Summary of *Fast Mapping and ...*

Victor Shu

The paper proposed an efficient approach for merging topological structures of genus-zero meshes and animating a morph between them. The proposed approach includes an efficient way of parameterizing meshes. The results are demonstrated through an application showing the morph between meshes.

Two kinds of mesh parameterization methods can generally be categorized into two kinds: Progressive method and relaxing method. In progressive methods, a mesh is progressively simplified to a base mesh, and the base mesh is trivial for spherical projection. The method preserves features of meshes, but it could depress the surface of the sphere. A relaxing method, on the other hand, updates vertex positions based on the curvature. It usually adopts global optimization strategies like minimizing flat energies, but it could also lead to triangle degeneration. The proposed approach can avoid triangle degeneration and foldover issues.

The proposed approach is made up of four steps. Simplification takes a genus-zero triangle mesh and simplifies it by collapsing edges. Parameterization and optimization insert the vertices back one by one. Feature alignment modifies the spherical representations of meshes and aligns the user-specified features. Remeshing and morphing generate a mesh that can represent both of the original meshes.

Instead of preserving features that could make a simplified mesh non-convex, the simplification of the proposed approach removes features in an early stage. The parameterization maps the vertices of a mesh onto a sphere. It first parameterizes the simplified mesh onto the sphere, which is trivial; then, it inserts vertices that are simplified back into the sphere iteratively. A spherical kernel is introduced to find the optimal position of the vertex. It also includes a way to measure the degree of distortion of each kernel point to place the vertex at the optimal position.

When inserting vertices back, the sphere could become unbalanced with many vertices in a small area, while other areas contain fewer vertices. The proposed approach introduces an optimization algorithm to reposition vertices after inserting goes for a while. After that comes the feature alignment. There are two steps in feature alignment. The rough alignment rotates the sphere to align the features roughly. It can prevent the exact alignment degenerating the triangles. The exact alignment moves the user-defined feature points of both spheres to their midpoints, or it introduces distortion.

As the original meshes are both mapped onto a sphere, a supermesh that contains features in both spheres is generated. This process is called remeshing. All vertices of the spheres are placed on a spherical surface, and the supermesh is constructed with a subdivision method. The supermesh sphere starts from an octahedron embedded in a unit sphere, and then it is iteratively subdivided until each subdivision contains at most one vertex from one of the original mesh. Then the same process is applied to the other original mesh.

The result of the approach is evaluated regarding performance and quality. The parameterization speed is 58K triangles per minute while other related methods' ranging from 5K to 14K. The speed of some parallelized implementation of other works can process around 150K triangles per minute, while the non-parallelized implementation of it can handle 30K. Overall, the optimization stage is the most expensive one in the spherical parameterization of the proposed approach. The figures in the paper show that the result has a high visual quality. The error between rendered images of original models and the proposed method is not high even on complex surfaces like fingers and toes.

In conclusion, the approach of the paper is very efficient and produces high-quality parameterization and morphing. There are still limitations in the method that artifacts exist in the results when a long and thin triangle of one sphere overlaps multiple triangles of another sphere. The future work includes introducing various radii to support different sizes of features, more exploration of multi-phase alignment, and parallelized implementation.