GLiMR Landmarks for Procrustes Analyses

GLiMR uses several different approaches to locate landmarks on an artifact's surface for use in Generalized Procrustes Analysis (GPA). There are many ways to skin a cat and these methods are still evolving. No one method is currently appropriate for all projectile points. The narrative that follows explains the inputs and outputs of the Procrustes tools. Resulting feature classes computed relative to Procrustes landmarks are highlighted in yellow and summarized in Figure 13.

The goal is to create a set of landmark points around the perimeter of the object and also landmarks on the face surface. Each landmark point is solved for X, Y and Z and assigned a Procrustes number. The objective is to have each Procrustes number refer to a topologically similar location on different projectile points.

Currently there are two main approaches for determining landmark positions: *Radial* and *Rectangular*. A perimeter segmentation approach is in development.

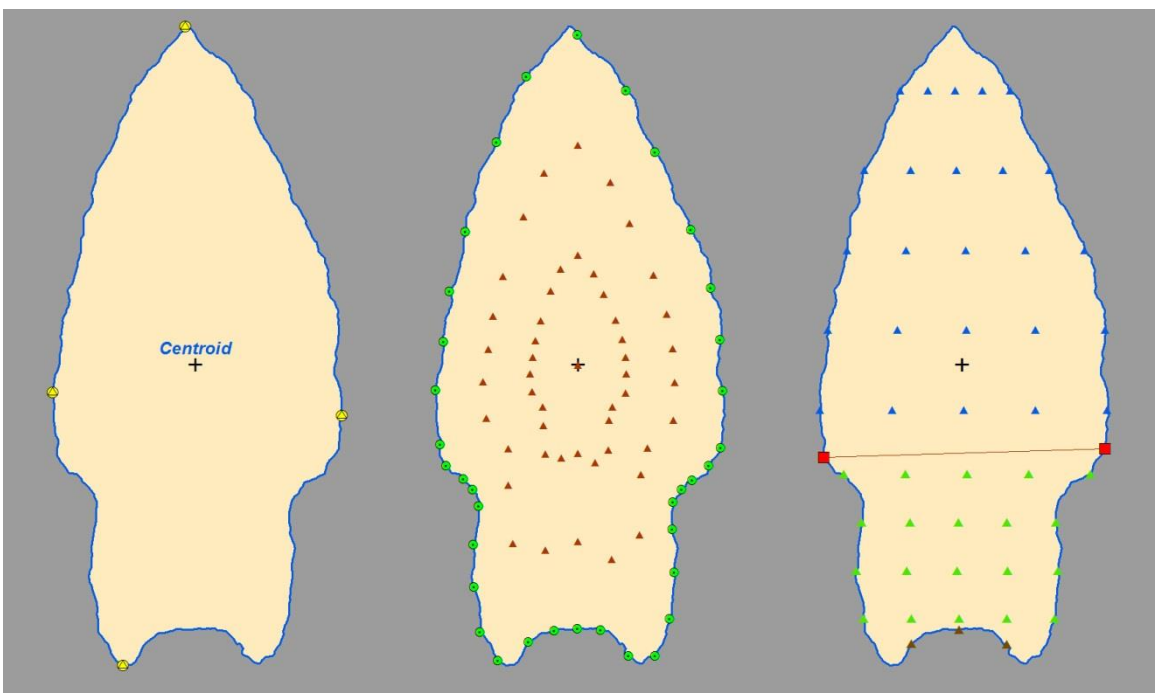

Figure 1. Outline, Radial and Rectangular landmark patterns

Although the most extreme cardinal points (top of tip, base of blade, farthest right, farthest left) are computed as *Outline_quad_pts*, they are not used in most landmark scenarios. Minor rotation of the point causes them to swap topologic position with other landmarks points. These points are in yellow on the *Outline* shown on the left side of Figure 1

Radial Landmark Points

- Radial Edge Points

The radial approach solves for the intersection of the artifact's outline and a fixed set of radial rays emanating from the centroid of the outline. To determine the radial edge points, GLiMR intersects a polyline feature class called *Starburst_edge* with the perimeter of the *Outline* feature class to yield a set of points (Figure 2 left image).

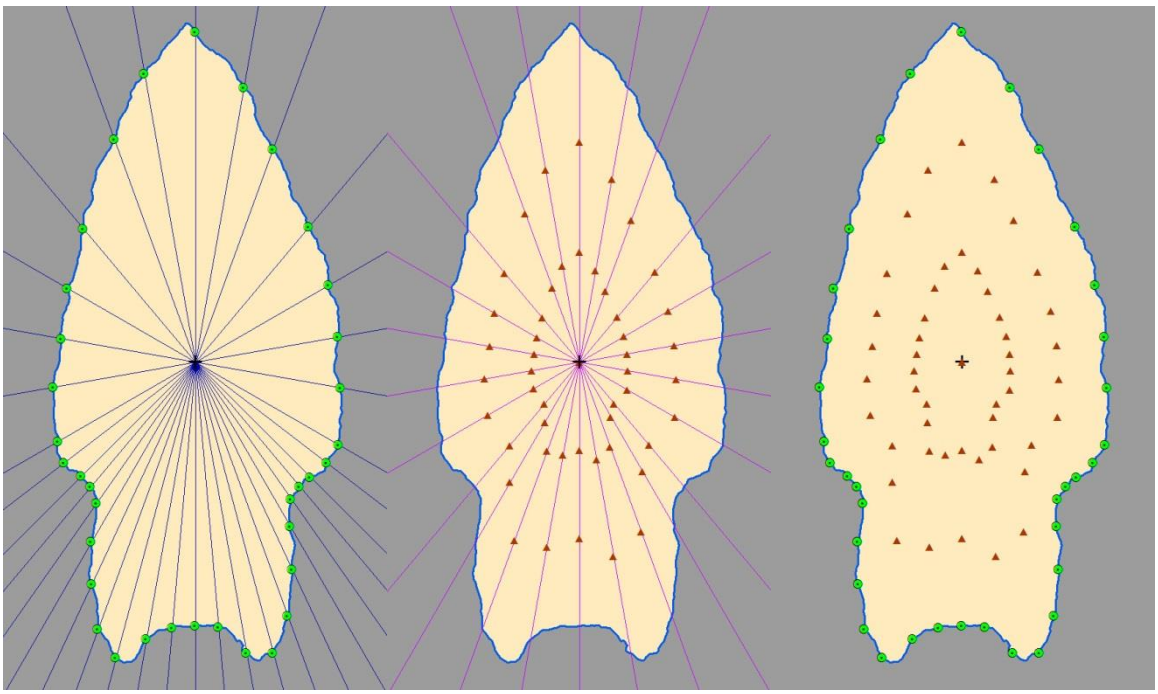

Figure 2. Radial rays and edge points, Radial rays and 2 internal rings and final radial result

The ray pattern for the edges is stored as a feature class named *Starburst_edge* in Template.gdb in the GLiMR_Templates directory. This feature class is copied to each local geodatabase and used. You may edit the number and position of radial rays in the *Starburst*_edge as needed. If the *Starburst_edge* or *Starburst_rings* feature class exists in a geodatabase, it is used. If not present it is copied from the Template.gdb geodatabase in the GLiMR_Templates directory.

The resulting points are placed in a feature class named *Procrustes_Radial_Edge_Pts* with the following attribute table.

Procrustes_Radial_Edge_Pts

OID *	Shape *	Type	ProcrustesPtNum	GKey	Label	X	Y	Z_Topo	Z_Smooth
1	Point Z	pcr_Ext_FC	1	297_1_1	Radial_1	-0.000001	26.451849	1.194838	1.262176
2	Point Z	pcr_Ext_FC	2	297_1_1	Radial_2	3.884099	22.027449	0.99968	1.311531
3	Point Z	pcr_Ext_FC	3	297_1_1	Radial_3	6.205299	17.048449	1.196399	1.521966
4	Point Z	pcr_Ext_FC	4	297_1_1	Radial_4	9.114199	10.861449	1.49198	1.546627
5	Point Z	pcr_Ext_FC	5	297_1_1	Radial_5	10.700899	6.177749	1.09724	1.408137
6	Point Z	pcr_Ext_FC	6	297_1_1	Radial_6	11.443599	2.017349	0.949482	1.172849
7	Point Z	pcr_Ext_FC	7	297_1_1	Radial_7	11.624699	-2.050251	0.813404	1.013943
8	Point Z	pcr_Ext_FC	8	297_1_1	Radial_8	11.494099	-6.636651	1.07075	1.026585

Figure 3. Attribute table for radial edge points (only part of records shown)

This attribute table includes the unique identifier GKey along with a unique Procrustes Number, Type and label for each point and along with its coordinates. Notice that for each point two different Z values are reported: Z_Topo is the actual elevation of the face at that point. Z_Smooth is the elevation of the smoothed surface fit to the point. The default smoothing is a 7mm circular moving average using Focal Statistics.

- Radial Internal Ring Points

A set of radially distributed internal ring points is also computed using a radial ray pattern feature class called *Starburst_rings* (also in the Template directory). The distance between the centroid and the edge along each ray is computed and equally divided to produce the number of rings specified. The default is 2 rings. (Figure 2 center image)

The *Starburst_rings* feature class typically has fewer rays than the *Starburst_edge* feature class. Figure 2 shows the *Starburst_edge* rays on the left have been densified to better quantify the haft area. If used for the internal ring points, oversampling would occur with many closely spaced internal ring points.

The resulting internal ring points are place in a feature class called *Procrustes_Radial_Rings_2*. The trailing digit _2 in the name refers to two rings specified. It would be _1 or _3 for one or three rings respectively.

- ***Pros*** *(Radial)*
  - Can yield the same number of edge points for most artifact shapes.
  - User can modify starburst ray pattern to adjust edge sampling in specific sectors.

- ***Cons*** *(Radial)*
  - Does not always place a point at absolute tip or base.
  - Some notch configurations 'shadow' some edges.
  - Non-uniform internal landmark placement

Each radial ray is projected from the centroid until it reaches the outline (edge) and a point is placed at the intersection of the ray and the outline. Certain notch configurations result in a ray intersecting the outline edge more than once. In this case, the intersection closest to the centroid is used and the ones further out are 'shadowed' and not used. This is to keep the number of Procrustes edge points the for every artifact.

Rectangular Landmark Points

A rectangular array of landmarks may be used only for stemmed points. This approach divides the artifact into a blade and a haft area and builds a rectangular array of row and column landmarks for each area. It is not appropriate for most notched points.

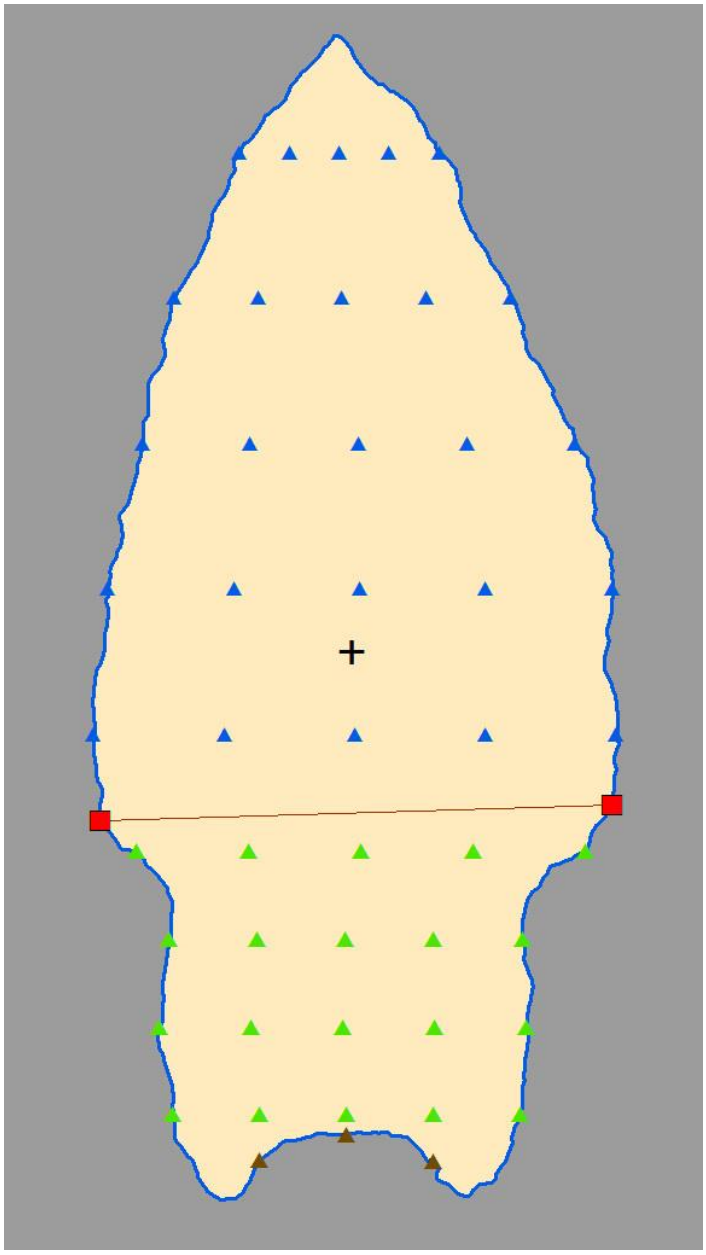

Figure 4. Rectangular Landmark Points

The blade and haft areas are processed independently. Each area is divided into rows and columns. The columns are spaced equally across the local width. The rows are spaced using a top and base setback,

then equally spaced along the Y axis within this region. In Figure 4 above, the blue triangles are the landmarks points, the green triangles are the haft landmarks and the three red landmarks at the base are additional base edge points. As you can see this scheme is not sampling the basal edge of the haft outline of this point adequately.

This approach requires determining blade and haft areas. This can be done for stemmed points by deriving a *haft collar* (grey line) from *haft points* (red squares shown in red in Figure 5). Haft points in turn are determined by the 'northernmost' quadrant point of the left and right reentrants (shown in green in Figure 5).

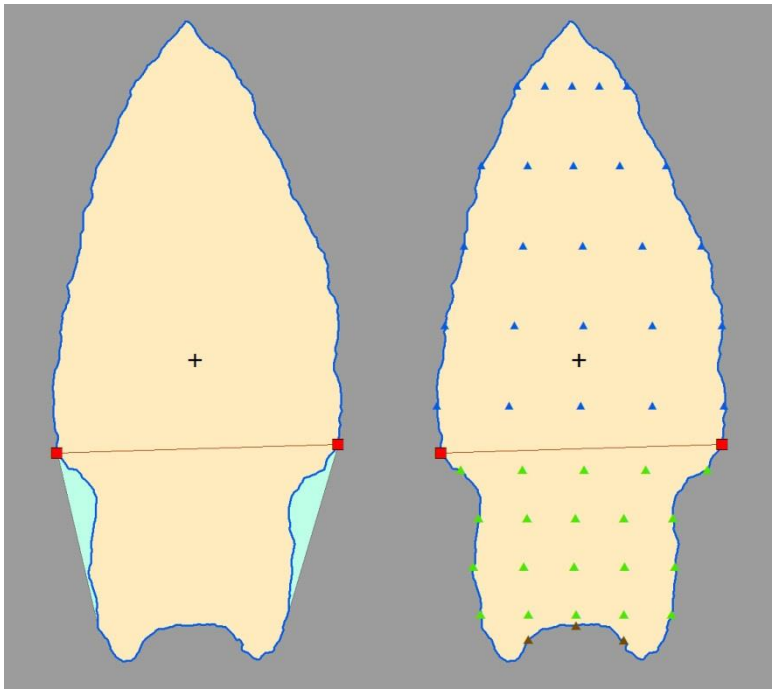

Figure 5. Haft Collar Determination

The midpoint of the haft collar give an approximation of the top of the blade-to-haft transition. The Y coordinate of this point is used, together with a setback to determine the row position for the bottom row on the blade and for the top row of the haft.

Creation of rectangular landmarks use setbacks to move the uppermost and lowermost row of the blade or haft array away from the top or bottom edge of the area. For example, as the position of the uppermost blade row approaches the tip, the narrowing width of the blade squeezes all of the columns together, resulting in oversampling a small area. By using a setback, the row can be moved away from the tip to an area of better areal distribution.

Setbacks are specified as a percent of the height of the blade or haft element. The blade height is measured from the blade tip to the Y value of the midpoint of the haft collar line. The haft height is similarly measured from the haft collar midpoint to the lowest Y value along the base outline.
The sample artifact in Figures 4 and 5 has a basal notch, so a haft base setback had to be increased to move the lowermost haft row above the basal notch. Procrustes requires the same number of landmarks for every artifact, so we cannot discard a landmark that lands in a notch, we have to adjust the row location. When analyzing a group of artifacts the most difficult shape will dictate your setbacks. Figure 9 has the four setbacks annotated.

The setbacks from the haft collar control the position of the lowermost landmark row of the blade and the uppermost landmark row of the haft. Haft-blade transitions occur over some distance and one side is commonly higher than the other resulting in a sloping haft collar line. The setbacks can be used to move rows away from the haft collar to avoid positioning a row that is measuring part haft and part blade. This may generate poorly correlative landmarks in GPA. The uppermost haft row in Figure 5 is such a row because it is still in the transition area. Increasing the setback for the haft top would correct this.

The rectangular scheme for blade and haft places landmarks at the left and right edges of the point and the surface in between, but does not sample the basal edge of the haft. The column positions of the internal (non-edge) landmarks of the bottom row are used to determine additional landmarks along the basal edge of the haft. These are shown in red on Figure 4.

The feature classes for the blade rectangular landmarks is named *Procrustes_Blade_Pts_5x5* where the _5x5 suffix references the number of rows and columns. Similarly the haft rectangular feature class is named *Procrustes_Haft_Pts_5x5* . Figure 6 shows the attribute table. Note that the Type field contains whether it is an edge or internal landmark and the Label field contains the row-column index within the array for internal landmarks.

Procrustes_Blade_Pts_5x5

OID *	Shape *	Type	ProcrustesPtNum	GKey	Label	X	Y	Z_Topo	Z_Smooth
7	Point Z	Blade_Edge_5x5	7	297_1_1	Edge_7	7.024699	15.618749	1.29456	1.532918
8	Point Z	Blade_Edge_5x5	8	297_1_1	Edge_8	-7.808401	15.618749	0.867169	0.890582
9	Point Z	Blade_Edge_5x5	9	297_1_1	Edge_9	3.885199	22.026449	0.996465	1.311098
10	Point Z	Blade_Edge_5x5	10	297_1_1	Edge_10	-4.922801	22.026449	1.479236	1.407559
11	Point Z	Blade_Internal_5x5	10001	297_1_1	Int_1_1	-5.584201	-3.604351	2.812044	2.667409
12	Point Z	Blade_Internal_5x5	20001	297_1_1	Int_2_1	0.166599	-3.604351	3.261374	3.236494
13	Point Z	Blade_Internal_5x5	30001	297_1_1	Int_3_1	5.917299	-3.604351	2.669269	2.513997

Figure 6. Example of rectangular landmark attribute table.

All of the attribute tables have the same structure, so they can be merged. As will be discussed in Hybrid Approaches below, selected landmarks out of different methods may be assembled to produce custom landmark schemes suited to your specific data set.

- ***Pros (Rectangular)***
    - Handles haft and blade area separately
    - Samples haft well
    - Uniform sampling of interior

- ***Cons (Rectangular)***
    - Stemmed points only
    - Requires haft collar points (derived from reentrants)
    - Poorly samples edges in haft-blade transition area

Perimeter Segmentation Landmarks

Perimeter segmentation landmark generation is not currently implemented in GLiMR, but is under development. A general discussion follows.

Some methods have been published that subdivide the perimeter length into equal segments and establish landmarks at segment boundaries. Although superficially attractive, this method has severe topological difficulties as the position of notches move about between different artifacts. A landmark on an blade edge in one artifact can appear deep in a notch is a different artifact, so it is not representing a similar topologic position. All landmark methods so far suffer from this problem to some extent.

Certain landmarks on the perimeter of notched points are attractive.

- Blade tip
- Tip of each tang
- Point on each notch closest to artifact centroid
- Midpoint of basal haft edge

Once these are established and verified, the perimeter between these points then could be further segmented and landmarks established. These have the benefit of not allowing a landmark to migrate like a flounder's eye to an inappropriate location.

Notched points require special treatment. A main requirement of Procrustes analysis is that the landmarks are positioned at key morphological positions that are topologically similar throughout a group of artifacts. Every artifact has to have the same number of landmarks, so a given landmark cannot be undefined in certain points.

Hybrid approaches

The radial scheme described above was developed because it could handle most points from foliate to stemmed to notched and yield a set of similar landmarks. The method did not required subdividing the point into haft and blade areas.

The rectangular approach was developed in recognition that the blade and haft might have different use and resharpening histories and should be examined separately. The radial approach did not adequately sample the haft. The rectangular approach was governed by row position (in Y dimension) and poorly sampled the haft-blade transition and had to have special treatment to sample the basal haft edge at all.

Procrustes Interior Landmarks

For our GPA and cluster analysis use, we built a hybrid assembly routine which combined radial edge landmarks with rectangular interior landmarks to yield a feature class named *Procrustes_Interior*. This is simply choosing subsets of landmarks and combining them and cleaning up the Procrustes numbering sequence.

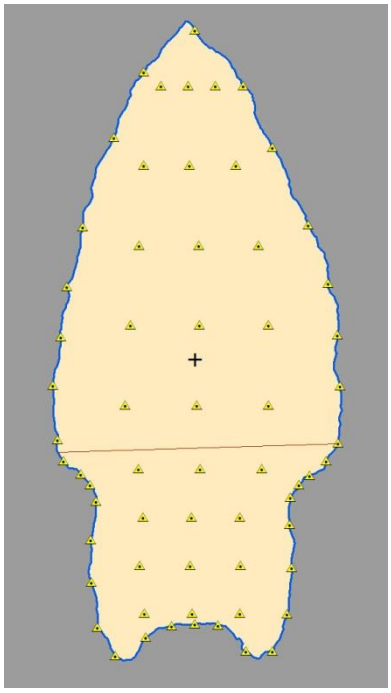

Figure 7. Procrustes Interior feature class (hybrid)

Procrustes_Interior

OBJECTID *	Shape *	GKey	Type	Ref	Label	ProcrustesPtNum	X	Y	Z_Topo	Z_Smooth
33	Point Z	297_1_1	RadialEdge	33	Radial_33	33	-7.928401	-11.323351	1.194465	1.703434
34	Point Z	297_1_1	RadialEdge	34	Radial_34	34	-9.135901	-9.136451	1.5374	1.504709
35	Point Z	297_1_1	RadialEdge	35	Radial_35	35	10.520099	-8.072851	0.739947	1.100054
36	Point Z	297_1_1	RadialEdge	36	Radial_36	36	-10.509201	-8.064451	0.892472	1.29346
37	Point Z	297_1_1	Blade5x5	1000	Blade_Int_1_1	37	-5.584201	-3.604351	2.812044	2.667409
38	Point Z	297_1_1	Blade5x5	1000	Blade_Int_1_2	38	-5.165101	2.803249	2.628574	2.611247
39	Point Z	297_1_1	Blade5x5	1000	Blade_Int_1_3	39	-4.427101	9.210849	2.677977	2.501141

1 (0 out of 63 Selected)

Figure 8. Procrustes Interior attribute table

By examining the Type, Ref (reference) and Label attributes for each point, you can determine its source. By using the Label field, you can extract specific row-column values.

The two hybrid feature classes *Procrustes_Interior* and *Procrustes_Exterior* are simply examples of piecing together specialized landmark data sets appropriate for a specific problem. Although these were assembled programmatically, it is fairly simple to build custom data sets either with queries in MS Access on the Procrustes Summary tables, or using ArcGIS database tools to select and merge the desired landmarks from the Procrustes Summary file geodatabases.

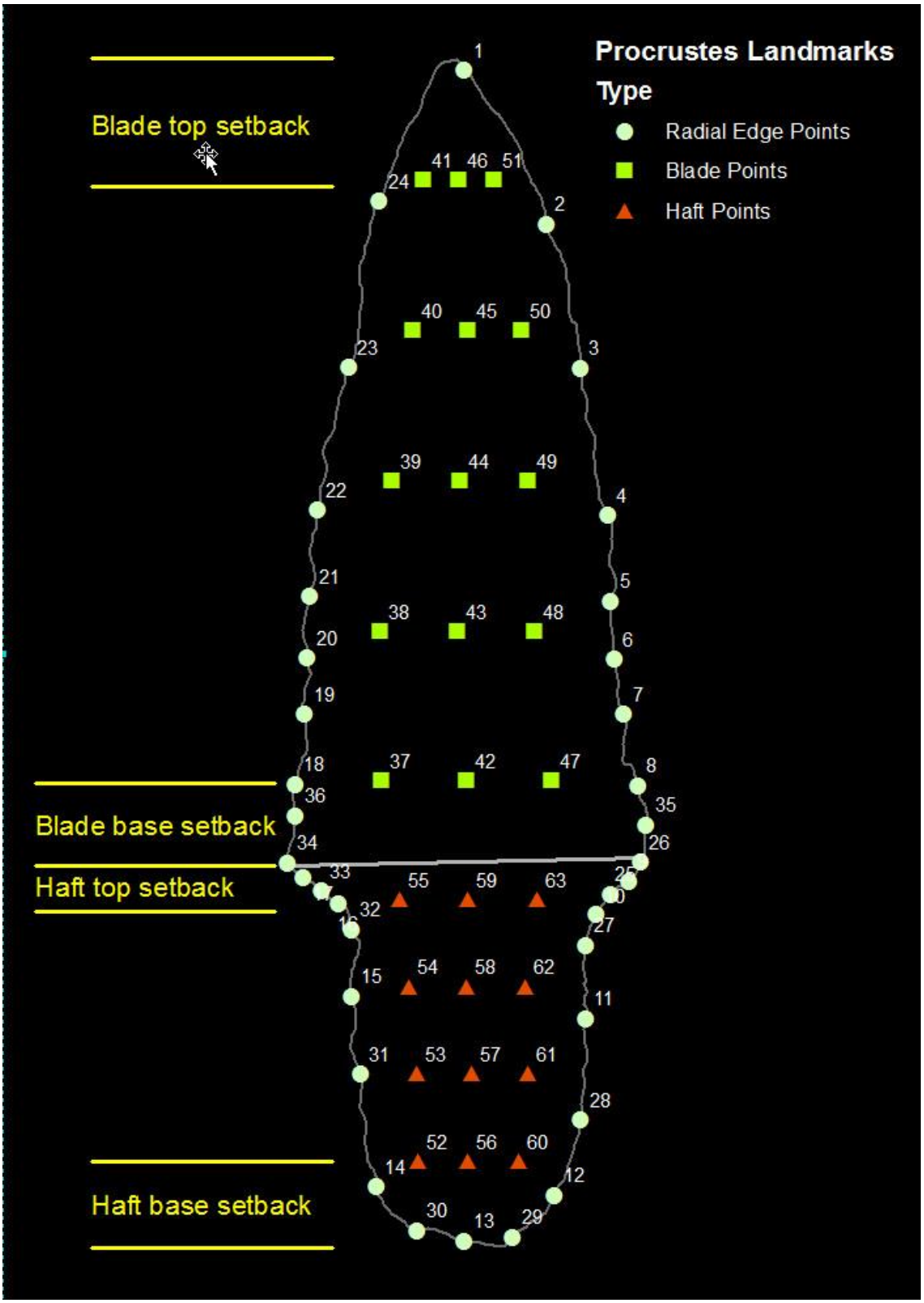

Figure 9. Hybrid Procrustes landmarks, setbacks and Procrustes numbers
(*Procrustes_Interior* feature class shown)

Procrustes Exterior Landmarks

The feature class *Procrustes_Exterior* is a hybrid containing radial exterior edge points and ellipse points characterizing the notches (reentrants) shape and orientation. It is designed to characterize planiform shape and characterize notches (reentrants).

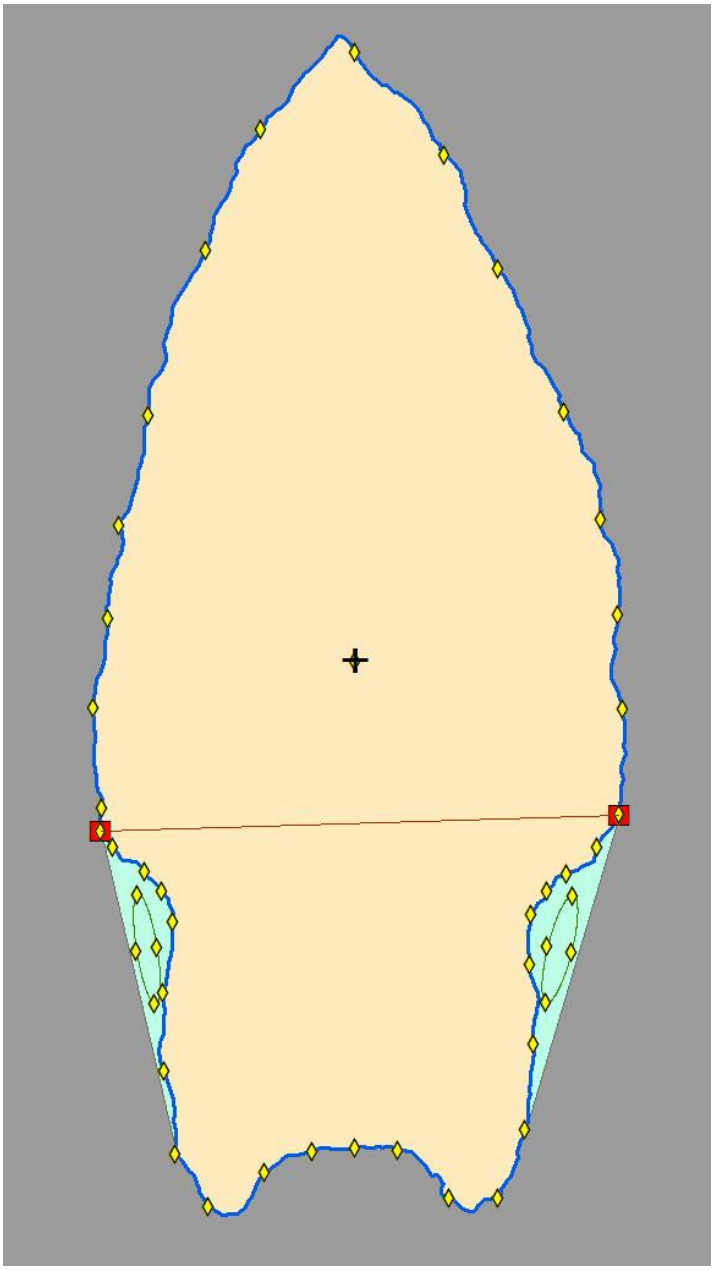

Figure 11. Procrustes Exterior Landmarks

This data set emphasizes the object shape and should be considered a 2D data set.

Re-computing landmarks with custom settings

The component feature classes for blade and haft landmarks and the *Procrustes_Interior* feature class may be recomputed with different settings for all geodatabases within the specified directory.

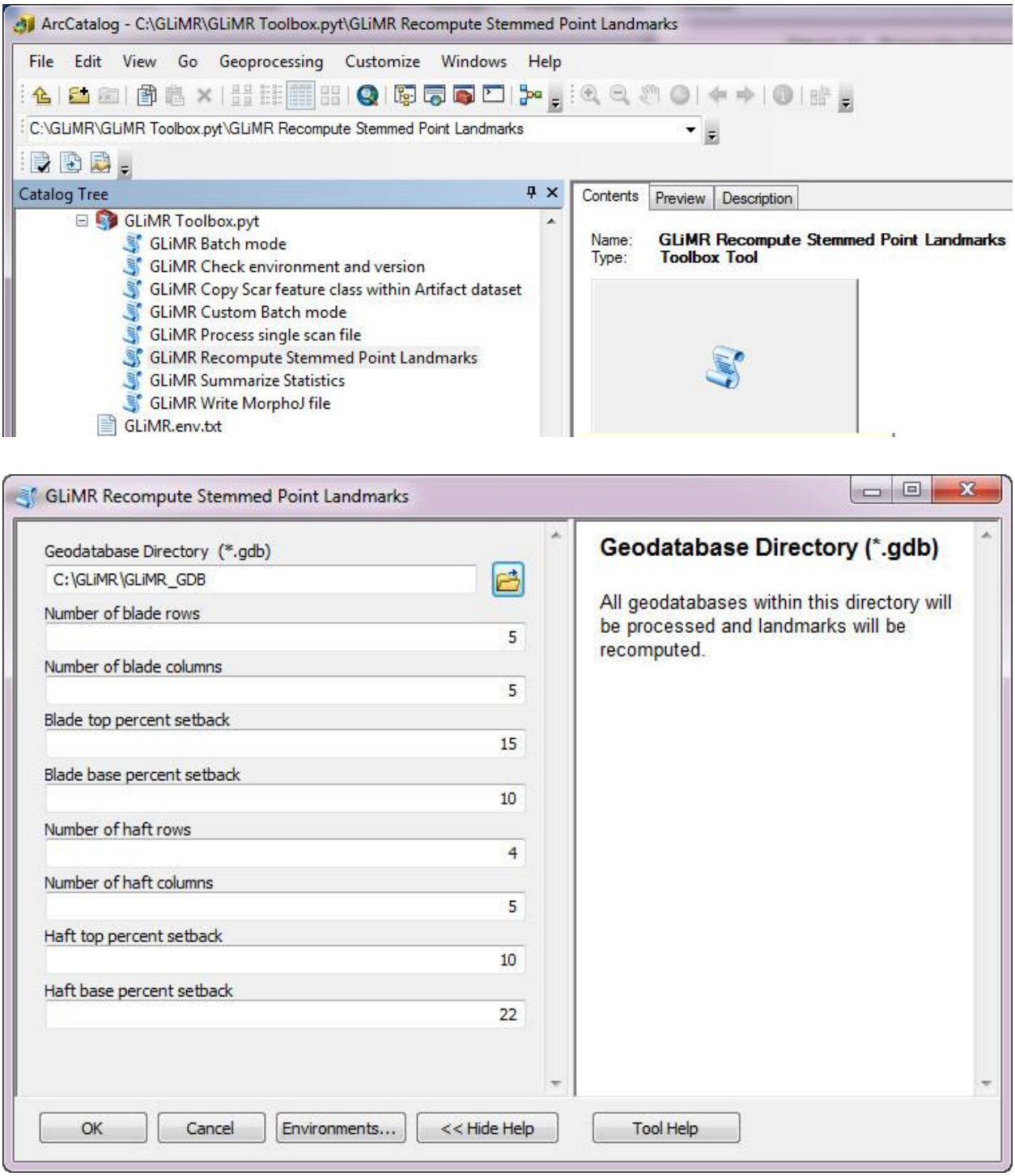

Figure 12. GLiMR Recompute Stemmed Point Landmarks tool

The GLiMR Recompute Stemmed Point Landmarks geoprocessing tool is found in GLiMR Toolbox.pyt toolbox in the GLiMR directory. Use ArcCatalog to navigate to the GLiMR Toolbox, then open the tool to display the dialog shown in Figure 12. The default settings are shown.

This tool expects that the artifacts being reprocessed have undergone initial processing either by GLiMR Batch node or GLiMR Process single scan file. The initial processing creates the Topo and Smoothed surface rasters and the Outline, Reentrant and quadrant and ellipse point feature classes used as Procrustes input.

Within the tool dialog shown, select the directory containing the geodatabases to be processed and set the rows, columns and setbacks for the haft and blade. Pressing **OK** will process all Procrustes data in the geodatabases in the specified directory and produce new feature classes related to rectangular and hybrid methods. The number of rows and columns are in the suffix of the feature class for blade and haft components.

Figure 13 below shows the feature class names generated by GLiMR.

Moving target…

GLiMR is an evolving tool. This document describes the status of the program as of November 2014. As we gather more experience on a wider set of artifacts we hope to refine and augment our approaches to lithic characterization.

Default feature classes

Below is a list of feature classes generated by GLiMR. The primary ones related to Procrustes landmarks are indicated with a grey background.

Contents | Preview | Description

Name	Type
BasePtsExt	File Geodatabase Feature Class
Blade_Haft	File Geodatabase Feature Class
Collars	File Geodatabase Feature Class
Curvature	File Geodatabase Raster Dataset
Curvature_Dip	File Geodatabase Raster Dataset
Curvature_Strike	File Geodatabase Raster Dataset
Flakes_ScarAuto	File Geodatabase Feature Class
Hull	File Geodatabase Feature Class
OrigPts	File Geodatabase Feature Class
Outline	File Geodatabase Feature Class
Outline_centroid	File Geodatabase Feature Class
Outline_quad_pnts	File Geodatabase Feature Class
Procrustes_Blade_Pts_5x5	File Geodatabase Feature Class
Procrustes_Blade_Tip_Pnt	File Geodatabase Feature Class
Procrustes_Exterior	File Geodatabase Feature Class
Procrustes_Haft_Basal_Pts_4x5	File Geodatabase Feature Class
Procrustes_Haft_Base_Pnt	File Geodatabase Feature Class
Procrustes_Haft_Pts_4x5	File Geodatabase Feature Class
Procrustes_Interior	File Geodatabase Feature Class
Procrustes_Radial_Edge_Pts	File Geodatabase Feature Class
Procrustes_Radial_Rings_2	File Geodatabase Feature Class
RE_Ellipse	File Geodatabase Feature Class
RE_Ellipse_pnts	File Geodatabase Feature Class
Re_entrants	File Geodatabase Feature Class
Re_entrants_quad_pnts	File Geodatabase Feature Class
Residual7mm	File Geodatabase Raster Dataset
Residual7mm_Cont_05	File Geodatabase Feature Class
ScarAuto	File Geodatabase Feature Class
Smoothed7mm	File Geodatabase Raster Dataset
Smoothed7mm_Cont	File Geodatabase Feature Class
Starburst_edge	File Geodatabase Feature Class
Starburst_rings	File Geodatabase Feature Class
TIN_edge	File Geodatabase Feature Class
TIN_Tri	File Geodatabase Feature Class
TIN_Tri_Near	File Geodatabase Feature Class
TIN_Tri_ScarAuto	File Geodatabase Feature Class
Topo	File Geodatabase Raster Dataset
Topo_Cont_05	File Geodatabase Feature Class
TopoExt	File Geodatabase Raster Dataset
Trend_Pts	File Geodatabase Feature Class

Figure 13. Feature class list

End notes on haft – blade determination

The rectangular methods described rely on splitting the face into haft and blade areas. The techniques to do so will evolve over time, but the current sequence is described below and certain dependencies are explained in detail.

Currently the line separating the blade and the haft is the haft collar line. The haft collar line connects the two haft points, which are the vertices nearest the blade tip (+ Y) of the left and right reentrant (Figure 5 left). Reentrants are the areas between the artifact *Outline* and the *Convex hull* and are created during initial processing. Each reentrant polygon has a *Type* attribute with the main labeled types being *Left*, *Right* and *Base.* GLiMR attempts to assign the main types automatically based on reentrant position and orientation.

One of the first quality control checks upon processing a projectile point for the first time is to check whether *Left*, *Right* and *Base* Type attributes are correctly assigned (if present). If the Type attribute needs to be corrected or assigned, it is important to correct the Re_entrants feature class in the geodatabase for the point in the **GLiMR_GDB_Edits** directory (not the main GLiMR_GDB directory).

The reason for this is based on the overall GLiMR workflow. The geodatabase (GDB) for each point in the GLiMR_GDB directory may be created, deleted or changed by GLiMR at any time during processing without requesting permission from the user. It is easier to recreate the parts and pieces from scratch than to track all user changes in 40 different feature classes. For each projectile point GLiMR maintains two GDBs, a working GDB in the GLiMR_GDB directory and a persistent GDB in GLiMR_GDB_Edits directory. The GDB in the working directory has all of the feature classes shown in Figure 13 and is used by all ArcMap files. The GDB in the persistent directory has only key feature classes which may have been manually edited.

Only a few feature classes need to be persistent such as *Re_entrants* and manual or edited flake scars. These persistent feature classes are stored in GDBs in the GLiMR_GDB_Edits directory. When a projectile point is initially processed, the *Re_entrants* feature class is created and copied to the corresponding GDB in the GLiMR_GDB_Edits directory. Any subsequent processing copies the persistent *Re_entrants* feature class back from the GLiMR_GDB_Edits directory into the working GLiMR_GDB directory.

For this reason, use the Re_entrants.MXD ArcMap file for a projectile point to examine the assignment of Left, Right and Base reentrants. This MXD color codes the reentrants based on the value of the Type attribute. If you are using the GLiMR Catalog application, scanning the Reentrant tab for the points is a fast way to identify potential problems.

Summary

Rectangular Procrustes landmarks placement is dependent upon separating blade and haft. This is accomplished based on the haft collar line connecting the left and right haft points. The haft points are the reentrant vertex closest to the blade tip for the Left and Right reentrant polygon. This requires that both the Left and Right reentrant polygons exist and are correctly defined by the Type attribute value.

So: ***Reentrants → Haft points → Haft line → Blade versus Haft determination → Landmarks***

If either Left or Right reentrant is not defined, no rectangular Procrustes landmarks will be generated.