Load Balancing for Smooth LODs

Michael Wimmer, Dieter Schmalstieg

December 9, 1998

Abstract

The paper shows how to derive an analytic formula for the following problem: given a set of objects in a continuous level of detail representation, and given a total number of polygons to display, determine the number of polygons to use for each object so that the best overall appearance is achieved. This improves on the situation of discrete levels of detail, where the problem has been shown to be equivalent to a constrained knapsack-problem.

Keywords: virtual environments, load balancing, level of detail, smooth LODs

1 Introduction

In order to display complex virtual environments with detailed geometric objects, Level Of Detail objects (LOD) can be used to reduce the overall complexity of the scene. LOD objects have similar visual appearance to the original object, but fewer vertices and are thus faster to render.

Previously, discrete sets of levels of detail [2] have been used to approximate objects at various viewing distances. More recently, continuous level of detail methods ([4, 3]) have been introduced that can generate approximations with an arbitrary number of vertices.

At runtime, rendering a scene includes the task of selecting an appropriate level of detail for each object. One approach determines a 'polygon budget', i.e., the number of polygons per frame that can be rendered interactively. In the case of discrete levels of detail, this means that for each object, an appropriate level of detail has to be selected, so that the total number of polygons does not exceed the polygon budget and the best possible overall appearance is achieved. Funkhouser et al. [1] have shown this to be an optimization problem equivalent to a constrained Knapsack problem

With the availability of continuous levels of detail, this process has to be re-evaluated. We show that it is possible to find a closed form solution to the problem which can be cheaply evaluated for each frame.

2 Problem Statement

As in [1], we associate a cost and a benefit with each level of detail of an object.Assuming that the benefits and costs of individual objects are independent,

let $f_i(x)$ denote the benefit associated with rendering an approximation consisting of x triangles of the i-th object. The cost of rendering this approximation is its number of triangles. The optimization problem then consists of maximising

$$f(x_0, \ldots, x_n) := f_0(x_0) + \cdots + f_n(x_n)$$

so that the polygon budget P is met:

$$x_0 + \cdots + x_n = P$$

which can also be expressed as $g(x_0, \ldots, x_n) = 0$, where

$$g(x_0, \ldots, x_n) := x_0 + \cdots + x_n - P$$

We will later see how to appropriately define $f_i(x)$.

3 Lagrange Multipliers

The optimization problem can easily be solved using Lagrange Multipliers. To do this, we have to build the extended benefit-function $F(x_0, \ldots, x_n, \lambda)$, where λ is the Lagrange Multiplier (a Lagrange Multiplier is inserted for every constraint):

$$F(x_0, \ldots, x_n, \lambda) := f(x_0, \ldots, x_n) - \lambda g(x_0, \ldots, x_n)$$

The solution can now be found by building the partial derivatives $F_{x_0}, \ldots, F_{x_n}, F_\lambda$, setting them to 0 and solving the resulting (hopefully linear) system of equations. In our case, we have

$$\frac{\partial F(x_0, \ldots, x_n)}{\partial x_i} = f_i'(x_i) - \lambda = 0 \tag{1}$$

and

$$\frac{\partial F(x_0, \ldots, x_n)}{\partial \lambda} = g(x_0, \ldots, x_n) = 0 \tag{2}$$

If we assume $f_i'(x)$ to be a continuous, monotonous function, we can find its inverse $f_i'(x)^{-1}$ and use that to solve for x_i:

$$x_i = f_i'(\lambda)^{-1} \tag{3}$$

Plugging this into (2) yields

$$f_0'(\lambda)^{-1} + \cdots + f_n'(\lambda)^{-1} = P \tag{4}$$

which we have to solve for λ so we get $x_0, \ldots, x_n$. This is as far as we can get without making any further assumptions about the benefit functions $f_i(x)$.

4 Choosing a Benefit function

We have already stated that the derivative of the individual benefit functions $f_i'(x)$ should be continuous and monotonous so that we can solve the system. It's actually the *derivative* of the function that is used to determine the shape of the benefit curve: a very high benefit should be assigned to the first few triangles that are displayed for an object, since they contribute very much to its visual appearance, and less benefit to later triangles, as they have a relatively low influence on appearance. So what we actually want is a benefit function with a decreasing derivative. Of course, the derivative has to be strictly positive (since adding triangles will never decrease the benefit). Choosing

$$f_i'(x) := w_i x_i^{-\alpha} \text{ with } \alpha, w_i > 0$$

where w_i denotes an arbitrary weight assigned to object i, fits the purpose. Solving for x_i as in (3) gives

$$x_i = (\frac{\lambda}{w_i})^{-\frac{1}{\alpha}}$$

which we can now plug into (4):

$$(\frac{\lambda}{w_0})^{-\frac{1}{\alpha}} + \cdots + (\frac{\lambda}{w_n})^{-\frac{1}{\alpha}} = P$$

and isolate and solve for $\lambda^{-\frac{1}{\alpha}}$:

$$\lambda^{-\frac{1}{\alpha}} = \frac{P}{\sum_{i=0}^{n} w_i^{\frac{1}{\alpha}}}$$

Thus, we obtain a formula for calculating individual triangle counts:

$$x_j = P\frac{w_j^{\frac{1}{\alpha}}}{\sum_{i=0}^{n} w_i^{\frac{1}{\alpha}}} \tag{5}$$

It can be observed that, as long as the weights are positive, the resulting triangle counts assigned to objects will always be positive.

5 Examples

We will now examine some specific choices for α and the weights w_i.

5.1 Funkhouser

Setting $\alpha = 3$, $w_i = 2A_i c$, where A_i denotes the projected area of the object, yields Funkhouser's benefit function:

$$f_i(x) = \int 2A_i c\frac{1}{x_i^3}\,dx = -A_i\frac{c}{x_i^2}$$

Choosing an integration constant of A_i even gives exactly the formula used by Funkhouser,

$$f_i(x) = A_i(1 - \frac{c}{x_i^2})$$

The optimal triangle count for each object can now be evaluated according to (5):

$$x_j = P \frac{\sqrt[3]{A_j}}{\sum_{i=0}^{n} \sqrt[3]{A_i}}$$

5.2 Schmalstieg

Setting $\alpha = 1$ and $w_i = A_i$ gives the benefit function

$$f_i(x) = \int A_i \frac{1}{x_i}\, dx = A_i \ln x_i$$

and yields the very simple formula for the triangle count

$$x_j = P \frac{A_j}{\sum_{i=0}^{n} A_i}$$

5.3 Faure/Wimmer/Wonka

We have observed that it might be a good idea letting w_i depend not only on the projected area A_i, but also on the number of available polygons, p_i, following the rationale that an object that is modeled with many triangles is approximated worse by a given number of triangles than an object with fewer triangles. Choosing

$$w_i = \beta A_i p_i^{\beta} \text{ where } \beta = \alpha - 1$$

gives the desired benefit function

$$f_i(x) = \int \beta A_i p_i^{\beta} \frac{1}{x_i^{\beta-1}}\, dx = -A_i (\frac{p_i}{x_i})^{\beta}$$

The resulting formula for the triangle counts for specific values of α can easily be derived from formula (5).

6 Caveats

The formula works very well if the number of total triangles for each object to be drawn at least equals the polygon budget P. If that is not the case, an iterative solution has to be used: whenever $x_i > p_i$ (where x_i is the number of triangles chosen for the object and p_i is the total number of triangles in that object), x_i is set to p_i, a new polygon budget is calculated as $P_{new} = P - p_i$ and the calculation is repeated for the remaining objects.

7 Acknowledgments

The work presented in this paper was sponsored by the Austrian Science Foundation (FWF) under contract no. P-11392-MAT. Many thanks to F. Faure and P. Wonka for their valuable input.

References

[1] Thomas A. Funkhouser and Carlo H. Séquin. Adaptive display algorithm for interactive frame rates during visualization of complex virtual environments. In James T. Kajiya, editor, *Computer Graphics (SIGGRAPH '93 Proceedings)*, volume 27, pages 247–254, August 1993.

[2] Paul S. Heckbert and Michael Garland. Survey of polygonal surface simplification algorithms. Technical report, CS Dept., Carnegie Mellon U., to appear.
URLhttp://www.cs.cmu.edu/ ph.

[3] Hugues Hoppe. Progressive meshes. In Holly Rushmeier, editor, *SIGGRAPH 96 Conference Proceedings*, Annual Conference Series, pages 99–108. ACM SIGGRAPH, Addison Wesley, August 1996. held in New Orleans, Louisiana, 04-09 August 1996.

[4] D. Schmalstieg and G. Schaufler. Smooth levels of detail. *Proceedings of VRAIS'97*, pages 12–19, 1997.