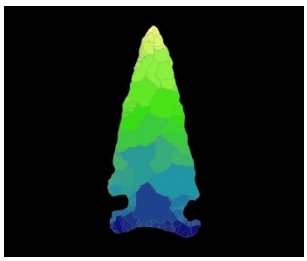
GIS-Based Lithic Morphometric Research (GLiMR) Software
Version 1.0

Loren G. Davis, Daniel W. Bean, Alex J. Nyers

Pacific Slope Archaeological Laboratory
Oregon State University,
Department of Anthropology
238 Waldo Hall
Corvallis, OR 97331



Creating the geometry metrics

Introduction

Overview Summary

- ***GIS Objects created for each projectile point***
- ***Triangles and Triangulated Irregular Networks (TINs)***
- ***Geometric properties of a triangle***
- ***Geometric properties of an entire TIN***
- ***Geometric properties of a flake scar polygon***
- ***Viewing GLiMR results***

Data generated by GLiMR

- ***GLIMR Feature Classes and their Attributes***
    - ***Points***
    - ***Outline***
    - ***Convex Hull***
    - ***Re-entrants***
    - ***TIN-based features***
        - ***Polygon meshes***
        - ***TIN mesh attributes***
- ***Raster data***
    - ***Topography***
    - ***Smoothed topography***
    - ***Residual topography***

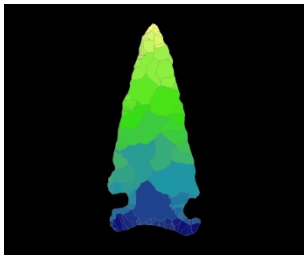
Creating flake scars
Aggregating flake scar statistics
Summary database tables

Methodology narrative
Landmark points
ArcGIS Geoprocessing steps

GIS-Based Lithic Morphometric Research (GLiMR) Software
Version 1.0

Loren G. Davis, Daniel W. Bean, Alex J. Nyers

Pacific Slope Archaeological Laboratory
Oregon State University,
Department of Anthropology
238 Waldo Hall
Corvallis, OR 97331



Introduction

The advent of 3D scanners allows the quantification of the detailed surface geometry of stone projectile points. Overall geometry measures such as thickness, slope and aspect can be determined at any location on the surface. In addition, flake scars can be determined either programmatically or manually and the flake scar position, pattern and geometry attributes within each flake scar can be measured and analyzed. The overall geometry of the point outline can also be quantified. GLiMR provides an initial framework of derived geometry metrics using generally available ESRI ArcGIS software tools. These metrics may then be exported for further analysis and classification with statistical software.

3D Scanners

Desktop 3D scanners acquire 3D point clouds of x,y,z data points. A typical scan may contain between 10,000 and 20,000 points. The point cloud data may be used in several ways. It can be assembled into seamless 3D polygon meshes for interactive display or split into two halves representing the top and bottom for surface analysis. Examining each of the two sides separately allows the use of GIS software for surface analysis.

The coordinate convention used throughout this paper orients a projectile point lying flat in the XY plane, with the blade tip pointing in the +Y direction. Z represents the thickness direction.

GIS approach

This paper examines surface and flake analyses that can be performed using off-the-shelf commercial GIS software. All of our analyses were performed using ESRI ArcGIS 10.2 and its 3D Analyst and Spatial Analyst extension modules. Although the procedures were automated using Python scripts and the ArcPy geoprocessing programming interface, similar analysis may be performed manually using the individual geoprocessing tools provided in ArcGIS user interface. We chose to only use surface analysis techniques available in ArcGIS, to avoid presenting ‘black-box’ software.

Overview Summary

To build morphometric data GLiMR creates and manipulates ArcGIS objects and their associated attribute tables. The ArcGIS 3D objects used include points, polylines, polygons, Triangular Irregular Networks (TINs), and grids. Every geometric element has associated tabular attributes used by GLiMR to store and make available computed properties.

The initial goal is to compute basic morphologic data from the scanned point data that represents the measured geometry, independent of interpretation. The morphologic data may be used alone or aggregated by overlay polygons (typically representing flake scars). The outputs are derived GIS geometry objects and their numerical attributes together with an associated MS Access database allowing export for analysis in other software.

GLiMR's only input is a text file containing (x, y, z) triplets for the points of one side of the scanned projectile point as ArcGIS points. The points are translated in X and Y such that their centroid is located at (0, 0). A set of objects and attributes shown below are then created based on these points.

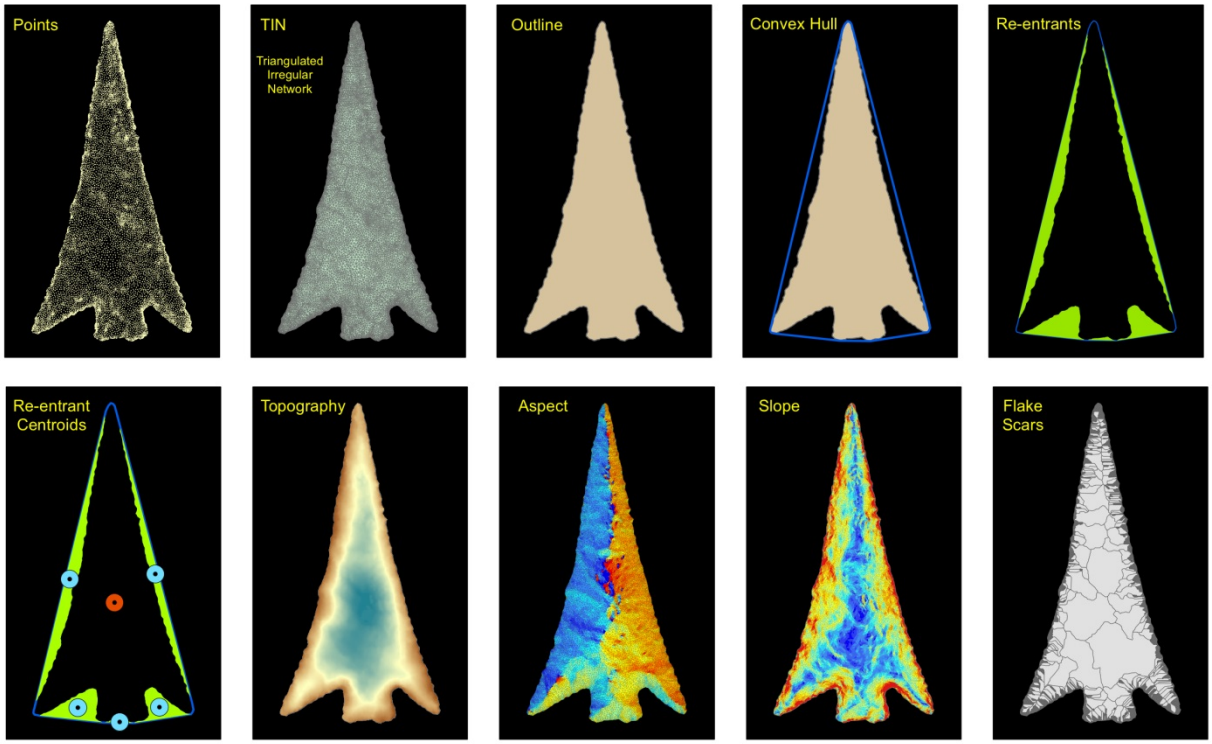


Figure 1 Some properties of an entire point

GIS Objects created for each projectile point

- **Points**
  One point is created for each x,y,z triplet of the input file. Points can be displayed in 2D or in 3D for visual quality control.

- **TIN**
  A Triangulated Irregular Network (TIN) is the set of triangles connecting all points such that each triangle is as close to equilateral as possible. The topography TIN represents a linearly interpolated elevation surface. Each triangle of the TIN has properties of slope, aspect, 2D area, 3D area, min, max and mean elevation, as well as volume above the zero plane. These properties are the basis of all derived surface metrics.

  The TIN is represented as an ArcGIS mathematical TIN object and is used to create 3D polygon or polyline representations of it. Polygons and polylines can have attributes which are used to store computed properties and can be symbolized or colored based upon these attributes.

- **Topography**
  Topography is a grid representing the surface elevation of the projectile point. The default topo is created as a grid with a cell size of 0.05 mm using a minimum curvature method.

- **Smoothed Topography**
  Smoothed topography is a grid of generalized topography created by using a 7mm moving average circle to compute a smoothed surface. This represents the generalized artifact shape, suppressing the small scale effects of dish-shaped flake scars. It is used in computing a residual topography surface and can also be used to compare generalized artifact shape between different objects.

- **Residual Topography**
  Residual Topography ('Residual') is created by subtracting the elevation of the smoothed topography from the original topography. It emphasizes <u>local</u> highs and lows and is used for automatic flake scar determination.

- **Other grids**
  Many other grids may be derived from the topography grid, such as slope and various curvature grids. Three curvature grids are computed by default. GLiMR uses grids in determining flake scars, but uses TIN data for generating all flake aggregation data.

- **Outline**
  The Outline is a polygon representing perimeter outline of the point. The outline has properties such as height, width, area, perimeter length and centroid. The outline is

created by processing the TIN to determine the point cloud perimeter. The centroid of the outline defines (0,0) origin for x, y coordinates.

- **Convex Hull**
  The convex hull is the smallest convex polygon containing all of the points in the XY plane. Think of it as shrink-wrapping your points in the XY plane. A convex hull is a generalization of the point shape without any notches or indentations.

- **Re-entrants**
  The difference between the outline and the convex hull define a set of polygons termed 'Re-entrants'. These polygons represent the notches and inward indentations along the edge of the outline. Re-entrant properties such as position, area and orientation are useful for analyzing and classifying side and basal notches.

- **Flake scar polygons**
  Flake scar polygons represent the boundaries between adjacent flake scars. The flake scar polygons may be manually created or automatically created by GLiMR.

  Flake scar polygons are used to compute the position, area and orientation of the scars themselves, and also to aggregate statistical data about elevation, slope, and aspect of the topographic surface beneath them.

  Manually drawing flake scar polygons accurately registered to a 3D scan is very time-consuming. Manually drawn scar polygons for hundreds of projectile points may not be practical in light of the five to ten hours each side of each point might require. In practice, creating flake scar polygons is much more complicated than simply sketching lines on a photo image.

  GLiMR will utilize manual flake scar polygons if provided, but also automatically creates flake scar polygons. The automatic scar polygons may be used as a starting point for manual editing of revised scar polygons or used as-is if appropriate to the current task.

Triangles and Triangulated Irregular Networks (TINs)

Geometric properties of a triangle

Each triangle between three adjacent scanned points have intrinsic properties of slope, aspect, 2D area (projected to plan view), 3D surface area, volume from the zero plane, as well as min, max and mean X, Y and elevation. Each triangle also has a spatial context relative to its location within the outline, such as the bearing and distance to the nearest edge. Because the (0, 0) origin is at the centroid of the projectile point outline, the XY position indicates position within the outline (e.g. positive X and Y indicate a position in the upper right quadrant of the point).

Geometric properties of an entire TIN

The entire TIN may be displayed colored by elevation, slope or aspect. This will show the selected property at the resolution of the scan with no filtering due to gridding or aggregation by flake scar polygons.

Geometric properties of a flake scar polygon

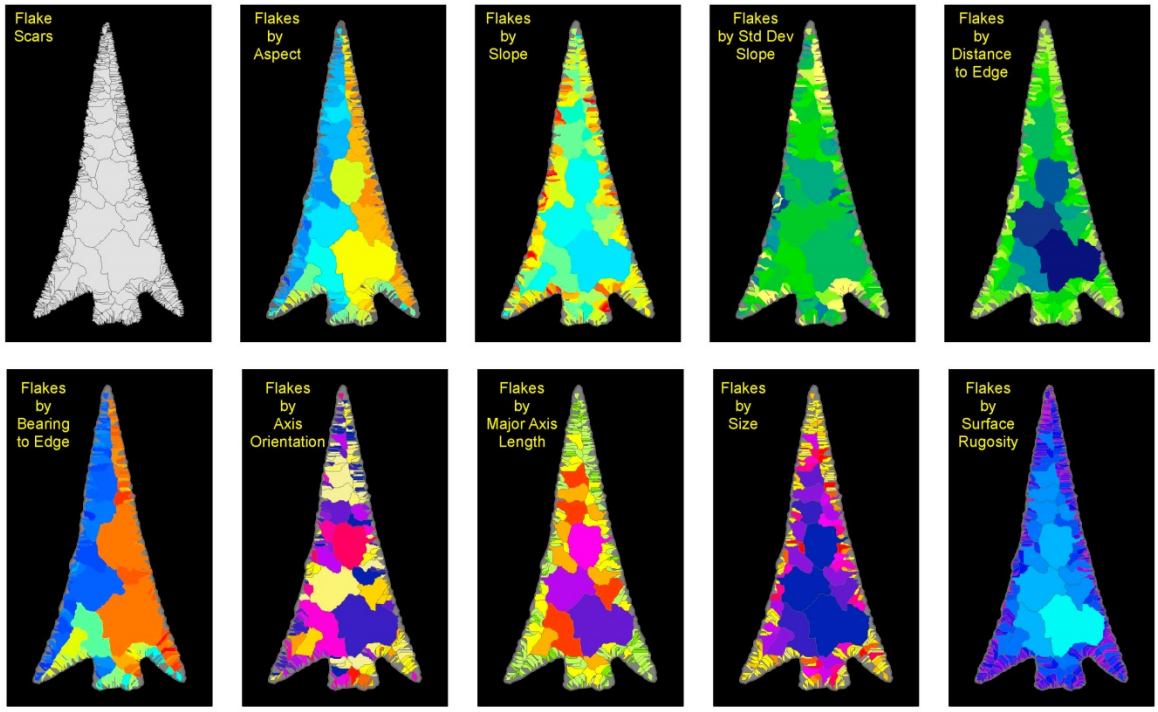


Figure 2 Flakes colored by geometric properties

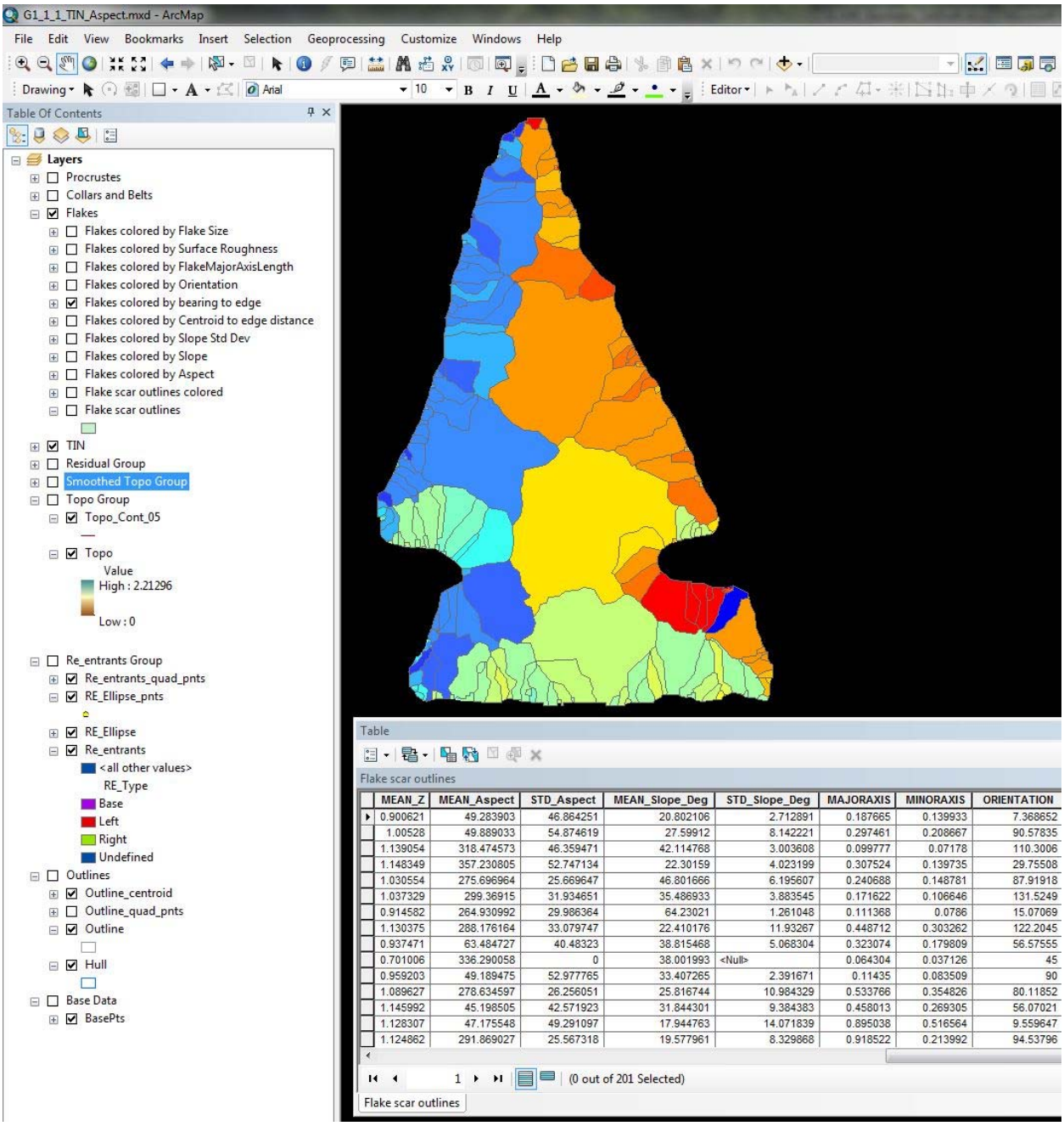
Each flake scar polygon overlies many triangles of the TIN representing the surface topography within that individual flake scar. Both intrinsic properties of the scar polygon and aggregate statistics of all of the triangles within the polygon can be determined.

Intrinsic flake scar properties include size, perimeter length, ellipse orientation, ellipse ratio, centroid position and bearing/distance to nearest edge of the projectile point.

Aggregate flake scar properties include mean, minimum, maximum and standard deviation of slope, aspect, and elevation. Surface roughness can be approximated by the ratio of the total flake 3D surface area to the total flake 2D area (corrected for slope).

Viewing GLiMR results

When GLiMR is run, it reads an ASCII file of scanned point coordinates and creates an ArcGIS geodatabase for that point including the objects listed above. It also creates an ArcMap (.mxd) file with links to the geometries in the geodatabase with appropriate symbology for many of the computed properties.



MEAN_Z	MEAN_Aspect	STD_Aspect	MEAN_Slope_Deg	STD_Slope_Deg	MAJORAXIS	MINORAXIS	ORIENTATION
0.900621	49.283903	46.864251	20.802106	2.712891	0.187665	0.139933	7.368652
1.00528	49.889033	54.874619	27.59912	8.142221	0.297461	0.208667	90.57835
1.139054	318.474573	46.359471	42.114768	3.003608	0.099777	0.07178	110.3006
1.148349	357.230805	52.747134	22.30159	4.023199	0.307524	0.139735	29.75508
1.030554	275.696964	25.669647	46.801666	6.195607	0.240688	0.148781	87.91918
1.037329	299.36915	31.934651	35.486933	3.883545	0.171622	0.106646	131.5249
0.914582	264.930992	29.986364	64.23021	1.261048	0.111368	0.0786	15.07069
1.130375	288.176164	33.079747	22.410176	11.93267	0.448712	0.303262	122.2045
0.937471	63.484727	40.48323	38.815468	5.068304	0.323074	0.179809	56.57555
0.701006	336.290058	0	38.001993	<Null>	0.064304	0.037126	45
0.959203	49.189475	52.977765	33.407265	2.391671	0.11435	0.083509	90
1.089627	278.634597	26.256051	25.816744	10.984329	0.533766	0.354826	80.11852
1.145992	45.198505	42.571923	31.844301	9.384383	0.458013	0.269305	56.07021
1.128307	47.175548	49.291097	17.944763	14.071839	0.895038	0.516564	9.559647
1.124862	291.869027	25.567318	19.577961	8.329868	0.918522	0.213992	94.53796

Figure 3 ArcMap screen of GLiMR results

Data generated by GLiMR

GLiMR generates a file geodatabase for each point and related ArcMap (.mxd) and image (.jpg) files, plus a summary MS Access database containing data for all projectile points.

GLiMR keys

All GLiMR tables carry an internal ID called GKey. GKey is unique for each scan file and consists of the concatenation of the Catalog number, the Face number and the scan Version.

For example, Catalog # 123, Face 2, Version 1 results in GKey = 'G123_2_1'

Catalog number is a unique long integer maintained by GLiMR. GLiMR tracks associated scan file names, locations and checksums, plus local artifact numbers or identifiers.

Face is simply the digit 1 or 2 for the two sides of the scanned object. A similar code tracks photos and allows face 3 to represent a side image.

Version defaults to 1 and is incremented if additional scans of the same side of the same object are obtained, perhaps with different settings or equipment.

GKey is useful for downstream analysis when data from multiple points are managed concurrently. Other ancillary data many be joined using GKey within the database.

Units

The internal units are those of the scan file. In all of our studies the units are millimeters, so lengths are in mm, area in mm^2. Default raster cell size is 0.05 mm (50 microns).

Angles are measured in degrees. Horizontal angles are presented as bearings with +Y ('north') = 0, increasing clockwise.

GLIMR Feature Classes and their Attributes

Points

The point feature class has no GLiMR attributes. The points are shifted in space so their origin is located at the centroid of the point outline.

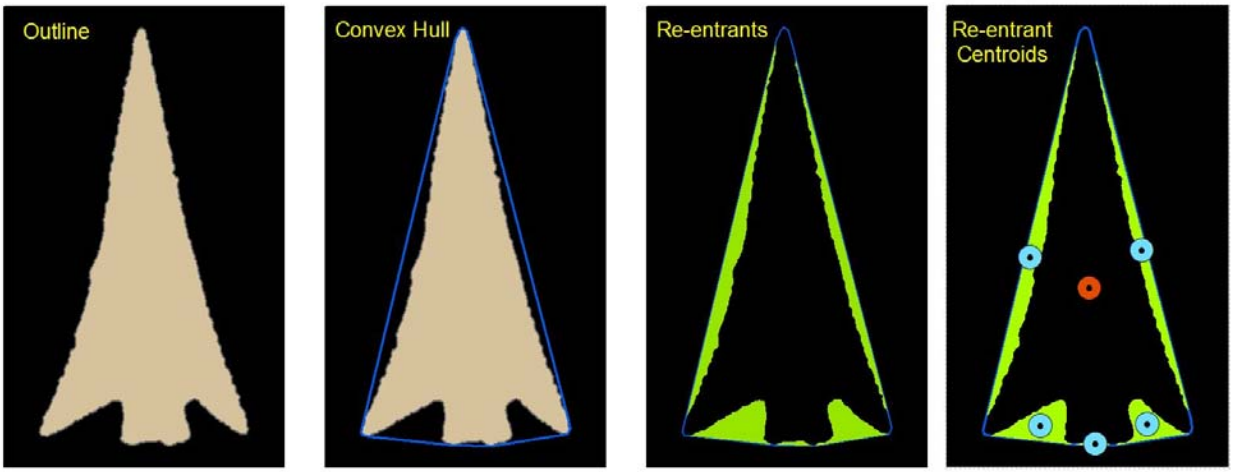


Figure 3 Outline, Convex Hull and Re-entrants

Outline

The Outline feature class is a single polygon representing the point outline.

Outlines

Field Name	Data Type	Description
GKey	Text	GLiMR key is the concatenated <Catalog + Face + Version> numbers.
Height	Number	Height of bounding box of artifact (Y dimension)
Width	Number	Width of bounding box of artifact (X dimension)
Half_Thickness	Number	Z range of scanned points
HW_ratio	Number	Height / Width Ratio
CENTROID_X	Number	X Centroid of outline (will be the X origin)
CENTROID_Y	Number	Y Centroid of outline (will be the Y origin)
Outline_Area	Number	Area of the Outline
Hull_Area	Number	Area of the Convex Hull
Outline_Length	Number	Length of the perimeter of the Outline
Hull_Length	Number	Length of the perimeter of Convex Hull
EXT_MIN_X	Number	Minimum dimension in X of the bounding box of the outline
EXT_MIN_Y	Number	Minimum dimension in Y of the bounding box of the outline
EXT_MAX_X	Number	Maximum dimension in X of the bounding box of the outline
EXT_MAX_Y	Number	Maximum dimension in Y of the bounding box of the outline
MinX_X	Number	X coordinate of point at Minimum X of any vertex of the outline
MinX_Y	Number	Y coordinate of point at minimum X of any vertex of the outline
MaxX_X	Number	X coordinate of point at maximum X of any vertex of the outline
MaxX_Y	Number	Y coordinate of point at maximum X of any vertex of the outline
MinY_X	Number	X coordinate of point at minimum Y of any vertex of the outline
MinY_Y	Number	Y coordinate of point at minimum Y of any vertex of the outline
MaxY_X	Number	X coordinate of point at maximum Y of any vertex of the outline
MaxY_Y	Number	Y coordinate of point at maximum Y of any vertex of the outline

Figure 4 Outline feature class attributes

The description of each Outline attribute is described on Figure 4, which is simply an image of the Access database table structure of GLiMR output.

The Height, Width and Half-Thickness are the dimensions of the bounding box of the point cloud. These represent the maximum range of X, Y and Z values and may differ from caliper measurements, especially if the scan file is rotated or the object itself is highly distorted. The measure tool within ArcMap may be used to accurately determine the distance between any two points of the scan.

The Half-thickness is the Z distance from the lowest value (which is zero because GLiMR shifts the points to make it so) to the thickest (highest Z) point. The perimeter points are not coplanar, so the zero plane is not the same for two faces of the same point. The sum of the two half-thicknesses for a point does ***not*** equal the thickness of the actual object. This version of GLiMR does not reconcile the scans of the two faces of a point in the z dimension.

The X and Y values of the centroid of the Outline are included. These will be very close (less than 100 microns) to (0,0). The source of the difference is discussed in the methods section.

The area and perimeter length of both the Outline and the Convex Hull are included in the Outline feature class attribute table. The Convex Hull values are included supply a reference area not influenced by notches and edge indentations. Comparing the perimeter lengths of the outline versus convex hull in edge sections away from notches may supply a metric of edge serration.

Various coordinates are also supplied for the maxima and minima in X and Y dimensions.

Convex Hull

The Convex Hull attribute table contains an area and length, but these data are included in the Outline table for convenience. The Convex Hull attributes also contain similar coordinate data for the maxima and minima in X and Y dimensions.

The Convex Hull and Outline feature classes each contain only one polygon object, but with many attributes. We will next be working with polygon feature classes such as re-entrants or flake scars that contain many polygon objects, each with many attributes.

Re-entrants

Re-entrant polygons represent the area outside of the point outline and inside of the Convex Hull outline. Re-entrants may be used to characterize notches and notch styles. Larger re-entrants include corner, side or basal notches if present.

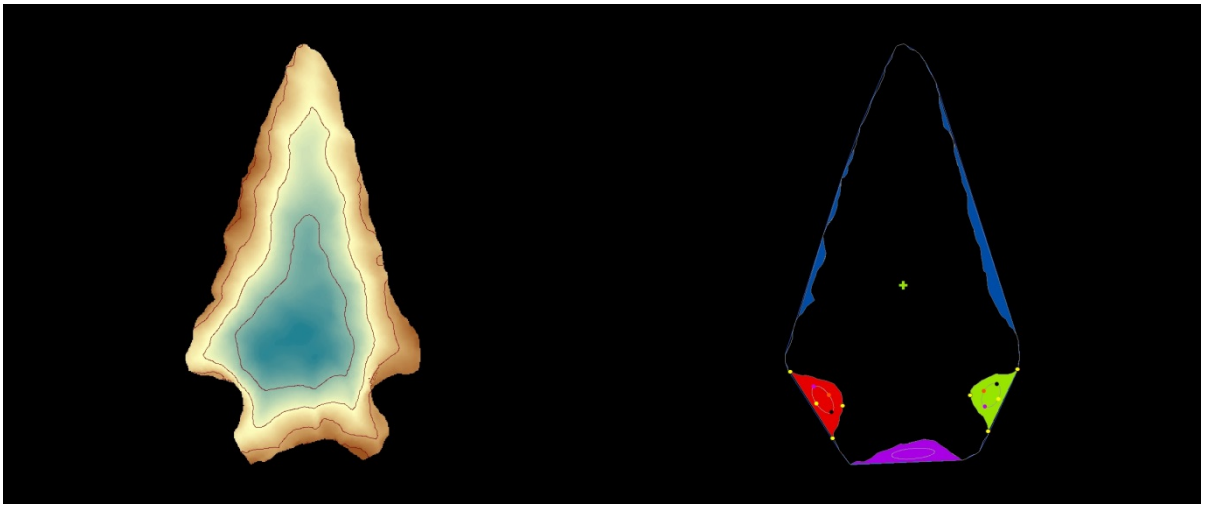


Figure 4 Point topography and re-entrants

A brief diversion into details is required to understand re-entrants. The outline and re-entrant polygons very nearly overlie on another if a point outline is somewhat convex. There will be a larger number of small edge indentations that would result in many very small re-entrants. The symmetrical difference geoprocessing tool used to create re-entrants has a cluster tolerance value representing a distance within which the polygons can be considered coincident. The current setting in GLiMR is 0.05 mm (50 microns). This has the effect of breaking the string-of-pearls initial re-entrants into non-touching polygons and eliminating tiny re-entrants.

Notched points commonly have one, two or three re-entrants significantly larger than the rest. These may represent basal-notched, symmetrical side-notched or corner-notched or both, respectively. Points with concave edges will also have re-entrants unrelated to notches. GLiMR creates an attribute called re-entrant type (‘RE_Type’) and attempts to set it to ‘Left’, ‘Right’ or ‘Base’ based on size, position and symmetry. In Figure 4, they are colored red, green and magenta respectively. Re-entrant type may be manually edited in ArcMap as needed and a copy of the edits will be maintained by GLiMR.

Reentrants

Field Name	Data Type	Description
GKey	Text	GLiMR key concatenated <Catalog + Face + Version> numbers. e.g. "G123_2_1"
OBJECTID	AutoNumber	Object ID in ArcGIS feature class
CENTROID_X	Number	X Centroid of reentrant
CENTROID_Y	Number	Y Centroid of reentrant
MAJORAXIS	Number	Length of major axis of ellipsoid fit to reentrant
MINORAXIS	Number	Length of minor axis of ellipsoid fit to reentrant
ORIENTATION	Number	Ellipse fit to re_entrant: orientation of major axis (longest axis) relative to X axi
EllipseRatio	Number	MajorAxis/(MajorAxis + MinorAxis) 0.5 for equant (round) ones and larger nu
RE_Size	Number	Diameter of largest inscribed circle that fits in flake
RE_Type	Text	Class of re-rentrant [Base, Left, Right, etc]
Pct_RE_Size	Number	Reentrant Size (diameter of largest circle with feature) Normalized to percent c
Pct_Centroid_X	Number	Normalized X coordinate of rentrant centroid. Normalized to percent of entire
Pct_Centroid_Y	Number	Normalized Y coordinate of rentrant centroid. Normalized to percent of entire
Pct_Hull_Area	Number	(Reentrant area / Convex Hull area) * 100.
MinX_X	Number	X coordinate of point at Minimum X of any vertex for this reentrant polygon
MinX_Y	Number	Y coordinate of point at minimum X of any vertex for this reentrant polygon
MaxX_X	Number	X coordinate of point at maximum X of any vertex for this reentrant polygon
MaxX_Y	Number	Y coordinate of point at maximum X of any vertex for this reentrant polygon
MinY_X	Number	X coordinate of point at minimum Y of any vertex for this reentrant polygon
MinY_Y	Number	Y coordinate of point at minimum Y of any vertex for this reentrant polygon
MaxY_X	Number	X coordinate of point at maximum Y of any vertex for this reentrant polygon
MaxY_Y	Number	Y coordinate of point at maximum Y of any vertex for this reentrant polygon
Reentrant_Length	Number	Length of the perimeter of the reentrant
Reentrant_Area	Number	Area of the reentrant

Figure 5 Re-entrant feature class attributes

Each reentrant has the attributes shown above including area, perimeter length, centroid position and coordinates for vertices representing the maxima and minima in the X and Y dimensions.

In addition, three scale-independent values are computed:

Pct_Centroid_X = Centroid_X / Width (Width is width of outline)
Pct_Centroid_Y = Centroid_Y / Height (Height is height of outline)
Pct_Hull_Area = Reentrant_area / Hull_Area (Hull area is area of Convex Hull)

These are useful for inter-point notch comparisons. Note that Pct_Centroid_X and Pct_Centroid_Y are independently scaled.

Reentrant size ('RE_Size attribute) represents the diameter of the largest circle that can be fit within the re-entrant. (GLiMR carries the diameter, not the radius described in ESRI documentation).

Reentrants also carry ellipse data (Orientation, MajorAxis, MinorAxis, Ellipse ratio) generated by ArcGIS Zonal Geometry tools. Paraphrasing from ESRI documentation: Zonal geometry tools approximate the geometry of each reentrant by creating an ellipse fixed at the centroid of each polygon. The eigenvalue and eigenvectors of each polygon are calculated. The orientation of the ellipse is in the direction of the first eigenvector. The ratio of the major and minor axes of the

ellipse is the same as the ratio of their eigenvalues. Refer to http://resources.arcgis.com/en/help/main/10.2/index.html#/How_Zonal_Geometry_works/009z000000ws000000/ for geoprocessing method details. A general description follows.

The general shape of each re-entrant is approximated by an ellipse centered on its centroid with properties of major axis length, minor axis length and orientation.

Orientation is defined as the angle between the +Y axis and the ellipse major axis. It has a range between 0 and 180 degrees. Orientation is not a vector, so an orientation of 30 degrees represents a line oriented on a 30º – 210º compass orientation with no direction implied. In GLiMR, orientation is presented as a bearing with zero being 'north' and angle increasing clockwise (This is different than the ArcGIS convention).

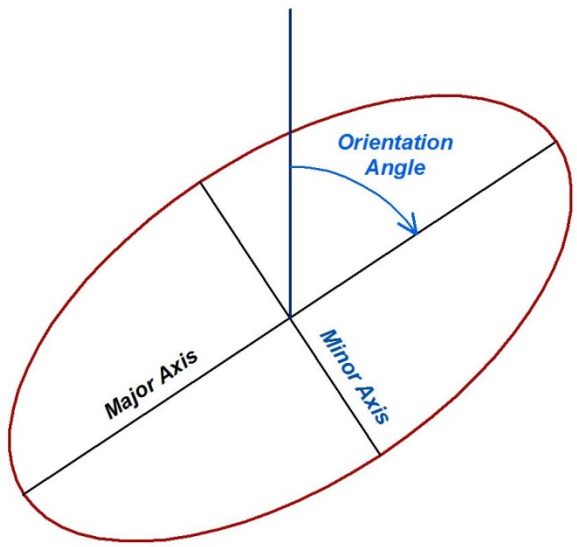


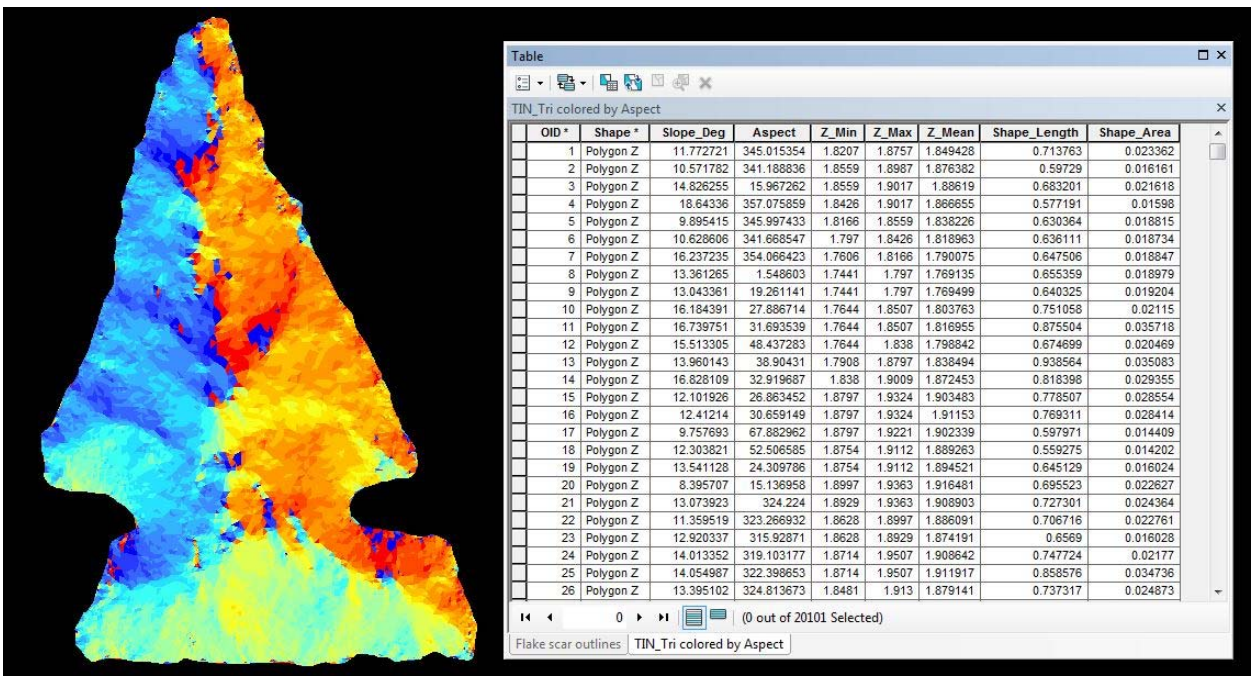
Figure 6 Ellipse orientation angle

A separate point feature class is created named 'RE_ellipse_pnts' that contains the coordinates of the four points defined by the intersection of the major and minor axes and the ellipse perimeter. These may be used in Procrustes approaches to notch analysis.

TIN-based features

The Triangulated Irregular Network (TIN) of the input point cloud is the basis of the bulk of the analyses computed by GLiMR. This TIN represents the topography of the face being examined.

The initial TIN produced by ArcGIS includes triangles filling the entire convex hull, including filling reentrant areas related to notches or concave segments of the point outline. A process called delineation (described in methods) trims away these extraneous triangles to yield the point outline. The delineated TIN used to create a polygon mesh representation and a polyline representation of the TIN. The polygon mesh feature class is the main object used for viewing and symbolizing the TIN.



Table

TIN_Tri colored by Aspect

OID *	Shape *	Slope_Deg	Aspect	Z_Min	Z_Max	Z_Mean	Shape_Length	Shape_Area
1	Polygon Z	11.772721	345.015354	1.8207	1.8757	1.849428	0.713763	0.023362
2	Polygon Z	10.571782	341.188836	1.8559	1.8987	1.876382	0.59729	0.016161
3	Polygon Z	14.826255	15.967262	1.8559	1.9017	1.88619	0.683201	0.021618
4	Polygon Z	18.64336	357.075859	1.8426	1.9017	1.866655	0.577191	0.01598
5	Polygon Z	9.895415	345.997433	1.8166	1.8559	1.838226	0.630364	0.018815
6	Polygon Z	10.628606	341.668547	1.797	1.8426	1.818963	0.636111	0.018734
7	Polygon Z	16.237235	354.066423	1.7606	1.8166	1.790075	0.647506	0.018847
8	Polygon Z	13.361265	1.548603	1.7441	1.797	1.769135	0.655359	0.018979
9	Polygon Z	13.043361	19.261141	1.7441	1.797	1.769499	0.640325	0.019204
10	Polygon Z	16.184391	27.886714	1.7644	1.8507	1.803763	0.751058	0.02115
11	Polygon Z	16.739751	31.693539	1.7644	1.8507	1.816955	0.875504	0.035718
12	Polygon Z	15.513305	48.437283	1.7644	1.838	1.798842	0.674699	0.020469
13	Polygon Z	13.960143	38.90431	1.7908	1.8797	1.838494	0.938564	0.035083
14	Polygon Z	16.828109	32.919687	1.838	1.9009	1.872453	0.818398	0.029355
15	Polygon Z	12.101926	26.863452	1.8797	1.9324	1.903483	0.778507	0.028554
16	Polygon Z	12.41214	30.659149	1.8797	1.9324	1.91153	0.769311	0.028414
17	Polygon Z	9.757693	67.882962	1.8797	1.9221	1.902339	0.597971	0.014409
18	Polygon Z	12.303821	52.506585	1.8754	1.9112	1.889263	0.559275	0.014202
19	Polygon Z	13.541128	24.309786	1.8754	1.9112	1.894521	0.645129	0.016024
20	Polygon Z	8.395707	15.136958	1.8997	1.9363	1.916481	0.695523	0.022627
21	Polygon Z	13.073923	324.224	1.8929	1.9363	1.908903	0.727301	0.024364
22	Polygon Z	11.359519	323.266932	1.8628	1.8997	1.886091	0.706716	0.022761
23	Polygon Z	12.920337	315.92871	1.8628	1.8929	1.874191	0.6569	0.016028
24	Polygon Z	14.013352	319.103177	1.8714	1.9507	1.908642	0.747724	0.02177
25	Polygon Z	14.054987	322.398653	1.8714	1.9507	1.911917	0.858576	0.034736
26	Polygon Z	13.395102	324.813673	1.8481	1.913	1.879141	0.737317	0.024873

0 (0 out of 20101 Selected)

Flake scar outlines | TIN_Tri colored by Aspect

Figure 7 TIN polygon feature class and attributes

The TIN polygon feature class represents topography, but may also be symbolized (colored) by any of its attributes. The figure above colors each triangle of the TIN by its aspect (bearing of downhill direction).

Each triangle has attributes of slope, aspect, area, perimeter length, and maximum, minimum and mean elevation. Slope is in degrees from horizontal and aspect is presented as a compass bearing in degrees.

These TIN attributes are the basis for all flake scar aggregate statistics.

Raster data

GLiMR creates several ESRI grid surfaces of topography, smoothed topography, residual topography and curvature grids. Grids are clipped to the point outline. These grids are used for visual display, contour generation and flake scar boundary determination. The flake scar aggregate statistics are generated from the TIN, not these grids.

Topography grid

A Topo grid is created from the original points. Internally GLiMR uses a cell size of 0.05 mm (50 microns) and a minimum curvature spline method (Spline with Barriers). This surface is extrapolated past the edge of the outline, and then clipped back to the outline to eliminate edge effects (details in Methods section).

Smoothed grid

A Smoothed grid is created from the extrapolated topographic surface using a 7mm moving average circle in the ArcGIS Focal Statistics geoprocessing tool. This produces a smooth representation of the projectile point without local flake scar irregularities. It is a generalized shape of the point.

Residual grid

A Residual grid is created by subtracting the Smoothed grid from the Topo grid. This accentuates the local <u>relative</u> highs and lows and is used to compute flake scar boundaries.

Other grid possibilities

Many other geoprocessing tools for grid analysis may be run starting with the Topo grid. Various curvature methods highlight flake and conchoidal fracture features nicely. Three curvature grids are computed: overall curvature, ‘dip curvature’ along the direction of steepest slope and ‘strike curvature’ along the direction of level slope. Strike and Dip curvature components are measured in directions perpendicular to one another in map view.

This concludes the basic geometry portion of the GLiMR analysis. The basic TIN data with elevation slope and aspect may be used for aggregation and statistics based on any set of overlay polygons. Although this document will focus on flake scar polygons, any polygonal subsets may be processed for aggregate statistics.

The outline, convex hull and reentrant polygons supply basic shape metrics and form a framework to sample the topographic or smoothed surface for further shape analysis or statistical methods. With the exception of re-entrant type (Left, Right, and Base notches) there is no ‘interpretation’ to this point, simply repeatable computed metrics.

About Flake Scars

Stone tools are created by the performance of a sequence of behavioral actions. Because the process of flintknapping is subtractive—percussion and pressure flaking remove different sized flakes from an objective piece and leave behind negative scars that mark their exact spot of removal—we can measure technological behaviors from the morphometric patterns held on an artifact's surfaces. Of course, tools that are created through an extended series of lithic reductions will superimpose new flake scars onto old ones, commonly removing any evidence of earlier stages of technological behavior. Because of this, some lithic tool manufacturing processes will retain only the final stages of manufacturing in a knapped artifact's flake scar patterns.

To analyze statistical data from flake scar patterns, GLiMR requires a set of flake scar polygons accurately registered to the 3D scan coordinate system. These may be produced manually or approximated via software.

Creating Flake scar polygons

Manual flake scars

On first glance, creating flake scars may simply appear to be an exercise of artifact study and careful drawing of interpreted scar boundaries on a projectile point photo. In practice it is significantly more complicated and time-consuming.

The first hurdle is accurately registering a photo of the projectile point to the associated 3D scan. The tolerances are very tight (0.3 mm is an unacceptably large error) and reconciling minor parallax issues between the orientation of the scan data and the camera position rapidly becomes a complex photogrammetric exercise. There are various tools of increasing sophistication to match imagery to topography within ArcGIS. Full ortho-rectification of your photo to the point cloud topography is time consuming and still may not yield a satisfactory match.

Once the photo is acceptably registered, polylines are drawn on the interpreted scar boundaries. These polylines must have their endpoints snapped exactly to adjacent flake scars so the result can be successfully converted into a polygon set with no gaps or overlaps. Our experience has been that flake scar polygons for one face of one projectile point may take well over five hours to create. GLiMR has a naming convention: Manual scar feature classes should be named starting with 'Scar', such as 'Scars_by LGD'

GLiMR-generated flakes scars

GLiMR automatically creates flake scar polygon features. The creation details use a watershed basin analogy and are described later in the Methodology section. Several different flake scar

polygons are created for each point using different sensitivity levels. They are named 'zScarAuto' with a suffix to show the sensitivity level used. One of these ('zScarAuto_05') is copied to 'ScarAuto' for use in aggregating flake statistics from the TIN. If one of the other auto scar feature classes is more appropriate, simply delete the old 'ScarAuto' and rename the appropriate 'zScarAuto' to something that starts with 'Scar' (within the same geodatabase as the point).

Any feature class beginning with 'Scar' will have an aggregated flake scar statistics feature class constructed using the same root name, but beginning with 'Flake'.

For example, ScarAuto will yield a Flakes_ScarAuto aggregated feature class. Similarly 'Scars_by_Fred' would yield 'Flakes_Scars_by_Fred'

Feature class names starting with Scars and Flakes are special to GLiMR and should be used with care.

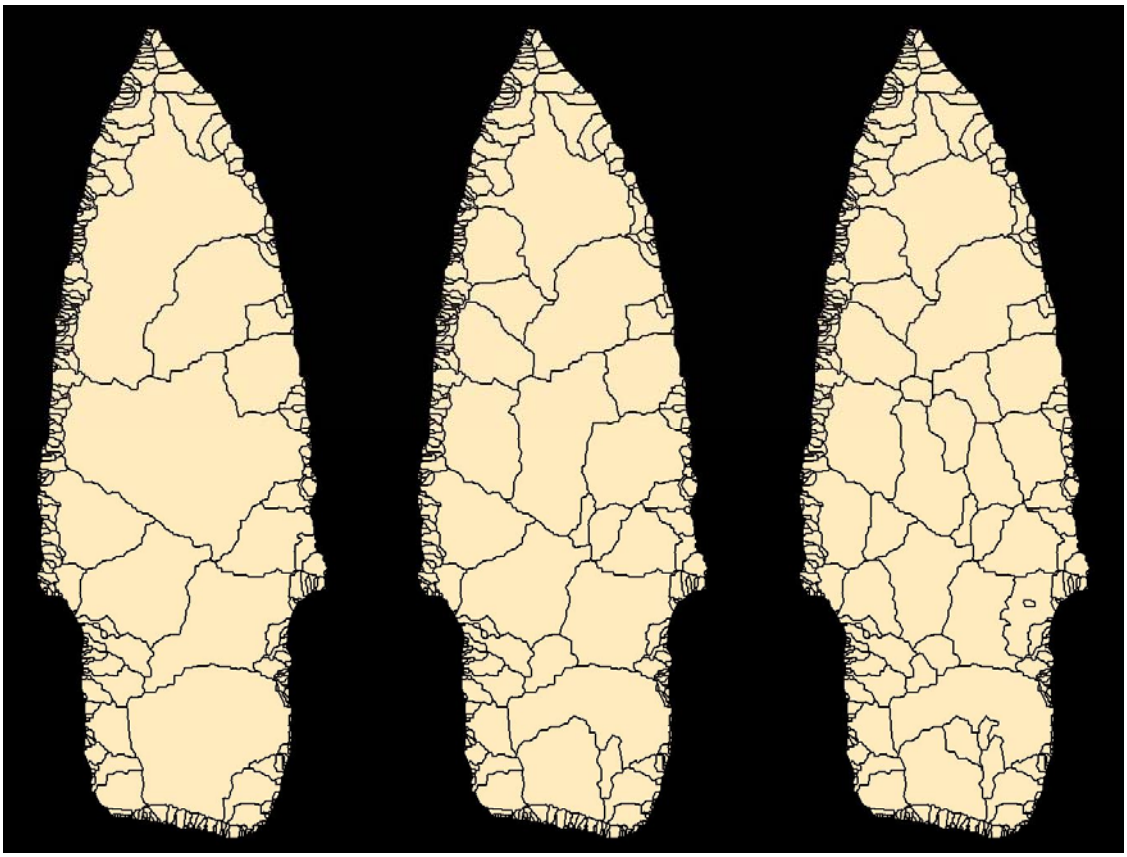


Figure 8 Different sensitivities for auto scar generation (zScarAuto)

Hybrid-Approach

GLiMR-created flake scars differ from manually interpreted flake scars. We are investigating a hybrid approach that starts with auto-scars and provides tools to help you edit them by

combining adjacent flakes or splitting flakes while maintaining rigorous topology. This feature is not included in GLiMR 1.0.

Luckily, the task of editing flake scars is topologically identical to editing land parcels in your county assessor's office. Merging and splitting adjacent lots with no gaps or overlaps is the same problem. ESRI has produced many specialized topology and parcel editing tools which may be applicable to the flake scar editing problem.

Persistent data

By default GLiMR will regenerate components of a geodatabase from scratch or regenerate missing ones for the current point. GLiMR will replace the entire geodatabase in some cases. Manually edited data such as manual flake scar polygons or manually classified re-entrant polygons are stored for each point in the special GDB_Edits directory of geodatabases. These data are not overwritten by GLiMR, but are copied into the current database when processing each point.

Any feature class in the GDB_Edits directory starting with 'Scar' (e.g. 'Scar_by_DWB') is considered a manual or edited flake scar polygon feature class and is copied from the persistent geodatabase into the geodatabase being processed. A corresponding aggregate feature class starting with 'Flake' (e.g. Flakes_Scar_by_DWB') will be created for the aggregate flake statistics.

Aggregating flake scar statistics

Aggregate statistics are computed within each flake scar polygon of the input 'Scar' feature class and added as an attribute to the analogous output 'Flake' feature class.

Each flake scar polygon is superimposed on the TIN triangles. Remember the TIN polygon meshes carries slope, aspect, 2D area, 3D area and elevation (thickness) data for each small triangle between scan points. If a scar polygon boundary splits a TIN triangle, the TIN triangle is split in two and only the portion inside the scar is used with appropriately revised attributes. This yields a set of TIN triangles with attributes corresponding exactly to the scar outline.

Within each scar, aggregate statistics are computed and added to the Flakes attribute table. The 2D area of the TIN triangles is not constant, so all values are area-weighted as appropriate. The result is a new feature class (Flakes_<name>) with all of the attributes previously described.

Summary database tables

GLiMR maintains a separate geodatabase for each projectile point. Each one of these will have feature classes of Outline, Convex Hull, Reentrants, Flakes, etc. Each of these tables also carries the GLiMR key field (GKey) that uniquely identifies it to a point, face and scan version.

To facilitate comparison between points, GLiMR builds a summary file geodatabase of many of these feature classes. It also creates a clone of this database as a MS Access database (a_GLiMR_Summary.mdb). This is accessible using Access 2010 or any other ODBC compliant database. GLiMR supplies an Access link database (GLiMR_Access_Link.accdb) with annotated table structures and queries to link and view the summary tables and append them to its local persistent data tables.

It is important to note that the summary tables are recomputed and replaced by GLiMR and so are **not** persistent. The Access link database (GLiMR_Access_Link.accdb) **is** persistent and may be used to warehouse results.

Queries within the Access link database provide an easy method to export filtered and formatted data to other software in a variety of formats.

At this point in time, other context and analytical data are stored in a separate database related by the catalog number embedded in GKey which carries (Catalog_Face_Version).

Methodology narrative

Constructing the geometry for the entire point

Data acquisition

To date, all scanned point data have been acquired using a NextEngine 3D Scanner HD. Point cloud acquisition, quality control, creation of seamless 3D meshes and partitioning them into subsets representing faces of one side of an artifact is covered in a separate document.

Surface representation

Our arbitrary convention for coordinate system orientation is that the long axis of the projectile point is aligned with the Y axis and the blade tip is toward +Y. The width of the artifact is aligned with the X axis. The thickness is aligned with the Z axis, +Z being thicker. We are working with the scan of one side of the artifact, so we are not dealing with true thickness, rather a distance above an arbitrary XY mid-plane through the artifact, informally referred to as half-thickness.

After initial quality control, the scanned point vertices are translated in XY such that the origin lies at the centroid of the point outline. The vertices are translated vertically such that the minimum z value equals zero and a plane fit to all points of the face is parallel to the XY plane..

GIS analysis software can create surface models from x,y,z points either as a Triangulated Irregular Network (TIN) or a gridded (raster) representation. We used both. A TIN interconnects each triplet of adjacent 3D points to form a set of 3D triangular faces.

Each face has properties of 2D area, 3D area, aspect and slope. Aspect refers to the bearing of the ‘downhill’ direction. Each face also has a min, max and mean elevation. The entire collection of 3D triangular faces is referred to as the TIN. All subsequent data are derived from these simple properties.

A TIN has the advantage of being a linear interpolation of all the data points and can follow irregular perimeters at the resolution of the input scanned data. A gridded surface resamples the data uniformly and is an approximation of the surface with different outcomes based on the method and grid cell size used.

We use TINs for surface properties such as slope, aspect and thickness. We used gridded surfaces for overview topography, surface extrapolations, smoothed surfaces, trend surface and residual surfaces for flake boundary determination. Flake polygons themselves were created using grids, but all geometry analyses within each flake were computed from the TINs to avoid introduced computational artifacts from the gridding algorithms.

We created both TIN and grid representations of artifact topography clipped to the perimeter outline of the artifact.

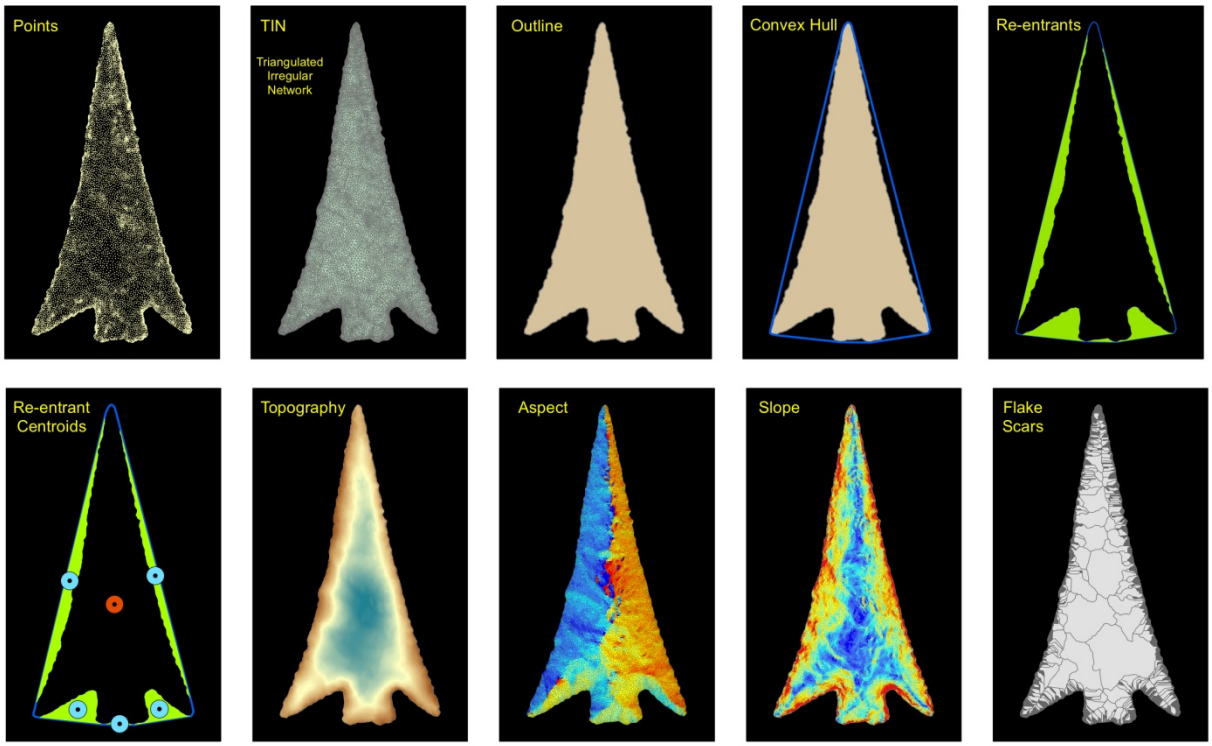


Figure 9 Properties of entire point

Entire artifact versus flake scar analysis

Computing and displaying representations of the entire artifact colored by a property are useful for visual study and shape analysis. A much more powerful data set can be derived by segmenting the artifact into flake scar polygons and aggregating geometric statistics with each flake scar. We will look next at entire artifact metrics compared to flake-based metrics.

Although this discussion will focus on metrics resulting for aggregating data based on flake polygons, other polygons may be used for aggregation. These may represent various domains or subsets of the artifact surface. An example set of domains might be based on flakes grouped by aspect, centroid position and bearing to closest edge to separate left, right and haft areas for separate analysis.

Triangular Irregular Network Surfaces

Most of the geometry metrics are derived from TIN surfaces, so a brief review of TINs follows.

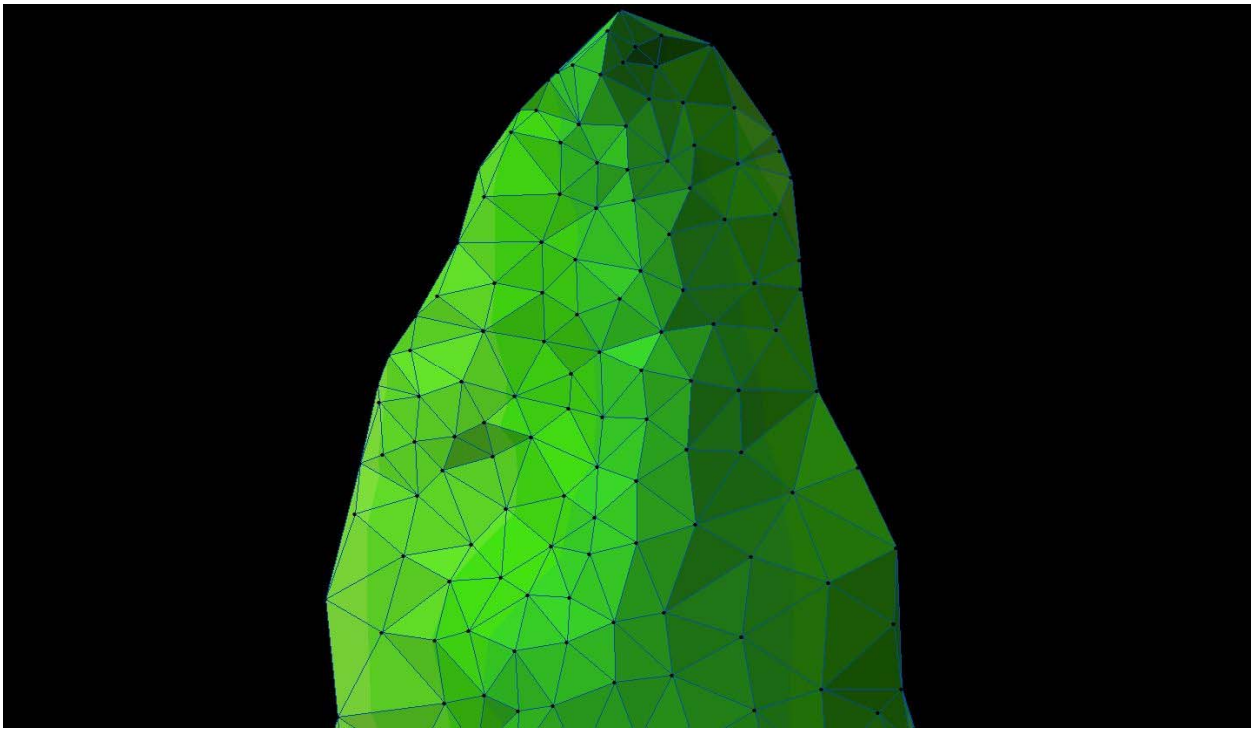


Figure 10 Detail of shaded TIN of point of artifact

A TIN is the network of 3D triangular faces, each defined by three vertices of scanned x, y, z points. All of the analyses herein use Delaunay triangulation. Each triangle of the TIN has the geometric properties shown in Table 1.

TIN_example

Field Name	Data Type	Description
OBJECTID	AutoNumber	
Slope_Deg	Number	Triangle slope in degrees
Aspect	Number	Triangle aspect direction (bearing) in degrees
Shape_Length	Number	Triangle perimeter length in mm
Shape_Area	Number	Triangle area in sq mm
Z_Min	Number	Minimum elevation of the triangle vertices
Z_Max	Number	Maximum elevation of the triangle vertices
Z_Mean	Number	Mean elevation of the triangle vertices

Figure 11 Geometric properties of a 3D triangle face

When a TIN is created it triangulates all points with the convex hull of the point set. A convex hull is the smallest convex polygon containing all of the points (in XY). Think of it as shrink-wrapping your points in the XY plane. The convex hull polygon itself turns out to be quite useful.

When a TIN triangulates all points within the convex hull, it includes unwanted triangles that overlie re-entrant areas on the artifact perimeter shown in Figure 11. These long, skinny triangles are subsequently removed by ‘delineating’ the TIN which entails removing all triangle edges greater than a give length. Specifying a length slightly greater than the maximum inter-

point distance of the scan removes triangles in reentrant areas and leaves triangles overlying the artifact intact. The perimeter of the remaining triangles yields the polygon representing the outline of the artifact.

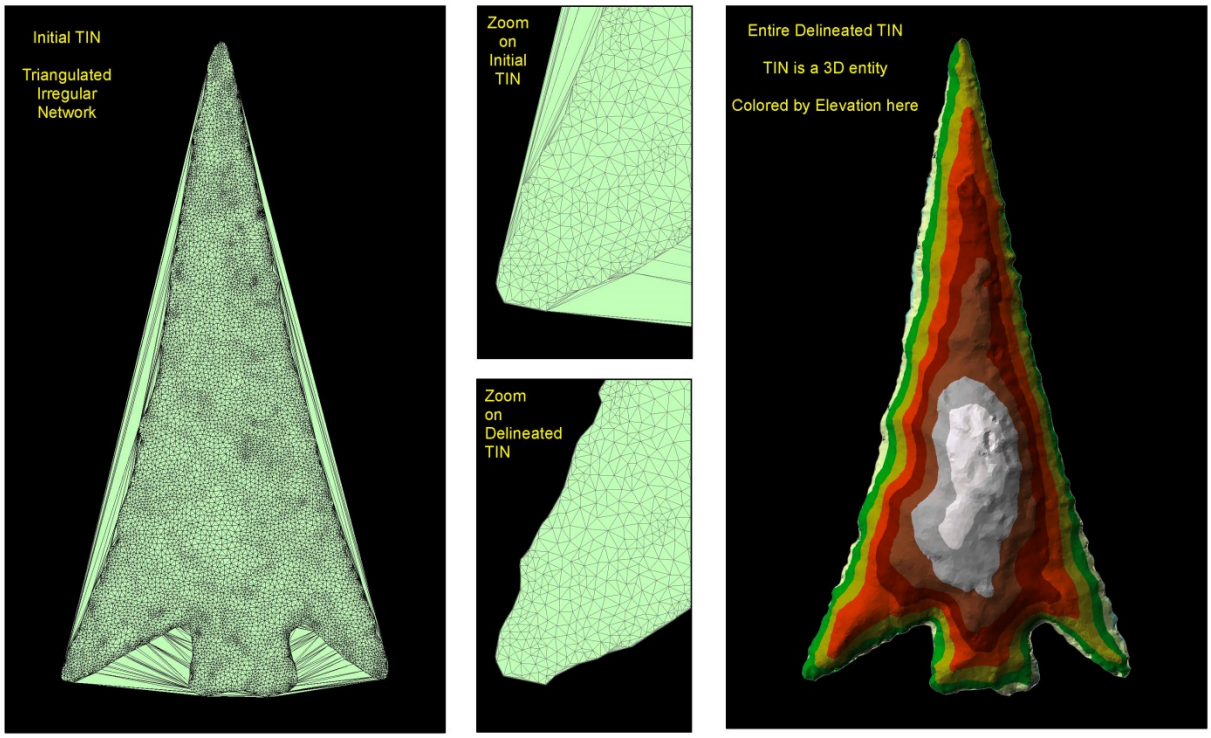


Figure 11 Creating and Delineating a TIN

The TIN within the outline of the artifact is a starting point for deriving other geometric values.

The Outline polygon yields basic height, width and area values. The associated Convex Hull polygon is the generalized shape. The symmetrical difference between the Outline and the Convex Hull is the starting point for re-entrant analysis.

Types of Metrics

Three primary groups of properties are examined:

- *Entire artifact 3D surface properties*
- *Entire artifact outline properties*
- *Flake-scar 3D properties*

Properties of 3D Surface of Entire Artifact

The entire artifact surface is described by the TIN, which is in turn made up of triangular faces connecting the scanned 3D vertices. Any of the properties may be analyzed for all faces in the artifact. An example histogram of aspect is shown in Fig 12

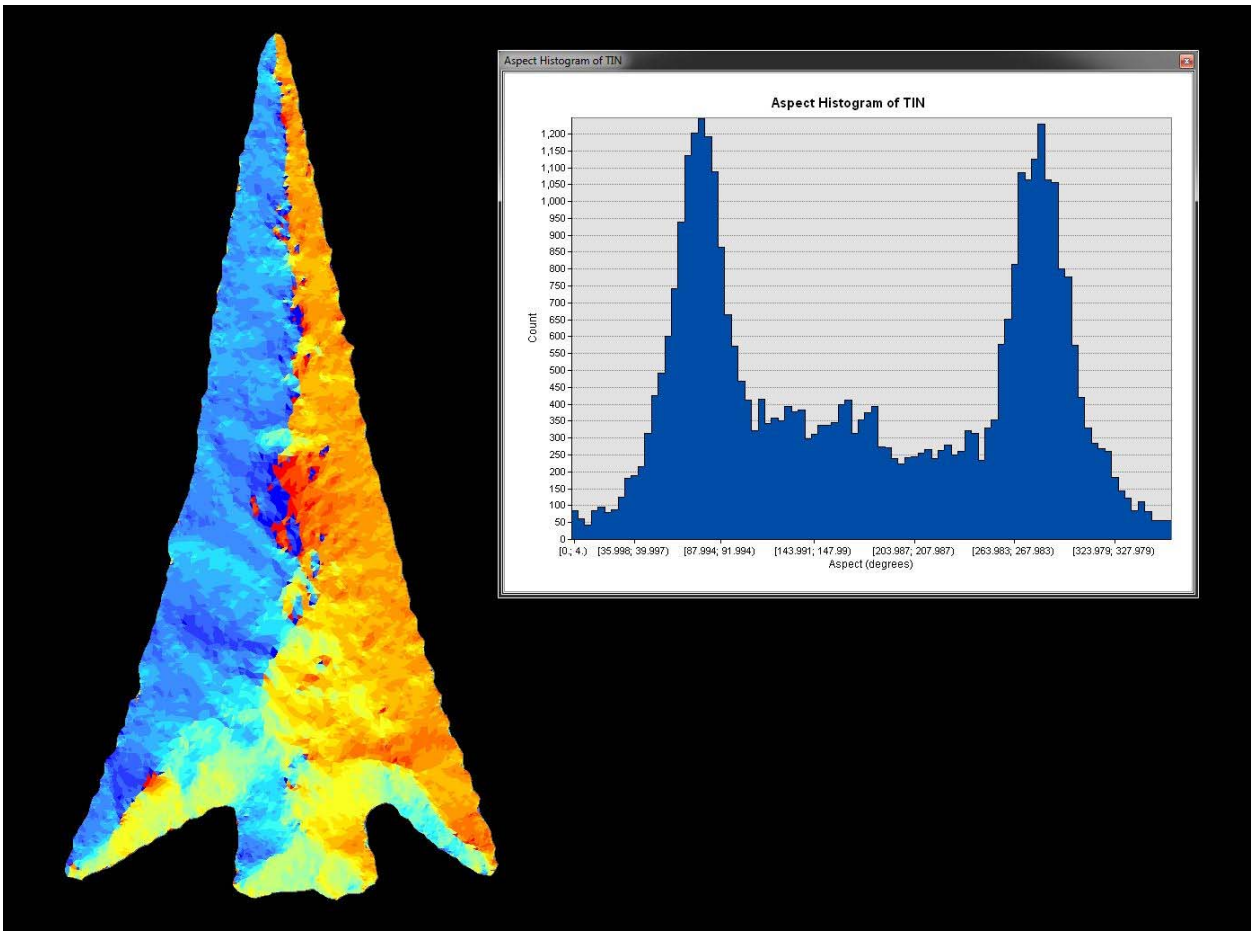


Figure 12 Aspect histogram of TIN of entire artifact

Categorization of triangle geometric properties is very useful visually, and very basic groupings can be devised based on the histograms. Filtered subsets may also be analyzed and compared for symmetry analysis.

The TIN is a complete representation of the 3D surface at the resolution of the scan. It provides a continuous topographic surface that can be sampled at any location. As a result, any arbitrary profile may be extracted and plotted in 2D as shown in Figure 13.

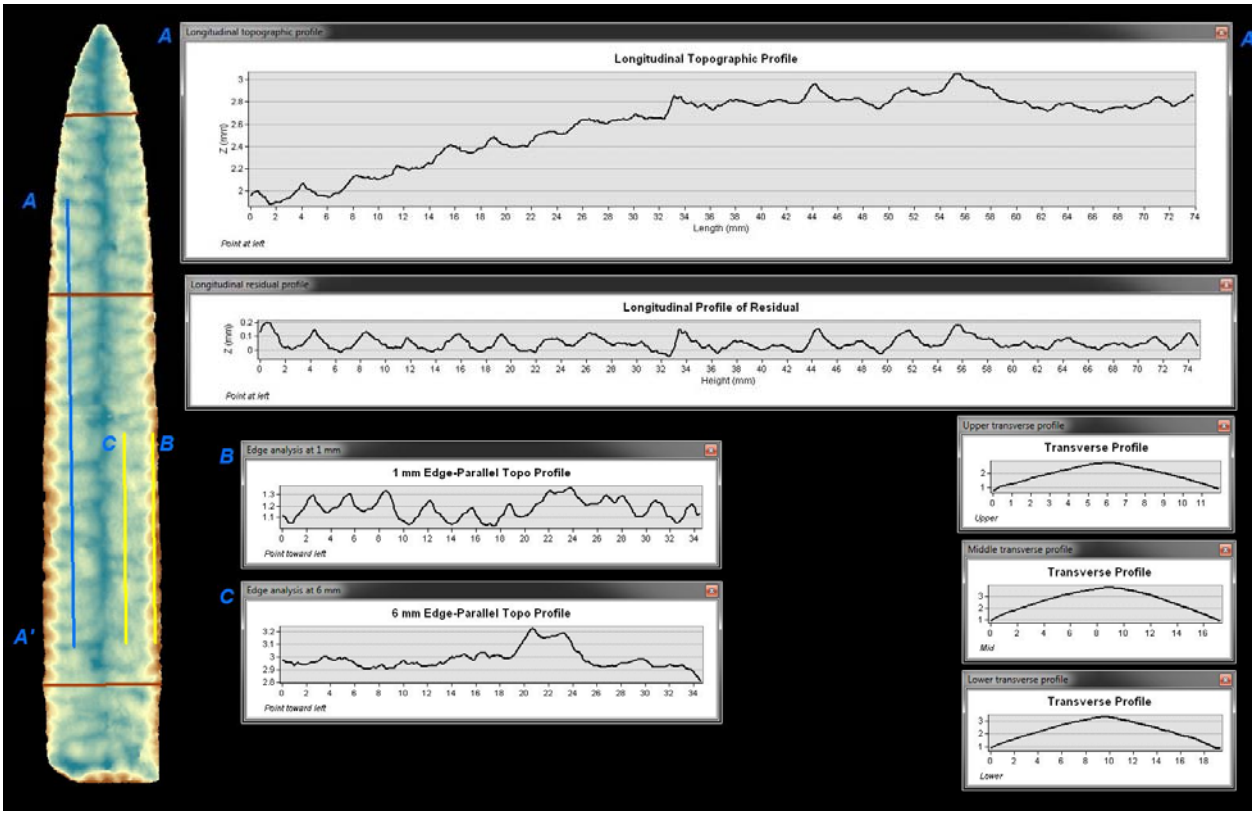


Figure 13 Arbitrary Profiles

Edge-parallel profiles may be extracted at different distance from the artifact edge to examine flake frequency and dimensions at any position. In areas of thickness (Z) variation, the residual surface may be more useful.

Properties of Artifact Outline

Three groups of properties of the outline will be examined. See Figure 14

- *Outline polygon*
- *Convex Hull*
- *Re-entrants*

The outline polygon is created from the delineated TIN perimeter. The centroid of this polygon is used as the origin for the XY coordinate system. This has the benefit of indicating what quadrant of the artifact a face lies simply by the sign of X and Y. The outline polygon has

further properties of area and perimeter length. Crenulated edges are reflected in longer outline perimeter lengths.

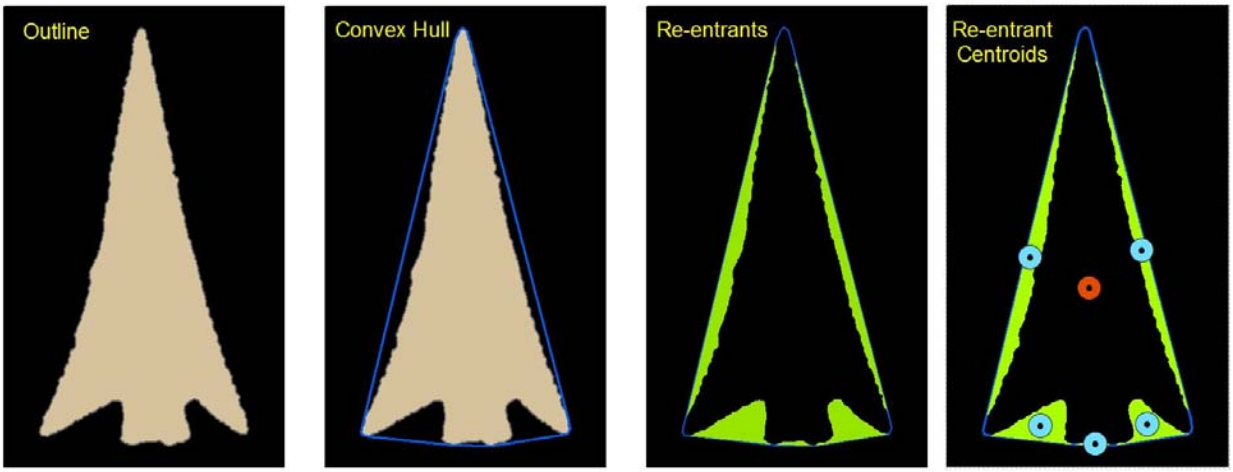


Figure 14 Artifact Outline Properties

Comparing the convex hull polygon and the outline polygon is very revealing regarding any indentation or notches around the perimeter of the artifact. We create a re-entrants polygon set representing areas of the convex hull not covered by the artifact outline. This represents the re-entrants in the sides of the artifact, but more importantly their number, sizes (area), shapes, orientations and positions (centroid XY). These can form a simple diagnostic for gross classification.

Outline properties are listed in Figure 4. In addition to these, reentrant feature classes also carry ellipse parameters approximating each re-entrant's orientation.

Various measures of edge crenulation may be computed by comparing the perimeter length to the area of the outline polygon or by examining the ratio of areas of the artifact outline versus the convex hull.

Determining Flake Scars Outlines

Conceptually speaking, flaked lithic artifacts bear many morphological similarities to island landscapes. When observed from an aerial view, a biface possesses features that are analogous to an island's geomorphic elements: flake scars are hydrological basins, the flake scar boundaries are the drainage divides; flake scars that are truncated by later flake scars produce closed hydrological basins; the bifacial margin that circumscribes the objective piece is akin to an island's shoreline, albeit not at a constant elevation.

The landscape topography analogy is a good one and is the starting point for using GIS tools to delineate flake scar polygons. In analyzing the artifact surface for flake boundaries, local topography is more important than absolute topography. A flake may be one of many on a sloped surface of the artifact. Removing the general slope prior to surface analysis enhances local topography of a given flake.

This is accomplished by residual surface analysis. A heavily smoothed surface is first created to reflect the general surface shape. This is done using focal statistics and a moving average filter with a 7mm diameter circle to produce a smoothed topographic surface. The smoothed surface represents the local average topography elevation.

The smoothed surface is then subtracted from the actual topography to yield a residual surface. The residual surface represents the relative elevation with the general slope effects removed. The residual provides a good surface to find the 'drainage divides' which will become our flake scar polygon boundaries.

ArcGIS geoprocessing routines were used to import the points, create the surface TIN and topographic, smoothed and residual surface grids. Specific workflow details are in the ArcGIS geoprocessing section later in this document. Watershed basin analysis geoprocessing routines are used with residual topography and derived flow directions to compute basin boundaries. These basin boundary polygons are used as flake scar boundaries. They represent local topographic ridges that form drainage divides, which is essentially what a flake scar boundary behaves like.

Care must be taken to insure that high relief ripples from conchoidal fracture (Wallner Lines) are not interpreted as flake scar boundaries. This will result in a set of nested "C"-shaped flake scars that are actually a single scar. This has only been a problem on very large flakes on artifacts made of glassy materials.

These routines may be adjusted to produce more fine scale flakes scars or fewer slightly generalized flake scars. The problem of 'one versus two' flakes as local relief gets low is a problem for both programmatic and manual flake delineation. The resulting flake scar boundary polygons may be overlaid on the artifact photo, colored TIN or grid surface representations to validate them visually.

The analyses in the rest of this section will look at the aggregate properties within each flake. The flake scar polygons are used to aggregate geometric properties of the TIN triangle beneath them. A manually digitized set of flake scar polygons may be used in lieu of the automatically determined ones.

Properties of individual flakes

Each flake polygon overlies many triangular faces of the TIN representing surface topography. The aggregated geometric properties of all of the triangles within each flake may be the sum, minimum, maximum, or mean of specific attribute or flake centroid location within the artifact. Figure 15 below lists flake-based aggregate metrics.

Flakes

Field Name	Data Type	Description
GKey	Text	GLiMR Key
OBJECTID	AutoNumber	Object ID in ArcGIS feature class
Centroid_X	Number	X centroid of flake
Centroid_Y	Number	Y centroid of flake
NEAR_DIST	Number	Distance from flake centroid to nearest edge
NEAR_X	Number	X coordinate of point on nearest edge
NEAR_Y	Number	Y coordinate of point on nearest edge
NEAR_ANGLE	Number	Bearing from flake centroid to nearest edge (0 = +Y; increasing cw)
FREQUENCY	Number	Number of TIN triangles in this flake
MIN_Z_Min	Number	Minimum Z value (lowest point on this flake)
MAX_Z_Max	Number	Maximum Z value (highest point on this flake)
Surface_Area	Number	Sum of the 3D surface area of all the TIN triangles making up the flake
MEAN_Z	Number	Mean Z value (average elevation of this flake)
MEAN_Aspect	Number	Mean aspect of flake (bearing in degrees of down slope direction) (0 = +
STD_Aspect	Number	Standard deviation of aspect for this flake
MEAN_Slope_Deg	Number	Mean slope of flake (in degrees)(0 = horizontal)
STD_Slope_Deg	Number	Standard deviation of slope for this flake
MAJORAXIS	Number	Ellipse fit to flake: major axis length (longest axis)
MINORAXIS	Number	Ellipse fit to flake: minor axis length (shortest axis)
ORIENTATION	Number	Ellipse fit to flake: orientation of major axis (longest axis) relative to X a
EllipseRatio	Number	Major Axis / (Major Axis + Minor Axis)
RoughnessByArea	Number	3D surface area / 2D area
FlakeSize	Number	Diameter of largest inscribed circle that fits in flake
Flake_Perimeter_Length	Number	Length of the perimeter of the flake
Flake_Area	Number	Area of the flake
Version	Text	GLiMR software version number

Figure 15 Flake properties

Each property will be described below. In practical use, some of these will be used as filters to isolate a specific part of the artifact prior to analyzing that subset of flake scars.

Flake Area represents the sum of the 2D areas of all underlying TIN triangles. This represents the plan view area in the XY plane. The geodatabase fieldname is Shape_Area.

Surface Area represents *t*he sum of the 3D areas of all underlying TIN triangles. A sloping triangle has a 3D surface area that is greater than the flat 2D triangle projected onto the XY plane. The fieldname in the geodatabase is SUM_SArea.

Aspect is the horizontal bearing (in degrees) of the downhill direction. Zero is in the +Y ('north') direction, typically representing the tip of the blade. Aspect increases clockwise, so 90 degrees is sloping right, 270 degrees is sloping left and 180 degrees is sloping toward the haft. The area-weighted mean and standard deviation of the aspect values of all of the underlying triangles are computed for each flake.

Slope is the rate of change of elevation expressed in degrees. The area-weighted mean and standard deviation of the aspect values of all of the underlying triangles are computed for each flake.

Elevation (Z): For each flake, we compute a Min Z, Max Z and a Mean Z. The Mean Z is the mean of all of the individual TIN triangle mean Z values.

The elevation (Z) of a flake represents the distance above the XY plane (z=0). The zero plane is set at the lowest z value of the scanned 3D point cloud input. This is referred to as half-thickness. The perimeter outline is actually a 3D polyline with the lowest point(s) at z = 0.

True thickness can only be determined with registered scans of both the upper and lower surfaces of an artifact in the same coordinate system. This paper speaks to a single upper surface scan only.

Centroid represents the X, Y position of the centroid of the flake scar boundary. The centroid is used both for flake position within the artifact and calculating its proximity to the closest edge. A consequence of setting the XY origin of the coordinate system at the centroid of the artifact is that the sign of X and Y of the flake centroid indicated what quadrant of the artifact the flake is located.

Flake orientation and shape

The shape and orientation may be described by the elliptical approximation of each flake scar. This is identical to the ellipse data generated for Re-entrants.

Major axis represents the length of the major ellipse axis.

Minor axis represents the length of the minor ellipse axis.

Orientation angle represents the angle between the Y axis and the major axis in degrees. Orientation values range from 0 to 180 degrees. Orientation is not a vector and does not imply one direction, just an axis orientation.

Ellipse ratio represents whether the overall shape of a flake is skinny or fat. It is calculated as the major axis length divided by the sum of the major and minor axis lengths. Its value ranges from 0.5 for nearly circular flakes to 1.0 for very long, skinny flakes.

Flake size represents the diameter of the largest circle that can be fit inside the flake. This metric is derived for the sadly-named 'thickness' parameter for a polygon calculated by ArcGIS Zonal Geometry.

Perimeter represents the length of the perimeter of the flake scar.

Bearing to edge represents the bearing (in degrees) from the flake centroid to the nearest point on the edge of the artifact. Zero is at +Y, typically toward the top of the artifact and increases clockwise. The bearing to edge is very useful for filtering flakes that are related to the same edge.

Distance to edge is the linear distance between the centroid of the flake and the nearest point on the artifact perimeter. This allows filtering of flakes proximal to an edge for separate analysis.

Nearest Edge is represented by two values Near_X and Near_Y which are the coordinates of the point of the closest edge to the flake.

Roughness is a measure of surface roughness within the flake. It is based on the fact that the 3D surface area of all of the TIN triangles underlying the flake increases as there are more ups

and downs within the flake surface. The flake itself is sloped, so final value has to be adjusted for the mean slope of the flake. It is computed as

((sum of the 3D Area) / (sum of 2D Area)) / cos (Mean_Slope).

Exploring the Aggregate data

ArcGIS provides many tools to filter, symbolize and graph the GLiMR output feature classes. The interactive relation between graphs and feature classes is powerful.

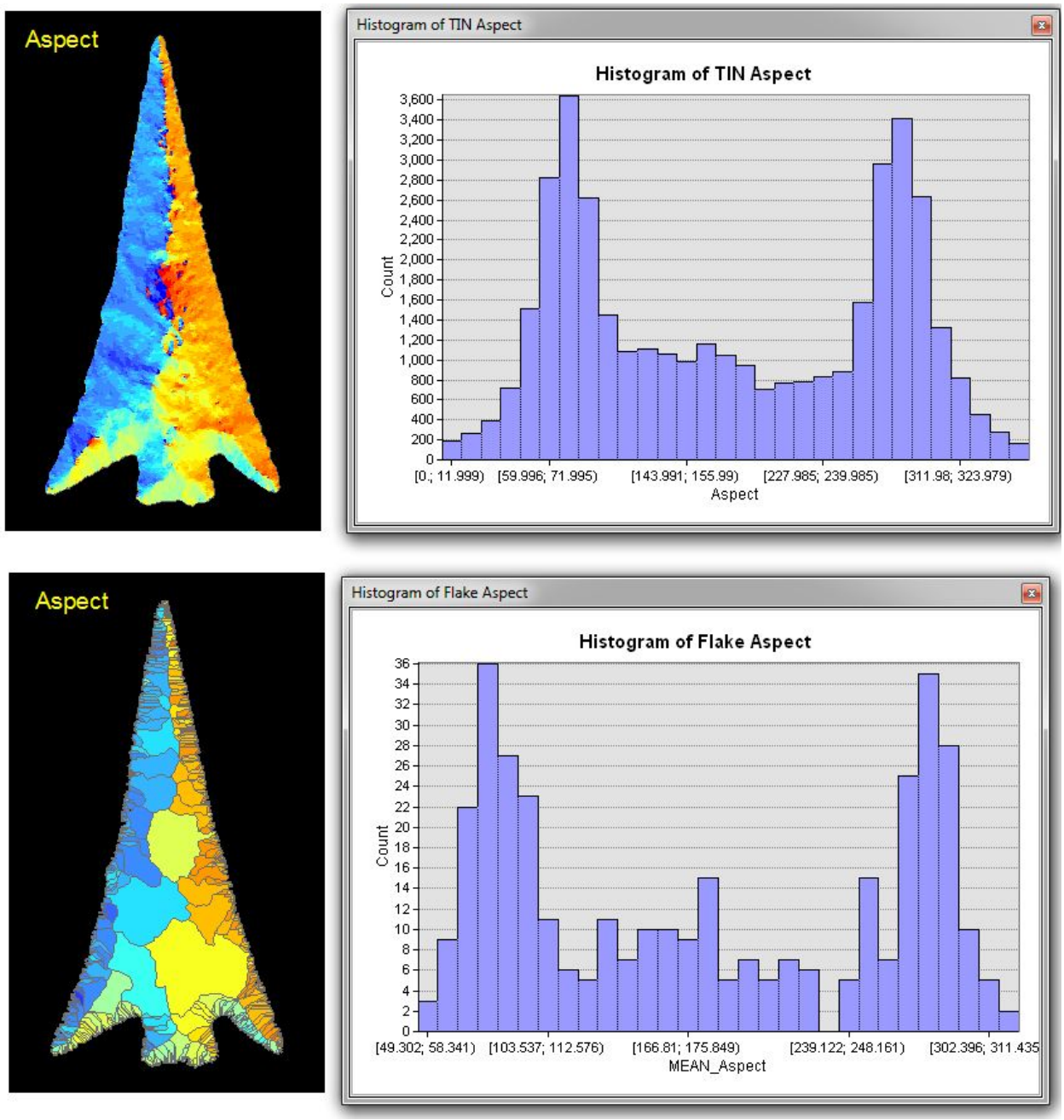


Figure 16 TIN versus Flake Histograms of Aspect

Aggregating geometric properties of the TIN triangles into flake properties allows more detailed analysis than just examining the TIN. The overall histograms for the entire point of the TIN versus flake properties appear similar.

The simplest way to examine flake geometry properties is to display the flake polygons colored by a given property.

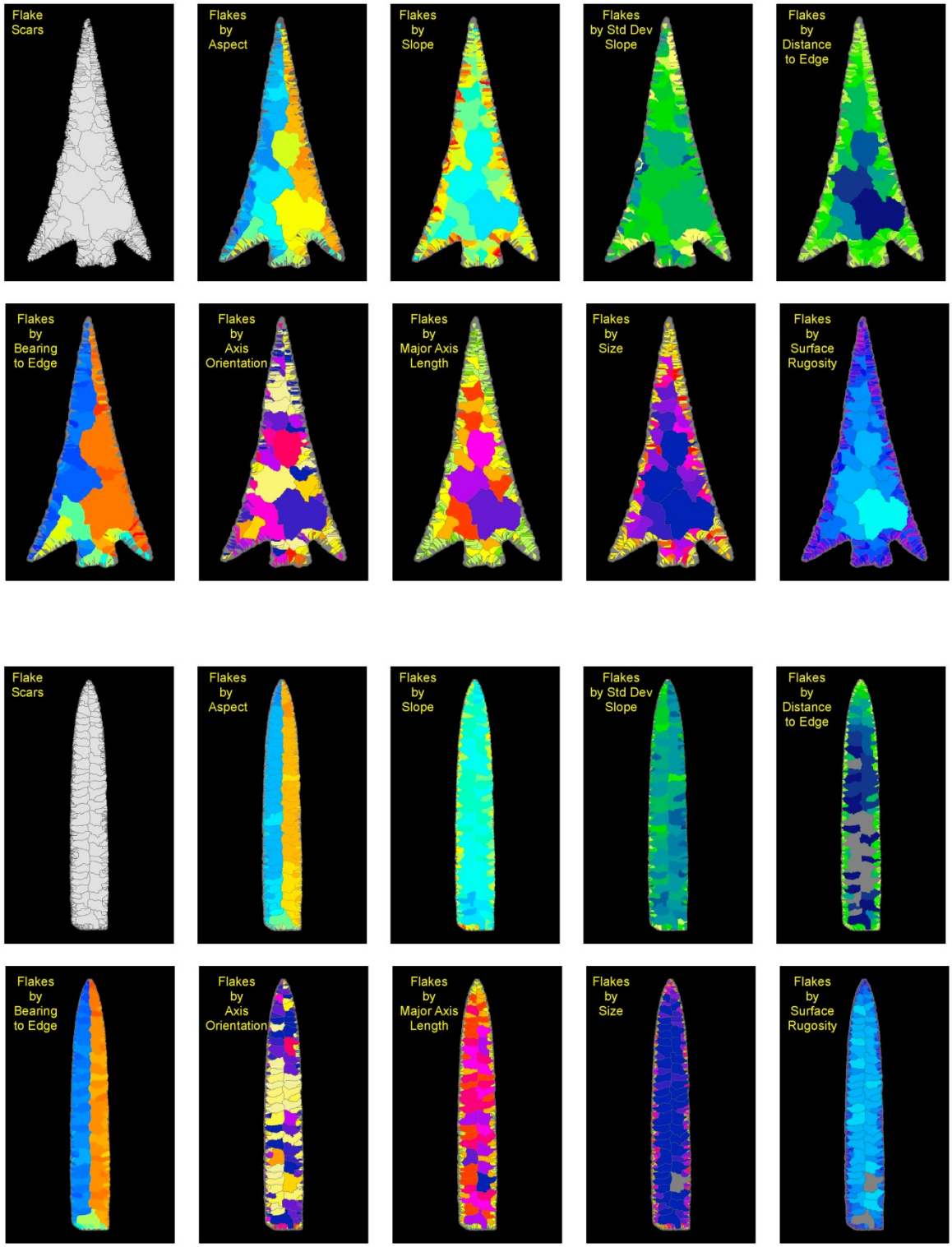


Figure 17 Examples of flakes colored by geometric properties

Initially, such displays may not be terrible illuminating. Histograms and multi-property scatter graphs combined with the selection capabilities of GIS provide more insight.

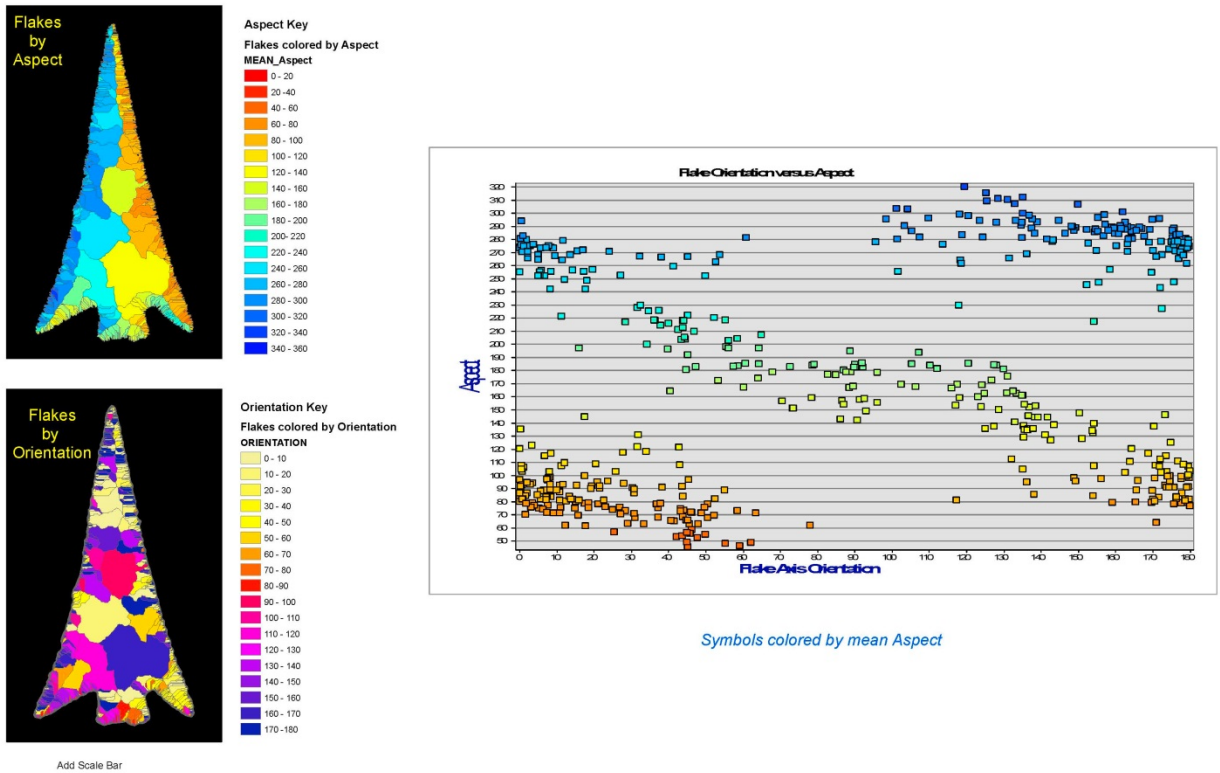


Figure 18 Flake Aspect compared with Flake Orientation

Figure 18 shows Flakes colored by Aspect and by major flake axis Orientation. The scatter plot demonstrates clustering which may be used to group similar subsets of faces.

Each flake polygon is represented by a row in the associated attribute table shown in Figure 19 Each column of the table is a geometric attribute value. One or more flakes may be selected graphically on the artifact display, on the graph display or in the attribute table.

Shape *	Id	POINT_X	POINT_Y	NEAR_DIST	NEAR_X	NEAR_Y	NEAR_ANGLE	MEAN_Slope_Deg	STD_Slope_Deg	MEAN_Aspect	STD_Aspect	MIN_Z_Min	MAX_Z_Max	MEAN_Z_Mean	SUM_SArea
Polygon	216	2.718755	-7.494323	5.47754	7.978088	-5.963684	73.773269	13.780987	8.127085	120.509171	52.698812	0.375	3.194	2.282606	69.089557
Polygon	171	-2.131588	-1.909592	4.509518	-6.431778	-0.551608	287.525923	17.314061	9.382136	257.061624	65.94945	0.3698	3.3148	2.57337	43.244957
Polygon	142	0.554806	4.672652	4.284923	4.701477	5.752319	75.405947	15.674157	7.686478	149.13326	126.062121	0.7209	3.3148	2.474813	35.056562
Polygon	222	-2.943668	-8.634438	3.861073	-3.250793	-12.48327	184.562362	20.948281	10.813083	229.924512	37.259488	0.6648	2.9898	2.224501	32.881151
Polygon	103	-1.33012	9.801633	2.857561	-4.182129	9.979675	273.572173	20.453492	10.474645	271.889523	52.794445	0.2402	2.4527	1.794384	15.437533
Polygon	170	4.093176	-2.537803	2.669127	6.704285	-1.984314	78.031903	23.303019	8.206933	89.070963	17.899236	0.7015	3.1719	2.229879	15.499689
Polygon	734	-0.950569	14.955323	2.625218	-3.458984	15.729675	287.15556	17.510392	9.145693	252.642167	77.729823	0.2837	2.2668	1.683088	12.100485

Figure 19 Portion of an attribute table for flake properties

Selections may also be made using SQL queries based on any combination of properties. A typical sequence may filter the flakes to a subset then determine properties for the group.

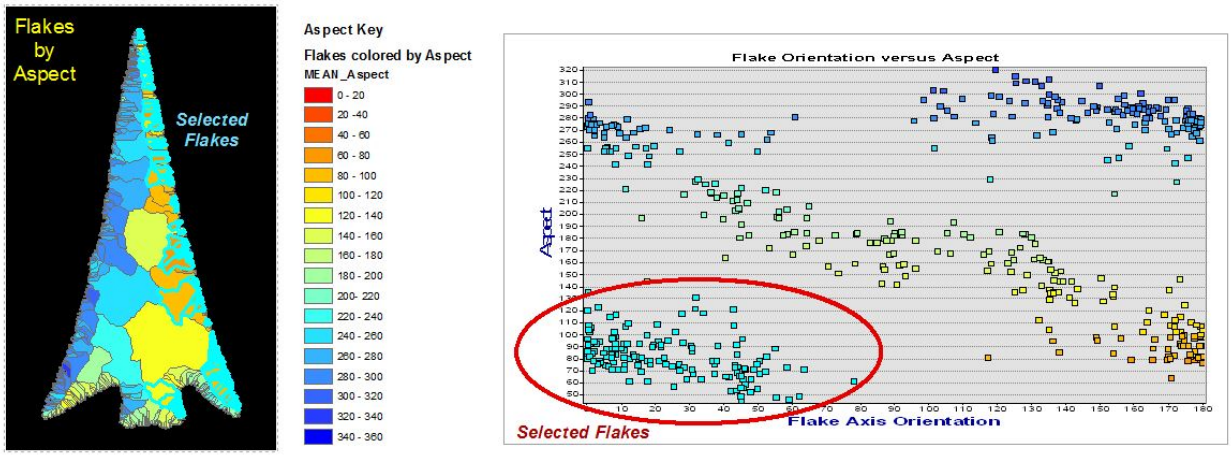


Figure 20 Interactive selection (note cyan highlight on both Flakes and Graph)

GLiMR is new and we do not know how it will be used. GLiMR supplies a starting point for exploring the details hidden in the geometry of an artifact. It supplies data for external analysis including multivariate statistical approaches and shape analysis.

Landmark Points

Many manual schemes have been devised for placing landmark points on projectile artifacts for comparison between different artifacts. GLiMR create a set of 3D landmarks located around the periphery of artifacts and at internal points within the artifacts.

GLiMR generates these user-configurable sets of landmarks for use in Procrustes analysis or similar shape analysis. The goal is to generate a group of points with (X,Y,Z) values and a landmark ID such that the landmark IDs are in a corresponding topological positions on different artifacts.

We chose a radial sampling consisting of concentric rings of points between the artifact centroid and the edge. This was chosen over methods that subdivide the perimeter equally because it is not as seriously affected by the position and depth of any notches.

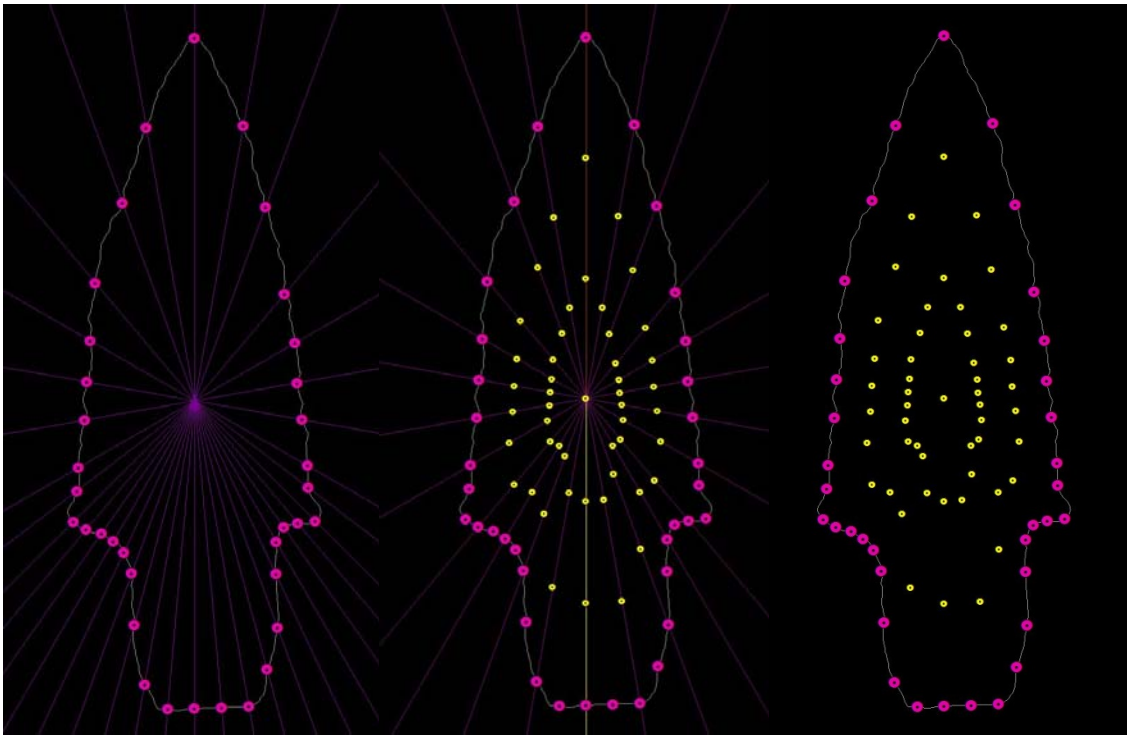


Figure 21 Procrustes L) Radials and edge points, C) Radials and internal ring points, R) All points

Conceptually, GLiMR projects radial rays from the centroid to the first intersection with the outline (edge) of the projectile point and creates a landmark point at that edge location. Then one or more points are interpolated between the centroid and the edge on the radial. Each point solves for the z value of the topography and smoothed topography of the projectile point surface and stores it in the attribute table.

Any number of rings of internal points may be specified. Two rings are shown in the example above. The radial set of lines for may be different for the edge and internal points. The number of radials may be varied and positioned to increase resolution at specific sectors as needed.
One additional Procrustes point is added at the extreme blade tip, specifically at the vertex with the maximum Y value. This important landmark is not necessarily a member of the radial edge points due to slight alignment discrepancies or distorted artifacts.

The mechanics of Procrustes point generation depends on two user-editable radial 'Starburst' feature classes shown in purple on Figure 21. They are named 'Starburst_Edge' and 'Starburst_Rings'. First the Starburst_Edge polylines are intersected with the Outline to produce the edge points (shown in purple). Next the Starburst_Rings polylines are a temporary set of edge points. New internal ring point(s) (shown in yellow) are then interpolated between the centroid to the temporary edge points. If one ring is specified, one point is placed halfway between the centroid and edge; if two rings are specified, two points are placed at 1/3 and 2/3 of the distance.

Two different starburst patterns are used to allow for denser sampling of the edge and sparser sampling of the interior. In the example in Figure 21, the Starburst_Edge (left panel) has more radials in the haft section than the blade section to increase sampling of the more complex haft area. The Starburst_Rings (center panel) has fewer radials in the haft area to prevent oversampling of the haft interior. These are at the discretion of the user.

There is one more adjustment to fine-tune interior ring point placement: An 'outward_shift_percent' variable, allows all ring points to be to be shifted outward by a given percent of the distance to the edge. For example, setting this to 20 would shift each internal ring point 20% of the way toward the edge.

Figure 22 below illustrates the resulting landmark points from points with different shapes.

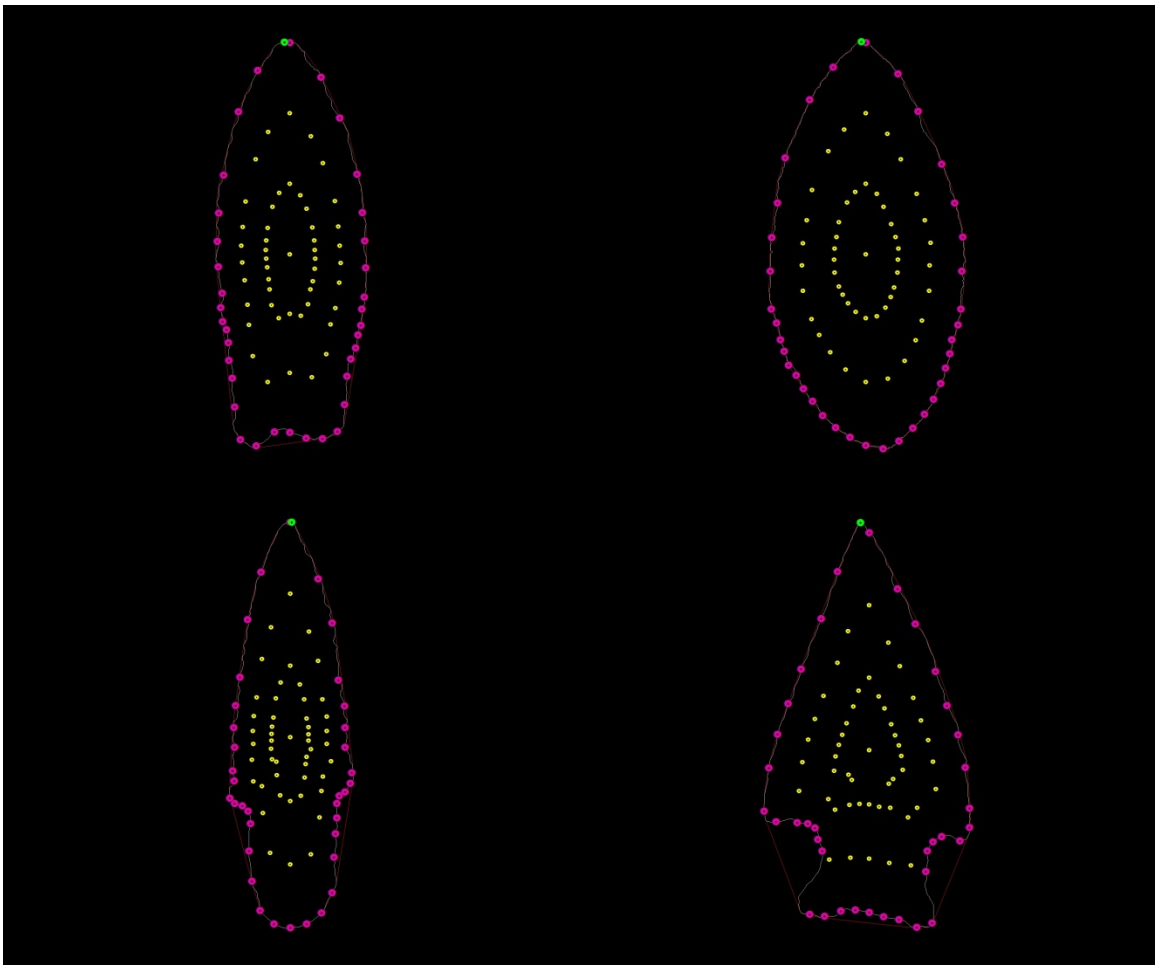


Figure 22 Procrustes landmarks on different points.

Each point has the same number of landmarks generated for it and summarized in the Procrustes Summary table. The actual number of landmarks per point varies dependent upon the starbursts used and the number of rings specified. Each landmark point has attributes including a point number, an X and Y, and the z values of both the topographic and smoothed topographic surface. The point number represents the same analogous position between artifacts. The point number is unique within each artifact and repeated for every different artifact. For example, point 7 exists only once in each artifact, but is present in every artifact at an analogous position.

Figure 23 below shows a portion of the Procrustes Summary table. An additional field called 'Label' is supplied that you may optionally populate with terms used for classification that you would like transferred to the output analysis ASCII text file.

ProcrustesSummary

GKey	ProcrustesPtNum	Type	Surface	Label	X	Y	Z_Topo	Z_Smooth
294_1_1	20017	Internal	Topo	foliate	-4.950	-5.900	4.754	4.641
294_1_1	20018	Internal	Topo	foliate	-5.864	-3.386	4.635	4.513
294_1_1	20019	Internal	Topo	foliate	-6.524	-1.151	4.290	4.198
294_1_1	20020	Internal	Topo	foliate	-6.330	1.116	4.382	4.006
294_1_1	20021	Internal	Topo	foliate	-5.765	3.328	4.463	3.966
294_1_1	20022	Internal	Topo	foliate	-4.869	5.802	4.341	4.018
294_1_1	20023	Internal	Topo	foliate	-3.340	9.177	4.570	4.298
294_1_1	20024	Internal	Topo	foliate	-2.134	12.102	4.590	4.238
295_1_1	-1	Tip	Topo	stemmed	2.110	31.395	1.933	1.655
295_1_1	0	Centroid	Topo	stemmed	0.000	0.000	3.151	3.153
295_1_1	1	Edge	Topo	stemmed	0.000	31.003	1.640	1.525
295_1_1	2	Edge	Topo	stemmed	4.770	27.051	1.902	1.817
295_1_1	3	Edge	Topo	stemmed	7.040	19.343	1.642	1.784
295_1_1	4	Edge	Topo	stemmed	8.844	10.539	0.717	1.177
295_1_1	5	Edge	Topo	stemmed	9.191	5.306	0.265	1.063
295_1_1	6	Edge	Topo	stemmed	10.190	1.796	0.635	0.860

Figure 23 A portion of the Procrustes Summary landmark table.

At this point in time a single stand-alone python program is supplied to write a formatted output of this table to a text file for input to MorphoJ software.

C. P. Klingenberg. 2011. MorphoJ: an integrated software package for geometric morphometrics. Molecular Ecology Resources 11: 353-357. doi: 10.1111/j.1755-0998.2010.02924.x

ArcGIS geoprocessing steps

An abridged sequence of geoprocessing commands is given below for those who want to delve deeper. The actual work is done in Python 2.7 calling ArcGIS 10.2 geoprocessing modules. To fully understand this you will have to examine the actual python code.

Throughout this section 'GDB' = geodatabase. Housekeeping file movement and internal bookkeeping is omitted. Some processing and flow control are summarized in blue text.

A GLiMR.env.txt environment file is read initially. It specifies the location for input and output.

GLiMR.env.txt environment file contents:

C:\GLiMR\GLiMR_XYZ
C:\GLiMR\GLiMR_GDB
C:\GLiMR\GLiMR_MXD
C:\GLiMR\GLiMR_Images
C:\GLiMR\GLiMR_Templates
C:\GLiMR\GLiMR_GDB_Edits
C:\GLiMR\GLiMR_Output

This is the GLiMR environment file.
The first seven lines contain the following directories
One directory path per line in the following order:

XYZ is the source for .xyz ASCII text scan files to be read
GDB is the root directory for all geodatabases to be created
MXD is the root directory for all .mxd ArcMap files to be created
Images is the root directory for all images to be created
Templates is the directory for all map template files. Templates staring with 'Template_' will create a jpg snapshot
GDB_Edits is the root directory for all persistent geodatabases. These typically hold edited data such as manual scars or re-entrant types
Output is the place files are written such as formatted Procrustes files or file audits

The GDB_Edits directory contains persistent data such as manually digitized scar feature classes and Re_entrants that may have had edits to Left, Right, Base RE_Type attributes.

The standard location for this file is C:\GLiMR

For every XYZ file in the GLiMR_XYZ directory:
Parse the file name to extract catalog number, face and version if supplied.

Geodatabase:
Create GKey . GKey = G123_2_1 for Catalog 123, Face 2, Version 1.
Delete preexisting GDB
arcpy.CreateFileGDB_management(gdb_dir, gdb_name)

Points:
arcpy.ASCII3DToFeatureClass_3d(xyz_file, 'XYZ', 'OrigPts', 'POINT', '1', '', '', '', 'DECIMAL_POINT')
Create initial outline and find centroid:
arcpy.CreateTin_3d(outTIN, '', in_point_fc + ' Shape.Z Mass_Points <None>', 'DELAUNAY')
arcpy.DelineateTinDataArea_3d(outTIN, max_length, 'PERIMETER_ONLY') (default max_length = 0.5)
arcpy.TinDomain_3d(outTIN, out_outline_poly, 'POLYGON')
< Python code to insure delineation did not fail >
Find centroid and extents:
arcpy.FeatureToPoint_management(outline_poly, out_centroid, 'CENTROID')
arcpy.AddGeometryAttributes_management(outline_poly, properties, length_unit, area_unit,
coordinate_system)
<Translate points in XY such that (0,0) is at the centroid of the outline>
Move points in Z to 'rotate' the points flat. This is a Z translation rather than true rotation.
arcpy.Trend_3d(inPointFC, 'Shape.Z', Trend_XYCentered', '0.05', '1', 'LINEAR', '')
arcpy.SplineWithBarriers_3d(inPointFC, 'Shape', '', '0.05', outRaster, '0')
arcpy.Minus_3d(Topo_XYCentered', Trend_XYCentered', Topo_Normalized')
ExtractValuesToPoints(inPointFC, Trend_XYCentered', 'Trend_Pts', 'INTERPOLATE', 'VALUE_ONLY')
ExtractValuesToPoints(inPointFC, 'Topo_Normalized', 'Pts_Flattened', 'INTERPOLATE', 'VALUE_ONLY')
Translate points in Z such that z Minimum = 0
Create an extended set of points to allow smooth slope extrapolation for zonal statistics
arcpy.ASCII3DToFeatureClass_3d(extFileNameandPath, 'XYZ', 'BasePtsExt', 'POINT',
'1', '', '', '', 'DECIMAL_POINT')
arcpy.SplineWithBarriers_3d('BasePtsExt', 'SHAPE', '#', 0.05, 'TopoExt')

Outline:
Create final Outline and find centroid:
arcpy.CreateTin_3d(outTIN, '', in_point_fc + ' Shape.Z Mass_Points <None>', 'DELAUNAY')
arcpy.DelineateTinDataArea_3d(outTIN, max_length, 'PERIMETER_ONLY') (default max_length = 0.5)
arcpy.TinDomain_3d(outTIN, out_outline_poly, 'POLYGON')
arcpy.FeatureToPoint_management(outline_poly, out_centroid, 'CENTROID')
arcpy.AddGeometryAttributes_management(outline_poly, properties, length_unit, area_unit,
coordinate_system)

```
arcpy.TinTriangle_3d(outTIN, outTIN_Tri, 'DEGREE', '1', '', '')
```

Add Outline attributes and do field name clean-up

```
arcpy.DeleteField_management(outlineFC, ['CENTROID_Z'])
arcpy.AddField_management(outlineFC, 'Height', 'DOUBLE')
arcpy.AddField_management(outlineFC, 'Width', 'DOUBLE')
arcpy.AddField_management(outlineFC, 'HW_ratio', 'DOUBLE')
arcpy.AddField_management(outlineFC, 'Half_Thickness', 'DOUBLE')
arcpy.AddField_management(outlineFC, 'GKey', 'TEXT', '', '', 25)
arcpy.AddField_management(outlineFC, 'Hull_Area', 'DOUBLE')
arcpy.AddField_management(outlineFC, 'Hull_Length', 'DOUBLE')
```

Compute Height, width, H/W ratio using an update cursor

Re-entrants:

```
arcpy.MinimumBoundingGeometry_management(in_outlineFC, 'Hull', 'CONVEX_HULL', 'NONE')
arcpy.SymDiff_analysis(inFeatures, updateFeatures, outFeatureClass, 'ONLY_FID', clusterTolerance)
arcpy.MultipartToSinglepart_management('Hull_Outline', 'in_memory\\Re_entrants')
arcpy.AddGeometryAttributes_management(inFeatures, properties, length_unit, area_unit,
                                       coordinate_system)
```

Ellipse generation:

```
outZonalGeometryAsTable = ZonalGeometryAsTable('Re_entrants', zoneField,
                                       zoneGeometryTable, zoneCellSize)
```

Add attributes:

```
inFeatureClass = 'in_memory\\Re_entrants'
arcpy.AddField_management(inFeatureClass, 'GKey', 'TEXT', '', '', 25)
arcpy.AddField_management(inFeatureClass, 'EllipseRatio', 'DOUBLE')
arcpy.AddField_management(inFeatureClass, 'RE_Size', 'DOUBLE')
arcpy.AddField_management(inFeatureClass, 'RE_Type', 'TEXT', '', '', 15)
arcpy.AddField_management(inFeatureClass, 'Pct_RE_Size', 'DOUBLE')
arcpy.AddField_management(inFeatureClass, 'Pct_Centroid_X', 'DOUBLE')
arcpy.AddField_management(inFeatureClass, 'Pct_Centroid_Y', 'DOUBLE')
arcpy.AddField_management(inFeatureClass, 'Pct_Hull_Area', 'DOUBLE')
```

<Populate these, correcting orientation angle to a bearing and rename some fields.

Work with notch array from re-entrants:
[OID@', 'RE_Type', 'RE_Size', 'Pct_Centroid_X', 'Pct_Centroid_Y', 'Pct_RE_Size', 'CENTROID_X', 'CENTROID_Y'] to determine Re-entrant type (Left, Right, Base) if possible.

Wade through re-entrants to determine quadrant points for each reentrant and outline, plus ellipse points for reentrants. Write point feature classes for each.

Create Procrustes points and haft collarand belt points based on outline and reentrants.

Rasters:

Create Topo grid:

Extract by Mask to clip extended topo to outline

```
TopoFile = ExtractByMask(inTopoExtFC, inOutlineFC)
TopoFile.save('Topo')
arcpy.Contour_3d(Topo + mm, 'Smoothed' + mm + '_Cont', 0.05)
```

Create Smoothed grid:

```
outFocal = FocalStatistics('TopoExt', 'Circle ' + str(radius) + ' CELL', 'MEAN', 'DATA')
outFocal.save('FocalC' + mm)
outExtractByMask = ExtractByMask('FocalC' + mm, 'Outline')
outExtractByMask.save('Smoothed' + mm)
arcpy.Contour_3d('Smoothed' + mm, 'Smoothed' + mm + '_Cont', 0.05)
```

Create Curvature grids:

```
arcpy.Curvature_3d('Topo', 'Curvature', 1.0, 'Curvature_Dip', 'Curvature_Strike')
```

Create Residual grid:

```
arcpy.Minus_3d('Topo', 'Smoothed7mm', 'Residual7mm')
arcpy.Contour_3d('Residual7mm', 'Residual7mm_Cont_05', 0.05)
```

Run Procrustes point generation code

Do housekeeping for persistent data (Reentrants and manual scars)

Flake Scars:

Create auto flake scars:

```
outFill = Fill(inResidualFC, zlimit)            # zlimit is iterated producing different zScarAuto_<zlimit>
outFill.save('tmp_Filled_Residual_' + inResidualFC)
outFlowDir = FlowDirection('tmp_Filled_Residual_' + inResidualFC)
outFlowDir.save('tmp_Flow_Dir_Filled_Residual_' + inResidualFC)
outBasin = Basin('tmp_Flow_Dir_Filled_Residual_' + inResidualFC)
outBasin.save(inResidualFC + '_Filled_Basin_Raster')
outBasin = BoundaryClean(inResidualFC + '_Filled_Basin_Raster', 'DESCEND', 'TWO_WAY')
outBasin.save(inResidualFC + '_Filled_Basin_Raster')
arcpy.RasterToPolygon_conversion(inResidualFC + '_Filled_Basin_Raster', outFlakeBasinFC, 'SIMPLIFY',
                                 'Value')
arcpy.AddField_management(inFeatureClass, 'GKey', 'TEXT', '', '', 25)
arcpy.AddField_management(inFeatureClass, 'Version', 'TEXT', '', '', 10)
```

Populate GKey and version

For every feature class starting with Scar* :

Aggregate flake scars from TIN:

```
arcpy.Identity_analysis (inFeatureClass, idFeatureClass, outFeatureClass, 'ONLY_FID')
arcpy.AddField_management(inFeatureClass, 'Aspect_wx', 'DOUBLE')
```

```
arcpy.AddField_management(inFeatureClass, 'Aspect_wy', 'DOUBLE')
arcpy.AddField_management(inFeatureClass, 'Z_wgt', 'DOUBLE')
arcpy.AddField_management(inFeatureClass, 'Slope_wgt', 'DOUBLE')
arcpy.AddField_management(inFeatureClass, 'SArea', 'DOUBLE')
arcpy.AddField_management(inFeatureClass, 'A_Slope_Sqrd', 'DOUBLE')
```

Compute area-weighted values in vector space to avoid average angle problems

Compute flake centroids:

```
arcpy.FeatureToPoint_management(inFeatureClass, outFeatureClass, 'CENTROID')
arcpy.AddXY_management(inFeatureClass)
```

Compute edge proximity:

```
arcpy.FeatureToLine_management(inFeatureClass, OutlineLine')
arcpy.Near_analysis(inFeatureClass, nearFeatureClass, '', 'LOCATION', 'ANGLE')
```

Change angles to bearings

```
arcpy.Statistics_analysis('TIN_TriID_' + inFlakePolyFC, 'Aggregate',
        [['Slope_wgt', 'SUM'], ['Aspect_wx', 'MEAN'],['Aspect_wx', 'STD'],
        ['Aspect_wy', 'MEAN'],['Aspect_wy', 'STD'], ['Z_Min', 'MIN'],['Z_Max', 'MAX'],
        ['Z_wgt', 'SUM'], ['SArea', 'SUM'], ['Shape_Area', 'SUM'], ['A_Slope_Sqrd', 'SUM']],
        flakeFIDfield)
inFeatureClass = 'Aggregate'
arcpy.AddField_management(inFeatureClass, 'MEAN_Z', 'DOUBLE')
arcpy.AddField_management(inFeatureClass, 'MEAN_Aspect', 'DOUBLE')
arcpy.AddField_management(inFeatureClass, 'STD_Aspect', 'DOUBLE')
arcpy.AddField_management(inFeatureClass, 'MEAN_Slope_Deg', 'DOUBLE')
arcpy.AddField_management(inFeatureClass, 'STD_Slope_Deg', 'DOUBLE')
```

Populate these fields

Append Aggregate data to flake attribute table

```
inFeatureClass = Flake
joinField = 'OBJECTID'
joinTable = 'Aggregate'
joinField2 = flakeFIDfield  # was 'FID_Basin'
fieldList = ['FREQUENCY', 'MEAN_Slope_Deg', 'STD_Slope_Deg', 'MEAN_Aspect', 'STD_Aspect',
            'MIN_Z_Min', 'MAX_Z_Max', 'MEAN_Z', 'SUM_SArea']
arcpy.JoinField_management (inFeatureClass, joinField, joinTable, joinField2, fieldList)
```

Compute flake ellipse data:

```
inFeatureClass = Flake
zoneField = 'Id'
zoneGeometryTable = ‘FlakeEllipse'
zoneCellSize = '0.05'
outZonalGeometryAsTable = ZonalGeometryAsTable(inFeatureClass, zoneField, zoneGeometryTable,
zoneCellSize)
```

Append Flake orientation data to basin attribute table

```
inFeatureClass = Flake
joinField = 'OBJECTID'
joinTable = ‘FlakeEllipse'
joinField2 = 'OBJECTID'
fieldList = ['THICKNESS', 'MAJORAXIS', 'MINORAXIS', 'ORIENTATION']

arcpy.JoinField_management (inFeatureClass, joinField, joinTable, joinField2, fieldList)
inFeatureClass = Flake
fieldName = 'RoughnessByArea'
arcpy.AddField_management(inFeatureClass, fieldName, 'DOUBLE')
fieldName = 'EllipseRatio'
arcpy.AddField_management(inFeatureClass, fieldName, 'DOUBLE')
```

Populate these fields

Transfer statistics:

```
arcpy.CreateFileGDB_management(self.gdb_dir, summary_gdb_short.strip('.gdb'))
arcpy.CreatePersonalGDB_management (self.gdb_dir, summary_mdb_short)
```

For every GDB:

```
        summarizeFC(workspaces, 'Flake', 'FlakeScarSummary')
        summarizeFC(workspaces, 'Outline', 'OutlineSummary')
        summarizeFC(workspaces, 're_entrants', 'ReentrantSummary')
        summarizeFC(workspaces, 'Procrustes', 'ProcrustesSummary')
        summarizeFC(workspaces, 'RE_Ellipse_pnts', 'EllipseSummary')
```

summarizeFC executes:

```
arcpy.Merge_management(fullFCs, self.summary_gdb + '\\' + out_summary_table_name)
```

Copy summary feature classes from File GDB to personal geodatabase (.mdb):

```
arcpy.FeatureClassToFeatureClass_conversion('FlakeScarSummary', self.gdb_dir + '\\' +
                                summary_mdb_short, 'FlakeScarSummary')
```

For complete understanding, review the Python code. GLiMR_BuildFramework.py builds the environment and calls class methods of the GLiMRClasses.py which do all the work.