

Spherical Attribute Images

COS598b Project #2

Robert Kalnins

Goal:

The aim of this project is to produce a tool for generating Spherical Attribute Images (SAIs) from 3D polygonal meshes, with the eventual hope that they might facilitate object recognition.

Approach:

To produce SAIs, it is necessary to generate and fit a ‘regular’ 2-simplex mesh to the input model. The SAI can then be extracted from the simplex mesh. The first part of the problem involves creation of the regular mesh, while the remainder concerns iterative deformation of the mesh to fit the model, subject to regularity constraints.

Mesh Generation:

We follow the approach in the literature to create a near-regular mesh by recursive subdivision of the 20 triangular faces of an icosahedron, followed by the extraction of the dual mesh. This mesh will have the desired connectivity of three neighbours per vertex and be almost entirely composed of hexagonal faces (except for 12 pentagons). This is accomplished by storing the icosahedron such that each edge is defined in CCW order, and each edge is associated with its neighbouring face. The recursive subdivision algorithm then ensures that the newly defined triangles also maintain this data structure. Once subdivision is complete, the vertices of the 2-simplex mesh are computed as the centroids of the triangular faces, whilst their connectivity and order are trivially determined from the edge order and edge to face pointers stored in the triangular mesh data structure. The mesh is then initialized by pushing its vertices out to the surface of the unit sphere, and scaling it to encompass the input model.

Mesh Deformation (see figures at end):

The mesh is fit to the model via iterative solution to Newton’s Law, where each mesh vertex is treated as a particle of unit mass, subject to forces arising from external forces due to attraction to the model, and internal forces which enforce the regularity of the mesh:

$$\frac{d^2 P_i}{dt^2} = -\mathbf{g} \frac{dP}{dt} + \vec{F}_{ext} + \vec{F}_{int}$$

The attractive force of the model is computed by an iterative closest point approach. At each iteration, the closest point on the model to the mesh vertex must be located. Since, in general, this point will lie on a model face, rather than coincide with a model vertex, it is not sufficient to simply locate the closest model vertex. The chosen solution is to store, with each model vertex, a list of all faces incident to the vertex. To compute the closest model point, the closest vertex is first located (by brute force) and then the closest point on each face incident to the vertex is computed. The closest such point is taken as the point of attraction. The force is proportional to the distance to this point. However, to avoid fitting a simplex mesh with a highly non-uniform vertex distribution, we don't exert the force in the direction of the nearest model point, but rather exert the force along the normal direction of the simplex mesh vertex with a magnitude proportional to the projection of the vector to the model point along this normal direction. This results in a much more uniform deformation of the mesh, and avoids the (fatal) possibility of the mesh becoming self-intersecting. In practice, this is wholly necessary, as the regularity condition requires that the projection of a vertex into the plane of its neighbors be a point contained by the triangle they form. And, since the normal of a vertex is defined as the normal to this plane, exerting the attractive force in this direction is clearly advantageous.

The internal force is defined to coerce each vertex such that its projection into the plane of its neighbors lies on their centroid. However, if we naively exert a force toward the centroid, successive iterations will eventually shrink the mesh away to a singularity, as this would seek to remove the curvature at every vertex. The key is to recall that the simplex angle at the vertex is identical regardless of where the vertex lies on the circumsphere of itself and its neighbors. Thus, we exert the force toward the point on the circumsphere whose projection into the neighbors' plane lies on the centroid. This allows the regularity condition to be satisfied, and also preserves the local curvature of the vertex.

The damping term, γ , is necessary to prohibit the system from oscillating about the solution, since our forces behave as springs about the model surface, and would lead to harmonic motion.

SAI (see figure at end)

The Spherical Attribute Image is now visualized by extract by generating a copy of the undeformed (spherical) mesh, and scaling the radius of each point by the simplex angle at the corresponding mesh vertex.

Issues:

The deformation of the mesh via the forces discussed above is not as straightforward as the papers suggest. The spring constants of the internal and external forces must be tweaked to within small windows, otherwise serious pathological behavior is observed. For instance, if the external force constant is too strong, the mesh will shoot past the model and fail to

spring back. Even more amusing is when the internal force constant is set too high, causing the mesh to enter harmonic oscillations within the plane of its surface eventually causing the mesh to fly apart. If one sets both these constants to reasonable values, however, problems can still arise due to the shape of the model. Depending upon how the model attracts the mesh, it may occur that the one part of the mesh is attracted underneath another part, eventually causing the mesh to self-intersect, and forcing the regularity assumptions to fail. At this point, the definitions of the simplex angle breakdown, and the mesh flies apart.

The papers make some mention of this, but don't really offer a good solution. One approach is to ensure the mesh is initialize to within rough approximation of the model so that attraction is unlikely to cause pathologies. However, there is no satisfactory way to do generally considering the model can have arbitrarily complicated shape and convexity.

The solution I came up with was to modify the regularity constraints. Rather than exerting the internal force so as to maintain local curvature whilst pushing vertices toward the centroids, I chose to push the vertices toward the centroids at a point whose curvature corresponds to a mean of the curvatures of the vertex and its neighbors. In practice, an 85% to 15% mixture of the local curvature and neighbors' curvatures prohibits the mesh from folding upon itself, whilst avoiding excessive smoothing out of sharp corners. Furthermore, by reducing the attractive constant, we slow the attractive acceleration, allowing the curvature smoothing more time prevent the mesh from folding upon itself. Finally, a generously high damping factor also helps to keep the mesh regular by limiting acceleration, allowing more temporally local conditions to smooth the mesh via the internal force (at the cost of longer convergence times).

Findings:

I believe that SAIs could be a powerful tool for model recognition. However, there seem to be a number of shortcomings associated the process of creating the fitted simplex meshes. Namely, the various parameters of the force equations must be meticulously tweaked to ensure that the mesh does not enter into a fatal pathological state. Furthermore, it is not clear that the arrived at parameters would work universally for all models. The success found from the local mean curvature approach is encouraging, but has the unfortunate effect of smoothing away high curvature features. This might be acceptable if increasing the number of

simplex mesh vertices can be tolerated. Clearly, though, all the measures that must be taken to ensure robust convergence lead to longer run times. This technique is computationally expensive.

Aside from the technical details, this method also suffers from one major draw back: the model to be fitted must be homeomorphically equivalent to a sphere. Attempting to fit to a torus, say, would be fatal, as the mesh would try to creep into, and around, the hole, leading to an amusing failure of the algorithm. Non-maniform surfaces will also cause problem in the general case, as the mesh will creep into the hole and attempt to fit the mesh to both sides of the surface. However, in practice, small holes don't cause the algorithm to break down.

Future Work:

I find the lack of robustness (and *long* run-times) to be somewhat discouraging, though the mean curvature approach is heartening. It may also be possible to increase performance significantly by storing the model in a spatial partitioning data structure to hasten closest point searches. As well, some research can be done into what additional data which can be stored in the nodes of the SAI (moments of inertia, etc.) as the finite, tractable, number of SAI correspondences make it ideal for additional oriented parameter comparisons. I'm not sure what I'm shall pursue in the final project...

References:

- H. Delingette. Simplex meshes: a general representation for 3d shape reconstruction. In Proc. of Int. Conf. on Computer Vision and Pattern Recognition (CVPR'94), pages 856-857, Seattle, USA, June 1994.
- H. Delingette. Simplex meshes: a general representation for 3d shape reconstruction. Technical Report 2214, INRIA, March 1994.
- H. Delingette. General object reconstruction based on simplex meshes. Technical Report 3111, INRIA, February 1997.
- H. Delingette. General object reconstruction based on simplex meshes. International Journal of Computer Vision, 32(2):111-146, September 1999.
- J. Montagnat and H. Delingette. Globally constrained deformable models for 3d object reconstruction. Signal Processing, pages 173-186, 1998.
- K. Ikeuchi and M. Hebert. Spherical Representations: from EGI to SAI. Tech. report CMU-CS-95-197, Computer Science Department, Carnegie Mellon University, October, 1995.



