.

From the Command Line

A list of the commands available while in this mode can be found in the "SliceO_TAG\Region.dll" section of appendix B.

The TAG Modes:
Snakes

In this mode you can use Active Contour (Snake) curves to segment the TAG images.

A Snake is a curve that will tend to minimize its energy. The energy of a Snake is composed of 2 parts: its internal energy, computed from its shape and curvature, and its external energy, computed from the gradient of the image under the curve.

In this mode you will be able to create Snakes either by placing the Snake's points directly on the image, or by extracting the contour of existing Tag data to create new Snakes.

From the Graphic Interface

Brush Select the brush used in the Display Area. The brush size gives the minimum radius of the Snake which in turn controls the "smoothness" of the curve. A bigger brush will create a smoother curve, a smaller brush will let the snake "hug" details more closely. This is a local property of the Snake. You can use different brush sizes to create or edit different portions of the Snake. This tool is also described in the "Interface tools" section of the introduction

TAG Value Select the current Tag value. The "Delete" Tag is used to delete all the Snakes under the cursor. When editing Snakes, only the Snakes associated with this Tag will be affected. This tool is also described in the "Interface tools" section of the introduction

Create New Snake Create a new Snake. The creation mode will remain effective as long as you don't minimize the Snake's energy.

Snake From TAG Create Snakes from the contours of every Tag surface in every selected image.

Polarity The Snake expects either:

- a light object on a dark background (+)
- a dark object on a light background (-)
- the Snake will adapt locally to the highest gradient disregarding its orientation (No Polarity).

Minimize Energy Minimize the energy of all the Snakes associated with the current Tag of all the selected images.

Snake To TAG Fill out the contours of the Snakes with their Tag colors.

Snake To GEOM Convert the Snake's curves to geometries (closed polylines) that can be visualized in the 3D modes and exported to any of the 3D formats supported by sliceOmatic.

Save to Script Create a script file that can be used to re-create the Snakes of the current Tag.

Propagate Uses the Snakes of the image selected by "slice" as a starting point for the Snakes of an adjacent image. The propagation can be done either on the image immediately preceding ("Up" button) or following ("Down" button) the selected image. Only the Snakes of the currently selected color will be propagated. This tool is also described in the "Interface tools" section of the introduction

From the Display Area

Creating a new Snake

To create a new Snake, you must select a Tag color and a brush. You then press the "Create" button and place points on the image with the left mouse button. The Snake points must form a first approximation of the contour of the object. **The order of these points must be clockwise around the object**. If you are delimiting a hole in the object, the points must be counter-clockwise. You can move existing points of the new Snake by clicking and dragging the point. You can add a new point between existing points by clicking on the segment joining the points. Otherwise, the points are added at the end of the Snake. The radius of each new point is the radius of the brush. This radius will determine the local radius of curvature of the Snake.

Minimizing the Snake's Energy

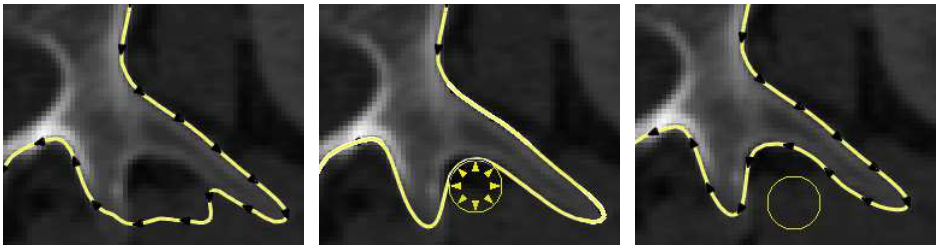
You then click on the "Minimize Snake" button to compute the Snake. The Snake will tend to latch to the regions of maximum gradient in the image that are in the neighborhood of the original curve, while preserving the conditions of local curvature. The middle mouse button can also be used to minimize the energy.

In "Create New Snake" mode

Mouse button	*Function*
Left	Insert (or drag) a control point.
Right	Erase your modifications.
middle	Minimize the current snake energy.

Editing the Snake

At anytime, you can edit the Snake's curve. The left mouse button is used to push the curve away from the cursor, while the right mouse button will attract the curve toward the center of the cursor. All the points on the curve touched by the cursor, either by pushing or pulling, will also have their radius attribute changed to the current brush radius. You may want to decrease the radius of a section of the curve to hug details more closely, or inversely, you can locally increase the radius to smooth out a section of the curve.

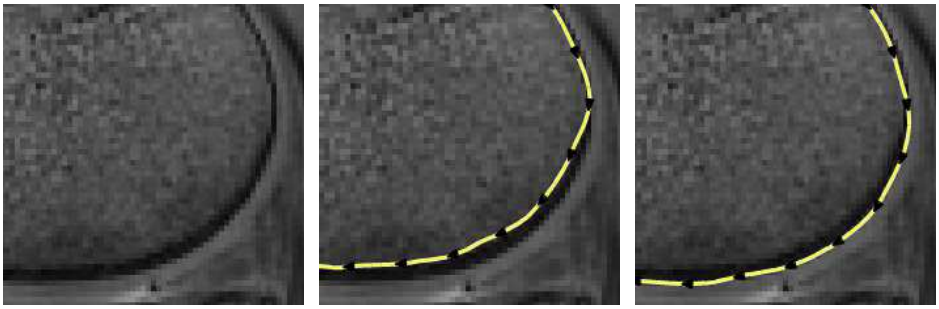


Not in "Create New Snake" mode

Mouse button	*Function*
Left	Edit the curve by pushing the points.
Right	Edit the curve by dragging the points toward the brush's center.
middle	Minimize the current snake energy.

Following the correct gradient

When the object you want to Tag is composed of higher value pixels than the surrounding background, the gradient of the pixels on the border of the object will point away from the Snake curve. SliceOmatic uses this information to help the Snake latch to the correct gradient. If, however, the intensities of the pixel are reversed (dark objects on a light background), you must tell it to the program. This is done with the "Polarity" buttons.



Deleting Snakes

You can delete a curve by pressing the “Delete” key while the cursor is over the curve, or you can select the “Delete” button and delete the curve by pressing the left mouse key while the cursor is over the curve.

From the Keyboard

The following keyboard key can be used while in this module:

Key Map	*Action*
Delete	Delete the Snake under the cursor.

From the Command Line

A list of the commands available while in this mode can be found in the "SliceO_TAG\Snake.dll" section of appendix B.

Technical Note:

When minimizing the Snake's energy, the following steps are used:

The Snake curve is subdivided in small segments about 3 pixel longs. Each segment is assigned a "radius" value. This radius is derived from the brush size used either to create the original points of the Snake, or to edit the Snake.

The internal energy of the Snake is then computed from the local curvature of the segments. This curvature is weighted with the "radius" value of the segments.

The external energy is computed from the gradient of the image under each segment, weighted by the angle between the segment and the gradient's orientation. To compute the correct angle, it is important that:

- The correct gradient polarity is selected.
- The Snakes around the organs are created clockwise, and the snakes around holes in the organs are created counter-clockwise.

When minimizing its energy, the snake will move to achieve the smoothest curve while "hugging" the best gradients on the image. The distance the segments can move to achieve this is controlled by the "capture range" of the segment. This value is derived from:

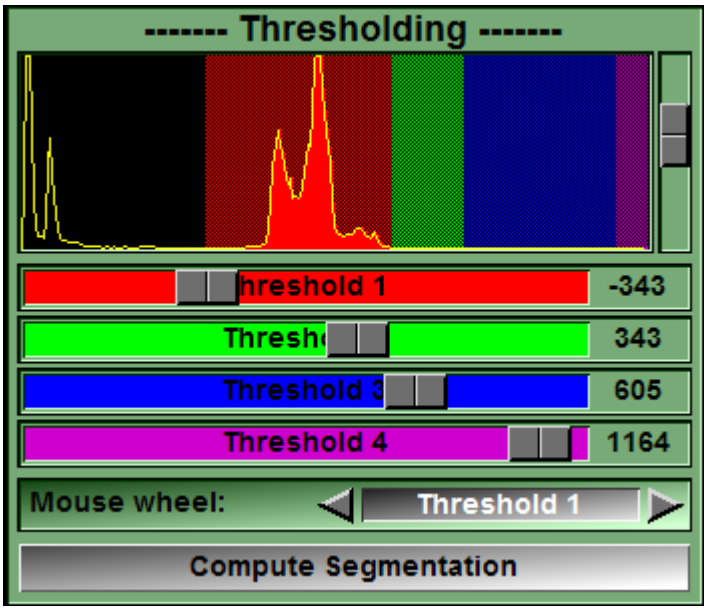
- The segment's local radius.
- The curve's polarity. A "no polarity" value will half the capture range.
- The curve's age. A curve segment that has been manually edited is probably close to where the user wants it, so its capture range is decreased.

The TAG Modes: Thresholding

In this mode, you create segmented TAG images from the GLI Images (Grey Level Images).

The segmentation is done by a simple grey level thresholding on the GLI images with four different thresholds.

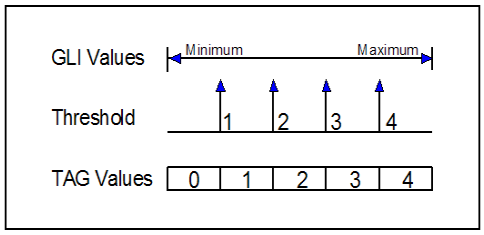
From the Graphic Interface



Histogram box Display the histogram of the grey values of the GLI image(s). The vertical axis represents the number of pixels having a particular value, while the horizontal axis represents the possible values of the pixels (from Minimum at left to Maximum at right). The colors under the line correspond to the segmentation values fixed by the sliders. When the cursor is over an image, the GLI value of the pixel under the cursor can be seen as a vertical bar on the histogram. This tool is also described in the "Interface tools" section of the introduction

Threshold slider

The 4 sliders are used to segment the GLI image by its values. The sliders range from the Minimum to the Maximum values of the GLI pixels. The sliders can be locked in position by pressing the “Scroll Lock” key while the cursor is over it.



- All GLI pixels with a value smaller than the first slider will have a Tag value of 0.
- All GLI pixels with a value in between the first slider and the second slider will have a Tag value of 1. Idem for Tag values 2 and 3.
- All GLI pixels with a value higher than the fourth slider will have a Tag value of 4.

Note:

The TAG values associated with each threshold slider (by default, 1 to 4) can be changed either with the “Up” or “Down” arrow keys when the cursor is over the threshold slider, or with the command line (see the “segment: threshold” command).

Mouse Wheel

You can use the mouse wheel to change the threshold values. You can select which of the threshold will be affected by the mouse wheel, or disable the mouse wheel control.

Preview

Toggle the preview mode. When on, the indicator on the button light up and the TAG displayed in the 2D windows will reflect the threshold selection.

Compute

Pressing the **Compute** button will perform the real segmentation and create the TAG images.

From the Display Area

The GLI value of the pixel under the cursor is highlighted in the interface’s histogram.

Mouse button	***Function***
Wheel	Can increase/decrease the threshold values

From the Keyboard

In addition to the keyboard shortcuts seen in section 5.2.0.3, the following keyboard keys , Specific to Thresholding, can also be used as a shortcut:

Key map	***Action (While the cursor is over one of the Threshold sliders)***

"Scroll Lock"	Lock or Unlock the slider under the cursor.
“Up Arrow"	Decrease the TAG value associated with the slider under the cursor
“Down Arrow”	Decrease the TAG value associated with the slider under the cursor

From the Command Line

A list of the commands available while in this mode can be found in the "SliceO_TAG\Threshold.dll" section of appendix B.

The 3D Modes

These modes only display and manipulate 3D objects. When one of these modes is active the 2D and TAG tools will not be available. Any window associated with a TAG Mode will be in 3D mode. The available modes are:

- **Display 3D**
  This Mode only displays the 3D objects. It has no other functions.

- **Contour Lines**
  This mode creates 3D contour lines around the TAG pixels in all the selected frames. The line can be composed of line segments or just points. Unlike the Shell modes, this mode does not impose any constraint on the frames it can use.

- **Shell TomoVision / Lorensen**
  These modes are 2 variations of the same thing: They are used to create a polygonal shell around the TAG voxels. The Lorensen algorithm will produce a a more detailed geometry with a LOT more polygons. If you need to decrease the number of polygons, you should use TomoVision's algorithm. Both algorithms will only be able to create a geometry from the frames of a single parent. And these frames must be aligned, parallel and all have the same image resolution and pixel metrics.

The mouse controls

The mouse is used to rotate, translate and scale the displayed objects. The mouse can be in one of three modes of controls: The Track Ball, the Classic or the Patient modes. These are explained in more detail in the "Transform" section of **The 3D Tools**.

From the Keyboard

There are a few keyboard keys that have been mapped to commands that can be used in all the 3D modules

Key	*Function*
Print_Screen	Create a snapshot of the display area

Also, if the "3D Slices" tool is enabled, then the following keys can be used to manipulate the slices:

Key	*Function*
Page UP / Page Down	Change current frame within its parent
Home / End	Change the current frame by changing the parent
F1 to **F4**	Select display modes "Grey", "Mixed", "Over" or "TAG"
"**+**" / "**-**"	Increase / decrease TAG opacity in Display Mode "Mixed".

From the Command Line

A number of commands affect the tags. These can be found in: "SliceO_System\3D.dll" section of **Appendix B**.

The 3D Modes:
Display 3D

This mode can be used to view 3D geometries and/or the frames in 3D using the 3D slice tool.

From the Graphic Interface

There is no interaction with the interface

From the Display Area

There is no display area interaction specific to this mode.

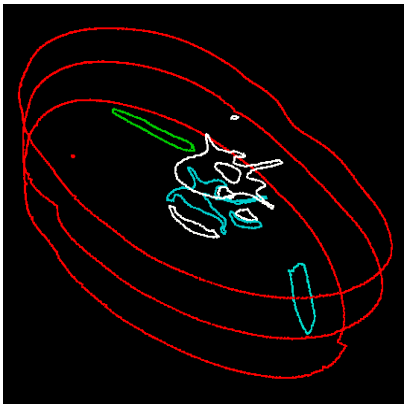
From the Keyboard

There is no keyboard interface specific to this mode.

From the Command Line

There are no command lines specific to this mode.

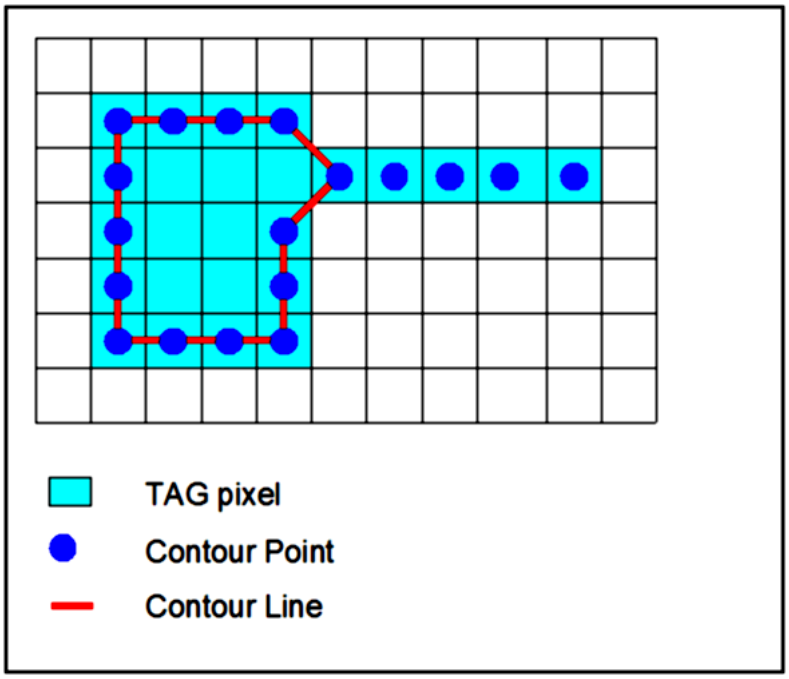
The 3D Modes:
Contour Lines



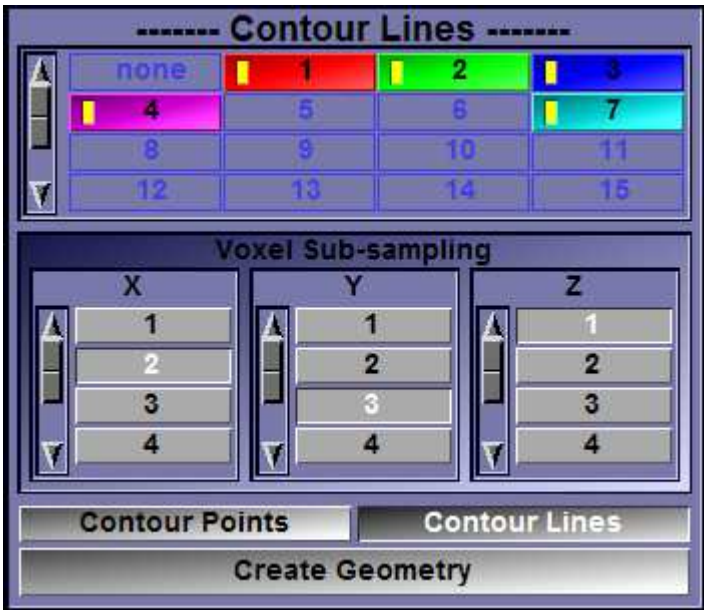
In this mode, you can visualize and save the contours of the Tag regions of the dataset volume. These contours can be either closed contour lines or just the points on these lines.

The line segments are ordered so that the Tag pixels of that segment are always on the right-hand side of the segment.

In the Line mode, if a Tag region is too thin, and the contour degenerates to a line (2 line segments share the same points), the degenerated segment will be eliminated. Thus, some points may appear in the Point mode but not in the Line mode. (See example below)



From the Graphic Interface



TAG List Select which Tag is going to be used to create the 3D surfaces. The yellow indicator on the Tag's button is lit if the Tag is selected.

Sub-sampling The number of points on the contours is controlled by the sub-sampling, but the polygons' contours will use ALL the voxels.

Contour Points Only the points on the contours of the objects will be displayed and saved.

Contour Lines The contours of the Tag regions will be displayed and saved as closed lines.

Create Geometry A geometrical object will be created from the contours (either points or lines). These objects can be visualized in the other sliceOmatic modes, or saved to a geometry file (See the File Menu section of **The Main Menu** and the Shading section of **The 3D Tools**).

From the Display Area

There is no Display Area interaction specific to this mode.

From the Keyboard

There is no keyboard interface specific to this mode.

From the Command Line

A list of the commands available while in this mode can be found in the "SliceO_TAG\Contour.dll" section of appendix B.

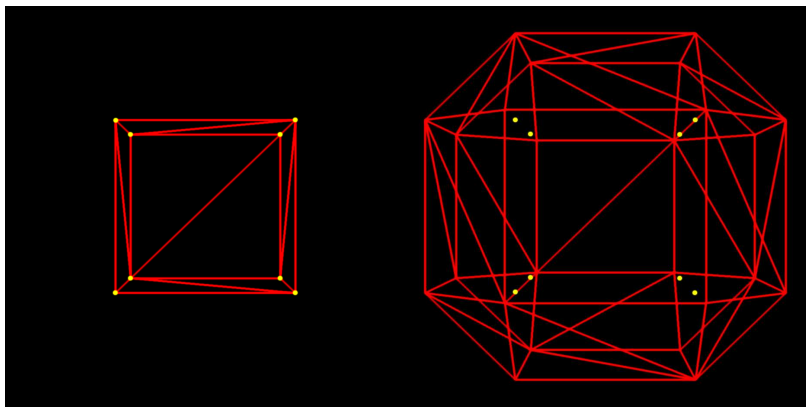
The 3D Modes:
Shell TomoVision

In this mode, you create 3D polygonal surfaces from the TAG images you created with the TAG Module.

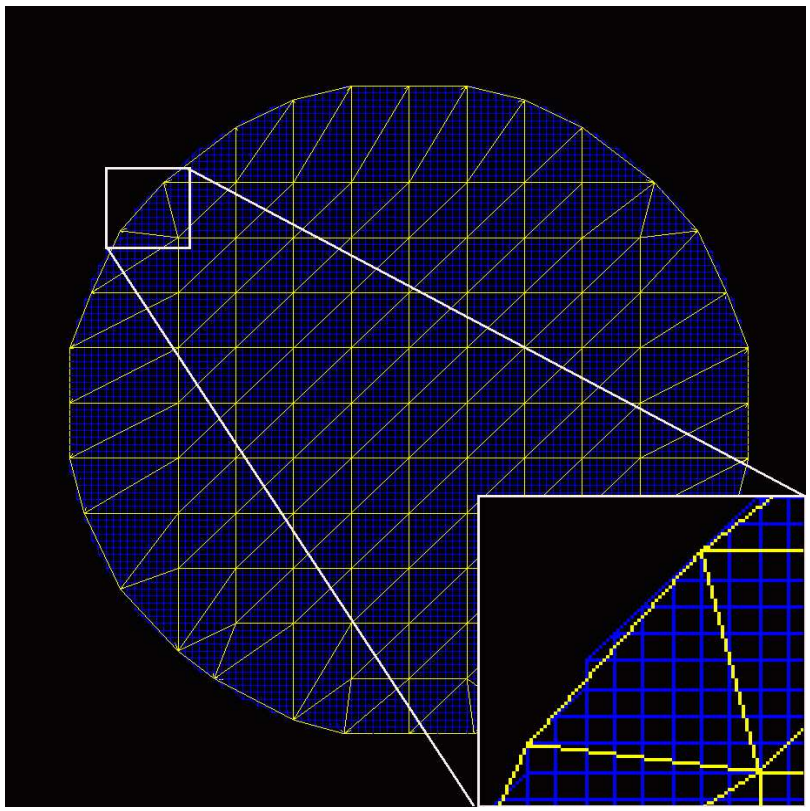
In TomoVision's algorithm, the center of each voxel is considered as a potential node for the reconstructed surface. If this node lies on the border between 2 tag values, it will be used to anchor the surface's polygons. It is also possible to use a sub-sampling of the voxels to limit the number of polygons. When using a sub-sampling, even though the creation of the geometry uses one in N voxels, a subsequent optimization phase replaces the polygons nodes close to the tag borders using all the voxels values.

The following images show the reconstruction of a 8 nodes cube using

a) the TomoVision algorithm.
b) the Lorensen algorithm and



Using sub-sampling for the x, y or z directions will decrease the number of polygons created. One point in N will be considered. Once the polygons are created, the geometry will be expanded using all the image's voxels. The polygons' nodes will be moved on the actual border between the tags, regardless of the sub-sampling value. So even though sub-sampling reduces the polygon count, it does not affect too much the surface's contour.

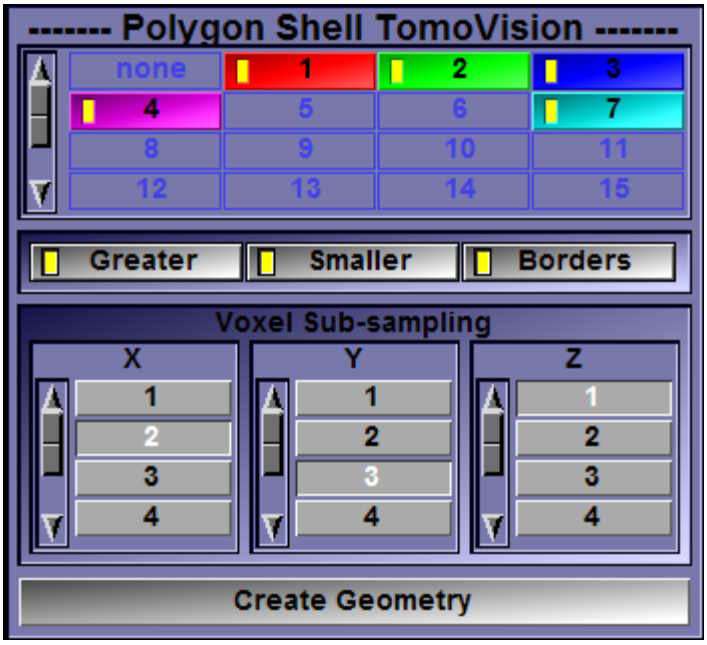


This example shows 2 Meshes created with the TomoVision algorithm: Using sub-sampling=1 (in blue, 14524 polygons) and sub-sampling=8 (in yellow, 444 polygons)

Note:

The shell computation is only available for 3D and sorted parallel groups.

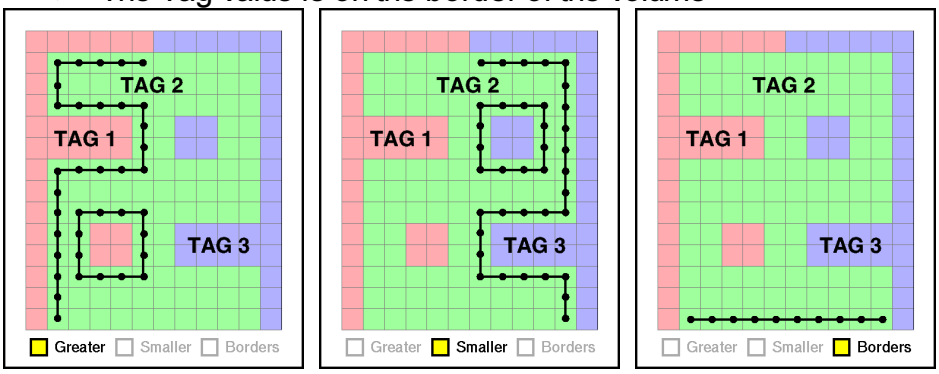
From the Graphic Interface



TAG Value Select the current Tag value

Conditions These buttons control the surface creation conditions. For each selected Tag value, a surface will be created on the border of a Tag pixel if:

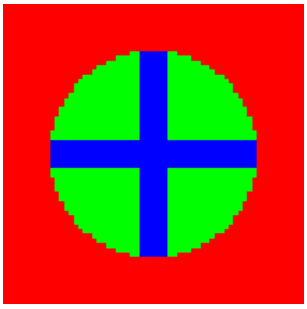
- The Tag value is greater than the Tag value of its neighbor
- The Tag value is smaller than the Tag value of its neighbor
- The Tag value is on the border of the volume



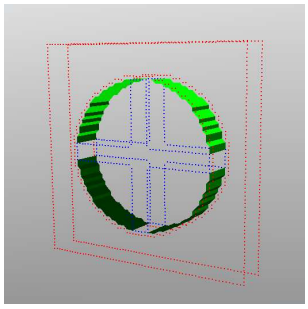
Voxel Sub-sampling The number of polygons created is controlled by the sub-sampling, but the polygons' contour will use ALL the voxels

Create Geometry Clicking this button will cause sliceOmatic to compute the 3D shell around all the selected Tags

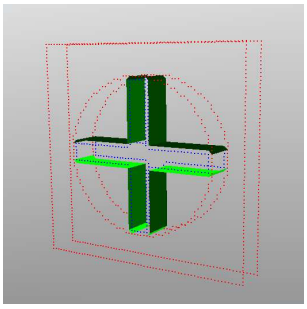
Example of reconstructions with different conditions.



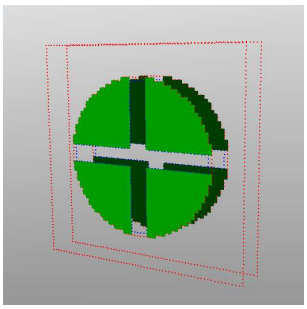
The **original TAG image**:
Outside = TAG 1,
circle = TAG 2,
cross = TAG 3



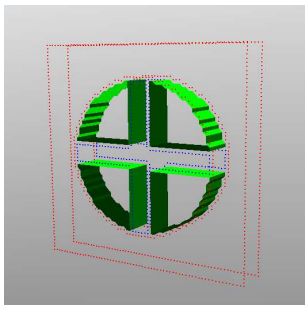
Shell of TAG 2 with the condition
Greater



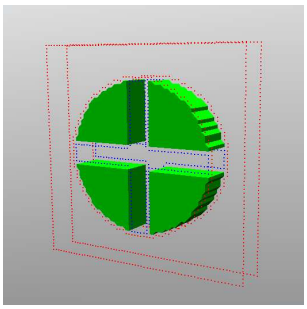
Shell of TAG 2 with the condition
Smaller



Shell of TAG 2 with the condition
Border



Shell of TAG 2 with the condition
Greater & Smaller



Shell of TAG 2 with the condition
Greater & Smaller & Border

From the Display Area

There is no Display Area interaction specific to this mode.

From the Keyboard

There is no keyboard interface specific to this mode.

From the Command Line

A list of the commands available while in this mode can be found in the "SliceO_TAG\Shell....dll" section of appendix B.

The 3D Modes:
Shell Lorensen

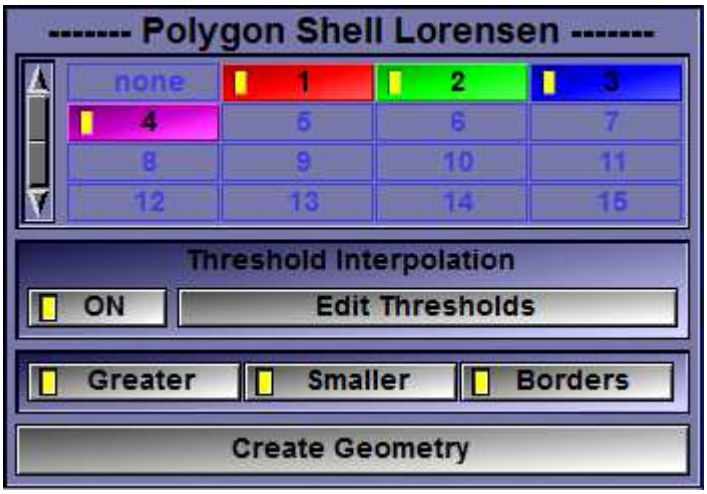
In this mode, you create 3D polygonal surfaces from the Tag images you created with the TAG Module.

In Lorensen's algorithm, the surface will be half a pixel wider than the center of the tag voxels. The surface will be slightly larger and contain more polygons than the surface created in the TomoVision mode.

Note:

The shell computation is only available for 3D and sorted parallel groups.

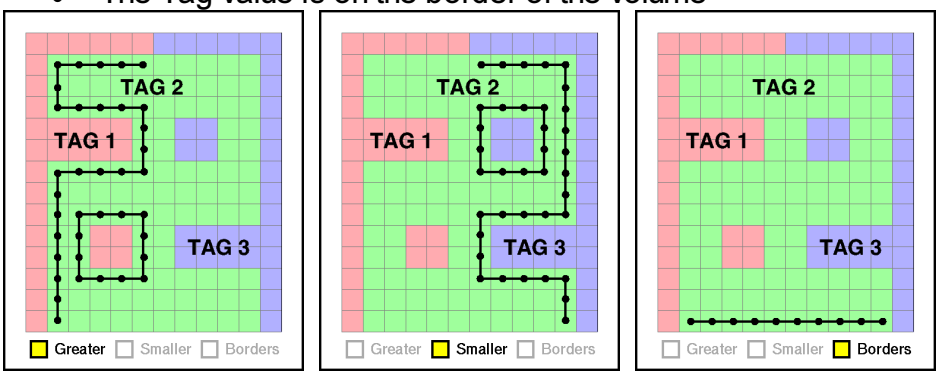
From the Graphic Interface



TAG Value Select the current Tag value. A surface will be created around each of the select Tags

Conditions These buttons control the surface creation conditions. For each selected Tag v a surface will be created on the border of a Tag pixel if:

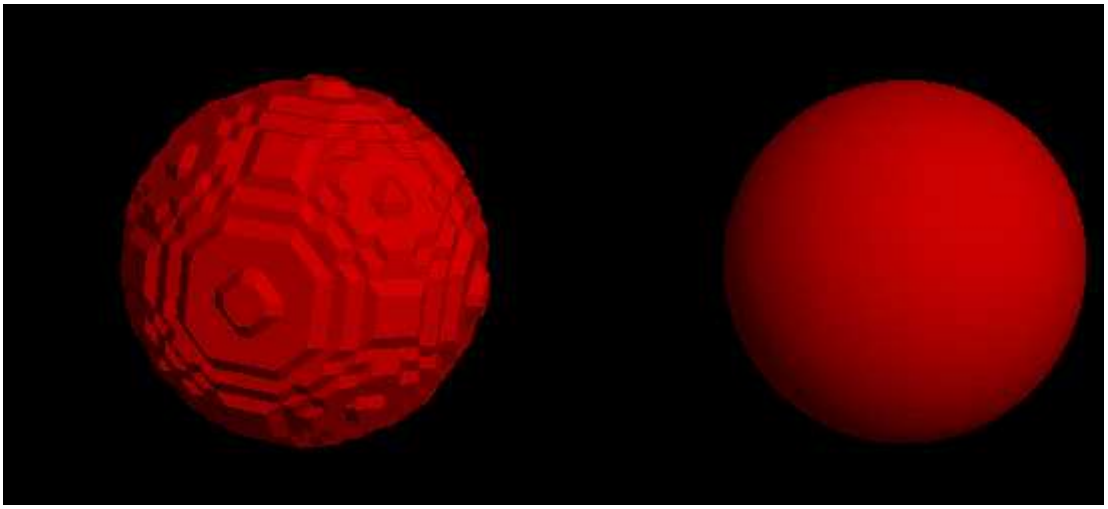
- The Tag value is greater than the Tag value of its neighbor
- The Tag value is smaller than the Tag value of its neighbor
- The Tag value is on the border of the volume



Threshold Interpolation

If your dataset has been segmented using one of the threshold based segmentat modes (Threshold, Region Growing, Histogram Segmentation...), then a threshol value has been assigned to each Tag. You can use these values to interpolate th position of the nodes of the reconstructed geometry between the voxels to obtain smoother surface.

When "Threshold Interpolation" is on, the "Edit Threshold" button is accessible. Activating this option will change the values displayed on the Tags buttons from t names, to their assigned threshold values. You can edit these values with a right on the Tag buttons.

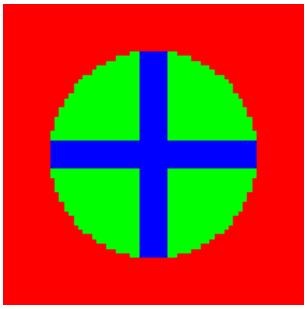


Threshold Interpolation: Off Threshold Interpolation: On

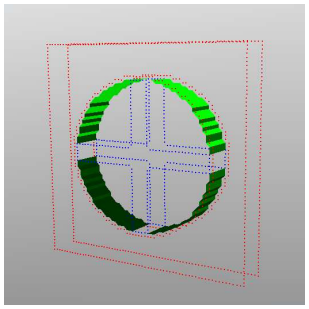
Create Geometry

Clicking this button will cause sliceOmatic to compute the 3D shell around al selected Tags

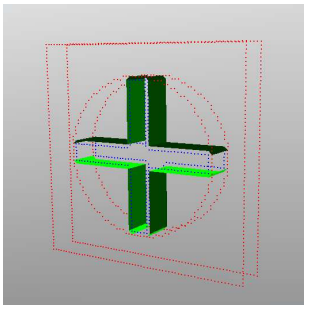
Example of reconstructions with different conditions.



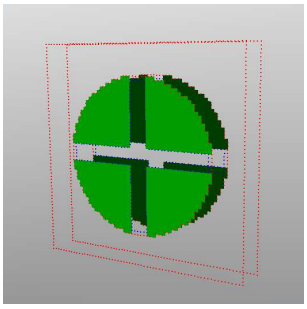
The **original TAG image**:
Outside = TAG 1,
circle = TAG 2,
cross = TAG 3



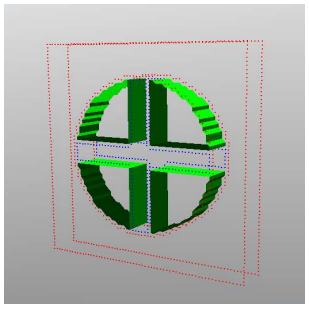
Shell of TAG 2 with the condition **Greater**



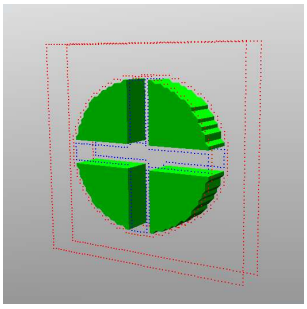
Shell of TAG 2 with the condition **Smaller**



Shell of TAG 2 with the condition
Border



Shell of TAG 2 with the condition
Greater & Smaller



Shell of TAG 2 with the condition
Greater & Smaller & Border

From the Display Area

There is no Display Area interaction specific to this mode.

From the Keyboard

There is no keyboard interface specific to this mode.

From the Command Line

A list of the commands available while in this mode can be found in the "SliceO_TAG\Shell....dll" section of appendix B.

The 2&3D Tools

The "2&3D" tools are tools that are relevant to all the modules. They are not limited to 2D or 3D modules, hence the "2&3D". These tools will always be available.

They include:

- **Memory Manager**
  The memory manager informs you of how much memory is available for the application and how much memory is used by different parts of the application. It is also responsible for automatically deleting low priority memory buffers when the application needs more space.

- **Point**
  This tool enables you to place points either on the 2D frames, or on the 3D surfaces.

- **Selection**
  This tool enables you to select/de-select frames. It is the only way to do so in 3D windows.

- **Snapshot**
  This tool is used to grab images of the Display Area and save them.

- **Units**
  This tool is used too select the units of measurement for distances, surfaces and volumes.

- **Windows**
  This tool enables you to open multiple windows. Each window can be used to display different frames in different modes.

The 2&3D Tools:
Memory Manager

This tool enables you to visualize and control the amount of memory used by sliceOmatic.

The memory used by the program is divided in 4 groups:

- The **sliceOmatic** memory. This is the memory used by the program without loading any images. This value is fixed and is estimated at 50 MBytes.
- The **Image** memory. To accelerate its graphic refresh, sliceOmatic keep copies of the images it displays. This is the memory used by the original slices and their copies.
- The **Undo** memory. This is the memory used to keep the undo operations.
- The **3D Geom** memory. This is the memory used by the 3D models.

In order to run, the program needs to fit into your computer's memory. That memory is composed of 2 parts: your RAM memory, and the swap space. The maximum size that the program can have is either the sum of these values, or the maximum addressable space in Windows (2Gb for 32 bits Windows) if this value is smaller than that sum.

When the program becomes too big to fit in the RAM memory, part of it will be "swapped" to the swap space on the hard drive and the program's performance will degrade. The automatic memory manager will try to prevent this by removing seldom used copies of the images from the image memory. As soon as the amount of memory used by the program reaches a critical "high water" mark, it will start cleaning the image memory until the memory usage falls under a safer "low water" mark.

You can also free up memory manually through the tool's graphic interface, you can "Cleanup" the image memory or delete the Undo and 3D Geom memory.

Doing a cleanup of the image memory will erase the copies of the images kept by the program. The only inconvenience this will cause is a small performance degradation since the next time the program needs to display these images, it will have to recreate them instead of fetching them from memory.

Deleting the Undo or the 3D Geom memory however will have some consequences: Deleting the Undo memory will remove any accumulated Undo. Deleting the 3D Geom memory will delete all 3D geometries from the program's memory.

Note:

To help graphic performances, the blow-up window is refreshed only when the processor is free. Thus, if the mouse is moved, the image in the blow-up window may not follow immediately.

From the Graphic Interface

Image Erase all the copies of the images kept in memory to accelerate refresh.

Undo Delete all Undo operations from the Undo buffer.

3D Geom Delete all the 3D geometries

Memory bar Indicate the amount of used memory. The total length of the bar is the maximum addressable memory space for the program. The yellow region represents the interval between the low water and the high water marks. The blue portion of the memory bar represents the Image memory, the cyan portion represents the Undo memory and the magenta portion represents the memory used by the 3D geometries.

From the Display Area

There is no Display Area interaction specific to this tool.

From the Keyboard

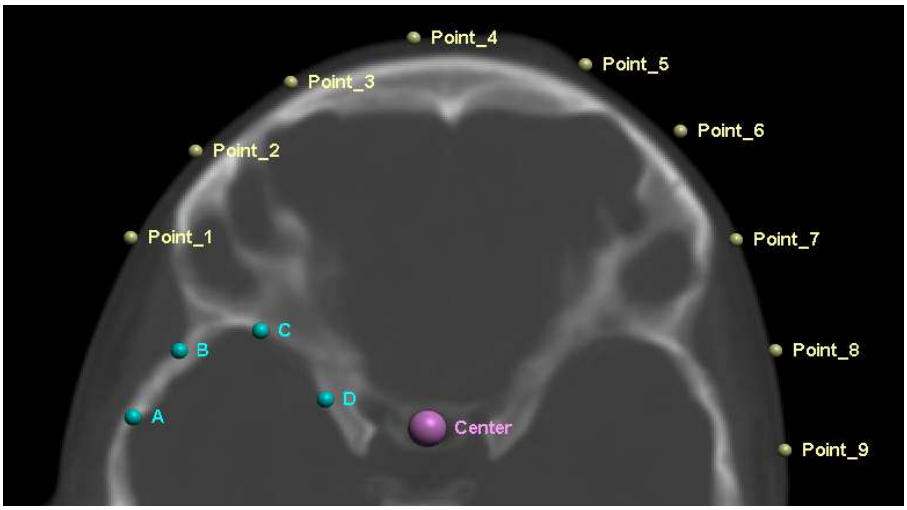
There is no keyboard interface specific to this tool.

From the Command Line

A list of the commands available for this tool can be found in the "SliceO_Tool\2&3D Memory.dll" section of **Appendix B**.

The 2&3D Tools:
Point

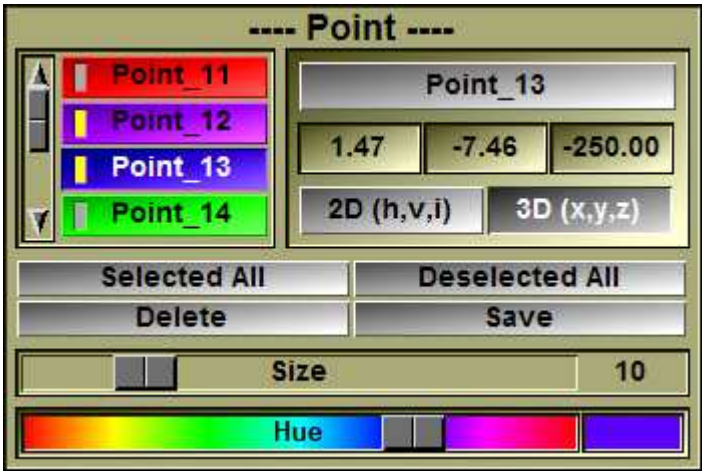
With this tool, you can place points on the 2D images. Points will be created under the cursor either with the left mouse button, or when you press the “Insert” key of the keyboard. Alternatively, pressing the right mouse button, or the “Delete” key of the keyboard will delete the point under the cursor (or the closest point if none is under the cursor). You can also use the mouse to drag a point.



The points are created in 3D space using the image’s origin, orientation and the pixel dimensions. The list of created points is shared between the Basic Module’s Point mode and the 3D Module’s Point mode. Points created on the 2D images will be visible in the 3D space and inversely, points created on the 3D surfaces will be visible on the 2D images.

Points can also be read from a file or manipulated from the command line.

From the Graphic Interface



Point List This is a list of all the Points. You can select the points you want to modify from this list. The indicator's light of the selected points is turned ON. The button of the last point to have been selected or unselected will appear pressed. It is the current point. Information on the current point will be displayed in the box at the right of the list. You can modify the position and name of the Current point from the controls in this box.

Name You can edit the name of a point by clicking on its name button. The button will be replaced by a text input window. Type in the new name and press "enter".

Position Box The 3D coordinates of the current point will be displayed here. Clicking in these boxes will enable you to modify their values from the keyboard.

Select All Select all the points from the list

Deselect All Deselect all the points from the list

Delete Delete all the selected points

Save Save all the selected points in a script file. A Save dialog box will appear. For each of the selected points, the following entries will be written in the file:
Point: "*name*" create *x y z*
Point: "*name*" color *R G B*
Point: "*name*" size *val*

Size Slider This slider is used to control the size of the selected points' graphic representation. At a size of zero (left) the point will be represented as a 2-pixels-wide point on the screen. At larger sides, a shaded sphere will be used to display the point.

Color Editor Upon creation, the points are assigned a random color. Afterward, using the Color Editor, you can change the color of all selected points.

From the Display Area

Points will be created under the cursor with the left mouse button, and deleted with the right mouse button. You can also use the mouse to drag a point.

From the Keyboard

The following keys can also be used while the cursor is over a frame:

Key	*Action*
Del	Delete the nearest point

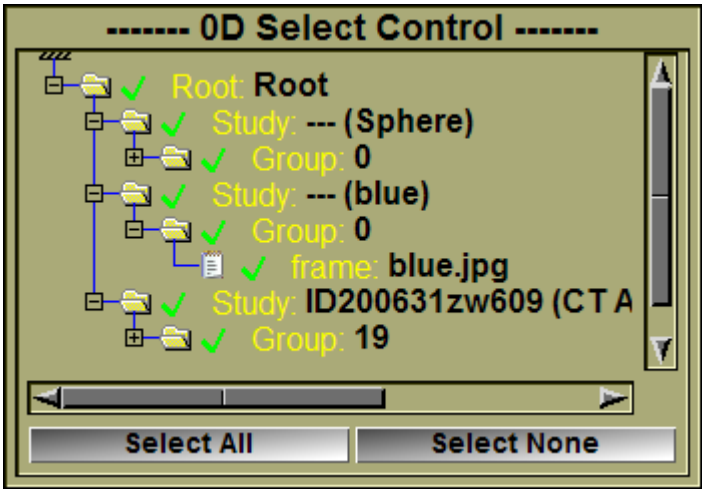
From the Command Line

A list of the commands available for this tool can be found in the "SliceO_Tool\2&3D Point.dll" section of **Appendix B**.

The 2&3D Tools:
Selection

This tool enables you to change the frame selection. The concept of "Selection" is explained in "Frame Selection" of **The Display Area** section.

From the Graphic Interface



You can select individual frames, or complete branches of the database tree just by clicking on their nodes. You can use the "Ctrl" key to toggle the current selection, and you can use the "Shift" key to select a range of frames. Selected frames will have a green check-mark. Yellow check-marks indicate branches that have some of their frames selected but not all.

Select All Select all frames

Select None De-select all frames

From the Display Area

Selecting frames in the Display Area will update the graphic interface of this tool. Selecting frames from the interface will automatically update the Display Area.

From the Keyboard

There is no keyboard interface specific to this tool.

From the Command Line

There is no command line or variables associated with this tool.

The 2&3D Tools:
Snapshot

You can save the content of the Display Area window with this tool.

You first select the format for the snapshot. The choice is between Windows' BMP format and TARGA. Each time you press the "Click!" button, an image of the Display Area window will be saved to the scratch directory (by default the "C:\Temp" directory). The image will be named "sliceO_*xx.ext*" where "ext" is either "tif", "png", "jpg", "gif", "bmp" or "tga" and "xx" is a sequential number assigned to the image upon creation.

Unless you use the command line interface with the "file_name" option, the program cannot save more than 100 images (from 00 to 99).

By default, clicking the "Click" button grab the entire sliceOmatic window. If you use the script command "snapshot: [t_window] click [filename]" to capture an image, then the "t_window" parameter is a template for the display window that is capture. That way you can have a screen capture of just one of the display windows.

For example: "snapshot: 1 click" will grab only the content of the main display window.

If you use a lossy compression, they variable $SNAPSHOT_QUALITY control the compression level (1=poor,100=best).

From the Graphic Interface

Click! Create an image with the content of the Display Area window.

Format Select the format of the snapshot image with this tool. The choices are: TIFF, PNG, JPEG, GIF, BMP and TARGA.

From the Display Area

There is no Display Area interaction specific to this tool.

From the Keyboard

The following commands can also be mapped to keyboard keys as a shortcut:

Key	*Action*
Print Scrn	Create an image with the entire sliceOmatic window.

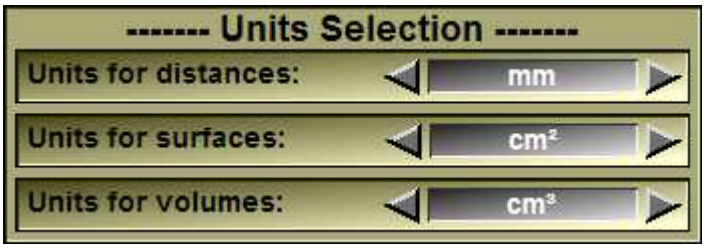
From the Command Line

A list of the commands available for this tool can be found in the "SliceO_Tool\2&3D Snapshot.dll" section of **Appendix B**.

The 2&3D Tools:
Units

A number of tools and mode display and save dimension information. You can change the units used for these. These are just cosmetic changes, the actual computations are all based on the units specified in the image's headers.

From the Graphic Interface



Distances The choices are: m, cm, mm or µm

Surfaces The choices are: m^2, cm^2, mm^2 or μm^2

Volumes The choices are: m^3, cm^3, mm^3 or μm^3

From the Display Area

There is no Display Area interaction specific to this tool.

From the Keyboard

There is no keyboard interface specific to this tool.

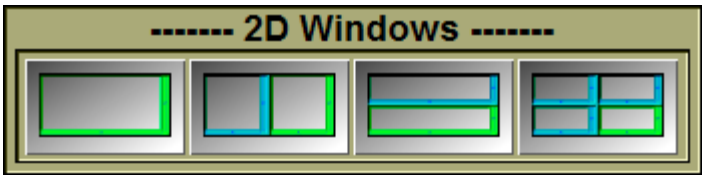
From the Command Line

There is no command line or variables associated with this tool.

The 2&3D Tools:
Windows

This tool enables you to control the windows in the Display Area

From the Graphic Interface



Single Only the lower right window is active. The other 3 windows are collapsed. The separator bars are located at the top and the left of the Display Area. This is the same state as the one you get when you enter sliceOmatic.

2 Horizontal Only the lower left and right windows are active. The other 2 windows are collapsed. The vertical separator bar is between the 2 windows, the horizontal is the top of the Display Area.

2 Vertical Only the upper and lower right windows are active. The other 2 windows are collapsed. The horizontal separator bar is between the 2 windows, the vertical is the left of the Display Area.

4 Windows All 4 windows are active. Both separator bars are between the windows.

From the Display Area

The first click in an unselected window will change the currently selected window to the window under the cursor.

From the Keyboard

There is no keyboard interface specific to this tool.

From the Command Line

A list of the commands available for this tool can be found in the "SliceO_Tool\2&3D Window.dll" section of **Appendix B**.

The 2D Tools

The "2D" tools are only relevant when working with 2D frames. They will be unavailable if the current window is associated with a 3D module.

They include:

- **blow-up**
  This tool displays a blow-up of the region under the cursor.

- **Color Map**
  This tool is used to change the color map used to display the GLI pixels in the frames.

- **Color Scheme**
  This tool enables you to change the contrast and intensity of the GLI pixels. Since we are displaying images that often have 16 bits of grey values on screen that can only display 8 bits of color intensity, we need this tool to adjust how the grey levels are displayed.

- **Frame Selection**
  This tool is used to change the current frame, change the current display mode or change the scaling of the selected frames.

- **Gamma**
  This tool is used too modify the "Gamma" of the screen.

- **Overlays**
  This tool enables you to display text overlays on top of the frames.

- **Pixel Info**
  This tool gives you information on the pixel under the cursor.

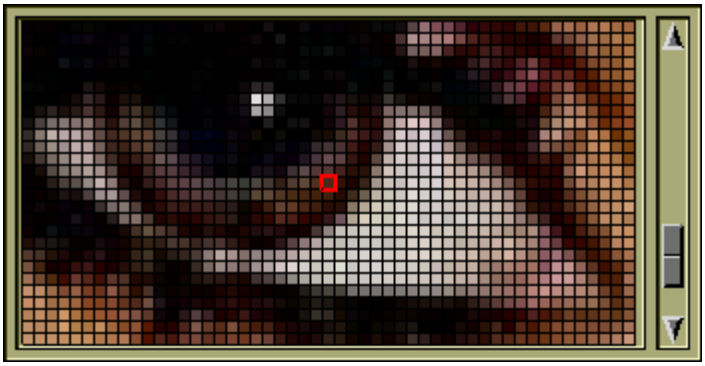
The 2D Tools:
Blow-Up

This tool gives a magnified view of the image under the cursor in the Display Area. The center red square in the Blow-Up area corresponds to the pixel under the cursor in the Image Display Area.

Note:

To help graphic performances, the blow-up window is refreshed only when the processor is free. Thus, if the mouse is moved, the image in the blow-up window may not follow immediately.

From the Graphic Interface



The Image A blow-up of the image, centered on the position of the cursor will be displayed here.

The Scale slider The magnification of the blow-up image can be modified with the slider located to its right. At its smallest magnification (cursor at the bottom), one pixel of the screen shows one pixel of the image.

From the Display Area

The image in the blow-up window is centered around the current cursor position when the cursor is over a frame.

From the Keyboard

There is no keyboard interface specific to this tool.

From the Command Line

There is no command line or variables associated with this tool.

The 2D Tools:
Color Map

If the GLI (Grey Level Images) are monochrome, you can display them using a color map. By default the standard grey level color map is used. The lowest values are black and the highest values are white.

From the Graphic Interface

The map buttons You have a choice of 4 color maps, the first 2 are standard and inverse grey levels, the second 2 use colors and can be modified by the user.

From the Display Area

There is no Display Area interaction specific to this tool.

From the Keyboard

There is no keyboard interface specific to this tool.

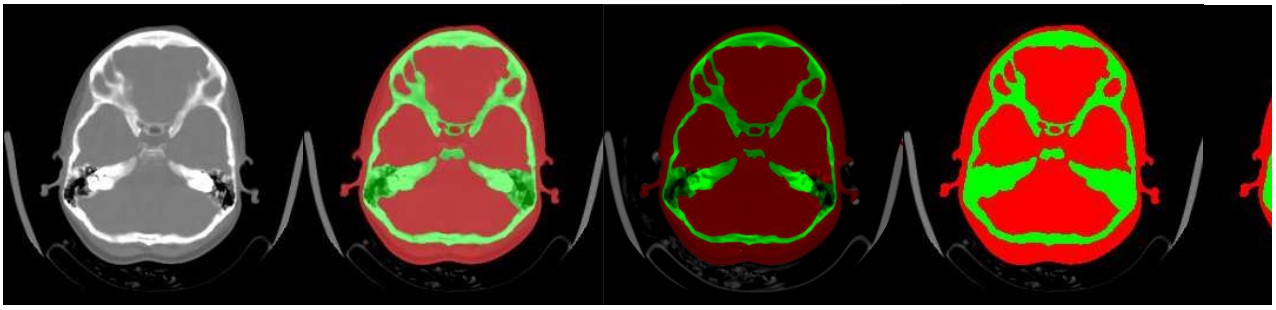
From the Command Line

A list of the commands available for this tool can be found in the "SliceO_Tool\2D Colormap.dll" section of **Appendix B**.

The 2D Tools:
Color Scheme

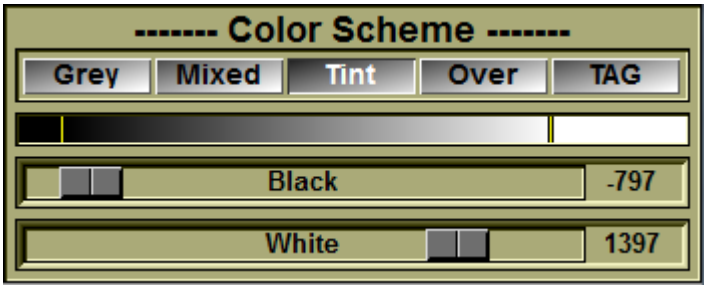
The color scheme will affect the appearance of the images in the display section.

The four buttons on the tool, **“Grey”**, **“Mixed”**, "**Tint**", **“Over”** and **“Tag”** control how the images are displayed in the Image Area. They are used to display either the GLI image, the TAG image or a combination of both.



Grey Mixed Tint Over

From the Graphic Interface



Grey Only the GLI image is visible.

Mix The Tag color image is mixed with the GLI image. By default, each image is half its normal intensity and the two are added together. The “mixed” value can be modified with the “shift +” or “shift -“ keys so that the TAG image will appear more or less dense.

Note:

Since rev-4a, the Mix mode is slightly different, if you prefer the old look, set the variable "$COLOR_CLASIC" to 1, or press "Shift F2".

Tint The GLI image is tinted with the TAG color. . The “tint” value can be modified with the “shift +” or “shift -“ keys so that the TAG image will appear more or less dense.

Note:

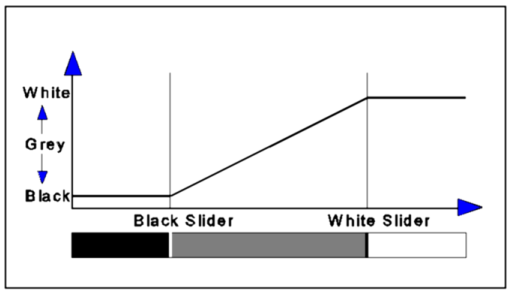
If the GLI values are black, the TAG color will not be seen.

Over The Tag color image is displayed over the GLI image. The GLI image is thus only visible in the regions where the Tag values are 0.

TAG Only the Tag color image is visible.

Color map This window displays the grey level color map used to display the GLI images.

Black / White These sliders are used to control the contrast and brightness of the GLI image.



- All values smaller than the value of the black slider are black.
- All values greater than the value of the white slider are white.
- All values in between are interpolated linearly.

From the Display Area

There is no Display Area interaction specific to this tool.

From the Keyboard

The following commands can also be mapped to keyboard keys as a shortcut:

Key map	*Action*
F1 **F2** **F3** **F4**	Set the color scheme mode for all windows: F1=Grey, F2=Mixed, F3=Tint and F4=Over
"SHIFT_MINUS" & **"SHIFT_PAD_MINUS** **"SHIFT_PLUS" &** **SHIFT_PAD_PLUS**	Decrease / increase the “mix” and tint density by .1 (from: 0.1 to 0.9)

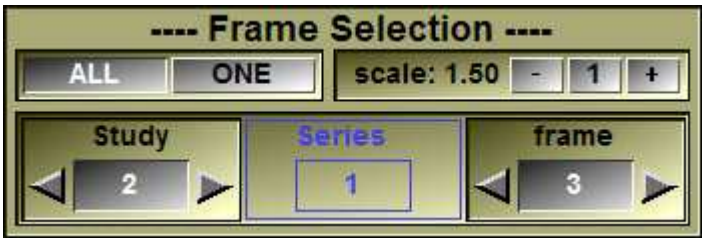
From the Command Line

A list of the commands available for this tool can be found in the "SliceO_Tool\2D Color_Scheme.dll" section of **Appendix B**.

The 2D Tools:
Frame Selection

With this tool, you control the images shown in the **Display Area**. You can change the scale of the displayed images, select the current image being displayed, or choose to display all the images.

From the Graphic Interface



ALL / ONE

These buttons are used to toggle the Display Mode of the images in the Display Area between **Mode One** (one image is displayed) and **Mode All** (all the images are displayed). For more information on these modes, refer to **Display Area: The 2D Window**. Pressing the space-bar will also toggle between these modes.

Scale

You control the size of the images with these buttons. The current scale factor being displayed is either a negative number, the shrink factor, in **Mode All**, or a positive number, the magnification factor, in **Mode One**.

The "+" and "-" buttons:
Clicking on the "+" or "-" buttons will increase or decrease the size of the images in the Display Area. Pressing the **"+"** or **"-"** keys on the keypad will have the same effect.

Class List

The program will look through the entire database tree to locate nodes that have more than 1 child. The classes associated with these children will each have a tool in the list.

Each tool that represents a class in the current branch of the tree that has more than one instance will be enabled. If a class has a tool in the list but does not have more than one instance in the current branch, it will be disabled.

For example, with the default "Root/Study/Series/Frame" tree:

- If you have more than 1 study under the root, then there will be a tool to select the studies.
- If you have only 1 series under the current study, then there is not tool for the series. However, if under a study other than the current one there is more than 1 series, then there will be a series tool but it will be disabled.
- If you have more than 1 frame under the current series, then you have a tool to select the frame.

From the Display Area

The values displayed in the interface will reflect the mode of the selected window (Mode One or Mode All), the scaling of the currently selected group and the study, group and image number of the image under the cursor.

From the Keyboard

The following commands can also be mapped to keyboard keys as a shortcut:

Key map	*Action*
Space	Toggle the display mode for the current 2D window between Mode ONE and Mode ALL
"=" & PAD_PLUS **"-" & PAD_MINUS**	Increase / decrease the magnification of the selected image(s)
PAGE_UP **PAGE_DOWN**	Go to the next / previous frame of the current class
HOME **END**	Change the frames by changing the current branch of the tree (change the parent of the current frame)
INSERT **DELETE**	Change the frames by changing the current branch of the tree (change the parent of the parent of the current frame)

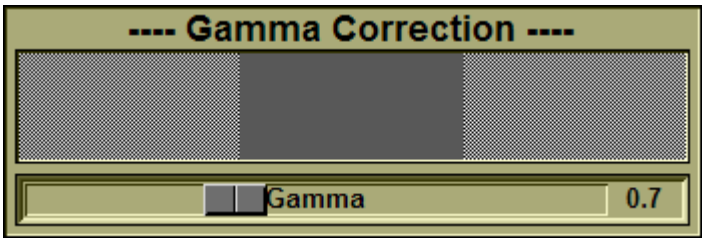
From the Command Line

A list of the commands available for this tool can be found in the "SliceO_Tool\2D Frame.dll" section of **Appendix B**.

The 2D Tools:
Gamma

The Gamma correction is used to compensate for the non-linear response of the screen. The grey-level values of the image are corrected to take the screen response into account.

From the Graphic Interface



Grey Image The Gamma correction value is set by comparing the 3 grey surfaces: two of these (left and right) are composed of a mix of black and white lines, the other (center) is a mid-level grey. The visual stimuli of all 3 surfaces should be the same.

Gamma slider This slider is used to select the Gamma correction value. Adjust the slider so that the 3 grey surfaces have the same intensity.

From the Display Area

There is no Display Area interaction specific to this tool.

From the Keyboard

There is no keyboard interface specific to this tool.

From the Command Line

A list of the commands available for this tool can be found in the "SliceO_Tool\2D Gamma.dll" section of **Appendix B**.

The 2D Tools:
Overlay

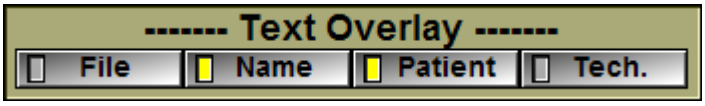
Patient and image information can be overlaid on the frames.

Note:

If the display size of the images is too small, only the image number from the File overlay will be displayed. Also, the Name, Patient and Technical overlays will not be displayed.

From the Graphic Interface

There are 4 levels of information you can display corresponding to the 4 buttons of this tool.



File Image sequence number in sliceOmatic and the file name (In white)

Name Patient name, patient id, hospital name, physician name, study, series and image number, date and time of acquisition (In yellow). Also, if the frames are “trusted”, labels identifying the Left, Right, Head, Feet, Anterior and Posterior sides of the patient will be displayed

Note:

If the "Anonymous" flag is set, then the patient's name will be "Anonymous", and no personal information about the patient will be displayed. See The Configuration Menu for more information on the Anonymous flag.

Patient Patient sex, age, and weight, radiologist name, comments (In green)

Technical Technical information on the frame format, scanner’s manufacturer and model, modality (CT, MR...), orientation, scan protocol, contrast agent, table position and height, scan time, slice position, slice thickness, pixel dimension and other information depending on the frame's modality (In cyan).

From the Display Area

There is no Display Area interaction specific to this tool.

From the Keyboard

There is no keyboard interface specific to this tool.

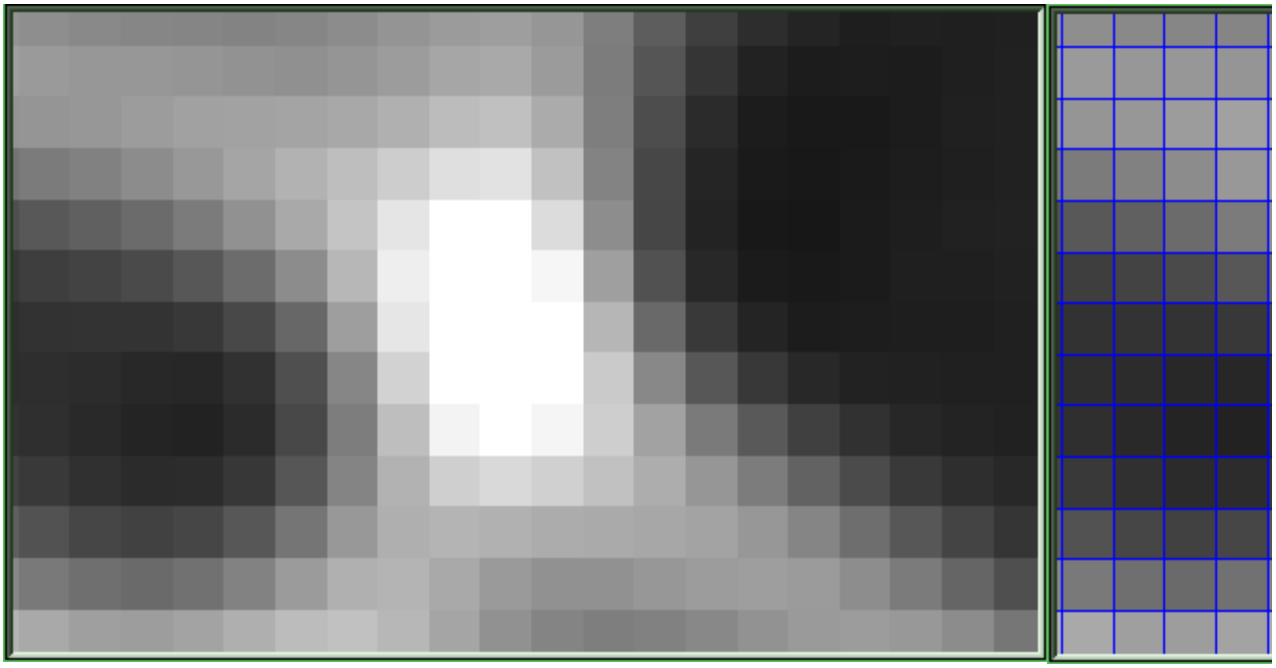
From the Command Line

A list of the commands available for this tool can be found in the "SliceO_Tool\2D

Overlay.dll" section of **Appendix B**.

The 2D Tools:
Pixel Grid

With this tool, you can display a grid to highlight the pixels when the image is magnified.



Pixel Grid: Off

Note:

To grid is only displayed at high magnification. If the image maginification is inferior to $GRID_MIN (default=5) the grid is not displayed.

From the Graphic Interface

Grid On/Off This button is used to turn the grid on or off

Grid color This slider is used to change the color of the grid

From the Display Area

There is no Display Area interaction specific to this tool.

From the Keyboard

The following commands can also be mapped to keyboard keys as a shortcut:

Key map	*Action*
"g"	Toggle the display of the pixel grid.

From the Command Line

A list of the commands available for this tool can be found in the "SliceO_Tool\2D Pixel Grid.dll" section of **Appendix B**.

The 2D Tools:
Pixel Info

In this tool, information about the pixel under the mouse pointer is displayed.

From the Graphic Interface

Pixel info This box displays the horizontal and vertical position of the cursor in the image. This position is given in image pixels. The value (0,0) is the lower left corner.

GLI info This box displays the numerical value of the pixel under the cursor. For CT images, this value should be in Houndsfield units. For color images, the 3 RGB values will be displayed.

Tag info This box displays the Tag value of the pixel under the cursor. The Tag value can range from 0 to TAG_MAX. (By default, TAG_MAX=127 and can be changed through the Preferences interface)

Coord. Info This box displays the 3D coordinates of the pixel under the cursor. This value is given in the patient's reference system (as defined in DICOM). It is computed from the image origin, the pixel dimensions and the gantry tilt.

From the Display Area

The values displayed in the interface are continuously updated to reflect any mouse movements.

From the Keyboard

There is no keyboard interface specific to this tool.

From the Command Line

There is no command line or variables associated with this tool.

The TAG Tools

The "TAG" tools are only relevant when working with TAG component of the frames. They will be unavailable if the current window is not associated with one of the segmentation (TAG) module.

They include:

- **TAG Color**
  This tool enables you to change the color of the different TAGs.

- **TAG Delete**
  This tool enables you to delete specific TAGs.

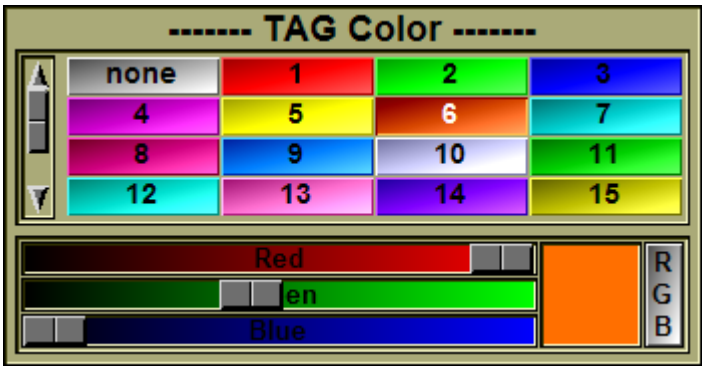
- **TAG Lock**
  This tool is used to lock/unlock TAGs. This is very useful when working with different segmentation techniques on the same frames. You can protect (lock) the work already done.

- **TAG Surface/Volume**
  This tool is used to compute the surfaces and volumes of the segmented images.

The TAG Tools:
Tag Color

The Tag Color tool is used to change the color associated with a Tag value.

From the Graphic Interface



Tag List Select a tag from the list.

Color Editor You can modify the color associated with the current Tag value with the 3 sliders.

From the Display Area

There is no Display Area interaction specific to this tool.

From the Keyboard

There is no keyboard interface specific to this tool.

From the Command Line

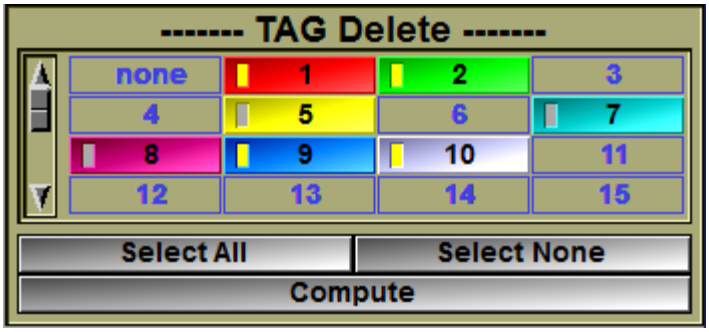
There is no command line or variables directly associated with this tool. However, you may want to look at the system dll "SliceO_System\Tag.dll" section of **Appendix B**.

The TAG Tools:
Tag Delete

The Tag Delete tool is used to delete all the pixels of a specific Tag value from the selected frames.

The interface will have enabled buttons for all the TAG values that exist in the selected frames. The TAG values that are to be deleted will have the button's yellow light activated. Pressing the "Compute" button will preform the deletion.

From the Graphic Interface



Tag List Select a tag from the list. Only TAG values that exist in the selected frames will have their buttons enabled.

Select All Select all enabled TAG value

Select None De-select all enabled TAG value

Compute Delete the selected TAG values from all the selected frames.

From the Display Area

There is no Display Area interaction specific to this tool.

From the Keyboard

There is no keyboard interface specific to this tool.

From the Command Line

There is no command line or variables directly associated with this tool. However, you may want to look at the system dll "SliceO_System\Tag.dll" section of **Appendix B**.

The TAG Tools:
Tag Lock

The TAG Lock tool is used to protect the results of your segmentation, enabling you to concentrate on segmenting a new tissue without fear of affecting the work already done.

By simply selecting the desired tags from the interface, you can lock them so that they will not be modified by any further operations. The corresponding tag buttons in the mode interfaces will be disabled.

From the Graphic Interface

Tag List Select a tag from the list to lock it.

Lock All Lock all the tags.

Unlock All Unlock all the tags

From the Display Area

There is no Display Area interaction specific to this tool.

From the Keyboard

There is no keyboard interface specific to this tool.

From the Command Line

A list of the commands available for this tool can be found in the "SliceO_Tool\Tag Lock.dll" section of **Appendix B**.

The TAG Tools:
Tag Surface/Volume

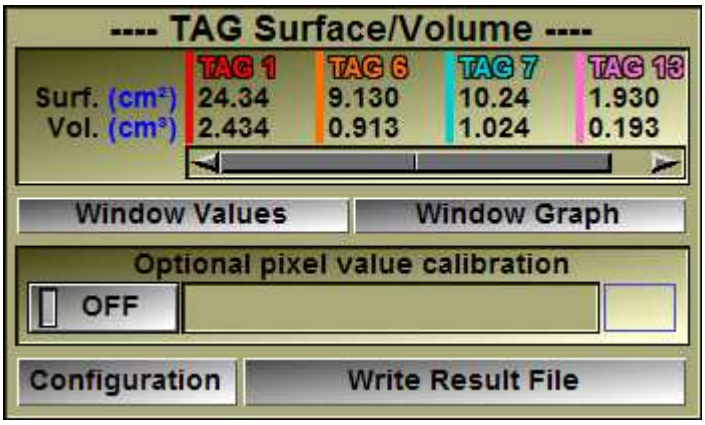
In this tool, you can compute the surface and volume of the Tag images for the selected frames. These results can be displayed numerically and in a graph and can be written to 2 different files: a spreadsheet compatible file or an ASCII text file.

By default the tool display the surfaces (in cm^2) and volumes (in cm^3) for all the tag values existing in the selected frames. You can change the units for these measurements with the "**Units**" tool, and you can change the list of displayed values with the "**Results: Display**" command.

WARNING:

Volumes are only computed for frames that have the same parent class in the database tree. If frames from more than 1 parent are selected a warning message will be displayed in the text window and the volume computation will give invalid results

From the Graphic Interface



Surface/Volume Some of the values computed by this tool are displayed in this window. If you want access to all the values computed, you need to open a secondary window with the "Window Values" button. You can control which of the values are displayed with the command line. By default, the surfaces and volumes are displayed.

Window Values

Clicking this button will cause a secondary window to be created. This window will contain all the values computed by the tool.

The values computed are:

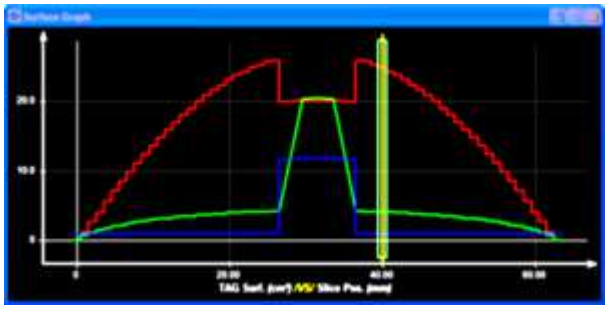
- "Surface Units", The surface of each TAG in surface units (see the Units tool)
- "Volume Units", The volume of each TAG in volume units (see the Units tool)
- "Surface Pixel", The surface of each TAG in pixels
- "Volume Voxel", The volume of each TAG in voxels
- "GLI Mean", The mean value of the pixel under each TAG in the pixel's units
- "GLI Min", The minimum value of the pixel under each TAG in the pixel's units
- "GLI Max", The maximum value of the pixel under each TAG in the pixel's units
- "GLI SD", The standard deviation of the values of the pixel under each TAG.

The values that are not selected to be saved in the result files will be displayed in grey. This can be changed either from the configuration menu or through the command line.

Only one such window can exist at any time, clicking this button while the window is already created will pop the associated window on top of the other windows.

Window Graph

Clicking this button will cause a secondary window to be created. This window will contain a graph of the surface of each tag on each slice as a function of the slice position in space.



Only one such window can exist at any time, clicking this button while the window is already created will pop the associated window on top of the other windows.

Calibration

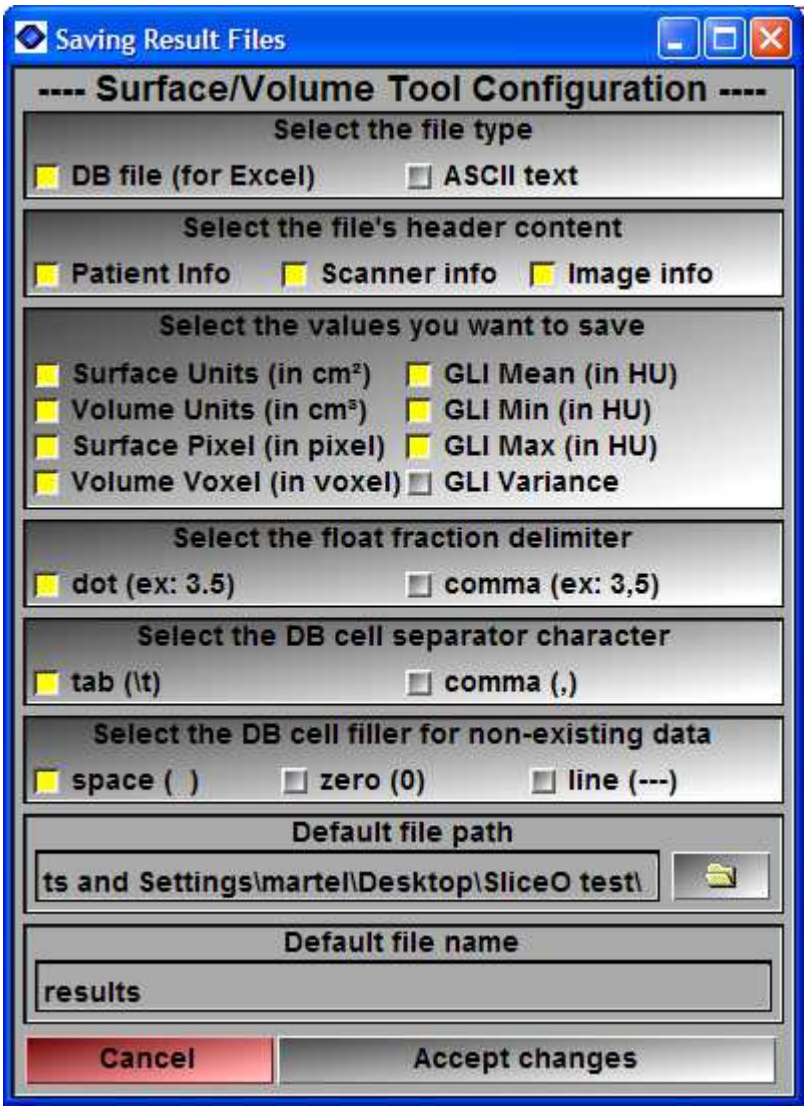
If you know what is the relationship between pixel values and something else, you can add a calibration file that will define this relationship. For example, if you know how to compute a tissue density from the CT Housfield units (HU), you can use a calibration file to let the program know about this. In this way, the mean, min and max values will be in densities instead of HU.

Configuration

Clicking this button will open a configuration menu. In this menu you will be able to specify the content and form of the result files you can create from this tool. The configuration menu is described in more detail further down.

Write Result Create a result file. The results can be either in standard ASCII text, or in a tabulated form compatible with Excel. You can set the default for this with the configuration menu, or changing the "save as type" parameter in the confirmation pop-up window.

The Configuration Menu



File Type Result files can either be in plain ASCII text (compatible with notepad or any text editors) or in "DB file", a tabulated form, compatible with Excel or most spreadsheet programs.

Headers

You can have 3 optional headers in the files you created:

The Patient Info header

For each patient used in creating the file we will have (if the information is present in the image's header):

- Patient Name
- Patient ID
- Patient sex
- Date of Birth
- Patient weight (Kg)
- Patient height (m)

The Scanner Info header

For each scanner used in creating the file we will have (if the information is present in the image's header):

- Modality
- Manufacturer
- Model

The Image Info header

For each image used in creating the file we will have (if the information is present in the image's header):

- Resolution X Y
- Pixel Dim. X Y Z (mm)
- Horiz. Dir. X Y Z
- Vert. Dir. X Y Z

Saved Measures

You can select which of the available measures you want to save in your result files.

Float Fraction

You can specify whether you want the fractions in your floating point values to be delimited by a dot (".") (example:12.34) or a comma (",") (example: 12,34). By default it is the dot.

Cell Delimiter

You can specify the character you want to use to mark the end of a cell in your spreadsheet result files. By default it is the "tab" character.

Cell Filler

You can specify what should be placed in cells that have no values. For example, in a result file for multiple frames, if a TAG is not present in all the frames, then in some cells we will not have surfaces and volumes for it.

Default Path

You can specify the default path for the results files. There is only one "default path" parameter in sliceOmatic, so by default, this value is set to the path of the last file read or written.

Default Name

You can set a default name for your result files.

Cancel

Close the configuration menu without saving any of the changes you just made.

Accept Changes Accept the changes you made and close the configuration menu. Please note that these changes only affect the current invocation of sliceOmatic. If you want to save the changes for future invocations of the program, you need to use the "Script Save As..." option of the File menu.

The Surface Area Computation

The surface area covered by a Tag value is computed by multiplying the number of pixels of that value by the surface area of one pixel.

The Volume Computation

Computing the volume of a Tag value for one image is fairly simple. We compute the surface area covered by the Tag value and multiply by the image thickness. For multiple images, a problem arises if the images overlap or if there are gaps between them.

Note:

In order to be able to compute a volume, the following criteria must be met:

- All selected frames are siblings (meaning that they are all children of the same parent in the database tree)
- All frames must be parallel
- All frames must have the same resolution
- All frames must have the same pixel dimensions

Jsing the notation:

= position in Z of the image A

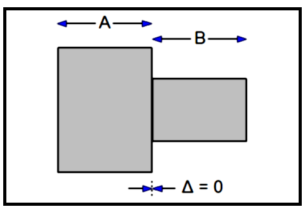
Z_A = position in Z of the image A
N_A = number of pixels of a certain Tag value in image A
P_A = surface of a pixel in image A
S_A = surface covered by a Tag value on the image A (= N_A P_A)
T_A = thickness of image A
Δ_{AB} = gap or overlap between the two images A and B
V_{AB} = volume of a Tag value between the two images A and B

The gap Δ between two images A and B is computed as follows:
$\Delta_{AB} = ABS(Z_A - Z_B) - (½T_A + ½T_B)$

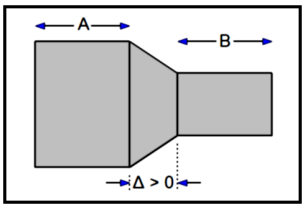
Depending on the value of Δ, there are 3 possibilities when computing the volume between 2 images:

$\Delta = 0$ There is no gap, and the computation of the volume of the Tag value between A and B is straightforward:

$V_{AB} = ½T_AS_A + ½T_BS_B$

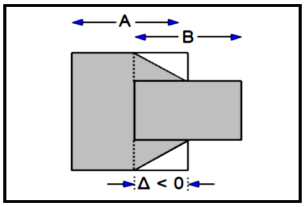


$\Delta > 0$ There is a gap between the slices and we have to interpolate the volume in the gap. This volume will be approximated by a truncated pyramid joining the volumes of both slices.



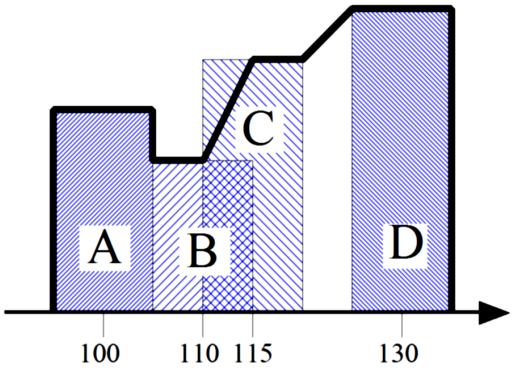
$$V_{AB} = \tfrac{1}{2}T_A S_A + \tfrac{1}{2}T_B S_B + \Delta_{AB}\,(\tfrac{1}{3}\,ABS(S_A - S_B) + MIN(S_A, S_B))$$

$\Delta < 0$ The images overlap and we have to compute the volume in this overlap region. The volume in the region without overlap is computed as before, and in the overlap we will again use the truncated pyramid.



$$V_{AB} = (\tfrac{1}{2}T_A + \tfrac{1}{2}\Delta_{AB})\,S_A + (\tfrac{1}{2}T_B + \tfrac{1}{2}\Delta_{AB})\,S_B - \Delta_{AB}\,(\tfrac{1}{3}\,ABS(S_A - S_B) + MIN(S_A, S_B))$$

Example: We will compute the volume of the Tag value 1 for 4 images:



All slices are 10.0 mm thick (T_a to T_D = 10.0)
All pixels are 0.25 mm^2 (P_A to P_D = 0.25)

Image A:
has 400 pixels of Tag value 1 (N_A = 400)
is positioned at Z = 100.0 (Z_A = 100.0)

image B:
has 300 pixels of Tag value 1 (N_B = 300)
is positioned at Z = 110.0 (Z_B = 110.0)

image C:
has 500 pixels of Tag value 1 (N_C = 500)
is positioned at Z = 115.0 (Z_C = 115.0)

image D:
has 600 pixels of Tag value 1 (N_D = 600)
is positioned at Z = 130.0 (Z_D = 130.0)

First, we compute the surface area covered by the Tag value 1 for the 4 images:

$S_A = N_A P_A = 400 * 0.25 = 100\ mm^2$
$S_B = N_B P_B = 300 * 0.25 = 75\ mm^2$
$S_C = N_C P_C = 500 * 0.25 = 125\ mm^2$
$S_D = N_D P_D = 600 * 0.25 = 150\ mm^2$

Then we compute the volume covered by the Tag value 1. This computation is divided in 5 steps:

- **Half the volume of image A:** (from Z=95 to Z=100)

$V_A = ½T_A S_A$
$= ½\ 10.0 * 100$
$= 500\ mm^3$

- **The volume between A and B:** (from Z=100 to Z=110)

the value of Δ is:
$\Delta_{AB} = ABS(Z_A - Z_B) - (½\ T_A + ½\ T_B)$
$= ABS(100 - 110) - (½ * 10 + ½ * 10)$
$= 0\ mm$
there is no gap! The volume is:
$V_{AB} = (½T_A S_A) + (½T_B S_B)$
$= (\ ½\ 10.0 * 100) + (½\ 10.0 * 75)$
$= 875\ mm^3$

- **The volume between B and C:** (from Z=110 to Z=115)

the value of Δ is:
$\Delta_{BC} = ABS(Z_B - Z_C) - (½\ T_B + ½\ T_C)$
$= ABS(110 - 115) - (½\ 10 + ½\ 10)$
$= -5\ mm$
the images overlap by 5 mm! The volume is:
$V_{BC} = (½T_B + ½\Delta_{BC})\ S_B + (½T_C + ½\Delta_{BC})\ S_C$
$- \Delta_{BC}\ (⅓\ ABS(S_B - S_C) + MIN(S_B, S_C))$
$= ((½ * 10.0 + ½ * -5) * 75) + ((½ * 10.0 + ½ * -5) * 125)$
$- -5 * (\ ⅓\ ABS(75 - 125) + MIN(75, 125)\ )$
$= 187.5 + 312.5 - -5 * (⅓ * 50 + 75)$
$= 958.33\ mm^3$

- **The volume between C and D:** (from Z=115 to Z=130)

the value of Δ is:
$\Delta_{CD} = ABS(Z_C - Z_D) - (½\ T_C + ½\ T_D)$
$= ABS(115 - 130) - (½\ 10 + ½\ 10)$
$= 5\ mm$
there is a 5 mm gap! The volume is:
$V_{CD} = ½T_C S_C + ½T_D S_D + \Delta_{CD}\ (⅓\ ABS(S_C - S_D) + MIN(S_C, S_D))$
$= (\ ½\ 10.0 * 125) + (½\ 10.0 * 150)$
$+ 5 * (⅓\ ABS(125 - 150) + MIN(125, 150))$
$= 625 + 750 + 5 * (⅓\ 25 + 125)$
$= 2041.66\ mm^3$

- **Half the volume of image D:** (from Z=130 to Z=135)

$V_D = ½T_D S_D$
$= ½\ 10.0 * 150$

$= 750\ mm^3$

The total volume is:

$V = V_A + V_{AB} + V_{BC} + V_{CD} + V_D$
$= 500 + 875 + 958.33 + 2041.66 + 750$
$= 5125\ mm^3$

The Mean Value Computation

The mean value is computed as:
Sum of all pixel values / Number of pixels = $\Sigma x_i / n$

The SD "σ" Computation

The standard deviation is computed using:
σ = sqrt(ABS(variance)).

The variance is computed using:

$$variance = \frac{\Sigma(x_i^2 - mean^2)}{n-1}$$

where

$$\Sigma(x_i^2 - mean^2) = x_1^2 - mean^2 + x_2^2 - mean^2 + ... + x_n^2 - mean^2 = x_1^2 + x_2^2 + ... + x_n^2 - n*mean^2 = \Sigma x_i^2 - n*mean^2$$

and:

$$mean = \Sigma x_i / n$$
$$mean^2 = (\Sigma x_i)^2 / n^2$$
$$n*mean^2 = (\Sigma x_i)^2 / n$$

So

$$Variance = \frac{\Sigma(x_i^2 - mean^2)}{n-1}$$

$$= \frac{\Sigma x_i^2 - (\Sigma x_i)^2 / n}{n-1}$$

$$= \frac{n*\Sigma x_i^2 - (\Sigma x_i)^2}{n*(n-1)}$$

From the Display Area

There is no Display Area interaction specific to this tool.

From the Keyboard

There is no keyboard interface specific to this tool.

From the Command Line

A list of the commands available for this tool can be found in the "SliceO_Tool\Tag Surface.dll" section of **Appendix B**.

The Calibration File

The calibration file is a script file containing commands specific to the pixel calibration.
A list of these commands can be found in the "SliceO_Tool\Tag Surface.dll" section of **Appendix B**.

The 3D Tools

The "3D" tools are only relevant when working with 3D windows. They will be unavailable if the current window is associated with a 2D or TAG module.

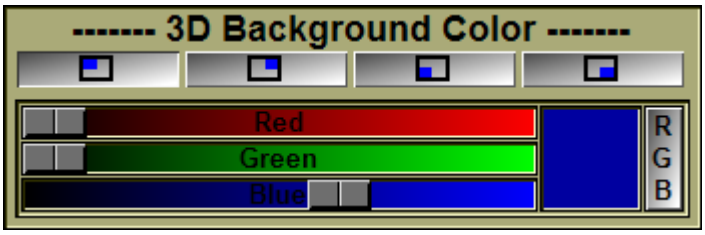
They include:

- **Background**
  This tool is used to control the color of the 3D window background.
- **Lights**
  This tool is used to control the lightning of the 3D geometries.
- **Measures**
  This tool gives you 3D measurements on the geometries present in the program.
- **Overlays**
  This tool is used to display 3D graphic objects such as axis systems or bounding boxes around the geometries..
- **Shading**
  This tool is used to control how the 3D geometries are represented
- **Slices**
  This tool is used too display the 2D frames in 3D space.
- **Smoothing**
  This tool is used to perform surface smoothing on the geometries produced by the "Shell" modules.
- **Transform**
  This tool is used to modify the "transformation Matrix". It enables you to position a camera and manipulate the 3D objects with the mouse.

The 3D Tools:
Background

This tool is used to control the background color. Each of the 4 corners of the background can be controlled individually. The background color is interpolated linearly from the values at the corners.

From the Graphic Interface



Corner Buttons Select one of the 4 corners of the background.

Color Edit You selected corner's color with these controls.

From the Display Area

There is no Display Area interaction specific to this tool.

From the Keyboard

There is no keyboard interface specific to this tool.

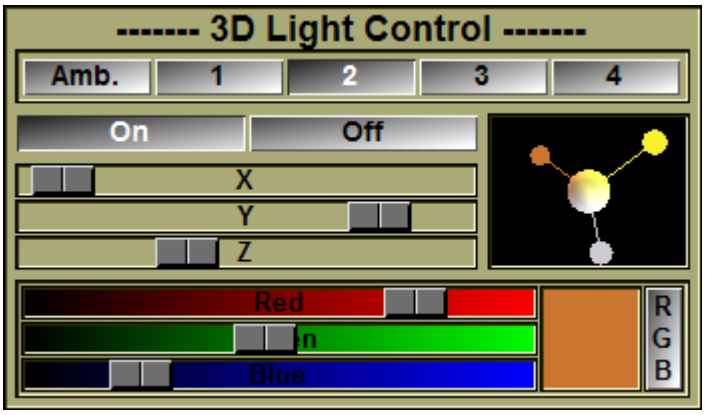
From the Command Line

A list of the commands available for this tool can be found in the "SliceO_Tool\3D Background.dll" section of **Appendix B**.

The 3D Tools:
Lights

The shading of the surfaces is determined by the light sources. This tool enables you to control 4 light sources and an ambient light.

From the Graphic Interface



Light List Select the light source that will be affected by the tool's controls.

On/Off The selected light source can be toggled on or off.

X/Y/Z Sliders Control the selected light direction. The lights are considered infinitely far from the objects. The sliders determine the direction from which the light comes.

Demo Image This demo shows how a centered sphere would be lighted.

Color Edit Change the light's intensity and color.

Note:

When the Ambient light is selected (Amb. Button), the On/Off buttons and the X/Y/Z Sliders are disabled.

From the Display Area

There is no Display Area interaction specific to this tool.

From the Keyboard

There is no keyboard interface specific to this tool.

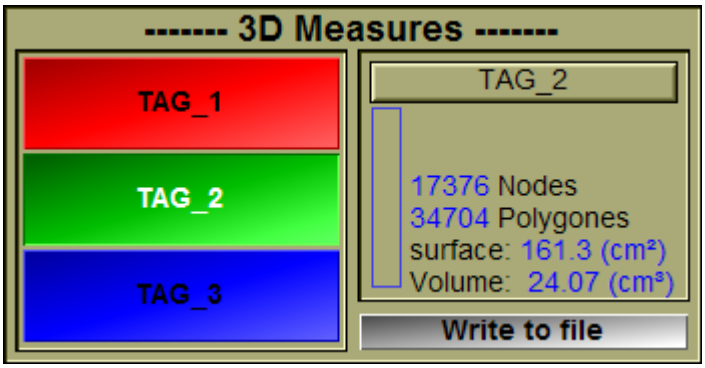
From the Command Line

A list of the commands available for this tool can be found in the "SliceO_Tool\3D Light.dll" section of **Appendix B**.

The 3D Tools:
Measures

This tool enables you to extract some measurements from 3D geometries.

From the Graphic Interface



Geometry List Select the geometry for which you want the measurements.

Results The measurements (number of nodes, number of polygons, surface of the polygons and volume of the geometry) will be displayed here.

Write The geometry's measurements can be written to a file.

From the Display Area

There is no Display Area interaction specific to this tool.

From the Keyboard

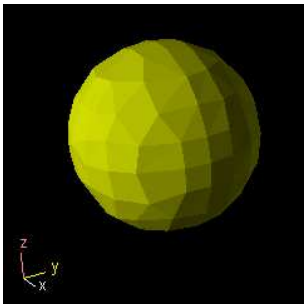
There is no keyboard interface specific to this tool.

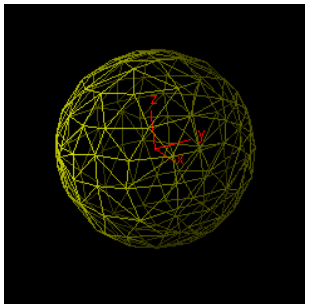
From the Command Line

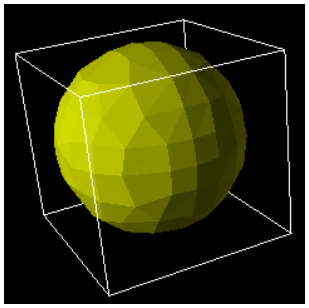
A list of the commands available for this tool can be found in the "SliceO_Tool\3D Measures.dll" section of **Appendix B**.

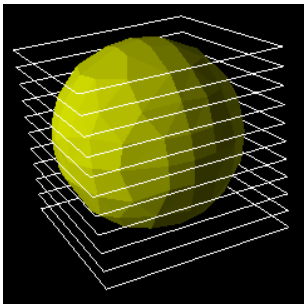
The 3D Tools:
Overlay

3D control boxes and axis information can be overlaid on the image.


Global Axis

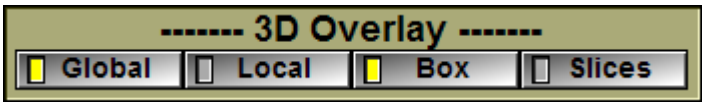

Local Axis


Box


Slices

From the Graphic Interface

There are 4 graphical objects you can display corresponding to the 4 buttons of this tool. The different objects displayed are:



Global	Show the global axis system at the bottom left corner of the window
Local	Show a local axis system at the center of each object
Box	Show the outline of the bounding box for the geometry
Slices	Show the outline of each slice.

From the Display Area

There is no Display Area interaction specific to this tool.

From the Keyboard

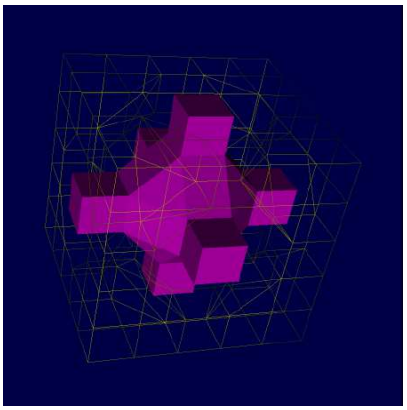
There is no keyboard interface specific to this tool.

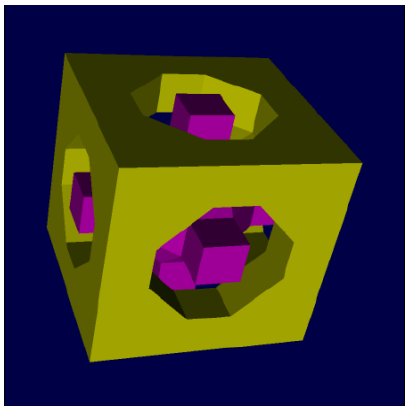
From the Command Line

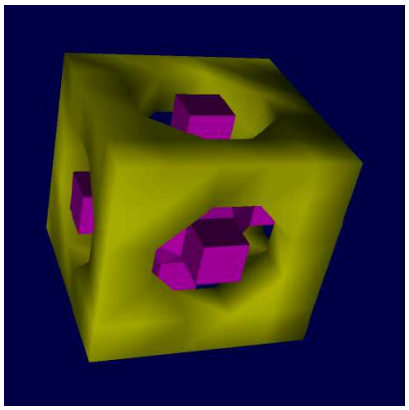
A list of the commands available for this tool can be found in the "SliceO_Tool\3D Overlay.dll" section of **Appendix B**.

The 3D Tools:
Shading

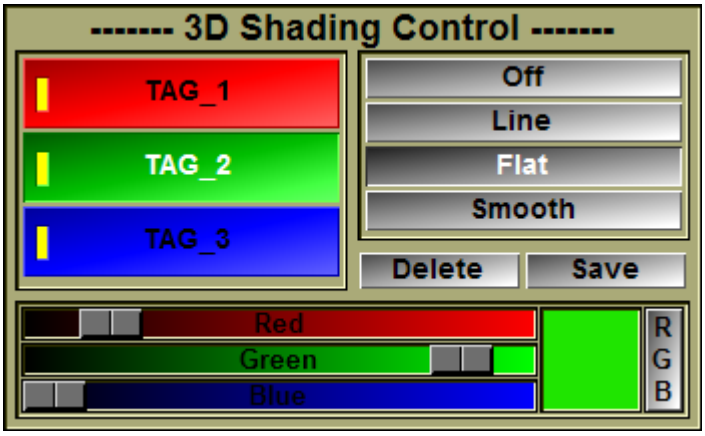
This tool enables you to control the appearance of all the 3D geometries in sliceOmatic. From this tool, you can also delete or save individual geometries.


Line


Flat


Smooth

From the Graphic Interface



Geometry List This list enables you to select the geometry on which the other controls act. The yellow indicator on the surface's button will be lit if the surface is visible (its shading is not "Off").

Shading List

Select the shading you want to apply on the selected geometry. If the geometry is composed of polygons, the choices are:

Off The surface is not shown

Line The surfaces' polygons are drawn as outlines

Flat Each of the surface's polygons is drawn with a color determined by its orientation with the light sources and the surface's color.

Smooth Colors are computed at the polygon's vertices and interpolated on the polygon's surface.

Note:

If the geometry is composed of points or lines, only the first 2 choices are available.

Delete Button

Delete the selected geometry.

Save Button

Save the selected geometry to a file. The surface saving dialog box will appear.

The available formats are:

mov Movie.BYU format. This old polygon format, developed by Brigham Young University for their MOVIE.BYU program in the 80's, is very simple, which makes it ideal to store simple polygonal objects such as those created in sliceOmatic. The format is described in more detail in appendix E. SliceOmatic can create ".dxf" files, but it cannot read them.

dxf Autocad file format. The geometries will be saved using the POLYLINE primitives. A definition of the DXF file format can be found in Wikipedia. SliceOmatic can create ".dxf" files, but it cannot read them.

stl & stl_ascii Stereolithography file format (binary version and ASCII versions). A definition of the STL file format can be found in Wikipedia.

Note:

STL format can only describe 1 object. If multiple objects are selected, sliceOmatic will create 1 file per object.

obj Wavefront OBJ file format. A definition of the OBJ file format can be found in Wikipedia. SliceOmatic can create ".obj" files, but it cannot read them.

wrl A VRML polygon file. A definition of the VRML file format can be found in Wikipedia. SliceOmatic can create ".wrl" files, but it cannot read them.

Color Edit Change the selected geometry's color (if the selected geometry has been recreated from a Tag, these controls will also change the Tag color).

From the Display Area

There is no Display Area interaction specific to this tool.

From the Keyboard

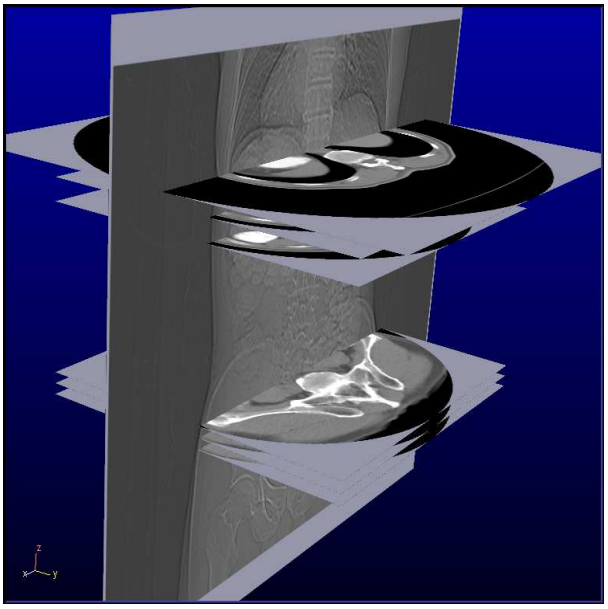
There is no keyboard interface specific to this tool.

From the Command Line

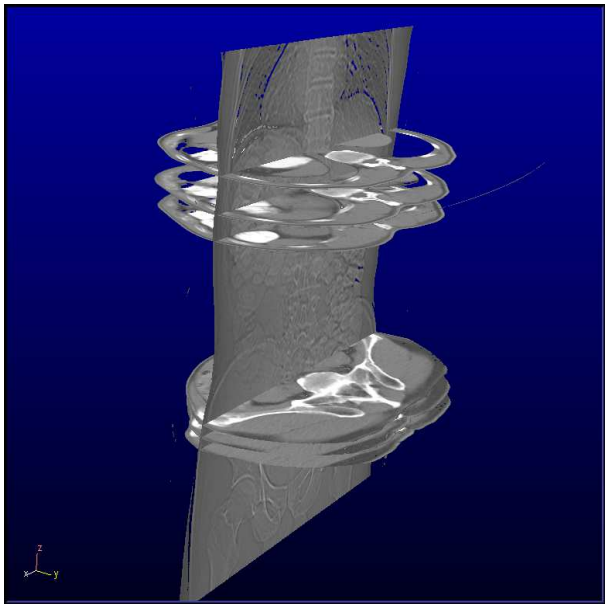
A list of the commands available for this tool can be found in the "SliceO_Tool\3D Shading.dll" section of **Appendix B**.

The 3D Tools:
3D Slices

This tool enables you to display the 2D slices in the 3D window. Only the selected frames will be displayed. You can change this selection with the "2&3D Selection" tool, or the keyboard shortcuts from the "2D Frame" tool.



Transparency: Off



Transparency: On

Note:

You should use this tool in conjunction with the "Selection" tool to select the frames you want to display.

Note:

If you modify the GLI or TAG frames in another window, the 3D slices will automatically be updated also. However this will slow down the segmentation process in all windows.

From the Graphic Interface

ON/OFF Enable/Disable the transparency mode.

Transparency The slider selects the transparency threshold. All pixels whose GLI value is smaller than the threshold will be transparent.

From the Display Area

There is no Display Area interaction specific to this tool.

From the Keyboard

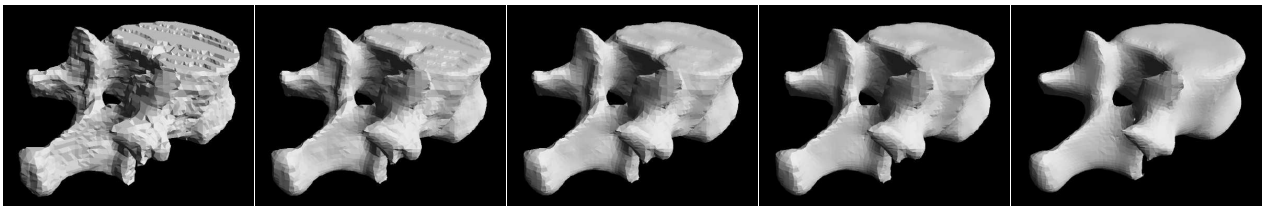
The shortcuts from the "2D Color Scheme" tool will affect the appearance of the slices.
The shortcuts from the "2D Frame" tool will modify the displayed slices.

From the Command Line

A list of the commands available for this tool can be found in the "SliceO_Tool\3D Slices.dll" section of **Appendix B**.

The 3D Tools:
Surface Smoothing

This tool enables you to smooth out any polygonal surface. Smoothing is done by computing displacement vectors that move the nodes of the geometry along the direction of their normal towards the mean value of their immediate neighbors. The nodes are only moved a fraction of the distance towards the mean value of their neighbors. Also, to prevent the surface from shrinking, a different factor is used if the nodes move outward (fact_out) or inward (fact_in). To obtain smoother surfaces, the process can be repeated multiples times (nb_step).



None Small Medium Large Huge

Smoothing does not affect the nodes themselves, instead, a displacement vector is associated with each node. Smoothing is not cumulative and a smoothing of “none” will remove any smoothing.

From the Graphic Interface

None Remove any smoothing done to the surface.

Small to **Huge** Smooth out the surfaces using progressively larger values for the nb_steps parameters. By default the parameters are set to:

	Fact in	Fact out	Nb Steps
Small	0.33	-0.35	5
Medium	0.33	-0.35	10
Large	0.33	-0.35	15
Huge	0.33	-0.35	30

From the Display Area

There is no Display Area interaction specific to this tool.

From the Keyboard

There is no keyboard interface specific to this tool.

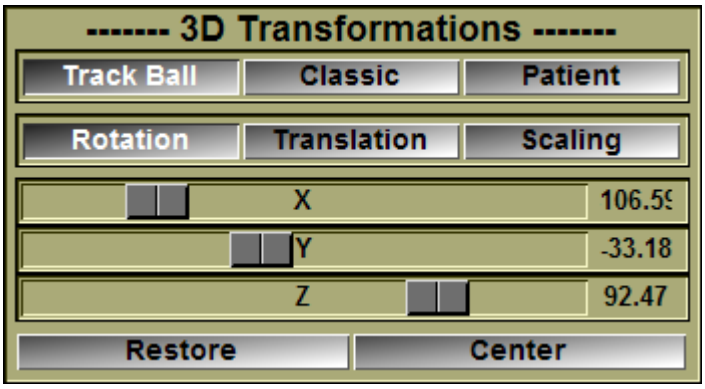
From the Command Line

A list of the commands available for this tool can be found in the "SliceO_Tool\3D Smooth.dll" section of **Appendix B**.

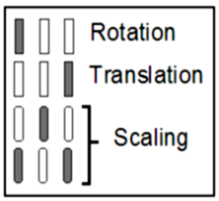
The 3D Tools:
Transform

This tool enables you to have a more precise control on the transformation matrix used to position the 3D Camera. The transformations applied with the mouse in the associated windows are immediately reflected here. Any modification to the matrix done with this tool will immediately affect the geometries in the associated windows.

From the Graphic Interface

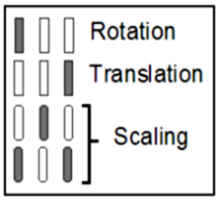


Mode Track Ball



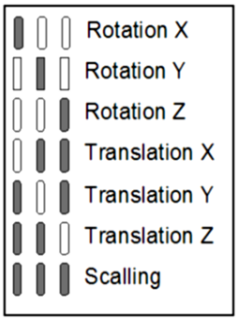
The left mouse button activates the trackball rotations. The mouse controls a point on the surface of a sphere anchored in the center of the 3D window. Moving the mouse drags the point and thus rotates the sphere around its center. The right mouse button activates the translations. The mouse motion drags the objects vertically and horizontally in the plane of the 3D window. The middle mouse button (or both the left and right) activates the scaling. The horizontal mouse motion controls the objects' scaling.

Mode Classic



The left mouse button activates the rotations. The vertical mouse motion will rotate the objects around the windows' horizontal axis. The horizontal mouse motion will rotate the objects around the window's vertical axis. The right mouse button activates the translations. The mouse motion drags the objects vertically and horizontally in the plane of the 3D window. The middle mouse button (or both the left and right) activates the scaling. The horizontal mouse motion controls the objects' scaling.

Mode Patient



In this interface mode, only the horizontal mouse motion is used, and all transformations are done in the patient axis system. Thus, a “z” rotation is a rotation around the patient’s head to foot axis. Each of the mouse buttons controls the rotation of the objects around one of the patient’s axis. The left button activates the “x” axis rotation, the middle button the “y” axis rotation and the right button the “z” axis. Each combination of 2 buttons controls the translation of the objects along one of the patient’s axis. The middle and right buttons activate the “x” translation, the left and right buttons the “y” translation and the left and middle buttons the “z” translation. Pressing all 3 mouse buttons at once activates the scaling.

Rotation The X/Y/Z sliders control the surfaces’ rotation around their axis. The rotations are applied in the following order: X then Y then Z.

Translation The X/Y/Z sliders control the surfaces’ translation along their axis.

Scaling The X slider is used to control the surfaces’ scaling factor.

X/Y/Z Sliders Control the rotation/translation or scaling applied to the transformation matrix.

Restore Reinitialize all transformation matrices and center all currently visible surfaces in the 3D window. (A surface is visible if its shading is either line, flat or smooth.)

Center Center the visible surfaces in the 3D window.

From the Display Area

The mouse is used to manipulate the 3D objects as described earlier.

From the Keyboard

There is no keyboard interface specific to this tool.

From the Command Line

A list of the commands available for this tool can be found in the "SliceO_Tool\3D Transform.dll" section of **Appendix B**.

The Additional Modules

The additional tools and modules are modules that were created for third parties. Placing one of these modules (they have the form of a DLL library) in one of the sliceOmatic directories is all that is needed for them to be included in the program.

These modules are available on the TomoVision web site, in the "Additional Modules" section of the sliceOmatic download page.

Depending on what the module contains, it should go in one of the following sub-directories under the main installation directory:
- SliceO_User_Class_1D to SliceO_User_Class_3D: If the module is a new class.
- SliceO_User_Tools: If the module contains an additional tool.
- SliceO_User_Mode_2D: If the module is working in a 2D window.
- SliceO_User_Mode_3D: If the module is working in a 3D window.

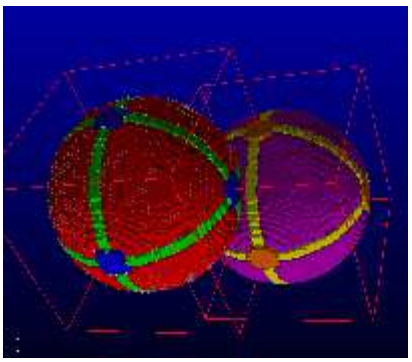
License:

Additional modules will only be loaded in the program if the **sliceOmatic Module** license is present. In demo mode they will not be available.

Additional modules can be protected by sliceOmatic's license system is so desired. If a developer does not want his module to be freely available, he can ask TomoVision for a license code. He will be provided with a code specific to his module and a license generating program to create user licenses for his module. The generated user licenses can then be read by the TomoVision_License program and incorporated in the license dongle.

Additional Modules:
Class: RR2D

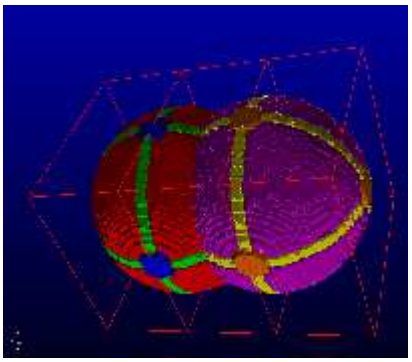
The RR2D module is a new class for sliceOmatic 5.0. This class, "Rigid Registration 2D", enable you to register slices in 2D (in the plane of the slices). It can be used, for example, to re-align series if you need to move a patient between acquisitions. This is a landmark based registration technique. You need to specify landmark points on all the objects that you want to register. the registration is done in 2D only, the slices will only be transformed in their "X-Y" plane.



Unaligned series

The registration will be applied to the children of the class. So if the class is the last before the frames, then a registration transformation will be computed for each frame. If the class is present before a Series class, then a transformation will computed for each complete series.

The first child of the RR2D class is the reference object. All the other objects will be aligned according to that one. To compute a translation, you need at least 1 marker on an object, to compute a rotation or a scale factor you need 2. All the objects are aligned in one operation using least square technique. So the reference points do not need to be present on all objects. As long as a reference point is present on any 2 objects, it will be used in the computation of the solution.

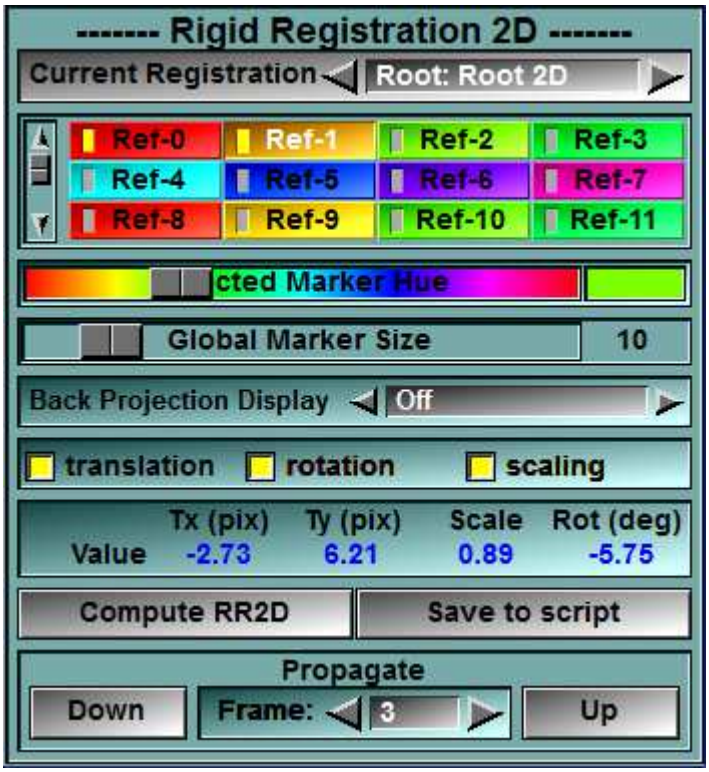


Aligned series

For example, if we want to align 3 objects using translations only, if we have marker #0 on objects 1, markers #0 and #1 on object 2 and marker #1 on object 3, we have enough markers on all objects to align objects 2 and 3 with 1 even though object 3 has no marker in common with object 1.

The RR2D class is enabled through the 2D Mode: "DB Class management" interface. Once you have a RR2D in your database tree, a new "Class Registration 2D" button will be present in the Modes menu. This button will open the module's interface.

From the Graphic Interface



Instance selection Select which of the instances of the class you want to work on.

Reference Points Select the current reference point. The current reference point has its button depressed. If you want to add a new reference point, first select the desired point from this list, then click on the desired position on the frames. Clicking on an existing point on the frames will make that point the current reference point.

A light on each button also indicate the reference points that exist on the current frame.

The mouse wheel can be used to change the current reference point.

Marker Hue You can change the hue of each reference point individually. The changes of hue will be applied to the current reference point.

Marker Size You can change the size of the reference points markers. The size changes are applied to all reference points.

Back Projection Enable the back projection lines. See below for a more detailed description of this function.

Registration Mode You can select the type of transformations that the registration will apply to the frames to align them.

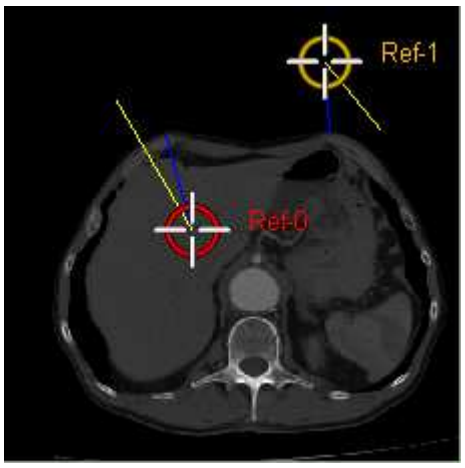
Registration Results Display the results of the registration for the current frame.

Compute Compute the registration for all the children of the class using a least square technique on a matrix of all the markers.

Save to script Create a file that can be used to re-create the same reference points.

Propagate All the measurement tools present on the starting frame will be copied on the target frame. This tool is also described in the "Interface tools" section of the introduction

The Back Projection lines



The back projection lines can be very useful to detect if a marker is misplaced.

It display a line that goes from the actual position of a marker to the compute position of the marker.

If you have multiple markers, then they are all used to compute the registration of an frame. The program then use the registration information to compute where the markers from one frame would be projected on another frame. The program then display a line from that projection position to the actual marker's position. If a marker is way out of place, this will show up as a longer line. You can then use that information to decide if you have misplaced that marker and can correct its position, or maybe remove it altogether.

In the mode "Selected Frame", the marker of the selected frame are projected on all the other frames, and the markers of all other frames projected on the selected frame. Yellow lines connect all the computed / actual marker pairs.

In the mode "All Frames", all markers from all frames are projected on all the other frames. Yellow lines connect markers pairs from markers present on the selected frame, and blu lines connect all the other pairs.

From the Display Area

The registration points are placed directly on the frames. Or if the display is in 3D, on a 3D geometry.

From the Keyboard

The following keys can also be used while the cursor is over a frame:

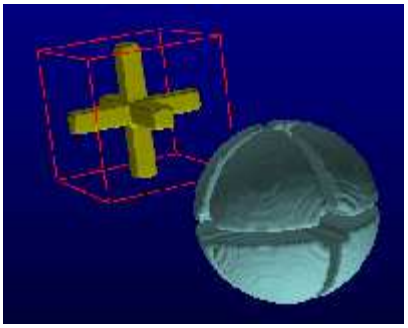
Key	*Action*
Del	Delete the nearest marker.

From the Command Line

There is no command line or variables directly associated with this class.

Additional Modules:
Class: RR3D

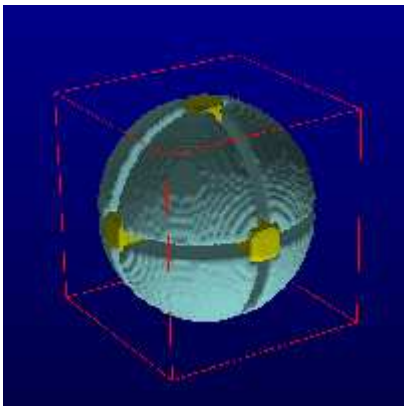
The RR3D module is a new class for sliceOmatic 5.0. This class, "Rigid Registration 3D", enable you to register series of images in 3D. This is a landmark based registration technique. You need to specify landmark points on all the objects that you want to register. the registration is done in 3D, you can allow for translation, rotations and/or scaling.



Unaligned series

The registration will be applied to the children of the class. So if the class is present before a Series class, then a transformation will computed for each complete series.

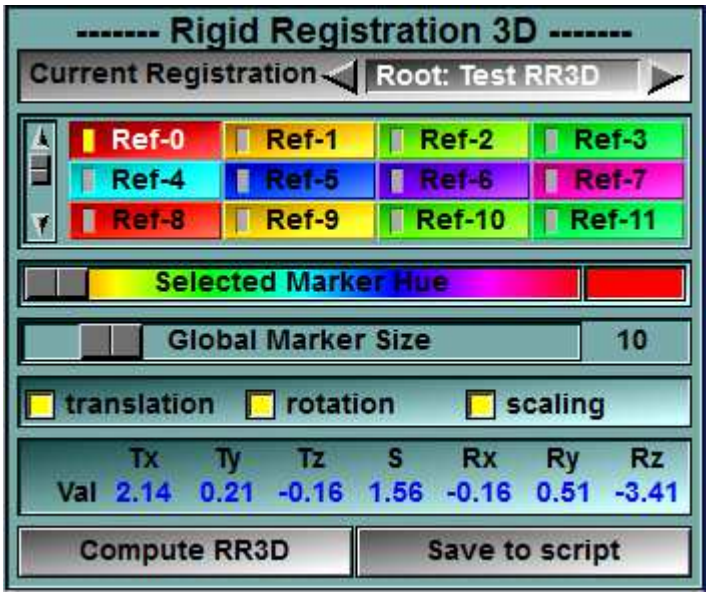
The first child of the RR3D class is the reference object. All the other objects will be aligned according to that one. To compute a translation, you need at least 1 marker on an object, to compute a rotation or a scale factor you need 2. All the objects are aligned must have reference points with the reference object.



Aligned series

The RR3D class is enabled through the 2D Mode: "DB Class management" interface. Once you have a RR3D in your database tree, a new "Class Registration 3D" button will be present in the Modes menu. This button will open the module's interface.

From the Graphic Interface



Instance selection Select which of the instances of the class you want to work on.

Reference Points Select the current reference point. The current reference point has its button depressed. If you want to add a new reference point, first select the desired point from this list, then click on the desired position on the frames. Clicking on an existing point on the frames will make that point the current reference point.

A light on each button also indicate the reference points that exist on the current frame.

The mouse wheel can be used to change the current reference point.

Marker Hue You can change the hue of each reference point individually. The changes of hue will be applied to the current reference point.

Marker Size You can change the size of the reference points markers. The size changes are applied to all reference points.

Registration Mode You can select the type of transformations that the registration will apply to the frames to align them.

Registration Results Display the results of the registration for the current frame. The results include the residual value after the least square computation.

Compute Compute the registration for all the children of the class using a least square technique on a matrix of all the markers.

Save to script Create a file that can be used to re-create the same reference points.

From the Display Area

The registration points are placed directly on the frames. Or if the display is in 3D, on a

3D geometry.

From the Keyboard

The following keys can also be used while the cursor is over a frame:

Key	*Action*
Del	Delete the nearest marker.

From the Command Line

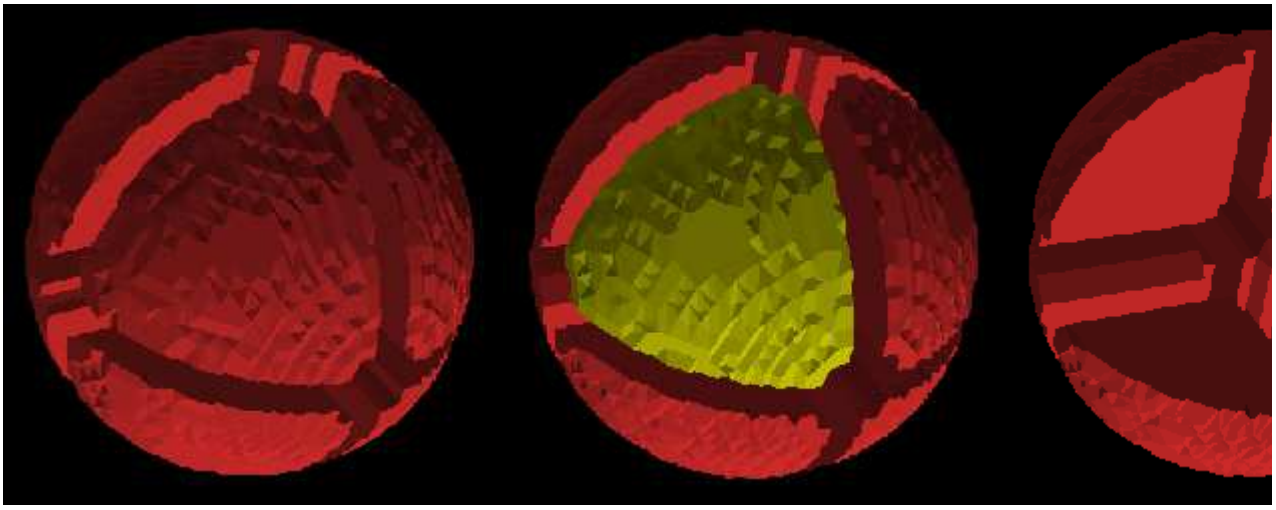
There is no command line or variables directly associated with this class.

Additional Modules:
Tool: 3D Cleanup

This tool enable you to view and delete unconnected 3D surfaces.

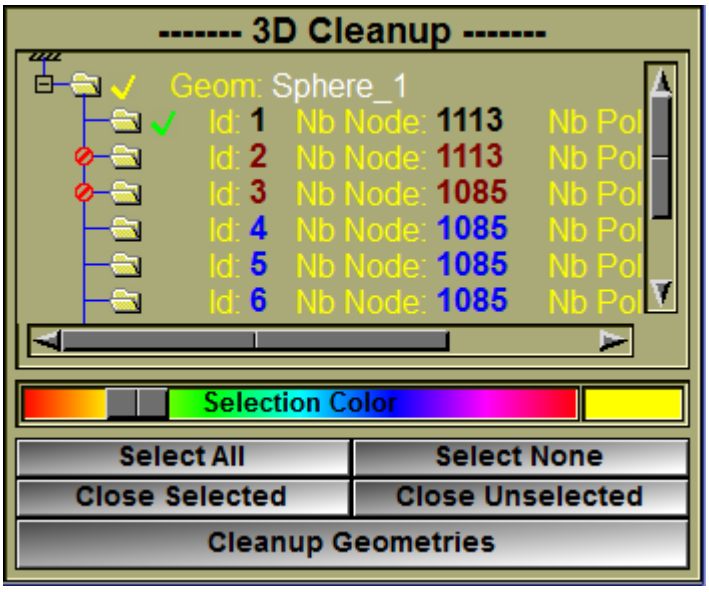
When you create a polygon surface from a 2D dataset, one geometry will be created for each TAG value. for each of these geometries, if the tagged voxels are not neighbours, then you may end up with multiple surfaces that have no contact with one another. Usually most of the small unconnected surfaces are resulting from noise in the TAG dataset.

This tool will present you the list of all unconnected surfaces for each geometry, sorted by decreasing size, and enable you to delete any of these.



8 unconnected surfaces 1 surface selected The surface has l

From the Graphic Interface



Surface list	The interface show you a list of all unconnected surfaces for each of the geometries in the database. For each geometry, the list of unconnected surfaces is sorted by number of polygons, and each element display the number of nodes and polygons of this surface. You can select surfaces by clicking on their lines, you can select all the surfaces in a geometry by clicking on the geometry's line. You can select a range of surfaces either by dragging a selection box, or using the "Shift" and the "Ctrl" keys. The selected surfaces will have a check-mark beside their icons, and they will have a special color in the 3D window.
Selection Color	You can modify the color given to the selected surfaces with the slider.
Select All / None	You can select all the surfaces, or deselect all the surfaces with these buttons.
Close Selected / Unselected	Mark all selected, or all unselected, surfaces for removal. Surfaces marked for removal will have a red slash symbol beside their icon.
Cleanup Geometries	Remove the surfaces marked for "closing" from the geometries.

From the Display Area

There is no Display Area interaction specific to this tool.

From the Keyboard

There is no keyboard interface specific to this class.

From the Command Line

There is no command line for this class.

Additional Modules:
Tool: 3D Displacement Burn-In

This tool enables you to use the displacement information associated with each node to actually change the node's position. The displacement will be added to the node's coordinates.

This is useful if you want to use multiple tools that compute displacement. since only one displacement can be associated with each node at a time, you can use this tool to convert the displacement information in actual change in the node's coordinates.

From the Graphic Interface

Compute Do the actual computation.

From the Display Area

There is no Display Area interaction specific to this tool.

From the Keyboard

There is no keyboard interface specific to this tool.

From the Command Line

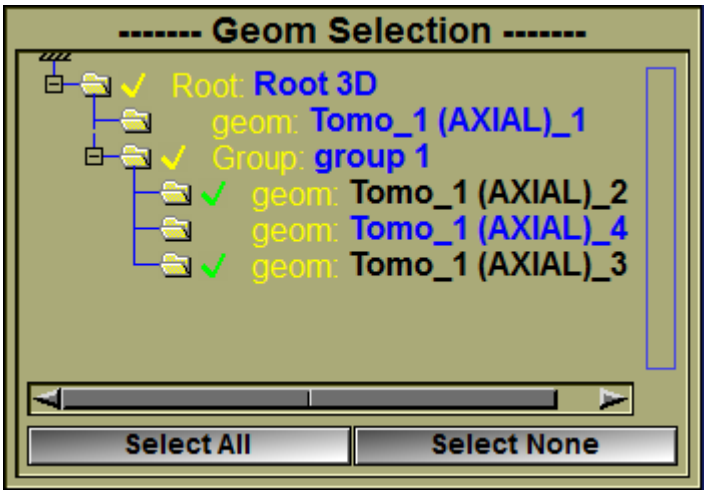
There is no command line or variables associated with this tool.

Additional Modules:
Tool: 3D Geometry Selection

This tool enables you to change the 3D geometry selection. Some tools will act only on the selected geometries. You can use this tool to control the geometries these tools will affect.

Unselected geometries are displayed in wire frame only.

From the Graphic Interface



You can select individual geometries, or complete branches of the database tree just by clicking on their nodes. You can use the "Ctrl" key to toggle the current selection, and you can use the "Shift" key to select a range of geometries. Selected geometrieswill have a green check-mark. Yellow check-marks indicate branches that have some of their geometries selected but not all.

Select All Select all geometries

Select None De-select all geometries

From the Display Area

There is no Display Area interaction specific to this tool.

From the Keyboard

There is no keyboard interface specific to this tool.

From the Command Line

There is no command line or variables associated with this tool.

Additional Modules:
Tool: 3D Relaxation

This tool enable you to get sub-voxel accuracy for some 3D surfaces.

The standard Lorensen 3D reconstruction algorithm create surfaces that are smoother tahn the surfaces you can create with sliceOmatic's version. This is because the standard algorithm create the surfaces directly from the 3D data using a threshold value. If the desired threshold is between 2 voxels, the created surface will go through the interpolated position between the voxels. Since sliceOmatic create its volumes from pre-segmented data, not necessarily created by thresholding, it can not interpolate between voxels.

This tool is intended to be used as a post-processing step. It will create a displacement map, associated with the geometry's nodes, to move the nodes to an interpolated threshold position.

Note:

This tool will only work if there are 3D datasets of voxels associated with the geometries. In other words, the geometries must have been created (with either the "3D Shell Lorensen" or "3D Shell TomoVision" modules) in the same session as the 3D relaxation step. When you save and reload a geometry, the dataset association is lost.

The tool work in 2 steps:

The first step, activated by clicking on the "Compute" button, it will compute the GLI values of the dataset associated at each node of all the selected geometries. It will also compute the GLI values at 2 positions along the normal at that node. The first position is "$RELAX_REACH" voxels in the direction of the normal. This will be the "out" position for "outside" the surface. The second position is "$RELAX_REACH" voxels in the direction inverse to the normal. This will be the "in" (for "inside") position. The tool will then display the ranges of GLI values for each of these positions. These ranges of GLI values will then be used to create a "Threshold" slider in the tool's interface.

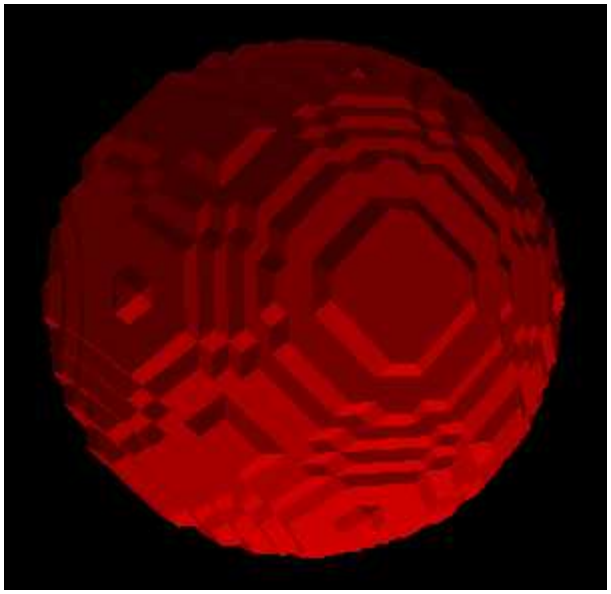
The second step is computed when you select a threshold value from the slider. The tool will compute, using the desired threshold and the "in" and "out" GLI values, a displacement for each node of the selected geometries.

Note:

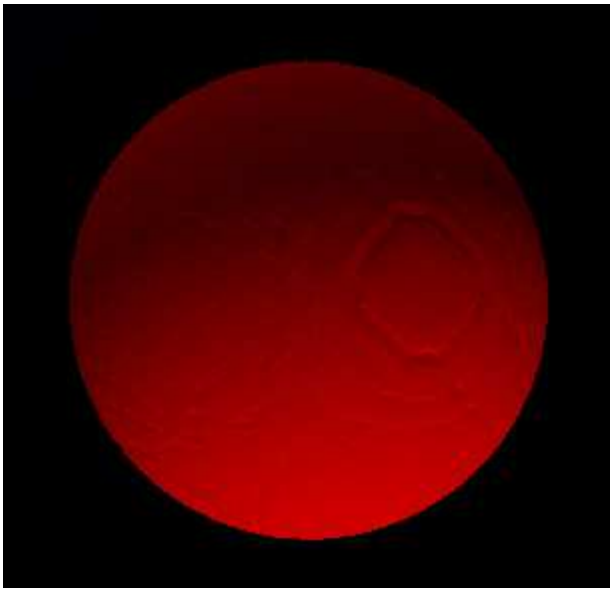
Please note that the "in" and "out" GLI values are computed from a mean value of all the voxels within a range of "$RELAX_RADIUS" voxels of the actual "in" and "out" positions. This enable a certain level of noise filtering.

Note:

It is important to note that this tool, like the "3D Smoothing" tool, compute a displacement that is associated with each node. There is only be one displacement for each node. So if you use both tools, they will overwrite one another's displacement. You will need to use the "3D Displacement Burn-In" tool to actually change the node's positions after computing the displacements of with the first tool before you can use the second tool.



Original Surface



After Relaxation

From the Graphic Interface

Compute This button start the pre-processing computation of the "in" and "out" values needed for the displacement interpolation.

Off This button disable the node displacement. Thus removing any relaxation you may have computed

Threshold changing the threshold value force a re-computation of the relaxation displacements.

From the Display Area

There is no Display Area interaction specific to this tool.

From the Keyboard

There is no keyboard interface specific to this class.

From the Command Line

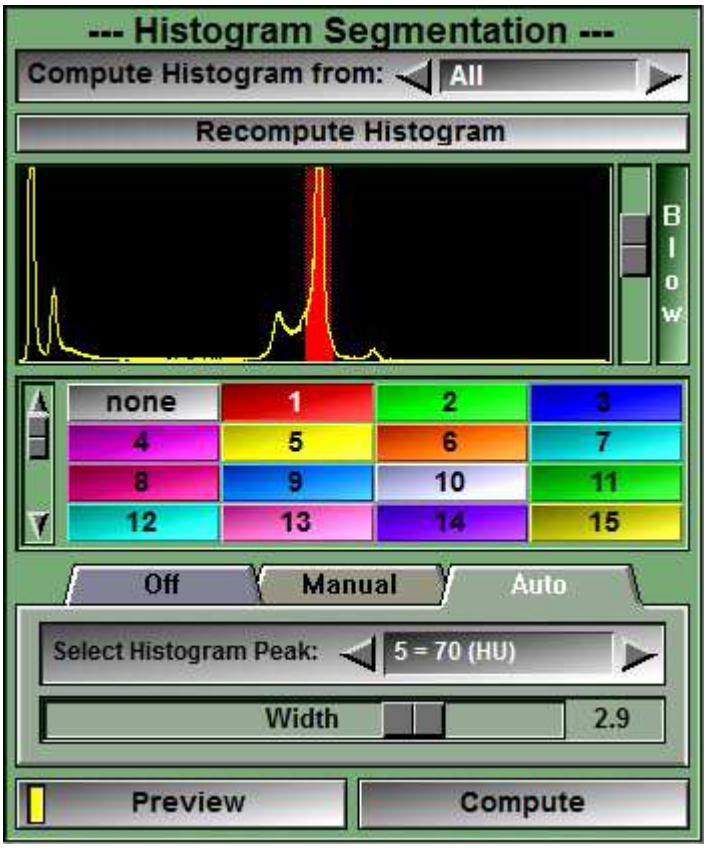
There is no command line for this class.

Additional Modules:
Mode TAG: Histogram Segmnetation

In this mode, you create segmented TAG images from the GLI Images (Grey Level Images).

The segmentation is done by a simple grey level thresholding on the GLI images. The program will attempt to detect Gaussian peaks in the histogram. You can then assign TAG threshold to these peaks.

From the Graphic Interface



Histogram From

By default the histogram is compute from the grey level value of all the pixels of the selected frames. But with this tool, you can restrict the pixels used to compute the histogram to the pixels under one of the existing TAG values in the image.

Recompute Histogram

If you modify the value of the preceding tool ("Compute Histogram from", you need to re-compute the histogram to make these actualize these changes.

Histogram box Display the histogram of the grey values of the GLI image(s). The vertical axis represents the number of pixels having a particular value, while the horizontal axis represents the possible values of the pixels (from Minimum at left to Maximum at right). The colors under the line correspond to the segmentation values fixed by the sliders. When the cursor is over an image, the GLI value of the pixel under the cursor can be seen as a vertical bar on the histogram. This tool is also described in the "Interface tools" section of the introduction

TAG Value Select the current Tag value. This tool is also described in the "Interface tools" section of the introduction

Segmentation Mode For each TAG value, you can select one of 3 segmentation modes:

- Off: The TAG is not used in the segmentation
- Manual: You manually select the lower and higher threshold values for this TAG
- Auto: The threshold values will be centred on one of the histogram peak. the width of the threshold window will be compute from the width of the peak at half it's height. The interface enable you to select the desired peak, and a factor used in computing the width of the threshold window.

Preview Toggle the preview mode. When on, the indicator on the button light up and the TAG displayed in the 2D windows will reflect the threshold selection.

Compute Pressing the **Compute** button will perform the real segmentation and create the TAG images.

From the Display Area

There is no Display Area interaction specific to this tool.

From the Keyboard

In addition to the keyboard shortcuts seen in section 5.2.0.3, the following keyboard keys , Specific to Thresholding, can also be used as a shortcut:

Key map	***Action (While the cursor is over one of the Threshold sliders)***
"Scroll Lock"	Lock or Unlock the slider under the cursor.
“Up Arrow"	Decrease the TAG value associated with the slider under the cursor
“Down Arrow”	Decrease the TAG value associated with the slider under the cursor

From the Command Line

Commands recognized in this module:

Histo: Compute
Compute the segmentation.

Histo: *t_tag* Mode (Off|Manual|Auto)
This will set the mode for all TAGs matching "t_tag".

Histo: *t_tag* Min *value*
This will set the lower threshold values for all TAGs matching "t_tag" if these TAGs are in mode manual.

Histo: *t_tag* Max *value*
This will set the lower threshold values for all TAGs matching "t_tag" if these TAGs are in mode manual.

Histo: *t_tag* Peak *value*
This will assign the peak "value" to all TAGs matching "t_tag" if these TAGs are in mode Auto.

Histo: *t_tag* Width *value*
This will assign the width "value" to all TAGs matching "t_tag" if these TAGs are in mode Auto.

Histo: *t_tag* Slider (*|min|max) (lock|unlock|toggle)
Lock or unlock the sliders for the lower and higher thresholds of the TAGs matching "t_tag"

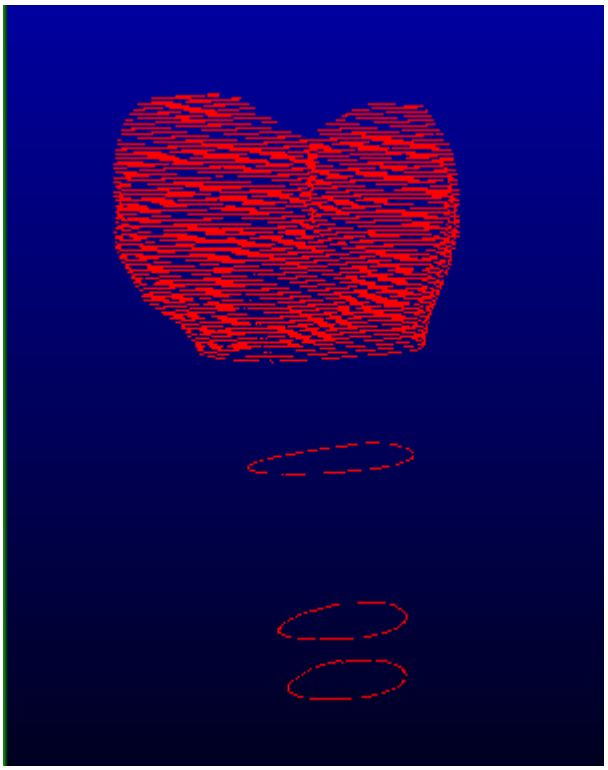
Templates:

t_tag is a template for the TAG values. It match either the TAG ID ("0" to "256") or the TAG label or "cur" for the current TAG value.

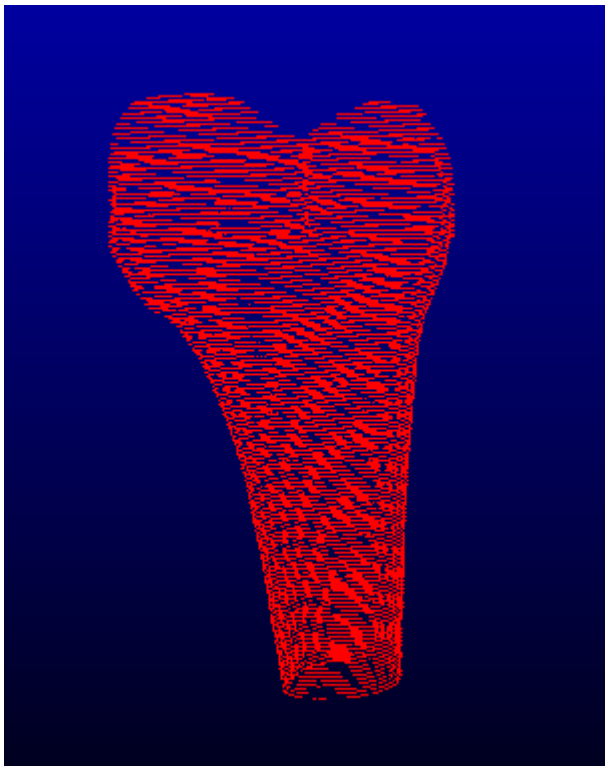
Additional Modules:
Mode TAG: TAG Interpolation

In this mode, you create Tag values on frames that do not have them by interpolating between frames that do.

This mode is useful for 3D structures that do not change much between slices (ex: the long bones of the leg). You only need to segment a few slices and then let this mode interpolate the missing slices from the data on the existing slices.

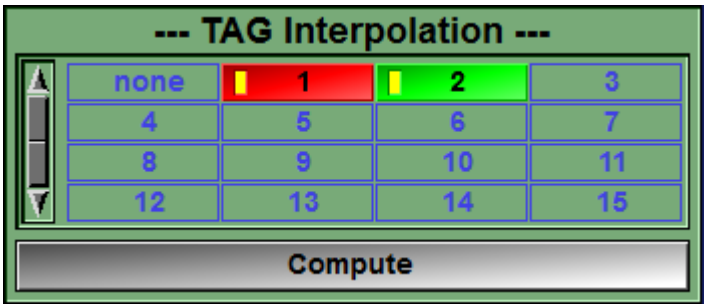


Before interpolation



After Interpolation

From the Graphic Interface



Tag List Select a tag from the list. Only TAG values that exist in the selected frames will have their buttons enabled.

Compute Pressing the **Compute** button will perform the interpolation and create the TAG images.

From the Display Area

There is no Display Area interaction specific to this tool.

From the Keyboard

There is no keyboard interface specific to this class.

From the Command Line

Commands recognized in this module:

Interpol**: Com**pute
Compute the segmentation.

Additional Modules:
Mode 1D: DB Class management 1D

In this mode you can change the composition of the database tree. There are 3 database trees. One each for 1D, 2D and 3D objects. The 1D objects are spectroscopy data, the 2D are images and the 3D are polygon geometries.

This mode enable you to manipulate the composition of the 1D database tree by selecting the 1D classes that will be used in the generation of the tree.

The interface for this mode is similar to the "DB Class Management 2D" mode of "The 2D Modes" section of the manual. The only difference is in the choice of available classes.

Additional Modules:
Mode 1D: DB File Management 1D

In this mode you can change close 1D files that have been opened by the program.

The interface for this mode is similar to the "close" tab of the "DB File Management 2D" mode of "The 2D Modes" section of the manual.

Additional Modules:
Mode 3D: DB Class management 3D

In this mode you can change the composition of the database tree. There are 3 database trees. One each for 1D, 2D and 3D objects. The 1D objects are spectroscopy data, the 2D are images and the 3D are polygon geometries.

This mode enable you to manipulate the composition of the 3D database tree by selecting the 3D classes that will be used in the generation of the tree.

The interface for this mode is similar to the "DB Class Management 2D" mode of "The 2D Modes" section of the manual. The only difference is in the choice of available classes.

Additional Modules:
Mode 3D: DB File management 3D

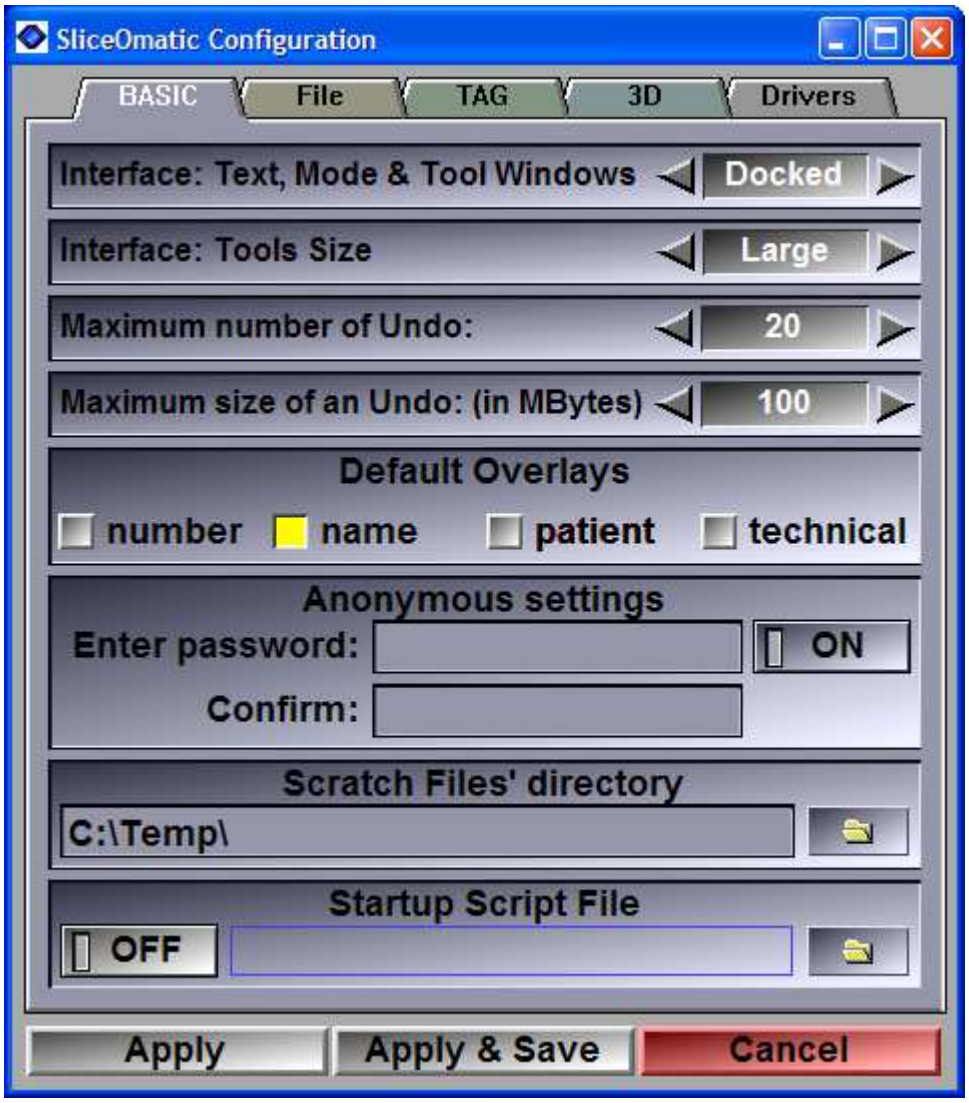
In this mode you can change close 3D files that have been opened and created by the program.

The interface for this mode is similar to the "close" tab of the "DB File Management 2D" mode of "The 2D Modes" section of the manual.

There is a tab for the geometries open by the program ("From File") and geometries created by the program ("From TAG").

The Configuration Menu

When you select the "Config" option from the "File" menu, a window with the configuration menu will pop-up. These configurations can be applied directly to the current session of sliceOmatic, or saved to a "SliceO_ini.scp" file for future sessions.



The TABs You can have access to 5 configuration pages: "Basic", "File", "TAG", "3D" and "Drivers". Each is described in details further down.

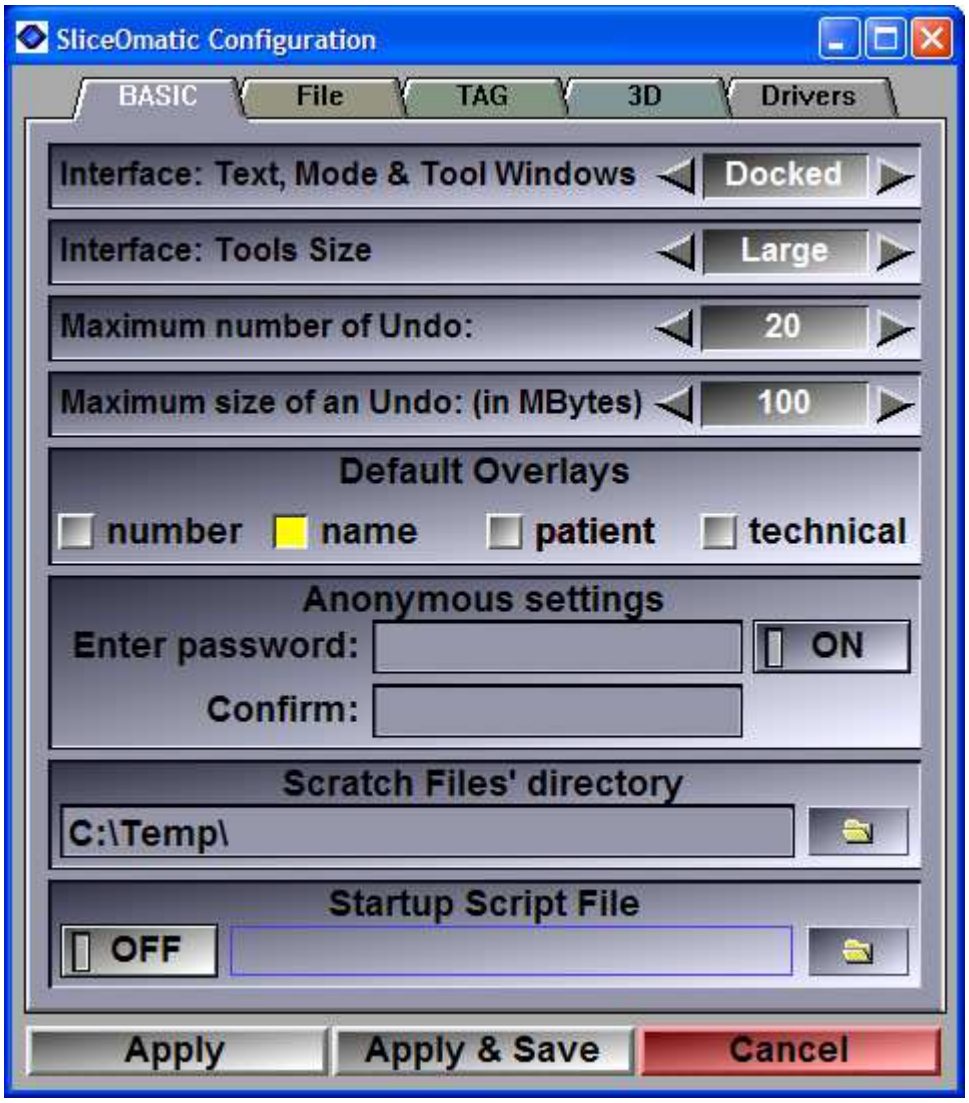
Tab Dependent The content of the configuration page is dependent on the currently activated tab.

Apply Apply the new settings to the current session of sliceOmatic and close the configuration menu.

Apply & Save Apply the new settings to the current session of sliceOmatic, save these values in the sliceO_ini.scp file for the next sessions and close the configuration menu. A description of that file is given in "The SlicO_ini.scp File" in the installation section.

Cancel	Discard all changes and close the configuration menu

Configuration Menu:
Basic



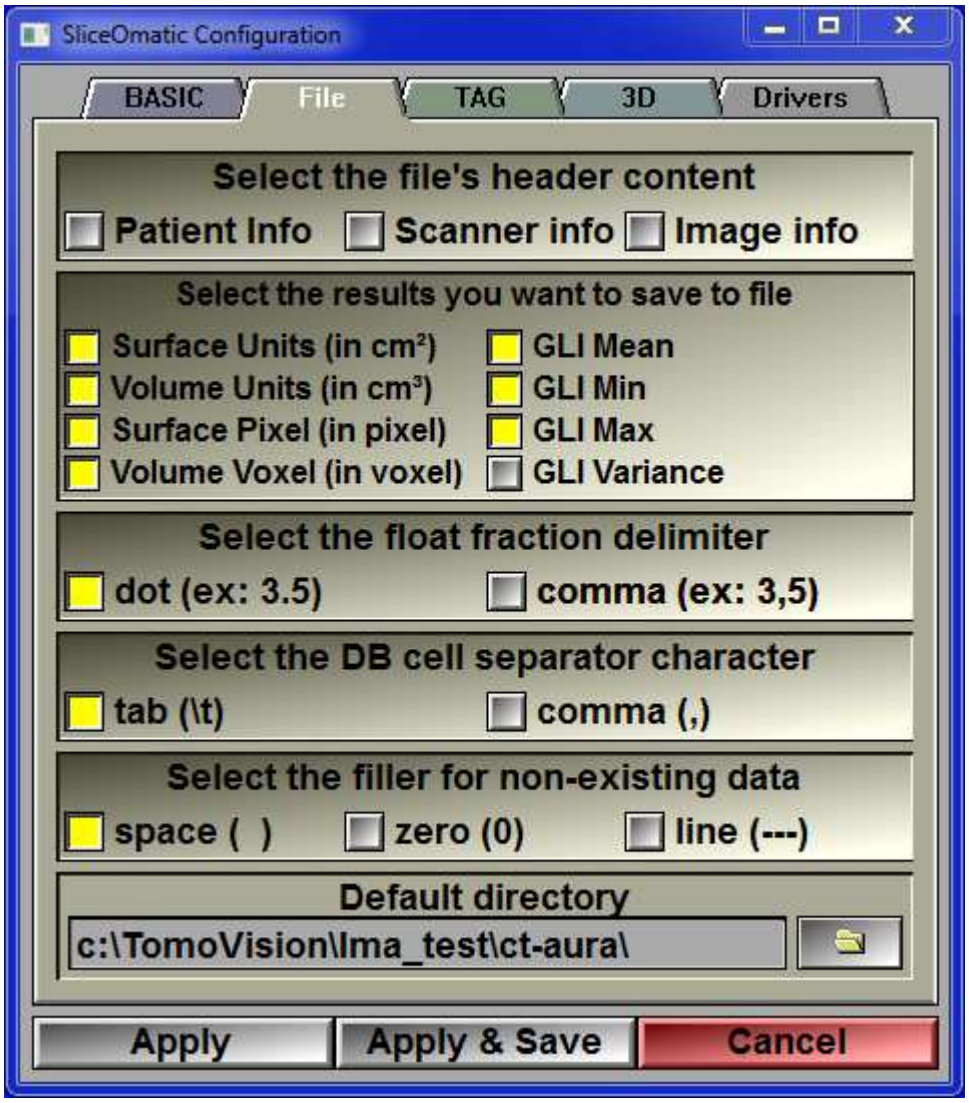
Floating Menus The 3 windows of the control area can be made floating or docked (the default).

Tool Size You can select the size of the interface's tools. If you are using a screen resolution of 1024x768 or less, it may be advantageous to use smaller interface tools. By default the large tools size is selected.

Undo Number You can select the maximum number of Undo that the program keeps in memory. In order to undo most of the operations, the program has to keep a copy of the images in memory. If your system is running low on memory, it might be a good idea to reduce the number of Undos. By default the number of Undo is set to its maximum value: 20.

Undo Size	Each individual Undo operation is limited in size. If an operation causes the associated Undo to be bigger than that size, a choice will be given to the user: cancel the operation, increase the maximum size of an Undo, or forgo the undo for the current operation (this choice will also remove the current Undo for any previous operations).
2D Overlays	The Overlays display information about the image, its patient and the acquisition modality. You can select which of these are displayed by default. You can always change the overlays through the “Overlay Tool”. By default the “Name” overlay is selected.
Anonymous	In many cases of blind studies, it is necessary that the user do not see the patient's name and ID. When the "Anonymous" flag is set, the program will not display any information on the patient. To set the flag, you need to enter a password. The same password will be necessary to remove the flag.
Scratch File	SliceOmatic creates temporary files, error log files and image snapshot files. You can specify in which directory you want these to be created. By default the scratch directory is “C:\Temp”.
Startup Script	You can have sliceOmatic execute a script at startup. This script usually contains some of your preferences such as Tag’s labels, segmentation threshold values, 3D light positions... By default the Startup Script is disabled.

Configuration Menu:
File



File headers

You can place 3 optional headers in the files you created:

The Patient Info header

For each patient used in creating the file we will have (if the information is present in the image's header):

- Patient Name
- Patient ID
- Patient sex
- Date of Birth
- Patient weight (Kg)
- Patient height (m)

The Scanner Info header

For each scanner used in creating the file we will have (if the information is present in the image's header):

- Modality
- Manufacturer
- Model

The Image Info header

For each image used in creating the file we will have (if the information is present in the image's header):

- Resolution X Y
- Pixel Dim. X Y Z (mm)
- Horiz. Dir. X Y Z
- Vert. Dir. X Y Z

Surf/Vol saving You can select which of the available measures from the TAG Surface/Volume you want to save tool in your result files.

Float Fraction You can specify whether you want the fractions in your floating point values to be delimitated by a dot (".") (example:12.34) or a comma (",") (example: 12,34). By default it is the dot.

Cell separator You can specify the character you want to use to mark the end of a cell in your spreadsheet result files. By default it is the "tab" character.

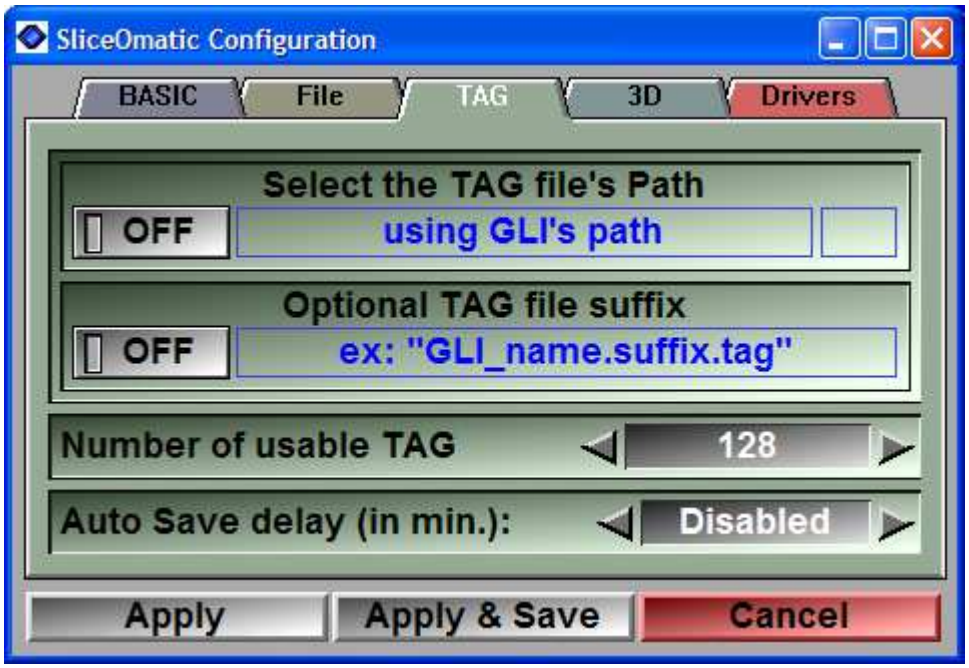
Cell Filler You can specify what should be placed in cells that have no values. For example, in a result file for multiple frames, if a TAG is not present in all the frames, then in some cells we will not have surfaces and volumes for it.

Default Dir. You can specify the default path for the results files. There is only one "default path" parameter in sliceOmatic, so by default, this value is set to the path of the last file read or written.

Note:

Most of the parameters in this page can also be changed in the configuration menu of the Tag_Surface/Volume tool.

Configuration Menu:
TAG



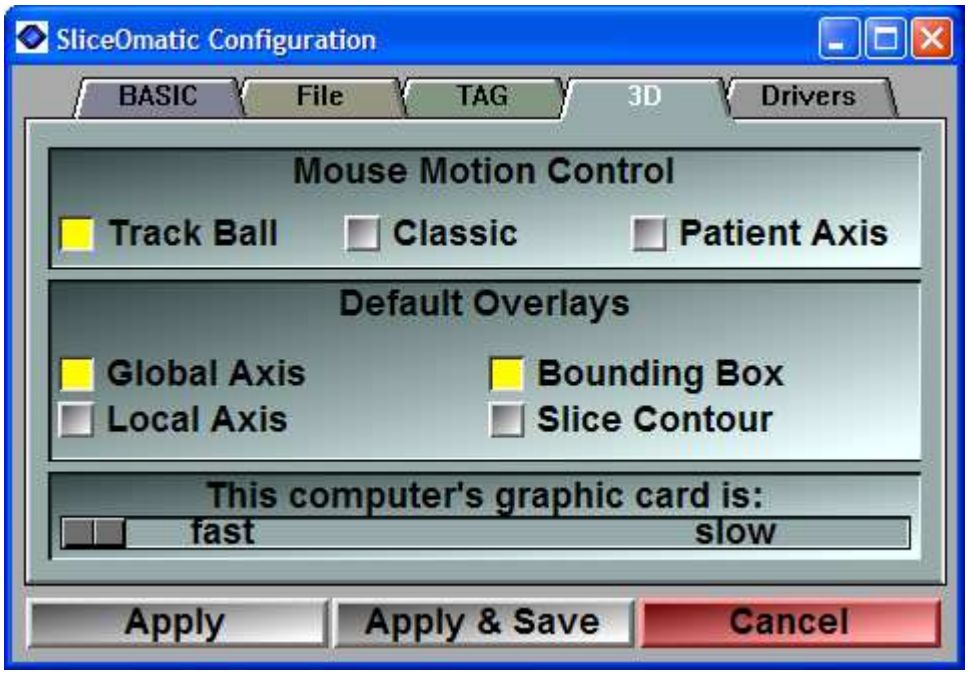
TAG path The Tag files can be placed either with the GLI images, or in a specified directory.

TAG Suffix You can add a suffix after the file name and before the ".tag" extension. The syntax will be: "GLI_name.suffix.tag". So if for example you have the GLI file: "toto.dcm" and you select "demo" as suffix, then the name of the associated TAG file will be: "toto.dcm.demo.tag". The suffix is useful if you want to have multiple users segmenting the same data. You just assign a different suffix to each user, and the can all share the same GLI files.

TAG Number The number of Tag values that you can use in the program can be changed. You have a choice of 16, 32, 64, 128 or 256 Tag values. If you are only using a few Tag values, decreasing the number of usable Tags will make for less crowded interfaces. The default value is 128.

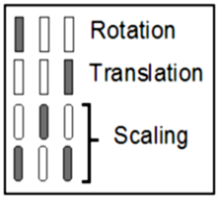
Autosave All modified Tag values will be saved to file automatically at regular intervals. You have a choice of values ranging between 1 and 120 minutes. By default, this option is disabled.

Configuration Menu:
3D

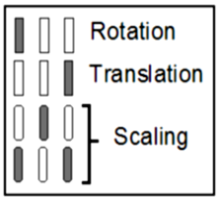


Mouse Motion

In the 3D window, the mouse can be used to perform rotation, translation and scaling of the objects. SliceOmatic gives you a choice of 3 interaction modes:

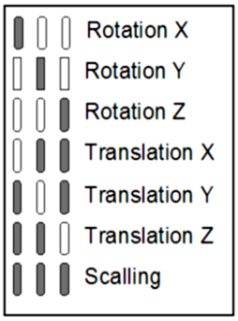


Trackball: The left mouse button activates the trackball rotations. The mouse controls a point on the surface of a sphere anchored in the center of the 3D window. Moving the mouse drags the point and thus rotates the sphere around its center. The right mouse button activates the translations. The mouse motion drags the objects vertically and horizontally in the plane of the 3D window. The middle mouse button (or both the left and right) activates the scaling. The horizontal mouse motion controls the objects' scaling.



Classic: The left mouse button activates the rotations. The vertical mouse motion will rotate the objects around the windows' horizontal axis. The horizontal mouse motion will rotate the objects around the window's vertical axis. The right mouse button activates the translations. The mouse motion drags the objects vertically and horizontally in the plane of the 3D window. The middle mouse button (or both the left and right) activates the scaling. The horizontal mouse motion controls the objects' scaling.

Patient Axis: In this interface mode, only the horizontal mouse motion is used, and all transformations are done in the patient axis system. Thus, a "z" rotation is a rotation around the patient's head to foot axis. Each of the mouse buttons controls the rotation of the objects around one of the patient's axis. The left button activates the "x" axis rotation, the middle button the "y" axis rotation and the right button the "z" axis. Each combination of 2 buttons controls the translation of the objects along one of the patient's axis. The middle and right buttons activate the "x" translation, the left and right buttons the "y" translation and the left and middle buttons the "z" translation. Pressing all 3 mouse buttons at once activates the scaling.

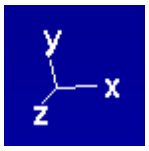


By default the "Trackball" interface is selected.

3D Overlays

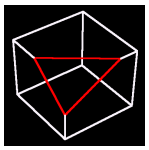
Even though these items are not exactly overlays, they are added on the 3D image and can be turned On or Off. You can also control these items with the "F9" and "F10" keys. By default the "Axis" and "Bounding Box" graphics are enabled.

Global Axis: This enables a 3D axis system in the bottom left corner of the window. The axises are in the patient coordinates system.

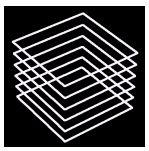


Local Axis: This enables a local 3D axis system in the center of each object.

Bounding Box: The exterior lines of the box containing all the images are drawn in white. If the clip planes intersect this box, the resulting clip contours are drawn in red.



Slice Contour: A white contour line is drawn around each image. This overlay can be combined with the Bounding Box.



Graphic Speed

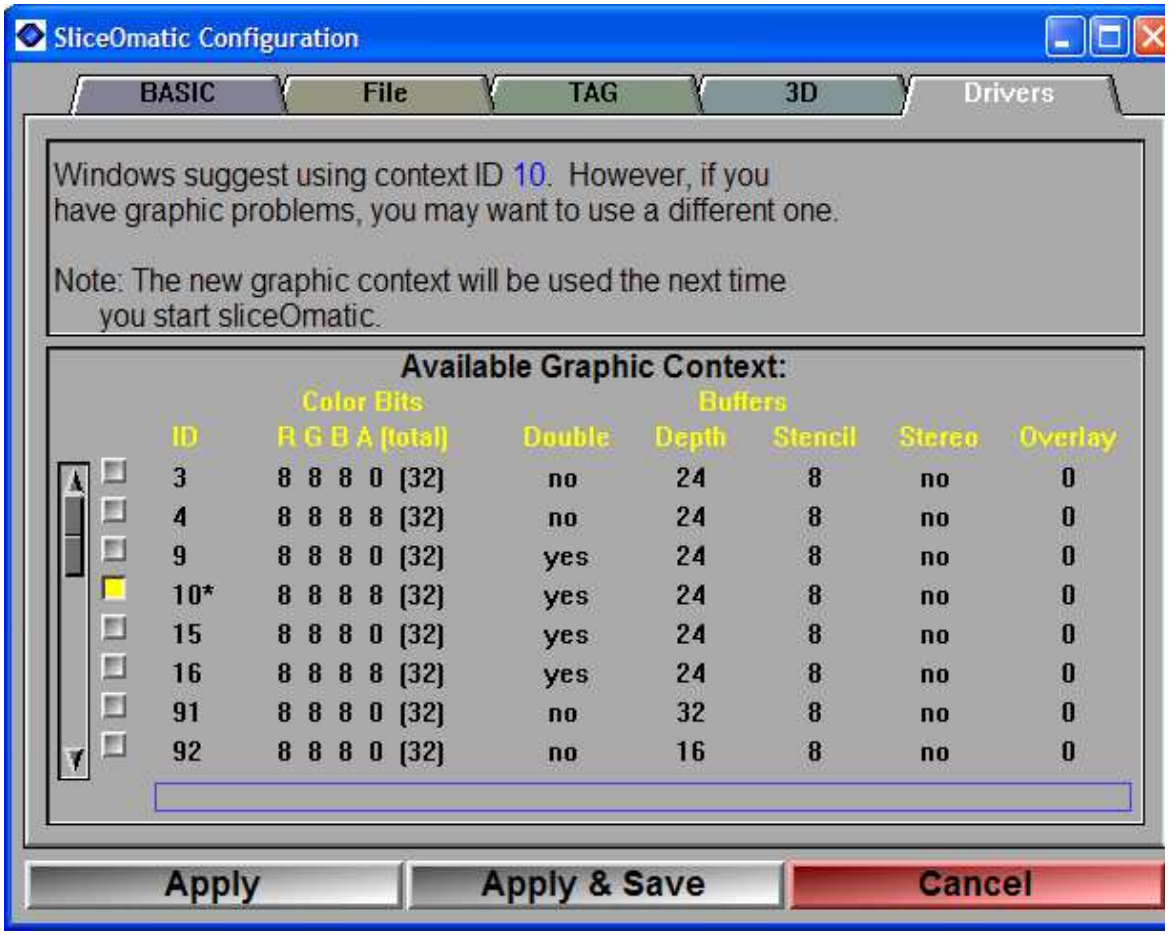
This slider controls the fragmentation of the 3D redraw process. Pushing this slider to the left will force the redraw to be done in multiple small fragments to help the interaction.

Configuration Menu:
Drivers

One of the first things that sliceOmatic does is ask Windows for a graphic context to talk to your graphic card. SliceOmatic specifies its needs (at least 16 bits of colors, a depth buffer, a stencil buffer and preferably a double buffer and an overlay buffer) and Windows returns what it believes is the best context to fulfill these needs. However, sometimes the context returned by Windows is not the best choice.

If you experience graphic problems, it may be due to a bad driver. If this is the case you should:

- Download and install the latest drivers available from the manufacturer of your graphic card.
- If that did not solve the problem, you can select a new context ID from the available contexts presented in the list.



Context ID List This is the list of all the available graphic contexts that meet the minimum requirements asked for by sliceOmatic. The list shows you the characteristics of each context. These are:

ID The context ID number used by Windows to identify this context. The context proposed by Windows is identified with a "*".

Color Bits The number of bits used for each color component: R, G, B and A (the transparency). The total number of bits is shown in parenthesis. SliceOmatic does not really use the □ buffer, so any context that has 8 bits of RGB colors would work.

Double Buffer Presence or absence of a double buffer. The double buffer helps reduce the image flickering when sliceOmatic redraws its screen. Select a context with a double buffer if it is available.

Depth Buffer This buffer is used to compute the back-face elimination in 3D graphics. It is also used for the MIPS computation. The usual values for this buffer are 16, 24 or 32 bits. You should select a context with at least 24 bits of depth buffer.

Stencil Buffer This buffer is used to display information and some tools on top of the images. SliceOmatic needs at least 4 bits of stencil.

Stereo Buffer The stereo buffer is used to create 3D effect with time multiplex liquid crystal glasses (such as “Crystal eyes”). This option is rarely supported by graphic cards, so don’t be disappointed if your card does not have a stereo buffer. You need to select a context with stereo buffer if you want to use the “Shutter” option in 3D viewing. If you do not have a stereo buffer, you will still be able to use the “Red/Blue” option to obtain a 3D effect with anaglyphic glasses.

Overlay Buffer This buffer can also be used to display information on top of the images. It is a much better way of doing it than the stencil buffers. Unfortunately, almost no graphic card supports the overlays anymore. So for the time being, sliceOmatic has stopped using the overlay buffers. The overlay information is still being displayed in the Context ID list, but that’s more for sentimental reason than anything... You do not need any overlay planes to run sliceOmatic.

Note:

If you change the graphic context, your choice will not be used until you restart sliceOmatic.

The Browsers

The Browsers

When reading files, you can select one of 3 TomoVision browsers:

The DICOM Browser
The Medi Browser
The Script Browser

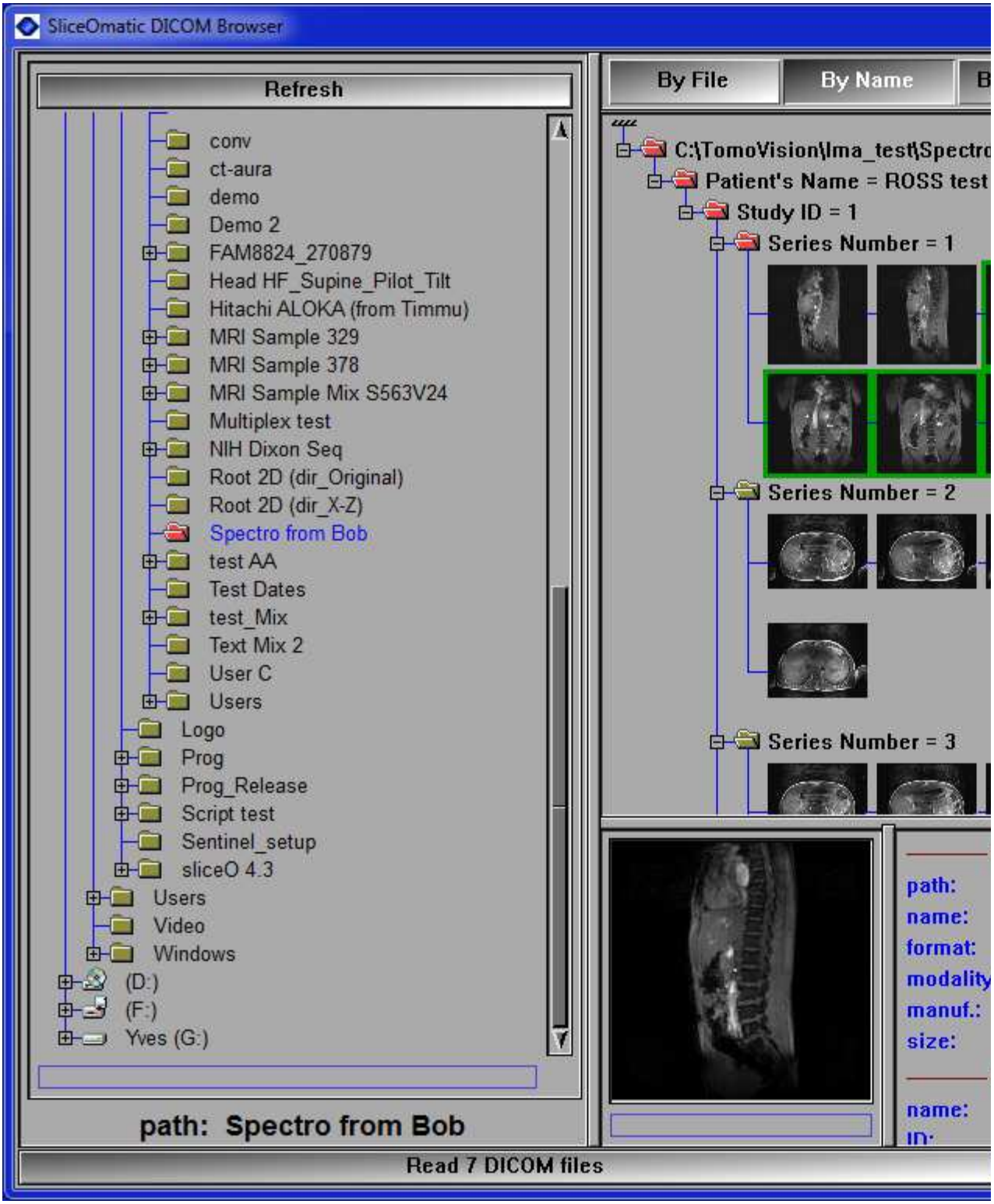
All 3 of these are based on the same program code. The all have a directory selection window, a file selection window, a preview window and a 2 buttons menu.

The Browsers:
The DICOM Browser

The DICOM browser is a sub-set of the Medi Browser. It is restricted to DICOM files. It will be slightly faster than the Medi browser and also use less memory than the Medi browser.

Note:

The DICOM browser will only see DICOM files that use the "Part 5" of the standard. This mean the files that have the string "DICM" at position 128 in the data stream. If you have files that do not have this string (either because they are not DICOM, or are a variation on the DICOM format), use the Medi Browser to locate them.

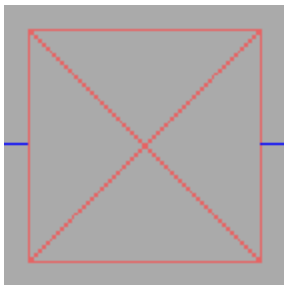


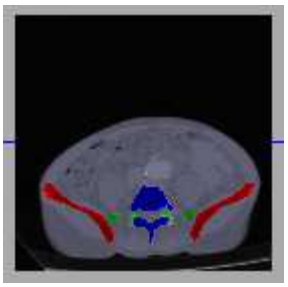
The user select a directory from the directory selection window on the left of the interface. The program will then parse the selected directory and its sub-directories for DICOM files. It will create sorting trees based on the sorting criteria provided by the program for all the DICOM files in the selected directory.

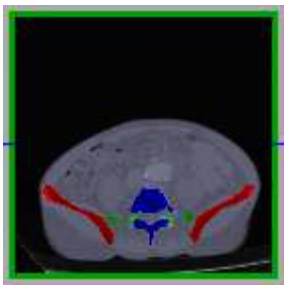
Note:

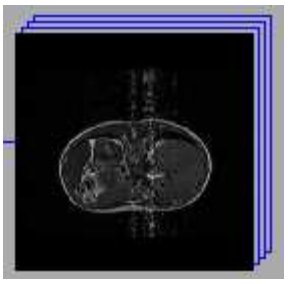
To access files on a remote computer, you first need to map the network drive containing that directory to a drive letter on the local system.

These trees can be viewed in the selection window. An icon will be displayed for each DICOM file in the tree. Icons for the files are read in a background thread of the program. If the file's icon has not yet been read, a red box is displayed instead. If a file contain more than one image, the icon will be of the first image in the file, and the icon will also represent a stack of images.


Icon not loaded yet


Normal Icon


Selected Icon


Multiple images in a file

You can change the default icon size with the "+" and "-" keys. The default icon size is 64x64, but it can be changed from a range of 16x16 to 128x128. Changing the icon size cause the program to dismiss all previously computed icons and re-load and compute them.

Each tree is structured according to different sorting criteria. The list of buttons at the top of the window enable you to display each of these trees.

The "By File" tree display the directory tree. Its root will be the selected directory. It will display all the directory structure under that root, with all the DICOM files inside each of the sub-directories.

All the other sort trees are created using specific tags in the DICOM file. The default trees are defined in the file "Browser_ini.scp" in the installation directory. For each level in the trees, multiple criteria can be used to sort the files. They are listed here in decreasing order of priority. This mean that if the first criteria used to sort 2 files yield the same result, then the second one is examined, etc....

The "By Name" tree has 4 levels:
- The patient name (0010,0010), and the patient sex (0010,0040)
- The study ID (0020,0010) and the image's modality (0008,0060).
- The series number (0020,0011), the series description (0008,103E), the image comment (0020,4000) and the image type (0008,0008).
- The acquisition number (0020,0012), instance number (0020,0013) and position in 3D space.

The "By Patient ID" tree has 4 levels:
- The patient ID (0010,0020)
- The study ID (0020,0010) and the image's modality (0008,0060).
- The series number (0020,0011), the series description (0008,103E), the image comment (0020,4000) and the image type (0008,0008).
- The acquisition number (0020,0012), instance number (0020,0013) and position in 3D space.

The "By Study ID" tree has 3 levels:
- The study ID (0020,0010) and the image's modality (0008,0060).
- The series number (0020,0011), the series description (0008,103E), the image comment (0020,4000) and the image type (0008,0008).

- The acquisition number (0020,0012), instance number (0020,0013) and position in 3D space.

The "By Date" tree has 4 levels:
- The study date (0008,0020), and the series date (0008,0021)
- The study ID (0020,0010) and the image's modality (0008,0060).
- The series number (0020,0011), the series description (0008,103E), the image comment (0020,4000) and the image type (0008,0008).
- The acquisition number (0020,0012), instance number (0020,0013) and position in 3D space.

Files can be selected by clicking on them or their parent icons in the sorting tree. Selected icons will have a green border. Clicking on an icon of a tree branch will select all the files under that branch in the hierarchy. The "Shift" and "Ctrl" keys can also be used when selecting files. If the "Shift"t key is pressed, then all the files between the previous click and the current one will be selected. If the "Ctrl" is pressed, then the file under the cursor will be added to the selection if it was not already selected, or removed from the selection if it was. You can also select multiple images by dragging a selection box over the icons. All the icons that touch the selection box will be selected. If the selection box include the icon for a tree branch, all the files in that branch are selected. If the "Ctrl" key is pressed while you drag a selection box, the files inside the box will be added to the current selection.

From these trees you can select one or multiple files. The selected files will be displayed in the preview window along with a text description of the main parameters of the file (patient name & ID, number of pixels, pixel dimensions...). If multiple files are selected a slider under the preview window will enable you to select which of the selected files will be previewed. If one of the selected file contain multiple images, then the slider can be used to view all the images in the file.

The user can then use the "read x DICOM files " button from the menu to read the selected files in sliceOmatic.

The browser can be closed by pressing the "Cancel" button in the menu or pressing the "Esc" key while the cursor is over the program, or by clicking on the close icon in the title bar.

Note:

If sliceOmatic is in mode Anonymous, the patient's name and ID will not be displayed.

You can change the default sorting trees with a "Browser.scp" file placed in the user's directory (c:\users\user_anme\AppData\Roaming\TomoVision). this file must contain a number of "browser: DICOM" commands. The first one has 2 arguments following the "DICOM" string: the id of the tree and the new title of the tree. The next commands have 3 to 4 arguments giving the id of the tree, the depth of the test and the test itself, either as a parameter name or the 2 element of a DICOM tag. You can multiple sorting parameters for each depth of each trees, they will be used in the order they are provided.

Example:

```
; -------------------------------------
; --- replace the sorting tree #3 -------
; -------------------------------------
browser: DICOM 3 "By Acquisition"
browser: DICOM 3 0 0010 0010 ; (0010,0010) = patient name
browser: DICOM 3 1 0020 0012 ; (0020,0012) = acquisition number
browser: DICOM 3 2 0020 0013 ; (0020,0013) = image number
browser: DICOM 3 2 pos_d     ; image position in 3D space
```

A Browser_DICOM.ini file to replace the "By Study ID" tree

The Browsers:
The Medi Browser

The Medi browser is used to select images to read in sliceOmatic. Contrary to DICOM browser It is not restricted to DICOM files but will recognize all the image format that sliceOmatic can support.

The user select a directory from the directory selection window on the left of the interface. The program will then parse the selected directory and its sub-directories for image files. It will create sorting trees based on the sorting criteria provided by the program for all the image files in the selected directory.

Note:

To access files on a remote computer, you first need to map the network drive containing that directory to a drive letter on the local system.

These trees can be viewed in the selection window. An icon will be displayed for each image file in the tree. Icons for the files are read in a background thread of the program. If the file's icon has not yet been read, a red box is displayed instead. If a file contain more than one image, the icon will be of the first image in the file, and the icon will also represent a stack of images.

You can change the default icon size with the "+" and "-" keys. The default icon size is 64x64, but it can be changed from a range of 16x16 to 128x128. Changing the icon size cause the program to dismiss all previously computed icons and re-load and compute them.

Each tree is structured according to different sorting criteria. The list of buttons at the top of the window enable you to display each of these trees.

The "By File" tree display the directory tree. Its root will be the selected directory. It will display all the directory structure under that root, with all the image files inside each of the sub-directories.

All the other sort trees are created using specific informations in the file. The default trees are defined in the file "Browser_ini.scp" in the installation directory. For each level in the trees, multiple criteria can be used to sort the files. They are listed here in decreasing order of priority. This mean that if the first criteria used to sort 2 files yield the same result, then the second one is examined, etc....

The "By Name" tree has 4 levels:
- The patient name
- The study ID or study number.
- The series number.
- The position in 3D space.

The "By Patient ID" tree has 4 levels:
- The patient ID (0010,0020)
- The study ID or study number.
- The series number.
- The position in 3D space.

The "By Study ID" tree has 3 levels:

- The study ID or study number.
- The series number.
- The position in 3D space.

The "By Data" tree has 4 levels:
- The study date, and the series date
- The study ID or study number.
- The series number.
- The position in 3D space.

Files can be selected by clicking on them or their parent icons in the sorting tree. Selected icons will have a green border. Clicking on an icon of a tree branch will select all the files under that branch in the hierarchy. The "Shift" and "Ctrl" keys can also be used when selecting files. If the "Shift"t key is pressed, then all the files between the previous click and the current one will be selected. If the "Ctrl" is pressed, then the file under the cursor will be added to the selection if it was not already selected, or removed from the selection if it was. You can also select multiple images by dragging a selection box over the icons. All the icons that touch the selection box will be selected. If the selection box include the icon for a tree branch, all the files in that branch are selected. If the "Ctrl" key is pressed while you drag a selection box, the files inside the box will be added to the current selection.

From these trees you can select one or multiple files. The selected files will be displayed in the preview window along with a text description of the main parameters of the file (patient name & ID, number of pixels, pixel dimensions...). If multiple files are selected a slider under the preview window will enable you to select which of the selected files will be previewed. If one of the selected file contain multiple images, then the slider can be used to view all the images in the file.

The user can then use the "read x image files " button from the menu to read the selected files in sliceOmatic.

The browser can be closed by pressing the "Cancel" button in the menu or pressing the "Esc" key while the cursor is over the program, or by clicking on the close icon in the title bar.

Note:

If sliceOmatic is in mode Anonymous, the patient's name, sex and ID will not be displayed.

You can change the default sorting trees with a "Browser_Medi.ini" file placed in the user's directory (c:\users\user_anme\AppData\Roaming\TomoVision). this file must contain a number of "browser: medi" commands. The first one has 2 arguments following the "medi" string: the id of the tree and the new title of the tree. The next commands have 3 arguments giving the id of the tree, the depth of the test and the test itself as a parameter name. You can multiple sorting parameters for each depth of each trees, they will be used in the order they are provided.

Example:

```
; ------------------------------------
; --- replace the sorting tree #3 -------
; ------------------------------------
browser: medi 3 "By Acquisition"
browser: medi 3 0 "patient_name"
browser: medi 3 1 "acquisition_num"
browser: medi 3 2 "image_num"
browser: medi 3 2 pos_d            ; image position in 3D space
```

A Browser_DICOM.ini file to replace the "By Study ID" tree

The Browsers:
The Script Browser

The Script browser is used to select sripts to read in sliceOmatic.

The user select a directory from the directory selection window on the left of the interface. The program will then parse the selected directory and its sub-directories for script files. It will create sorting trees based on the sub-directory of the selected directory.

Note:

To access files on a remote computer, you first need to map the network drive containing that directory to a drive letter on the local system.

The tree can be viewed in the selection window.

Files can be selected by clicking on them or their parent icons in the sorting tree. Selected icons will have a green border. Clicking on an icon of a tree branch will select all the files under that branch in the hierarchy. The "Shift" and "Ctrl" keys can also be used when selecting files. If the "Shift"t key is pressed, then all the files between the previous click and the current one will be selected. If the "Ctrl" is pressed, then the file under the cursor will be added to the selection if it was not already selected, or removed from the selection if it was. You can also select multiple images by dragging a selection box over the icons. All the icons that touch the selection box will be selected. If the selection box include the icon for a tree branch, all the files in that branch are selected. If the "Ctrl" key is pressed while you drag a selection box, the files inside the box will be added to the current selection.

From these trees you can select one or multiple files. The selected files will be displayed in the preview window. If multiple files are selected a slider under the preview window will enable you to select which of the selected files will be previewed.

The user can then use the "read x script files " button from the menu to read the selected files in sliceOmatic.

The browser can be closed by pressing the "Cancel" button in the menu or pressing the "Esc" key while the cursor is over the program, or by clicking on the close icon in the title bar.

Appendices:
A: The Command Line Syntax

The command line is used to control the program. Command lines can be typed in the input line of the text window or placed in a script file and read into the program with the script file mechanism.

Script file have the "**.scp**" extension.

A script file is composed of a number of command lines.

The syntax of the command line is:

Everything on a line following a "#" is a comment and is discarded by the program.

The command lines are used to input statements.

Statements are composed of words, keywords and operators separated from each other by spaces, tabs or end of lines. They MUST be separated from each other to be recognized. If a word contains a space or a special character, it should be protected with double quotes.
ex: `Read: path C:\Program Files\toto` is invalid
`Read: path "C:\Program Files\toto"` is valid

Each statement should be on a line by itself.

Multiple statements can be grouped together with the {} syntax. All the statements between the opening bracket "{" and the closing bracket "}" are considered as 1 statement.
ex: `echo: Bonjour` is 1 statement.

```
{
     Echo: 1
     Echo: 2
     Echo: 3
}
```
is also 1 statement.

The order of operations inside a statement can be changed with the uses of "()".
Example: `$a = 1 + ( 2 * 3 )` is different than `$a = ( 1 + 2 ) * 3`

The variables

Variables can be used anywhere in a script instead of an actual value. The first character of a variable must be "$". A value is assigned to a variable with the "=" character
ex: `$a = Bonjour` will assign the string "Bonjour" to the variable $a
`echo: $a` will echo the string "Bonjour"

Variables values can be either a character string, an integer value, a float value or a vector of values.

ex:
```
$a = Bonjour
$b = 124
$c = 3.1416
$d = 1  2  3.25  126
```

Variables can be used in mathematical expressions. The following operators can be used: "**+**", "**-**", "**/**", "*****". Each operator and its operand must be separated from the others by spaces.

ex:
```
$b = ( $a + 3 ) * 2
```

There are 2 kind of variables: system variables and user variables.

System variables are already defined by the program. They can be used both to access a system value or to change a system value. Some system variables are read-only, and some are only defined under certain conditions.

For example, the variable $TAG is used to represent the TAG value of the pixel under the cursor. This variable is only defined when the cursor is over an image. You can use: "`echo: $TAG`" to get the pixel's TAG value, or "`$TAG = 1`" to assign the value 1 to the TAG value. And "`$TAG = $TAG + 1`" will increment the tag value.

If the user attempts to assign a value to a read-only variable, a warning message will be displayed.

The name of all system variables are in uppercase characters. We suggest that you use lowercase characters for the names of your local variables.

A list of all the defined system variables is given further down.

The macros

A statement can be assigned to a macro. The syntax is:

```
@macro_name = statement
```

The first character of a macro must be "**@**". A macro is executed simply by entering its name.

ex:
```
@toto = echo: Bonjour
```
will assign the statement "echo: Bonjour" to the macro @toto

```
@toto
```
will execute the macro @toto

The commands statements

A command statement is composed of a command keyword followed by its associated values. Command keywords are terminated by a ":" character. The syntax is:

```
keyword:  value_1 [...]
```

The identification of keywords is not case sensitive. The number of values associated with a keyword depends on the keyword.

A list of all the available command statements and their associated templates is given further down.

The control statements

The "for" loop

```
for variable in ( value [...] )
     statement
```

ex:
```
for $a in ( 1 2 3 4 5 )
     echo: $a
```
will echo the successive values taken by the variable $a, these are: 1, 2, 3, 4 and 5

ex:
```
for $IMAGE_CUR in ( $IMAGE_ALL )
     snapshot: c:\temp\snap_$IMAGE_CUR
```

will take a snapshot of the screen for all the images in the current group and name these snapshot C:\temp\snap_1 to c:\temp\snap_n (where n is the last image). In this example, $ALL_IMAGES is a vector of values, and $IMAGE_CUR will successively be assigned each of these values. Each time, the assignment will cause the current image to be changed and the window to be redrawn to reflect these changes. (Also see system variable assignment.)

The "if" statement

```
if ( expression )
     statement
```

If the value of expression is evaluated to a value different than "0" then the test is valid and the statement is executed.

A number of comparison operators are available for the tests: "**==**", "**<**", "**>**", "**<=**" and "**>=**". These operators will be evaluated to "**0**" if the test fails and "**1**" otherwise.

Adding "else" to the "if" statement

```
if ( expression )
     statement1
else
     statement2
```

ex:
```
if ( $a >= 1 )
     echo: a
else
```

```
echo: b
```
will only echo "a" if the value of the variable $a is greater or equal to 1, otherwise it will echo "b".

Keyboard mapping

You can assign predefined operations to a keyboard key with the command statement: "`Key: key_def map action`". The available "actions" are listed after each tool and mode in this manual and in Appendix B: The available keyboard shortcuts.

But also, you can assign a statement or a macro to a keyboard key with the syntax:

`key: key_def macro statement`

ex: `key: a macro flood: $TAG + 1`
will cause the TAG value under the cursor to be incremented when the key "a" is pressed.

Or:

`key: key_def macro macro_name`

ex: `key: b macro @toto`
will cause the macro @toto to be executed when the key "b" is pressed.

The templates

Some commands can accept templates to match one or multiple entities. They use a "wild cards" syntax similar to the one used by UNIX:

? replaces one character
ex: `toto_?` Will match toto_1, toto_a but not toto or toto_12

***** replaces 0 to many characters
ex: `toto_*` Will match toto_1, toto_a, toto_12 but not toto

[abc] replaces one character by a, b or c
ex: `toto_[12]` Will match toto_1, toto_2 but not toto_a or toto_12

[!abc] replaces one character by any character but a, b or c
ex: `toto_[!12]` Will match toto_1 but not toto_1 or toto_2

[a-c] replaces one character by a character from a to c
ex: `toto_[0-9]` Will match toto_1 and toto_2 but not toto_a or toto_12

So if, for example, you to delete all the Tags between 20 and 59 on the presently selected images, you could replace the "**t_tag**" in the following command:

```
Tag: t_tag del
```

with the string "**[2-5]?**"

```
Tag: [2-5]? del
```

The templates used in sliceOmatic's command statements are:

"**t_file**" stands for "file template". It is a template to match the names of the files read in the program (without the path).

"**t_frame**" stands for "frame template". It is a template to match the names of the frames. "*" will match all frames.

"**t_geom**" stands for "geometry template". It is a template to match the names of either the TAG derived geometries (names "1" to "256") or the geometries read from files.

"**t_light**" stands for "light template" The valid light names are: "amb", "1" to "4".

"**t_mask**" stands for "mask template", this template is used to identify one or more geometries in the Geometrical Mask mode.

"**t_merge**" stands for "merge template", this template is used to identify one or more merge values in the Morpho mode. The valid merge values are: "1", "2", "3" and "4".

"**t_point**" stands for "point template", this template is used to identify one or more points.

"**t_tag**" stands for "tag template", this template is used to identify one or more tags. A tag name can be either it's number (from "0" to "256") or its label if one has been associated with the tag through a "tag: label" command. The value "cur" stand for the currently selected TAG value.

"**t_wind**" stands for "window template". A window name is its ID ("1" to "4"). The value "cur" stand for the currently selected window

Appendices:
B: The Variables and Commands

Each DLL can have a set of variables and a set of script commands it recognizes. In this section, all the available variables and commands will be sorted by the DLL they come from.

The Variables

Variables can be used in the command lines instead of values, or you can assign values to the variables.

ex:
```
echo: $TAG_MAX
```
report "256"

and the lines:
```
$TAG_NB = 256
$TAG_NB = $TAG_MAX
tag: number 256
tag: number $TAG_MAX
```
are all equivalent. They will all set the number of accessible TAGs to 256.

Variables start with the character "$".

Variables can be of different types, they can have one value, or multiple (array). The values can be integers, floating point or character strings. In the following lists, we will use the following abbreviations for the variable types:

- **I_8** Signed character (8 bits)
- **I_{16}** Signed short (16 bits)
- **I_{32}** Signed integer (32 bits)
- **I_{64}** Signed long integer (64 bits)
- **U_8** Unsigned character (8 bits)
- **U_{16}** Unsigned short (16 bits)
- **U_{32}** Unsigned integer (32 bits)
- **U_{64}** Unsigned long integer (64 bits)
- **F_{32}** Floating point value 32 bits
- **F_{64}** Floating point value 64 bits
- **S** Character String.
- **V** Vector. A vector has 3 F_{32} components: x y z
- **C** Color. A color is a 32 bits unsigned integer where each component (RGBα) uses 8 bits.
- **A** Array. An array is composed of 1 or more values.
- **R** Read Only. A Read only variable cannot be changed through the command line.
- **P** Pointer. A pointer is an address to an element in the program. Pointers cannot be used or modified through the command lines.
- **U** Unlisted. Unlisted variable do not appear in the list of system variables.

Note:

"**P**" (pointers) and "**U**" (unlisted) variables are listed here mainly for completion's sake. Though they cannot be used in commands, they can be used in user modules.

For each variable, we give:
• The name of the variable in green.
• The abbreviation for the type of the variable (between parenthesis).
• A description of the variable.

The Commands

Please remember that only the first 3 characters of a command and of most of its arguments are used to identify it. To make this clearer, only the necessary first 3 characters of the commands in the following command lines will be in **bold**.
so the command:
interface: **siz**e **smal**l
is equivalent to
int: siz smal

The commands and most arguments (except those in *italics*) are not case sensitive, so
Int: Size SMALL
is equivalent to
int: size small

Appendices: B: Variables and Commands: B: The System DLLs

SliceO_System\3D.dll

System Variables defined in this library:

$GEOM_TAG_NB	(I_{32},R)	Number of potential 3D geometries created from TAG data (through modes such as (Contour Line, or Shell). (= $TAG_NB)
$GEOM_FILE_NB	(I_{32})	Number of 3D geometries not created from TAG (probably imported in sliceOmatic from a 3D format such as MOVIE.BYU).

Commands recognized in this library:

Geom: *t_surf* Write (mov|dxf|stl|stl_ascii|obj) *file_name*

Write all the geometries matching the template "t_surf" to the file "file_name" using the selected file format. The available formats are:

mov Movie.BYU format. This old polygon format, developed by Brigham Young University for their MOVIE.BYU program in the 80's, is very simple, which makes it ideal to store simple polygonal objects such as those created in sliceOmatic. The format is described in more detail in appendix E

dxf Autocad file format. The geometries will be saved using the POLYLINE primitives. A definition of the DXF file format can be found in Wikipedia.

stl Stereolithography file format (binary version and ASCII versions). A definition of the STL file format can be found in Wikipedia.

stl_ascii

Note:
STL format can only describe 1 object. If multiple objects are selected, sliceOmatic will create 1 file per object.

obj Wavefront OBJ file format. A definition of the OBJ file format can be found in Wikipedia.

Geom: *t_surf* Delete

Delete all the geometries matching the template "t_surf".

SliceO_System\A.dll

System Variables defined in this library:

$FRAME_TOTAL	(U_{32})	Total number of frames.
$FRAME_LIST	(U_{32})	List of all the frames (1, ..., n)
$FRAME_OVR	(P)	Pointer to the frame under the mouse (or NULL if the mouse is not over a frame).

$FRAME_CUR	(P)	Array of pointers to the current frames. One entry for each window.
$MOUSE_WND	(P)	Position of the mouse in window, infinite and frame coordinates.
$MOUSE_INF	(P)	
$MOUSE_FRM	(P)	

SliceO_System\Brush.dll

System Variables defined in this library:

$BRUSH_MAX	(U_{16},R)	Maximum number of brushes (Read only)
$BRUSH_NB	(U_{16})	Number of brushes actually used. (Default=$BRUSH_MAX)
$BRUSH_LIST	(A,U_{16})	List of all the brushes (1,...,n)
$BRUSH_CUR	(U_{16})	Index of the current brush. (1 to $BRUSH_NB)
$BRUSH_SIZE	(A,U_{16})	Array of brush sizes, one per brush.
$BRUSH_RADIUS	(U_{16},R)	Maximum brush radius. (Read only)
$BRUSH_BIG	(U_{16})	Size of the big square brush (created with the middle mouse button).

Commands recognized in this library:

Brush: Current *value*
Assign a value to the variable $BRUSH_CUR used to specify the current brush.

SliceO_System\Computer.dll

System Variables defined in this library:

$COMPUTER_NAME	(S,R)	The name of the computer running the program
$COMPUTER_THREAD_MAX	(U_{16},R)	The maximum number of thread that can be run (= Number of CPU + Hyper Threading)
$COMPUTER_THREAD_NB	(U_{16})	The number of concurrent thread that will use in multi-thread operations
$COMPUTER_THREAD_ID	(A,U_{32})	Vector of thread IDs (only valid during multi-thread operations)

SliceO_System\Database.dll

System Variables defined in this library:

$DB_CLASS_NB	(U_{32})	Number of defined classes
$DB_TREE_NB	(U_{32})	Number of classes used in the database tree
$DB_FILE_NB	(U_{32})	Number of files loaded in the program
$DB_FILE_PT	(A,P)	Array of pointers to the loaded files structures
$DB_ROOT_PT	(P)	Pointer to the root class in the database tree

Commands recognized in this library:

Database: List
List all the classes currently available.

Database: Tree [*str_1 ... str_n*]
Re-create the database tree of images using the classes specified. If no class is specified, only the root and frames classes will be used.

Database: Stool
Print out the content of the database tree.

SliceO_System\Document.dll

Commands recognized in this library:

Document: Append *file_name*
Open the document "file_Name" and append information at its end. If no path is specified, the default path (`$DEFAULT_FILE_PATH`) will be used.

Document: Close
Close the document .

Document: Open *file_name*
Open the document "file_Name". If the file was already existing, its content will be erased. If no path is specified, the default path (`$DEFAULT_FILE_PATH`) will be used.

Document: Write *string1 ...*
Write the strings to the opened document .

Note:

The "Document" mechanism can be used to save information from the program's variables.
ex:

```
document: open test.txt
document: write The current brush is: $BRUSH_CUR
document: close
```

SliceO_System\File.dll

Commands recognized in this library:

File: *t_file* Dim *x y z*
Change the pixel dimensions for all the files matching the templates "t_file". If one of the parameters "x", "y" or "z" is "0", then that parameter will be left unchanged.

Note:

The "z" dimension of a pixel is the slice thickness. So, for example, changing the slice thickness of all the images loaded in sliceO to 2mm can be done with:

```
file: * dim 0 0 2
```

File: *t_file* D_H *x y z*

Change image's horizontal direction vector for all the files matching the templates "t_file". If one of the parameters "x", "y" or "z" is "0", then that parameter will be left unchanged.

File: *t_file* D_V *x y z*

Change image's vertical direction vector for all the files matching the templates "t_file". If one of the parameters "x", "y" or "z" is "0", then that parameter will be left unchanged.

File: *t_file* Org *x y z [t]*

Change the image's origin for all the files matching the templates "t_file". If one of the parameters "x", "y" "z" or "t" is "0", then that parameter will be left unchanged.

Templates:

t_file is a template for the file's names

SliceO_System\GLI.dll

System Variables defined in this library:

$GLI_COLOR_COMPONENT	(U_8)	Component used in a color (RGB) image: Max(=0), Red(=1), Green(=2), Blue(=3) or Composite(=4).
$GLI_COLOR_DISPLAY	(U_8)	Red, Green or Blue components are displayed as B&W (=0) or color (=1).

SliceO_System\Interface.dll

System Variables defined in this library:

$INTERFACE_SIZE	(U_8)	Specify the size of all interface tools in the program (0,1 or 2). (Default=2)
$INTERFACE_FLOAT	(U_8)	Specify if the interface windows are docked (=0) or floating (=1) (Default=0)
$INTERFACE_HELP_KEY	(S)	Define which key is associated with the help function. (Default="F12")
$INTERFACE_HELP_INDEX	(S)	Give the path to a local copy of the HTML Index file. If none is given, we use the TomoVision web pages. (Default=none)
$INTERFACE_TOOL_WIDTH	(I_{16})	Give the size (in pixels) of the control area. (Default=350)
$INTERFACE_TEXT_HEIGHT	(I_{16})	Give the minimum height (in pixels) of the text window. (Default=120)
$INTERFACE_MENU_HEIGHT	(I_{16})	Give the height (in pixel) of the menu bar. (Default=26)

$INTERFACE_TITLE_FONT...	Specify the appearance of little text in the interface (FONT_NAME, FONT_SCALE and FONT_WEIGHT)
$INTERFACE_TOOL_FONT...	Specify the appearance of all other text in the interface (FONT_NAME, FONT_SCALE and FONT_WEIGHT)

Commands recognized in this library:

Interface: Size (small|medium|big)
Assign a value to the variable **$INTERFACE_SIZE** used to specify the size of all the tools used in sliceOmatic. This is useful if your screen resolution is insufficient to display all the tools you want at the same time.

SliceO_System\Key.dll

System Variables defined in this library:

$KEY_NAME	(A,P,R,U)	Array of pointers to structure containing the definition of all the available keys on the keyboard
$KEY_CODE	(A,P,R,U)	Array of pointers to structure containing all the remapped keys.

Commands recognized in this library:

Key: List
List all currently mapped keys

Key: *name* Map *script_command*
Associate a command to a specific key.
ex:
```
key: g map "Grid: on toggle"
key: SHIFT_F5 map "overlay: file toggle"
```

Note:
The list of all available key names is provided in Appendix C.

SliceO_System\OpenGL.dll

System Variables defined in this library:

$OPENGL_CONTEXT	(U_{32})	Specify the driver used by the program.
$OPENGL_FLAG	(U_{32})	Bit field of graphic modes.
$OPENGL_SPEED	(U_{32})	Split 3D redraw.
$OPENGL_TEXTURE_SIZE	(U_{32})	Maximum texture size (in pixels, must be a power of 2). (Default=512)
$OPENGL_STATE	(U_{32},U)	Current state of OpenGL

Commands recognized in this library:

OpenGL: Context *value*
Assign a value to the variable `$OPENGL_CONTEXT` used to specify the graphic driver used by the program. (See the Drivers tab of **The Configuration Menu**.)

OpenGL: Flag *value*
Assign a value to the variable `$OPENGL_FLAG`. The flag is a "or" of the different supported values.
The supported values are (in hexadecimal):

- 0x01 = Index mode. TAG images are displayed using color index rather than RGB values.

OpenGL: Speed *value*
Assign a value to the variable `$OPENGL_SPEED` used to split big 3D geometries in multiple parts to keep real-time when manipulating 3D objects. (1 <= value < 100). By default value=1.

SliceO_System\Read.dll

System Variables defined in this library:

$READ_PATH	(S)	Macro for the GLI path when reading
$READ_PATIENT	(S)	Macro for the GLI file name to use
$FILE_TAG_DIR_FLAG	(U_8)	1 if a directory other than the one of the original GLI file is used
$FILE_TAG_DIR_NAME	(S)	Path of the directory where the TAG files are to be saved
$FILE_TAG_SUFFIX_FLAG	(U_8)	1 if a suffix is to be added to the TAG file's name
$FILE_TAG_SUFFIX_NAME	(S)	Suffix to add to the TAG file's name
$DEFAULT_FILE_PATH	(S)	Default file's path
$DEFAULT_ICON_PATH	(S)	Path to the icons used in sliceO
$DEFAULT_USER_PATH	(S)	Path to the users directory
$DEFAULT_PIPE_PATH	(S)	Path to the sliceO_5.dat file
$STARTUP_SCRIPT	(S)	Path to a script to read at startup (default=none)

Commands recognized in this library:

Read: Startup *file_name*
Read a startup script. This command is used internally by the program to read an optional script at startup. (See the Basic tab of **The Configuration Menu**).

Read: Scratch *path_name*
Specify the scratch directory to use to create temporary files. By default, the scratch directory is "c\temp".

Read: Image *file_name* [t=*offset*]
Read an image file in the program. The file can contain one or multiple images (depending on the file format). SliceOmatic can read a number of different file formats.

A detailed list is available on our web site. You can also specify an offset that will be added to the original images positions.

Read: Script *file_name*
Read and execute a script file.

Read: Geom *file_name*
Read a 3D geometry in the program. SliceOmatic can read files in MOVIE.BYU and STL formats.

Read: Patient *patient_name*
Associate the value specified by "patient_name" to the variable `$PATIENT` used to read files

Read: Path *path_name*
Associate the value specified by "path_name" to the variable `$PATH` used to read files

SliceO_System\Regional.dll

System Variables defined in this library:

`$REGIONAL_FRACTION`	(U_8)	Character used to specify how a floating point value is represented. (Default = '.')
`$REGIONAL_SEPARATOR`	(U_8)	Character used to specify the character used to mark the end of a cell in the spreadsheet file format. (Default, = '\t')
`$REGIONAL_FILLER`	(S)	Character used to specify what string should be used to express results that have no values. (Default = " ")

Commands recognized in this library:

Regional: Fraction (.|,)
Assign a value to the variable `$REGIONAL_FRACTION` used to specify how a floating point value is represented. The default value is ".".

Regional: Separator (tab|,)
Assign a value to the variable `$REGIONAL_FRACTION` used to specify the character used to mark the end of a cell in the spreadsheet file format. By default, the "tab" character is used.

Regional: Filler *string*
Assign a value to the variable `$REGIONAL_FRACTION` used to specify what string should be used to express results that have no values. By default the "space" (" ") character is used.

Note:

A description of these parameters can also be found in the File tab of **The Configuration Menu.**

SliceO_System\Select.dll

System Variables defined in this library:

$SELECT_FRM_NB	(U_{16},R)	The number of frames selected in the current window.
$SELECT_GLI_MIN	(F_{32})	The minimum GLI value among the selected frames.
$SELECT_GLI_MAX	(F_{32})	The maximum GLI value among the selected frames.

Commands recognized in this library:

Select: *t_frame* Set (on|off|toggle)
All the frames matching the templates "t_frame" will have their selection flag set t on, off or toggled.

Templates:

t_frame is a template for the frame's names

SliceO_System\Tag.dll

System Variables defined in this library:

$TAG_MAX	(U_{16},R)	Maximum number of TAG. (= 256)
$TAG_NB	(U_{16})	Number of TAG accessible (16 <= $TAG_NB <= $TAG_MAX)
$TAG_CUR	(U_{16})	Current tag value (0 <= $TAG_CUR <= $TAG_NB)
$TAG_ENABLE	(A,U_8)	Array of "Enabled" flags for all tags. Used in 3D
$TAG_LABEL	(A,S)	Array of tag labels.
$TAG_SELECT	(A,U_8)	Array of "Select" flags for all tags
$TAG_RED	(A,U_{16})	Array of red, green and blue colors for all tags (use 8 higher bits)
$TAG_GRN	(A,U_{16})	
$TAG_BLU	(A,U_{16})	
$TAG_ALPHA	(A,F_{32})	Array of transparencies for all tags
$TAG_MIX	(F_{32})	Mix value for the "Mixed" display mode
$TAG_AUTOSAVE	(U_{16})	Delay between auto-saves (0=off)

Commands recognized in this library:

Tag: Current *value*
Assign a value to the variable $TAG_CUR used to specify the current TAG.

Tag: Number *value*
Assign a value to the variable $TAG_NB used to specify the total number of TAG used by the program. Value must be a power of 2 between 16 and 256 (16, 32, 64, 128 or 256). By default the number of TAB available is 128. (See the TAG tab of **The Configuration Menu**.)

Tag: Grab

Assign the TAG value under the cursor to the variable `$TAG_CUR` used to specify the current TAG (the cursor must be over a frame).

Tag: Mix *value*
Assign a value to the variable `$TAG_MIX` used to specify the the ratio of TAG color to GLI color used in the "Mixed" color mode. (See Color Scheme in The 2D Tools section)

Tag: Autosave *value*
Assign a value to the variable `$TAG_AUTOSAVE` used to specify the delay between automatic saving of the TAG images. (See the TAG tab of **The Configuration Menu**.)

Tag: Suffix (off|*string*)
Assign values to the variable `$FILE_TAG_SUFFIX_FLAG` and `$FILE_TAG_SUFFIX_NAME` used to add a suffix to the saved tag file's names. (See the TAG tab of **The Configuration Menu**.)

Tag: Path (off|*path_name*)
Assign values to the variable `$FILE_TAG_DIR_FLAG` and `$FILE_TAG_DIR_NAME` used to specify where the tag files are saved. (See the TAG tab of **The Configuration Menu**.)

Tag: *t_tag* Color *R G B*
Assign a color to a tag value.

Tag: *t_tag* Delete
Delete the tag values matching the template "t_tag" from the selected frames.

Tag: *t_tag* Change *value*
Change the tag value of all tags that match the template "t_tag" to the value "value" for all selected frames.

Tag: *t_tag* Label *string*
Assign the label "string" to all the tags matching the template "t_tag".

Templates:

t_tag is a template for the TAG values. It matches either the TAG ID ("0" to "256") or the TAG label or "cur" for the current TAG value.

SliceO_System\Undo.dll

System Variables defined in this library:

$UNDO_MAX	(U_{16},R)	Maximum number of UNDO. (= 40)
$UNDO_NB	(U_{16})	Current number of possible Undo. (0 <= `$UNDO_NB` <= `$UNDO_MAX`)
$UNDO_SIZE	(U_{32})	Maximum size of each undo (in bytes) (Default=10Mb)
$UNDO_TAG	(A,U_8)	Array of TAGs that are modified by the latest Undo/Redo operation.

Commands recognized in this library:

Undo: Number *value*
Assign a value to the variable `$UNDO_NB` used to specify the maximum number of undo operations kept. (0 <= value <= `$UNDO_MAX`). A value of 0 turns the undo to off.

Undo: Size *value*
Assign a value to the variable `$UNDO_SIZE` used to specify the maximum size of any undo operation. Value is specified in Mbytes. (1<= value <= 100)

Undo: Undo
Perform one "undo" operation.

Undo: Redo
Perform one "redo" operation.

SliceO_System\Variable.dll

Commands recognized in this library:

Variable: List [mode ...]
List the variables that match the "mode" arguments. Mode can be one of: user, system, char, short, int, long, float, double, vector, color, string, array, unsigned, pointer or readonly.

ex:
```
Var: list system unsigned short
```
will list all U_{16} system variables.

Variable: Print "*var_name*"
Report "the name of the variable" = "the value of the variable".

Note:

You must use quotes ("") to protect the variable name, otherwise it will be replaced by its value before the "Variable" command even receives it!

SliceO_System\Write.dll

System Variables defined in this library:

`$WRITE_HEADER` (U_8) Bits field flag for the creation of headers.
- 0x01 = Patient Info
- 0x02 = Scanner Info
- 0x04 = Image Info

Commands recognized in this library:

Write: Header (patient|scanner|image) (on|off)
Enable or disable the creation of optional headers when writing output files. (See the File tab of **The Configuration Menu**.)

Write: TAG [*mode*]
Save all modified TAG files. The "mode" argument is reserved for sliceOmatic's use.

Appendices: B: Variables and Commands: B: The Class DLLs

SliceO_Class\DICOM_Tree.dll

Commands recognized in this library:

Selector**: Dis**criminate ***[t_level] group element***
Set the "Discriminate" parameter for the different levels of the class.

Templates:

t_level is a template for the class level values. ("1", "2"...)

SliceO_Class\Multiplex.dll

Commands recognized in this library:

Multiplex**: Tag [*t_level] channel_id***
Set the "Tag" parameter for the different levels of the class. This parameter is used to select which of the channels will have the associated TAG files. The TAG files are accessible by all channels, but when they are saved, the file name is based on the selected channel.

Multiplex**: Ass**ociate ***[t_level] group element***
Set the "Associate" parameter for the different levels of the class.

Multiplex**: Dis**criminate ***[t_level] group element***
Set the "Discriminate" parameter for the different levels of the class.

Templates:

t_level is a template for the class level values. ("1", "2"...)

Appendices: B: Variables and Commands: B: The Tool DLLs

SliceO_Tool\2&3D Memory.dll

Memory: Clear (*|images|undo|geom)
Clear the memory buffers for one (or all) of the categories.

SliceO_Tool\2&3D Point.dll

System Variables defined in this library:

$POINT_NB	(U_{32})	Total number of points.
$POINT_ON	(U_8)	Bit field to select which class of points are visible.
$POINT_LABEL	(U_8)	The labels are visible (=1) or not (=0). (Default = 1)
$POINT_CUR	(P)	Pointer to the current point
$POINT_PT	(P)	Array of pointers to the points.

Commands recognized in this library:

Point: Label (on|off|toggle)
Make the point's label visible or not.

Point: *t_point* Color *R G B*
Change the color of the points matching the template t_point.

Point: *t_point* Delete
Delete the points matching the template t_point.

Point: *t_point* Position *x y z*
Change the position (3D) of the points matching the template t_point.

Point: *t_point* Size *value*
Change the size of the points matching the template t_point.

Point: *t_point* Save *file_name*
Save the points matching the template t_point to a script file.

Templates:

t_point is a template for the point names

SliceO_Tool\2&3D Snapshot.dll

System Variables defined in this library:

$SNAPSHOT_FORMAT_NB	(U_8,R)	Number of formats supported (=5).
$SNAPSHOT_FORMAT_CUR	(U_8)	Format of the snapshot files (0=TIFF, 1=PNG, 2=JPE

$SNAPSHOT_QUALITY ($U_8$) Compression quality (1=poor, 100=best)

Commands recognized in this library:

Snapshot: Format (TIFF,PNG,JPEG,GIF,BMP|TARGA)
assign a value to the $SNAPSHOT_FORMAT variable that control the format used when creating the snapshot files. Supported formats are:

- TIFF ($SNAPSHOT_FORMAT=0)
- PNG ($SNAPSHOT_FORMAT=1)
- JPEG ($SNAPSHOT_FORMAT=2)
- GIF ($SNAPSHOT_FORMAT=3)
- BMP ($SNAPSHOT_FORMAT=4)
- TARGA ($SNAPSHOT_FORMAT=5)

Snapshot: [t_window] Click [file_name]
Create a "Snapshot" of a specific window (specified by the t_window template) or the complete interface (if no template is specified) and save it to the file "file_name". If "file_name" is not specified, the file will be save to the scratch directory using the name "sliceO_xx.yyy". Where xx is an increasing number from 00 to 99 and yy is the extension for the selected file format (".tif", ".png", ".jpg", ".gif", ".bmp" or ".tga").

Templates:

t_window is a template for the window. It match either the widow ID ("1" to "4") or "cur" for the currently selected window.

SliceO_Tool\2&3D Window.dll

System Variables defined in this library:

$WINDOW_MAX	(U_{32},R)	Maximum number of windows (=10)
$WINDOW_NB	(U_{32})	Current number of windows (1 <= $WINDOW_NB <= $WINDOW_MAX)
$WINDOW_LIST	(U_{32})	List of all windows (1, 2, ..., n)
$POINT_CUR	(P)	Pointer to the current window
$POINT_OVR	(P)	Pointer to the window under the cursor
$POINT_PT	(P)	Array of pointers to the windows.

Commands recognized in this library:

Window: Config (split|floating|free)
Specify the window's modes (at this time only "split" is supported):

- Split: the Display Area is split in 4 windows
- Floating: you can have up to $WINDOW_MAX windows. They are floating but limited to the Display Area.
- Free: you can have up to $WINDOW_MAX windows. They are floating and are not limited to the Display Area.

Window: Split *horizontal vertical*
Specify the vertical and horizontal divider in the Display Area. This will reformat the 4

windows.

Window: *t_window* Mode (one|all|toggle)
Specify the current mode of the windows matching the template "t_window"

Window: *t_window* Color R G B
Specify the color of the borders of windows matching the template "t_window"

Window: *t_window* State (one|off)
Specify the current state of the windows matching the template "t_window"

Window: *t_window* Position *left top width height*
Specify the position of the windows matching the template "t_window"

Window: *t_window* Sliders *horizontal vertical*
Specify the value of the horizontal and vertical sliders of the windows matching the template "t_window"

Templates:

t_window is a template for the window. It match either the widow ID ("1" to "4") or "cur" for the currently selected window.

SliceO_Tool\2D Color_Map.dll

System Variables defined in this library:

`$COLORMAP_GLI_INVALID`	(C)	Displayed color for invalid GLI pixel
`$COLORMAP_GLI_FILL`	(C)	Displayed color for GLI pixel outside the FOV
`$COLORMAP_GLI_CUR`	(U_{16})	Currently used colormap (Default=0)
`$COLORMAP_GLI_NB`	(U_{16})	Number of defined colormaps (Default=5)
`$COLORMAP_GLI_SIZE`	(U_{16})	Size of each colormap (Default=4096)
`$COLORMAP_GLI_PT`	(P)	Array of pointers to the GLI colormaps.

SliceO_Tool\2D Color_Scheme.dll

System Variables defined in this library:

`$COLOR_SCHEME`	(U_8)	Curently selected color scheme (0=Grey, 1=Mixed, 2=Tint, 3=Over, 4=Tag)
`$COLOR_VAL_MIN`	(F_{32})	"Black" slider;s value
`$COLOR_VAL_MAX`	(F_{32})	"White" slider's value
`$COLOR_CLASIC`	(U_8)	0=new style, 1=old style of "Mix" mode

Commands recognized in this library:

Color: Grey

Assign the value 0 (=Grey) to the $COLOR_SCHEME variable.

Color: Mix [*value*]
Assign the value 1 (=Mixed) to the $COLOR_SCHEME variable. If "value" is present, assign its value to the variable $TAG_MIX that control the mix of GLI and TAG colors The accepted range
for "value" is from 0.1 to 0.9.

Color: Over
Assign the value 2 (=Over) to the $COLOR_SCHEME variable.

Color: Tag
Assign the value 2 (=TAG) to the $COLOR_SCHEME variable.

Color: Black *value*
Assign the value "value" to the $COLOR_VAL_MIN variable that control the color of the GLI images. (see Color Scheme in **The 2D Tools**)

Color: White *value*
Assign the value "value" to the $COLOR_VAL_MAX variable that control the color of the GLI images. (see Color Scheme in **The 2D Tools**)

SliceO_Tool\2D Debug.dll

Commands recognized in this library:

Debug: Database
List the content of the database classes (this is equivalent to the "Database: List" command)

Debug: Filter (on|off|toggle)
Change the displayed GLI image from the normal image to the "Filtered" image. The filtered image is used in some modes such as Morpho and Snakes. It is the first step in the computation of the image's gradients. Before the gradient are computed a 3x3 smoothing filter is applied on the images.

Debug: Gradient (amp|x|y|z) (on|off|toggle)
Change the displayed GLI image from the normal image to the "Gradient" image. The gradient image is used in some modes such as Morpho and Snakes. You can display either the intensity of the gradient, or the amplitude of one of the direction component of the direction of the gradient.

Debug: WS (on|off|toggle)
Change the displayed TAG image from the normal image to the Watreshed "WS" image. The watershed is compute in the Morpho mode.

Debug: Merge (1|2|3|4) (on|off|toggle)
Change the displayed TAG image from the normal image to the one of the Morpho merge images.

SliceO_Tool\2D Frame.dll

System Variables defined in this library:

$FRAME_SCALING	(F_{32})	Scaling factor to apply when increasing or decreasing images (Default = 1.5)

Commands recognized in this library:

Frame: Scale (*value*|inc|dec)
Change the scaling factor for all the selected frames

Frame: Current (*value*|inc|dec)
Change the current frame by selecting one of the children of the current parent of the current frame.

Frame: Offset Pos *x y z*
Offset the position of the current frame.

Frame: Offset d *value*
Offset the position of the current frame by adding an offset value in the "z" direction.

Frame: Offset t *value*
Change the time value of the current frame.

Frame: Parent (*value*|inc|dec)
Frame the current frame by making one of the parent class the current class.

Frame: Frame *value*
Change the current frame by selecting one of the frames of all the available frames. This command enable to go through all the frames in the DB.
example:

```
for $cur in ( $FRAME_LIST ) {
      echo: Frame -> $cur
      frame: frame $cur
}
```

SliceO_Tool\2D Gamma.dll

System Variables defined in this library:

$GAMMA_CUR	(F_{32})	Curent Gamma value (Default = 1.0)
$GAMMA_NB	(U_{16},R)	Number on entries in the Gamma table (= 256)
$GAMMA_ALPHA	(A,F_{32})	Array of `$GAMMA_NB` Gamma map values used to remap the "Aplha" color values (`glPixelMapfv( GL_PIXEL_MAP_A_TO_A, nb, alpha))`
$GAMMA_PT	(A,F_{32})	Array of `$GAMMA_NB` Gamma map values used to remap the R G and B color values (`glPixelMapfv( GL_PIXEL_MAP_R_TO_R, nb, pt))`

Commands recognized in this library:

Gamma: Set *value*
Assign the value to the `$GAMMA_CUR` variable that set the current Gamma correction value (0.3 <= Gamma <= 3.0)

SliceO_Tool\2D Pixel Grid.dll

System Variables defined in this library:

`$GRID_ON`	(U_8)	Display (=1) or not (=0) a grid on top of the image's pixels. (Default=0)
`$GRID_MIN`	(U_8)	Minimum scale value for the grid to be displayed (Default=5)
`$GRID_COLOR`	(C)	Color of the grid

Commands recognized in this library:

Grid: On (on|off|toggle)
Assign a value to the variable `$GRID_ON`.

Grid: Min *value*
Assign a value to the variable `$GRID_MIN`.

Grid: Color *R G B*
Assign a value to the variable `$GRID_COLOR`.

SliceO_Tool\2D Overlay.dll

System Variables defined in this library:

`$OVERLAY_2D`	(U_{32})	Bit field that control the 2D overlay display: • x01: file overlay on/off • x02: name overlay on/off • x04: patient overlay on/off • x08: technical overlay on/off

Commands recognized in this library:

Overlay: 2D *bit_field*
Overlay: (file|name|patient|technical) (on|off|toggle)
Assign a value to the `$OVERLAY_2D` variable.

SliceO_Tool\3D Background.dll

System Variables defined in this library:

`$BACKGROUND_CORNER`	(A,C)	Array of 4 colors. One for each corner of the windows.

Commands recognized in this library:

Background: (top|bottom) (left|right) *R G B*
Assign a value to the elements of the $BACKGROUND_CORNER variable.

SliceO_Tool\3D Light.dll

System Variables defined in this library:

$LIGHT_MAX	(U_{16},R)	Maximum number of light sources. (= 8)
$LIGHT_NB	(U_{16})	Current number of defined lights (Default=5)
$LIGHT_LIST	(A,U_{16})	List of all the lights (1,..., n)
$LIGHT_STATE	(A,U_8)	Array of light states (0=Off, 1=On, 3=Ambient)
$LIGHT_COLOR	(A,C)	Array of light colors
$LIGHT_DIR	(A,P)	Array of pointer to the light's directions

Commands recognized in this library:

Light: *t_light* State (on|off)
Assign values to the elements of $LIGHT_STATE specifying the state of the light sources. (0=Off, 1=On).

Light: *t_light* Dir *x y z*
Assign values to the elements of $LIGHT_DIR specifying the direction of the light sources.

Light: *t_light* Color *R G B*
Assign values to the elements of $LIGHT_COLOR specifying the color of the light sources.

Templates:

t_light is a template for the light names ("amb", "1" to "4")

SliceO_Tool\3D Measures.dll

Commands recognized in this library:

Measure: *t_geom* 3D *file_name*
Write to file "file_name" the measures for all the geometries matching "t_geom".

Note:

If optional headers have been requested (see File in The Configuration Menu), they will be written to the file.

Templates:

t_geom is a template for the 3D geometries

SliceO_Tool\3D Overlay.dll

System Variables defined in this library:

$OVERLAY_3D ($U_{32}$) Bit field that control the 2D overlay display:
- x01: global overlay on/off
- x02: local overlay on/off
- x04: box overlay on/off
- x08: slice overlay on/off

Commands recognized in this library:

Overlay**: 3D *bit_field***
Overlay**: (glo**bal**|loc**al**|box|sl**ices**) (on|off|tog**gle**)**
Assign a value to the $OVERLAY_3D variable.

SliceO_Tool\3D Shading.dll

Shading:

SliceO_Tool\3D Slices.dll

Slice: Transparency ***value***
Set the transparency level for the 3D slices. if value=0, the slices will have no transparency. (0 <= value <= 1.0)

SliceO_Tool\3D Smooth.dll

System Variables defined in this library:

$SMOOTH_NB ($U_{16}$,R) Number of "Smoothing" levels. (= 5)
$SMOOTH_LIST (A,$U_{16}$) List of the parameters (1, ..., n)
$SMOOTH_PARAM (A,$U_{16}$) Array of number of recursion used in each smoothing levels.

SliceO_Tool\3D Transfo.dll

System Variables defined in this library:

$TRANSFORM_INTERFACE ($U_{16}$) Type of 3D mouse interface. (0=Track, 1=Classic, 2=Patient)

Commands recognized in this library:

Transform**: Mo**de **(tra**ck**|cla**ssic**|pat**ient**)**

Assign a value to the $TRANSFORM_INTERFACE variable. (See Transform in **The 3D Tools**)

Transform: *t_window* Camera *x y z*
Place the observer at the position x, y, z for all the windows matching the template "t_window".

Transform: *t_window* Center *x y z*
Center all the windows matching the template "t_window" around the point x, y, z.

Transform: *t_window* FOV *angle*
Change the camera Field Of View "angle" to all the windows matching the template "t_window". Accepted values for "angle" range from 0.0 to 179.0 degrees. A value of 0 will yield an orthogonal projection. The default Camera has a FOV of 27 degrees.

Transform: *t_window* Matrix *a11 a12 a13 a14 a21 ... a44*
Directly set the matrix elements. The 16 parameters of this command will be inserted in the 4x4 transformation matrix.of all the windows matching the template "t_window".

Transform: *t_window* Pivot *angle x y z*
Rotate "angle" degrees around the direction specified by x, y, z all the windows matching the template "t_window".

Transform: *t_window* Restore
Restore the transformation matrix all the windows matching the template "t_window".

Transform: *t_window* Rotate *x y z*
Rotate all the windows matching the template "t_window".

Transform: *t_window* Scale *value*
Scale all the windows matching the template "t_window".

Transform: *t_window* Translate *x y z*
Translate all the windows matching the template "t_window".

Templates:

t_window is a template for the window. It match either the widow ID ("1" to "4") or "cur" for the currently selected window.

SliceO_Tool\Tag Lock.dll

System Variables defined in this library:

$TAG_LOCK (A,$U_8$) Array of "lock" values. (0=unlock, 1=lock)

Commands recognized in this library:

Tag: *t_tag* Lock
Assign the value "lock" (=1) to all the elements of $TAG_LOCK that match the template

"t_tag".

Tag: *t_tag* Unlock
Assign the value "unlock" (=0) to all the elements of `$TAG_LOCK` that match the template "t_tag".

Templates:

t_tag is a template for the TAG values. It match either the TAG ID ("0" to "256") or the TAG label or "cur" for the current TAG value.

SliceO_Tool\Tag Surface.dll

System Variables defined in this library:

$RESULT_BACKWARD	(U8)	Backward compatibility flag (1=compatible with 4.3)
$RESULT_NAME	(A,S)	Array of result computation name
$RESULT_UNIT	(A,S)	Array of result computation units
$RESULT_ENABLE	(A,C_8)	Array of flag for the display/output of the result computations (0x01=in output file, 0x02=displayed)

Commands recognized in this library:

Results: Backward (on|off|toggle)
Enable or disable the backward compatibility mode. If enable, the database results will have a "frame number" column, and also a column for each empty tag.

Results: Calib (on|off|toggle)
Enable or disable the calibration of the pixel values.

Results: Calib File *file_name*
Read a calibration file. The calibration file is a script file containing commands specific to the pixel calibration. The commands used in the calibration file are explained further down.

Results: Display *t_measure* (on|off|toggle)
Enable or disable the display of the measures matching the template "t_measure" in the tool's window. By default, only the "Surface Units" and "Volume Units" measures are displayed.

Results: Enable *t_measure* (on|off|toggle)
Enable or disable the saving of the measures matching the template "t_measure" in the result files. By default, only the "GLI Variance" measure is disabled.

Results: Measure List
list the names of all the measures performed by this tool. These names can then be used in the "results: enable" and "results: display" commands

Results: Name *string*
Specify a default name for the result file.

Results: Type (text|db)
Specify a default file type for the result file. The choice is between "db" (spread sheet format) or "text" (simple ASCII text format).

Results: Write [text|db] *file_name*
Write the surface/volumes to the result file "file_name", in either "db" (spread sheet format) or "text" (simple ASCII text format). If no format is specified, the default value is used.

The Calibration File

The calibration file is a script file containing commands specific to the pixel calibration.

Results: Calib Units *string*
Specify the name of the new units described by the calibration data

If the relationship between the GLI pixel values and the new calibrated values is a simple linear ,transformation, then we can express the calibration with 2 values: offset and scale: new value = (GLI_Value + offset) * scale

Results: Calib Offset *value*
Specify the offset of the calibration curve (new value = (GLI_Value + offset) * scale).

Results: Calib Scale *value*
Specify the scale of the calibration curve (new value = (GLI_Value + offset) * scale).

If the relationship is more complex, we express the calibration with a series of point on calibration curve. All GLI values not specifically defined will be linearly interpolated or extrapolated from the values specified.

Results: Calib Point *val_GLI val_new*
Specify a point on the calibration curve.

Appendices: B: Variables and Commands: B: The Mode DLLs

SliceO_2D\Measure.dll

Commands recognized in this library:

Measure: ***t_measure*** **Del**ete
Delete all the measurement instrument matching the template "t_measure".

Measure: ***t_measure*** **Sav**e ***file_name***
Create a script file to re-create all the measurements matching the template "t_measure".

Measure: ***t_measure*** **Wri**te ***file_name***
Create a text file containing the measures from all the measurements matching the template "t_measure".

Measure: ***t_frame*** **Caliper** ***name a_x a_y ...***
Measure: ***t_frame*** **Distance** ***name a_x a_y ...***
Measure: ***t_frame*** **Dist_to_Line** ***name a_x a_y ...***
Measure: ***t_frame*** **Profil** ***name a_x a_y ...***
Measure: ***t_frame*** **Protractor** ***name a_x a_y ...***
Measure: ***t_frame*** **ROI** ***name a_x a_y ...***
Measure: ***t_frame*** **Ruler** ***name a_x a_y ...***
Create the specified measurement tool named "name" with the parameters "a_x", "a_y" etc on the frame matching "t_frame".

Measure: **Pro**pagate [**up**|**down**]
Propagate the result from the current frame to the next frame in the group, either up (the default) or down.

Templates:

t_measure is a template for the measure names. **t_frame** is a template for the frame names.

SliceO_3D\Contour.dll

System Variables defined in this library:

`$CONTOUR_MODDE`	(U_8)	Contour mode. 0=points, 1=lines.
`$CONTOUR_SUBSAMPLING`	(A,U_{16})	Array of 3 sub-sampling values (x,y and z). Sub-sampling values goes from 0 (no sub-sampling) to 10 (1 pixel every 10).

Commands recognized in this library:

Contour: Mode (line|point)
Assign a value to the `$CONTOUR_MODE` variable. 0 = point mode, 1 = line mode.

Contour: Sampling (x|y|z) value
Assign values to the x,y and z components of the `$CONTOUR_SUBSAMPLING` array.

Contour: Compute [line|point]
Compute the contour curve. If the "line" or "point" parameter is present, it define the type of values. Otherwise, the `$CONTOUR_MODE` variable is used.

SliceO_3D\Shell_....dll

System Variables defined in this library:

$SHELL_LORENSEN_MODDE	(U_8)	Shell reconstruction mode. This is a bit field with the values: • x01 = Mode Ouside • x02 = Mode Inside • x04 = Mode Border
$SHELL_TOMO_MODDE	(U_8)	
$SHELL_TOMO_SUBSAMPLING	(A,U_{16})	Array of 3 sub-sampling values (x,y and z). Sub-sampling values goes from 0 (no sub-sampling) to 10 (1 pixel every 10).

Commands recognized in this library:

Shell: Mode (inside|outside|border) (on|off)
Change the value of the specified bit in both `$SHELL_LORENSEN_MODE` and `$SHELL_TOMO_MODE`.

Shell: Sampling (x|y|z) value
Assign values to the x,y and z components of the `$SHELL_TOMO_SUBSAMPLING` array.

Shell: Compute (TomoVision|Lorensen)
Compute the polygon shell according to the select model.

SliceO_Tag\Flood.dll

Commands recognized in this library:

Flood: 2D cursor
Flood the region under the cursor with the current TAG value.

Flood: 2D *x y*
Flood the region starting at point x, y of the current frame with the current TAG value.

SliceO_Tag\Mask.dll

System Variables defined in this library:

$GEOM_MASK_CAPTURE	(F_{32})	Capture radius (in pixels). If clicking within this distance of mask's anchor point, the point itself is considered clicked.
$GEOM_MASK_GLOBAL	(U_8)	The masks are either local (=0) or global (=1). This variable is used as default value when creating new masks.
$GEOM_MASK_FILL	(U_8)	The masks are either outlines (=0) or filled (=1). This variable is used as default value when creating new masks.
$GEOM_MASK_NB	(U_{16})	The number of created masks.
$GEOM_MASK_LIST	(A,P)	Array of pointers to the masks.
$GEOM_MASK_ACTIVE	(P)	Pointer to the active mask.

Commands recognized in this library:

Mask: Radius *value*
Change the default value (=10.0) of the radius for the "Surface" shape.

Mask: *t_mask* Tag *tag_num*
Assign the TAG value "tag_num" to all the masks matching the template "t_mask".

Mask: *t_mask* Mode **(glo**bal**|loc**al**|out**line**|fil**led**)**
Assign the TAG value "global/local" or "outline/filled" to all the masks matching the template "t_mask".

Mask: *t_mask* Delete
Delete all the masks that match the template "t_mask".

Mask: *t_mask* Save ***file_name***
Create a script file to re-create all the masks matching the template "t_mask".

Mask: *t_frame* Circle *name a_x a_y ...*
Mask: *t_frame* Polygon *name a_x a_y ...*
Mask: *t_frame* Quadrant *name a_x a_y ...*
Mask: *t_frame* Rectangle *name a_x a_y ...*
Mask: *t_frame* Spline *name a_x a_y ...*
Create the specified masks named "name" with the parameters "a_x", "a_y" etc on the frame matching "t_frame".

Mask: Propagate [**up**|**down**]
Propagate the result from the current frame to the next frame in the group, either up (the default) or down.

Templates:

t_mask is a template for the mask names. **t_frame** is a template for the frame names.

SliceO_Tag\Morpho.dll

System Variables defined in this library:

$MORPHO_MERGE_MAX	(U_{16},R)	Maximum number of "Merge" buttons.
$MORPHO_MERGE_NB	(U_{16})	Actual number of "Merge" buttons.
$MORPHO_MERGE_LIST	(A,U_{16})	List of the merge parameters (1, ..., n)
$MORPHO_MERGE_CUR	(U_{16})	Current "Merge" value.
$MORPHO_PARAM_DIST	(A,F_{32})	Array of "Distance" parameters for the merges.
$MORPHO_PARAM_SURF	(A,F_{32})	Array of "Surface" parameters for the merges.
$MORPHO_LINE_MODE	(U_8)	Display mode for the watershed lines (0=off, 1=small, 2=thick).
$MORPHO_LINE_COLOR	(C)	Color of the watershed lines.

Commands recognized in this library:

Morpho: Line (off|small|thick)
Assign a value to the variable `$MORPHO_LINE_MODE` that control the appearance of the watershed lines.

Morpho: Color R G B
Assign a value to the variable `$MORPHO_LINE_COLOR` that control the color of the watershed lines.

Morpho: Param *t_merge surf diff*
Assign values to all the elements of `$MORPHO_PARAM_SURF` and `$MORPHO_PARAM_DIST` that match the template "t_merge". These variables control the merge of the watersheds.

Morpho: [*t_frame*] Compute *t_merge*
Compute the merge of the watersheds for all merge matching the template "t_merge" for all the frames matching the template "t_frame"

Morpho: [*t_frame*] Close *size*
Morpho: [*t_frame*] Dilatation
Morpho: [*t_frame*] Erosion
Morpho: [*t_frame*] Open *size*
Perform Erosion, Dilatation, Open or Close Mathematical Morphology operation on all the frames matching the template "t_frame". The Open and Close operation also have a "size" parameter that control the reach of the operation

Morpho: Propagate [up|down]
Propagate the result from the current frame to the next frame in the group, either up (the default) or down.

Templates:

t_merge is a template for the merge numbers ("1" to "4")
t_frame is a template for the frame names.

SliceO_Tag\Region.dll

System Variables defined in this library:

$REGION_OVERWRITE	(U_8)	Bit field that control how regions overwrite Tag values already present in the pixels (Grow 2D and 3D only): x01 = The region does not overwrite a pixel with a Tag value equal (=) to the current Tag value. x02 = The region does not overwrite a pixel with a Tag value superior (>) to the current Tag value. x04 = The region does not overwrite a pixel with a Tag value inferior (<) to the current Tag value. The default value is x03 (>=)
$REGION_TAG_FLAG	(A,U_8)	Array of bit-field modes for each TAGs. x001 = The lower threshold is on x010 = the hight threshold is on
$REGION_TAG_MIN	(A,F_{32})	Array of lower threshold values for all TAGs
$REGION_TAG_MAX	(A,F_{32})	Array of higher threshold values for all TAGs
$REGION_WHEEL_MODE	(U_8)	Mode of the mouse wheel (0=off, 1=, 2=).
$REGION_PREVIEW_ON	(U_8)	Preview mode (0=off, 1=on).
$REGION_PREVIEW_COLOR	(C)	Color of the preview.

Commands recognized in this library:

Region: Color *R G B*
Assign a value to the variable `$REGION_PREVIEW_COLOR` that define the color of the Region Growing preview.

Region: Preview (on|off|toggle)
Assign a value to the variable `$REGION_PREVIEW_ON` that define if a preview is shown.

Region: Mode (paint|2d|3d)
Set the sub mode of the Region Growing mode. The choices are "paint", "2D or "3d".

Region: Wheel (off|min|max)
Assign a value to the variable `$REGION_WHEEL_MODE` that define the action of the mouse wheel.

Region: *t_tag* Min (on|off) [*value*]
Assign values to the element matching "t_tag" in the variables `$REGION_TAG_FLAG` and `$REGION_TAG_MIN`. The `$REGION_TAG_FLAG` element is defined by the "on/off" parameter, the `$REGION_TAG_MIN` element is only assigned the vale "value" if the "on/off" parameter is "on". This will set the lower threshold values for all TAGs matching "t_tag".

Region: *t_tag* Max (on|off) [*value*]
Assign values to the element matching "t_tag" in the variables `$REGION_TAG_FLAG` and `$REGION_TAG_MAX`. The `$REGION_TAG_FLAG` element is defined by the "on/off" parameter, the `$REGION_TAG_MAX` element is only assigned the vale "value" if

the "on/off" parameter is "on". This will set the lower threshold values for all TAGs matching "t_tag".

Region: *t_tag* Slider (*|min|max) (lock|unlock|toggle)
Lock or unlock the sliders for the lower and higher thresholds of the TAGs matching "t_tag"

Region: Propagate [up|down]
Propagate the result from the current frame to the next frame in the group, either up (the default) or down.

Templates:

t_tag is a template for the TAG values. It match either the TAG ID ("0" to "256") or the TAG label or "cur" for the current TAG value.

SliceO_Tag\Snake.dll

Commands recognized in this library:

Snake: [*t_frame*] *t_tag* Radius *value*
Assign the radius "value" to all the points of all the Snakes associated with the Tag matching the "t_tag" template on the frames matching the "t_frame" template.

Snake: [*t_frame*] *t_tag* Capture *value*
Assign the capture range "value" to all the Snakes associated with the Tag matching the "t_tag" template on the frames matching the "t_frame" template. The capture range determines how much the Snake can move to latch on the best gradient.

Snake: [*t_frame*] *t_tag* Delete *value*
Delete all the Snakes associated with the Tag matching the "t_tag" template on the frames matching the "t_frame" template.

Snake: [*t_frame*] *t_tag* Save *file_name*
Create a script file to re-create all the snakes matching the tags in the template "t_tag" from the frames matching the template "t_frame".

Snake: *t_frame t_tag* Create *nb_pts radius capture*
Create a new snakes on the frame matching "t_frame" for the tag matching t_tag. This new snake will have nb_pts points with a default radius of "radius" and a default capture range of "capture". The points for the snake will be given by a series of "snake: point" commands.

Snake: Point *id x y radius*
Add a point to a newly created snake.

Snake: Propagate [up|down]
Propagate the result from the current frame to the next frame in the group, either up (the default) or down.

Snake: Tag
Perform the "Snake to Tag" operation on the snake matching the selected TAG value.

Snake: Geom
Perform the "Snake to Geom" operation on the snake matching the selected TAG value.

Templates:

t_tag is a template for the TAG values. It match either the TAG ID ("0" to "256") or the TAG label or "cur" for the current TAG value.
t_frame is a template for the frame names.

SliceO_Tag\Threshold.dll

System Variables defined in this library:

$THRESHOLD_MAX	(U_{16},R)	Maximum number of threshold sliders (= 4)
$THRESHOLD_NB	(U_{16})	Actual number of threshold sliders. (Default = $THRESHOLD_MAX)
$THRESHOLD_LIST	(A,U_{16})	List of the thresholds (1, ..., n)
$THRESHOLD_GLI	(A,F_{32})	Array of threshold values for each threshold sliders.
$THRESHOLD_TAG	(A,U_{16})	Array of associated TAG values for each threshold sliders.
$THRESHOLD_WHEEL	(U_8)	Mode of the mouse wheel (0=off, 1 to $THRESHOLD_NB).

Commands recognized in this library:

Threshold: Threshold *id value* [*tag*]
Assign a value to the element "id" of the variables $THRESHOLD_GLI and $THRESHOLD_TAG if "tag" is present.

Threshold: Compute [*val_1*...]
Compute the segmentation for currently selected images. If threshold values are given, they will be used for the segmentation, otherwise, the values fixed by the interface are used.

Threshold: Slider *t_id* (lock|unlock|toggle)
Lock or unlock the sliders matching the template "t_id".

Threshold: Wheel (off|1|2...)
Assign a value to the variable $THRESHOLD_WHEEL_MODE that define the action of the mouse wheel.

Appendices:
C: The Keyboard Shortcuts

You can assign a command or a macro to a keyboard key. The mechanism to assign a command to a key is:

Key: *name* Map *script_command*

Where "name" is the key's name.

Key Names

Here is a list of the available key names. Note that the names are case sensitive, so "SHIFT_ENTER" is valid while "Shift_Enter" is not.

Normal keyboard keys:

Key	Shift + Key	Alt + Key	Ctrl + Key
a ... z	A ... Z	ALT_A ... ALT_Z	CTRL_A ... CTRL_Z
0 ... 9	SHIFT_0 ... SHIFT_9	ALT_0 ... ALT_9	CTRL_0 ... CTRL_9
ENTER	SHIFT_ENTER	ALT_ENTER	CTRL_ENTER
BACKSPACE	SHIFT_BACKSPACE	ALT_BACKSPACE	CTRL_BACKSPACE
TAB	SHIFT_TAB	ALT_TAB	CTRL_TAB

Normal keyboard keys that do not have "Shift", "Alt or "Ctrl" variations:

Key					
`	~	!	@	#	$
^	&	*	(	)	-
_	=	+	[	{	]
}	\	|	;	:	'
"	,	<	.	>	/
?	SPACE				

Arrow and Center keys:

Key	Shift + Key	Alt + Key	Ctrl + Key
PRINT	SHIFT_PRINT	ALT_PRINT	CTRL_PRINT
UP	SHIFT_UP	ALT_UP	CTRL_UP
DOWN	SHIFT_DOWN	ALT_DOWN	CTRL_DOWN
LEFT	SHIFT_LEFT	ALT_LEFT	CTRL_LEFT
RIGHT	SHIFT_RIGHT	ALT_RIGHT	CTRL_RIGHT

INSERT	SHIFT_INSERT	ALT_INSERT	CTRL_INSERT
HOME	SHIFT_HOME	ALT_HOME	CTRL_HOME
PAGE_UP	SHIFT_PAGE_UP	ALT_PAGE_UP	CTRL_PAGE_UP
DELETE	SHIFT_DELETE	ALT_DELETE	CTRL_DELETE
END	SHIFT_END	ALT_END	CTRL_END
PAGE_DOWN	SHIFT_PAGE_DOWN	ALT_PAGE_DOWN	CTRL_PAGE_DOWN

Key Pad keys:

Key	Shift + Key	Alt + Key	Ctrl + Key
PAD_DIVIDE	SHIFT_PAD_DIVIDE	ALT_PAD_DIVIDE	CTRL_PAD_DIVIDE
PAD_MULTIPLY	SHIFT_PAD_MULTIPLY	ALT_PAD_MULTIPLY	CTRL_PAD_MULTIPLY
PAD_DELETE	SHIFT_PAD_DELETE	ALT_PAD_DELETE	CTRL_PAD_DELETE
PAD_MINUS	SHIFT_PAD_MINUS	ALT_PAD_MINUS	CTRL_PAD_MINUS
PAD_PLUS	SHIFT_PAD_PLUS	ALT_PAD_PLUS	CTRL_PAD_PLUS
PAD_PERIOD	SHIFT_PAD_PERIOD	ALT_PAD_PERIOD	CTRL_PAD_PERIOD
PAD_0 ... PAD_9	SHIFT_PAD_0 ... SHIFT_PAD_9	ALT_PAD_0 ... ALT_PAD_9	CTRL_PAD_0 ... CTRL_PAD_9
PAD_ENTER	SHIFT_PAD_ENTER	ALT_PAD_ENTER	CTRL_PAD_ENTER

F keys:

Key	Shift + Key	Alt + Key	Ctrl + Key
F1 ... F12	SHIFT_F1 ... SHIFT_F12	ALT_F1 ... ALT_F12	CTRL_F1 ... CTRL_F12

Keys already mapped

SliceOmatic makes extensive use of the key mapping. Here's the list of keys that are already mapped for you. (You can also get this list with the "key: list" command),

Key	Mapped Command
F1 F2 F3 F4	Color: Grey Color: Mixed Color: Over Color: Tag
F5 ... F10	Brush: Current 1 ... Brush: Current 6
0 and PAD_0 ... 9 and PAD_9	Tag: Current 0 ... Tag: Current 9

PAD_PERIOD PAD_DELETE	Tag: Grab
SPACE	Window: 0 Mode toggle
CTRL_Z CTRL_Y	Undo: Undo Undo: Redo
PRINT	Snapshot: Click
= -	$TAG_MIX = $TAG_MIX + 0.1 $TAG_MIX = $TAG_MIX - 0.1
g	Grid: On toggle
ENTER	Flood: 2D toggle
PAD_PLUS PAD_MINUS PAGE_UP PAGE_DOWN HOME END	Frame: Scale inc Frame: Scale dec Frame: Current inc Frame: Current dec Frame: Parent inc Frame: Parent dec
SHIFT_F5 SHIFT_F6 SHIFT_F7 SHIFT_F8	Overlay: File toggle Overlay: Name toggle Overlay: Patient toggle Overlay: Technical toggle
SHIFT_F9 SHIFT_F10 SHIFT_F11 SHIFT_F12	Overlay: Global toggle Overlay: Local toggle Overlay: Box toggle Overlay: Slice toggle
a s d	Morpho: Line 0 Morpho: Line 1 Morpho: Line 2
q w e r	Morpho: * Compute 1 Morpho: * Compute 2 Morpho: * Compute 3 Morpho: * Compute 4
ALT_D ALT_1 ALT_2 ALT_3 ALT_4 ALT_5 ALT_6 ALT_7 ALT_8 ALT_9 ALT_0	Debug: Database Debug: Filter toggle Debug: Gradient Amp toggle Debug: Gradient X toggle Debug: Gradient Y toggle Debug: Gradient Z toggle Debug: WS toggle Debug: Merge 1 toggle Debug: Merge 2 toggle Debug: Merge 3 toggle Debug: Merge 4 toggle

Note:

The key F12 is also used by sliceOmatic, but the mechanism that was used for assigning the key F12 to the interactive help system is different. The variable "$INTERFACE_HELP_KEY" is used to identify the help key. By default we have:

$INTERFACE_HELP_KEY = F12

Appendices:
D: The TAG File Format

The TAG file format is loosely based on the university of Waterloo IM format.

The image file is composed of three sections:
- The header
- the image binary data.

The Header

The header is entirely composed of lines of ASCII text. Each line is terminated by the characters <CR> and <LF> (0x0D and 0x0A). The header is terminated by a <FF> character (0x0C).

Everything on a line following a "*" will be considered as comments and can be disregarded by the program.

The header is composed of a series of keywords value pairs. The keyword and values are separated by "**:**". Each pair of keyword and values are separated by one or more separation characters. The recognized separators are: " " (space), "," (comma), "\t" (tab) or "\n" (new-line). You can use lowercase or uppercase indifferently in the keywords, the program converts all the keywords characters to uppercase before parsing the header.

The recognized keywords and their permitted values are:

x:	"X" resolution (in pixels).
y:	"Y" resolution (in pixels).
z:	Number of images in the file.
type:	Gives the size of each pixel, the values supported by the program are BYTE or SHORT (only BYTE is used for the ".tag" files).
org_x: **org_y:** **org_z:**	Position in "x", "y" and "z" of the center of the top left pixel of the image.
dim_x: **dim_y:**	Total dimension in "x" and "y" of the image (in millimeters).
inc_x: **inc_y:**	Distance between 2 consecutive pixels in "x" and "y" of the image (in millimeters).
epais:	Slice thickness.
dir_h_x: **dir_h_y:** **dir_h_z:**	X, y and z components of the horizontal direction vector (in patient system).
dir_v_x: **dir_v_y:** **dir_v_z:**	X, y and z components of the vertical direction vector (in patient system).
uid:	Unique number used to make sure this tag is associated with the correct GLI image.
chksum:	Checksum of the original GLI image.

```
x:256    y:256    z:9      type:BYTE
org_x:-204.2221  org_y:-181.8909  org_z:-250.0000
inc_x:0.7105     inc_y:0.7105     epais:5.0000
dir_h_x:1.0000     dir_h_y:0.0000     dir_h_z:0.0000
dir_v_x:0.0000     dir_v_y:1.0000     dir_v_z:0.0000
uid:AFCCCAC6 chksum:09F1588D bin:256
* number of echos:        0
^L
```

Sample TAG header

The Image Data

The image data is written in binary form. There are X x Y x Z pixels in the image data. The values for X, Y and Z come from the header. Each pixel takes 1 or 2 bytes according to the value associated with the "type" keyword. The pixels are written using the algorithm:

```
for each image k
        for each line j (starting at the top)
                for each pixel i (starting at the left)
                        write pixel [k][j][i] ;
```

Appendices:
E: The MOVIE.BYU file format

The MOVIE.BYU file format is an old format used to described 3D objects.

The Header

The first line of the format contains 4 integer values:

- The number of objects in the file
- The number of nodes in the file
- The number of elements in the file (points, lines and polygons)
- The number of edges in the file

The part list

The next lines contain a series of 2 integers per object in the file. For each object we have the index of the first and last elements in the connectivity array that describe this object.

The Nodes

Next, we have the X, Y and Z positions of all the nodes in the geometries. Originally, these float values (E12.5 in FORTRAN) should be placed 6 to a line, but for clarity, we only place 1 node (3 values) on each line and the formating is not important.

The connectivity array

Last, we have a series of integer vales describing each element in the objects. These are index in the node array (the first one being "1"), Each element is created by connecting one or more nodes together, the connectivity tell us how many and which nodes. The "how many" is given by having the last node of each element being negative. For example a triangle created by connecting the first 3 nodes will be expressed as: "1 2 -3". Originally, the format requested that the values of the connectivity array be written 10 integers per lines. We only place the connectivity of one element per line.

Note:

The original MOVIE.BYU program was written in FORTRAN. In those days, file formats were rather strict, integer values in the MOVIE.BUY format had to be I8, meaning 8 characters long (no more, no less) and float had to be E12.5 (12 characters, with 5 for the fraction). The nodes had to be written 6 values to a line, while most integers were 10 values to a line. Nowadays, the programming languages are much more flexible and sliceOmatic does not respect these strict constraints.

Note:

The original MOVIE.BYU file format could describe elements having 3 to 8 nodes (it could be used to create finite element meshes). In sliceOmatic, we have extended the element definition to 1 and 2 nodes elements (point and line segments) but we will not accept shapes with more than 4 nodes (quads).

Example (using the original MOVIE.BYU syntax):

```
 1       8       6      24
 1       6
0.00000e+00 0.00000e+00 0.00000e+00 1.00000e+00 0.00000e+00 0.00000e+00
1.00000e+00 1.00000e+00 0.00000e+00 0.00000e+00 1.00000e+00 0.00000e+00
0.00000e+00 0.00000e+00 1.00000e+00 1.00000e+00 0.00000e+00 1.00000e+00
1.00000e+00 1.00000e+00 1.00000e+00 0.00000e+00 1.00000e+00 1.00000e+00
1       4       3      -2       5       6       7      -8       1       2
6      -5       2       3       7      -6       3       4       8      -7
4       1       5      -8
```

The 6 faces of a cube as 4 nodes polygons

Example (using sliceOmatic syntax):

```
 1    8      8      8
 1    8
-1 -1 -1
 1 -1 -1
 1  1 -1
-1  1 -1
-1 -1  1
 1 -1  1
 1  1  1
-1  1  1
-1
-2
-3
-4
-5
-6
-7
-8
```

The 8 corners of a cube as points

```
 1    8      12     24
 1    12
-1 -1 -1
 1 -1 -1
 1  1 -1
-1  1 -1
-1 -1  1
 1 -1  1
 1  1  1
-1  1  1
1 -2
2 -3
3 -4
4 -1
5 -6
6 -7
7 -8
8 -5
1 -5
2 -6
3 -7
4 -8
```

The 12 edges of a cube as line segments

```
 1    8      6      24
 1    6
-1 -1 -1
 1 -1 -1
 1  1 -1
-1  1 -1
-1 -1  1
 1 -1  1
 1  1  1
-1  1  1
1 2 3 -4
8 7 6 -5
2 6 7 -3
4 8 5 -1
3 7 8 -4
1 5 6 -2
```

The 6 faces of a cube as 4 nodes polygons