Abstract

This document is the user manual for Pablo and was made according to the Pablo version 2. This user manual instructs how to install and use this research software to do model fitting and statistical analysis.

Chapter 1

Introduction

1.1 What is Pablo?

Pablo is a research software package developed by MIDAG (the Medical Image Display and Analysis Group) at UNC-Chapel Hill. This software can deal with varieties of medical image applications, mainly focusing on s-rep model fitting, segmentation and statistical analysis. It takes binary images and distance maps as the input, and transforms them into s-reps, which are the underlying structure in representing the 3D objects using this software. In addition, it is a powerful tool to visualize both the s-reps and the images which they are fit to: it can display the 3D structure from any angle (supporting the operation of rotation, zoom in/out, pan, mirror, etc.); it can display the 3D image in three 2D cut planes (Axial plane, Sagittal plane, Coronal plane); and it can display the target object as a binary image or as a set of boundary tiles. Moreover, it allows the users to edit any single atom in s-reps to improve the whole model.

An s-rep is a quasi-medial skeletal model, in our case with no branches. As described in Ch.3 [Damon] of [Siddiqi & Pizer], skeletal models consist of a folded sheet (or space curve) called the skeleton with spoke vectors proceeding from each skeletal point to the implied object boundary in such a way that no spokes cross and the union of the spokes forms the object interior. The mathematics of skeletal models is available in the just cited chapter.

A skeletal model is quasi-medial if for each spoke, (i) its length is approximately the same as that of its partner proceeding from the same location but the other side of the fold; (ii) the spoke is approximately orthogonal to the implied boundary; (iii) the spokes proceeding from the fold curve of the skeleton approximately meet a crest point of the implied boundary. We represent these continuous s-reps discretely by sampling each side of the skeletal sheet by a regular grid of spokes and by sampling the fold curve regularly with spokes, producing three subgroups of spokes that we call “top, bottom, and fold” spokes respectively. A continuous s-rep can be derived from the discrete s-rep by a spoke interpolation method consistent with skeletal geometry [Tu & Vicory, CVIU].

Rather than being derived from the boundary of an object, an operation sensitive to noise [Chs. 4-7 of Siddiqi & Pizer medial book], discrete s-reps are fit to an object's boundary by an iterative optimization process. The optimization is initialized by an s-rep derived from a warp of a reference model [Tu, SPL] and has an objective function made up from terms penalizing each of the following: (i) mismatch of interpolated spoke ends to the boundary of the object being fit to; (ii) deviation from orthogonality of the spoke ends to the object boundary; (iii) deviation of the fold spoke ends and directions from behavior consistent with a crest; (iv) regularity of all three subgroups of the discrete spokes; (v) measures of spoke crossing behavior; (vi) optionally, -log(probability density of the spoke tuples) if such a probability function has been computed from training. Typically the optimization is followed by a spoke by spoke refinement to tighten the fit of the interpolated s-rep's implied boundary to the object boundary.

This software fits an s-rep to an underlying image in either of two ways: automatically and manually. For the automatic way, it computes the similarity transformation for the whole model, and then transforms the spokes according to the computed transformation to get the fitted model. For the manual way, the positions of the atoms, the directions and the lengths of the spokes are under the user's control.

1.2 Intended readership

This document covers the use for the shape researchers:

- To fit models: Users will adjust a pre-built model (an initial s-rep) to a target image. These users just need to know about the properties of the target image and adjust the parameters in the configuration file to get a good fit.
- To do statistical analysis: Pablo has the toolbox to do statistical analysis using CPNS. Based on the enough good fitted models, CPNS can get a mean model and its variations. Pablo provides the function to visualize the modes of variations when a shape space is provided.

1.3 How to use this document?

- Chapter 2 introduces some necessary concepts related to Pablo. It will help the users to better understand s-rep object models and Pablo and get a better fit.
- Chapter 3 describes how to install Pablo.
- Chapter 4 introduces the GUI for Pablo that enables users to get started quickly.
- Chapter 5 contains tutorials for model fitting.
- Chapter 6 introduces the statistical tools.

1.4 Related documents

The s-rep is the underlying structure to represent a model using Pablo. All the documents related to Pablo and s-reps can be found here: http://midag.cs.unc.edu/. Pablo software can be downloaded from http://www.nitrc.org/projects/sreps.

Here are some suggested papers.

[1] Sungkyu Jung, Ian L Dryden, J.S. Marron, “Analysis of principal nested spheres”, Biometrika, 2012.

[2] Derek Merck, Gregg Tracton, Rohit Saboo, Joshua Levy, Edward Chaney, Stephen Pizer, and Sarang Joshi, "Training Models of Anatomic Shape Variability," *Medical Physics*, vol. 35, no. 8, pp. 3584-3596, 2008.

[3] Stephen M Pizer, Sungkyu Jung, Dibyendusekhar Goswami, Jared Vicory, Xiaojie Zhao, Ritwik Chaudhuri, James N Damon, Stephan Huckemann, and J.S. Marron, "Nested Sphere Statistics of Skeletal Models," in *Innovations for Shape Analysis: Models and Algorithms,* (M. Breuss, A. Bruckstein, and P. Maragos, eds.), pp. 93-113, 2013.

[4] K Siddiqi & SM Pizer, *Medial Representations,* Springer 2008.

[5] L Tu, J Vicory, S Elhabian, B Paniagua, JC Prieto, JN Damon, R Whitaker, M Styner, SM Pizer, “Entropy-based Correspondence Improvement of Interpolated Skeletal Models,” To appear, *Comp. Vis. And Image Understanding (2015)* .

[6] Liyun Tu, Dan Yang, Jared Vicory, Xiaohong Zhang, Stephen M. Pizer and Martin Styner, “Fitting Skeletal Object Models Using Spherical Harmonics Based Template Warping,” *Signal Processing Letters*, volume 22, issue 12, Sept. 2015.

Chapter 2

Terms and definitions related to Pablo

2.1 The s-rep model: slab and quasi-tubes

The s-rep is the representation of a 3D object in by a skeletal model, which captures the boundary and interior of anatomic objects and is especially suited for the statistical analysis from a population of these objects. A continuous s-rep consists of a folded skeletal locus, which is a sheet or curve, and non-intersecting spokes from the skeletal locus to the boundary such that the union of the spokes is the interior of the s-rep. As represented in the computer, the s-rep consists of an array of samples of the skeletal locus and spokes. A continuous s-rep can be interpolated from this discrete s-rep by a method described in [Tu & Vicory, CVIU]. The discrete s-rep consists of a sampled grid forming the folded skeletal surface, each with a spoke reaching from each grid point to an associated boundary point. When two (or at the object crest, three) spokes share a grid point, this collection of spokes is called an atom. In a skeletal atom one spoke points to the top half of the object, one spoke points to the bottom of the object, and if there is a third spoke at the fold of the sheet, it points to the object crest, connecting the top to the bottom. These spokes, approximately normal to the boundary and touching the boundary, contain all information representing the interior and surface structure of the objects.

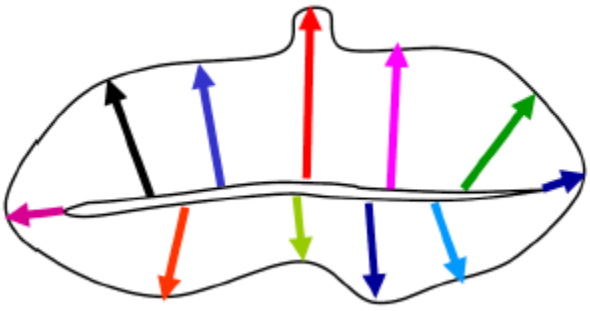


Figure 1 Skeletal model in 2D [1]

Figure 1 shows the 2D skeletal model (picture comes from Reference [1] listed in Section 1.4). From this picture, we can clearly see none of the spokes cross each other; the union of the

(interpolated) spokes forms the interior of the object, the union of the interpolated spoke tips forms the boundary, and the union of the interpolated spoke tails forms the skeletal surface.

The intrinsic geometric features of s-reps can be used in a variety of applications, such as classification, segmentation and registration, which achieved good performance. This software provides the functions to fit s-rep models to the anatomic objects, and to compute probability distributions on these s-reps. These can be used to train and classify different classes of the anatomic object, and to apply hypothesis testing to determine the most significant geometric variations between different classes, etc.

There are two types of s-reps as shown in Figure 2.

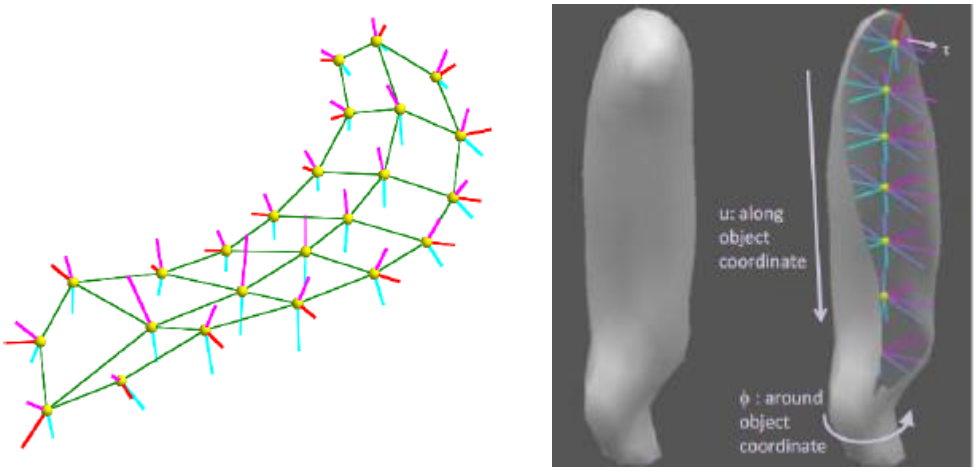


Figure 2 Slab (left) and Quasi-tube (right) s-reps [1]

2.1.1 Slabular s-reps

For slabular s-reps, shown in the left picture of Figure 2, it is formed by sampling a folded two-sided skeletal surface. At each atom (the yellow balls in Figure 2), there are two or three spokes pointing approximately normally to the boundary. Three types of spokes need to be taken into consideration: up-side spoke (the magenta spoke), down-side spoke (the cyan spoke), and the crest spoke (the red spoke), which only appears at the fold curve (the exterior of the grid). Suppose each side of the skeletal surface is sampled into an $m \times n$ grid, it is easy to get there are $m \times n$ up-side spokes and down-side spokes, respectively, and $2m + 2n - 4$ creast spokes. All the remaining spokes in the continuous domain are interpolated from the spokes in discrete s-reps.

The slabular s-reps are used to model the objects such as the kidney, most muscles, the heart, the bladder, etc.

2.1.2 Quasi-tube s-reps

For the quasi-tube s-reps, shown in the right picture of Figure 2, the skeletal surface takes the form of a bent cylinder of infinitesimal radius with hemispherical caps at both ends. The axis of the cylinder is a space curve, and the spokes emanate from each axis-orthogonal cut of the cylinder end in a common plane. The spokes in a given plane need not all be of the same length. Quasi-tube s-reps are used to model roughly tubular objects such as the esophagus and the rectum. Quasi-tubes are not fully supported in the distributed Pablo program. Contact J.C. Prieto at UNC if you want a version that supports quasi-tubes.

2.2 The standard orders of s-rep fitting: the s-rep initialization stage and the spoke stage

The aim of s-rep fitting is to find good s-rep models to match the binary images. It frequently involves two procedures: initialization stage and spoke stage. Another stage that may be used is the shape space stage, and infrequently, with manual corrections, the atom stage. Details in fitting will be talked about in Section 5.2.

2.2.1 S-rep initialization stage

Before fitting, a reference s-rep needs to be provided. However, this initial s-rep is only a common template and may disagree with the target binary image with large differences. In this condition, fitting directly using this template may get a local optimum.

A step called s-rep initialization stage deals with this situation [3]. In this stage, From both the s-rep template's implied boundary and target binary image's boundary a predecessor program to Pablo first selects a set of landmarks based on the thin-shell demons method [Zhao] or spherical harmonics [Styner, MICCAI]. From these boundary point pairs, a thin plate spline transformation is computed. Then the boundary-end and skeletal-end points of each spoke of the reference s-rep are transformed to produce an initialized s-rep.

2.2.2 Spoke stage

This is a stage following an initialization stage or other stages that provide an s-rep that needs refining. This includes the shape space stage (if statistics are provided) or an atom stage (if

statistics are not provided). In the spoke stage, spokes are treated independently. The restrictions of spokes like not crossing with each other, preferably being nearly normal to the boundary are applied here, but the main variation is the lengthening or shortening of the spoke to make the interpolated spoke ends fit the boundary well.

2.3 Other Pablo stages of s-rep fitting

2.3.1 Shape space stage

The shape space stage is based on the statistical analysis. Statistical analysis provides us a mean s-rep model and a limited number of important eigenmodes of shape variations. In this stage, what to optimize is the coefficients of these eigenmodes. A linear combination formed by the mean together with the weighted eigenmodes (in the Euclideanized space formed using CPNS statistical analysis) will yield a fitted model. It is easy to see all the atoms in s-reps will change together in this optimization.

The shape space stage is preceded by CPNS analysis of s-reps that have fit their respective images. This analysis, described in section 6.1.1, yields the mean, eigenmodes, and principal variances needed by the shape space stage.

2.3.2 Atom stage

In the unusual cases that the initialization stage followed by spoke stage refinement produces an inferior fit, a shape space stage followed by a spoke stage refinement can be attempted. Only when both of these approaches fail to produce a satisfactory s-rep fit to the input image is an alternative repair mechanism necessary. In this approach certain of the skeletal atoms are manually put roughly into place, and then the atom stage is put to use. Several parameters are related to the spokes at each atom: three parameters for the atom's skeletal position, three parameters for each of the two or three spoke directions, and two or three parameters for the spoke lengths. The atom stage is a process to optimize these parameters, atom by atom. One atom is randomly chosen while all the other atoms are fixed, and all the spokes at this atom will change together. Then the optimization moves to the next atom until covering all the atoms at least twice.

All successful fits can be analyzed by CPNS, and the result can be used with the shape space stage to obtain better fitted s-reps for the previous failures.

Chapter 3

Getting Started

3.1 Where to download Pablo?

To get the access of Pablo, the users will need to register an account on www.nitrc.org. From there, go to the Pablo page at http://www.nitrc.org/projects/sreps, where both a Linux and a Windows version, and some other related tools are provided.

3.2 Required external libraries

Pablo is running based on some external libraries. Before installing Pablo, install these libraries first:

- Insight Segmentation and Registration Toolkit(ITK) v4

  ITK is an open-source, cross-platform tools for medical image analysis. It can be downloaded at http://www.itk.org/ITK/resources/software.html.

- Visualization Toolkit(VTK) v5.8

  VTK is an open-source, freely available software for 3D computer graphics, image processing, and visualization. The latest version of this software can be obtained at http://www.vtk.org/download/. Note that version 5.8 is used by the Palo team. A different version could give problems.

- Fast Light Toolkit(FLTK) v1.1.10

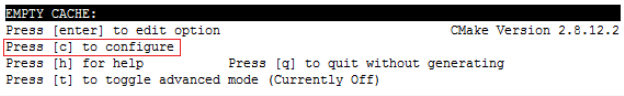
  FLTK is a cross-platform C++ GUI toolkit. It provides modern GUI functionality without bloat and supports 3D graphics via OpenGL and its built-in GLUT emulation. Version 1.1.10 can be downloaded at http://www.fltk.org/software.php .

3.3 How to install Pablo?

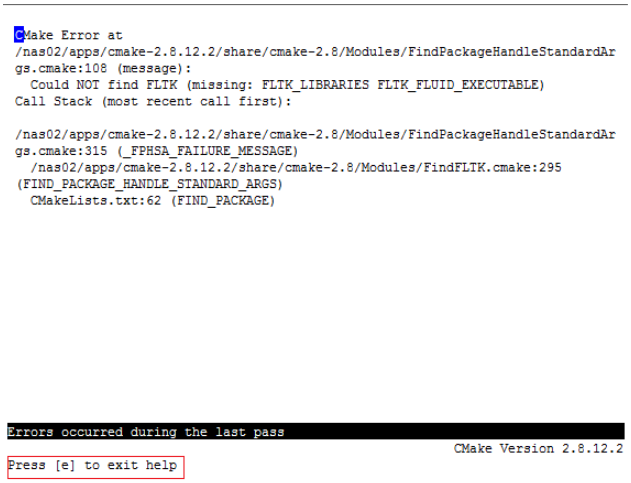
If the users download the Windows version, just open the Pablo folder and double click "*bin_pablo.exe*". Then the interface for Pablo will pop out.

If the Linux version is downloaded, it needs to be built from the source. Explained below is how to perform this process.

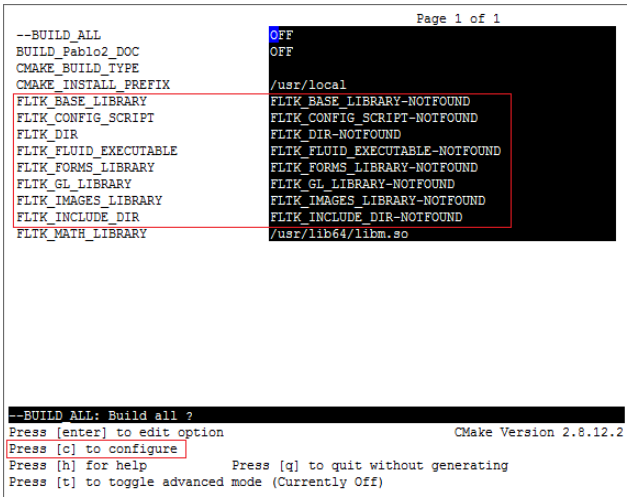
1. Create a build directory for ITK. Go to this directory and use the *ccmake* command ("*ccmake* ***sourcefile_directory***") and *make* command to build ITK. Press "c" for configure and "g" for generate when the screen asks to do that.
2. Create build directories for VTK and FLTK the same as the steps of ITK.
3. Create a build directory for Pablo. Go to this directory and type "*ccmake* ***sourcefile_directory/Pablo2***". Press "c" when necessary.



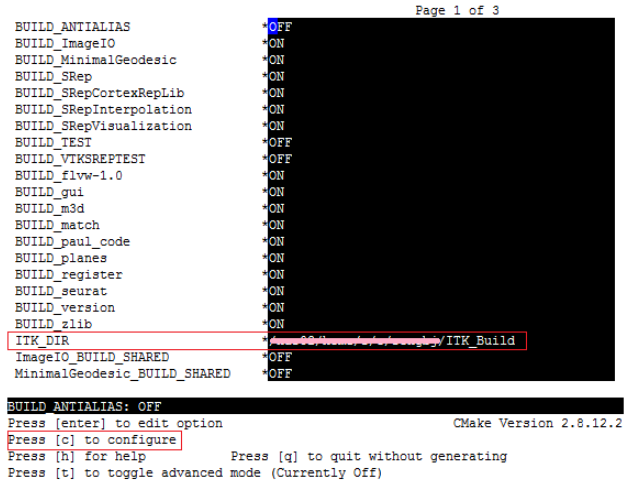
4. Then the user will receive an error message, which says "FLTK can't be found". Press "e" to give the path of FLTK to Pablo manually.



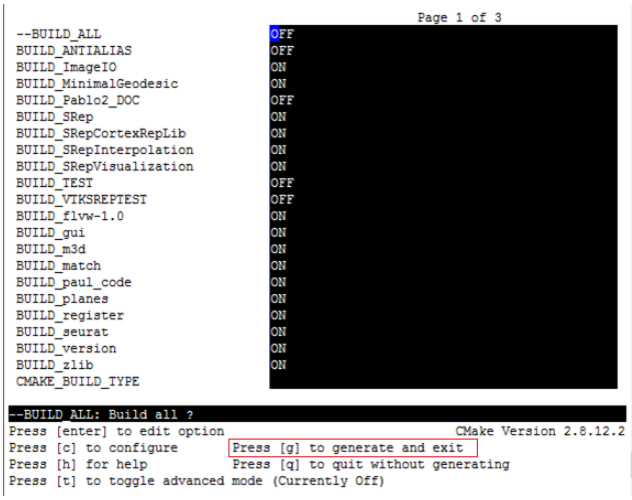
5. Fill the entries in the red box in the picture below as the build directory of FLTK. Then press "c" to continue.



6. Configuration stops at this screen. We can see Pablo finds the correct path of the ITK. Press “c”.



7. Press “g” to generate and exit.

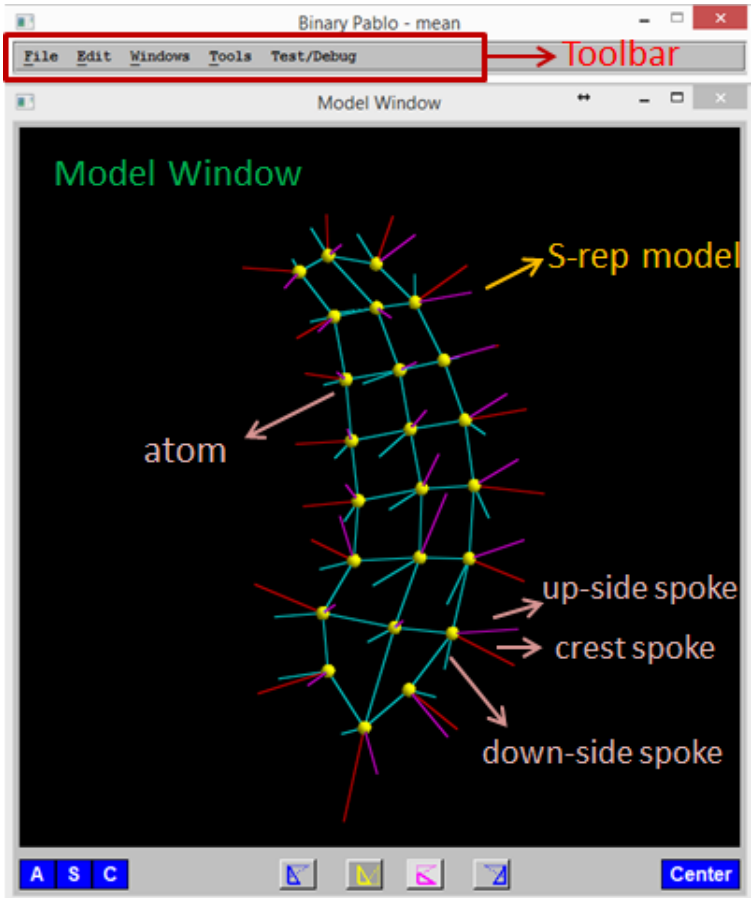


8. Then type "*make*" at the command line. It will reach 100% if succeeds.

Chapter 4

Pablo Graphics-User Interface (GUI)

In this chapter, the Pablo GUI will be introduced in details. To run Pablo, for the windows users, go to the downloaded directory of Pablo and double click the executable file "*bin_pablo.exe*"; for Linux users, go to the build directory of Pablo and type "*./bin_pablo &*". If the user is connecting remotely to a Linux machine from a Windows machine, the X-Win32 software must be loaded on the Window machine.



4.1 Mouse Operations on S-rep Model

This part describes the mouse actions on the s-rep model in the Model Window.

- Basic operations on the whole model
  - Rotate a model: drag with mouse left button

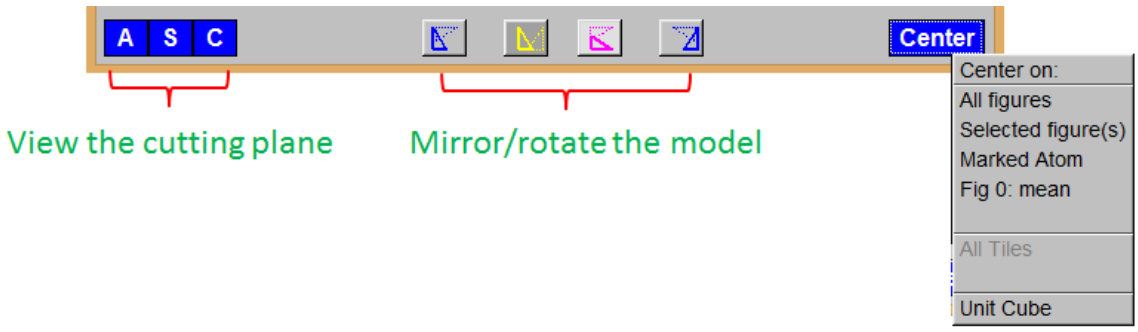
  - Zoom in/out a model: drag with mouse right button
  - Pan a model: drag with mouse wheel. For the 2-button mouse, hold down the left shift key and drag with mouse left button at the same time.
- Select/deselect the atoms
  - Hold down the “Ctrl” key and drag with mouse left button. Users will see a black dotted line indicating the region. All the atoms within the rectangle will be affected. An unselected atom’s center is a small white ball. When selected, a large yellow ball is drawn.
- Operations on the selected atoms
  - The basic operations (zoom in/out, rotate, pan) will only work on the selected atoms when moving the mouse and holding down the “Alt” key (or Ctrl and shift keys in the Linux version of Pablo) at the same time. That is to say, hold down the “Alt” and drag with mouse left button, right button and wheel to rotate, zoom in/out, and pan the selected atoms.
- Operations on the spoke length of selected atoms
  - Hold down the “Shift + Alt” at the same time and drag with mouse right button to change the spoke lengths of the selected atoms. All the spokes of the selected atoms will be changed together.

4.2 Model Window

- View the s-rep models

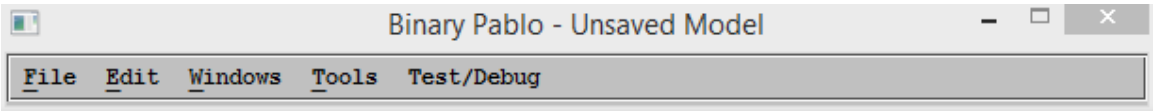
  S-rep models can be visualized in the Model Window. Users can interactively manipulate the pose/position/zoom of the models by translating, rotating, scaling, selecting and marking the atoms or spokes. See Section 4.1 Mouse Buttons.
- View the binary images

  Binary images (with value 1 inside the object and 0 outside) and its tile set can be also viewed in the Model Window.



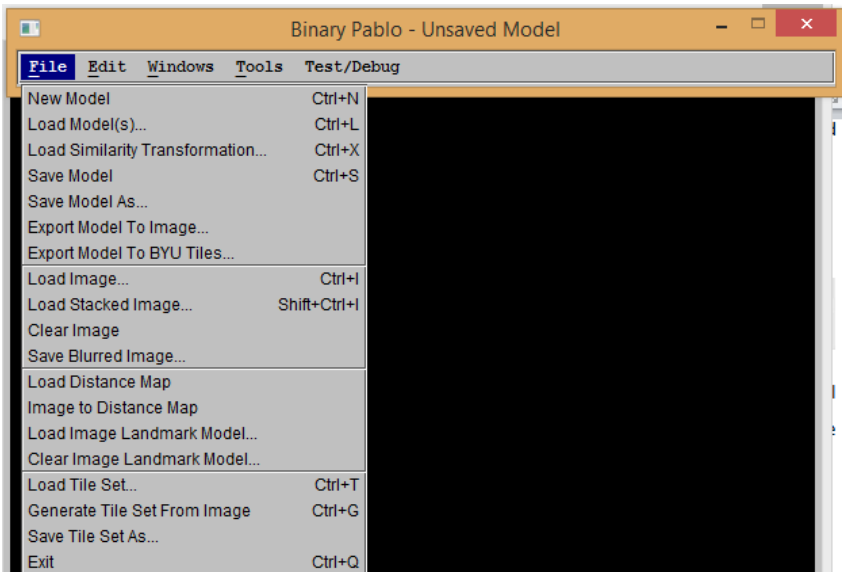
At the bottom of the model window, there are some shortcut icons that can be used. On the left, button A, S and C allow users to view the axial plane, sagittal plane and coronal plane of the image that has been loaded. In the middle, there are four buttons that allow users to mirror or rotate the s-rep models or images. On the right, clicking on “Center” will pop up a drop-down list. It centers the models and images on the user-defined positions, with possibilities including all figure, selected figures, marked atom, etc. The one that most frequently used is center on all figures.

4.3 Menu bar



In the menu bar, five drop-down menus are here: File, Edit, Windows, Tools, Test/Debug. They will be described in the following subsection respectively. The name of the current model (like Unsaved Model in the above picture) is displayed in the title bar.

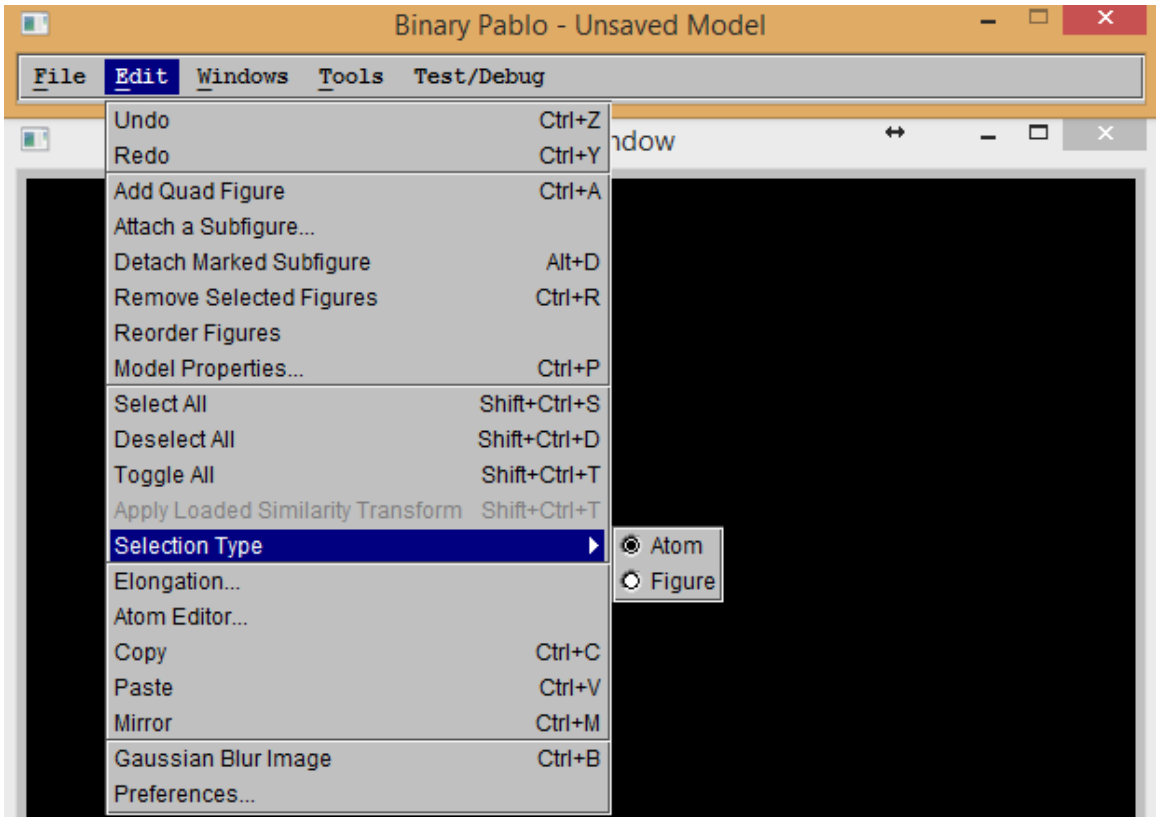
4.3.1 File



- **File->New Model** deletes the current model(s).
- **File->Load Model(s)…**reads an s-rep model (*.m3d) from disk. If one model is already loaded, replace the current model. Be careful if it asks whether to save changes to the current model.
- **File->Load Similarity Transformation…** reads a similarity transformation from disk in order to be performed on the current model. Before loading the transformation file, a model must have been already loaded.
- **File->Save Model** creates or overwrite a file with the current model. Note that this new file will record the selected and marked atoms.
- **File->Save Model as…**is same as “Save Model” except that it pops out a file browser to save the current model according to a user designated path rather than the file which the model was read from.
- **File->Export Model to Image…**converts the current s-rep model (*.m3d) to a binary image file (in the format *.mhd and *.raw together) containing a pixel value of 1 inside and on boundaries and 0 elsewhere. This is used when some operations need to be done on the interior of the s-reps-implied object interior. So instead they need to work on the binary image.
- **File->Export Model to BYU Tiles…**writes the s-rep model to a .byu file, which represents the tile set for the implied boundary.

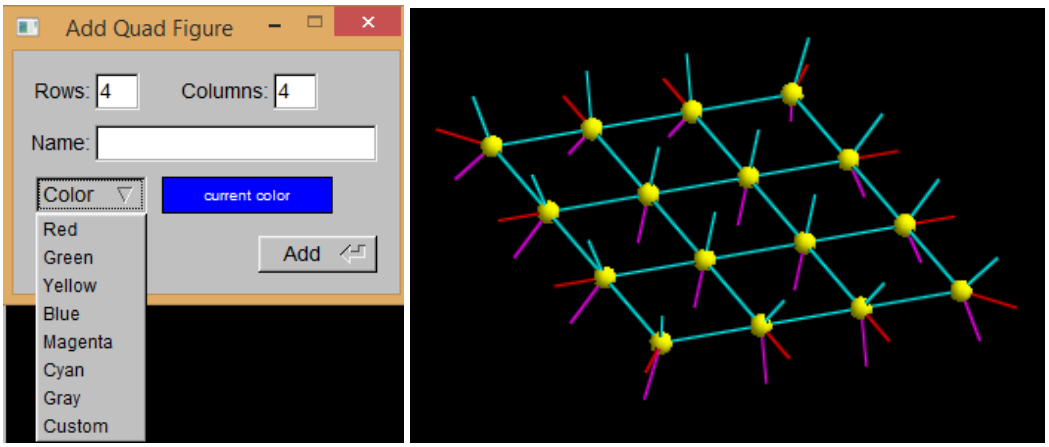
- **File->Load Image…**loads the binary image from disk. If one binary image is already loaded, just replace it. Pablo accepts the binary images with the filename extensions like .raw3, .pim, .gipl, .mha, .mhd, etc.
- **File->Load Stacked Image…**this function has been deprecated and should not be used.
- **File->Clear Image** deletes the current image.
- **File->Save Blurred Image…** creates a file to record the newly blurred images which is produced by *Edit->Gaussian Blur Image*
- **File->Load Distance Map** loads the distance image with the zero level surface representing the object. This function is presently not functional.
- **File->Image to Distance Map** converts the binary image to distance map. This function is presently not functional.
- **File->Load Image Landmark Model…**loads the image with landmarks, which correspond to specific s-rep spokes.
- **File->Clear Image Landmark Model…**deletes the landmarks in an image.
- **File->Load Tile Set…**loads an object boundary tile set from disk. If one tile set is already loaded, just replace it. The tile set is scaled to the same unit box as its corresponding image.
- **File->Generate Tile Set from Image** generates the tile set from the current binary image. The tile set is an arbitrary set of 3D triangles representing a 3D surface.
- **File->Save Tile Set as…**saves the current tile set to disk.
- **File->Exit** exits the program. Be careful when it asks whether to save changes to the current model.

4.3.2 Edit

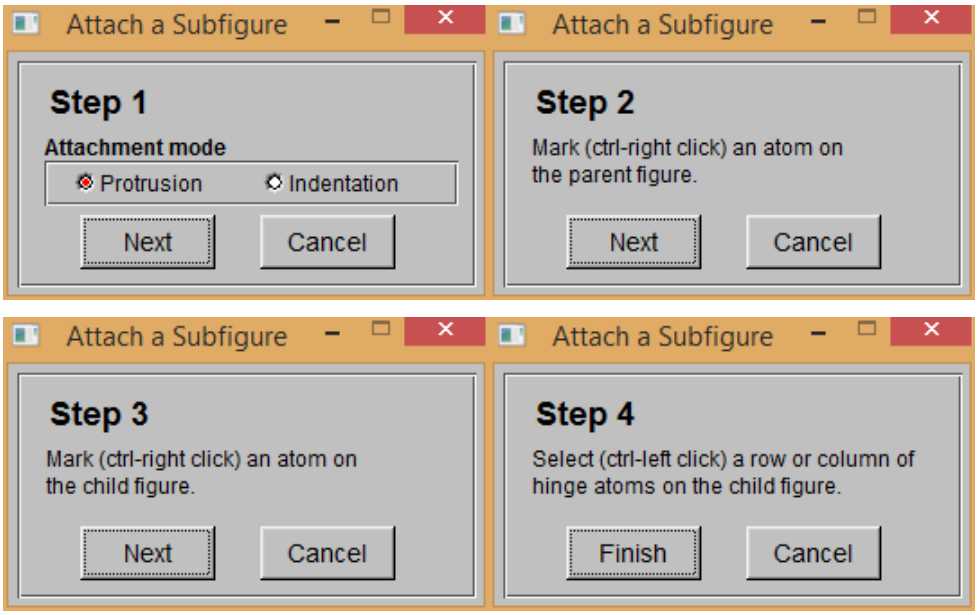


- **Edit->Undo** cancels the last edit made to the atoms, as if the last edit never happened. Multiple edits (the maximum value can be modified in *Edit->Preferences->Other*) can be undone in succession. Almost all edits can be undone, including *Add Quad Figure…, Remove Selected Figures…, Atom Editor* and *Load Model*.
- **Edit->Redo** reverses the effect of an Undo.
- **Edit->Add Quad Figure*…** creates a skeletal model of size $m \times n$, where m is the number of rows and n is the number of columns. Colors and text names can be assigned to the models. Click “Add” to create the model. This dialog box won’t disappear after clicking the “Add”, so multiple models can be added efficiently.

  *To better understand the atom editing, two properties of an atom need to be defined: atom selection and atom marking. Only selected atoms can be edited; the other atoms (including the marked ones) won’t be changed during the process of editing. Marking function is used to mark an atom. The details of how to select/unselect/mark/unmark an atom and how to edit an atom will be illustrated in Section 4.7.

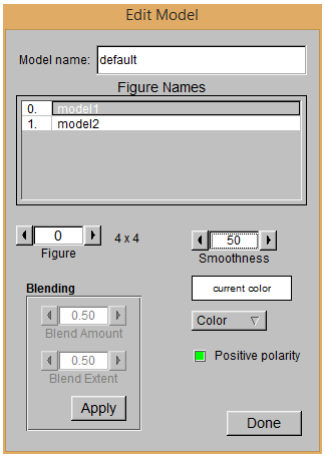


- **Edit->Attach a Subfigure…** merges two independent models into a parent-child relationship. Users can do it following the instructions Pablo pops out.

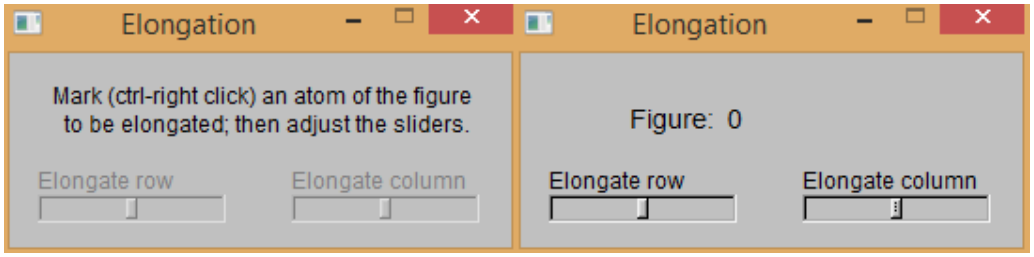


- **Edit->Detach Marked Subfigure** detaches a figure with the parent-child relationship to two independent models. Before detaching, users need to mark an atom on the subfigure to be detached.
- **Edit->Remove Selected Figures** deletes the selected figures. Hold down "Ctrl" key and drag with mouse left button to select a model, and then remove it. If parts of the model's atoms are selected, this model can't be removed. Because the subfigure capabilities of Pablo are not now fully functional, this feature should presently not be used.
- **Edit->Reorder Figures** reorders all the models. Users need to have multiple models before reordering them.

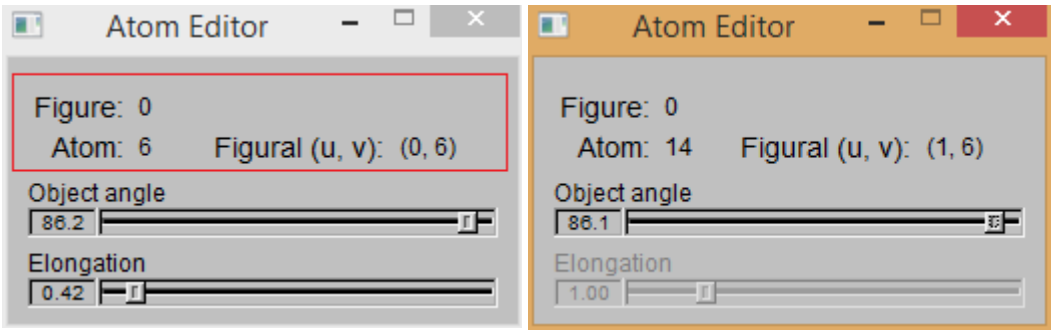
- **Model Properties…** displays a dialog box to show and edit the properties of the models, including the model names, model colors, smoothness, and the size of models (the size only be read; it can't be edited). The changes will take effect until "Done" button is pressed. Models can also be named and colored when they are created by *Add Quad Figure.*



- **Edit->Select All** selects all the atoms in the models regardless of their current selection state.
- **Edit->Deselect All** deselects all the atoms in the models regardless of their current selection state.
- **Edit->Toggle All** reverses all the atoms' selection state (select the unselected atoms, and deselect the selected atoms). This is useful when performing some operation on one set of atoms and then another operation on the remaining atoms.
- **Edit->Apply Loaded Similarity Transform** applies the loaded similarity transform (*File->Load Similarity Transformation)* to the current models.
- **Edit->Selection Type** controls what can be selected with one click in the Model Window. If Selection Type is set to Figure, users can select/deselect an entire model with only one click on one atom. This helps users to select/deselect multiple models easily and quickly. If Selection Type is set to Atom, only one atom will be selected or deselected at one time.
- **Edit->Elongation…** elongates/shortens a row and/or column of the model. Users first need to mark an atom to activate this dialog box, and then they need to drag the slider to elongate/shorten the row and/or column.



Edit->Atom Editor… edits atoms. Mark an atom that needs to be edited to activate the dialog box. After the scrollbar is activated, users can move the slider and watch the changes of the atom in Model Window. The Model Window will be updated as users change the object angle and elongation. Note that elongation only is the parameter of crest spokes. If the atoms not located at the surface curve are selected, the elongation tab will become gray to be disabled.



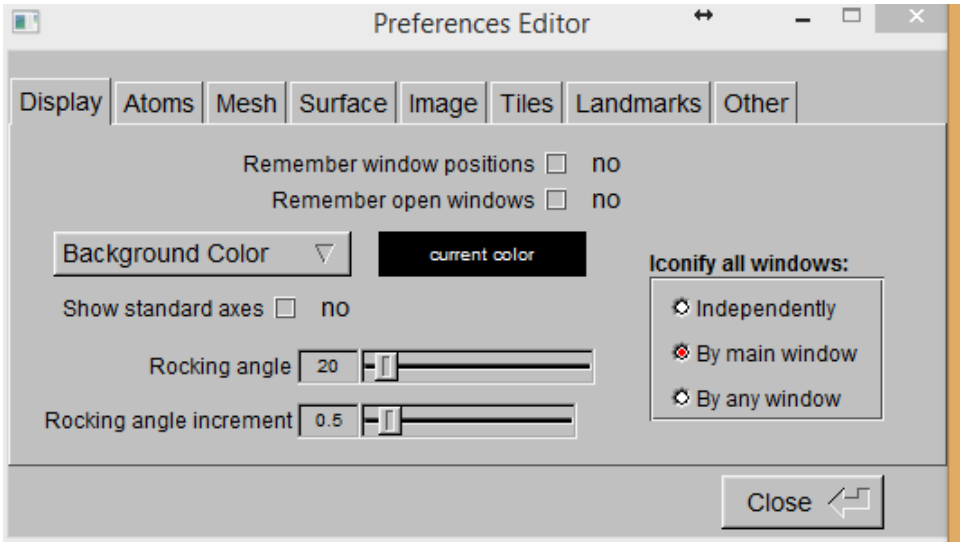
The object angle is half the angle between up-side and down-side spokes. Its range is from 0 degree to 90 degrees. 0 degree means the two spokes are overlapping, and 90 degrees mean spreading the two spokes as far as they will go. The elongation is the length of crest spoke. The larger this value is, the more highly curved the surface curve is at the crest.

The Atom Editor also provides the information of Figure ID, the Atom Id and the Atom position of the marked atom.

- **Edit->Copy** copies the current selected models into the cut buffer.
- **Edit->Paste** makes a copy of the models in the cut buffer. Pablo will deselect the models that are currently selected and then set the pasted models as the newly selected ones. The copy operation should always be followed by a paste operation.
- **Edit->Mirror** flips the selected models horizontally (around the X coordinate), not affecting the Y and Z coordinates. This operation is useful because the bilateral symmetry of vertebrates can be used to model a structure on one side of the body, and then mirror it as a starting point to model the structure on the other side of the body. If

mirroring around the other coordinates is necessary, rotate the model to make the mirror plane to be X coordinate, mirror it, and then rotate the model back to its original orientation.

- **Edit->Gaussian Blur Image** uses the Gaussian kernel to blur the images shown in the Model Window, making the images look smoother.
- **Edit->Preferences…** pops out a dialog box to set the preferences.
  - Display Tab



Remember window positions: records the size and the current position of each Pablo window for the next use when Pablo starts.

Remember open windows: records the Pablo windows which are displayed now for the next use when Pablo starts.

Background Color: changes the background color of the Model Window. The default color is black.

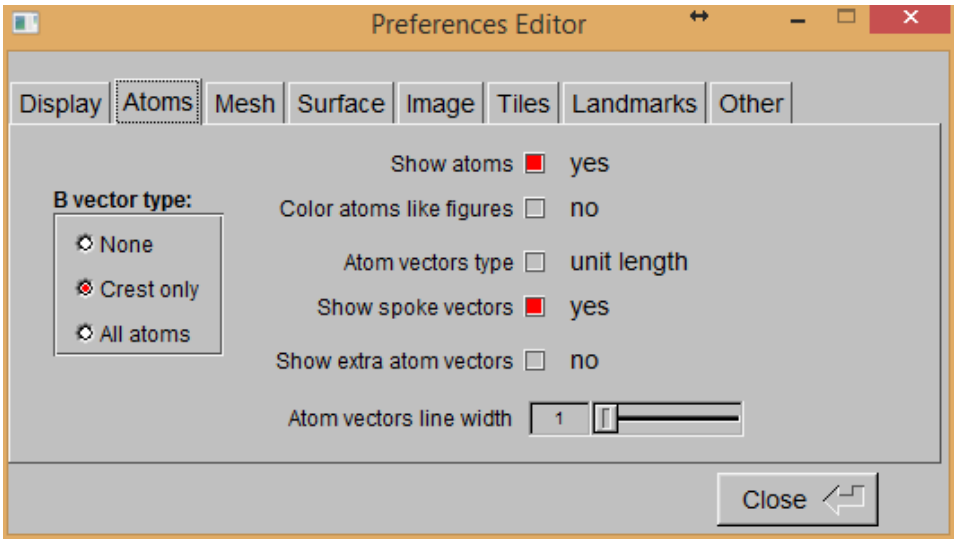
Show standard axes: shows the colored primary X, Y, Z axes. X is colored red; Y is colored magenta; Z is colored blue. The plane composed by X and Y axes is considered to be co-planar with the axial slices of the 3D images, and the Z axis is orthogonal to these slices.

Rocking angle: controls the degrees that the camera will rock (rotate from one side to the other side) during display of an s-rep. See the rock motion mode in the "*Display control tab*" in section 4.4.

Rocking angle increment: controls the rotational degrees the camera will rock in each frame of the rocking sequence. The larger this number is, the faster the camera will rock.

Iconify all windows: controls the ways to maximize, minimize or restore the windows. If “Independently” is selected, treat every window independently. If “By main window” is selected, bring the window to full size of small size or restoring the window size will work on all windows, while these operations on the other windows will only work on themselves. If “By any window” is selected, the operations on any window will work on all the windows.

- Atoms Tab



B vector type: B vectors are the vectors that bisect the upside and downside spokes for that atom. This item controls which atoms these vectors are displayed for. It can be defined as “None”, “Crest only” and “All atoms”. The default should be “crest only”.

Show atoms: controls whether to show the atoms in the Model Window.

Color atoms like figures: controls whether to set the atoms’ color to their figure’s color.

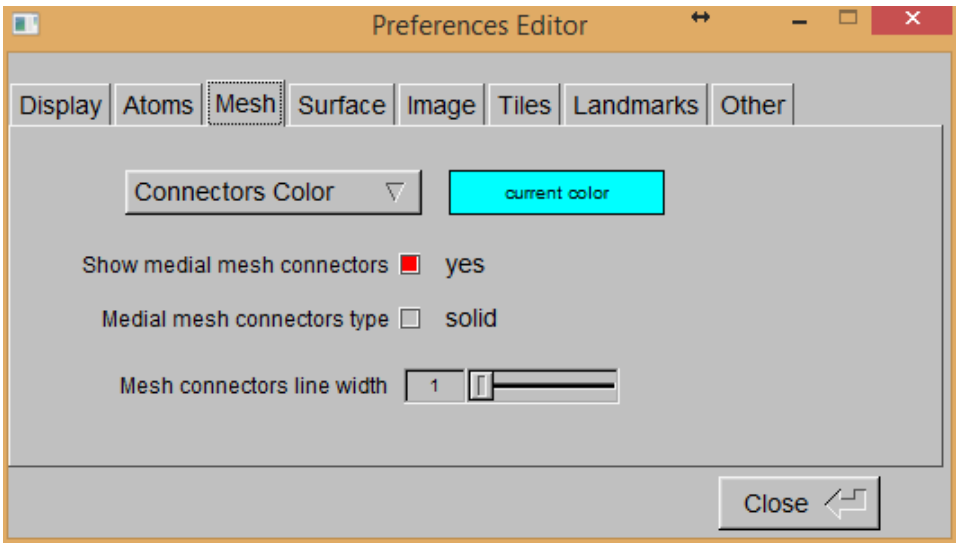
Atom vectors type: This command has been deprecated.

Show spoke vectors: controls whether to show the spokes in the Model Window.

Show extra atom vectors: controls whether to show the extra atom vectors, which are the normal to the skeletal sheet at each atom.

Atom vectors line width: adjusts the thickness of the spokes.

- Mesh Tab



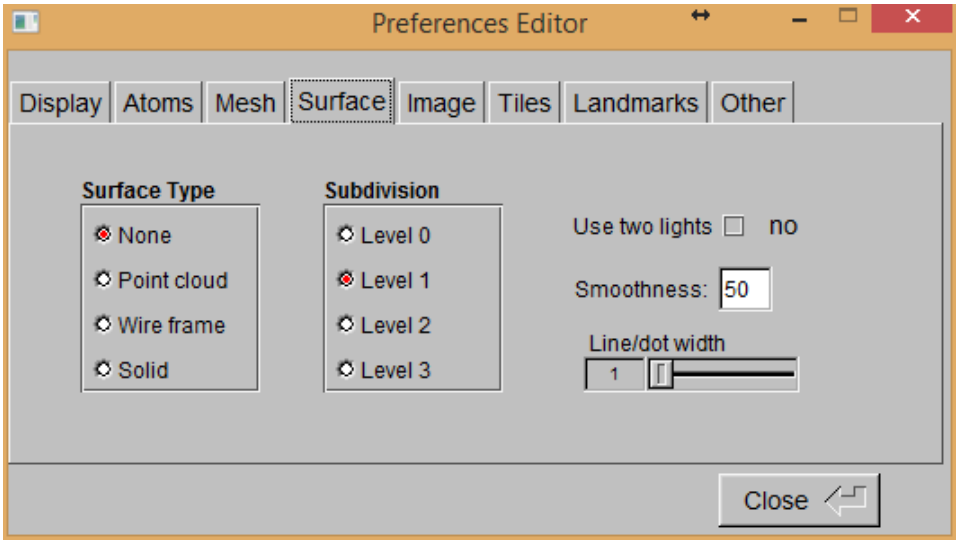
Connector Color: changes the color of the mesh which is connecting the atoms. All references to the "medial mesh" should be understood as "skeletal mesh".

Show medial mesh connectors: controls whether to draw the mesh which is connecting the atoms.

Medial mesh connector type: changes the connector type between solid and dotted.

Mesh connectors line width: adjusts the thickness of the connectors.

- Surface Tab



Surface Type: selects the visualization type of the implied boundary by s-reps.

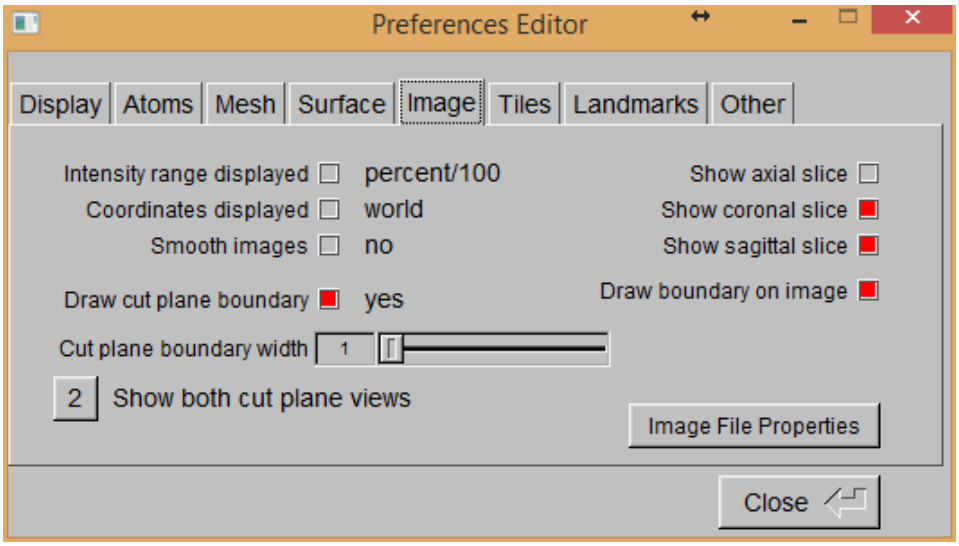
Subdivision: implies the level that the space between neighboring spokes is divided to form the surface or tile set. The larger this number is, the finer the surface is and the more time it takes to compute. Level 0 specifies that the mesh shows just the spokes in the computer representation of the s-rep.

Use two lights: changes the camera model to two source lights.

Smoothness: sets the smooth parameter to smooth the surface.

Line/dot width: sets the thickness of the surface line/dot.

- Image Tab



Intensity range displayed: this command determines whether image intensities are displayed as their raw values or as percentage of the maximum intensity in the image.

Coordinates displayed: changes between the world coordinate and model coordinate.

Smooth images: controls whether to smooth the image.

Show axial slice: controls whether to show the image's axial slice.

Show coronal slice: controls whether to show the image's coronal slice.

Show sagittal slice: controls whether to show the image's sagittal slice.

Draw cut plane boundary: controls whether to draw the intersection of the object boundary with a cut plane when displaying the image in the cut plane (see *Windows->Atom Cut Planes* in section 4.3.3).

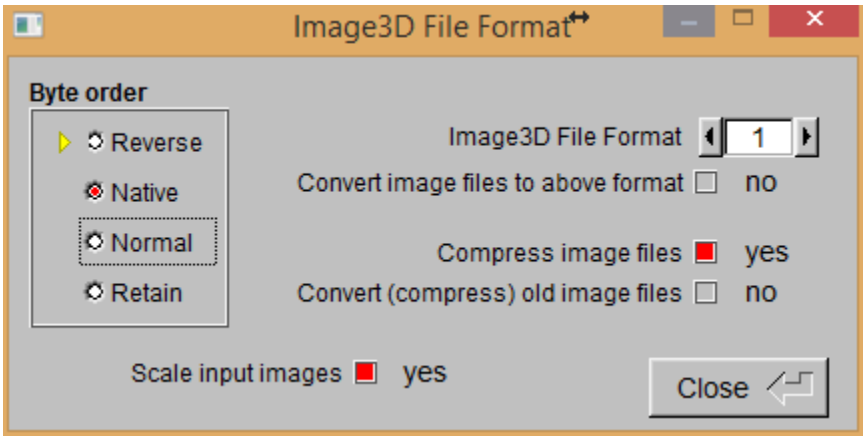
Cut plane boundary width: sets the thickness of the object boundary displayed on the cut plane.

Draw boundary on image: controls whether to draw on the image the s-rep-implied object boundary's intersection with the displayed image slice.

Show both cut plane views: controls where to show the cut planes. Change among "show only cut plane windows", "show cut planes only in model window" and "show both cut plane views". See *Windows->Atom Cut Planes* in Section 4.3.3 for more information.

Image File Properties

In this tab, there are various settings for how images are saved.



Byte order: the order that the bytes are stored in.

Image3D File Format: this command was created to allow the use of an old format for the Image3D File when a new format became the default. Consider this command deprecated.

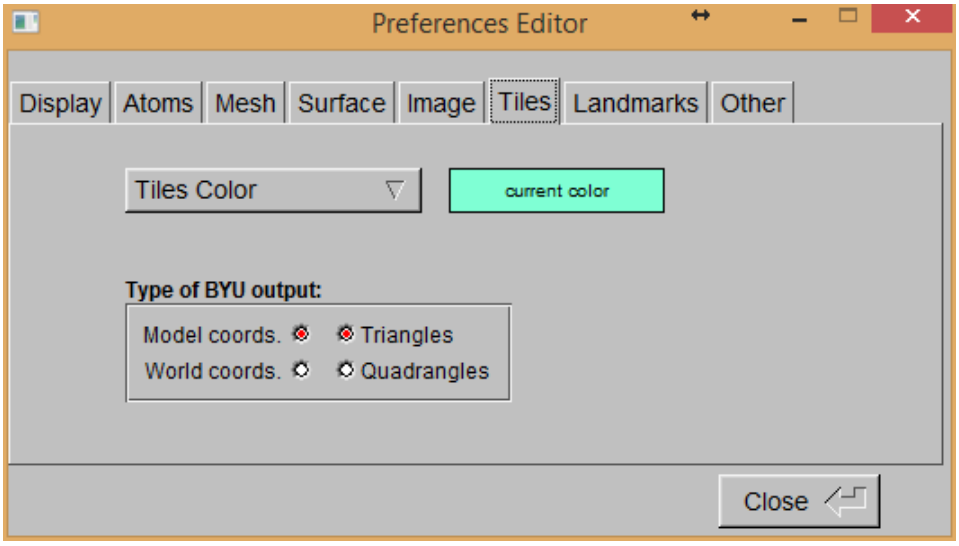
Convert image files to above format: controls whether to convert image files to the format set above. Just as above, consider this command deprecated.

Compress image files: controls whether to compress image files.

Convert (Compress) old image files: controls whether to convert the old image flies.

Scale input images: controls whether to spatially scale input images to the $[0,1]^3$ space native to Pablo. The default is to do that scaling.

- Tiles Tab

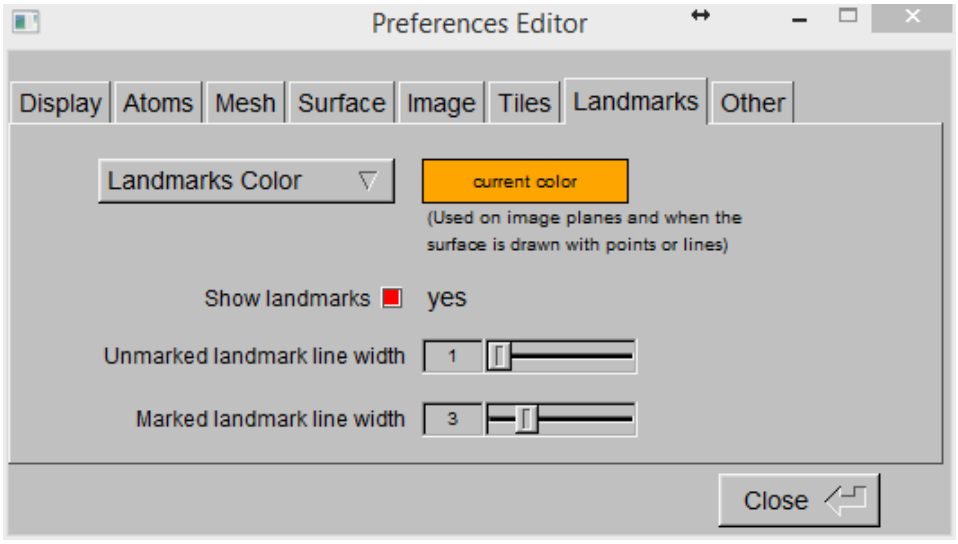


Tiles Color: sets the color of tile set.

Type of BYU output: sets the type of BYU output. Users can choose the coordinates from Model coordinates and World coordinates. Also, the mesh of the tile sets can be set to a Triangle mesh or Quadrangle mesh.

- Landmarks Tab*

*Landmarks are a set of sparely sampled well-corresponding points on organs or regions of interest across a population of shape instances. They can be selected manually or automatically. Landmark is very important to various kinds of researches such as the registration, the statistical analysis of the differences among sets of similar shapes, etc.



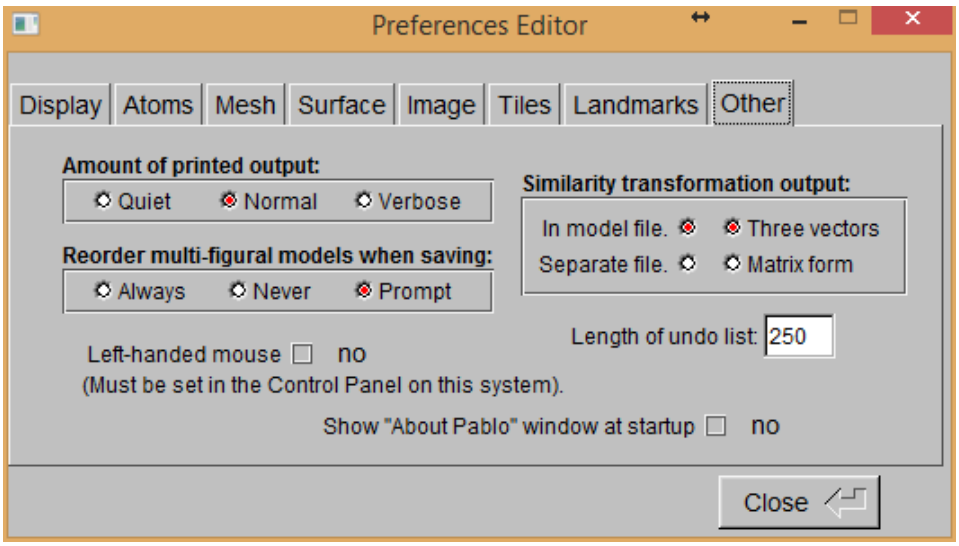
Landmarks Color: sets the color of landmarks.

Show landmarks: controls whether to show landmarks.

Unmarked landmark line width: controls the thickness of the unmarked landmark lines.

Marked landmark line width: controls the thickness of the marked landmark lines.

- Other Tab



Amount of printed output: controls how much information will be shown in the console window when Pablo is running. “Quiet” means no output. “Verbose” will print a lot more information about what it is doing than “normal”.

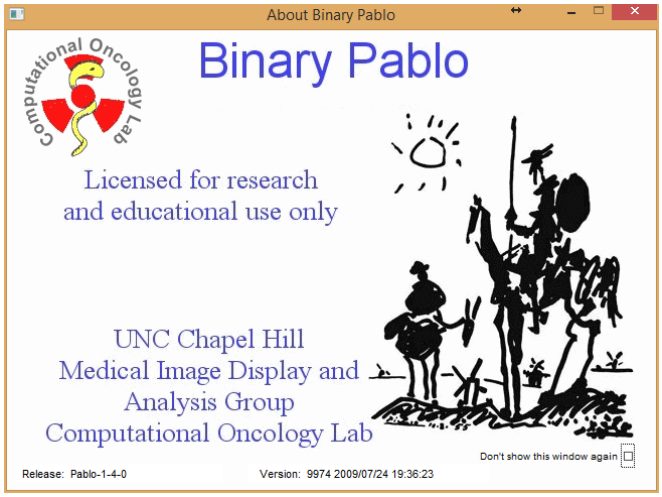
Reorder multi-figural models when saving: if an anatomic entity is made from two or more non-interacting s-rep grids (one is not a subfigure of another), this feature controls how to reorder the models when saving them.

Similarity transformation output: controls where and in what form to save the similarity transformation output.

Left-handed mouse: used for the left-handed users. This must be set in the control panel on the system.

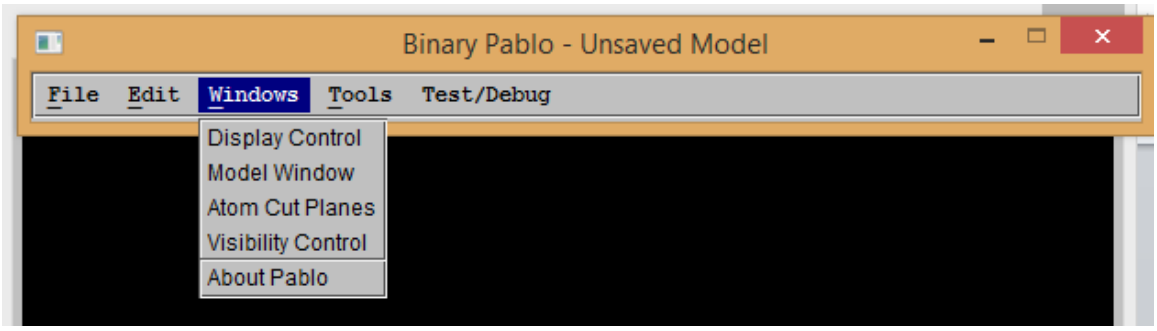
Length of undo list: sets the length of the undo list in *Edit->Undo*.

Show “About Pablo” window at startup: controls whether to show the “About Pablo” window at startup. If “yes” is selected, the below picture will be shown.



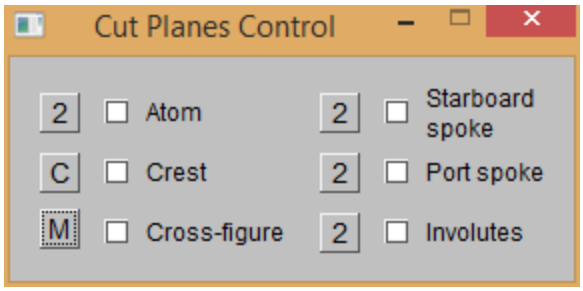
4.3.3 Windows

The Windows menu will show or hide some tabs.

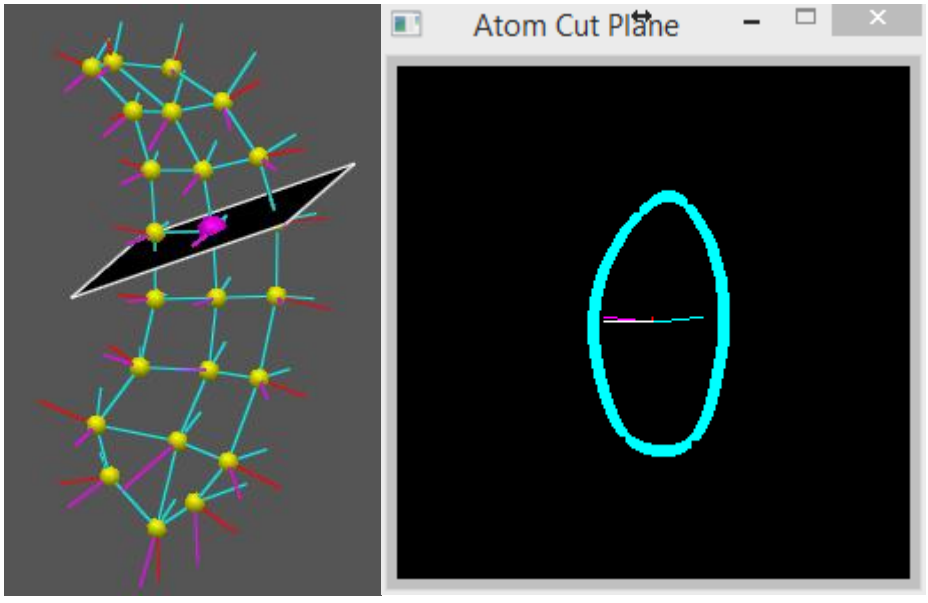


- **Windows->Display Control** shows/hides the Display Control tab. This part will be talked about in the Section 4.4 because it will be frequently used.
- **Windows->Model Window** shows/hides the Model Window, which is the place to view the s-reps, binary images, or the tile sets.
- **Windows->Atom Cut Plane*...** displays the checkboxes for six kinds of atom cut planes, which cut an atom and contain various different spokes. First mark the atom (Ctrl and right-mouse-button) whose cut planes need to be shown, and then this tool helps to visualize the image boundaries near the marked atom.

  * As shown in the example below, Pablo allows displays in planes defined by an atom. These planes are called “cut planes”. The image slice is displayed both relative to the model and face on.

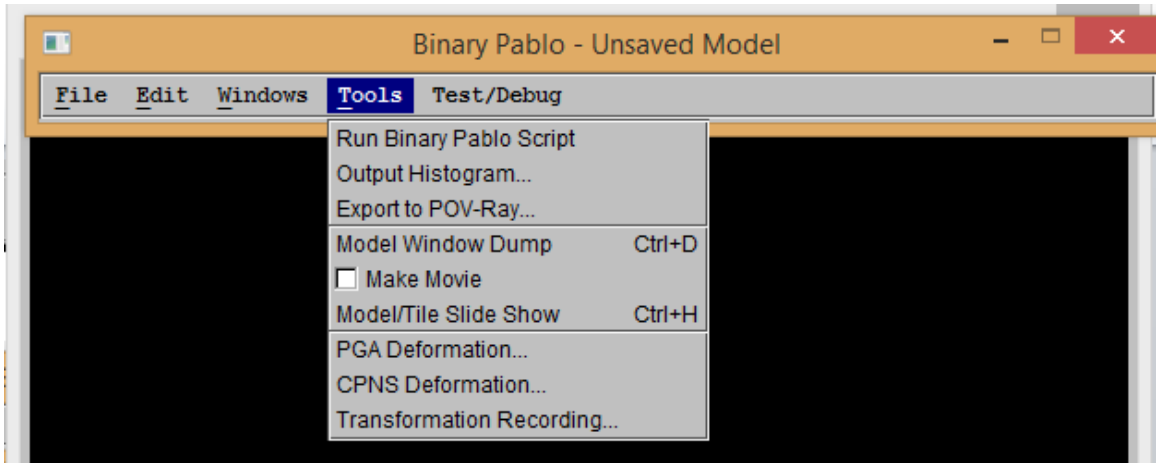


When any of these checkboxes is selected, a 2D plane of the image slice going through the marked atom will be displayed. The curve represents the implied boundary; the lines represent the spokes projecting on the 2D cut plane. The intersection between marked atom and the slab aligns with the atom's geometry. There are six kinds of atom cut planes (orientations): Atom cut plane, Crest cut plane, Cross-figure cut plane, Starboard spoke cut plane, Port spoke cut plane and involutes cut plane. There are three ways to show the cut planes: in the picture above, "C" means showing the cut plane in a separate window; "M" means showing the cut plane in the Model Window; "2" means show the cut plane in the above two windows.

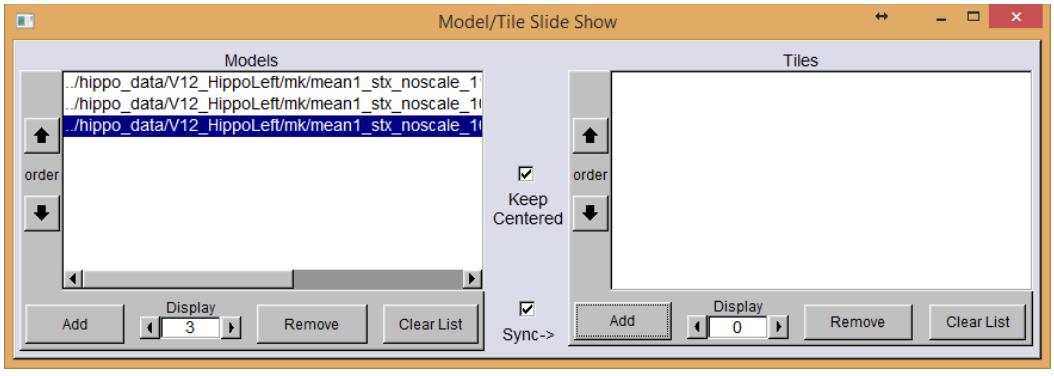


- **Windows->Visibility Control** shows/hides the visibility control tab. This part will be talked about in the Section 4.5 because it will be frequently used.
- **Windows->About Pablo** shows the information about Pablo, including the version, the developed team, etc.

4.3.4 Tools

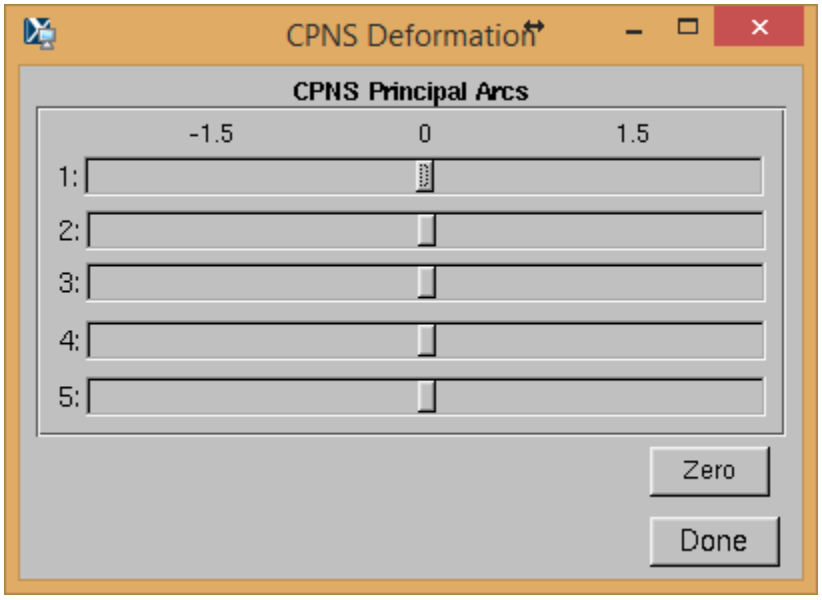


- Tools->Run Binary Pablo Script
- Tools->Output Histogram…
- Tools->Export to POV-Ray…
- **Tools->Model Window Dump** prints a screen shot of the Model Window.
- Tools->Make Movie
- **Tools->Model/Tile Slide Show…**helps users to view and compare the models and tile sets conveniently. Click the “Add” button to add all the models/tiles that need to be compared; click the “Removed” button to remove the unwanted ones. “Clear List” helps to remove all the models/tiles with one click. If the users want to reorder the models/tiles, click the upward arrow or downward arrow to change the order. Highlight one of the model/tile names and press the up/down arrow on the keyboard to see the model/tile variations in the Model Window.



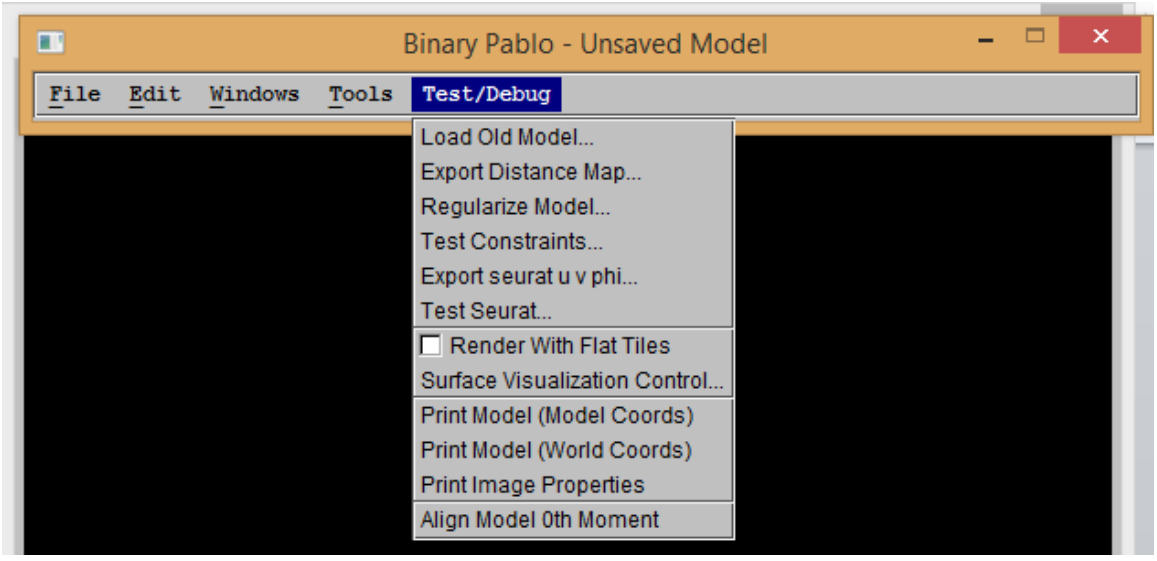
- **Tools->PGA Deformation…** is similar to “*Tool->CPNS Deformation…*”. It works with the statistical distribution generated by PGA, which is an older method.

- **Tools->CPNS Deformation…** a tool to observe the contributions of the first five Eigen modes generated by CPNS to the mean model. A model containing the distribution generated by CPNS is needed; otherwise this window will be disabled. CPNS is done over a number of fitted s-rep models. See Section 6.1 to see how to do CPNS. Press “Zero” button to set the sliders to zero position.



- Tools->Transformation Recording…

4.3.5 Test/Debug



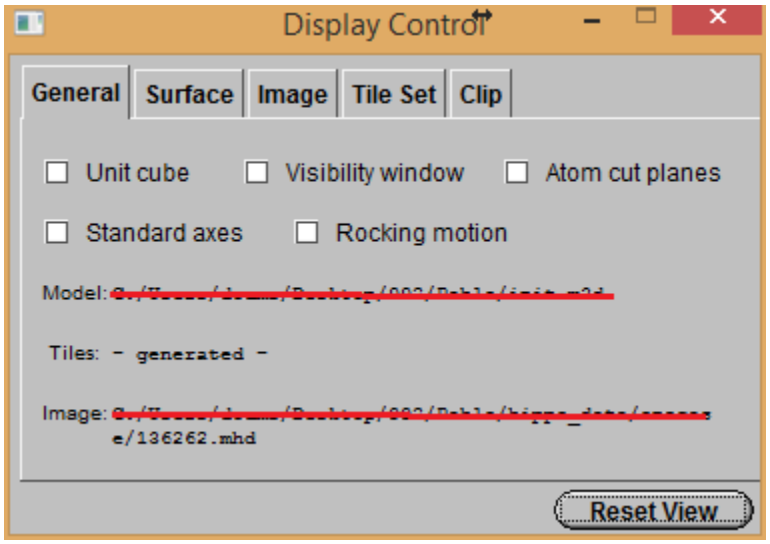
- **Test/Debug->Load Old Model…** loads a model with the format .mod, which is an older version than .m3d.

- **Test/Debug->Print Model (Model Coords)** prints the model information at the console under the model coordinates.
- **Test/Debug->Print Model (World Coords)** prints the model information at the console under the world coordinates.
- **Test/Debug->Print Image Properties** prints the binary image information at the console.

4.4 Display Window

Display Window is one of the most frequently used windows in Pablo. It controls the appearance of the objects in the Model Window.

- General Tab



Unit Cube: shows a unit cube with white edge at the original point.

Visibility window: pops out the Visibility Window. See Section 4.5.

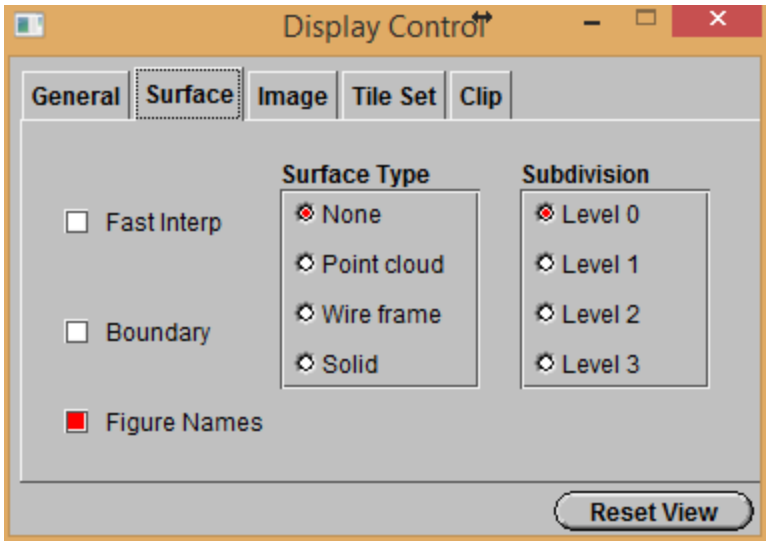
Atom cut planes: pops out the Atom Cut Plane Window. See Section 4.3.3.

Standard axes: shows the colored primary X, Y, Z axes.

Rocking motion: continually rotates the camera to change the view point. It helps users to capture smooth model animations or observe the models/images/tile sets.

Reset View: sets the camera parameters to the default values.

- Surface Tab



Fast Interp: uses the fast interpolation algorithm to speed up the render of the implied boundaries by s-reps.

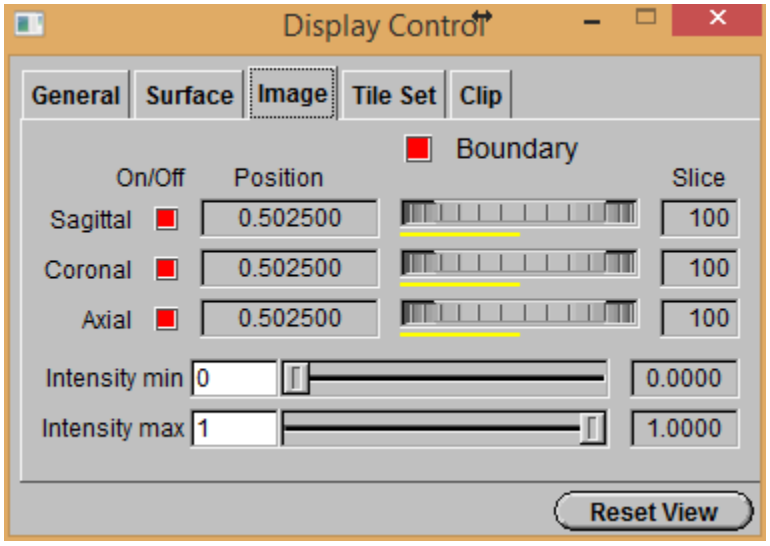
Boundary: shows the 2D intersection of each displayed image X-Y-Z plane with the model's surface.

Figure Names: controls whether to show the model names.

Surface type: displays a surface interpolated from the spokes. The surface type includes None, Point cloud, Wire frame, or a tiled opaque surface.

Subdivision: implies the level that the space between neighboring atoms is divided to form the surface or tile set, that is, the density of the dots or tile vertices. The larger this value is, the smoother the surface is and the more time it takes to compute.

- Image Tab



Boundary: shows the 2D intersection of each displayed image X-Y-Z plane with the model's surface. It's the same button as the Boundary in the *Surface tab*.

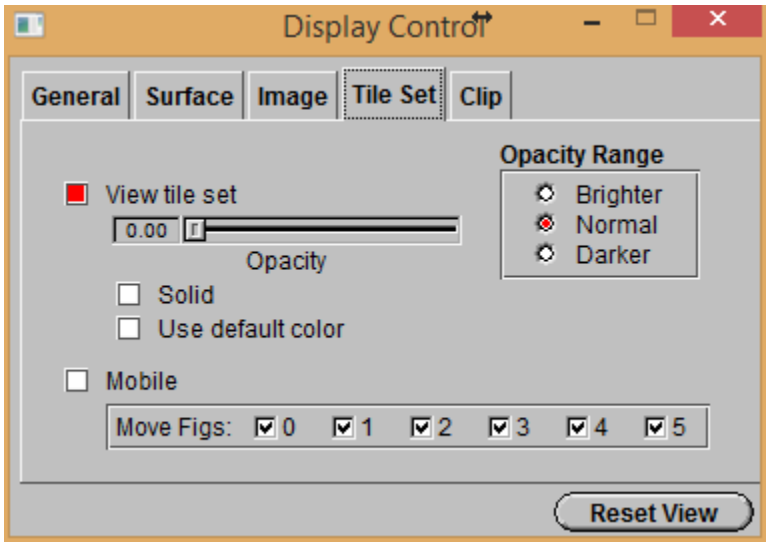
On/Off: controls whether to show the three cutting planes (Sagittal, Coronal, and Axial).

Position: drags the slider to control the positions of the cutting planes. The images live in the $[0, 1]^3$ cubic, and the range of cutting planes is [0.0025, 0.9975].

Slice: Pablo numbers each cutting plane from 0 to 199.

Intensity min/max: sets the minimum and maximum intensity values of the images. Draw the pixels below the min as black, above the max as white.

- Tile Set Tab



View tile set: controls to show/hide the tile set.

Opacity: sets the transparency of the tile set. 1 makes the tile set opaque and 0 makes the tile set invisible.

Solid: sets the surface of the tile set to solid surface.

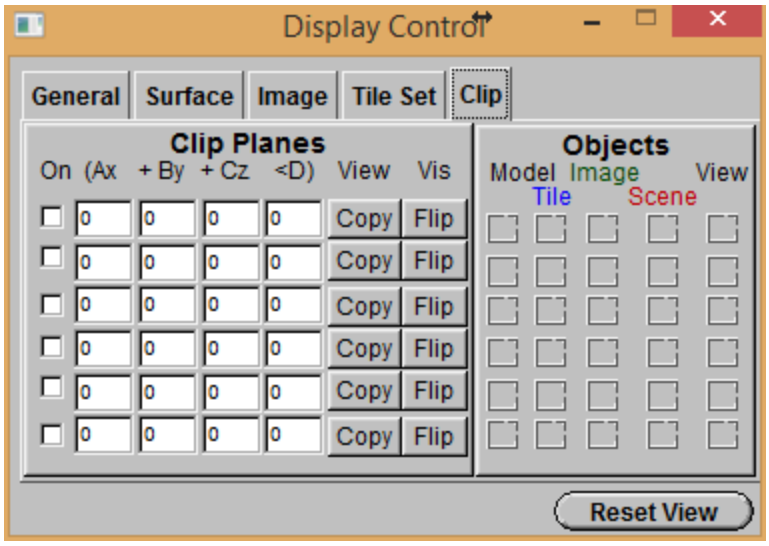
Use default color: controls whether to set the tile set color to the default color.

Opacity Range: sets the opacity range. The order of the brightness is Brighter, Normal, and Darker.

Mobile: (?)

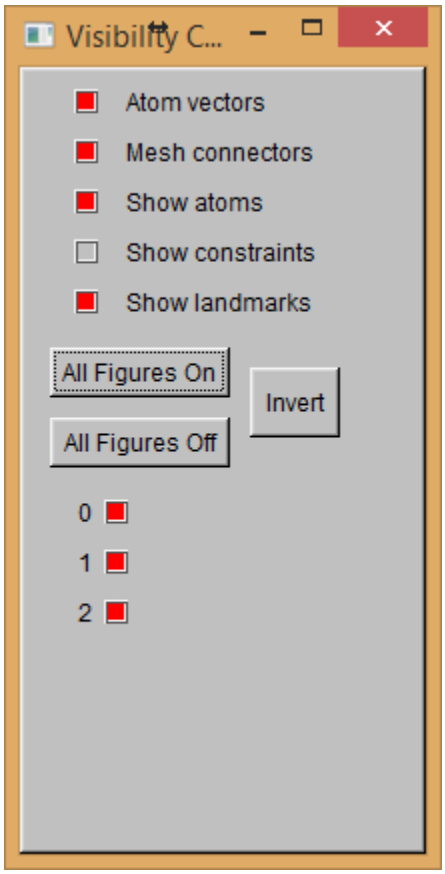
- Clip Tab

In this tab, users can define the clip planes for the images.



4.5 Visibility Window

Visibility window controls the visibility of the models.



Atom vectors: controls whether to show the spokes.

Mesh Connectors: controls whether to draw the meshes which are connecting the neighboring atoms.

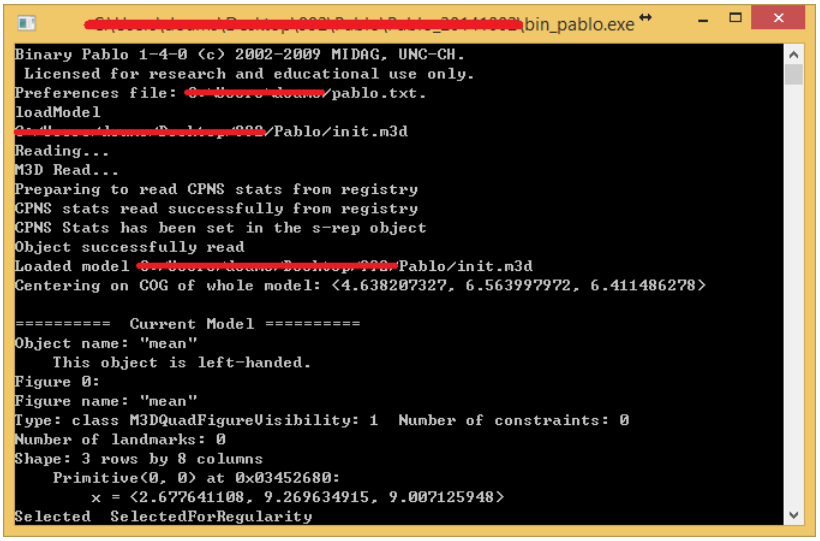
Show atoms: controls whether to show the atoms.

Show constraints: (?)

Show landmarks: controls whether to show the landmarks.

All Figures On/Off/Invert: makes the models visible, invisible or reverse the current status. Clicking the checkboxes at the bottom also works for showing/hiding the corresponding models.

4.6 Console Window



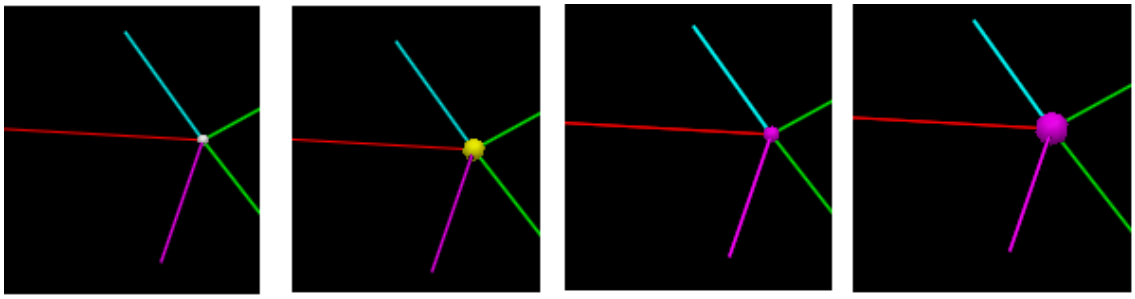
Text messages are printed here as diagnostics. Users can view the filename loaded, operation messages, error messages, model information, and etc.

4.7 Editing Models

Sometimes s-rep models need to be edited, especially when the fitted results are not good enough. Before the models been edited, an atom needs to be selected or marked.

In order to select an atom, hold down the “Ctrl” key and click the mouse left button; do it one more time to deselect this atom. In order to mark an atom, hold down the “Ctrl” key and click the mouse right button; do it one more time to unmark this atom.

Atoms change their color or size to show the status for manipulation. The unselected atom's center is a small white ball, which is connected to the neighboring atoms by the meshes. When selected, a large yellow ball is drawn. When marked, a magenta ball is drawn.



Model edit involves the atom edit (position) and spoke edit (direction, length, angle):

- Atom edit: Select the atoms that need to be edited. Hold down the "Alt" key, and drag with the mouse wheel to move the selected atoms to the best positions.
- Spoke edit with directions and length: Select the atoms that need to be edited. Hold down the "Alt" key, and drag with the mouse left button to change the directions of the spokes; drag with the mouse right button to change the length of the spokes. Note that all the spokes will change together when you move the mouse.
- Spoke edit with angles and crest spoke length: If the spoke edit described above still can't be satisfied, Pablo allows changing the spoke angles and the crest spoke length. See *Edit->Atom Editor* in Section 4.3.2 for more information.

The edited model can be seen as a good new starting point to do fitting again. With a better initial model, fitting result and statistics obtained from CPNS can be significantly improved.

Chapter 5

Fitting a Model

Fitting an s-rep to an object description is one of the most important applications of Pablo. Obtaining a good fitted model is the basis for the further researches, such as registration, segmentation, classification, etc. In this chapter, the procedure of fitting a model will be introduced.

5.1 File format of .m3d

Before fitting, users would be better to understand the meaning of the content of the .m3d file.

S-reps, which live in a cubic region $[0, 1]^3$, are saved in the ASCII files with the extension .m3d. Users can use a text editor to view or edit this kind of files. Model files use the nested curly brackets to describe the content of each part. Note that the open brace must appear on the same line as the name of each part, and the close brace must be on a separate line. The attributes in each part can be assigned to the associated values by the character "=". All the assignment lines must end with a semicolon. The names of the attributes are reserved keywords; only the programmers may create or edit them.

There are some information contained in these files: the model type (1 for slabular model and 0 for quasi-tube model), the number of rows and columns, the image intensities format (positivePolarity = 1 for bright object on a dark background, and positivePolarity=0 for dark object on a bright background), the parameters of each discrete spoke (its tail position on the skeletal sheet as x, y, z ; its orientation defined by a unit 3-tuple as u_x, u_y, u_z; its length as r), a similarity transformation that map it to an appropriately scaled, rotated and translated space, the location of bounding box, etc.

Some .m3d files are the result of the statistical analysis of CPNS. In this kind of files, the information talked above are the attributes of the mean s-rep generated by CPNS. Besides this mean information, these files also include the information about the eigenmodes, which represents the variations of the population of s-reps. In .m3d file, the last part named "CPNSStats" includes this statistical information.

5.2 Signed distance image

Pablo tries to fit an s-rep to some kind of data which represent the boundary of the object, such as the binary images with value 1 inside the object and 0 outside, the surface tile set with polygonal planar tiles, etc. These data need to be transformed into a signed distance image with the zero-level surface representing the boundary of the object being smooth. In Section 5.4.3, the production of a smoothed signed distance image will be introduced.

5.3 Configuration file

Configuration file is one of the inputs of the fitting command. A configuration file is an ASCII file that indicates the stages of fitting to be applied and the coefficients for the terms of objective function in each stage. Users can directly adjust the values of these coefficients to improve the results of fitting.

All of the stages, including spoke stage, shape space stage and atom stage, involve the optimization of the objective functions. Note that the initialization stage is performed independently so the parameters related to this stage are not included in the configuration file. Generally speaking, there are mainly two kinds of terms for the objective functions. One is the image mismatch term. It involves letting the tips of the s-rep spokes touch the boundary of the object. The other is the geometric atypicality term. It involves getting a good geometric s-rep. As the coefficients of the terms increase, the quality of the s-rep will be worse. So there is a compromise between the fidelity to the signed distance image and the geometric properties of the s-reps.

Configuration file uses "#" to comment the rest of a line, and uses the format of "KEY = VALUE" to do the assignment for each parameter. The values in the square brackets on an assignment line indicate the optional values for this parameter.

In most cases, the atom stage is not necessarily used if the initialization stage followed by a spoke stage or a shape space stage followed by a spoke stage works well. To skip this stage, simply set the iterations of the corresponding stage to 0. Only when the model generated by the suggested procedure above cannot match the ground truth well, the atom stage would be applied as a mechanism. Also, the shape space stage is only usable when the statistical model generated by CPNS is acceptable, whereas the TPS-based initialization is more often to be used.

After that, the fitted models generated by TPS-based initialization followed by a spoke stage would be used to obtain a better statistical model.

For the names of the parameters, there are three kinds of prefixes ("figure", "atom", "spoke"("sRep")) to indicate which stage they are related to. In the following part, these three prefixes will be replaced by the character "*" if this parameter appears in all of the three stages. Next, the meaning of each term will be explained.

5.3.1 DATA part of the configuration file

The data describing the images must have the cubic voxel and be in .raw3 format with binary values 0 and 1. "tileSurfaceLevel" and "dilatoinVoxels" are two parameters in DATA part.

- **tileSurfaceLevel**: set the density of the tile set vertices. This value is from 0 to 3. The higher this value is, the smoother the surface is and the slower the fitting process is.
- **dilationVoxels**: set the degree to dilate very thin voxels. 0 is recommended.

5.3.2 INITIALIZATION STAGE of the configuration file

- **doMethodOfMoments**: This is used to roughly align the initial template to the object image before fitting. It involves a similarity transformation. Configuration file uses three bits to indicate the format of similarity transformation. The least significant bit (with the value 1) represents translation; the second least significant bit (with the value 2) represents scale; and the third least significant bit (with the value 4) represents rotation. Users can simply add the corresponding values of translation, scale and rotation to set the desired similarity transformation. For example, if the similarity transformation contains translation and rotation, just set this value to 5.

5.3.3 Parameters in spoke stage, shape space stage and atom stage

- ***ResetModelBefore**: control whether to reset the model to the original input before the indicated stage starts.
- ***Iterations**: set the number of iterations. If users want to disable figure/atom/s-rep stage, just set this value of the corresponding stage to 0.
- ***SurfaceLevel**: set the density of the tile set vertices. The optional values are 0-3.
- ***AverageNeighborMatch**: penalize the grid of the skeletal sheet being irregular. If the result s-rep does not have a nice grid, increase this value.

- ***ImageMatch**: let the tips of the spokes touch the boundary of the image, which means the fidelity to the signed distance image. Increase this value if the s-rep does not match the image very well.
- ***LandmarkMatch**: control the fidelity to the landmarks.
- ***ModelMatch**: if the initial model is good enough, give a large value to this term; otherwise, give a small value to this term.
- ***CurvinessPenalty**: penalize the skeletal sheet being too curved.
- ***RSradPenalty**: penalize the spokes being too close to cross with each other.
- ***ImageNormalMatch**: penalize the directions of the spokes being non-orthogonal to the boundary of the images which are fitted to.
- ***ImagePlaneOrienAtEnd**: let the planes spanned by the upside spoke, downside spoke and crest spoke be orthogonal to the skeletal sheet. If these planes are nearly co-plane with the skeletal sheet, increase this value.
- ***ImageVertexMatch**: penalize the crest spoke not being in the crest location.

5.3.4 Parameters only in shape space stage

- **figureDoScale**: control whether to do some scaling in the optimization process in the figure stage.
- **figureMahalanobisMatch**: in the figure stage, penalize the fitted model being far from the mean obtained by CPNS analysis. The distance of difference is measured by the Mahalanobis distance.
- **figureMaxPGAModes**: set the maximum number of PGA/CPNS eigenmodes that will be used to form the basis of the shape space.

5.3.5 Parameters only in spoke stage

- **spokeRadiusMatch**: penalize the length of the upside spoke and downside spoke at the same atom being far different.
- **spokeAtomRestrictMovement**: penalize the position of the atom being moved during the s-rep stage.
- **spokeThetaFactor**: turn on (with the value 1)/off (with the value 0) the ability of the spokes to change their angles during the s-rep stage.
- **spokeScaleFactor**: turn on (with the value 1)/off (with the value 0) the ability of the spokes to change their length during the s-rep stage.

- RSradPenaltyNorm
- RSradPenaltyThreshold
- **spokeIterations**: This parameter controls the number of iterations to optimize quasi-tube models. If users fit a slab model, just leave this parameter alone.
- weightMatchAtEnds
- WeightNormalMatchAtEndMidSpoke
- saveDistanceVectors

5.4 Steps to fit a slab model

The ways to fit a slab model and a quasi-tube model are different. But both of them will use the command line to do the fitting job. The command line asks to provide some inputs including the initial s-rep to be fit, the object signed distance image to be fit to, the location to save the fitted result, and a configuration file that controls the parameters of fitting. The following steps are the procedure to fit a slab model.

5.4.1 Create an initial model

Fitting process requires a starting point, an initial model, to do the optimization. Use the "Edit->Add Quad Figure" to create the initial model. Users need to define the size, the number of rows and columns, of the regularly spacing grid. This size should be chosen properly so that the spokes at each atom can roughly capture the structure of the object image. Note that in the initial model, the tails of the spokes at the same atoms share the same position. In later optimization, those spoke tails may be separated from each other. Also in the initial model, the upside spoke and the downside spoke at the same atom have the same length. In the later optimization, the length of those spokes may be changed and be different.

5.4.2 Initialization stage: Alignment to object data

Given the initial model created in section 5.4.1, the initialization stage uses thin-plate-spline based method to initially align this s-rep template to the object data that are ultimately fit to. Such alignment operation works on the SPARM model. Users can download this SPARM software from https://www.nitrc.org/projects/spharm-pdm. The detailed procedure of the initialization stage is as follows:

1) Get the SPHARM model from the initial s-rep template.

i) Load the initial s-rep to Pablo (File->Load Models...).

ii) Export this model to binary image (File->Export Model To Image), and save it as .mhd file (e.g., test.mhd).

iii) Go to the build directory of the SPHARM. Type the command:

./bin/SegPostProcessCLP *binary_image_of_s-rep output_image* --rescale --space *values_of_x_y_z_axis*

(e.g., ./bin/SegPostProcessCLP *directory*/test.mhd *directory*/test_PP.mhd --rescale --space 0.005, 0.002, 0.002)

iv) Type the command:

./bin/GenParaMeshCLP *output_image_of_the_previous_step output_para_file output_surf_file* -- iter *number_of_iterations*

(e.g., ./bin/GenParaMeshCLP *directory*/test_PP.mhd *directory*/test_para.vtk *directory*/test_surf.vtk --iter 1000)

Note that the first input of this command is the output of the previous step, and the second and third inputs are the directory of the outputs. The flag of --iter defines the number of iterations, and 1000 is recommended.

v) Type the command:

./bin/ParaToSPHARMMeshCLP *output_para_file_from_the_previous_step output_surf_file_from_the_previous_step prefix_name* --subdivLevel 10 --spharmDegree 12 --flipTemplateOn --flipTemplate *harmonic_coefficients_file*

(e.g., ./bin/ParaToSPHARMMeshCLP *directory*/test_para.vtk *directory*/test_surf.vtk *directory*/Test_ --subdivLevel 10 --spharmDegree 12 --flipTemplateOn --flipTemplate *directory*/*name*.coef)

After this step, the file named Test_SPHARM.vtk is the SPHARM representation of the initial s-rep model.

2) Roughly align the object data to the SPHARM model obtained in step 1).

i) Use a simple alignment method, such as Procrustes alignment, to initially align the object data to the initial template.

ii) Note that if the template is the mirror image of the object data, users need to flip the template first. The program called “MeshMath” in the SPHARM software can do this work. For example, type the command under the build directory of SPHARM to flip the initial template:

./bin/MeshMath *directory*/Test_SPHARM.vtk *directory*/Test_SPHARM_flip.vtk --scaleMesh -1 -1 1

iii) The .vtk file cannot be read by most of the software. In order to do the alignment operation, the .vtk files can be first translated to .meta files for easy reading. After alignment, the .meta files need to be converted back to .vtk files. The SPHARM software provides the programs to convert the files between .vtk and .meta formats. The usage of the two commands is:

./bin/VTK2Meta *sourcefile.vtk outputfile.meta*

./bin/Meta2VTK *sourcefile.meta outputfile.vtk*

3) Use thin-plate-spline program to wrap the initial template to the object image. (where to download?)

The thin-plate-spline program will wrap the initial s-rep model to better fit an object image based on the relationship between the reference SPHARM model and the target SPHARM model. It requires users to provide four inputs: the SPHARM model of the initial template, the SPHARM model of the aligned object data, the initial s-rep model, as well as the directory of the wrapped output model. Go to the build directory of the Correspondence software. Type the following command:

./bin/tps_srep *reference_SPHARM object_image_SPHARM reference_template outputfile_wrapped_model*

(e.g., ./bin/tps_srep *directory*/Test_SPHARM.vtk *directory*/object_image_SPHARM.vtk *directory*/test.m3d *directory*/outputfile_wrapped.m3d)

This wrapped s-rep model named outputfile_wrapped.m3d is the output of the initialization stage, and is also the input to the shape space/spoke/atom stages for Pablo fitting.

5.4.3 Create the signed distance image

Pablo takes the signed distance image as the input, and the wrapped s-rep obtained from the previous step need be fit to this signed distance image. When the input image is binary image, the transformation to the signed distance image is necessary.

Note that if the format of the image description is .vtk rather than .mhd, some software like PolyDataToImageData may be needed to transform the data from .vtk to .mhd. Download this software from http://www.vtk.org/Wiki/VTK/Examples/Cxx/PolyData/PolyDataToImageData. The format to use the command is:

./build_location_of_PolyDataToImageData/PolyDataToImageData *sourcefile.vtk output.mhd*

1) Transform to a non-smooth signed distance image

   Pablo provides the program "Image2SignedDistanceMap" to convert the binary image to the signed distance image. Go to the build directory of Pablo, and type the command

   ./Image2SignedDistanceMap *sourcefile_binary_image.mhd*

   Here the input is the binary image that needs to be transformed, e.g. image.mhd, and the output will be automatically generated, which has the same name with the input file plus a suffix "-ddm", e.g. image-ddm.mhd.

2) Smooth the signed distance image while retaining the zero level set

   Pablo requires that the zero level surface of the signed distance image should be smooth. However, the output of the Image2SignedDistanceMap program has the jagged boundary and cannot satisfy this requirement. In order to create a smooth signed distance image from the previous step, an anti-aliasing program called antiAliasWrapper provided by Pablo should be used. This is a matlab code and can be found in the Pablo\antiAlias folder. This function will be invoked like:

   antiAliasWrapper(input_file, output_destination, tolerance);

   The input_file is the file containing the non-smooth signed distance image (e.g. image-ddm.mhd), and the output_destination is the location to save the smoothed signed distance image. If the tolerance field is left out, the default value of [0.5, 0.5] will be taken. This parameter constrains the distance between the implied boundary and the center of the voxels. The default value of [0.5, 0.5] is recommended, because it keeps the boundary within the same voxel. A larger value will yield a smoother signed distance image, but with the larger movements of the implied boundary.

5.4.4 Fitting a wrapped s-rep to the signed distance image

With a wrapped s-rep and the corresponding signed distance image, fitting can be done by the command line. Go to the build directory of the Pablo, and type the command:

./bin_pablo -po -cs *location_of_the_configuration_file* -image *location_of_antiAliased_signed_distance_image* -model *location_of_wrapped_s-rep* -outModel *location_of_output_model*

(e.g., ./bin_pablo -po -cs *directory*/config.txt -image *directory*/image-ddm.mhd -model *directory*/test.m3d -outModel *directory*/output.m3d)

The flag of “-po” specifies that the optimization process will be done without showing the Pablo display window; the flag of “-cs” specifies the next input is the location of the configuration file; so do the flags of “-image”, “-model” and “-outModel”. Use the flag “-h” to look for help and see the description of all flags. The output of this command is output.m3d, which is the final result of fitting. Note that if users want to avoid applying the atom stage or shape space stage, set the iterations of the corresponding stage in the configuration file to zero.

5.4.5 Compare the result s-rep vs. the input image

Users need to know the quality of the fit. Pablo provides two ways to visualize the s-rep model against the object image. Before visualization, both the s-rep and the image need to be loaded to Pablo.

The first method is to view the contours on the image slices. Select the Image tab under “Window->Display Control”, and turn on the “Boundary” option. Users can choose the slices that will be displayed, and slide the slider to observe the curve on the image slices where it intersects the s-rep boundary.

The second method is to view the tile set of the image. Select “File->Generate Tile Set from Image”, and the tile set will be generated in a few seconds. Note that, the three slices had better turn off for better observation. Click the Image tab under “Windows->Display Control”, and turn off the Sagittal/Coronal/Axial slices. Then click the Tile Set tab, change the opacity of the tile set in order to compare the s-rep model and the tile set easily.

After comparison, users can try to adjust the parameters in the configuration file or try to apply different stages to improve the result. However, sometimes some manual edit to the atoms on the result will be needed. See section 4.7 to know how to edit the models. A new round of fitting may need to be applied on this edited model to get a better result.

5.5 Good fits and bad fits

Most of the time, Pablo can get a good fit using the command line. Figure 3 is a good fitting result: all the spokes are touching the boundary; the spokes are approximately perpendicular to the boundary; and the fitted model is close to the ground truth.

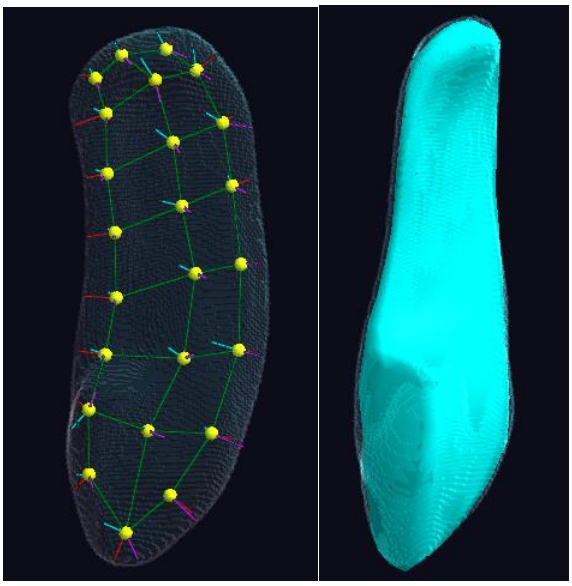


Figure 3 A good fit

However, sometimes a bad fit will be got if the parameters or methods are not used properly. Users need to know what a bad fit is. Figure 4 shows one kind of bad fits. Some atoms are located outside the tile set. In this situation, in order to make the atoms' spokes touch the boundary, Pablo has to make the spokes pointing inside (wrong direction). To deal with this problem, users can select these problematic atoms, move them inside the tile set, rotate the spokes to make them pointing outside, and do fitting again.

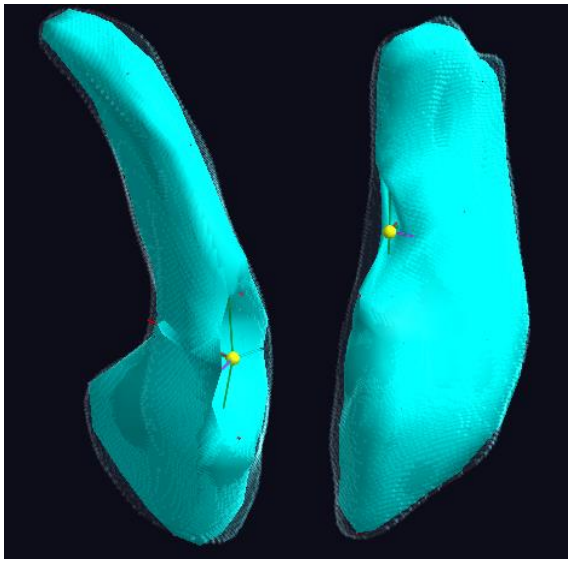


Figure 4 Bad fits: spokes pointing inside

Figure 5 shows another kind of bad cases. These atoms are located on the boundary, which causes the length of some spokes equal to 0. To deal with this problem, users need to move

these atoms inside the tile set, prolong the spokes whose length are equal to 0, and do fitting again.

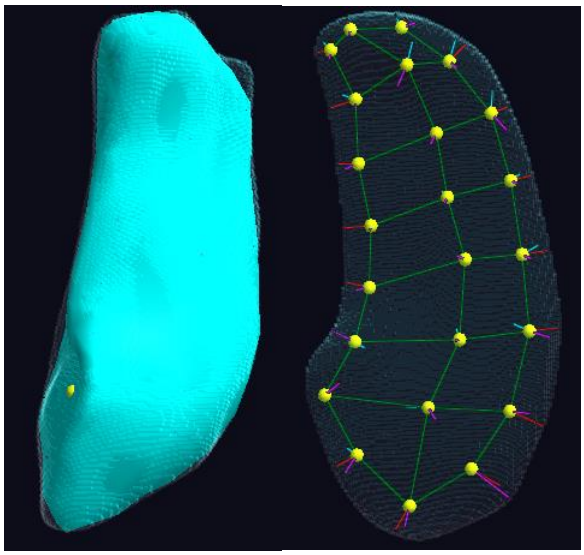


Figure 5 Bad fits: atoms locating at the boundary

5.6 Steps to fit a Quasi-tube model (empty)

Chapter 6

Statistical tools

6.1 Statistical analysis via CPNS

6.1.1 What is CPNS?

CPNS (Composite Principal Nested Spheres), proposed by Sungyu Jung [4], is a method to do the statistical analysis based on a population of the models. Unlike PCA, which works well on the Euclidean features, CPNS recognizes that most of the features in s-reps are non-Euclidean features, which live on the abstract spheres. This method computes a Euclideanized feature value (such as a spoke direction) corresponding to each non-Euclidean feature value. After all of the non-Euclidean features have been Euclideanized, all of the features including Euclidean and Euclideanized ones are appropriately weighted. Then PCA is used to get a mean s-rep and modes of variation in the Euclidean and Euclideanized features. This approach uses the backwards analysis: at each step decrease the dimension of the spheres until a 1-sphere has been arrived at.

The .m3d files produced by CPNS contain not only the basic information of s-rep model, but also the statistical information, including the mean s-rep, the eigenmodes and principal variance, and the polar systems for each sphere and subsphere.

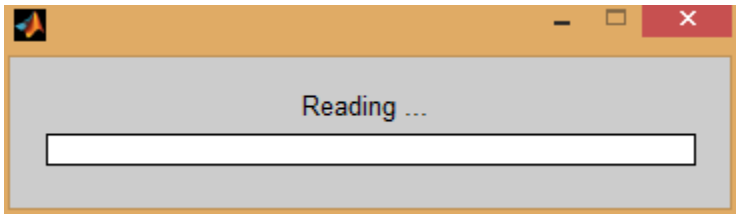
Note that CPNS can be used not only for s-reps but also for lots of other objects that live in a Cartesian product of a Euclidean space and a collection of spheres.

6.1.2 How to do CPNS?

CPNS in Pablo is implemented in Matlab. Before doing CPNS, some decent fits are needed. At the very beginning of the fitting, users may only have limited decent fits. Simply pick some reasonable fits and throw away the other fits that are not very good. Pablo may need a number of rounds of fitting and CPNS to get the results improved.

CPNS code can be found in the downloaded folder, which is "\sreps\Pablo_matlab\shapeStat". This folder and its subfolders contain all the functions related to CPNS. The following is the steps to do CPNS using Matlab.

1) Run Matlab; add the folder "\sreps\Pablo_matlab\shapeStat" and its subfolders to the search path. This is implemented by the command:
   >>addpath(genpath('*directory*\sreps\pablo_matlab\shapeStat'));
2) Put all of the s-rep models that will be involved in CPNS into a same folder.
3) Run the function:
   <<calculateCPNS(dataDir, outputFilename, optionPNS);
   Users need to give the directory where the s-reps are stored and the directory where the output file will be saved. The meaning of the three inputs can be found in the file calculateCPNS.m.
4) Matlab will pop up a window, telling the users how many files are read and showing the progress of CPNS.



5) The result file can be found in the designated directory. The output .m3d file contains a mean s-rep and the modes of variation. Users can use "Tools->CPNS Deformation" to visualize the contribution of the eigenmodes.
6) This mean s-rep obtained by CPNS would be much better than the previous initial template. So use this mean to refit all of the cases again. This time, the irregularity penalty (like "*AverageNeighborMatch" and "*RSradPenalty") can be lowered, and the geometric penalty of s-rep (like *ModelMatch, *ImageMatch, *ImageNormalMatch) can be enhanced. This will yield many more decent fits, from which users can compute a new mean and modes of variation via CPNS.
7) This new mean obtained in step 7) would be used to fit the rest of the population. Pick all of the decent fits and compute new statistics (a new mean and eigenmodes) via CPNS. Repeat doing the fitting using this new mean. This repeated process can be expected to yield decent fits for most of the population.