Constructing Canonical Regions for Fast and Effective View Selection *

Wencheng Wang[1,2] Tianhao Gao[1,2]
[1]State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences
[2]University of Chinese Academy of Sciences
{whn, gaoth}@ios.ac.cn

Abstract

In view selection, little work has been done for optimizing the search process; views must be densely distributed and checked individually. Thus, evaluating poor views wastes much time, and a poor view may even be misidentified as a best one. In this paper, we propose a search strategy by identifying the regions that are very likely to contain best views, referred to as canonical regions. It is by decomposing the model under investigation into meaningful parts, and using the canonical views of these parts to generate canonical regions. Applying existing view selection methods in the canonical regions can not only accelerate the search process but also guarantee the quality of obtained views. As a result, when our canonical regions are used for searching N-best views during comprehensive model analysis, we can attain greater search speed and reduce the number of views required. Experimental results show the effectiveness of our method.

1. Introduction

View selection aims at finding the best views that can watch much information about the model, and so promoting model understanding. The pipeline is by first sampling some viewpoints around the model, generally on a sphere enclosing the model, called *a viewing sphere*, and then evaluating the views of the sampled viewpoints to get the best ones. Although many methods have been proposed to improve view evaluation significantly, not many works study the sampling strategy [23, 27] and their shortcoming of very possibly missing real best views prevents their use, to be discussed in detail in Section 2. As a popular strategy, viewpoints are evenly and densely sampled. Obviously, this is inefficient, because many regions on the viewing sphere are unlikely to contain best viewpoints. More seriously, poor views may even be mis-evaluated as high-quality views due to the unsuitable preference on view evaluation by existing methods, which is also a main reason why it is still a hot topic to study more effective view evaluation methods.

In this paper, we address the challenge of obtaining the regions on the viewing sphere that are very likely to contain best views. Sampling views in these regions can avoid distributing poor candidate views, and so preventing existing methods from producing pseudo best views. Besides, the reduced search space for the best views can speed up view selection. We call these regions as *canonical regions*, as they are related to canonical views for model investigation, to be discussed in the following. Our motivation is based on the study of human representation and processing of visual information, and its role in recognizing models [1, 31, 33]. According to the study, a model is understood by investigating its individual parts and the relations between the parts. Thus, if many parts can be well watched from a viewpoint, the viewpoint is very possible to be a best one. As for watching a part, its canonical views are most preferred because these views are stable and capable of producing more meaningful and understandable images for the viewer [3, 15, 22, 32, 36]. However, due to occlusions between the parts, the canonical views of the part may be hidden. Considering this, we sample some views around the part to approximate the observability of the part from the viewing sphere. Therefore, clustering the sampled views of the parts, we can find the regions that each contain many good views of the parts. Clearly, any a viewpoint in these regions can well watch many parts, because a part can be well watched by the viewpoints in the neighborhood of its good views. These regions are our canonical regions. By the similar reasoning, the viewpoints in the other regions are not very possible to well watch many parts, and so they can be excluded from the best view search.

In constructing canonical regions, it is necessary to divide the model into parts. For this, we resort to the study on semantic segmentation for model understanding. By the study, it is known that human beings prefer to watch a hand-

*Corresponding author: Wencheng Wang. This work is supported in part by National Natural Science Foundation of China (61379087),the Knowledge Innovation Program of the Chinese Academy of Sciences, and the European Unions Seventh Framework Programme (FP7/2007-2013) under grant agreement (612627).

ful of object-like segments in a view, because they are much easier to interpret and process by higher-level visual routines [5]. These findings were also validated by the results of the Princeton Benchmark study on human segmentations [7]. In [17] and [38], algorithms have been proposed to produce decomposed segments that effectively match human perception. As a result, we only need to divide the model into a small number of parts for constructing our canonical regions.

With regard to canonical views, their definition is subjective and not easy to quantify. The general way for computing canonical views is always by user studies [16] and presents numerous difficulties. To efficiently implement our method, we develop an analytic solution to rapidly approximate canonical views. It is by our observation that if a shape and its bounding box do not differ significantly, they would share very similar best views; and if the ratios between the length, width, and height of the box are maintained by the ratios of their respective 2D image projections in a view, the view is the optimal one, because the geometric information of the box are represented as much as possible in this view. These assertions are validated by our user studies. Thus, our solution is to use bounding boxes of the segmented parts to approximately compute their canonical views, to be discussed in Section 3. This would not reduce the quality of the resulted best views, as our aim is to optimize the search space, and the best views are finally determined by applying existing view evaluation methods in the constrained search space.

In our experiments, we can configure every canonical region to occupy less than 5% of the surface of the viewing sphere and obtain very satisfactory results. When only one best view is required, we can effectively get a canonical region for such a search, and get an acceleration of about $6 \sim 12$ times. When N-best views are required, we can perform the search over 40 times faster, as listed in the experimental results.

In sum, we present a new method to optimize view search, which can be integrated with all existing view evaluation methods. Our contributions are in the below.

- Presenting a method to effectively get the regions that are very likely to contain best views.
- View selection is sped up considerably, and the quality of the obtained views is guaranteed.
- The required views for N-best view search can be reduced and obtained much fast.

In the remainder of the paper, related works will be briefly discussed in Section 2. Afterwards, we will first introduce model decomposition and canonical view approximation of boxes in Section 3 and 4, two basic measures for constructing our canonical regions, and then discuss canonical region construction in Section 5 and its application for best view search in Section 6. In section 7, experimental results are discussed, and finally, the conclusion is summarized in Section 8.

2. Related Work

2.1. View selection methods

For evaluating views effectively, many view descriptors have been proposed using the displayed information, including geometric information such as curvatures [2, 23], mesh saliency [18] and surface regions of interest determined by some measurement [19], visual information such as opacity and colors for volume rendering [4, 14], semantic information when the information of interest is not geometric information in some applications [8, 11, 23, 30, 35], and even the information in the model-making procedure [6, 20]. Such information is always measured using information theory to estimate the view quality [9, 25, 27]. In [26], the results of a user study were utilized to optimize the parameters of a general model for viewpoint quality, in which many geometric and semantic attributes are used together. Surveys of them can be found in [23, 24].

Although view evaluation has been improved significantly, not many works study the view search strategy to promote view selection. As introduced in [23], principal component analysis (PCA) and normal clustering are used to improve the search. By PCA, the model is approximated by an oriented box, thus establishing a local Cartesian frame by the three axes of the box. Then the candidate views are sampled as the vectors whose components are the eight combinations of $(\pm1, \pm1, \pm1)$ in this coordinate system. However, as stated in [21], PCA-alignment is performed by solving for the eigen-values of the covariance matrix, which captures only second order model information. When the alignment of higher frequency information is strongly correlated with the alignment of the second order components, such a sampling strategy can achieve good results. Unfortunately, such an assumption is invalidated for many models. As for normal clustering, it is based on the assumption that the more the normal point in some direction, the more the model's surface is visible from that direction. Obviously, this is not very effective for view selection, e.g., the best view for a cuboid will watch only its biggest facet, not its three facets. A strategy used a lot is to perform a coarse sampling to locate potential optimal views and then employ a gradient-descent optimization to search for local optimal views [27]. As the model information does not vary continually with the viewpoint movement, this strategy may miss the best views as the coarse sampling is sparse in general. In [29], it is proposed for volume visualization by decomposing an entire volume into a set of feature component and then finding a globally optimal viewpoint through the compromise between locally optimal viewpoints for the components. It seems like our method. However, it is to use the view selection methods for mesh models to promote volume visualization, where the features are modeled in meshes, and it is still by uniformly distributing viewpoints on the viewing sphere to search for best views. Thus, to guarantee the quality of the obtained views, the popular way in existing methods is to sample viewpoints evenly and densely on the viewing sphere.

Different from existing methods, our method tries to get the regions that are very likely to contain best viewpoints on the viewing sphere. Thus, the search space for best views can be reduced for acceleration, and the quality of the obtained views can be guaranteed by avoiding distributing poor candidate viewpoints to correct the unsuitable preference of view evaluation methods. For example, when the method in [27] via the viewpoint Kullback-Leibler distance for view evaluation is applied in our canonical regions, it's obtained views are more informative, and otherwise, it is not, as illustrated in Figure 1 for watching a cuboid.

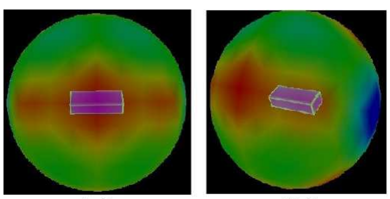


Figure 1. The method in [27] with its sampling strategy for views is not very effective to get the best view with much information. Its best view for watching a cuboid can watch only two facets (a), not the canonical view to watch three facets as we do (b). The colors on the viewing sphere represent the view scores evaluated by the method in [27], where the colors change from red, through green, to blue, corresponding to the view scores decreasing from the highest to the lowest gradually.

2.2. N-best views

N-best views are always required for comprehensive model analysis, owing to the occlusions between the parts of the model. The aim here is to obtain as few high-quality views as possible, to encapsulate all the information of interest about the model [4, 9, 23, 26, 27, 35, 37]. Although the view descriptors used in different methods have significant differences, they always follow the same strategy to search N-best views. First, they sample some viewpoints as seeds, and then iteratively add viewpoints until the model information can be mostly encapsulated by the selected views [27]. For example, the method in [27] is initialized by placing six sampling points on the viewing sphere, with angular coordinates of $(0,0),(\pi/2,0),(-\pi/2,0),(0,\pi/2),(0,-\pi/2),(0,\pi)$, and remaining the sampled points with views that are qualified. Afterward, new points are iteratively generated on the viewing sphere by a binary combination of the obtained points in the set of N-best views. If qualified, these points are added to the set. This process continues until no additional points can be added. Clearly, poor quality seeds will increase the number of required views, and generating many low quality views to evaluate will waste computation time. However, the blindness in the seed distribution process cannot be corrected in existing methods. With our method, such blindness can be avoided, and the canonical regions can optimize the initialization and searching processes, because the views in the canonical regions are very likely to be high quality, and finding views in different canonical regions can expedite the convergence of the search process. As a result, we can obtain significant increases in processing speed, and even reduce the number of required views, as shown in our experiments.

2.3. Segmentation

Model segmentation has been extensively studied, and a survey can be found in [28]. This body of research is an important step towards model understanding. Since detailed features contribute little to model understanding, interest in semantic segmentation for model understanding has increased in recent years; typically, it is used for extracting high-level features containing important information about the model. As discussed in [17, 38], they can produce decomposed segments to effectively and automatically match human perception, not only in the cutting boundaries but also in the number of segments. In our method, we rely on Lai's method [17] to develop a simple method for model decomposition. Though our results are slightly inferior in quality to the results of the methods [17, 38], this will not cause problems for our method, because our construction of canonical regions is an approximation computation, without a very high requirement on decomposition.

3. Segmentation

In using Lai's method [17] for model decomposition, seeds are evenly distributed on the model surface. Subsequently, the seeds are based to classify the facets of the model for segmentation, where some seeds may be automatically classified into the same segment. This results in a segment quantity that is well matched to human perception. Considering human beings prefer to watch a handful of object-like segments in a view, as attested in [5, 7], we generally distribute 15 seeds for segmentation. In this distribution, the seeds are placed as far apart as possible, according to Lai's method for coarse-scale segmentation.

After the seeds are distributed, the decomposition is by the following steps. Firstly, the initial segments are obtained by deciding whether a given face belongs to the region attached to a seed. If a random walk starting at a face has a higher probability of reaching this seed than reaching any other seeds, this face is classified to the region attached to the seed. Afterwards, neighboring segments are investigated to determine if they can be merged according to their similarities, to get the final decomposition. For meaningful decomposition, probability computations are performed to measure the dihedral angle between adjacent faces for graphical models, and measure Gaussian and mean curvatures on both sides of a given edge for engineering models. For details regarding this method, please refer to [17].

As discussed in Section 2, we do not require high-quality segmented parts for constructing our canonical regions. Thus, following the basic idea of Lai's method, we simplify some computations to accelerate the process. This is accomplished by computing only the geodesic distances from the facets to the seeds, instead of the corresponding computation in Lai's method, for classifying facets. As illustrated in Figure 2 for a model with 40K triangles, the left shows the results of Lai's method, and the right shows those of our simplified method, where our seeds are marked with red boxes. Although our segmented results are not high in quality, they can still effectively represent their corresponding model contents, such as the head and legs. As a benefit of this, we can get the segmented results quickly, e.g, only 0.193 second was taken to get our result in Figure 2 on our personal computer during the experiments, while Lai's method needed 2.1 seconds for its result in Figure 2 on a similar computer. Moreover, we need not solve a large matrix as required in Lai's method when treating large models, so that we can easily process large models without a significant storage requirement.

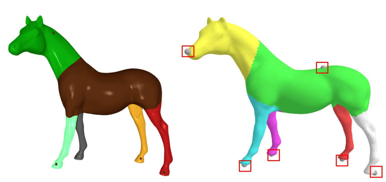


Figure 2. Segmentation of the Horse Model. The left shows the results of Lai's method [17], and the right shows those of our simplified method

4. Canonical View Approximation of Boxes

As we know, the segmented parts are investigated to know the relations between them for model understanding, so that it is more cared for their lower frequency information rather than their higher frequency information in the investigation, which represents their main characteristics, as studied in [13]. Because their bounding boxes via PCA can well represent their lower frequency information, we can use their bounding boxes as their proxies to approximate their canonical views for investigating their main characteristics. For this, we derive an analytical algorithm, which is based on our observation that in the human being's preferred views of a box, the ratios between the length, width, and height of the box are well preserved by their respective projections in the corresponding images. This is attested in our user studies, provided in the supplementary materials.

Our algorithm is by weak perspective projection, as illustrated in Figure 3, instead of perspective projection for computation efficiency [12]. By a simple deduction, we can get a canonical view direction $\vec{\boldsymbol{n}}$ by solving the following equation (1) to have $\left|\vec{\boldsymbol{L}}\right| : \left|\vec{\boldsymbol{W}}\right| : \left|\vec{\boldsymbol{H}}\right| = \left|\vec{\boldsymbol{l}}\right| : \left|\vec{\boldsymbol{w}}\right| : \left|\vec{\boldsymbol{h}}\right|$, where $\vec{\boldsymbol{L}}, \vec{\boldsymbol{W}}$ and $\vec{\boldsymbol{H}}$ are the length, width, and height of the box, $\vec{\boldsymbol{l}}, \vec{\boldsymbol{w}}$ and $\vec{\boldsymbol{h}}$ are its projected length, width, and height in the orthogonal projection plane, and λ is the parameter for normalizing $\vec{\boldsymbol{n}}$.

$$\begin{bmatrix} \vec{\boldsymbol{L}} & \vec{\boldsymbol{W}} & \vec{\boldsymbol{H}} \end{bmatrix}^T \cdot \vec{\boldsymbol{n}} = \lambda \left(\left|\vec{\boldsymbol{L}}\right|, \left|\vec{\boldsymbol{W}}\right|, \left|\vec{\boldsymbol{H}}\right| \right) \tag{1}$$

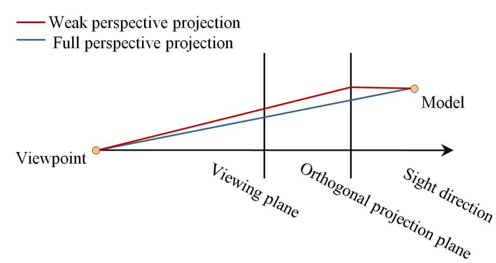


Figure 3. Illustration of weak perspective projection and full perspective projection.

5. Canonical Region Construction

As discussed in Section 1, our canonical region construction is by sampling some views of the bounding boxes of the segmented parts, and making clusters of these views to effectively estimate the regions on the viewing sphere that can well watch many parts. In details, our construction is by the following steps.

1) The model is decomposed into many segmented parts with the measure in Section 3.

2) For every part, its bounding box is built via PCA, and so its canonical views are approximated with the measure in Section 4.

3) For every part, many views are sampled and evaluated their effectiveness for watching the part, and then they are mapped onto their viewpoints on the viewing sphere, where the occluded views are excluded.

4) The viewpoints on the viewing sphere are clustered with every cluster covering a small region.
5) The clusters are ranked by their effectiveness to watch the segmented parts. When a cluster is more effective to watch the parts, it will be ranked ahead.
6) Finally, the regions covered by the clusters that are ranked ahead are detected to be the canonical regions for investigating the model.

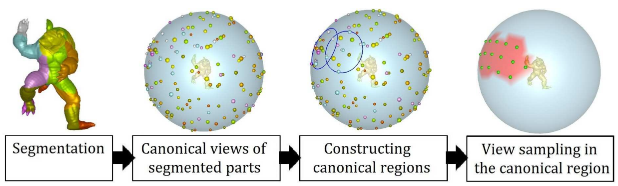


Figure 4. The pipeline for view search with our method applied.

In Figure 4, it is illustrated for constructing canonical regions and sampling views in these regions for best view search.

5.1. View sampling and weighting

In our work, it is necessary to sample some views around a part to estimate its observability from the viewing sphere. For this, we made an investigation by testing three sampling manners, provided in the supplementary materials. As illustrated in Figure 5 via the sampled views on a box face, our adopted manner is by sampling the canonical views in black, the face views in red from the face centers of the box and in the directions vertical to the faces, and the interpolation views in blue from the middle points of the lines connecting the face centers and the box corners, whose directions are interpolated between those of their related face views and canonical views. This is because the obtained clusters of these views can almost cover the viewing sphere surface, enough to find the regions that are very likely to contain best views.

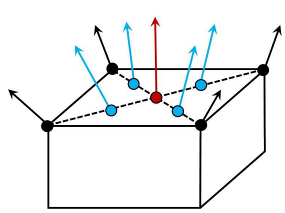


Figure 5. Our view sampling manner on a face of a box.

For the sampled views, we assign weights to them to describe their effectiveness on watching the part. Here, some factors are considered, including the volume of the part, the viewing distance, and the view type, because a larger part may contribute more for model understanding, a nearer part can be investigated more clearly, and canonical views, face views and interpolated views have different effectiveness for watching the part. The weight for view v, $weight(v)$, is computed by the bounding box of the part, and in the following equation (2).

$$weight\,(v) = w_{vol} \times w_{dis} \times w_{type} \quad (2)$$

where $w_{vol} = \sqrt[3]{Vol\,(p)\,/Vols}$ with $Vol\,(p)$ being the volume of the bounding box of the part, and $Vols$ the sum of the volume of the bounding boxes for all the segmented parts. $w_{\mathrm{dis}} = \frac{1}{1+e^{-Z}}$ with $Z = -\frac{R}{2}\left(\frac{1}{d} + \frac{1}{d-2R}\right)$, where d is the distance from the viewpoint on the viewing sphere to the starting point of the view on the bounding box, $d \in [0, 2R]$, and R is the radius of the viewing sphere. w_{type} is computed by its view type. If it is a canonical view, its $w_{type} = 1.0$. If it is a face view, its $w_{type} = \frac{S_{face}}{S_{box}}$, where S_{face} is the area of its related box face, and S_{box} the sum of the areas of the three faces sharing a corner of the box. If it is an interpolation view, its w_{type} is the average of the w_{type} values of its related canonical view and face view.

5.2. Clustering

For the sampled views of the parts, we map them to their viewpoints on the viewing sphere, where hidden views are excluded. Then, we make clusters with k-means clustering, expecting every cluster to contain as many viewpoints for different parts as possible, and in a size as small as possible. For convenience, we implement our solution in an approximate manner by controlling the size of every cluster. Using our tests, satisfactory results can be always obtained by allowing the bounding ellipse for a cluster to occupy less than 5% of the viewing sphere surface. Our clustering algorithm is in the below.

The clustering algorithm

1) As the initial, 8 seeds are distributed on the viewing sphere, which are the 8 canonical viewpoints of the bounding box of the whole model.
2) According to the seeds, the mapped viewpoints of the segment parts are classified into clusters by the distances from them to the seeds.
3) Seeds are updated with the following measures, as illustrated in Figure 6, to perform the above step 2) iteratively until the bounding ellipses for the clusters each have the area near a set threshold, ek. For example, we let $ek = 5\%$ of the area of the viewing sphere surface in our tests.
   - The weighted centers of the clusters are taken to replace their old seeds, computed by their respectively contained viewpoints' location with their view weights.

- When two clusters are much near to each other, say the distance between their centers are shorter than a threshold, e.g. $es = 5\%$ of the radius of the viewing sphere in our tests, these two clusters are merged, and one of their related two seeds is given up randomly.
- When a cluster has its bounding ellipse with a larger area than the threshold ek, we find the viewpoint in the ellipse that is the farthest from the seed of this cluster, and add it as a new seed.

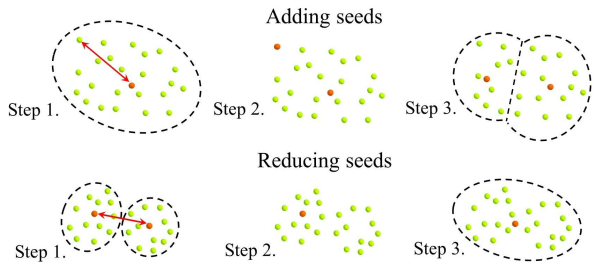


Figure 6. Two operations for seed updating, with the steps (1), (2) and (3) in a sequence. One is for adding seeds, and the other for reducing seeds. Here, the red points are the seeds.

5.3. Ranking

In ranking the clusters, we mainly consider two factors, the effectiveness of the views in a cluster to watch their related segmented parts, and the projection area of the model when viewed from the center of the cluster since larger projection is more helpful for investigating the model in general, and then train a Bayes classifier based on logistic regression to give scores to the clusters.

After the weights of the views in a cluster c is summed, represented as $W(c)$, and the projection area of the model when viewed from the center of the cluster is obtained, represented as $A(c)$, the score for the cluster, $Score(c)$, is computed by the following equation (3).

$$Score(c) = \frac{1}{1+e^{\theta_0+\theta_1 W(c)+\theta_2 * A(c)}} \quad (3)$$

where $\theta_0 = 0.8, \theta_1 = 0.8$ and $\theta_2 = 1.2$. The values of θ_0, θ_1 and θ_2 were learned by training a logistic regression classifier [10]. In the training, two models were selected from every class of the 19 model classes in Princeton Segment Benchmark [7], and every selected model was treated by the measure in Subsection 5.2 to get clusters. Then, three volunteers were invited to mark whether a cluster contains best views to watch the model.

After the clusters are all given scores, the clusters with higher scores are more possible to well watch many segmented parts, and they are determined as our canonical regions.

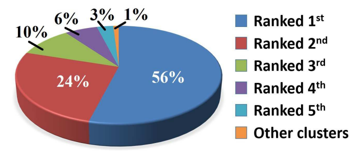


Figure 7. The statistics of the user's best viewpoints inside the ranked regions.

5.4. Validation

To validate the effectiveness of our canonical regions, we conducted a user study with 30 models, and invited 10 participants who must treat 3D models in their work. For a model, participants selected its best views respectively after interactively watching the model around it. Then, we investigated the correspondence between the participants' selections and our ranked regions. The statistics listed in Figure 7 show that their selections are mostly inside the regions that are ranked the first several ones. This means that our method is very effective for best view selection.

6. For Best View Search

In many applications, it is required only one best view for a model. In this case, we can try to perform view selection in the canonical regions ranked the first and the second, as they take up 80% of the user's selections in the statistics in Figure 7. In applying a view selection method in these two regions, we first use the method to evaluate the views of the two centers of these regions to get the better one, and then only sample candidate views in the region with the better view for best view search. In this way, computation can be saved much and without reducing the quality of the obtained view.

For searching N-best, we try to search the best views in different canonical regions. By this, similar views from a same canonical region can be avoided to get the succinct set of N-best views. (A canonical region is small in area, very possible for its views to watch similar parts.). The detailed steps are as followed. At first, we select the cluster ranked the first, and then iteratively add clusters one by one from the remaining clusters until it is evident that the views from these clusters can almost cover the model information. For a cluster to be added, it must be the one among the remaining clusters likely to add the most information lacked in the selected clusters. This procedure is based on the corresponding work in [27]. In our implementation, we use the segmented parts as the representatives to measure the model information. When all segmented parts can be watched, the search ends. For every added cluster, we simply use the center of its bounding ellipse as the viewpoint to add to the N-best views, in our current implementation for efficiency and without loss of quality. This is because the bounding ellipses are very small in area, so that the center is very likely to watch the related parts well.

7. Results and Discussion

We implemented our method and some view evaluation methods to test our effectiveness in improving view selection. They are the methods by evaluating view entropy (VE) [34], mesh saliency (MS)[18], and viewpoint saliency Kullback-Leibler distance (vSKL)[27]. We conducted the experiments on a personal computer equipped with an Intel i7-2600 CPU, 4GB RAM, and an nVidia GT420 GPU. To compute saliency values for mesh saliency and viewpoint saliency Kullback-Leibler distances, we employed the CUDA computing platform for fast computation. Using the results, we discuss our performance in the following subsections.

7.1. Effectiveness

Table 1 lists the best views for the tested models. They were selected by the three view evaluation methods, with and without our method applied to constrain the search space. Clearly, with our method applied, the selected best views were generally effective at showing many contents of the models, preventing existing methods from producing pseudo high-quality views, as those marked in '?' in Table 1. This demonstrates that we can reliably obtain high-quality views.

7.2. Efficiency

In Figure 8, we list the acceleration times for existing methods to locate best views, when our method was applied to constrain the search space, where the rate for sampling views on the viewing sphere keeps the same everywhere. Here, the time cost includes the time required for generating our canonical regions when our method was applied. The statistics show that we can speed up existing methods by about 6 $\sim$ 12 times. In the statistics, we do not include the time cost of measurements that produce reusable results during preprocessing, such as mesh saliency computed during preprocessing.

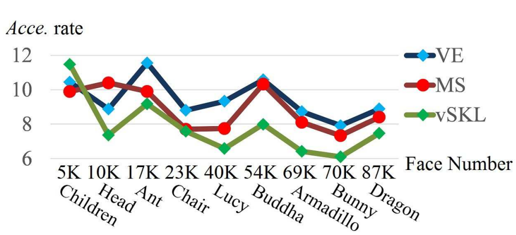
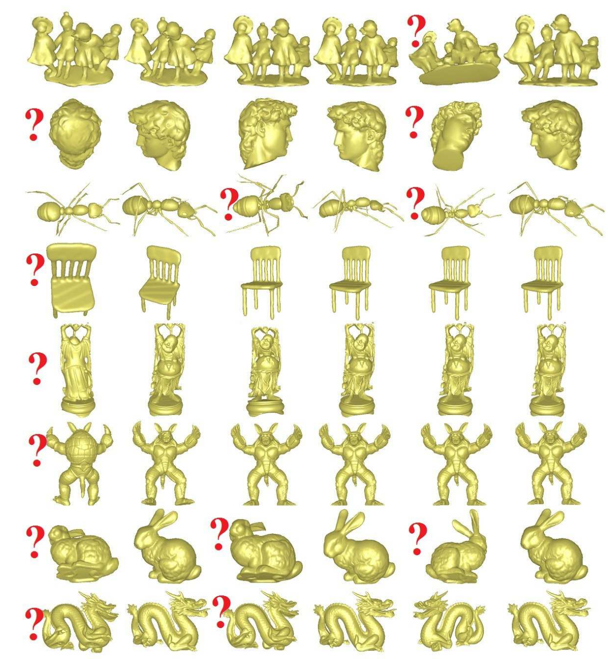


Figure 8. Efficiency increases for existing methods using our method.

Table 1. The obtained best views with or without our method applied.

VE		MS		vSKL	
Non$^{\#}$	Ours*	Non$^{\#}$	Ours*	Non$^{\#}$	Ours*



Non$^{\#}$: the view evaluation methods in use without our method applied.

7.3. N-best views

For searching for N-best views, we compared our method with a method that used semantic meaning [26] and the vSKL method [27]. In [26], the mean shift measurement is used to evaluate views, and the views with the highest evaluation scores in the remaining views are gradually selected and added to the N-best views. Although its selected views are able to watch the model well, some may be very similar, reducing the succinctness of the selected N-best views. As illustrated in Figure 9 (a), among the N-best views selected by this method for the Lucy model (copied from the paper), the first and sixth are very similar in terms of their visible content; a similar scenario exists for the second and fifth views. For our N-best views of this model in Figure 9 (b), five views are sufficient.

With the vSKL method, the seed-based strategy was adopted to add views gradually. Because of the blindness in the seed distribution process, this method may increase the number of views required and demand a significant amount of time, as discussed in Section 2. In contrast, our method can get N-best views succinctly, as shown in Figure 10. To compare efficiency, Table 2 lists the statistics for generating the N-best views in Figure 10. As shown, ours can perform

Figure 9. Comparison between our method and the semantic meaning based method [26] for searching N-best views. With the method in [26], seven views were generated (a); our method required only five views, thus reducing the number of required views. This is because our selection of views from different canonical regions is very helpful to prevent selecting similar views.

over 40 times faster than the vSKL method.

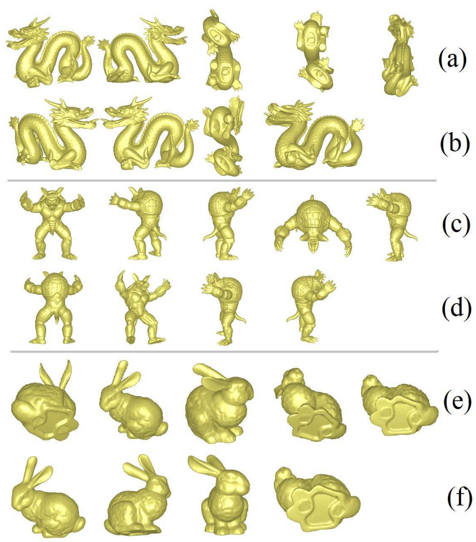


Figure 10. Comparison between our method and the vSKL method [27] for searching N-best views. The results in (a), (c), and (e) were obtained with the vSKL method, and the others with ours.

Table 2. Running time (seconds) for generating N-best views in Figure 10.

Model	Dragon	Armadillo	Bunny
vSKL	35.93	30.37	28.01
Ours	0.73	0.62	0.62
Acce.	48.15	47.78	44.47

Limitation. Our method aims to obtain the views that can effectively watch many contents of the model. This may generate views that are not suited to human being's perception habits. As illustrated in Figure 11, we may select the view to watch the table from under it, as the legs can be watched more from there. This is related to semantic computation, needing a further study in the future.

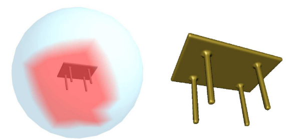


Figure 11. Our failed views to watch the table from under it, which are not suited to human being's perception habits.

8. Conclusion

Best view selection has been studied extensively in the literature. However, these research efforts employ brute-force search strategies to densely sample viewpoints around the model and check them individually. As a result, search efficiency is reduced, and poor views may be misidentified as high-quality views, owing to the shortcomings of existing view evaluation methods. In this paper, we addressed the challenge of finding the regions that are very likely to contain best views. Thus, the best views can be searched within constrained regions to obtain faster results and the guarantee of high-quality views. This is accomplished by decomposing the model into meaningful parts, and using the canonical views of those meaningful parts to build canonical regions. In this study, we also developed an analytic method to rapidly approximate canonical views of meaningful parts. Experimental results show our method's ability to significantly accelerate existing methods and guarantee the high quality of obtained views; moreover, our method can reduce the number of views required for N-best view search and achieve an acceleration of over 40 times.

In our current implementation, we mainly focus on the views that can effectively watch the geometric features of the model, which are extensively used in applications. When the contents of the model do not strongly correspond to its geometric features, it is necessary to find suitable segmentation methods, such as graph-based segmentation, to successfully approximate the contents. Subsequently, our method can be applied. This is an interesting issue to study in the future.

References

[1] I. Biederman and P. C. Gerhardstein. Viewpoint-dependent mechanisms in visual object recognition: Reply to tarr and bülthoff (1995). 1995.

[2] T. C. Biedl, M. Hasan, and A. López-Ortiz. Efficient view point selection for silhouettes of convex polyhedra. *Computational Geometry*, 44(8):399–408, 2011.

[3] V. Blanz, M. J. Tarr, H. H. Bülthoff, and T. Vetter. What object attributes determine canonical views? *Perception-London*, 28(5):575–600, 1999.

[4] U. D. Bordoloi and H.-W. Shen. View selection for volume rendering. In *IEEE Conference on Visualization(VIS'05)*, pages 487–494, 2005.

[5] J. Carreira and C. Sminchisescu. Constrained parametric min-cuts for automatic object segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10)*, pages 3241–3248, 2010.

[6] H.-T. Chen, T. Grossman, L.-Y. Wei, R. M. Schmidt, B. Hartmann, G. Fitzmaurice, and M. Agrawala. History assisted view authoring for 3d models. In *ACM Conference on Human Factors in Computing Systems*, pages 2027–2036, 2014.

[7] X. Chen, A. Golovinskiy, and T. Funkhouser. A benchmark for 3d mesh segmentation. *ACM Transactions on Graphics*, 28(3):73:1–73:12, 2009.

[8] X. Chen, A. Saparov, B. Pang, and T. Funkhouser. Schelling points on 3d surface meshes. *ACM Transactions on Graphics*, 31(4):29:1–29:12, 2012.

[9] M. Feixas, M. Sbert, and F. González. A unified information-theoretic framework for viewpoint selection and mesh saliency. *ACM Transactions on Applied Perception*, 6(1):1:1–1:23, 2009.

[10] A. Freedman, D. Statistical models: theory and practice. *Cambridge university press*, 2009.

[11] H. Fu, D. Cohen-Or, G. Dror, and A. Sheffer. Upright orientation of man-made objects. *ACM Transactions on Graphics*, 27(3):42:1–42:7, 2008.

[12] L. Hajder, Á. Pernek, and C. Kazó. Weak-perspective structure from motion by fast alternation. *The Visual Computer*, 27(5):387–399, 2011.

[13] D. D. Hoffman and W. A. Richards. Parts of recognition. *Cognition*, 18(1):65–96, 1984.

[14] G. Ji and H.-W. Shen. Dynamic view selection for time-varying volumes. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1109–1116, 2006.

[15] T. Kamada and S. Kawai. A simple method for computing general position in displaying three-dimensional objects. *Computer Vision, Graphics, and Image Processing*, 41(1):43–56, 1988.

[16] T. Kol, J. Liao, and E. Eisemann. Real-time canonical-angle views in 3d virtual cities. In *Vision, Modeling and Visualization Conference(VMV'14)*, pages 55–62, 2014.

[17] Y.-K. Lai, S.-M. Hu, R. R. Martin, and P. L. Rosin. Rapid and effective segmentation of 3d models using random walks. *Computer Aided Geometric Design*, 26(6):665–679, 2009.

[18] C. H. Lee, A. Varshney, and D. W. Jacobs. Mesh saliency. *ACM Transactions on Graphics*, 24(3):659–666, 2005.

[19] G. Leifman, E. Shtrom, and A. Tal. Surface regions of interest for viewpoint selection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'12)*, pages 414–421, 2012.

[20] S. Lienhard, M. Specht, B. Neubert, M. Pauly, and P. Müller. Thumbnail galleries for procedural models. *Computer Graphics Forum*, 33(2):361–370, 2014.

[21] K. Michael, F. Thomas, and R. Szymon. Rotation invariant spherical harmonic representation of 3d shape descriptors. In *Eurographics Symposium on Geometry Processing (SGP'03)*, pages 156–164, 2003.

[22] S. Palmer, E. Rosch, and P. Chase. Canonical perspective and the perception of objects. *Attention and performance IX*, pages 135–151, 1981.

[23] O. Polonsky, G. Patané, S. Biasotti, C. Gotsman, and M. Spagnuolo. Whats in an image? *The Visual Computer*, 21(8-10):840–847, 2005.

[24] M. Sbert, M. Feixas, J. Rigau, M. Chover, , and I. Viola. Applications of information theory to computer graphics. *Eurographics Tutorial*, pages 840–847, 2007.

[25] M. Sbert, D. Plemenos, M. Feixas, and F. González. Viewpoint quality: measures and applications. In *Eurographics conference on Computational Aesthetics in Graphics, Visualization and Imaging*, pages 185–192, 2005.

[26] A. Secord, J. Lu, A. Finkelstein, M. Singh, and A. Nealen. Perceptual models of viewpoint preference. *ACM Transactions on Graphics*, 30(5):109:1–109:12, 2011.

[27] E. Serin, S. Sumengen, and S. Balcisoy. Representational image generation for 3d objects. *The Visual Computer*, 29(6-8):675–684, 2013.

[28] A. Shamir. A survey on mesh segmentation techniques. *Computer Graphics Forum*, 27(6):1539–1556, 2008.

[29] T. Shigeo, F. Issei, T. Yuriko, and N. Tomoyuki. A feature-driven approach to locating optimal viewpoints for volume visualization. In *IEEE Conference on Visualization(VIS'03)*, pages 495–502, 2003.

[30] D. Sokolov and D. Plemenos. Virtual world explorations by using topological and semantic knowledge. *The Visual Computer*, 24(3):173–185, 2008.

[31] M. J. Tarr and H. H. Bülthoff. Image-based object recognition in man, monkey and machine. *Cognition*, 67(1):1–20, 1998.

[32] M. J. Tarr and D. J. Kriegman. What defines a view? *Vision Research*, 41(15):1981–2004, 2001.

[33] M. J. Tarr, P. Williams, W. G. Hayward, and I. Gauthier. Three-dimensional object recognition is viewpoint dependent. *Nature neuroscience*, 1(4):275–277, 1998.

[34] P.-P. Vázquez, M. Feixas, M. Sbert, and W. Heidrich. Viewpoint selection using viewpoint entropy. In *Vision, Modeling and Visualization Conference(VMV'01)*, pages 273–280, 2001.

[35] T. Vieira, A. Bordignon, A. Peixoto, G. Tavares, H. Lopes, L. Velho, and T. Lewiner. Learning good views through intelligent galleries. *Computer Graphics Forum*, 28(2):717–726, 2009.

[36] D. Weinshall and M. Werman. On view likelihood and stability. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):97–108, 1997.

[37] H. Yamauchi, W. Saleem, S. Yoshizawa, Z. Karni, A. Belyaev, and H.-P. Seidel. Towards stable and salient multi-view representation of 3d shapes. In *IEEE Conference on Shape Modeling and Applications (SMA'06)*, pages 40–46, 2006.

[38] J. Zhang, J. Zheng, C. Wu, and J. Cai. Variational mesh decomposition. *ACM Transactions on Graphics*, 31(3):21:1–21:14, 2012.