Effective Removal of Detail for a Multiresolution Viewer

Kari Pulli (Univ. of Washington, Seattle, WA) Tony DeRose (Pixar Animation Studios)

Abstract. Several authors have described how to construct a multiresolution surface model by adding wavelet coefficients to a base model in arbitrary order. We present a method that allows one to remove detail from a wavelet description, which is important for interactive viewing of multiresolution models.

Introduction. Wavelet analysis has been shown to be an effective method for automatically generating accurate models of geometric models at different levels of resolution [4]. An especially attractive feature of wavelet-based models is that the most accurate version of the model already contains all the coarser approximations, so memory is not wasted when storing several models of the same object at different resolutions.

An interactive viewer of multiresolution models may be used to render different amounts of detail depending on factors such as the the resolution of the output and the speed of the graphics engine. Should the user zoom in to an object, more detail can be added, but if the user zooms out, a coarser version that is faster to display can be rendered with little or no degradation in perceived image quality. Whereas adding or removing detail is usually a discrete event that may cause visible "popping", wavelet techniques allow detail to be added or removed in a continuous fashion.

Meyers [3] and Lounsbery [2] showed how the detail contained in a wavelet coefficient can be added into a polygon and a polyhedral surface model, respectively. However, these authors did not present a method for incrementally removing detail to reduce the polygon count. Certain *et. al.* [1] have recently developed a multiresolution surface viewer that stores models at different resolutions and then adds detail starting from the most appropriate stored model. In this paper we present a method for removing detail directly from the current representation without having to regrow the model from a lower detail version.

Adding detail. Let's first review how detail is added into a polygon [3]. In Figure 1 we have a coarse representation of a piecewise linear polygon, and we are about to add a wavelet centered on the 17^{th} vertex. First we need to add to the polygon all the wavelet vertices that it does not already have (vertices 12, 14, 17). Then we need to add to the wavelet the vertices of the polygon that lie within the support of the wavelet (vertex 21). Finally, we just add the nonzero values of the wavelet vertices to the polygon (vertices 14, 16, 17, 18, 20, 21). Note that depending on which vertices already exist in the polygon, we may have to add anywhere from 0 to 7 new vertices for a single wavelet.

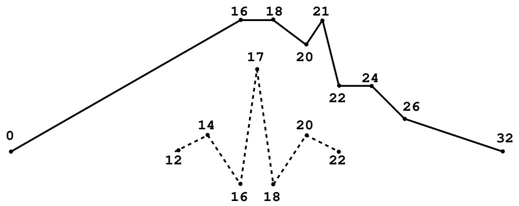

Figure 1 Adding a wavelet creates several new vertices.

Removing detail. Removing the geometry of a wavelet is easy: add to the wavelet the vertices of the polygon within the support of the wavelet and subtract all the wavelet vertex values from the polygon vertices. The geometry is simpler, but rendering and manipulating the simpler version is as expensive as before. We may be able to remove some of the vertices, but only when the vertex is not required by any of the remaining wavelets.

Our solution for determining which vertices can be removed is to use reference counting: each time we insert a wavelet, we increase the reference count of the wavelet's vertices (in the example for the vertices 12, 14, 16, 17, 18, 20 and 22, but not 21). When a wavelet is removed, we reduce the reference count. All vertices with zero count can be safely removed.

Generalizing to surfaces. Using this idea for surfaces is somewhat more complicated. As illustrated in Fig. 2, suppose we add a wavelet in the middle of the large triangle (**ijk**). In order to avoid cracks between the triangles when rendering, it is not enough to add only the triangles and vertices of the wavelet into the mesh; we also need to recursively add the "parent" triangles and vertices (**iln**, **lmn**), as well as their parent (**ijk**) and siblings (**jlm**, **mkn**). For more detail, see [2, 1]. As in the curve case, the reference count of each wavelet vertex is increased.

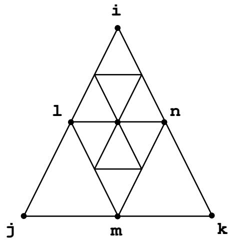

Figure 2: Adding a surface wavelet.

Removing detail is also more complicated for surfaces. Removing the geometry of the wavelet is analogous to the curve case, as is the maintenance of reference counts. However, a vertex cannot be removed when its reference count goes to zero: it may be a parent vertex for some finer level wavelet in the mesh. Four child triangles may be collapsed into a parent triangle if and only if their children do not exist in the mesh and the counts at the vertices that split the parent's edges are zero. Note that collapsing four triangles may lead to a cascade of further simplifications of the mesh.

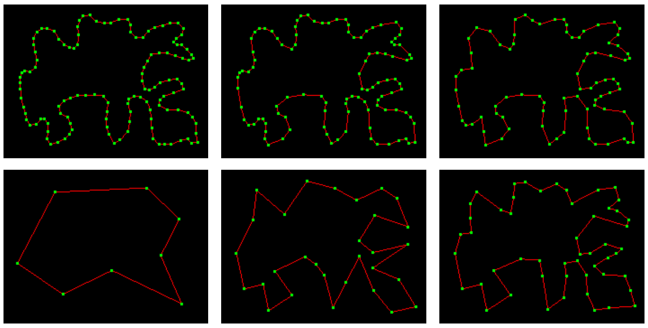

Figure 3 Removing detail from a 2D polygon. The sequence proceeds clockwise starting from left upper corner.

References

[1] A. Certain, J. Popović, T. DeRose, T. Duchamp, and W. Stuetzle. Interactive multiresolution surface viewing. In *Proceedings of SIGGRAPH '96*, August 1996.

[2] M. Lounsbery. *Multiresolution analysis for surfaces of arbitrary topological type*. PhD thesis, Department of Computer Science and Engineering, Univ. of Washington, September 1994.

[3] D. Meyers. Multiresolution tiling. *Computer Graphics forum*, 13(5):325–340, 1994.

[4] Eric Stollnitz, Tony DeRose, and David Salesin. *Wavelets for Computer Graphics: Theory and Applications*. Morgan–Kaufmann, 1996.