# Information Theory Tools for Viewpoint Selection, Mesh Saliency and Geometry Simplification

Mateu Sbert, Miquel Feixas, Pascual Castelló and Miguel Chover

**Abstract** In this chapter we review the use of an information channel as a unified framework for viewpoint selection, mesh saliency and geometry simplification. Taking the viewpoint distribution as input and object mesh polygons as output vectors, the channel is given by the projected areas of the polygons over the different viewpoints. From this channel, viewpoint entropy and viewpoint mutual information can be defined in a natural way. Reversing this channel, polygonal mutual information is obtained, which is interpreted as an ambient occlusion-like quantity, and from the variation of this polygonal mutual information mesh saliency is defined. Viewpoint entropy, viewpoint Kullback-Leibler distance, and viewpoint mutual information are then applied to mesh simplification, and shown to compare well with a classical geometrical simplification method.

## 1 Introduction

In computer graphics, several viewpoint quality measures have been applied in areas such as scene understanding [29, 36, 30, 9], scene exploration [15, 33], image-based

Mateu Sbert
Graphics and Imaging Laboratory, Universitat de Girona, Building P4, Campus Montilivi, 17071-Girona, Spain, e-mail: mateu@ima.udg.edu

Miquel Feixas
Graphics and Imaging Laboratory, Universitat de Girona, Building P4, Campus Montilivi, 17071-Girona, Spain, e-mail: feixas@ima.udg.edu

Pascual Castelló
Computer Graphics Group, Universitat Jaume I, Campus Riu Sec, 12071-Castelló, Spain, e-mail: castellp@lsi.uji.es

Miguel Chover
Computer Graphics Group, Universitat Jaume I, Campus Riu Sec, 12071-Castelló, Spain, e-mail: chover@lsi.uji.es

modeling [37], and volume visualization [2, 34, 38]. The basic question underlying the viewpoint selection study and application is “what is a ‘good’ scene viewpoint?” Obviously, this question does not have a simple answer. Palmer et al. [28] and Blanz et al. [1] have presented different experiments demonstrating that observers prefer views (called canonical views) that avoid occlusions and that are off-axis, salient, stable, and with a large number of visible surfaces. In this chapter, the quality of a viewpoint is associated to the idea of representativeness.

Most common polygonal simplification methods use some technique based on a geometric distance as a measure of quality between an original mesh and the one obtained from simplification [16, 17, 10, 11, 7]. With these methods we can achieve meshes that are very similar to the original. In general, one of the most important advantages of geometry-oriented methods is their low temporal cost. This fact makes them suitable for scanned models, since these models are composed of thousands or even millions of polygons. In contrast, image-based simplification methods carry out a simplification guided by differences between images more than by geometric distances. In other words, their goal is to create simplified meshes that appear similar according to visual criteria [24, 20, 26, 39]. These methods present a high temporal cost compared to geometric ones. 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 and walkthroughs.

In this chapter, several viewpoint measures (viewpoint entropy, viewpoint mutual information, and viewpoint Kullback-Leibler distance) [36, 9, 32] that can be defined from an information-theoretic framework are presented to deal with viewpoint selection, mesh saliency, and polygonal simplification.

## 2 Related Work

We briefly summarize some related work that is not based on information theory since information-theoretic measures are analyzed in the next section. In Plemenos and Benayada [29], the quality of a viewpoint $v$ of a scene is computed using a heuristic measure. Polonsky et al. [30] describe a number of different ways to measure the goodness of a view of an object. Sokolov et al. [33] present two different exploration algorithms guided by the total curvature of a visible surface. Based on the investigation on canonical views, Gooch et al. [14] present a new method for constructing images, where the viewpoint is chosen to be off-axis, and Lu et al. [25] obtain the viewing direction from the combination of factors such as saliency, occlusion, stability and familiarity. Lee and Varshney [23] have introduced the saliency as a measure for regional importance for graphics meshes and Kim and Varshney [21] presented a visual-saliency-based operator to enhance selected regions of a volume. Ruiz et al. [31] presented a voxel saliency measure based on the gradient of voxel obscurances.

In recent years, the most important improvement in geometry-oriented simplification methods was the incorporation of mesh attributes such as color, normals and textures [16, 17, 10, 11, 7, 12]. With respect to the image-based simplification methods, Lindstrom et al. [24] was the first to address 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 the pixels of all the images. Then, all edges are sorted by the total error induced in the images and after that the edge collapse that produces the least error is chosen. Lindstrom et al. used 20 viewpoints in their implementation to compute that error. The main advantage of this 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 them. On the other hand, its main disadvantage is the high temporal cost. Karni et al. [20] proposed a metric to capture the visual difference between two approximations, the average of the norm of the geometric distance between models and the norm of the Laplacian difference. Luebke et al. [26] presented a method to perform a view-dependent polygonal simplification using perceptual metrics. Zhang et al. [39] proposed a new algorithm that takes visibility into account. This approach defines a visibility function between the surfaces of a model and a surrounding sphere of cameras.

## 3 Viewpoint Information Channel

In this section, an information channel between a set of viewpoints and the set of polygons of an object is introduced to define a set of viewpoint measures.

A viewpoint selection framework is constructed from an information channel $V \rightarrow Z$ between the random variables $V$ (input) and $Z$ (output), which represent, respectively, a set of viewpoints $\mathscr{V}$ and the set of polygons $\mathscr{Z}$ of an object [9]. This channel, called viewpoint channel, is defined by a conditional probability matrix obtained from the projected areas of polygons at each viewpoint and can be interpreted as an observation channel where the conditional probabilities represent the probability of "seeing" a determined polygon from a given viewpoint. Viewpoints will be indexed by $v$ and polygons by $z$. The capital letters $V$ and $Z$ as arguments of $p(.)$ will be used to denote probability distributions. For instance, while $p(v)$ will denote the probability of a single viewpoint $v$, $p(V)$ will represent the input distribution of the set of viewpoints. The three basic elements of the viewpoint channel are:

- Conditional probability matrix $p(Z|V)$, where each element $p(z|v) = \frac{a_z(v)}{a_t}$ is defined by the normalized projected area of polygon $z$ over the sphere of directions centered at viewpoint $v$ ($a_z(v)$ is the projected area of polygon $z$ at viewpoint $v$ and $a_t$ is the total projected area of all polygons over the sphere of directions). Conditional probabilities fulfil $\sum_{z \in \mathscr{Z}} p(z|v) = 1$. The background is not taken into account but it could be considered as another polygon.

- Input distribution $p(V)$, which represents the probability of selecting each viewpoint, will be obtained from the normalization of the projected area of the object at each viewpoint. This can be interpreted as the probability that a random ray originated at $v$ hits (or "sees") the object. The input distribution can also be interpreted as the importance assigned to each viewpoint $v$.
- Output distribution $p(Z)$, given by

  $$p(z) = \sum_{v \in \mathscr{V}} p(v) p(z|v), \tag{1}$$

  which represents the average projected area of polygon $z$ (i.e., the probability of polygon $z$ to be hit (or "seen") by a random ray cast from the viewpoint sphere).

From the previous definitions, the viewpoint entropy [36] and viewpoint mutual information [9] can be defined. The viewpoint entropy (VE) of viewpoint $v$ is defined by

$$H(Z|v) = -\sum_{z \in \mathscr{Z}} p(z|v) \log p(z|v). \tag{2}$$

VE measures the degree of uniformity of the projected area distribution at viewpoint $v$. The maximum viewpoint entropy is obtained when a certain viewpoint can see all the polygons with the same projected area. The best viewpoint is defined as the one that has maximum VE [36]. The conditional entropy $H(Z|V)$ of the viewpoint channel is given by the average of all viewpoint entropies. Both entropies $H(Z|v)$ and $H(Z|V)$ tend to infinity when polygons are infinitely refined. This makes these measures very sensitive to the discretization of the object [9].

The mutual information of channel $V \to Z$, that expresses the degree of dependence or correlation between the set of viewpoints and the object [9], is defined by

$$I(V;Z) = \sum_{v \in \mathscr{V}} p(v) \sum_{z \in \mathscr{Z}} p(z|v) \log \frac{p(z|v)}{p(z)} = \sum_{v \in \mathscr{V}} p(v) I(v;Z), \tag{3}$$

where $I(v;Z)$ is the viewpoint mutual information (VMI) given by

$$I(v;Z) = \sum_{z \in \mathscr{Z}} p(z|v) \log \frac{p(z|v)}{p(z)}. \tag{4}$$

VMI gives us the degree of dependence between the viewpoint $v$ and the set of polygons, and it is interpreted as a measure of the quality of viewpoint $v$. Consequently, mutual information $I(V;Z)$ gives us the average quality of the set of viewpoints. Quality is considered here equivalent to representativeness. In this context, the best viewpoint is defined as the one that has minimum VMI. High values of the measure mean a high dependence between viewpoint $v$ and the object, indicating a highly coupled view (for instance, between the viewpoint and a small number of polygons with low average visibility). On the other hand, the lowest values correspond to the most representative or relevant views, showing the maximum possible number of polygons in a balanced way.

In Viola et al. [38], it has been shown that the main advantage of VMI over VE is its robustness to deal with any type of discretisation or resolution of the volumetric dataset. The same advantage can be observed for polygonal data. Thus, while a highly refined mesh will attract the attention of VE, VMI will be almost insensitive to changes in the mesh resolution [8, 9]. In the volume rendering field, Bordoloi et al. [2] and Takahashi et al. [34] used an extended version of viewpoint entropy and Viola et al. [38] introduced the viewpoint mutual information.

The behavior of VMI (4) is now compared with the one of VE (2). To compute these viewpoint quality measures, we need a preprocess step to estimate the projected area of the visible polygons of the object at each viewpoint. In all the experiments, the objects are centered in a sphere of 642 viewpoints built from the recursive discretisation of an icosahedron and the camera is looking at the center of this sphere. Note that all the measures analyzed here are sensitive to the relative size of the viewpoint sphere with respect to the object. The viewpoint sphere is built in the following way: first, the smallest bounding sphere of the model is obtained and, then, the viewpoint sphere adopts the same center as the bounding sphere and a radius three times the radius of the bounding sphere. To show the behavior of the measures analyzed, the sphere of viewpoints is represented by a color map, where red and blue colors correspond respectively to the best and worst views. Note that a good viewpoint corresponds to a high value for VE and to a low value for VMI.

Figure 1 shows the behavior of VE and VMI measures. Columns (a) and (b) show, respectively, the best and worst views, and columns (c) and (d) show two different projections of the viewpoint spheres. Observe how VE chooses to "see" the most highly discretised parts of the cow, such as head and legs. While the worst views for the VE measure correspond to the ones that see the less discretised parts, in the VMI case a true restricted view is obtained [9].

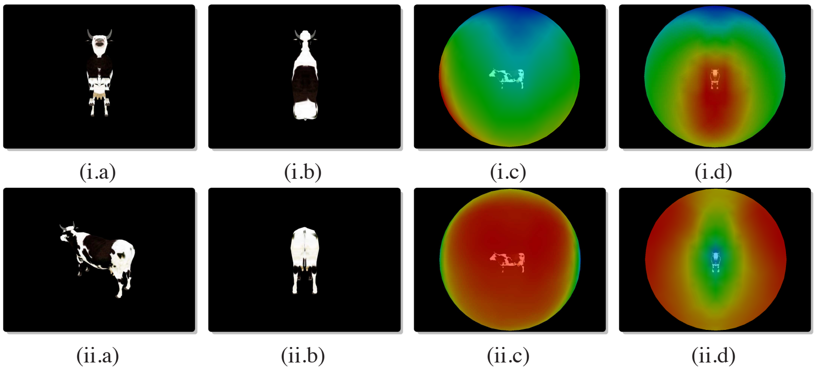


**Fig. 1** (a) The most representative and (b) the most restricted views, and (c-d) the viewpoint spheres obtained respectively from the (i) VE and (ii) VMI measures. Red colors on the sphere represent the highest quality viewpoints and blue colors represent the lowest quality viewpoints.

Another information-theoretic measure, the viewpoint Kullback-Leibler distance, can also be introduced [32]. The viewpoint Kullback-Leibler distance (VKL) of viewpoint $v$ is defined by

$$D_{KL}(p(Z|v),a(Z)) = \sum_{z\in\mathscr{Z}} p(z|v)\log\frac{p(z|v)}{a(z)}, \tag{5}$$

where $a(z)$ is the normalized area of polygon $z$ obtained from the area of polygon $z$ divided by the total area of the object.

The VKL measure is interpreted as the distance between the normalized distribution of projected areas and the "ideal" projection, given by the normalized distribution of the actual areas. Note that, in this case, the background could not be 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 views of high representativeness means to minimize VKL. Note the difference between VMI (4) and VKL (5), due to the fact that in the last case the distance is taken with respect to the actual areas.

The different behavior between VKL and VMI is shown in Figure 2. As we have noted, the main difference between VMI and VKL is that while the former computes the distance between the projected areas of the polygons and the average area "seen" by the set of viewpoints, the later calculates the distance with respect to the actual areas of polygons. Due to this fact, the behavior of VKL is outstandingly affected by the existence of many non visible or poorly visible polygons, as in the case of the lady of Elche model.

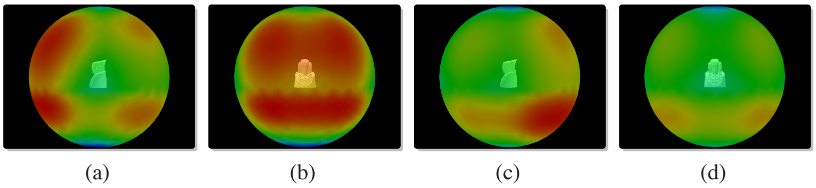


**Fig. 2** Viewpoint spheres obtained respectively from the (a-b) VKL and (c-d) VMI measures.

## 4 Best View Selection

In order to understand or model an object, we are interested in selecting a set of representative views which provides a complete representation of the object. In this section, a VMI-based algorithm is applied to the selection of the $N$ best representative views. This algorithm can also be applied to object exploration [9]. Other algorithms of viewpoint selection and scene exploration based on viewpoint entropy can be seen in [36, 35, 37, 15].

With the goal of obtaining the best representation of the object using the minimum number of views, a viewpoint selection algorithm based on VMI is presented. If we look for a good set of views within the set of viewpoints, we will obtain the most representative set by selecting the views such that their mixing (merging) minimizes the VMI, that is, the distance between $p(z|v)$ and the target distribution $p(Z)$. This mixing provide us with a balanced view of the object.

Thus, the algorithm proposed should select the $N$ viewpoints so that their merging $\widehat{v}$ minimizes the viewpoint mutual information $I(\widehat{v};Z)$. Due to the fact that this optimization algorithm is NP-complete, a greedy strategy is adopted by selecting successive viewpoints that minimize $I(\widehat{v};Z)$. This algorithm permits us to find in an automated and efficient way the minimal set of views which represent the object or scene.

The algorithm proceeds as follows. First, we select the best viewpoint $v_1$ with distribution $p(Z|v_1)$ corresponding to the minimum $I(v;Z)$. Next, we select $v_2$ such that the mixed distribution $\frac{p(v_1)}{p(\widehat{v})}p(Z|v_1)+\frac{p(v_2)}{p(\widehat{v})}p(Z|v_2)$ will minimize $I(\widehat{v};Z)$, where $\widehat{v}$ represents the clustering of $v_1$ and $v_2$ and $p(\widehat{v})=p(v_1)+p(v_2)$. At each step, a new mixed distribution $\frac{p(v_1)}{p(\widehat{v})}p(Z|v_1)+\frac{p(v_2)}{p(\widehat{v})}p(Z|v_2)+\ldots+\frac{p(v_n)}{p(\widehat{v})}p(Z|v_n)$, where $p(\widehat{v})=p(v_1)+p(v_2)+\ldots+p(v_n)$, is produced until the VMI-ratio given by $\frac{I(\widehat{v};Z)}{I(V;Z)}$ is lower than a given threshold or a fixed number of views is achieved. This ratio can be interpreted as a measure of the goodness or representativeness of the selected viewpoints. Figure 3 show the six best views obtained with the VMI-based selection algorithm for two different models. It is important to note that the best views for the selected models (Figure 3(a)) are not the ones our intuition would expect as more representative. This is due to the fact that, from a purely geometric approach, the best views of Figure 3 correspond to the viewpoints that their projected area distribution is more similar to the average projected area distribution (target distribution). This problem can be tackled by introducing perceptual criteria to select the best views (see [9]).

## 5 View-based Polygonal Information and Saliency

As we have seen in Section 3, the information associated with each viewpoint has been obtained from the definition of the channel between the sphere of viewpoints and the polygons of the object. Now, the information associated with a polygon will be defined as the contribution of this polygon to the mutual information of that channel. To illustrate this new approach, the reversed channel $Z \rightarrow V$ is considered, so that $Z$ is now the input and $V$ the output [9, 13].

From the Bayes theorem $p(v,z)=p(v)p(z|v)=p(z)p(v|z)$, the mutual information (3) can be rewritten as

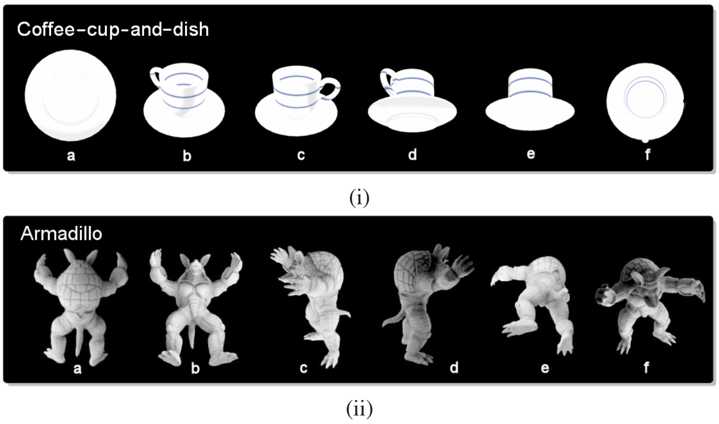


**Fig. 3** From (a) to (f), the six most representative views selected by the VMI-based algorithm for the (i) coffee-cup-and-dish and (ii) armadillo models.

$$\begin{aligned} I(Z;V) &= \sum_{z\in\mathscr{Z}} p(z) \sum_{v\in\mathscr{V}} p(v|z) \log \frac{p(v|z)}{p(v)} \\ &= \sum_{z\in\mathscr{Z}} p(z) I(z;V), \end{aligned} \tag{6}$$

where $I(z;V)$ is the polygonal mutual information (PMI) of polygon $z$, defined by

$$I(z;V) = \sum_{v\in\mathscr{V}} p(v|z) \log \frac{p(v|z)}{p(v)}. \tag{7}$$

PMI represents the degree of correlation between the polygon $z$ and the set of viewpoints, and can be interpreted as the information associated with polygon $z$. Analogous to VMI, low values of PMI correspond to polygons that “see” the maximum number of viewpoints in a balanced way, i.e., $p(V|z)$ is close to $p(V)$. The opposite happens for high values.

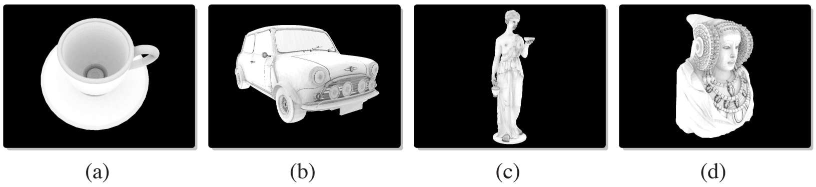


**Fig. 4** View-based polygonal information for the (a) coffee-cup-and-dish, (b) mini, (c) Hebe and (d) lady of Elche models.

In Figure 4, we show the polygonal information maps of (i) the coffee-cup-and-dish, (ii) mini, (iii) Hebe and (iv) lady of Elche models. To obtain these images, the PMI has been normalized between 0 and 1 and subtracted from 1. Thus, low values of PMI, corresponding to non-occluded or visible (from many viewpoints) polygons, are represented by values near 1 in the grey-map, while high values of PMI, corresponding to occluded polygons, are represented by values near 0 in the grey-map. Observe that these maps look as an ambient occlusion or obscurance map (see [22, 6, 40, 18]). For more details see [13].

Itti et al. [19] maintain that visual attention is saliency-dependent and use a saliency map to represent the conspicuity or saliency at every location in the visual field by a scalar quantity and to guide the selection of attended locations. In [23], mesh saliency is captured from surface curvatures and is considered as a perception-inspired measure of regional importance and has been used in graphics applications such as mesh simplification and viewpoint selection.

In our framework, the view-based mesh saliency of a polygon is defined as the average dissimilarity between this polygon and its neighbors, where the dissimilarity between two polygons is obtained from the variation of mutual information when two polygons are clustered [9]. The polygonal dissimilarity between $z_i$ and $z_j$ is defined by

$$D(z_i, z_j) = JS\left(\frac{p(z_i)}{p(\hat{z})}, \frac{p(z_j)}{p(\hat{z})}; p(V|z_i), p(V|z_j)\right), \tag{8}$$

where the second term is the Jensen-Shannon divergence [1] between $p(V|z_i)$ and $p(V|z_j)$ with weights $\frac{p(z_i)}{p(\hat{z})}$ and $\frac{p(z_j)}{p(\hat{z})}$, respectively. Hence, two polygons are “similar” when the JS-divergence between them is small.

The saliency of a polygon $z_i$ is defined by

$$S(z_i) = \frac{1}{N_o} \sum_{j=1}^{N_o} D(z_i, z_j) \geq 0, \tag{10}$$

where $z_j$ is a neighbor polygon of $z_i$ and $N_o$ is the number of neighbor polygons of $z_i$. A polygon $z$ will be salient if the average of JS-divergences between $z$ and its neighbors is high. On the other hand, a polygon at the center of a smooth region will have probably low saliency since the polygons of this region will present small visibility differences with respect to the set of viewpoints.

[1] The Jensen-Shannon (JS) divergence is defined by

$$JS(\pi_1, \pi_2, \ldots, \pi_n; p_1, p_2, \ldots, p_n) = H\left(\sum_{i=1}^{n} \pi_i p_i\right) - \sum_{i=1}^{n} \pi_i H(p_i), \tag{9}$$

where $H(.)$ is the entropy of a probability distribution, and $p_1, p_2, \ldots, p_n$ are a set of probability distributions defined over the same alphabet with prior probabilities or weights $\pi_1, \pi_2, \ldots, \pi_n$, fulfilling $\sum_{i=1}^{n} \pi_i = 1$. The JS-divergence measures how “far” are the probabilities $p_i$ from their mixing distribution $\sum_{i=1}^{n} \pi_i p_i$, and equals zero if and only if all the $p_i$ are equal [3].

Figure 5 shows the behavior of the saliency measure. The most salient parts are represented in red and the least salient ones in blue. For instance, the handle of the coffee cup and the nose, mouth and eyes of the other models are the most salient surfaces.

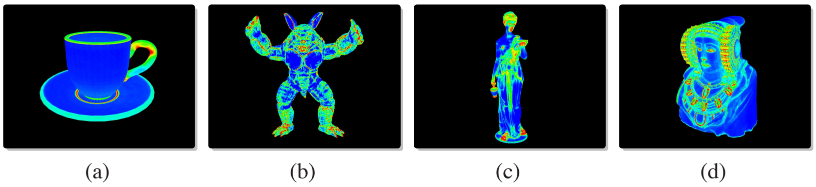


**Fig. 5** Mesh saliency for the (a) coffee-cup-and-dish, (b) armadillo, (c) Hebe, and (d) lady of Elche models.

Similarly to Lee et al. [23], where mesh saliency was used to select the best views, a method to calculate the saliency of a viewpoint is now proposed. Up to now we have calculated the saliency of a polygon, however we can convey this information to the sphere of viewpoints, using the conditional probabilities of the inverse channel.

The viewpoint saliency is defined by

$$S(v) = \sum_{z \in \mathscr{Z}} S(z) p(v|z). \tag{11}$$

Figure 6 shows the most and least salient views for the coffee-cup-and-dish and armadillo models. Observe how the most salient views show us the most salient parts of each object.

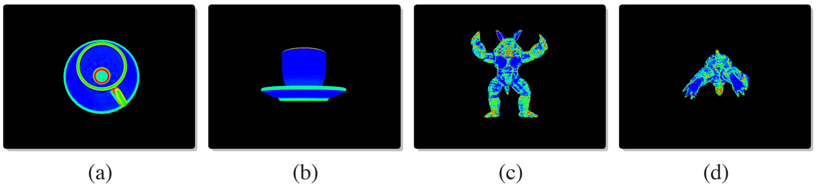


**Fig. 6** The (a,c) most salient and (b,d) least salient views for the coffee-cup-and-dish and armadillo models.

## 6 Viewpoint-Based Error Metric

In this section, an error metric based on viewpoint selection measures is presented for polygonal simplification [4, 5]. This metric can be used to evaluate the cost of a decimation operation. The edge collapse is chosen as the decimation operation, although any other simplification operation could be performed such as removing a vertex, replacing a cluster of vertices by a single one and contracting an edge.

Taking into account that the viewpoint selection measures express the accessible information about an object from a particular viewpoint, the variation of a given viewpoint measure can provide us with an error metric to guide the simplification process. Thus the simplification error deviation for edge collapse $e$ from all viewpoints $V$ is defined by

$$C_e = \sum_{v \in \mathcal{V}} |I_v - I'_v|, \tag{12}$$

where $I_v$ represents the viewpoint selection measure before the edge collapse $e$ and $I'_v$ afterwards.

To test our simplification method, we use VE, VKL, and VMI. As we have seen, these measures are based on the distribution of areas of polygons seen from a viewpoint. The area of the background is included as the polygon number 0. This fact allows us to preserve the silhouette better. Note that the main implication of considering the projected areas is that the hidden geometry will be initially removed, because if a polygon is not seen from any point of view, its simplification will not introduce error.

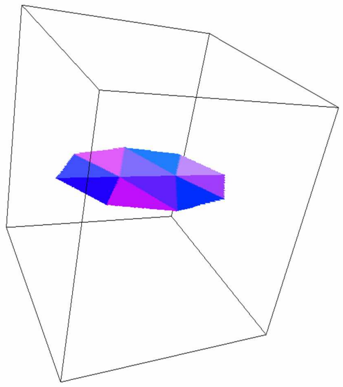


**Fig. 7** An example of 8 camera positions surrounding the Test Model. Camera positions correspond to the 8 vertices of the cube. This allows covering the whole object because the viewpoints are equidistant from each other.

Figure 7 shows the original Test model and how the viewpoints are distributed around it. These viewpoints are associated with the vertices of the Cube in which the object is inscribed. Figure 8 shows the VMI for the original Test model using the 8 viewpoints shown in Figure 7. As can be seen, the different viewpoints have the same VMI. This is because the object is wholly seen from every viewpoint and each viewpoint sees the same as any other. Note that this is a very special situation because the object is quite simple and regular. Normally, in more complex models every viewpoint will have a different VMI.

Figures 9 and 10 illustrate how VMI can be employed to conduct the simplification. Figure 9 shows the Test model after performing the best edge collapse $e$ and Figure 10 after performing the worst edge collapse $e'$. The best edge collapse belongs to the lowest simplification error $C_e$ (12) and the worst to the highest. As can be observed all the VMI values for every viewpoint decreased after an edge collapse (see for instance Figure 8(a) compared to Figure 9(a) or 10(a)). This is because the visible area did not increase in both cases and also the complexity is always reduced during the simplification process. But in a more general case, it is possible that after an edge collapse some previously hidden parts of the mesh may now appear, thus increasing the visible area. If we pay attention to Figure 8(b) and compare this same viewpoint after the best edge collapse (see Figure 9(b)), it can be appreciated that although the number of triangles is reduced (T=8), the visible area remains the same. The simplification error for this viewpoint using VMI is $C_e$=0.004097-0.003651=0.000446. If we analyze the same viewpoint in the worst edge collapse operation (see Figure 10(b)), it can be seen that although the number of triangles is less reduced, the total visible area is somewhat decreased. The simplification error for this viewpoint is $C_{e'}$=0.004097-0.003372=0.000725, which is higher than the error committed in the best edge collapse.

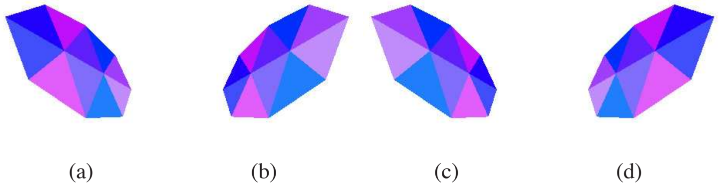


**Fig. 8** Original Test model. T=10. $I(v,O)$=0.004097 where $v = \{1,..,8\}$. Only 4 viewpoints are shown because the rest are symmetric.

Nevertheless, due to the fact that VKL considers the actual area of polygons, after an edge collapse, normally one or two polygons will be removed, thus decreasing the total actual area. This will change the value for VKL after an edge collapse. Therefore the error committed will be distinct from zero. The consequence is that even hidden polygons will have error when simplifying and will not be completely removed during the initial steps of the algorithm. Hidden polygons will be removed according to their actual area. Thus, the smallest polygons will be simplified before, preserving the main features of the object in its internal parts.

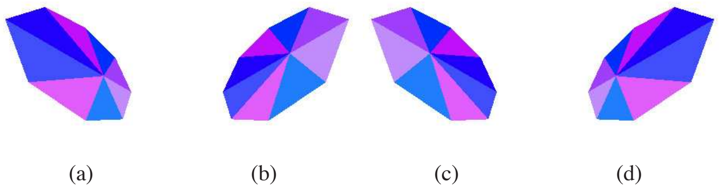


**Fig. 9** Test model after performing the best edge collapse $e$ using VMI. T=8. $C_e$=0.002573.

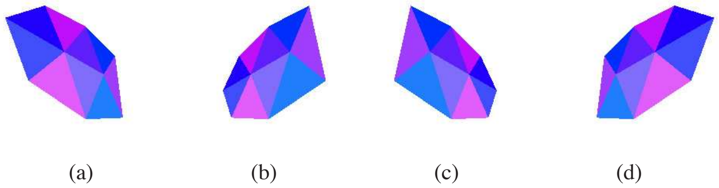
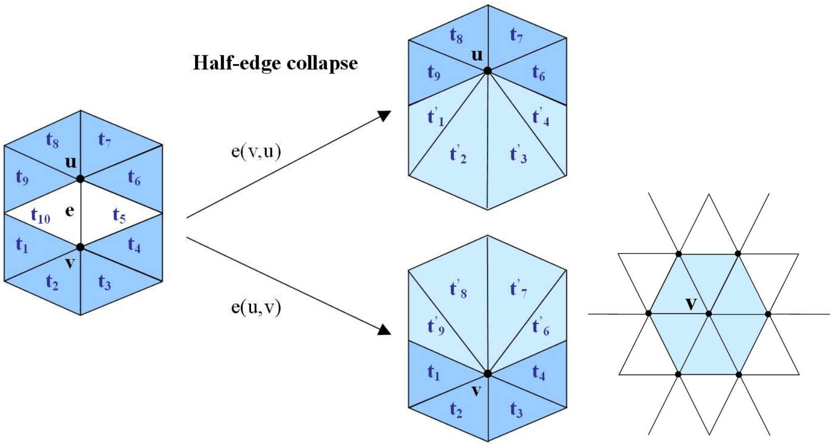


**Fig. 10** Test model after performing the worst edge collapse $e'$ using VMI. T=9. $C_{e'}$=0.006228.

## 7 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 $v$ (see Figure 11(a)). 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. Furthermore, the half-edge representation is useful for progressive transmission. 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. In any case, the general edge collapse operation can be applied to our algorithm. However, a strategy is required to compute the position of the resulting vertex. Brute force selection of edges can introduce mesh inconsistencies. In order to avoid these artifacts, we only take into account the edges which have at most two adjacent polygons, that is, 2-manifold edges. And we also consider boundary edges, i.e. edges which have one single adjacent polygon.

The best half-edge collapse is the decimation operation chosen in our algorithm. Note that the cost of collapsing vertex $u$ to $v$ may be different than the cost of collapsing $v$ to $u$. In our strategy in order to determine the best orientation of an edge collapse, we would have to render the two possibilities and compute that error. However, this would increase considerably the number of renderings and consequently the number of framebuffer readings. Therefore the temporal cost would be penalized. To avoid that, we used the approach developed by Melax [27] that takes into



(a) The half-edge collapse operation

(b) Edges adjacent to vertices adjacent to vertex $v$.

**Fig. 11** In this example edge $e$ is collapsed into vertex $u$ (see $e(v,u)$), but it is also collapsed into $v$ (see $e(u,v)$). Triangles $t_{10}$ and $t_5$ are removed.

account polygon normals. Within this approach, the two orientations $e(u,v)$ and $e(v,u)$ are calculated and finally the orientation that produces a minor change in the curvature of the local region around the edge collapse is applied. Hence the simplification error deviation is only computed for that orientation. In Figure 12 we show a summary of the simplification algorithm.

Both VE and VMI can be calculated iteratively allowing us to speed up their calculation in the algorithm. VE and VMI are computed from the projected areas and the total projected area. The background is considered to be another polygon, and thus the total projected area is always the image resolution. Moreover, only a few polygons change after an edge collapse. Therefore VE and VMI can be computed for the entire object at the beginning and then their initial values can be updated successively. VKL considers the projected areas and one more distribution, the actual area of polygons. In this case, VKL cannot be computed iteratively because the background plays no role, that is, it is not considered as another polygon. The total projected area is always the image resolution. However, the total actual area is not a constant value. This means that after an edge collapse the total actual area will change because some polygons were removed.

The simplification algorithm maintains a heap of edge collapses, sorted by the simplification error cost. In fact, it is an iterative method, so the edge collapse operation is applied until the desired approximation is obtained. At each operation, the edge collapse $e$ that has the least deviation $C_e$ (12) is chosen.

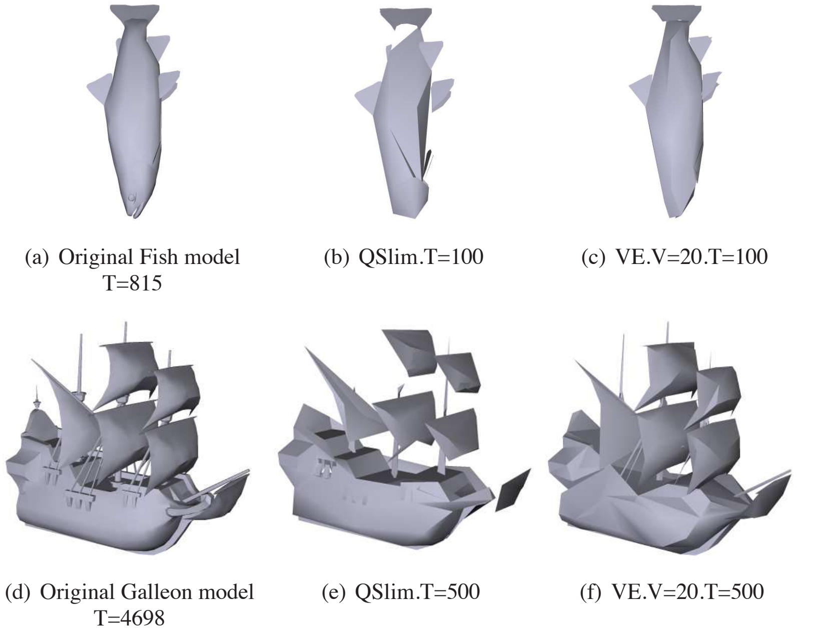
At each iteration, the edge cost must be evaluated for the entire set of remaining edges. An edge collapse in our algorithm could, in principle, affect the cost of any remaining edge. But this case does not always happen to each edge. At each step, we only choose a small group of edges affected by an edge collapse and then the cost is recalculated for this set of edges. These edges are the ones adjacent to the vertices adjacent to the vertex $v$ resulting from a half-edge collapse (see Figure 11(b)). In order to avoid performing unnecessary edge collapse calculations, after applying an edge collapse, each edge that should be recalculated is simply marked as dirty. Such edges are really recalculated only when they reach the top of the heap. If the edge extracted from the heap is dirty, it is simply discarded. Then, its cost is recomputed and inserted into the heap again.

```
// Compute initial viewpoint selection
// measure for mesh $M$
Compute $I_v$, where $v = \{1,..,n\}$

// Build initial heap of edge collapses
for($e \in M$ )
   Choose the best orientation of $e$
   Perform collapse $e$
   Compute $I_v$, where $v = \{1,..,n\}$
   Compute collapse cost $C_e = \sum_{v=1}^{n} |I_v - I'_v|$
   Insert the duple $(e,C_e)$ in heap $h$
   Undo collapse $e$
end for

// Update mesh $M$
while (heap $h$ not empty)
   Delete from heap $h$ edge $e$ with lowest $C_e$
   Perform collapse $e$
   Recalculate cost for the neighborhood of $e$
   and update their location in heap $h$
end while
```

**Fig. 12** Pseudo-code of the viewpoint-based simplification algorithm.

Our experiments were performed with low complexity models from CAD programs. All models were simplified on an Intel Xeon 2.8GHz with 4GB RAM and an NVIDIA 8800 GT 512MB graphics card from 20 viewpoints using 256x256 resolution images. The results obtained with the viewpoint-driven simplification method were compared to the results with QSlim v2.1 [10], using the best half-edge collapse, at the same level of simplification. We chose QSlim because it is a well-known purely geometric algorithm, freely available, which produces high quality simplifications.

We have implemented the root mean square error (RMSE) of the pixel-to-pixel image difference defined in [24] 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.



(a) Original Fish model T=815

(b) QSlim.T=100

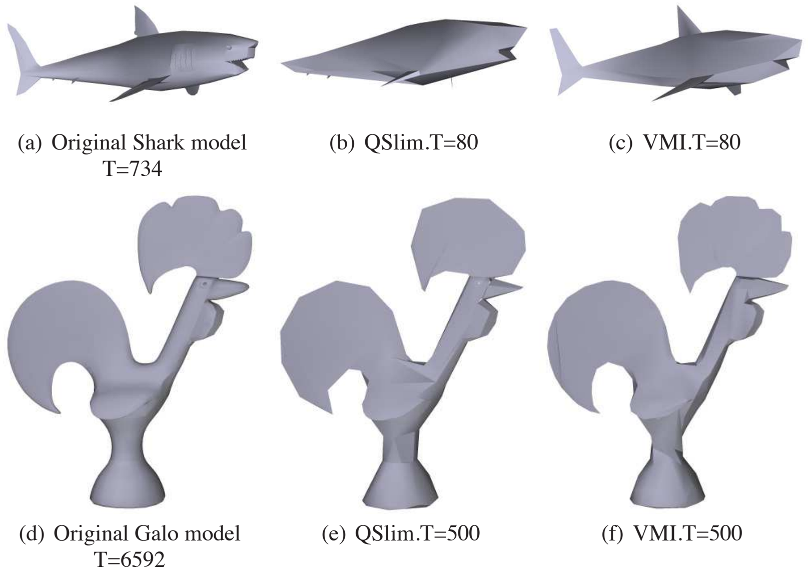
(c) VE.V=20.T=100

(d) Original Galleon model T=4698

(e) QSlim.T=500

(f) VE.V=20.T=500

**Fig. 13** Results for Fish and Galleon models. V indicates the number of viewpoints and T the number of triangles.



(a) Original Shark model T=734

(b) QSlim.T=80

(c) VMI.T=80

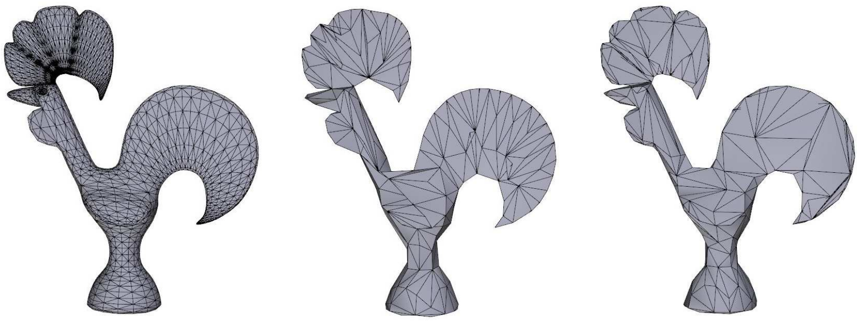
(d) Original Galo model T=6592

(e) QSlim.T=500

(f) VMI.T=500

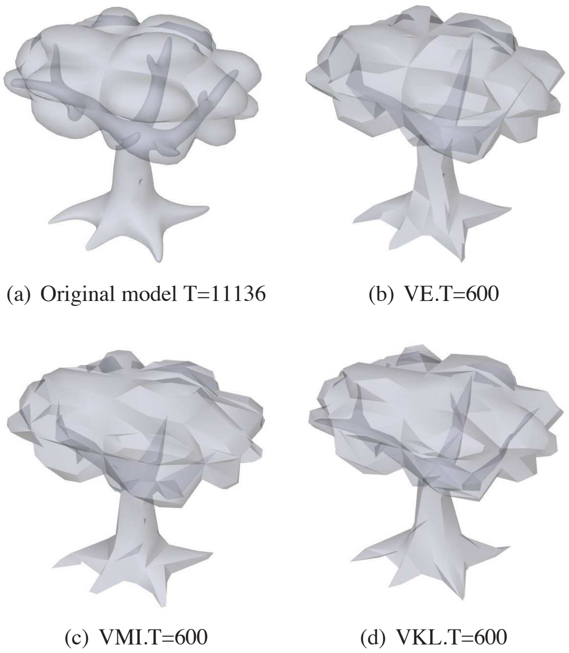
**Fig. 14** Results for Shark and Galo models.

We perform a comparison between QSlim, viewpoint entropy, and mutual information for several models, the results of which appear in Figures 13 and 14 as well as in Table 1.



(a) Original model T=6592 (b) VE.T=500 (c) VMI.T=500

**Fig. 15** Galo model rendered in wireframe.



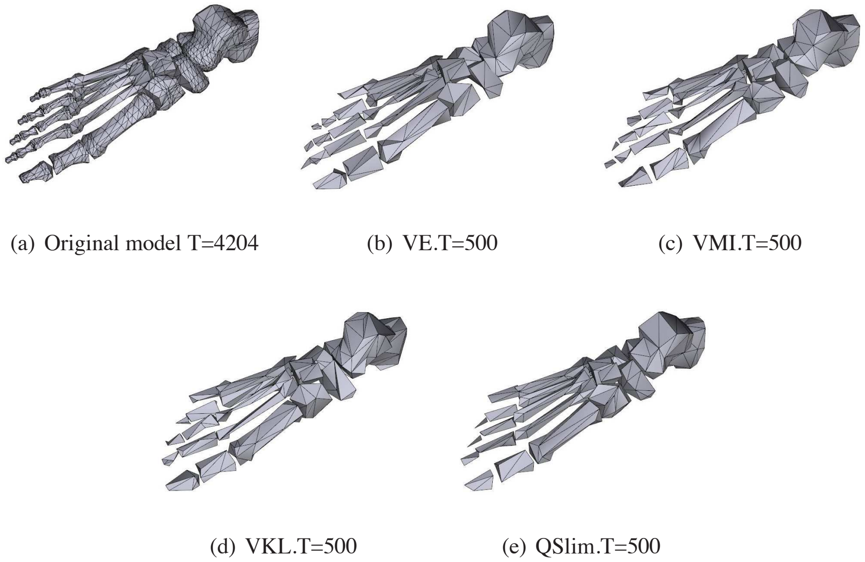
(a) Original model T=11136 (b) VE.T=600

(c) VMI.T=600 (d) VKL.T=600

**Fig. 16** Simpletree model rendered in transparency.

Figure 15 shows how VE and VMI work when they are applied to simplification. VE tends to balance the size of triangles. This means that all triangles in the simplified model will have more or less the same area. However, VMI increases the simplification in flat regions such as the base and the tail. This behavior is desirable and allows VMI to reduce the simplification in other parts of the model, preserving better the visual similarity of the simplified model. In Figure 16, we also show how VE and VMI behave when the object has hidden parts, for instance the Simpletree model which has some branches hidden by the treetop. As can be seen in this figure, both VE and VMI simplify those hidden branches more deeply whereas VKL preserves them better.

Figures 17, 18, and 19 show the results for different models, a more detailed description of these results can also be found in Table 1. For the models analyzed, VE, VMI and VKL achieve much better simplification than QSlim. The difference between both VMI and VE with respect to QSlim is much more significant if the model presents lots of hidden interiors, in which case VMI and VE can accomplish much better simplifications than QSlim and even than VKL, because as we explained previously, VKL preserves better those internal parts. However, this implies a visual error slightly higher in VKL than in VE and VMI.

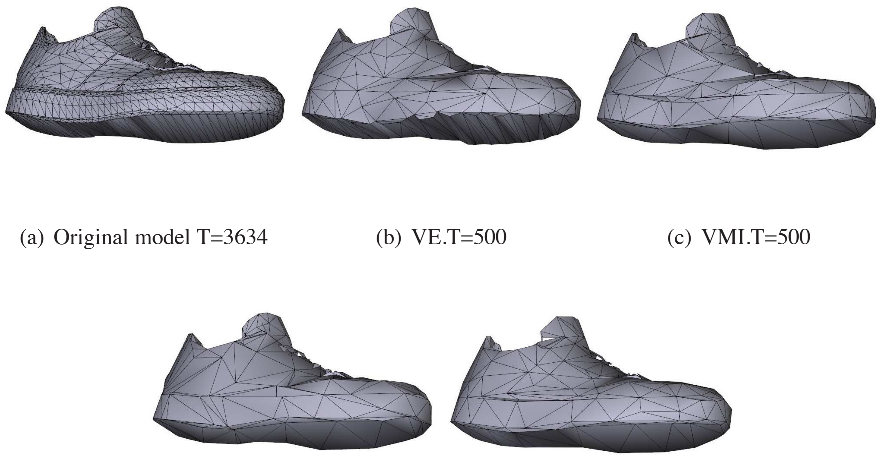


(a) Original model T=4204 (b) VE.T=500 (c) VMI.T=500

(d) VKL.T=500 (e) QSlim.T=500

**Fig. 17** Footbones model.

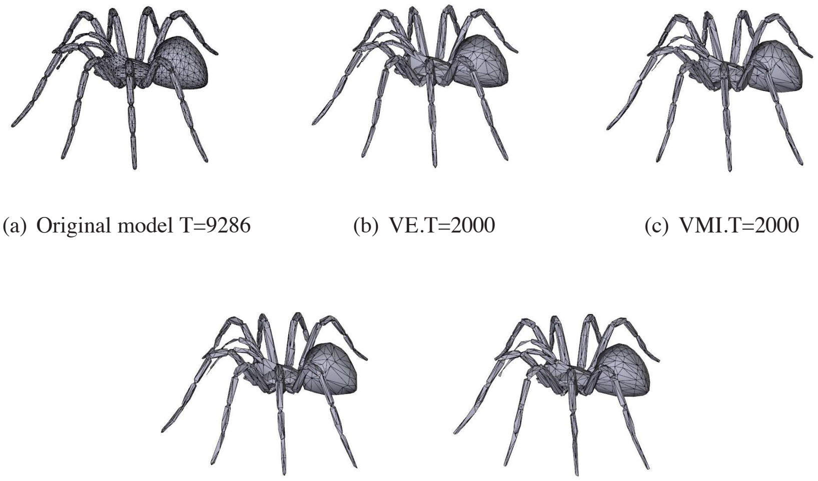
In Table 1, we observe that for all the cases the visual error is much higher for QSlim than for VE, VMI and VKL. As shown in this table the visual error is lower in the VMI case for most models. An analysis of the temporal cost is also shown in this table. This cost is proportional to the complexity of the model and to the final



(a) Original model T=3634 (b) VE.T=500 (c) VMI.T=500

(d) VKL.T=500 (e) QSlim.T=500

**Fig. 18** Tennis_shoe model.



(a) Original model T=9286 (b) VE.T=2000 (c) VMI.T=2000

(d) VKL.T=2000 (e) QSlim.T=2000

**Fig. 19** Big_spider model.

number of triangles demanded. However, the QSlim algorithm is extremely fast. Its times for these models are less than a second. In any case, our method produces high quality simplifications according to visual similarity. The time difference between VMI and VE lies in the calculation performed to obtain the mean projected area of polygons, which is not necessary in VE. Therefore, the temporal cost of VE is a bit lower. The temporal cost of VKL is higher than the temporal cost of VE and VMI because we cannot exploit the iterative calculation for VKL as we carried out in the case of VE and VMI.

**Table 1** Results for QSlim (QS), VE, and VMI measuring visual error (RMSE) and simplification time in seconds.

| Model | Triangles | | RMSE | | | | Time | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Initial | Final | QS | VE | VMI | VKL | QS | VE | VMI | VKL |
| Shark | 734 | 80 | 33.4 | 14.8 | 14.7 | 18.3 | 0.02 | 14.48 | 15.09 | 14.42 |
| Fish | 815 | 100 | 22.9 | 12.1 | 13.9 | 13.8 | 0.03 | 15.71 | 16.67 | 16.54 |
| Tennis_shoe | 3634 | 500 | 9.8 | 8.9 | 8.3 | 8.4 | 0.04 | 83.01 | 95.08 | 145.36 |
| Footbones | 4204 | 500 | 14.5 | 14.3 | 14.5 | 14.3 | 0.04 | 83.65 | 84.11 | 140.87 |
| Galleon | 4698 | 500 | 36.8 | 23.4 | 17.1 | 17.9 | 0.04 | 96.05 | 100.20 | 120.60 |
| Galo | 6592 | 500 | 12.4 | 9.4 | 8.4 | 10.8 | 0.08 | 137.61 | 142.24 | 182.63 |
| Big_spider | 9286 | 2000 | 11.5 | 10.1 | 10.6 | 11.4 | 0.12 | 228.08 | 251.92 | 323.23 |
| Simpletree | 11136 | 600 | 20.7 | 17.2 | 16.6 | 17.8 | 0.20 | 321.06 | 332.49 | 427.24 |

**Acknowledgements** This work has been supported by TIN2007-68066-C04-01 and TIN2007-68066-C04-02 of the Ministry of Education and Science (Spanish Government).

## References

1. V. Blanz, M. Tarr, and H. Bülthoff. What object attributes determine canonical views? *Perception*, 28:575–599, 1999.
2. U. D. Bordoloi and H.-W. Shen. Viewpoint evaluation for volume rendering. In *IEEE Visualization 2005*, pages 487–494, 2005.
3. J. Burbea and C. R. Rao. On the convexity of some divergence measures based on entropy functions. *IEEE Transactions on Information Theory*, 28(3):489–495, May 1982.
4. P. Castelló, M. Sbert, M. Chover, and M. Feixas. Viewpoint-based simplification using f-divergences. *Information Sciences*, 178(11):2375 – 2388, 2008.
5. P. Castelló, M. Sbert, M. Chover, and M. Feixas. Viewpoint-driven simplification using mutual information. *Computers & Graphics*, 32(4):451 – 463, 2008.
6. P. Christensen. Ambient occlusion, image-based illumination and global illumination. *Photorealistic RenderMan Application Notes, Note 35*, 2002.
7. J. Cohen, M. Olano, and D. Manocha. Appearance-preserving simplification. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 115–122, New York, NY, USA, 1998. ACM Press.
8. M. Feixas. *An Information-Theory Framework for the Study of the Complexity of Visibility and Radiosity in a Scene*. PhD thesis, Universitat Politècnica de Catalunya, Barcelona, Spain, Desember 2002.

9. M. Feixas, M. Sbert, and F. González. A unified information-theoretic framework for viewpoint selection and mesh saliency. *ACM Trans. Appl. Percept.*, 6(1):1–23, 2009.
10. M. Garland and P. Heckbert. Surface simplification using quadric error metrics. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 209–216. ACM Press/Addison-Wesley Publishing Co., 1997.
11. M. Garland and P. S. Heckbert. Simplifying surfaces with color and texture using quadric error metrics. In *VIS '98: Proceedings of the conference on Visualization 1998*, pages 263–269, Los Alamitos, CA, USA, 1998. IEEE Computer Society Press.
12. C. González, P. Castelló, and M. Chover. A texture-based metric extension for simplification methods. In *Proc. of GRAPP 2007, Barcelona, Spain*, pages 69–77, 2007.
13. F. González, M. Sbert, and M. Feixas. Viewpoint-based ambient occlusion. *IEEE Computer Graphics and Applications*, 28(2):44–51, 2008.
14. B. Gooch, E. Reinhard, C. Moulding, and P. Shirley. Artistic composition for image creation. In *Rendering Techniques*, pages 83–88, 2001.
15. C. A. Gran, P. P. V. Alcocer, and M. F. González. Way-finder: Guided tours through complex walkthrough models. *Comput. Graph. Forum*, 23(3):499–508, 2004.
16. H. Hoppe. Progressive meshes. *Proceedings of SIGGRAPH 96*, pages 99–108, August 1996. ISBN 0-201-94800-1. Held in New Orleans, Louisiana.
17. H. Hoppe. New quadric metric for simplifying meshes with appearance attributes. In *VIS '99: Proceedings of the 10th IEEE Visualization 1999 Conference*, Washington, DC, USA, 1999. IEEE Computer Society.
18. A. Iones, A. Krupkin, M. Sbert, and S. Zhukov. Fast, realistic lighting for video games. *IEEE Computer Graphics and Applications*, 23(3):54–64, 2003.
19. L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998.
20. Z. Karni and C. Gotsman. Spectral compression of mesh geometry. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 279–286, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.
21. Y. Kim and A. Varshney. Saliency-guided enhancement for volume visualization. *Transactions on Visualization and Computer Graphics*, 12(5):925–932, 2006.
22. H. Landis. Renderman in production. In *Course notes of ACM SIGGRAPH*, 2002.
23. C. H. Lee, A. Varshney, and D. W. Jacobs. Mesh saliency. *ACM Transactions on Graphics*, 24(3):659–666, 2005.
24. P. Lindstrom and G. Turk. Image-driven simplification. *ACM Transaction Graphics*, 19(3):204–241, 2000.
25. A. Lu, R. Maciejewski, and D. S. Ebert. Volume composition using eye tracking data. In *Proceedings of EuroVis 2006*, pages 655–662, 2006.
26. D. P. Luebke and B. Hallen. Perceptually-driven simplification for interactive rendering. In *Proceedings of the 12th Eurographics Workshop on Rendering Techniques*, pages 223–234, London, UK, 2001. Springer-Verlag.
27. S. Melax. A simple, fast, and effective polygon reduction algorithm. *Game Developer*, pages 44–48, Nov. 1998.
28. S. Palmer, E. Rosch, and P. Chase. Canonical perspective and the perception of objects. *Attention and Performance IX*, pages 135–151, 1981.
29. D. Plemenos and M. Benayada. Intelligent display techniques in scene modelling. new techniques to automatically compute good views. In *International Conference GraphiCon'96*, 1996.
30. O. Polonsky, G. Patanè, S. Biasotti, C. Gotsman, and M. Spagnuolo. What's in an image? *The Visual Computer*, 21(8-10):840–847, 2005.
31. M. Ruiz, I. Boada, I. Viola, S. Bruckner, M. Feixas, and M. Sbert. Obscurance-based volume rendering framework. In *Proceedings of IEEE/EG International Symposium on Volume and Point-Based Graphics 2008*, pages 113–120, 2008.

32. M. Sbert, D. Plemenos, M. Feixas, and F. González. Viewpoint quality: Measures and applications. In *Computational Aesthetics 2005 - First Eurographics Workshop on Computational Aesthetics in Graphics, Visualization and Imaging (CAGVI '05)*, pages 185–192, Aire-la-Ville, Switzerland, May 2005. Eurographics Association.
33. D. Sokolov, D. Plemenos, and K. Tamine. Methods and data structures for virtual world exploration. *The Visual Computer*, 22(7):506–516, 2006.
34. S. Takahashi, I. Fujishiro, Y. Takeshima, and T. Nishita. A feature-driven approach to locating optimal viewpoints for volume visualization. In *IEEE Visualization 2005*, pages 495–502, 2005.
35. P. P. Vázquez. *On the Selection of Good Views and its Application to Computer Graphics*. PhD thesis, Universitat Politècnica de Catalunya, April 2003.
36. P. P. Vázquez, M. Feixas, M. Sbert, and W. Heidrich. Viewpoint selection using viewpoint entropy. In T. Ertl, B. Girod, G. Greiner, H. Niemann, and H.-P. Seidel, editors, *Proceedings of Vision, Modeling, and Visualization 2001*, pages 273–280, Stuttgart, Germany, November 2001.
37. P.-P. Vázquez, M. Feixas, M. Sbert, and W. Heidrich. Automatic view selection using viewpoint entropy and its applications to image-based modelling. *Computer Graphics Forum*, 22(4):689–700, 2003.
38. I. Viola, M. Feixas, M. Sbert, and M. E. Gröller. Importance-driven focus of attention. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):933–940, 2006.
39. E. Zhang and G. Turk. Visibility-guided simplification. In *VIS '02: Proceedings of the conference on Visualization 2002*, pages 267–274, Washington, DC, USA, 2002. IEEE Computer Society.
40. S. Zhukov, A. Iones, and G. Kronin. An ambient light illumination model. In *Rendering Techniques*, pages 45–56, 1998.