Source Unfolding Algorithm

Joe Crouch, Mark Doss, Taylor Johnson, Kira Kiviat, Justin Lanier

Arizona Summer Program 2010

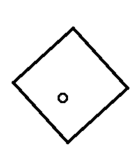
Introduction

- Given a source point on a piecewise flat manifold and a vector **v** in its tangent space, the exponential map returns the point on the manifold a distance $\|\mathbf{v}\|$ along the geodesic with initial direction given by **v**.
- The development is the inverse relation, taking points from the manifold to the corresponding points in the tangent space.
- While the development computes point-by-point, our source unfolding algorithm attempts to calculate the development a chunk at a time.

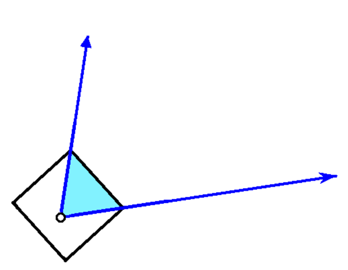
Definitions

- A **frustum** is an intersection of half-spaces with the origin on their boundary. They are represented here as ordered lists of vectors defining the lines of intersection.
- A **facet** is an n-dimensional convex polyhedron, represented here as a collection of vertices and information about connections.
- A **ridge** is an $(n-1)$-dimensional convex polyhedron that forms the boundary between adjacent facets.

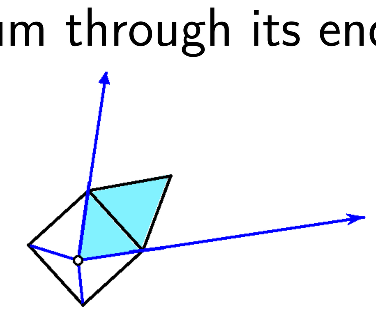
Example of Algorithm



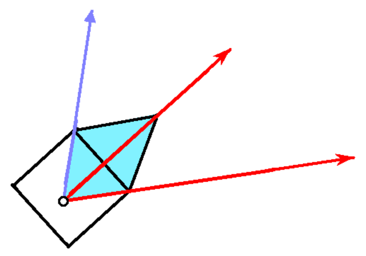
Choose a point in one of the facets.



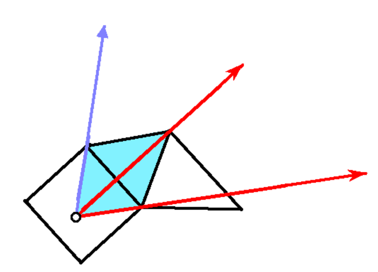
Choose a ridge to develop across, and create a frustum through its endpoints.



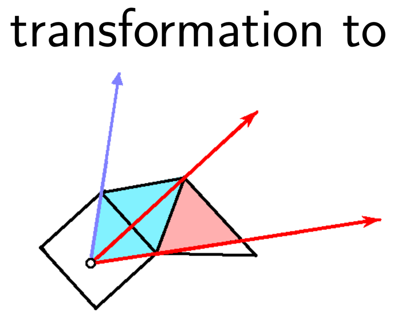
Find the facet adjacent across this ridge, and the affine transformation to place it. Add this facet to the unfolding.



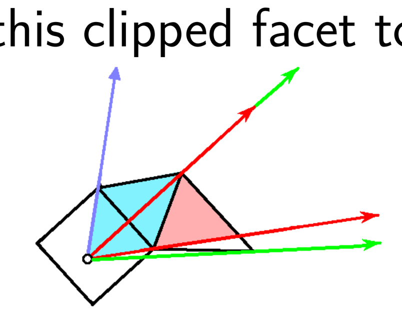
Choose a ridge on this facet to develop across, and create a frustum through its endpoints. Intersect this frustum with the old one.



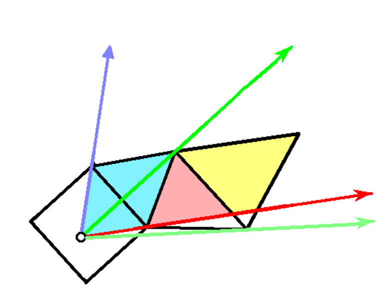
Find the facet adjacent across this ridge, and the affine transformation to place it.



Clip this new facet with the current (intersected) frustum. Add this clipped facet to the unfolding.

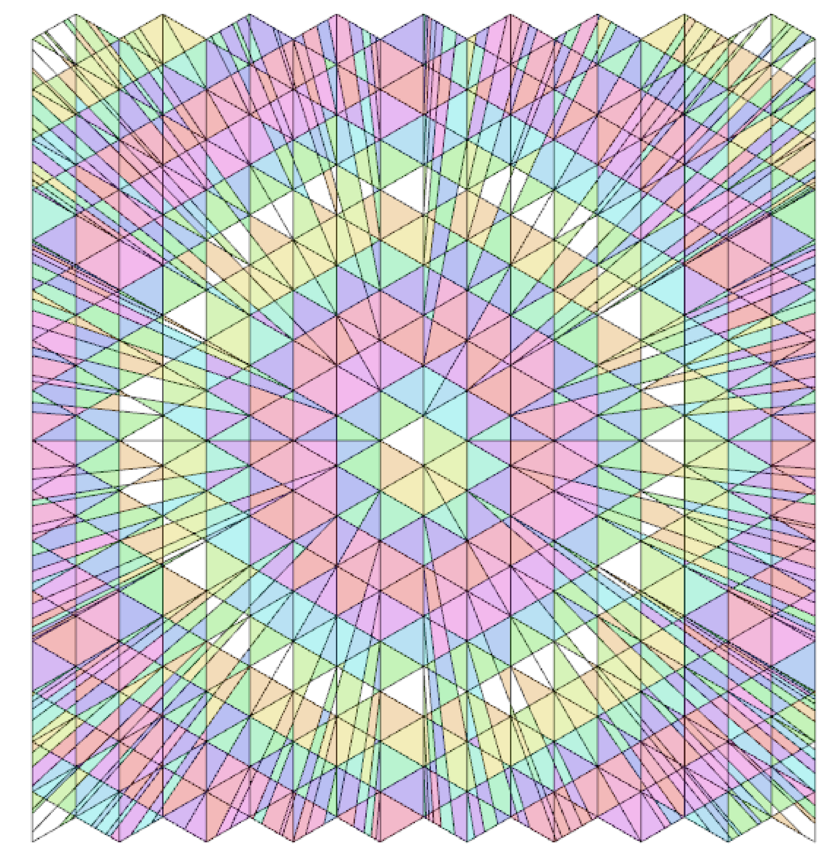


Choose a ridge to develop across and create a frustum through its endpoints.



Intersect the two frustums, find the adjacent facet, and clip this facet with the new frustum. Add this clipped facet to the unfolding and continue.

Unfolding of an Icosahedron



Pseudocode

Initial

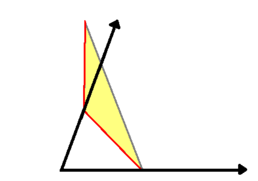
INPUT:
  length-data, combinatorics, source
START:
  find the facet F that contains source
  $\mathcal{V}$:= a full frustum
  specify orthonormal basis $u_1, \ldots, u_n$
  S := affine transformation taking source to origin and orienting $u_1, \ldots, u_n$
  Loop(F, $\mathcal{V}$, S)
END

Loop Subroutine

INPUT:
  facet F, frustum $\mathcal{V}_0$, affine transformation S
START:
  clip SF with $\mathcal{V}_0$ and add to the unfolding
  construct a list of ridges to develop across
  for each ridge in the list:
    form a new frustum through the endpoints of the ridge
    $\mathcal{V}$:= intersection of this frustum and the input frustum
    find the facet F' adjacent to F across the ridge
    T := affine transformation taking F' to F
    find the vertex on F' not on the ridge
    if (this vertex is inside the chosen bounds):
      Loop($\mathcal{F}$, $\mathcal{V}$, ST)
  end for loop
END

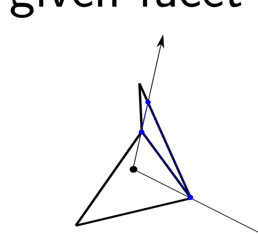
Subroutines

Choosing Ridges to Develop. Only those ridges with adjacent facets that intersect the current frustum (and are not the current facet) should be used. For instance, in two dimensions, this situation could occur:

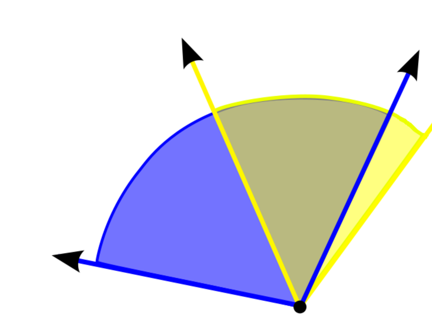


Here the two edges marked in red should be ignored.

Facet Clipping. Facets are clipped by frustums, which are defined by vectors from the origin. This is done by taking the convex hull of all points of intersection between a given facet and frustum.



Intersecting Frustums. To find the intersection of two frustums, first find the vectors of each frustum that are interior to the other. Add to this list the new vectors determined by the intersections of the two frustums' bounding facets that are interior to both frustums. Finally, order this list cyclicly into a new frustum.



Acknowledgements

This poster was mentored by David Glickenstein and Danny Maienschein, whose help is acknowledged with much appreciation.

This material is based upon work supported by the National Science Foundation under VIGRE Grant No. 0602173.