Viewpoint Entropy-Driven Simplification

P. Castelló
Universitat Jaume I
Campus Riu Sec
E-12071, Castelló, Spain
castellp@lsi.uji.es

M. Sbert
Universitat de Girona
Campus Montilivi
E-17071, Girona, Spain
mateu@ima.udg.es

M. Chover
Universitat Jaume I
Campus Riu Sec
E-12071, Castelló, Spain
chover@lsi.uji.es

M. Feixas
Universitat de Girona
Campus Montilivi
E-17071, Girona, Spain
feixas@ima.udg.es

ABSTRACT

In this paper, a new viewpoint-based simplification approach is proposed for polygonal meshes. This approach is driven by an information-theoretic metric, viewpoint entropy, which measures the amount of information from a scene or object that arrives at a certain viewpoint. Our algorithm applies the best half-edge collapse as a decimation criterion and uses the variation of viewpoint entropy to measure the edge collapse error. Compared to pure geometric-based simplifications, the models produced by our method are closer to the original model according to visual similarity. Our approach also achieves a higher simplification in hidden interiors, by being able to remove them all and to leave the visible areas of the mesh intact. Models generated by CAD applications can benefit from this feature, since these models are usually constructed by assembling smaller objects which can become partially hidden during joining operations. The main application of our method is for video games where models come from CAD applications and are geometrically not very complex, a few thousand polygons at the most, and in which visual similarity is the most important requirement.

Keywords

Simplification, level-of-detail, viewpoint selection, Information Theory.

1. INTRODUCTION

Most common simplification methods use some technique based on a geometric distance as a quality measure between an original mesh and the one obtained from simplification. With these methods we can achieve meshes that are very similar to the original. On the other hand, image-based simplification methods carry out a simplification guided by differences between images more than by geometric distances. That is, their goal is to create simplified meshes that appear similar according to visual criteria. Thus, the applications that can benefit from image-based methods are those in which the main requirement is visual similarity. Examples of such applications are video games, vehicle simulations, walk-throughs, etc. A reduced number of applications, however, require exact geometric tolerances with regard to the original model. For this type of applications it would be better to consider some simplification method based on a pure geometric measure. Examples of such applications include collision detection and path planning for part insertion and removal.

Geometric methods are suitable for scanned models, which are composed of thousands of polygons. However, in video games models usually come from CAD applications and it is very useful to simplify models that are not very complex to a lower level, typically a few hundred polygons. This is where our approach could be taken into consideration.

In this paper, we introduce a new viewpoint-based simplification method which uses viewpoint entropy [Vaz01], a measure of the geometric information of a scene or object seen from a certain point of view. This method uses the best half-edge collapse as a decimation criterion and measures the variation of viewpoint entropy to quantify the cost of collapse. Experimental results show our method yields better visual performance than QSlim-based simplifications [Gar97]. Our algorithm also offers very good results even at early stages of simplification, where achieves a higher simplification in hidden interiors.

This paper is organized as follows. In Section 2, we survey the previous work and basic information-theoretic measures. In Section 3, we define the simplification error metric based on viewpoint entropy to measure the cost of an edge collapse. In Section 4, we describe our simplification algorithm. In Section 5, we show the results of our experiments and finally, in Section 6, conclusions and future work are presented.

2. BACKGROUND

In this section we review related work, basic information-theoretic measures and information-theoretic viewpoint selection measures.

Related Work

The most important improvement in geometry-oriented simplification methods in recent years was the incorporation of mesh attributes such as color, normals and textures. For example, Hoppe extended his initial work [Hop96] by proposing a new quadric metric that incorporates colors and texture coordinates [Hop99], and the QSlim algorithm [Gar97] was also extended with those attributes in [Gar98]. Cohen et al. [Coh98] developed an algorithm based on edge collapses that samples the vertex position, normal and color attributes of the original mesh and then converts them to normal and texture maps. This algorithm is based on a texture deviation metric.

Lindstrom et al. [Lin00] addressed the problem of visual similarity by developing a pure image-based metric. Basically, their method determines the cost of an edge collapse operation by rendering the model from several viewpoints. The algorithm compares the rendered images to the original ones and adds the mean-square error in luminance across all pixels of all images. Then it sorts all edges by the total error induced in the images and chooses the edge collapse that produces the least error. They used 20 viewpoints in their implementation to compute that error. The main advantage of their method is that the metric provides a natural way to balance the geometric and shading properties without requiring the user to perform an arbitrary weighting of these attributes. On the other hand, its main disadvantage is the low speed it achieves.

Luebke et al. [Lue01] presented a method to perform a view-dependent polygonal simplification using perceptual metrics. These metrics derive from a measure of low-level perceptibility of visual stimuli in humans. Later on, Williams et al. [Wil03] extended this work for lit and textured meshes. Zhang et al.[Zha02] proposed a new algorithm that takes visibility into account. Their approach defined a visibility function between the surfaces of a model and a surrounding sphere of cameras. The number of cameras increases both accuracy and calculation time. They used up to 258 cameras. In order to guide the simplification process, they combined their visibility measure with the quadric measure introduced by Garland et al. [Gar97].

Recently, Lee et al. [Lee05] introduced the idea of mesh saliency as a measure of regional importance for graphics meshes. This measure was incorporated into mesh simplification. Basically, their approach consists in generating a saliency map, and then simplifying by using this map in the QSlim algorithm as in [Zha02]. The new edge collapse cost is that of the quadric times the saliency of this edge.

Information-Theoretic Measures

Let $\mathcal{X}$ be a finite set, let X be a random variable taking values x in $\mathcal{X}$ with distribution $p(x)=Pr[X=x]$. Likewise, let Y be a random variable taking values y in $\mathcal{Y}$. The *Shannon entropy* $H(X)$ of a random variable X is defined by

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x). \tag{1}$$

This is also denoted by $H(p)$ and measures the average uncertainty of a random variable X. All logarithms are base 2 and entropy is expressed in bits. The convention that $0 \log 0 = 0$ is used.

The *mutual information* (*MI*) between X and Y is defined by

$$I(X,Y) = \sum_{x \in \mathcal{X}} p(x) \sum_{y \in \mathcal{Y}} p(y \mid x) \log \frac{p(y \mid x)}{p(y)}. \tag{2}$$

This is a measure of the information shared by X and Y. It can be seen that $I(X,Y)=I(Y,X)\geq 0$.

The *relative entropy* or *Kullback-Leibler distance* between two probability distributions p and q defined over the same set is given by

$$KL(p \mid q) = \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)}, \tag{3}$$

where the convention that $0 \log 0 = 0$, $p(x) \log \frac{p(x)}{0} = \infty$ if $p(x)>0$, and $0 \log \frac{0}{0} = \infty$ is used. The relative entropy $KL(p|q)$ is a measure of the inefficiency of assuming that the distribution is q when the true distribution is p [Cov91].

Information-Theoretic Viewpoint Selection Measures

Information-theoretic-based viewpoint selection metrics have been successfully applied in computer graphic areas, such as scene understanding and virtual exploration [Vaz01, Vaz03, Sbe05] and volume visualization [Bor05, Tak05]. In this section, we review the *viewpoint entropy* [Vaz01, Vaz03] and the *viewpoint Kullback-Leibler distance* [Sbe05] which have been used to compute the best viewpoints of a scene. Recently, the *viewpoint mutual information* has been introduced to select the best views in volume rendering [Vio06] and for polygonal meshes [Fei06].

Viewpoint entropy, based on the Shannon entropy (1), has been defined [Vaz01] from the relative area of the projected polygons over the sphere of directions centered at viewpoint *v*. Thus, the viewpoint entropy was defined by

$$H_v = -\sum_{i=0}^{N_f} \frac{a_i}{a_t} \log \frac{a_i}{a_t}, \tag{4}$$

where N_f is the number of polygons in the scene, a_i is the projected area of polygon *i* over the sphere, a_0 represents the projected area of background in open scenes, and $a_t = \sum_{i=0}^{N_f} a_i$ is the total area of the sphere.

The best viewpoint is defined as the one that has maximum entropy.

In [Sbe05], a viewpoint quality measure based on the *Kullback-Leibler distance* (3) has been defined by

$$KL_v = -\sum_{i=1}^{N_f} \frac{a_i}{a_t} \log \frac{\frac{a_i}{a_t}}{\frac{A_i}{A_t}}, \tag{5}$$

where a_i is the projected area of polygon *i*, $a_t = \sum_{i=1}^{N_f} a_i$, A_i is the actual area of polygon *i* and $A_T = \sum_{i=1}^{N_f} A_i$ is the total area of the scene or object. In this case, the background is not taken into account. The minimum value 0 is obtained when the normalized distribution of projected areas is equal to the normalized distribution of actual areas. Thus, selecting high quality views means minimizing KL_v.

3. SIMPLIFICATION ERROR METRIC BASED ON VIEWPOINT ENTROPY

In this section, we present a new error metric based on the viewpoint entropy that can be used to evaluate the cost of an edge collapse and hence to drive the simplification process.

Viewpoint entropy (4) is based on the distribution of polygon areas seen from a viewpoint. Thus, if a polygon is not seen from any point of view its contribution to the formula is zero and the geometry that is hidden will initially be removed.

Given a particular viewpoint, we can consider the following: if a simplification is produced near the silhouette and increases the whole area seen, the overall value of viewpoint entropy will have been changed. So if we want to keep the silhouette of the model we must try to reduce this change. Note that the area of the background is included in the formula as polygon 0. This allows viewpoint entropy to preserve the silhouette better.

Due to the above seen characteristics of viewpoint entropy and the fact that it expresses the accessible information about the object from a given viewpoint, the variation of this measure for each viewpoint can provide us with an error metric to guide the simplification process. Taking into account these facts, the *simplification error metric* is defined by the sum of variations of viewpoint entropy for all viewpoints *V*:

$$c = \sum_{v \in V} \left| H_v - H'_v \right|, \tag{6}$$

where H_v is the viewpoint entropy before an edge collapse and H'_v is the viewpoint entropy afterwards.

With respect to its computation, several techniques have been analyzed in [Cas06]. More specifically, the OpenGL histogram, the hybrid SW-HW histogram and the occlusion query were studied. The best technique was found to be the hybrid SW-HW histogram in current hardware. This technique takes advantage of the PCI Express bus symmetry. A different color is assigned to each polygon and the whole object is sent for rendering. Next, a buffer read operation is performed, and then this buffer is analyzed pixel by pixel to retrieve data about its color. Using an RGBα color codification with a byte value for each channel, up to 256^4 polygons can be calculated with only one single rendering pass. We used this technique during the simplification process.

4. SIMPLIFICATION ALGORITHM

The simplification process, like many other simplification algorithms, is based on the edge collapse operation. However, we use the half-edge collapse operation. According to this, the remaining vertex for an edge collapse *e(u,v)* is vertex *u* or *v* (Figure 1). By using half-edge collapses it is possible to reuse the simplification process in order to generate multiresolution models. These models can use the current hardware in a more efficient way because no new vertices are added to the original model. The main disadvantage is a slight loss of quality of the final mesh, although the complexity of the simplification algorithm is reduced because we do not have to compute the position of the new vertex *v'* resulting from the edge collapse.

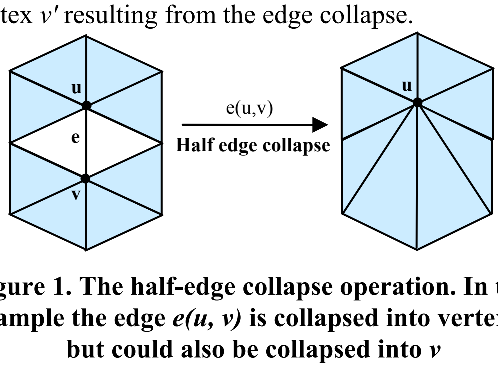


Figure 1. The half-edge collapse operation. In this example the edge *e(u, v)* is collapsed into vertex *u*, but could also be collapsed into *v*

We only take into account the edges that have at most two adjacent polygons, that is to say 2-manifold edges. And we also consider boundary edges, i.e. edges that have one single adjacent polygon.

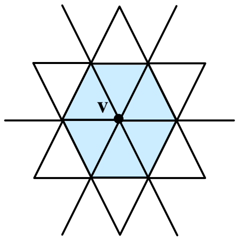


Figure 2. Edges adjacent to vertices adjacent to vertex *v*

This allows us to keep the shape of the mesh better during the simplification process. In addition, we compute the best half-edge collapse. To do this, we use the approach developed by Melax [Mel98], which takes into account polygon normals. We calculate the two possibilities *e(u,v)* and *e(v,u)* and finally we apply the direction that produces a minor change in the curvature of the local region around the edge collapse.

Edge Collapse Error

In the previous section, we defined the error induced by an edge collapse as the sum of the differences in viewpoint entropy before simplifying and after simplifying.

To speed up its calculation, we can make use of the fact that viewpoint entropy can be iteratively calculated. Viewpoint entropy is a calculation from the projected areas, and only a few polygons change after an edge collapse. So viewpoint entropy can be computed at the beginning for the entire object and then this initial viewpoint entropy can be successively updated. In our implementation we have exploited this feature.

We choose the edge collapse that has the least deviation *c* (6). It is important to determine some parameters, since the quality of the results could change. We have performed measurements with 20 regularly distributed viewpoints and rendered 256x256 resolution images. Higher values increase quality, but also significantly raise the temporal cost.

In [Cas05] different hardware techniques were analyzed for geometric visualization using standard OpenGL running on current GPUs. In this study it is shown that the vertex buffer objects technique is the best suited to dynamic geometry. So we used this technique to render our images.

We found that more accurate results are obtained with many viewpoints than with just a few, although the computational cost is obviously higher. At each iteration the edge cost has to be evaluated for the entire set of remaining edges. An edge collapse in our algorithm could, in principle, affect the cost of any remaining edge. But this does not always happen to each edge. At each iteration we only choose a small group of edges that are affected by an edge collapse and then the cost for these edges is recalculated. These edges are the ones that are adjacent to the vertices adjacent to the vertex *v* resulting from a half-edge collapse (Figure 2). In our experiments, if we consider the whole set of edges of the model, the temporal cost is increased around 20 times, but we obtain results that are not significantly better. In Figure 3 we show a summary of this algorithm.

```
/* Compute Viewpoint Entropy for the
original mesh M */
Compute H_v where v={1,..,n}
/* Build initial priority queue of edge
collapses */
for( e ∈ M )
  Perform collapse e
  Compute H'_v where v={1,..,n}
  Compute collapse cost  c = Σ_{v=1}^{n} |H_v − H'_v|
  Insert the duple (e, c) in queue q
  Undo collapse e
end for
/* Update the mesh */
while (queue q not empty)
  Delete from queue q the edge e with lowest
  c
  Perform collapse of e
  Recalculate cost of every edge in the
  neighborhood of the transformation of e
  and update their location in queue q
end while
```

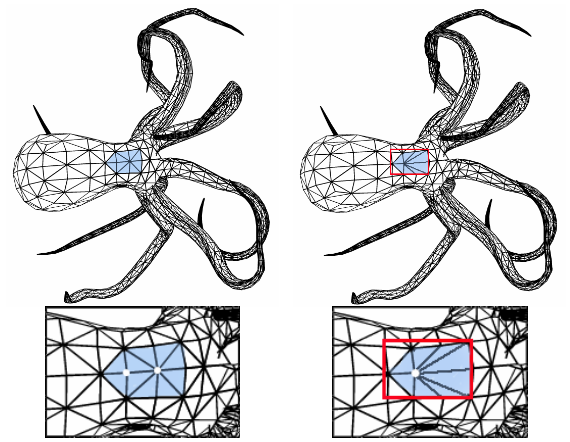
Figure 3. Pseudo-code of our viewpoint entropy-driven simplification algorithm

Updating Projected Areas

In order to compute the entropy of a viewpoint we need the projected areas of every polygon from this viewpoint. The bottleneck resides in the pixel-to-pixel analysis performed by the hybrid SW-HW histogram [Cas06] to obtain these areas, due to the memory transfer cost. Therefore, we can reduce this overload if instead of analyzing the whole image we restrict the area of reading to a window that only includes the polygons surrounding the edge collapse (Figure 4).

To obtain this window, first we determine the bounding box that includes the polygons surrounding the edge collapse and then we project this bounding box onto the screen.

This method allows us to reduce the temporal cost of the algorithm by around 10 times, but it can lead to some slight loss of quality. This is mainly due to the fact that after an edge collapse some hidden polygons may appear and we could not measure their contribution to the formula.



(a) (b)

Figure 4. Image (a) shows the Octopus model, the triangles surrounding the edge collapse are marked in blue. Image (b) shows in red the window which is used to obtain the new projected areas for blue triangles

5. RESULTS

We carried out our tests with low complexity models from CAD programs. All models were simplified from 20 viewpoints on a Pentium Xeon 2GHz with 1GB RAM and an NVIDIA 7800 GTX 512MB graphics card. We compared the results obtained at the same simplification level to the results with QSlim [Gar97], a well-known geometric-based method and freely available, using the best half-edge collapse. The images shown were obtained using different viewpoints from those used during the simplification process.

Model	Triangles		Error	
	Original	Final	H_v	QSlim
Fish	815	100	0,05	0,09
Galleon	4 698	500	0,11	0,22
Fracttree	4 806	1 200	0,08	0,12
Galo	6 592	500	0,03	0,05
Octopus	8 468	500	0,09	0,16
Big_porsche	10 474	1 000	0,04	0,10
Unicycle	13 810	1 000	0,03	0,07

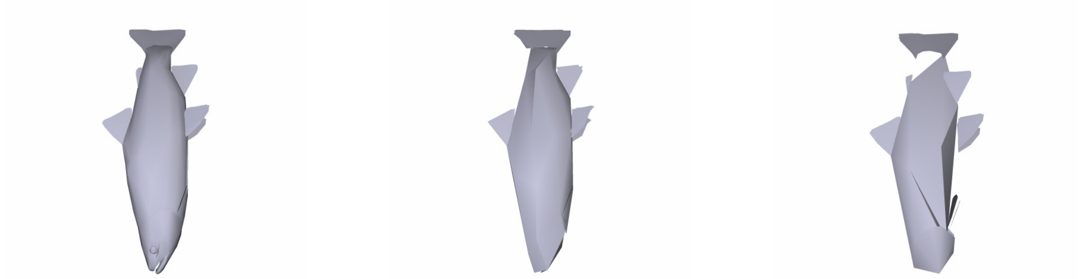
Table 1. Errors measured with Metro for all models

Model	Triangles		Error	
	Original	Final	H_v	QSlim
Fish	815	100	11,40	22,83
Galleon	4 698	500	17,74	36,84
Fracttree	4 806	1 200	30,19	34,10
Galo	6 592	500	9,03	12,40
Octopus	8 468	500	17,35	25,84
Big_porsche	10 474	1 000	7,48	8,28
Unicycle	13 810	1 000	10,32	11,06

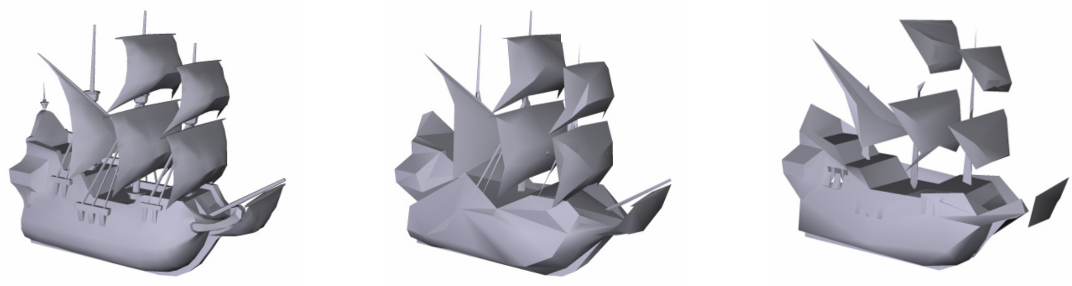
Table 2. Errors measured with RMSE for all models

Model	Triangles		Time	
	Original	Final	H_v	QSlim
Fish	815	100	11.16	0.02
Galleon	4 698	500	92.64	0.06
Fracttree	4 806	1 200	96.30	0.08
Galo	6 592	500	152.29	0.08
Octopus	8 468	500	224.89	0.09
Big_porsche	10 474	1 000	299.47	0.13
Unicycle	13 810	1 000	451.76	0.20

Table 3. Simplification times measured in seconds for all models

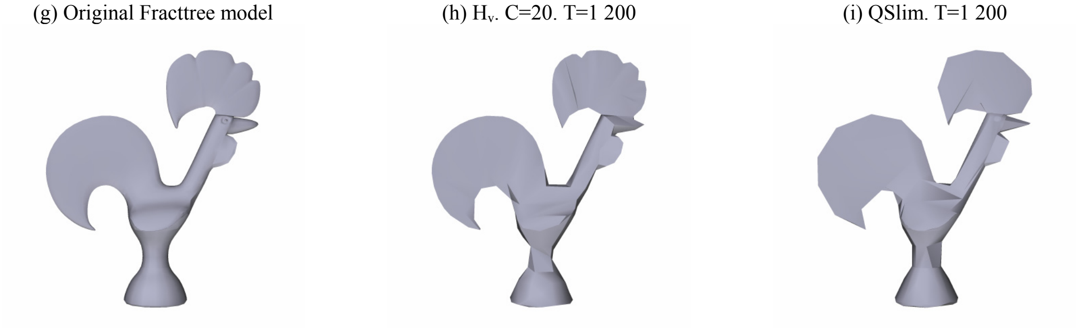


(a) Original Fish model (b) H_v. C=20. T=100 (c) QSlim. T=100

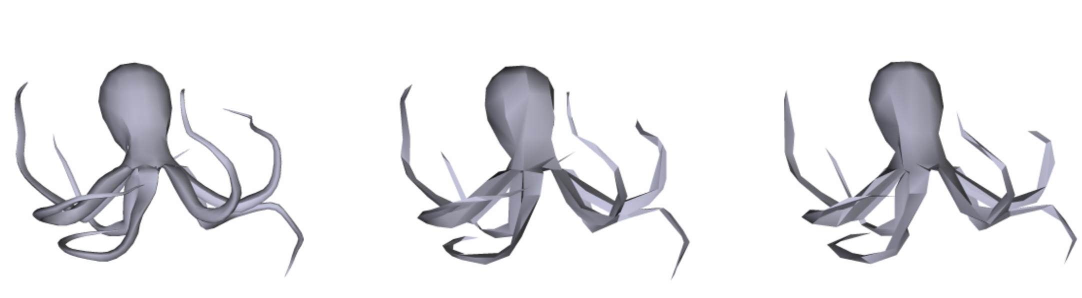


(d) Original Galleon model (e) H_v. C=20. T=500 (f) QSlim. T=500

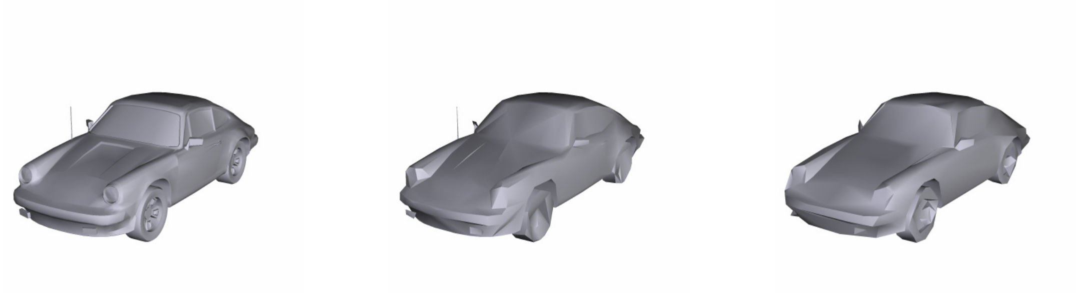
(g) Original Fracttree model | (h) H_v. C=20. T=1 200 | (i) QSlim. T=1 200



(j) Original Galo model | (k) H_v. C=20. T=500 | (l) QSlim. T=500



(m) Original Octopus model | (n) H_v. C=20. T=500 | (o) QSlim. T=500



(p) Original Big_porsche model | (n) H_v. C=20. T=1 000 | (o) QSlim. T=1 000

(s) Original Unicycle model | (n) H_v. C=20. T=1 000 | (o) QSlim. T=1 000

Figure 5. Results for all models. C indicates the number of viewpoints and T the number of triangles

In Table 1 and 2 we present the error committed in our experiments. Table 1 analyzes the visual error and Table 2 shows the geometric error. We have implemented the root mean square error (RMSE) of the pixel-to-pixel image difference defined in [Lin00] to measure the mean visual error between the original and the simplified model. This error was taken using 24 viewpoints and 512x512 resolution images. We must emphasize that each viewpoint was different from the one used during the simplification and the resolution was higher. Clearly, the visual error committed in our method is quite low compared to QSlim, and can even be 50% lower, as shown in the case of the Fish model. We have also measured the geometric error using the mesh comparison tool called Metro [Cig98], and our results are rather better than the geometric method used for comparison purposes. This makes us highly confident about our approach. For example, the geometric error committed in the Galleon, Big_porsche and Unicycle models using H_v are 50% less than with QSlim.

In Table 3 we show an analysis of the temporal cost of our method. This cost is proportional to the complexity of the model and the final number of triangles demanded. However, the QSlim algorithm is extremely fast. Its times for these models are less than a second.

In Figure 5 we show the results for all the models analyzed. The H_v achieves much better simplification than QSlim. For example, in the Fish model the tail and the mouth shape is kept better, and in the Galleon model the same can be said for the sails and the masts. In the Fracttree model there are more branches, while in the Galo model the crest and the tail, in the Octopus model the tentacles, in the Big_porsche model the headlights and the aerial and, finally, in the Unicycle model the spokes are all far better represented.

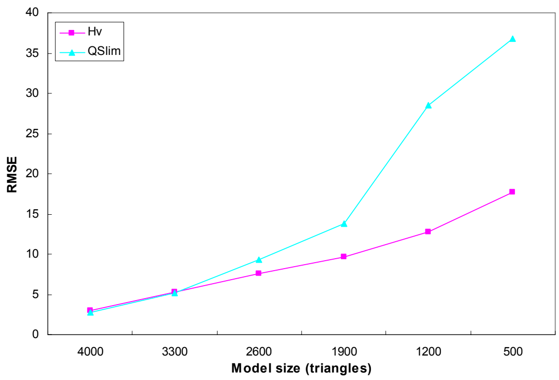


Figure 6. RMSE errors for the Galleon model

In Figures 6 and 7 we show how H_v acts at several degrees of simplification for the Galleon model. We have measured the RMSE and the geometric error. In Figure 6 we show that as we increase the level of simplification the difference between H_v and QSlim becomes larger and the visual quality of H_v is much higher. In Figure 7 we show that the geometric error of H_v is also lower than QSlim, except during the very first stages. This can be accounted for by the fact that H_v is a global measure, and it is possible that, in these stages, QSlim could often be better because it evaluates the error locally.

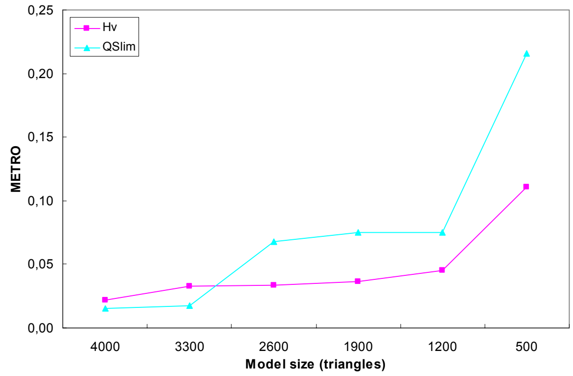
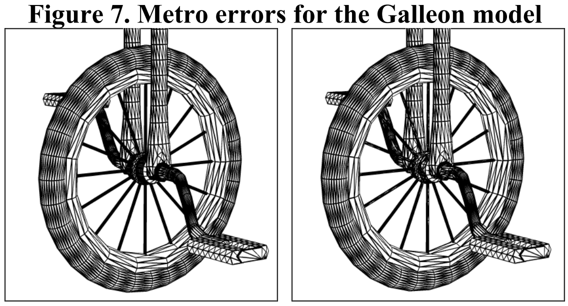
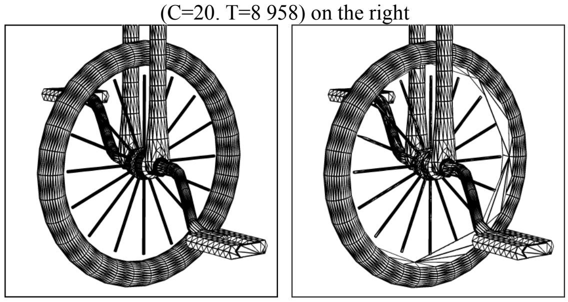


Figure 7. Metro errors for the Galleon model



(a) Original Unicycle model on the left and simplified with H_v (C=20. T=8 958) on the right



(b) Original Unicycle model on the left and simplified with H_v (C=20. T=8 958) on the right, both without rim

Figure 8. Close-ups of Unicycle model simplified with H_v at very early degree of simplification

Finally, in Figure 8 we show how the H_v acts at very early simplification levels for the Unicycle model. In this case we analyze the Unicycle model since it presents hidden interiors on the inner part of the tire which is in contact with the rim. As shown in this figure, H_v achieves a great level of simplification in this region (see 10(b) on the right). The model is simplified by around 35% and is visually the same. At this level, most simplifications focused on hidden interiors.

We also have conducted some experiments with more viewpoints and our results slightly improve but the temporal cost increases substantially. Therefore, we think that with 20 viewpoints we already have excellent results, and thus it is a good compromise between quality and efficiency.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a new mesh simplification method based on viewpoint entropy. Our method performs a simplification with lower visual and geometric error than QSlim. In addition, it achieves very good results with CAD models. These models are composed of different pieces that are assembled together, thus presenting a lot of hidden zones and this is where our algorithm hits harder. In general, the main drawback of image-based methods is the high temporal cost. Our approach, based on viewpoints, also has a high cost compared to geometric-based simplifications. However, we have shown that our method achieves better simplifications by taking into account visual similarity and even it improves geometric error in most cases.

Finally, it could be very interesting to make a study with other metrics based on information theory such as Kullback-Leibler distance and mutual information, which can also be applied to the simplification framework. And it could also be useful to incorporate mesh properties such as color and texture.

7. REFERENCES

[Bor05] Bordoloi U., Shen H.W. Viewpoint evaluation for volume rendering. Proc. of IEEE Visualization 2005, Minneapolis, USA, May 2005.

[Coh98] Cohen J., Olano M., Manocha D. Appearance preserving simplification. Proc. SIGGRAPH '98 (1998), vol. 32, pp. 115-122.

[Cas05] Castelló P., Ramos J.F., Chover M. A comparative study of acceleration techniques for geometric visualization. Proc. of ICCS 2005, pp. 240-247.

[Cas06] Castelló P., Sbert M., Chover M., Feixas M. Techniques for computing viewpoint entropy of a 3D scene. Proc. of ICCS 2006, pp. 263-270.

[Cig98] Cignoni P., Rocchini C., Scopigno R. Metro: measuring error on simplified surfaces. Computer Graphics Forum, vol. 17(2), pp. 267-274, June 1998.

[Cov91] Cover T.M., Thomas J.A. Elements of Information Theory. Wiley Series in Telecommunications, 1991.

[Fei99] Feixas M., del Acebo E., Bekaert P., Sbert M. An information theory framework for the analysis of scene complexity. Computer Graphics Forum (Eurographics'99 Proceedings, Milan, Italy), 18(3), pp. 95-106, 1999.

[Fei06] Feixas M., Sbert M., González F. A unified information-theoretic framework for viewpoint selection and mesh saliency. IIiA-06-06-RR, Institut d'Informàtica i Aplicacions, Universitat de Girona (2006).

[Gar97] Garland M., Heckbert P. Simplification using quadric error metrics. Proc. SIGGRAPH '97 (1997), vol. 31, pp. 209-216.

[Gar98] Garland M., Heckbert P. S. Simplifying surfaces with color and texture using quadric error metrics. IEEE Visualization 98, Los Alamitos, CA, USA, 1998, pp. 263-269.

[Hop96] Hoppe H. Progressive meshes. ACM SIGGRAPH 1996. New York, USA, pp. 99-108.

[Hop99] Hoppe H. New quadric metric for simplifying meshes with appearance attributes. IEEE Visualization 1999, October 1999, pp. 59-66.

[Lue01] Luebke D.P., Hallen B. Perceptually-driven simplification for interactive rendering. Proc. of the 12th Eurographics Workshop on Rendering Techniques (London, UK, 2001, pp. 223-234.

[Lin00] Lindstrom P., Turk G. Image-driven simplification. ACM TOG 19, 3 (July 2000), pp. 204-241.

[Lee05] Lee C.H., Varshney A., Jacobs D.W. Mesh saliency. ACM Trans. Graph. 24, 3 (2005), pp. 659-666.

[Mel98] Melax S. A simple, fast, and effective polygon reduction algorithm. Game Developer (Nov. 1998), pp. 44-48.

[Rig05] Rigau J., Feixas M., Sbert M. Shape complexity based on mutual information. IEEE International Conference on Shape Modeling and Applications (2005).

[Sbe05] Sbert M., Plemenos D., Feixas M., Gonzalez F. Viewpoint quality: Measures and applications. Computational Aesthetics in Graphics, Visualization and Imaging (2005).

[Tak05] Takahashi S., Fujishiro I., Takeshima Y., Nishita T. A feature-driven approach to locating optimal viewpoints for volume visualization. Proc. of IEEE Visualization 2005, (at Minneapolis, USA) (2005), pp. 495-502.

[Vio06] Viola I., Feixas M., Sbert M., Gröller E. Importance-driven focus of attention. IEEE Trans. on Visualization and Computer Graphics, Vol. 12, No. 5, September/October 2006

[Vaz01] Vázquez P.P., Feixas M., Sbert M., Heidrich W. Viewpoint selection using viewpoint entropy. Proc. of Vision, Modeling, and Visualization 2001 (Stuttgart, Germany) (2001), pp. 273-280.

[Vaz03] Vázquez P.P., Feixas M., Sbert M., Heidrich W. Automatic view selection using viewpoint entropy and its application to image-based modeling. Computer Graphics Forum (Dec. 2003).

[Wil03] Williams N., Luebke D., Cohen J.D., Kelley M., Schubert B. Perceptually guided simplification of lit, textured meshes. Proceedings of the 2003 ACM SIGGRAPH, New York, USA, pp. 113-121.

[Zha02] Zhang E., Turk G. Visibility-guided simplification. Proc. of IEEE Visualization 2002 (Nov. 2002), vol.31, pp. 267-274.