Manual for Fitting S-reps and Computing Shape Statistics via S-reps

Stephen M. Pizer **September 16, 2013**

S-reps are form of skeletal representation that capture the interior of an object in a way that a) provides an object-related coordinate system for the points in the object interior and near exterior; b) supports statistics on object geometry. This document teaches

1) What the s-rep computer representation is.
2) How to prepare object descriptions implying an object boundary for being fit by an s-rep.
3) How to fit s-reps to an individual object description.
4) How to fit s-reps to a collection of object descriptions so that the fits correspond in regard to their interior points.
5) How to generate a probability distribution on object geometry from a collection of corresponding s-reps.
6) How to generate a classification from two classes of training s-reps.
7) How to classify a new s-rep given the classification generated in method #6.
8) How to test hypotheses on differences between s-rep features from two classes.

Some of the sections in the following describe procedures that are presently in use but for which new procedures are anticipated soon. The present methods are described in the italic font, as are comments on what is anticipated as the new procedure.

I. The Computer Representation of S-reps

A slabular s-rep (see below) is a folded, non-branching skeletal sheet with a spoke vector emanating from every sheet position (to the implied object boundary) such that the spokes do not cross (within the interior of the object). It useful to consider the spokes in three classes: northside (from the top side of the folded sheet), southside, and at the fold (crest of the implied boundary). The fold spokes are adjacent to both a northside and a southside spoke. As illustrated in Fig. 1, an s-rep is discretely represented in the computer as a rectangular m×n grid of northside spokes, a rectangular grid of the same size of southside spokes, and a chain of 2m+2n-4 spokes at the fold (i.e., at the grid end indices). The remaining spokes in the (continuous) s-rep are interpolated from these discrete spokes using interpolation mathematics specialized to s-reps.

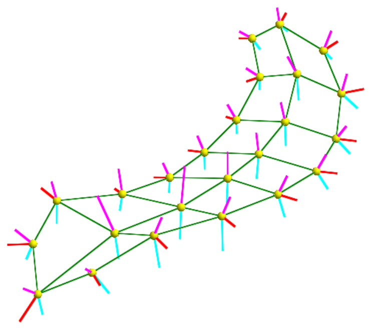

Fig. 1. An s-rep for a hippocampus

A file containing a discretely represented s-rep has .m3d as the extension in its filename. In that file northside spokes are labeled 0, southside spokes are labeled 1, and fold spokes are labeled 2.

The interpolated spokes are understood in an s-rep-based coordinate system, called (u,v). The u value of each northside or southside discrete spoke is the integer that is its first index (zero-origin) in the grid. Likewise, the v value of each northside or southside discrete spoke is the integer that is its second index (zero-origin) in the grid. Other u and v values for northside spokes are linearly interpolated between these integers. While the fold spokes are stored in the file in the rectangular array with extreme values of the first or second index of the grid (or both), they are understood to have a cycle of length 2m+2n-4, and interpolations can occur among any quad 4-tuple of adjacent spokes in that cycle.

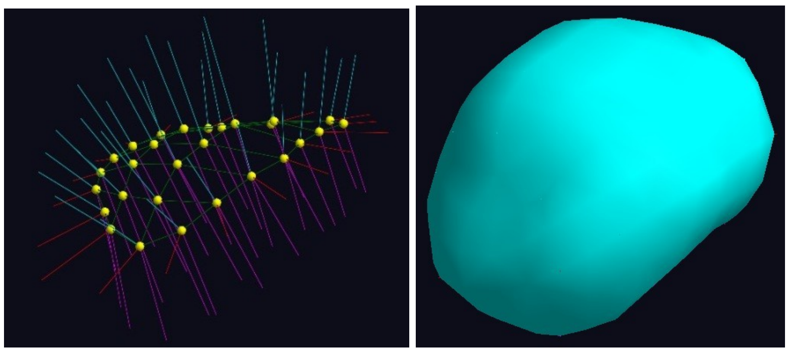

Fig. 1. A discrete s-rep for a prostate (left) and its implied boundary (right). In this illustration the northside and southside spokes as well as the corresponding spokes at the fold all share a common skeletal position, but this is not always the case.

Interpolation of spokes between the outer northside spokes to the fold spokes and from the fold spokes to the outer southside spokes are done by a different interpolation scheme than for the proper northside and southside regions. These spokes are understood to have (u,v) values that are xx

Positions in the interior and near exterior of an object are understood in “s-rep coordinates”, i.e., object-relative coordinates, when they are given by the u,v of the spoke along which they fall and , the fraction of the spoke length relative to the skeletal position. That is, at the skeleton =0, at the implied boundary =1, and halfway between these points =½.

Quasi-tubular s-reps, formed as curve of wheels of spokes, will be more completely described here once they are fully supported.

Discrete representations of s-reps are stored in ascii files that have .m3d as their extension. They describe an s-rep in a coordinate region of [0,1]^3. The main things these files contain are the s-rep type (slabular or quasitube and version), the values of m and n, the values of each discrete spoke (in the $[0,1]^3$ coordinates), and a similarity transform to map it to an appropriately scaled, translated, and rotated space. Each discrete spoke is specified by its tail position (on the skeleton), as x,y,z; its orientation given by a unit 3-tuple u_x, u_y, u_z; and its length r.

Some .m3d files are the result of statistical analysis of a population of s-reps. In that case the parts of the file described above represent the mean s-rep generated by that analysis. These files also include information about the principal modes of variation of the population of s-reps – this will be described in section V.

II. **Preparing Boundary Descriptions for Fitting S-reps**

The object data to which an s-rep is to be fit somehow represents the boundary of the object. Possibilities are piles of planar contours, binary images (with value 1 inside the object and 0 outside), and surface tiled with polygonal planar tiles. These need to be transformed into a signed distance image with the zero level surface representing the object being smooth. The following describes programs that are available to do this transformation.

A. **Transformation to a signed distance image that is not necessarily smooth**

From a binary image that does not have cubic voxels, you first need to interpolate that to a binary image that does have cubic voxels. That binary image need not be cubical. For speed reasons, we recommend that you use arrays with no more than 256 voxels on a side, and for accuracy reasons we recommend that you use no fewer than 64 voxels on a side. From that use the program Image2SignedDistanceMap to produce a signed distance image with the same size voxels.

From a tiled image (if you have triangular tiles) use the program Voxelise from Mathworks to create a binary image, and then use the steps above to turn the binary image into a signed distance image. Alternatively, there are probably packages on the web that include a program to turn a tiled image directly to a signed distance image.

From a pile of contours, use the PLUNC program xx to create xx. Then xx

B. **Smoothing a signed distance image while retaining the zero level set**

Pablo, the program that fits s-reps to distance images requires as input a distance image whose level surfaces, especially the zero level surface implying the object boundary, are smooth. To create a smooth signed distance image from one that represents the jagged boundary of a binary image or that comes from a tiled surface that is nonsmooth at the tile edges, use the anti-aliasing program antiAliasWrapperForPablo(input_file, output_destination, tolerance) with the input being the file containing the nonsmooth signed distance image and the output destination receiving the resulting smooth signed distance image. If the tolerance field is left out, the default is taken. To produce a file to which you can fit an s-rep using Pablo, the input image needs voxels that have the same size (w) in each of the three cardinal directions: x, y, and z.

The major parameter in the anti-aliasing program is the tolerance of the constraint in the distance of the resulting implied boundary to the center of the voxels in the implied boundary in the input. The default is set to 0.5w, where w is the voxel width, thereby keeping the boundary within the same voxel. We recommend using the default value. Using a larger value will yield a smoother boundary at the expense of larger movements of the implied boundary.

III. Fitting an S-rep to an Individual Object Description

Fitting an s-rep to an object description is done using the program Pablo. Pablo also is used to manually edit s-reps and to visualize both s-reps and the images to which they are fit. It is also used to visualize the modes of variation when a shape space is provided; discussion of this is left to section V. The editing and visualization operations are controlled by menu items and keyboard keys and mouse buttons as described in Appendix 1 in the manual entitled "Binary Pablo Developer's Manual".

While fitting can be manually engaged, it is almost always engaged by a command line that specifies the s-rep to be fit, the object distance image to be fit to, the location to store the result s-rep, and a "configuration file" that controls the fitting. The configuration file is an ascii file that specifies the stages of fitting to be applied and the parameters for each stage. The items in the configuration file will be discussed as each stage of fitting is presented.

Command line: `bin_pablo -po -cs config/config-srep.txt -image images<binary_image>.gipl -model atom_stage.m3d -outModel srep_stage.m3d`. There are many flags that can be chosen besides the "-po" flag specifying that optimization (fitting) is to be done without opening the Pablo display windows) and "-cs" specifying that what is next is the location of the configuration file. The flag "-h" yields a description of all of the flags that can be specified.

Almost all of the stages involve an optimization of an objective function. The configuration file specifies which terms are summed to form the objective function and what the weights of these terms are. Generally speaking, there are two types of terms: "image mismatch terms", i.e., those involving properties of the fit of the s-rep spoke ends to the object signed distance image, and "geometric atypicality terms", i.e., those involving geometric properties of the s-rep itself. As each term increases, the quality of the s-rep is taken to be worse. The precise meaning of these terms can be found in the paper [Merck 2007], which can be found at the website http://midag.cs.unc.edu/MIDAG_FS.html.

A. Creating an initial model

Creating a basic m-rep is done using the "Add Quad Figure" item in the "Edit" menu of the program Pablo. The user selects the size of the grid, i.e., the values of m and n implying the number of discrete spokes in the model. The number should be chosen so that with regular spacing of the spokes the curvatures of the boundary surfaces will be captured at the spoke tips.

In an initial model the grid-interior discrete spokes come in northside & southside pairs that share x,y,z positions of their tails (locations that are on the skeleton). Also, the grid-exterior discrete spokes come in northside, fold, and southside triples that share x,y,z positions of their tails (locations that are on the skeleton grid edge). The tuples of spokes sharing a skeletal position are called an "atom". Later operations may separate these spoke tails in an atom from each other.

In all atoms in an initial model the length of the northside and southside spokes in an atom are the same. Later operations may make these lengths somewhat different.

B. Alignment to object data

You should do your first fitting on an object that is somehow characteristic for the population of objects you will ultimately want to fit. That is, is should somehow be a median case.

Fitting an s-rep to object data in the form of a signed distance image begins with taking an s-rep to be fit and aligning it to the signed distance image. The alignment always involves translation and rotation (although either alone is supported) and can optionally also involve uniform scaling. The alignment method translates the s-rep model so that the center of mass of its implied object interior is the same as the center of mass of the distance image's implied object. It rotates the model about that center of mass (and scales the model about that center if that option is chosen) so that the criterion chosen from the following two is met.

1) Moments
2) Min distance magnitudes at boundary

For option 1, these alignment operations are accomplished by Pablo using the configuration file stage called the "Initialization Stage". The parameter "doMethodOfMoments" is a series of bit flags representing the different initialization options: 1 for translate, 2 for scale, and 4 for rotation. Simply add the numbers for the desired options (for example, a parameter value of 5 would be used for translation and rotation but no scaling).

C. Fitting an aligned s-rep to a first object description

With an aligned s-rep, the first stage of fitting works atom by atom while maintaining the common length of a northside spoke and its southside partner. Manual first.

Keeping a reasonable geometry at this stage requires the atoms to be heavily constrained. Iterative optimization of the objective function is used to modify the spoke parameters directly.

The irregularity penalty (unfortunately labeled "atomAverageNeighborMatch" in the configuration file) and spoke crossing penalty (unfortunately labeled "atomRSradPenalty" in the configuration file) need high weights at this stage. The mismatch between spoke ends and the actual object boundary (atomImageMatch) and the spoke directions deviation from orthogonality to the actual object boundary (atomImageNormalMatch) can have relatively lower weights at this stage.

Using the printout of the objective function terms.

This fitting to a first object description is the most difficult. Often it benefits from some manual editing of the atoms on the result of the first attempt, followed by a new alignment of this model, followed by an atom stage optimization initialized from this model. Indeed, this process may need repeating.

Understanding the quality of the fit requires visualization of s-rep against the object description. This can be done in two ways, described in section D.

D. Visualizing the S-rep vs. the Input Object Description

There are two ways to visualize the current s-rep against the object description: as contours on image slices, or by comparing the surfaces implied by the s-rep and image description.

Assuming the model and image have been loaded into Pablo, the s-rep can be visualized as contours on image slices as follows. First, choose the “Display Control” from the Pablo “Windows” menu. In the “Surface” tab, select the “Boundary” option. You should now see a colored curve on your image slices where it intersects the s-rep boundary. The “Image” tab provides controls for choosing which slice directions to display as well as for moving through these slices.

To visualize the s-rep and object descriptions as surfaces, you will choose “Display Control” from the Pablo “Windows” menu. Navigate to the “Images” tab and make sure all of the image slices are turned off. Next, under the “Surface” tab, select “solid” as the surface type. Next, select “Generate Tile Set From Image” under the “File” menu. You should now be able to compare the two surfaces. Various tile set properties (color, opacity, etc) can be modified in the “Tile Set” tab in the Display Control.

Fig. xx. Visualizations

E. Segmenting an S-rep from an Image

If instead of having a known, already segmented object, one wishes to extract an object from an image, a version of Pablo known as “Greyscale Pablo’ is applicable. Pablo supports reading an image in the following formats. The description of how segmentation to this image proceeds is left to a separate manual.

F. Fitting to later object descriptions

Once you have a decent fit to an object that is characteristic of the population, all the rest of the fittings are much easier. Pick a few cases that are still reasonably characteristic of the population. For each use the s-rep fit to the first object as the base model. For each target object distance image, align the base model to the image and deform the result using the result as the initialization of an atom stage. Simply throw away, for the time being, fits that are not very good. (visually or according to the values of objective function terms).

From the remaining fits, the decent ones, use the CPNS method described in section V to compute a mean s-rep, and refit all of the cases fit so far, successfully or not, by an alignment of this mean to the respective distance image followed by an atom stage. In these refits you can distinctly lower the irregularity penalty and raise the reference penalty (called “atomModelMatch” in the configuration file), which penalizes difference from the initializing s-rep. This will yield many more decent fits, from which you should compute a new mean via CPNS.

This new mean should be used to align and initialize fits to the rest of your population. Again choose from all of the fits (including the previously selected ones) the decent ones and compute a new mean via CPNS. Repeat all of the fits using this new mean to start. This process can be expected to yield decent fits to most members of your population.

Spoke stage

IV. Fitting S-reps to a Collection of Object Descriptions

The adequate fits that you have so far obtained, i.e, for most of the members of your population, do not have their spokes adequately in correspondence to generate the most informative probability distributions. *Later we will be providing code to optimize this correspondence directly by entropy minimization.* But at present we use the following approach to achieve both correspondence and adequate fits to all (excepting perhaps one or two) of the cases in the population of object distance images.

V. Generating a p(s-rep) Probability Distribution

A. Alignment of the s-reps

At least translational and rotational alignment of the s-reps from which the probability distribution is to be computed is necessary. If normalization of the size of the object is desired, the resulting probability distribution will not reflect size difference; otherwise it will (the normal desire). If you do want the probability distribution to reflect size differences, be careful that you do not input s-reps in the $[0,1]^3$ coordinate system but rather those that are scaled up or down to reflect their actual anatomic sizes. Producing these properly scale s-reps can be done as follows. xx

The alignments that are done must not be done via centers of mass, as was done for initializing an s-rep to an object signed distance image. Rather the CPNS method depends on the fact that they are done relative to the center of mass of the spokes' skeletal positions. The translational alignment for this purpose is done automatically by the CPNS code. Rotational alignment xx

B. Composite Principal Nested Spheres (CPNS)

CPNS is a method, implemented in Matlab, that, like PCA (principal component analysis) yields a mean s-rep and modes of variation. However, it recognizes that most of the features in an s-rep live on abstract spheres and as such are non-Euclidean features. There are also some Euclidean features, i.e., which live in a flat space. The method, developed by Sungyu Jung and described in [Jung papers], operates by determining, for each non-Euclidean entity (such as a spoke direction) xx a polar system (a north pole and a specified latitude) for each component sphere by computing best fitting subspheres, and thereby it computes a Euclideanized feature value corresponding to each non-Euclidean feature value. It also uses this “backwards” (decreasing in dimension of the sphere) approach to compute a mean s-rep that for each feature is within the cluster of feature values in the population.

After all the non-Euclidean features have been Euclideanized, all of the features, Euclidean and Euclideanized, each with mean zero, are appropriately weighted to make them commensurate, and PCA is applied to the resulting scaled features (this is the “compositing” step). This yields eigenmodes and principal variances in the Euclidean and Euclideanized features.

Transforming from the mean plus eigenmodes weighted by chosen scores to an s-rep requires knowing not only the eigenmodes and principal variances, but also the factors by which the features were made commensurate, the means of the originally Euclidean features, and the polar systems for each sphere and subsphere. All of this information, together with the mean s-rep, is presented in the .m3d file produced by CPNS.

CPNS has one setting and one parameter, both with default values. The setting is whether the best fitting subspheres are chosen among only geodesic (“great”) subspheres or among both great and small subspheres, depending on a hypothesis test as to whether the data noticeably falls near a small subsphere rather, for example, to being in a clump. The choice mode is the default. However, when the data is known to be clumped or it is important that the same choice be made in many applications of CPNS, as in a hypothesis test based on permutations, then the great circle mode should be chosen.

The parameter, used only when the choice mode is set, is the p-value of the hypothesis test, specifying how certain (how low a p value) the analysis needs to be that a small subsphere is correct choice before it makes that choice. The default is p=0.05, but smaller values are recommended when the choice for a small subsphere should be more conservatively made.

Modes visualization

CPNS is applicable not only to s-reps but also to other entities that are understood to live on a Cartesian product of a Euclidean space and a collection of spheres. This includes other object representations, including boundary point distribution models.

VI. **Computing a Classification Direction from 2 Classes of S-reps**

Classification between two classes of s-reps can be done using a method that finds a separating manifold in the feature space and then classifies according to which side of the manifold

VII. **Classifying an S-rep**

Given a separating direction vector in the feature space of Euclidean and Euclideanized variables derived from an s-rep, together with a threshold along the separating direction, classification of a new s-rep involves computing the required representation, projecting the result orthogonally onto the separating vector, and comparing the project result to the threshold. The DWD-based classification software whose use was described in section VI provides the polar system data and mean according to which the non-Euclidean variables in an s-rep can Euclideanized, the separation vector and threshold, and the feature weights needed to do the projection. The program xx uses this information to take an s-rep in its normal representation and produce the resulting class for that s-rep.

VIII. **Hypothesis Testing on S-reps**

Xx

IX. **Filling in the Parameters in a Pablo Configuration File**