Improved Billboard Clouds for Extreme Model Simplification

I.-T. Huang, K. L. Novins and B. C. Wünsche

Graphics Group, Department of Computer Science, University of Auckland, Private Bag 92019, Auckland, New Zealand.

ihua006@ec.auckland.ac.nz, novins@cs.auckland.ac.nz, burkhard@cs.auckland.ac.nz

Abstract

Computer generated scenes are usually represented by polygon models. While graphics hardware capabilities have advanced rapidly in recent years many natural scenes are still much too complex to be rendered in real-time as polygon models. A solution is to simplify such models by reducing the number of polygons in them. However, for many complex models, such as models of trees, a polygon reduction is difficult to achieve. Recently a new image-based method called "Billboard Clouds" has been suggested for extreme model simplification. The idea is to replace a complex model by a set of texture mapped images (billboards) of it. The algorithm can deliver impressive results but has several severe limitations: the set of possible planes is restricted to a discretised plane space and the number of resulting billboards can not be controlled directly. In this paper we present a new algorithm for generating billboard clouds using k-means clustering. We show that our algorithm is fast and offers improved performance and user control.

Keywords: model simplification, billboard clouds, texture mapping, real-time rendering

1 Introduction

Modern graphics hardware is purpose build for the fast rendering of large numbers of texture mapped polygons. Consequently polygon meshes are a popular representation of models in Computer Graphics. However, despite recent advances in graphics hardware many natural scenes represented by polygons are still much too complex to be rendered interactively. A possible solution is to use mesh reduction techniques which simplify polygon models by reducing the number of polygons in them. Examples are polygon merging techniques which combine coplanar or nearly coplanar polygons, vertex or edge elimination techniques which delete vertices or edges and retriangulate the resulting holes, and edge collapsing techniques which replace edges by vertices obviating the need for retriangulation. A survey of mesh reduction techniques can be found in [1].

While mesh reduction techniques work well for some models, other models, e.g. models of trees, can only be reduced to a small number of polygons by severely changing the shape and topology of the model. An alternative approach is to use an image-based model simplification. An example is billboards which are popular in Computer Games and represent the complex geometry of a model by a single texture mapped polygon depicting the model's image from a certain view point. The advantages of image-based simplification methods are that complex geometry can be represented efficiently by texture maps and that it is not necessary to manage texture coordinates during mesh simplification.

If a model is approximately rotational invariant the illusion of a 3D model within an interactive application can be achieved by rotating the corresponding billboard so that it always faces the current view point. However, if the appearance of the model differs considerably for different view points this approach does not work. A more complex representation without this drawback are light fields which are 4D functions which encode 2D images of the model from different view points [2]. New views of the model can be generated from arbitrary camera positions without depth information simply by combining and resampling the available images stored in the light field. However, light fields are rather complex to implement and in order to get a good quality reproduction they must encode images from many different viewpoints resulting in a large data set. Recently "Billboard Clouds" have been proposed which combine ideas from mesh reduction techniques and image-based model simplification [3]. Billboard clouds represent models as a set of planes with texture and transparency maps. The resulting models can be efficiently rendered using modern graphics hardware and in contrast to mesh simplification methods no mesh connectivity information is necessary for their computation.

The next section explains billboard clouds for extreme model simplification. We identify the limitations of the algorithm and suggest in the subsequent section an improved implementation which uses a variation of k-means clustering in order to find optimal billboard planes. We conclude with an analysis of the efficiency and approximation quality of our algorithm.

2 Billboard Clouds

Billboard clouds represent models as a set of planes with texture and transparency maps. The original algorithm proposed by Décoret et al. uses a geometric error threshold and a greedy method to select suitable representative planes from a discrete plane space. The selection of planes is based on a so-called density measure that incorporates three constraints: all vertices of faces must be within a specified error distance, the area of faces projected onto the plane should be maximised, and a plane must contain all nearly tangential faces in its neighbourhood.

2.1 Shortcomings and Suggested Improvements

The original algorithm for generating billboard clouds described in [3] contains several limitations. Since the plain space is discreet it must have a high resolution in order to enable the algorithm to find planes with a low approximation error. However, a high resolution plane space increases the running time of the algorithm since more planes must be tested. In this paper we make the plane space redundant and instead suggest a variation of k-means clustering in order to find optimal billboard planes for a set of faces.

Another limitation of the algorithm is that the user can not specify the desired number of billboards. Instead several parameters have to be selected (the plain space resolution and a geometric error and a measure of tangentiality for the selection of planes) and the algorithm will find a billboard cloud fulfilling these constraints. For many Computer Graphics application it is desirable to be able to specify the complexity of a model. The algorithm presented in this paper will provide this capability.

3 An Improved Algorithm for Generating Billboard Clouds

The main idea of our algorithm is to let the user specify the number k of desired planes and to use a variation of k-means clustering to find a suitable set of k billboard planes.

3.1 K-Means Clustering

K-means clustering is a popular method for partitioning data points into disjoint subsets so as to minimise a sum-of-squares criterion [4]. The basic idea is to first assign the data points (at random) to the K sets. Then the centroid is computed for each set and the data points are assigned to the centroids according to a distance metric. These two steps are repeated until there is no further change in the assignment of the data points.

In our case we want to find k planes and assign triangular faces to them in a way which minimises a given error metric. We choose as error metric the distance of a triangle's centroid to a plane. The distance is computed as the sum of the Euclidean distances of the triangle vertices to the plane. We can omit the division by three necessary to compute the average value since linearly scaling the metric does not influence the result of the clustering algorithm.

Next we have to find best fit planes for each cluster of triangles. We achieve this using Singular Value Decomposition (SVD) [5] and our distance metric for triangles. A drawback of this method is that the SVD only considers vertices of triangles but not their normals. Hence it is possible that all triangles assigned to a best fit plane are perpendicular to it. In this case the projected area of all triangles is zero and the billboard contains no information about the model. However, we found that despite this drawback the method works well in practice [8].

The proposed k-means clustering algorithm gradually approaches a locally optimal solution but might not stop at a stable state since "flipping" of triangles between clusters can occur. Hence we compute for each step the total distance of triangle centroids to their associated billboards. The k-means clustering algorithm terminates if a local minimum for this total distance is reached.

3.2 Assignment of Triangles to Empty Clusters

During the proposed clustering algorithm some clusters might end up with no triangles assigned. We detect such empty clusters in each iteration and assign the "worst" triangle from all other clusters to it. In this case the "worst" triangle is the one which has the largest angle between its normal and the normal of its best fit plane. As a result the algorithm will compute a new best fit plane containing this triangle.

3.3 Initialisation

An important aspect of our algorithm is its initialisation, i.e. the initial distribution of billboard planes. Our first implementation used randomly distributed and oriented planes. However, we found that the quality of results varied drastically when repeatedly running the algorithm. Hence we developed a more advanced initialising procedure which consists of three steps as described in the following paragraphs.

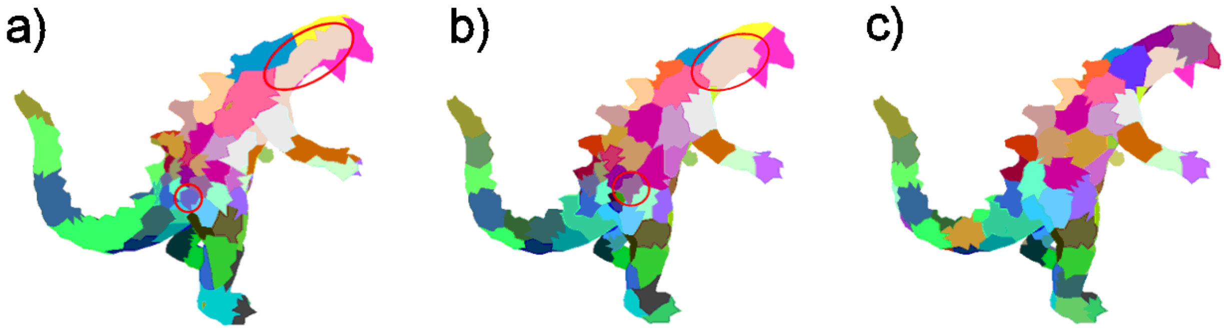

Figure 1: Distribution of clusters after step 1 of the initialisation procedure (a), after step 1 and 2 (b), and using all three steps of the initialisation procedure (c).

STEP 1 - Minimal Discrete Energy Distribution of Tangent Planes on a Bounding Sphere:

The first step of the initialisation procedure starts with computing a bounding sphere of the model. We then distribute k points over the sphere's surface using a minimal discrete energy method [6,7], compute the sphere's tangent planes at these points, and assign triangles to these planes (clusters) using our minimal distance criterion. Figure 1 (a) shows the distribution of clusters after step 1 of our initialisation procedure. It can be seen that the coverage of clusters can vary strongly (e.g. compare the two clusters indicated by the red circles).

STEP 2 - Cluster Coverage Variance Reduction:

In order to reduce the differences in coverage we employ another k-means algorithm which uses the following distance metric and centroid definition: The centroid of a cluster is computed by projecting all triangles onto its best fit plane and by determining the centroid of the triangle closest to the centroid of all projected triangle vertices. The distance metric is given by the average distance of the vertices of a triangle to the centroids of the cluster. The improvement achieved by adding this step to the initialisation procedure is illustrated in figure 2 (b). However, the results are still not satisfactory because the k-means algorithm optimises coverage only locally.

STEP 3 - Iterative Removal of Minimum Coverage Clusters:

In order to reduce the differences in coverage even further we employ an iterative procedure which removes in each step the smallest cluster, redistributes the triangles to the remaining clusters, and creates a new cluster using the plane of the triangle which is furthest away from the centroid of the largest cluster. The procedure is stopped when the variance of cluster radii reaches a local minimum. Here we define the cluster radius as the largest distance between the centroid and any vertex of that cluster. Figure 1(c) demonstrates that the procedure dramatically reduces variances in cluster coverage.

4 Crack Reduction

The algorithm described so far often creates billboard clouds with clearly visible cracks (see figure 2 (b)). There are several possible reasons for this:

- The texture map representing the projection of two adjacent triangles can contain a gap between them due to alias problems.
- Because of the way best fit planes are computed some triangle information can be missed when projecting them onto billboards (e.g. if a triangle is orthogonal to its best fit plane).
- Because each triangle is projected onto only a plane, there will be cracks between billboards as indicated by the dashed ellipsoid in figure 3.

In order to reduce the number and severity of cracks we project some triangles onto multiple planes. We determine the triangles which must be projected onto multiple planes by first computing for each plane the smallest envelope containing all triangles of that cluster. Next we determine all pairs of envelopes which intersect. Two envelopes intersect if each envelope contains a triangle which has at least one vertex lying inside the other envelope. For all intersecting envelopes we project those triangles which lie inside their intersection onto both planes. If only a part of a triangle lies within the envelope of another plane we project only that part. This is achieved by using a stencil buffer for its rendering. The process is illustrated in figure 4. More implementation details are described in [8]. Figure 2 shows a polygonal model (a) and its billboard cloud representation before (b) and after (c) crack reduction.

Figure 2: A polygonal model (a) represented by a billboard cloud before (b) and after (c) crack reduction.

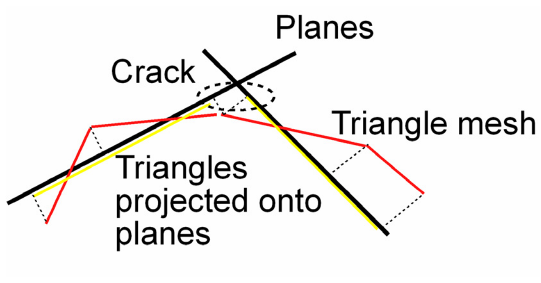

Figure 3: Crack between billboards.

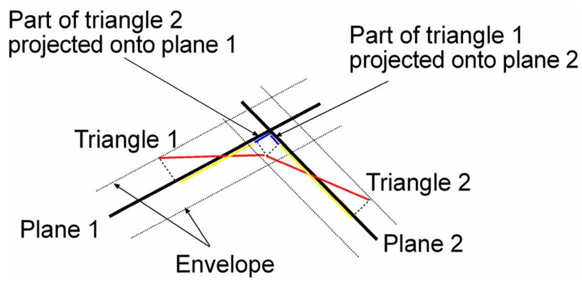

Figure 4: Crack removal by multiple projections.

5 Results

In order to evaluate the quality of the results produced with our algorithm we compute four metrics for the visual error between a model and its billboard cloud. Each error is computed by summing the error for 50 different view directions obtained by evenly distributing 50 points over a sphere [6,7]. We use the following four error metrics to quantify the error between an image of the original model and the corresponding image of a billboard cloud obtained from a given view direction:

Colour error: The colour error between the two images with the pixels $\mathbf{p}_{ij}$ and $\mathbf{q}_{ij}$ is defined as

$$\sum_{i,j \text{ where } p_{ij} \text{ and } q_{ij} \text{ show the model}} \left\| \mathbf{p}_{ij} - \mathbf{q}_{ij} \right\|_2$$

We express pixel colours in CIE Luv coordinates since this colour space is perceptually uniform, i.e. the Euclidean distance between two colours is always proportional to their perceptual distance [9].

Depth error: The depth error between the two images is defined as the sum of the Euclidean distances between the depth buffer values of all pixels showing the model in both images.

Gap error: The gap error between the two images is defined as the number of pixels where the original model is drawn but not the corresponding billboard cloud.

Coverage error: The coverage error between the two images is defined as the number of pixels where the billboard cloud is drawn but not the original model.

We have compared the visual quality of billboard clouds generated with our algorithm for different numbers of billboards. Figure 5 (a) shows results obtained for the model in figure 2 (a). It can be seen that both the colour error and the depth error decrease nearly monotonically. Both errors seem to converge towards a non-zero value, which indicates that it is not possible to completely reconstruct a polygonal model with billboard clouds. The results also suggest that for a given polygon model there is a certain number of billboards beyond which little improvement in visual quality is achieved. In this

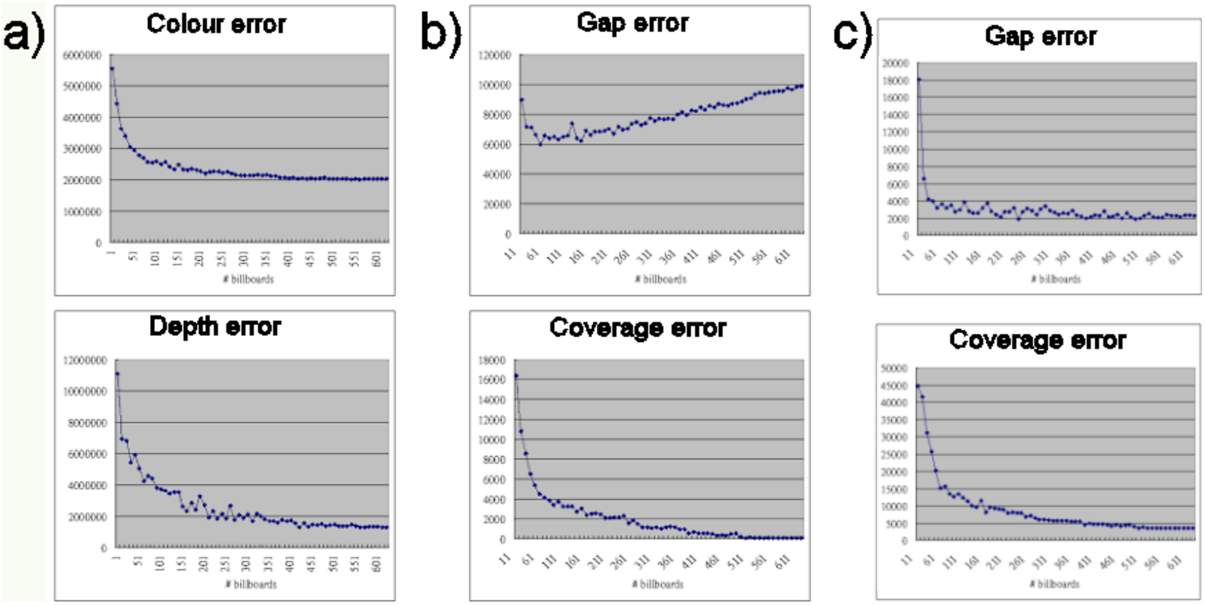

Figure 5: Error measures for the model shown in figure 2 (a) computed without (a,b) and with (c) crack reduction.

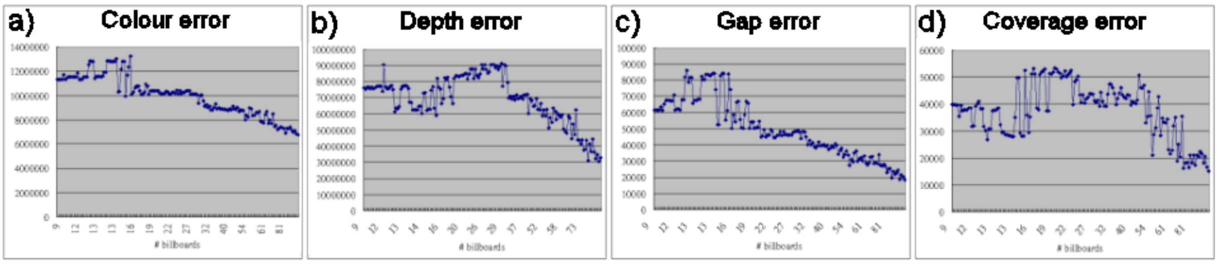

Figure 6: Error measures for the model shown in figure 2 (a) computed using the original billboard cloud algorithm by Décoret et al. [3].

example using 150 billboards seems to give a good balance between visual quality and simplicity of the representation.

Figure 5 shows the results for our algorithm without (b) and with (c) crack reduction, respectively. It can be seen that without crack reduction the gap error (number of missing pixels) increases approximately linear with an increasing number of billboards. Using our crack reduction algorithm dramatically reduces the number of cracks at the expense of creating erroneous pixels showing the model. However, the number of excess pixels reduces with an increasing number of billboards.

We have also compared our algorithm with the original billboard cloud algorithm by Décoret et al. [3]. The results of applying that algorithm to the model shown in figure 2 are displayed in figure 6. Since the algorithm by Décoret et al. does not allow the specification of a fixed number of billboards we have used a plane space of fixed resolution and varied the error tolerance values in order to get results for different numbers of billboards. It can be seen that the colour, depth and gap error reduce with an increasing number of billboards. The colour error is consistently larger than with our algorithm. The gap error is smaller than for our algorithm but the depth error is again considerably larger. This can be partially explained by the fact that the original algorithm doesn't differentiate between front and back faces and front facing polygons can be mapped onto back facing billboards. Hence when rendering a model produced with the original algorithm back face removal must be disabled. This has the effect that back faces cover most gaps between front facing polygons and the corresponding pixels are not counted in our gap error measure but in the depth error measure. In contrast our algorithm allows a differentiation between front and back faces and we can enable back face removal for billboard clouds of solid models. Additional measurements show that the execution speed of our algorithm is comparable to that of Décoret et al. when using a low resolution plane space and it is by an order of magnitude faster than their algorithm executed using a high resolution plane space.

6 Conclusion

We have presented a new algorithm for generating billboard clouds which allows the user to specify an arbitrary number of billboards. Our algorithm uses a variation of k-means clustering and makes it possible to find arbitrarily oriented and positioned best fit planes for billboards. Initial results suggest that our algorithm is faster than the original billboard cloud algorithm by Décoret et al. and that it produces more truthful results. More experiments are needed in order to compare the algorithms for different models and different plane space resolutions.

7 References

[1] B. C. Wünsche, A Survey and Analysis of Common Polygonization Methods & Optimization Techniques, *Machine Graphics & Vision*, 6(4), (1997), 451-486.

[2] M. Levoy and P. Hanrahan, Light Field Rendering, *Computer Graphics (SIGGRAPH '96 Proceedings)*, (1996), 31-42.

[3] X. Décoret, F. Durand, F. X. Sillion, and J. Dorsey, Billboard Clouds for Extreme Model Simplification, *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2003)*, 22(3), (2003) 689-696.

[4] J. A. Hartigan and M. A. Wong, A K-Means Clustering Algorithm, *Applied Statistics*, 28(1), (1979), 100-108.

[5] The Math Forum, Ask Dr. Math – Orthogonal Distance Regression Planes, http://mathforum.org/library/drmath/view/63765.html, (last accessed 2nd September 2004).

[6] E. A. Rakhmanov, E. B. Saff and Y. M. Zhou, Minimal discrete energy on the sphere, *Mathematical Research Letters*, 1(6), (1994) 647-662.

[7] D. A. Katanforoush and M. Shahshahani, Distributing Points on the Sphere, *Experimental Mathmatics*, 12(2), (2003), 199-209.

[8] I.-T. Huang, "Improved Billboard Clouds for Extreme Model Simplification", MSc thesis, Department of Computer Science, University of Auckland, November 2004 (to be published).

[9] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes, Computer Graphics – Principles and Practice, Addison-Wesley, 2nd edition (1992).