Selectively Refinable Progressive Meshes: The ideal data structure for complex numerical tasks?

S. Seeger, G. Häusler

Complex numerical tasks such as correlation problems in several dimensions, solving partial differential equations with lots of (spatial) boundary conditions or the visualization of highly detailed models with interactive control all face the same difficulty: The great amount of input data is too large to be handled in real-time. The only way to overcome this problem is to reduce the input data. Progressive Meshes (PM) [1] are in our opinion the ideal way to perform this task for surface data: In describing the original triangle mesh as a small base mesh plus an ordered set of vertex split operations one gets a continuous hierarchy of triangle meshes. All data points are ordered with respect to their importance which is defined by an arbitrarily choosable criterion. This criterion is typically of geometric nature so that each mesh in this hierarchy can be the best approximation of the original mesh for a given number of vertices. However, for many applications we often are not interested in a best approximation of the whole input data but in a best approximation of *interesting parts* of the data, which are defined by the application at runtime and have not to be of geometric nature. For example, for the visualization of a highly detailed model we are only interested in those parts that we can see: triangles on the backface of the model or triangles that are smaller than a pixel on the screen have not to be rendered. But these criteria change at runtime when the model is rotated or zoomed. Another example is the alignment of two range images of an object that have been taken from different viewpoints (called registration). This is a typical correlation problem where we are in particular interested in the overlapping parts of the range images, which are determined only during runtime.

Selectively Refinable PM [2] are in our eyes the ideal data structure for these tasks. They allow to skip some of the vertex split operations (in the ordered set of split operations) of a PM while always maintaining a valid triangle mesh. In this way it is possible to locally restrict refining or coarsening of the mesh in a best approximation sense to regions of interest.

Until now we have implemented an optimized converter for transforming a standard triangle mesh into a PM data structure and a viewer for the visualization of the PM. The viewer permits to modify the resolution of the triangle mesh in real-time. In addition, a given number of frames per second while moving an arbitrary model can be guaranteed. The model is coarsened (or refined) until the desired number of frames per second is reached. In this way the viewer always chooses the optimal size of the model for interactive control so that we are independent of the size of the original model, the speed of the computer we use, or how many applications are running in parallel on our machine.

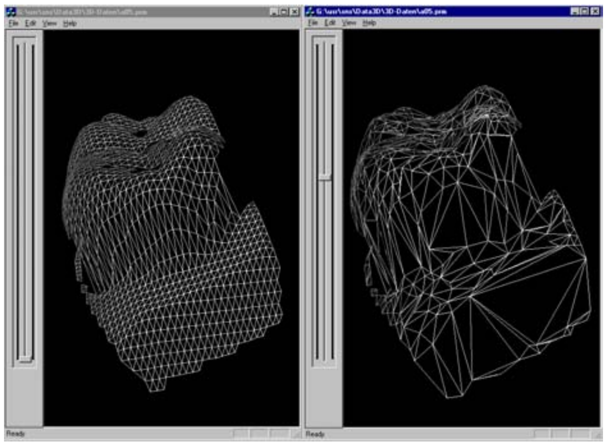

Figure 1: PM viewer with real-time resolution control.

An amazing feature of the PM data structure is that it needs less memory than a standard triangle mesh for which the coordinates of all vertices (3 floats per vertex) and vertex indices of all triangles (3 integers per triangle) are saved. Another feature of a PM is the possibility to progressively transfer it over low-bandwidth communication lines as e.g. the internet: first the small base mesh is transferred followed by the vertex split operations. At each time the transferred data define a valid triangle mesh which can be already rendered. Of course the *Progressive Mesh* gets its name from this feature.

In future work we will build up the data structure of Selectively Refinable PM on top of the implemented PM. Then we will be ready to combine this ideal data structure for complex numerical tasks with our registration approach described in [3].

[1] H. Hoppe, Progressive Meshes, *Siggraph '96 Proceedings*, pp. 99-108, 1996.

[2] H. Hoppe, View-dependent Refinement of Progressive Meshes, *Siggraph '97 Proceedings*, pp. 189-198, 1997.

[3] S. Seeger and G. Häusler, A Robust Multiresolution Registration Approach. In H. Niemann, H.-P. Seidel, and B. Girod, editors, *Vision, Modeling and Visualization '99*, pp. 75-82, 1999.

GG