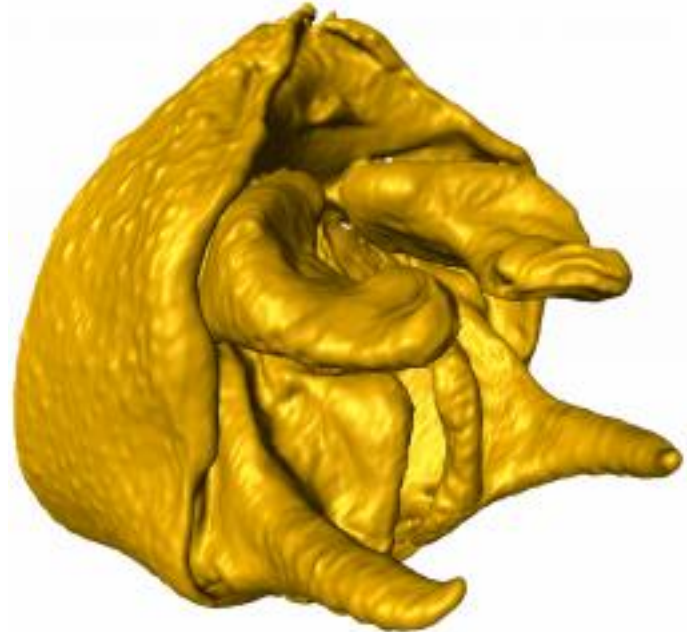


# SPHARM

Version 1.4

A Matlab Executable to perform  
Spherical Harmonic Analyses of  
Triangular Mesh Surfaces



Li Shen

Center for Neuroimaging, Division of Imaging Sciences, Department of Radiology, Center for  
Computational Biology and Bioinformatics, Indiana University School of Medicine, 950 W  
Walnut St, R2 E124, Indianapolis, Indiana 46202

[shenli@iupui.edu](mailto:shenli@iupui.edu)

Mark A. McPeck

Department of Biological Sciences, Dartmouth College, Hanover, New Hampshire 03755 USA

[mark.mcpeek@dartmouth.edu](mailto:mark.mcpeek@dartmouth.edu)

Hany Farid

Department of Computer Science, Dartmouth College, Hanover, New Hampshire 03755 USA

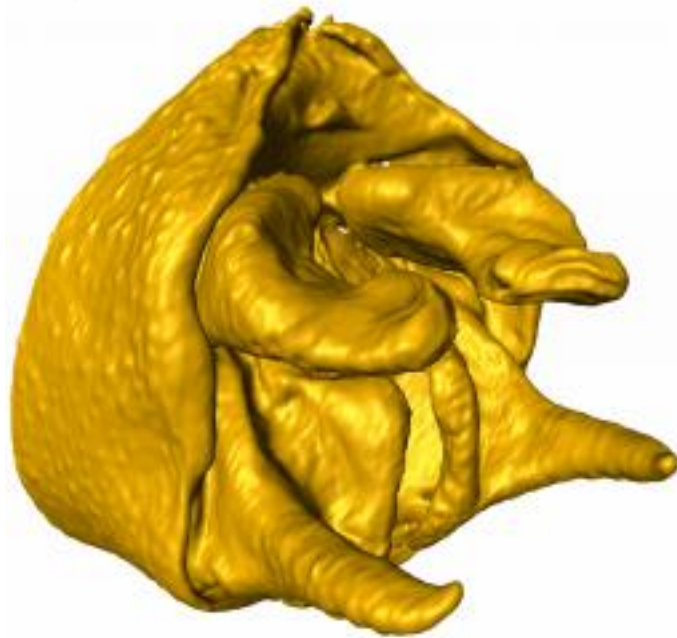
[hany.farid@cs.dartmouth.edu](mailto:hany.farid@cs.dartmouth.edu)

## License and Acknowledgments

This software is distributed under the [GNU General Public License, Version 3](#), 29 June 2007. End users use this software at their own risk. End users are responsible for checking that the software is producing expected output from these analyses. The authors are not liable for any defects in the performance of this software. Please report any detected bugs [here](#).

Also, if you substantially modify or add to the capabilities of this software, the [authors](#) would like to hear about your changes. We are always looking for improvements in the package and would like to make the best techniques available at the download site for others to use.

The production of this software was supported by National Institutes of Health grant NIH/NIBIB R03EB008674-01 to Li Shen and National Science Foundation grant IBN-0516104 to Mark A. McPeck and Hany Farid.



SPHARM is a software application to perform spherical harmonic analyses of triangular mesh representations of the surfaces of objects as described in Shen et al. (2009). [Wikipedia](#) has a general introduction to [spherical harmonics applications](#) in many different fields. SPHARM is written in [Matlab](#)<sup>®</sup> (The Mathworks, Inc., Natick, Massachusetts) and is distributed as a Windows executable and as a collection of Matlab scripts.

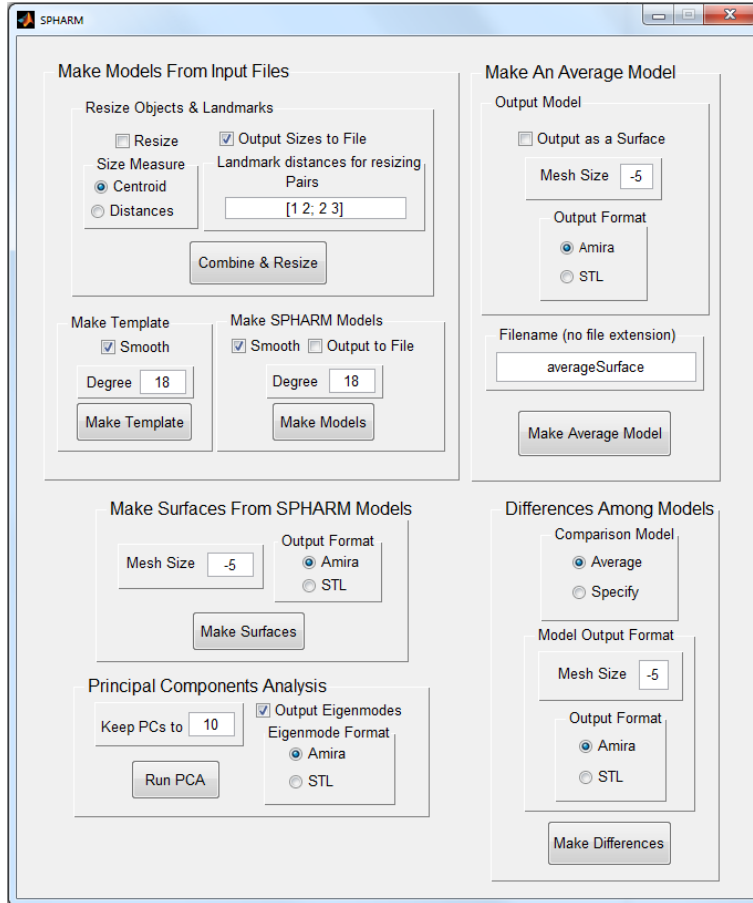
## Installation

To install SPHARM, copy the **SPHARM\_pkg.exe** file to a directory of your choice and extract the contents of the file to that directory. A number of files will be copied to the directory. If you have a full installation of Matlab installed on your computer, you can run SPHARM version 1.4 by simply double-clicking on **SPHARM1p4.exe**. If you do not have a full version of Matlab installed on your computer, you must install the [Matlab Compiler Runtime \(MCR\)](#) on your computer. An installation for the MCR for Matlab version R2013a is included in the **SPHARM\_pkg.exe** file. You should be given the option to install the MCR by running **SPHARM\_pkg.exe**. Simply follow the directions given by the installer. After installing the MCR, simply double click on **SPHARM1p4.exe** to run the program.

Two other files included in the distribution file are required to run SPHARM. These are **spa.exe** and **msvcrt.dll**. These are library files that are needed for some intermediate calculations. These files must be in the same directory as either the Matlab script files (if you are running SPHARM directly from scripts) or **SPHARM1p4.exe** (if you are running SPHARM from the Windows executable).

Example data sets are included in a data subdirectory in the **SPHARMDData.zip** file. These are three triangular mesh files (files with the extension ‘.m’) of damselfly cerci (see McPeck et al. 2008) and associated landmarks (‘.landmarkAscii’ files).

# Running SPHARM



Once you have installed the MCR, you can run SPHARM by double clicking on the **SPHARM1p4.exe** file or running the SPHARMGUI.m script in Matlab. When you fire the program, the SPHARM Graphical User Interface (GUI) will appear. Analyses are run from the SPHARM GUI, and progress output is listed to the Windows command window.

All analyses are performed from this Graphical User Interface (GUI). The GUI has five major panels ordered from top to bottom. They are:

1. Make Models From Input Files
2. Make Surfaces From SPHARM Models
3. Principal Components Analyses
4. Make An Average Model
5. Differences Among Models

Each panel controls a separate task in an overall spherical harmonics analysis or analyses that utilize spherical harmonics models. Each of these tasks will be discussed in sequence. In this document, only the practical steps of running the software will be considered. For more information about the mathematical and computational workings of spherical harmonics analyses or the algorithms implemented, please consult Shen et al. (2009) and references cited therein.

## Data Format

At present, SPHARM accepts triangular mesh data in only two formats: STL and Matlab. The first is the familiar STL format that is produced by many CAD programs. The second is as a Matlab m text file of vertices and faces.

These formats are based on the output produced by [Amira](#)<sup>®</sup> (Visage Imaging, Inc., Carlsbad, California, USA), the software we use to produce triangular mesh data. Please contact [us](#) if the software you use to produce triangular mesh data cannot generate these formats and would like to see if we can add those formats to the input of SPHARM. We cannot guarantee anything, but it does not hurt to ask.

SPHARM accepts vertices, faces and landmarks of a triangular mesh in one or two files as described below.

If the data are in two input files, one file must contain either an ASCII STL data structure or a Matlab structure containing the vertices and faces of the triangular mesh (in that order), and the other file containing the landmarks. The names of the files must match exactly. The triangular

mesh in the STL must have and '.STL' file extension, or the Matlab structure of vertices and faces must have a '.m' file extension. The second file must be a file of landmarks having a '.landmarkAscii' file extension (see example data files) but the same file name.

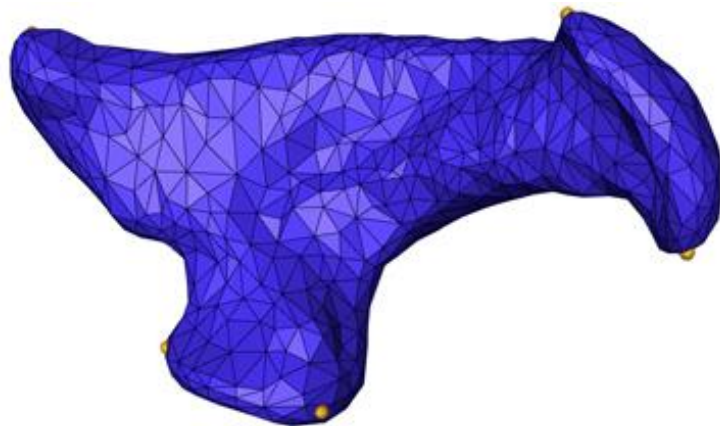
So for example, if you have a triangular mesh in STL format saved in a file called **modell.STL**, you must have a file of landmarks in the same directory with the name 'modell.landmarkAscii'. Because this format is widely used in CAD programs, we do not describe its structure further here. If you have trouble converting STL models using the tools below, please let [us](#) know.

We will describe the Matlab format further here, because it is less familiar. If you have a Matlab triangular mesh in **modell.m**, you must have a file of landmarks in the same directory with the name 'modell.landmarkAscii'.

A Matlab structure format bundles the vertices and faces of the triangular mesh into one data structure. Please open one of the example files to see the data format.

```
% this is a matlab file which can be loaded by
% load "Easp_ME_6_4.m"
```

```
surface = struct('vertices', [856.399 532.41 454.388; ...
855.271 545.967 454.366; ...
882.376 551.071 449.063; ...
...
740.515 398.195 1123.64; ...
759.597 409.625 1121.28; ...
], 'faces', [13 1 2; ...
13 3 6; ...
13 2 3; ...
...
999 1002 1001; ...
999 1000 1002; ...
]);
p = patch(surface);
```



The triangular mesh data in **Easp\_ME\_6\_4.m** are listed here (with most vertices and faces not listed), and the visual representation of this mesh is shown next to the file listing. The file can begin with any number of blank lines and Matlab comments (lines beginning with '%'). The first line of data starts on the line beginning with 'surface = struct('. This defines the beginning of the Matlab structure. Everything between the '(' on this line and the ')' near the end of the file are the data for the triangular mesh named 'surface' in this file. Each file must have this exact definition of the structure (i.e., 'surface = struct(' to begin the structure).

The triangular mesh of this object is defined by the vertices and faces listed in this data structure. Vertices are points scattered across the surface of the represented object. Faces are the triangles defined by the vertices that cover the surface of the object. If the file contains N vertices, it must also contain 2(N-2) faces: the number of vertices and faces in a complete triangular mesh

In the structure, the keyword 'vertices' identifies that the vertices for the triangular mesh follow. The '[' immediately after 'vertices' begins the list of vertices. Each vertex is listed as a triplet giving its three dimensional (3D) x, y and z coordinates. The first is listed on the line immediately after the '[', and the rest are listed one per line in sequence until all are listed. Matlab format requires that ';' ... terminates each line. The vertices are identified by their order in this file from 1 to N. The line immediately after the last vertex begins with ']' to signify the end of the vertices.

This line then continues with 'faces [' to identify the beginning of the list of 2(N-2) faces. The three integers in a face identify the three vertices that define the triangle of that face. The complete list of faces must construct a closed surface. The list of faces ends when a new line starts with ']' immediately after the last face. In [Easp\\_ME\\_6\\_4.m](#), this line also terminates the structure (i.e., '];'), because the list of landmarks are in a separate file. Amira produces the subsequent line of 'p = patch(surface);' but this is ignored by SPHARM and is not necessary.

The landmarks of the object are a separate set of points on the surface of the object that identify homologous points across a set of objects (see Zelditch et al. 2004 for a full discussion of landmarks). Amira produces a separate file to store landmarks. [Easp\\_ME\\_6\\_4.landmarkAscii](#) is listed here:

```
# AmiraMesh 3D ASCII 2.0

define Markers 7

Parameters {
    NumSets 1,
    ContentType "LandmarkSet"
}

Markers { float[3] Coordinates } @1

# Data section follows
@1
650.682 607.458 632.607
582.402 571.192 797.036
687.445 389.492 1086.93
911.024 414.062 1016.77
993.932 621.327 495.161
923.262 419.796 675.338
777.261 356.417 558.65
```

SPHARM ignores everything in this file except the lines listing three double precision numbers. These are the x, y and z coordinates of the seven landmarks for this object. Three of the seven landmarks can be seen as small yellow dots on the triangular mesh of this object above (the other four are on the back side of the object). Any format that simply lists landmarks in order with three double precision numbers separated by spaces per line can be read by SPHARM. Note that no line terminators follow the landmark coordinates. The lines must only contain three numbers separated by spaces

Alternatively, SPHARM can read a Matlab structure file that contains the vertices, faces and landmarks all in one file. For [Easp\\_ME\\_6\\_4](#) this file would have the following format.

```
surface = struct('vertices', [856.399 532.41 454.388; ...
855.271 545.967 454.366; ...
882.376 551.071 449.063; ...
...
740.515 398.195 1123.64; ...
759.597 409.625 1121.28; ...
], 'faces', [13 1 2; ...
13 3 6; ...
13 2 3; ...
...
999 1002 1001; ...
```



```

999 1000 1002; ...
], 'landmarks', [650.682 607.458 632.607; ...
582.402 571.192 797.036; ...
687.445 389.492 1086.93; ...
911.024 414.062 1016.77; ...
993.932 621.327 495.161; ...
923.262 419.796 675.338; ...
777.261 356.417 558.65; ...
]);

```

The vertices are listed, then the faces, and finally the landmarks. If landmarks are given in the file with the vertices and faces, the landmark file (if present) for the object is ignored.

All objects to be included in a SPHARM analysis must be in a single directory. And all objects must have the same number of vertices (and therefore faces) and homologous landmarks.

## Make Models From Input Files

The first major panel controls the algorithms for creating the Spherical Harmonic (SPHARM) Models that are used for subsequent analyses. This panel contains three subpanels that control the three steps needed to produce a set of registered SPHARM models. These three subpanels are:

1. Resize Objects & Landmarks
2. Make Template
3. Make SPHARM Models

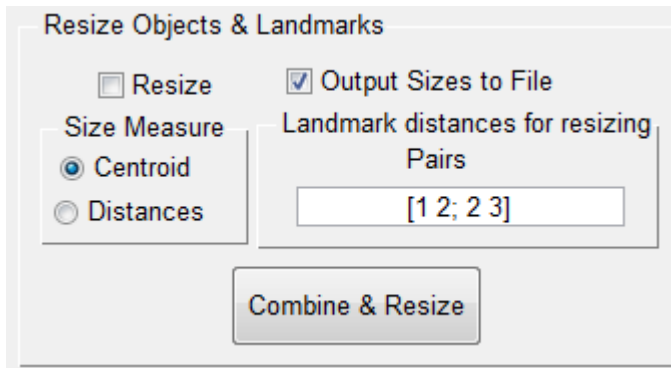
We will discuss each of these in turn in the following.

### Resize Objects & Landmarks

The tasks controlled by the Resize Objects & Landmarks subpanel are to read the raw triangular mesh and landmark data and to perform the size transformations of the data in preparation for spherical harmonic analyses.

SPHARM can resize the triangular mesh objects by two different methods. The first is by the centroid size of the object, and the second is by a set of distances among the landmarks of the object. Each object will be resized if the Resize checkbox is checked, and the method used will

depend on which radio button is selected in the Size Measure area. See Zelditch et al. (2004) for a discussion of “size” in morphometric analyses.



If you want to perform a SPHARM analysis without resizing the objects, leave the Resize checkbox unchecked.

To resize objects by their centroid size, check the Resize checkbox and choose the Centroid radio button in the Size Measure area.

To resize objects by some combination of the sum of distances among the landmarks, check the Resize checkbox, choose the Distances radio button in the

Size Measure area, and code the combinations of landmarks to use for the distance calculations in the text field in the Landmark distances for resizing area. SPHARM begins with this text set to specify the sum of the distances between landmarks 1 & 2 and between 2 & 3. The entire collection of pairs must be enclosed by square brackets, numbers in a pair must be separated by a space, and each pair of landmarks must be separated by a semicolon. The landmarks are identified by their order they were read from the file. The distances between each pair of landmarks is calculated, these distances are summed, and the object is resized so that this summed distance is 1.0 for every object. Some examples of how to identify distance combinations in the text field are:

<u>Distance combination</u>	<u>Text in text field</u>
Distance(1-2)	[1 2]
Distance(2-3) + Distance(5-6)	[2 3; 5 6]
Distance(1-2) + Distance(4-6) + Distance(4-7)	[1 2; 4 6; 4 7]

Once the Resizing method is set, click the Combine & Resize button to choose the files to be processed for the SPHARM analysis. This will open a standard file chooser dialog. Navigate to the directory containing the .m files or the .STL files to include in the analysis, and choose all that should be processed. The default file format is .m. If you have .STL files, simply change the File Type at the bottom of the Open dialog box to see all the .STL files.

SPHARM reads each .m file or .STL file, depending on what was chosen, into memory (and read the landmark file if landmarks are not contained in the .m file), translates the position of each object in coordinate space so that its centroid is located at the origin (i.e., [0.0, 0.0, 0.0]), resizes the triangular mesh and the landmark positions according to the method chosen, and then writes the resulting data to a new file that is in binary .mat format (a file that can only be opened by Matlab). New files are written to disk because of the enormous amount of data in a triangular mesh. These files will be used in subsequent steps in the process.

SPHARM names the new files according to the sizing method chosen. The base file name of the original .m file is retained, but various suffixes are appended to this base file name to identify what result the file contains. All files produced in this step will have ‘\_OL’ and ‘\_2O’ appended to the file name to signify that the file contains both the Object (i.e., vertices and faces) and Landmarks (hence “\_OL”) and has been translated to the origin (hence ‘\_2O’). Thus, if no resizing is done, this step would produce a file named **Example\_OL\_2O.mat** from a pair of files named **Example.m** and **Example.landmarkAscii**. If the object is resized by centroid size, the file



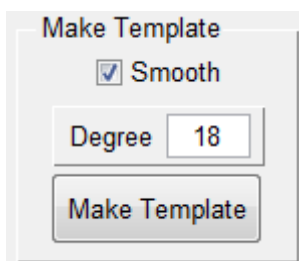
name will have the additional ‘\_CS’ appended (e.g., [Example\\_OL\\_2O\\_CS.mat](#)). If a landmark distance measure is used, the file will have ‘\_DS’ appended (e.g., [Example\\_OL\\_2O\\_DS.mat](#)).

You can also output the size metric used to resize each object by checking the Output Sizes to File checkbox. This will produce an ASCII text file that is simply a list of the file names of the objects followed by the appropriate number for each, i.e., centroid size or distance depending on which resizing metric you choose.

The files produced in this step are the raw data used in subsequent steps of the analysis. In subsequent steps, the user is always asked to choose what files to include in the step. Therefore, the user at this step can produce many different versions of each object that are resized by different methods. Note that the user must rename files produced by different combinations of landmark distances.

## Make Template

The tasks controlled by the Make Template panel of the GUI are to apply the smoothing algorithm and then calculate the spherical harmonic model for an object that will be used as a template in subsequent analyses.



All objects must have a smoothing algorithm applied to them to optimize the positions of vertices across the surface of an object for subsequent analyses; see Shen et al. (2009) and references therein for a description of the purposes and implementation of the smoothing algorithm. This smoothing process is the most time consuming computational step in the entire process, and once smoothing has been performed on an object, the process does not need to be repeated. Smoothing will be performed if the Smooth checkbox is checked.

The spherical harmonic model can be calculated to different user defined degrees, which are set by the integer in the Degree text field. This number specifies the level of detail to be modeled by the spherical harmonic representation. Larger numbers will have greater detail modeled. This value is preset to 18 which would produce a model containing  $(18+1)^2 = 361$  complex valued coefficients (i.e., each coefficient is a [complex number](#)) for each of the three spatial dimension represented in the model, for a total of 1083 complex spherical harmonic coefficients in the final model. A model of degree 18 will capture a tremendous level of detail.

This step does the processing to produce a template object. One object in the analysis group must serve as a template to align and register all the objects in the analysis. Any object in the set can be used as the template. When the Make Template button is clicked, the user will be asked to choose a file to use as the template. If smoothing has not been done to the objects, the appropriate file produced by the Size Objects & Landmarks step should be chosen. So for example, if an analysis of objects resized by centroid size is to be done, and the Example object will be used as the template object, the [Example\\_OL\\_2O\\_CS.mat](#) file should be chosen. The smoothing algorithm will be applied to this object, and a spherical harmonic model will be calculated to the specified degree. Two files will be produced from this step: one will be called [template\\_Example\\_OL\\_2O\\_CS\\_smo.mat](#) which is the result of the smoothing algorithm, and the other is [template\\_Example\\_OL\\_2O\\_CS\\_15\\_0\\_des.mat](#), which contains the spherical harmonic model calculated from this smoothed object. The file name of this file identifies the degree to which the model was calculated. If other models are produced to different degrees of this same object, they will produce different file names that specify the degree.

Because the smoothing step is so time consuming, models can be recalculated from the intermediate file containing the smoothed object without repeating the smoothing. To do this,

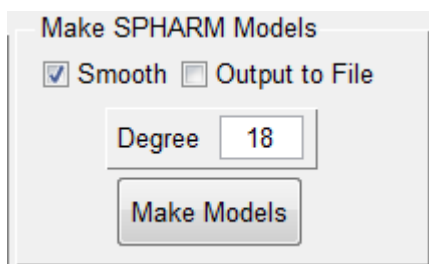
simply uncheck the Smooth checkbox and choose the smoothed file of the template to build a model to a different degree.

The progress of the analysis is displayed in the Windows command window. This analysis may take several minutes depending on the number of vertices in the triangular mesh.

The **template\_\*.des.mat** file (where \* represents a wildcard for the file name of the original object) produced in this step will be used in subsequent analyses to register the other models.

## Make SPHARM Models

The tasks controlled by the Make SPHARM Models panel are for each object to apply the smoothing algorithm, calculate the SPHARM model, and register each object to the template.



The controls for this step are very similar to those in the Make Template step. Check the Smooth checkbox to first apply smoothing to each object, and specify the degree of the resulting spherical harmonic model in the Degree text field. When the Output to File checkbox is checked, a separate text file is produced that contains the spherical harmonic coefficients of all objects included. This file will be useful when using the spherical harmonic coefficients as data in other analyses not performed by SPHARM.

Clicking the Make Models button first brings up a file chooser dialog to select the template file to use for this analysis. The file chosen should be a **template\_\*.des.mat** file produced in the Make Template step that was rendered to the same degree as will be used here.

Once the template file is chosen, a file chooser dialog asks the user to choose the set of files to be processed in this step. Again, these are the files produced by the Size Objects & Landmarks step. All these files should be sized according to the same metric (including the template as well). Thus they all should have the same file name suffix. Also, the file from which the template was made should also be included so that a registered file is produced.

This step produces three files for each object. The first contains the results of the smoothing process (**\*\_smo.mat** where \* is the wildcard signifying the name of the file brought into this step). The second contains the raw spherical harmonic model (**\*\_des.mat**). The third contains the final spherical harmonic model that has been registered to the template object (**\*\_reg.mat**). This third registered file will be used in subsequent analyses because the models of all the objects included in this step of the analysis have been registered to a common object and thus the spherical harmonic coefficients in each are directly comparable. Coefficients are not comparable across models without this registration step (see Shen et al. 2009).

If the Output to File checkbox is checked, the user is asked for the filename where the spherical harmonic coefficients of the registered models should be written. If the specified file already exists, the contents of this file will be overwritten with this output.

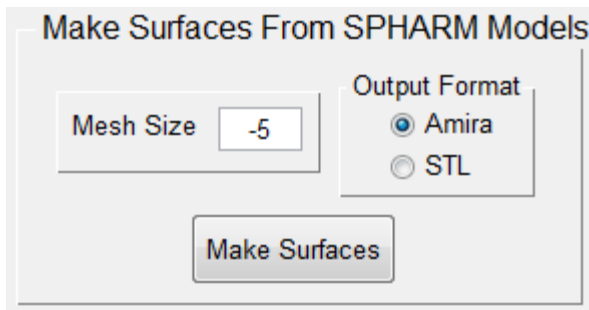
Because the smoothing step is so time consuming, models can be recalculated from the intermediate files containing the smoothed objects without repeating the smoothing (the **\*\_smo.mat** files). To do this, simply uncheck the Smooth checkbox and choose the smoothed files produced by previous analyses at the appropriate window to build a model set to a different degree. Remember that a template produced to the appropriate degree must also be used, and so the user will also have to run the Make Template step again as well if a template of the

appropriate degree is not present. This would calculate the spherical harmonic model to the specified degree and register the objects to one another without smoothing.

The progress of the analysis is displayed in the Windows command window. This analysis may take several minutes to several hours depending on the number of vertices in the triangular meshes, whether smoothing is performed and the number of objects included.

## Make Surfaces From SPHARM Models

The task controlled by the Make Surfaces from SPHARM Models panel is to produce a set of vertices and faces from a spherical harmonic model that is saved as a file in either Amira or STL format.



For this analysis, a coverage of vertices is placed on a unit circle to a density specified by the integer in the meshsize text area, and then a spherical harmonic model of an object is used to reconstruct the shape of the object using this set of vertices. The meshsize parameter determines the size and pattern of the grid of triangles placed on the surface when the model surface is back calculated from the spherical harmonic model.

If the meshsize parameter is set to 0, the vertex positions of the original vertices are used.

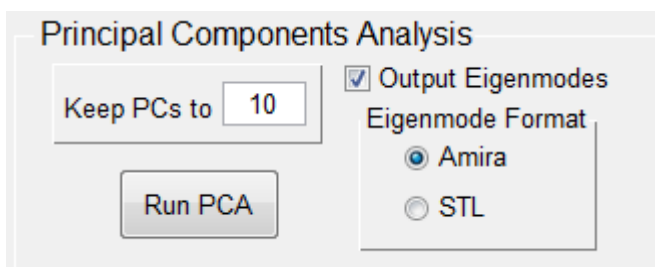
If the meshsize parameter is set to a positive integer, say +X, the reconstructed mesh will be a X\*X grid on the unit circle. So if meshsize is set to 32, a 32\*32 mesh grid will be placed on the unit circle.

If the meshsize parameter is set to a negative integer, an icosahedron subdivision to the specified level will be applied to the unit circle (see Teanby 2006 for this method). So if meshsize is set to -5, an icosahedron subdivision level 5 will be used for the reconstruction.

When the Make Surfaces button is pushed, the user is prompted to identify files from which reconstructed surfaces are to be made. These will be *\*reg.mat* files. These shape files are written in Amira HxSurface or ASCII STL format, depending on the radio button chosen in the Output Format subpanel. They have the same file name as the input files for this analysis, but with a '.surf' file or '.stl' file extension, respectively.

## Do Principle Components Analysis

The task controlled by the Principle Components Analysis panel of the GUI is to perform a principle components analysis (PCA) on a set of spherical harmonic models to ordinate them in spherical harmonic shape space.



This step performs a PCA on the covariance matrix of spherical harmonic coefficients using the princomp function of Matlab. The analysis is done on both real and imaginary parts of the coefficients.

When the user pushes the Run PCA button, the user is first queried to identify the

objects to include in the analysis. These should be *\*reg.mat* files that have all be registered to the same template and calculated to the same degree. An output file named *PCSummary.dta* is then written to the directory where the files are located. This file contains the eigenvalues, percent of variance explained by each eigenvalue, and the PC scores for each object. The PC scores can be used to ordinate the objects in the resulting reduced spherical harmonic space. The number of PCs reported is taken from the Keep PCAs to text area. The user can change this value before the analysis. Because of the tremendous number of spherical harmonic coefficients and because coefficients cannot be attributed to specific features of the object, the eigenvectors are not reported.

If the Output Eigenmodes checkbox is checked, the user is then prompted to identify a directory where the eigenmode surfaces for these PCs will be written. The eigenmode surface is the object that corresponds to a specific position along a PC axis while being at 0 on all other PC axes. Eigenmodes are very useful for identifying the features of shape that vary along each PC axis. Eigen mode shapes are produced for five points along each of the PC axes. These points are  $\pm 2$  standard deviations from the origin,  $\pm 1$  standard deviation from the origin, and at the origin. The standard deviation along each axis is the square root of the eigenvalue. These shape files are written in Amira or STL format. Their names identify the PC and the position along the axis they represent.

## Make An Average Model

The screenshot shows a software dialog box titled "Make An Average Model". It has several sections: "Output Model" containing a checkbox "Output as a Surface", a "Mesh Size" input field with the value "-5", and an "Output Format" section with two radio buttons, "Amira" (which is selected) and "STL". Below the "Output Model" section is a "Filename (no file extension)" input field containing the text "averageSurface". At the bottom of the dialog is a button labeled "Make Average Model".

In many applications I have found over the years that I needed an average model of a set of objects. Therefore, in this version of SPHARM, we have added a routine that calculates an average model and writes that model to the disk for later processing.

The average model is calculated by simply taking the arithmetic average of all the corresponding spherical harmonic coefficients of the set of models to be averaged. Therefore, the models to be averaged must have been derived to the same degree when the models were made originally. In other words, the models must have the same number of spherical harmonic coefficients. Thus, if you wish to average across a set of models that you previously derived to different degrees, you will first have to return to the Make SPHARM Models panel, and rederive the models to the same degree; in this case you can derive the models from the smoothed intermediates saved in the original derivation (i.e., uncheck the Smooth checkbox).

When you push the Make Average Model button, the algorithm will ask you to choose a set of registered models to be averaged. You specify the file name for this averaged model in the Filename (no file extension) textbox in the panel. The averaged model will then be

written to the same directory containing the registered models as a Matlab *\*.mat* file: the program would place the average model in a file called *averageSurface.mat* in the example here.

If you would also like a version of the file that can be visualized in other software packages, you can also have the program save a version of the file in either Amira or STL format. To do this, check the Output as a Surface checkbox, specify the Mesh Size to be rendered (see the *Make*

[Surfaces From SPHARM Models](#) section for an explanation of mesh size specifications), and choose a format to output in the Output Format radio button group. The resulting surface will be in the corresponding file of the name specified. Thus, using the settings shown in the panel here, a mesh of an icosahedron subdivision level 5 would be used for the reconstruction and the resulting Amira formatted surface would be placed in the file named [averageSurface.surf](#).

## Differences Among Models

The Differences Among Models panel produces a set of surfaces that are used to visualize the areas of difference of a set of models from a reference surface. This analysis produces a set of models in which each vertex and face is color coded to represent these differences.

The screenshot shows a software panel titled "Differences Among Models". It has three main sections: "Comparison Model" with radio buttons for "Average" (selected) and "Specify"; "Model Output Format" with a "Mesh Size" input field containing "-5"; and "Output Format" with radio buttons for "Amira" (selected) and "STL". A "Make Differences" button is at the bottom.

To perform this analysis, the user must specify the comparison model to be used as a reference: this is the surface to which all the other surfaces will be compared. If the user chooses the Specify radio button, the user will be prompted to choose a specific model to use as the reference. If the user chooses the Average radio button, the average model will first be calculated for the models to be compared, and this average model will be used as the reference model. In addition, the user must specify the mesh size with which to render the various models for visualization (see the [Make Surfaces From SPHARM Models](#) section for an explanation of mesh size specifications), and specify an output format.

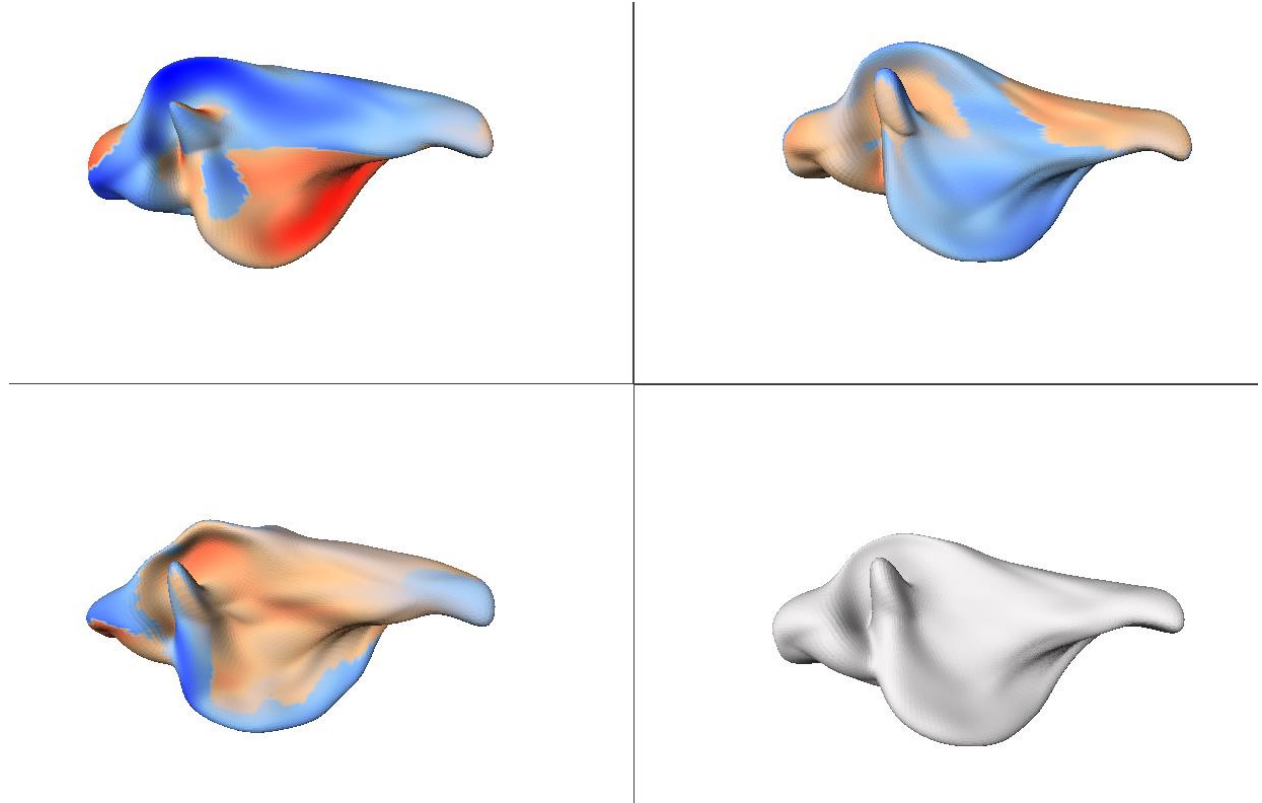
When the Make Differences button is pushed, the user will be prompted to choose the set of models to be compared (and to choose a reference model if the Average radio button is highlighted). This analysis only makes sense on a set of registered models. The distance of each vertex of the rendered reference surface to the centroid of that surface is first calculated. Then the distances of all vertices of each surface to the centroid of the reference surface is calculated. Vertex colors for each surface are then specified by the quantitative difference in distance between the vertices on the

model relative to the corresponding vertices on the reference surface. The maximum absolute difference in distance among all vertices on all models is used to set the range of values specified for color. Quantitative distance values are rescaled to have a range from 0 to 255, with a difference in distance of zero for a vertex receiving a value of 127.5, values less than 127.5 being vertices that are closer to the centroid for the model in question than the reference surface, and values greater than 127.5 being vertices that are closer to the centroid on the reference surface. These values can then be used in other software to specify a color range for visualization.

The color values are calculated for vertices and for faces by averaging the three values for the corresponding vertices. The algorithm first produces a surface of the model and saves this to disk in a file named *\*mod.surf* or *\*mod.stl* depending on the file format specified. Two other files are also saved to disk that contain the values for each vertex (in the file *\*modSurfaceFieldVerts.am*) and for each face (in the file *\*modSurfaceFieldFaces.am*). These files can then be used to render visualizations of the surfaces in which the surfaces are colored based on their differences from the reference surface.

An example of this visualization is given below for the cerci of three Hawaiian *Megalagrion* species. The average of the three models was used to specify the reference surface, which is

shown in the lower right panel and colored all white (i.e., that is the color specified for 127.5 in the colormap used). The surface is colored so that darker shades of blue correspond to vertices and faces that are closer to the centroid than the corresponding vertices and faces on the reference, and darker shades of red correspond to vertices and faces that are farther from the centroid than the corresponding vertices and faces on the reference.





## References

- [McPeck, M. A., L. Shen, J. Z. Torrey, and H. Farid. 2008. The tempo and mode of 3-dimensional morphological evolution in male reproductive structures. \*American Naturalist\* 171:E158-E178.](#)
- McPeck, M. A., L. Shen and H. Farid. 2009. The correlated evolution of 3-dimensional reproductive structures between male and female damselflies. *Evolution* 63:73-83.
- Shen, L., H. Farid, and M. A. McPeck. 2009. Modeling 3-dimensional morphological structures using spherical harmonics. *Evolution* 63:1003-1016.
- Teanby, N. A. 2006. An icosahedron-based method for even binning of globally distributed remote sensing data. *Computers and Geosciences* 32:1442-1450.
- Zelditch, M. L., D. L. Swiderski, H. D. Sheets, and W. L. Fink. 2004. *Geometric morphometrics for biologists: a primer*. Elsevier Academic Press, San Diego, California.