Finding Canonical Views by Measuring Features on the Viewing Plane

Wencheng Wang[*] , Liming Yang[^] and Dongxu Wang[†]
[*]State Key Laboratory of Computer Science, Institute of Software, CAS, Beijing, China
E-mail: whn@ios.ac.cn Tel: +86-10-62661611
[^]State Key Laboratory of Computer Science, Institute of Software, CAS, Beijing, China
[†]College of Information Engineering, Xiangtan University, Xiangtan, China

***Abstract*— Canonical views are referred to the classical three-quarter views of a 3D object, always preferred by human beings, because they are stable and able to produce more meaningful and understandable images for the viewer. Unlike existing methods to measure features in the 3D space for view selection, this paper proposes to measure features on the viewing plane, taking into account the influence of feature deformation due to perspective projection on view evaluation. Meanwhile, we try to have more features perceptible instead of having preferred features displayed more in a good view, which is aimed by existing methods. As a result, we can effectively obtain canonical views with only geometry computation, without troublesome semantic computation, which are always needed in existing techniques for obtaining good views.**

I. INTRODUCTION

View selection aims at obtaining good views that can help the user to understand the object. From psychophysical investigation on the impact of the aspect of an object on the quality of recognition [1], most people prefer canonical views when they watch an object, where *canonical views* are referred to the classical three-quarter views of a 3D object. This is because canonical views are stable and able to produce more meaningful and understandable images for the viewer [2]. The general approach for view selection is by distributing sample points on a sphere bounding the object, called *a viewing sphere*, and then taking the points as viewpoints to evaluate their corresponding views for finding good views. Clearly, view evaluation plays a key role in this process.

Existing view evaluation methods can be classified into two categories, by measuring either geometric information or semantic meaning. The methods by geometric information are easy to implement, and many kinds of geometric attributes have been investigated, such as visibility, curvatures, and silhouettes [3], but they are not always effective. As for the methods by semantic meaning, they try to first construct the correspondence between geometric features and object contents with semantic meaning, and then evaluate each view by measuring its related semantic meaning [3]. They can get canonical views in general. However, it is not an easy task to assign semantic meaning to geometric features, which prevents these methods from applications.

This paper presents a new view evaluation method. It is based on the observation that object deformation on the viewing plane due to perspective projection have much influence on object understanding, but not considered in existing methods which always measure object features in the 3D space, and a canonical view tries to display as many kinds of features as possible, not as existing methods do to have their preferred features displayed more. Thus, we design a view-dependent curvature for measuring features on the viewing plane, and use the standard deviation to improve entropy computation for having more kinds of features visible, unlike existing techniques only employing entropy computation for view evaluation. As a result, we can effectively find canonical views as the methods using semantic computation, but with only geometry computation, as illustrated in Fig.1.

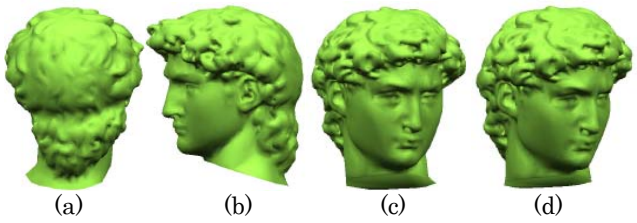


Fig.1 Our new method can find good views as the methods using semantic meaning, but use only geometry computation. Here, the first three views are obtained by (a) maximum visible mean curvature [3], (b) mesh saliency [4], and (c) semantic computation [3]. The fourth view is by our new method, which is very like the view in (c).

II. RELATED WORK

View quality measures have been applied in many fields. Among them, most are for polygonal models since objects are always approximated with polygons. In this paper, our proposed view evaluation method is also discussed for polygonal models. In the following paragraphs, we mainly discuss related work for polygonal models and the methods using curvatures.

For polygonal models, it is convenient to investigate geometric features for view evaluation, such as visibility [5], depth maps [6], curvatures and silhouettes [3] [7] [8]. These features are often measured by integrated with the information theory to estimate the view quality [9]. Surveys of these techniques can be found in [3]. Among them, curvatures are very popular and always regarded very effective for view

evaluation. They include the Gaussian curvature distribution over the entire surface of the object [7] or the visible portion of the surface [3], and Gaussian-weighted mean curvatures computed in a multi-scale manner to obtain stable salient features [4]. In comparison with these methods, our method measures the perceptibility of features on the viewing plane to find the views that can show different kinds of features as many as possible, unlike them trying to have their preferred features shown more in their obtained views and measuring features in the 3D space, which is not effective to measure the perceptibility of features due to their deformation by perspective projection. Results will show that we can effectively find canonical views, superior to existing methods with geometric features measured only.

In many applications, there is more important information than merely the geometry of the object [10]. Thus, there are methods developed to evaluate views using semantic meaning. Some of these [3] [10] [11] first segment the object and assign semantic values to the segmented parts thereof, where either larger parts are given higher values according to the assumption that larger parts have a higher likelihood to include some semantic meaning, or semantic-oriented segmentation is taken. Measurements are then carried out using these values. Other methods try to derive functions or statistical learning machines to reflect accurately the correspondence between geometric features and semantic meaning [12] [13], and even leverage the results of a user study to optimize the parameters of a model for viewpoint searching that is a combination of attributes known useful for view selection [14]. Such derivation, however, often requires manual intervention or time-consuming training. Recently, a method was proposed to use web images to find the best view of a 3D model [15], by exploring human perception experiences to derive human beings' viewing preferences for similar 3D models. Using these methods, the selected views are generally satisfactory, if the object is segmented properly or the derived learning machines are robust. However, to the best of our knowledge, segmentation computation is always difficult and expensive, and poorly segmented results have an adverse effect on viewpoint selection. Furthermore, with regard learning machines, they are highly dependent on the training samples, and it is not easy to collect enough suitable samples, e.g. it cannot be guaranteed that there are enough web images for a 3D model. With our new method, despite not executing any semantic computation, we can still find good views that are very like those produced by the methods using semantic meaning, when there is much correspondence between geometric features and semantic meaning, which is always true in many applications, especially when we watch an object about which we have no knowledge beforehand.

Curvatures have been studied extensively across a large spectrum of applications. We found that, in non-photo-realistic rendering, they are always used to extract important features, such as ridges and valleys [16], and suggestive contours [17]. Considering that human observers are highly sensitive to high luminance variation and the perceptual effect thereof, some methods take into account the shading variation on the viewing plane to extract features such as apparent ridges [18] and photic extremum lines [19], to enhance the representation and understanding of 3D objects. Partly motivated by the work in [18] [19], we design a view-dependent curvature to consider the effect of the perspective projection on displaying objects, to ensure that salient features are well perceptible in good views. As we are only concerned about using curvature information to distinguish features for view evaluation, not the actual shapes and locations of the features as required for non-photo-realistic rendering, our computation can be simplified. As verified by experiments, our proposed method works very well for view selection.

III. VIEW-DEPENDENT CURVATURE

To efficiently measure the perceptibility of 3D features on the viewing plane, we define a view-dependent curvature to compute, showing the normal variation of the geometric features against the viewing plane. Here, we follow the framework of [18] to compute the view-dependent curvature, but adopt two different operations, to well represent the appearance of features on the viewing plane. The first is to use perspective projection instead of parallel projection, and the second is to project the normal onto the viewing plane, while not process normal information in the object space as done in [18]. In the following subsections, we will first introduce the computation of our view-dependent curvature, and then compare it with two popularly used curvatures in view evaluation by some tests, to show that our view-dependent curvature can be more effective in revealing salient features.

A. Computation

Our view-dependent curvature is computed at every pixel after a normal field is constructed on the viewing plane, where every pixel carries a 3D normal from the object.

Constructing the normal field.

The normal field is constructed by assigning normals to pixels. The steps are given below.

(1) The visible parts of the object are projected onto the viewing plane, with the 3D normals of the visible vertices assigned to their corresponding projection pixels. If several visible vertices are projected onto the same pixel, only the closest vertex to the viewpoint has its normal assigned to the pixel.

(2) Pixels covered by the visible parts, but not projections of 3D vertices, obtain their 3D normals by linearly interpolating the normals at their respective neighboring projection pixels of visible vertices. Here, a case may occur that the neighboring projection pixels may come from different surfaces, and so artifacts may be caused. Fortunately, such a case is seldom to occur in practice, because the projection of a facet on the viewing plane is very small and the surface for a salient feature occupies a large projection area in general. Such artifacts will have little impact on view evaluation.

(3) Pixels not covered by the object projection are assigned the normal vertical to the viewing plane.

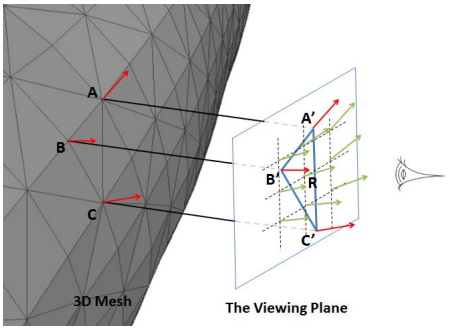


Fig. 2 Constructing the normal field on the viewing plane.

Without loss of generality, it is given an example to show the above steps in Fig 2. Here, the visible vertices A, B, and C, of a 3D object are projected onto the viewing plane, and their normals are assigned to their projection pixels A', B', and C', respectively, e.g., normal $\boldsymbol{n}(A)$ in red is assigned to Pixel A'. Thereafter, each pixel covered by the object projection, say pixel R, obtains its normal by interpolation with the normals of its neighboring projection pixels. Here, the interpolation can be executed via the grids to organize the pixels on the viewing plane, as shown by the grid lines on the viewing plane in Fig 2.

Curvature computation.

We compute the view-dependent curvature at every pixel. Here, we adopt the standard techniques in [20] to estimate such curvatures.

The curvature operator S is defined as

$$S(\boldsymbol{r}) = D_{\boldsymbol{r}}(\boldsymbol{n})$$

where $D_{\boldsymbol{r}}(\boldsymbol{n})$ is the directional derivative of the normal $\boldsymbol{n}$ at a pixel, say R, along vector $\boldsymbol{r}$ in its tangent plane S, known as a Weingarten map or the shape operator, which is a linear map from the tangent plane at R to a tangent of the Gauss sphere parallel to the tangent plane.

Given a choice of basis, S can be represented as a symmetric 2*2 matrix, and the maximum and minimum principal curvatures at pixel R, k_1 and k_2, respectively, are the eigenvalues of S, where $|k_1| \geq |k_2|$. Then, the mean curvature at pixel R is computed as $(k_1+k_2)/2$, and it is taken as our view-dependent curvature at this pixel.

B. Comparison Tests

To investigate the effectiveness of our view-dependent curvature in representing features, we carried out experiments in comparison with two other curvatures, the Gaussian curvature [3] and the radial curvature [17], which are used widely in existing methods for view evaluation. These two curvatures are both computed at points on the object surface in the 3D space, while the radial curvature is view-dependent and the Gaussian curvature is not. The Gaussian curvature at a point is the product of the two principal curvatures at the point, whereas the radial curvature at a point is the normal curvature of the surface in the direction defined as the projection of the viewing vector onto the tangent plane at the given point.

Two models were tested and the results are illustrated in Fig 3, where the curvature values are assigned different colors to highlight the distribution of the curvature

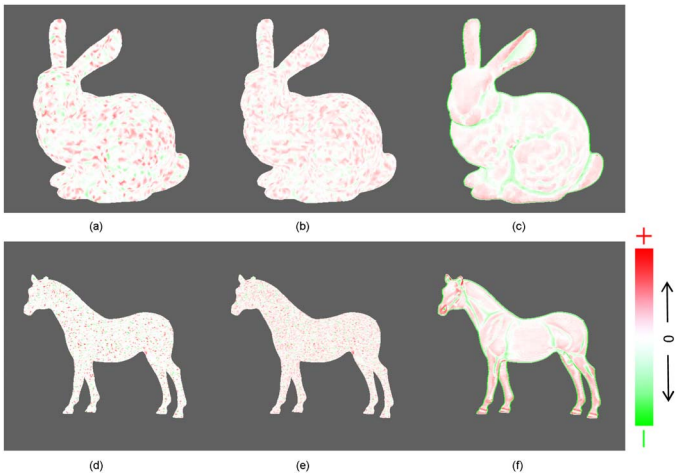


Fig. 3 By our curvature, the valleys and ridges for salient features can be effectively perceptible, superior to using the Gaussian curvature and the radial curvature, where the images (a) and (d) are by the Gaussian curvature, images (b) and (e) by the radial curvature and images (c) and (f) by our curvature.

information, and positive and negative values are depicted in red and green respectively. From the results, the valleys and ridges can be effectively perceptible to distinguish salient features using our view-dependent curvature, but not using the other two curvatures. This is because that our curvature is more efficient to represent larger features while ignoring smaller features owing to the scaling effect of perspective projection. As point out in [3], larger features are always more important to carry object contents. Thus, our curvature provides a solid base for efficient view evaluation.

IV. VIEW EVALUATION

View evaluation is for obtaining good views that each can well display the object contents for human beings to understand the object efficiently. In this paper, we let curvatures represent the object contents, and design our view evaluation method as followed. We divide the value range of our curvatures into sub-ranges, called *bins*, and take every bin as an event to measure with the information theory.

Though the Shannon entropy has been used widely and can achieve high values when the investigated events each appear in similar probabilities, it cannot be guaranteed as many events occurring as possible. Thus, we use the standard deviation to enhance view evaluation, trying to have more events to display in a good view. This is benefited from that the standard deviation on a global basis has its values varied with the number of investigated events.

Thus, by checking the standard deviations of the views, we get the candidate views that can have as many contents as possible to display. Then, we get the good view for watching the object, which has the highest entropy value among the candidate views. In the following, we will first introduce the computation for the Shannon entropy and the standard deviation, and then discuss the effectiveness of our view evaluation method and its parameters.

A. *The Shannon Entropy*

The Shannon entropy is computed as $E(\boldsymbol{X}) = -\sum_{i=1}^{n} p_i \log p_i$ for a discrete random variable $\boldsymbol{X}$, with the event set of $a_1, a_2, \ldots, a_m$, where $p_i = P[\boldsymbol{X} = a_i]$, and $0 \log 0 = 0$ is allowed for continuity.

In our method, we have our bins correspond to the events $a_1, a_2, \ldots, a_m$, and compute the Shannon entropy in the followed steps.

1) We count the pixels in these bins to obtain $m_1, m_2, \ldots, m_m$ corresponding to the events $a_1, a_2, \ldots, a_m$ respectively.

2) Let $SM = \sum_{i=1}^{n} m_i$, and compute $p_i = \frac{m_i}{SM}$, $i=1, 2, \ldots, n$, to obtain the Shannon entropy value for the investigated view, where n is the number of bins.

B. *The Standard Deviation*

The standard deviation for a view, s_i, is computed in the following formulae:

$$\overline{m} = \frac{\sum_{i=1}^{n} m_i}{n} \quad (1)$$

$$s_i = \sqrt{\frac{\sum_{i=1}^{n} (m_i - \overline{m})^2}{n-1}} \quad (2)$$

By formulae 1 and 2, we know that when the pixels covered by the object are in a constant number, with the visible bins being more, the numbers of m_i for the visible bins will differ smaller in general, and so the standard deviation tends to be smaller. As illustrated in Fig 4, when the David head model has more contents displayed with each in more similar amounts in a view, the corresponding standard deviation for the view has smaller values. However, the projection area of the object also has much influence on computing the standard deviation. When the projection area is smaller, the visible bins tend to be fewer, which may also have the standard deviation be smaller. On the contrary, when the projection area is larger, more visible bins tend to appear and the count differences between these bins may be larger to lead to a higher standard deviation value. As a result, a view with a bigger projection area and more visible bins may have a higher standard deviation value than another view with a smaller projection area and fewer visible bins, which is not what we expect. Because the projection areas in different views may differ much, we think it is reasonable that a view with more visible bins should have its standard deviation neither much high nor much low. Thus, we take the averaged standard deviation of all the investigated views as a criterion to search for the views that would have as many contents displayed as possible. This means that a view is more possible to be a good view if its standard deviation is nearer the averaged one. Thus, we generally select the candidate views as the first several nearest views with their respective standard deviation to the averaged standard deviation of all the investigated views. We found this works very well by a lot of tests.

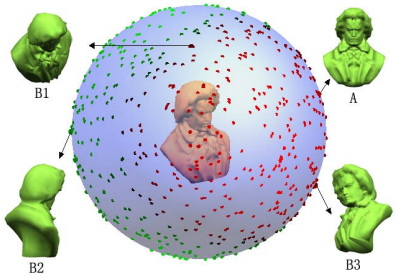
Standard Deviation:	*868.3693*	*729.4034*	*623.6733*
Num. of bins:	*385*	*423*	*462*

Fig. 4 The view tends to have a smaller standard deviation value when it has more visible bins and the bins each appear in more similar amounts.

C. *Effectiveness*

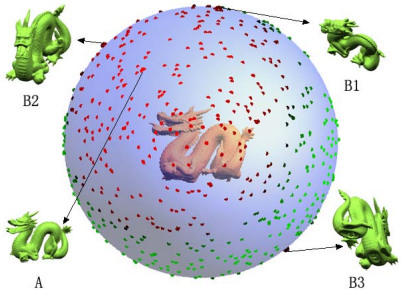
In Fig.5, two models are tested to show the effectiveness of our method for finding good views. Here, 1000 viewpoints are randomly sampled on the viewing sphere, and we display the view with the highest value of Shannon entropy, and the first three nearest views by their respective standard deviation to the averaged one of all these 1000 views. Clearly, the views with the highest Shannon entropy value are not always good views, and our selected views are good, very like the canonical views for the tested models.

In our view selection method, there are two parameters to consider, *pixel resolution* of the viewing plane, and *the*



View	*Shannon entropy*	*Diff. of standard deviation*#
A	4.0003	1067.15
B1	3.20404	4.09766
B2	2.83349	4.39893
B3	3.2637	5.875
Our selected view: B3		

Diff. of standard deviation#: the difference between the standard deviation of the view and the averaged standard deviation for all the tested views.



View	*Shannon entropy*	*Diff. of standard deviation*#
A	3.79104	1228.14
B1	2.95031	0.677246
B2	2.96902	1.32568
B3	2.82056	1.64209
Our selected view: B2		

Fig. 5 Good views can be efficiently found by our method, which are not always the views with the highest Shannon entropy value. Here, A is the view with the highest Shannon entropy value. B1, B2, B3 are the three nearest views with their standard deviation to the averaged standard deviation of all investigated views.

number of bins for dividing the value range of our curvature. In the following, we discuss them respectively.

Pixel resolutions

The pixel resolution may impact the appearance of features. If the pixel resolution is higher, smaller features have more chances to display. Otherwise, the lower pixel resolution may only allow larger features to appear. It is not as clear, however, what pixel resolutions are better to ensure that sufficient important features are displayed for effective view evaluation. This is because that the important features for different models may vary significantly in size, and even within a model, various important features may have different sizes. As a general rule, the pixel resolution should not be too low or too high. If it is too low, too many object features may be omitted. On the contrary, when the resolution is too high, very detailed features will be displayed, and in turn to obscure the perceptibility of important features. We carried out several experiments to investigate the effect of pixel resolutions on view evaluation. Some of the results are listed in Fig 6, where 512 curvature bins are used.

Fig. 6 Best views obtained for the tested models with different pixel resolutions, where 200 views are evaluated per model with the viewpoints distributed randomly over the surface of the viewing sphere, which has a radius three times the radius of the bounding sphere of the model, and has the model located at its center, as is the case in [9].

Table 1. The statistics for finding the good views in the third column in Fig 6.

Models	Num. of triangles	Time (second)
Beethoven	5.00k	59.9
David head	10.49k	61.5
Cup with saucer	32.75k	77.9
Dragon	168.82k	50.7
Motor	22.07k	33.1

From the results in Fig 6, it can be seen that for a single model, several pixel resolutions may be suitable to obtain good views, but for different models, such pixel resolutions may differ. By a survey of these results, the pixel resolution of 512 * 512 could be a good choice for all the tested models, because their corresponding good views are all like canonical views. Since the tested models vary greatly in shape and feature size, such a pixel resolution may be regarded suitable for most models. Thus, in all our experiments, we used the viewing plane of 512*512 pixels in general, except the cases that are specifically mentioned otherwise. In Table 1, it is listed the timings for finding the good views in the third column in Fig 6, where the configuration for the used computer is given in Section V.

Bins

The number of bins has impact on computing the probabilities of the events and the standard deviation values of views, and so influences the efficiency of view evaluation. If fewer bins are used, the statistics computation will be coarser to lower view evaluation. On the contrary, if more bins are used, the computation cost will be higher to lower the efficiency of finding good views. As we know, with more bins, the count difference for every bin between neighboring views tends to be smaller, and so the evaluation values will change more smoothly between neighboring views. With such smoothness, it will be easy to select good views. By testing many models with various numbers of bins, it is found that when more than 256 bins are used, the Shannon entropy values and the standard deviation values change much smoothly between neighbor views. Considering the balance between computation cost and view selection efficiency, we generally use 512 bins for our view evaluation, whose associated changes between neighboring views are always enough smooth to efficiently select good views while not much time and storage cost are required.

V. Results And Discussion

We made tests on a PC with an Intel Core2 Duo E6559 @ 2.33GHz CPU, 2G RAM and an Nvidia Geforce 8600GTS GPU, where projection and normal field construction are computed on the GPU, and other computations on the CPU. For every model tested, we generally sampled 200 viewpoints uniformly over the surface of the viewing sphere around the object, and evaluated their corresponding views to select the best one. In the experiments, we set the radius of the viewing sphere to three times the radius of the bounding sphere of the object, as is the case in [9], to ensure that the object is observed completely from any viewpoint.

By the results in Fig 1, Fig 5, Fig 6 and Fig.7, it is clear that our method is very effective in finding canonical views, superior to the existing methods via measuring geometric information such as visible mean curvatures [3], mesh saliency [4] and depth [6], and very like the methods using semantic meaning [3][13]. In Fig 8, more results are displayed to show the effectiveness of our method, where some results are not very fitted with the viewing habits of human beings in

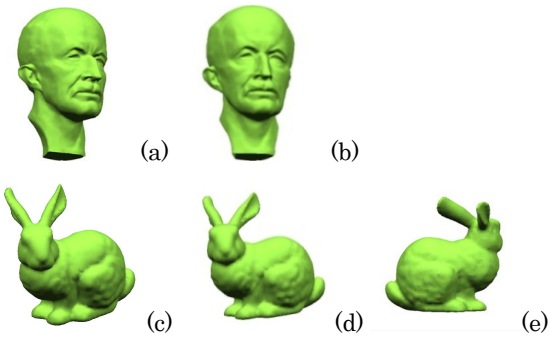


Fig.7 Results of comparing our method with a method by semantic computation [13] and a depth-based method [6], where (a) and (c) are obtained by our method, (b) and (d) copied from [13], and (e) from [6].

ordinary living, but they really like canonical views, showing more contents. In Table 1 and Fig.8, we also list the time cost for our method to get best views of these models, which show that our method has similar efficiency as existing methods for view selection. For example, it is reported in a paper published in 2011 that their proposed method averagely needs 46 seconds to treat a model for its alignment [21], a similar task as finding best views.

Because our method is still based on geometric computation, it may fail when object contents cannot be efficiently distinguished with geometric features, especially the curvatures. These cases can be classified into two categories, with the first having repeated features with many curvatures, and the second containing many small geometric features to reduce the appearance of important features. In these cases, local geometric features are mainly for representing small-scale object contents, but not their large-scale and more important object contents. As illustrated in Fig 9, for the armadillo model, local curvatures can well represent its repeated tiled pieces of its back shell, but not the back shell, so that our selected best view will watch its back mainly (View B2), which is the same of the Igae model as its bob, sideburns and neck section contain many small features. Fortunately, for the nearest several views with their respective standard deviation values to the averaged standard deviation value, when they are sorted with their entropy values from high to low, the second or third views may be satisfactory in general, more like canonical views, since our method tries to give higher entropy values to the views that can show more contents. Thus, in viewpoint selection, we may provide several candidate views, and let the user select his best one.

In our method, we try to find canonical views by measuring the perceptibility of features on the viewing plane, while not care the orientation of the model on the viewing plane. Thus, it is not guaranteed that our obtained view is upright. Upright views are much related with the viewing experiences of human beings in the ordinary living, while not with the features. It is better to find upright views by semantic meaning, as done in [12] [13].

In summary, our method can efficiently find canonical views by measuring geometric information only, without the troublesome task of semantic computation. Though it may fail in some cases that the correspondence is weak between geometric features and the object contents that the

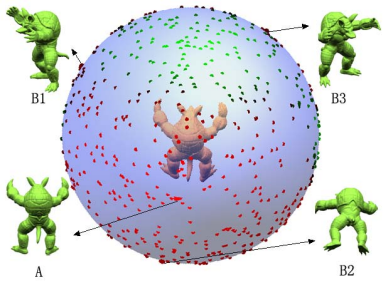
Fig. 8 More results by our method and the cost time (seconds) for finding them, showing the effectiveness of our method.

viewer wishes, as discussed in [22], where the methods by measuring geometric information cannot take effect, there are a lot of cases that such correspondence is strong, such as image-based modeling [23]. In particular, when we watch a model without any beforehand knowledge about it, we generally investigate it through its geometric features. In these cases, the canonical views are always preferred by human beings. Thus, our method can easily find applications.

VI. Conclusions

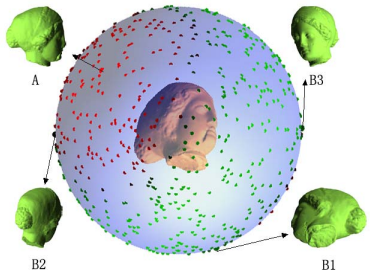
This paper presents a new view evaluation method that can find good views to efficiently display object contents, very like the canonical views as most people prefer to. It is motivated by the observation that people understand an object by viewing its projection on the viewing plane, and trying to watch as many object contents as possible in a view. Thus, unlike existing methods to perform view evaluation in the 3D space and try to have their preferred features displayed more in a good view, our method evaluates views on their corresponding viewing plane and tries to have as many kinds of features perceptible as possible. Results have shown the effectiveness of our method. As our method uses only geometry computation, it is very easy to implement, and so helpful to spread its use in applications.

Viewpoint selection is much related with object understanding. When the correspondence is very weak between interesting object contents and geometric features, our new method may be ineffective. Fortunately, in many applications, people always understand an object through watching its geometric features, so our method can easily find applications. As for the case that interesting object contents are less related with geometric features, the framework of our new method can be also used to have more interesting



View	Shannon entropy	Diff. of standard deviation
A	2.99506	545.32
B1	2.51994	0.235352
B2	2.60626	0.700195
B3	2.48367	1.03418

Our selections: The first: B2; The second: B1; The third: B3.



View	Shannon entropy	Diff. of standard deviation
A	4.56084	1284.13
B1	3.8176	1.31787
B2	3.91646	1.4043
B3	3.85429	2.18066

Our selections: The first: B2; The second: B3; The third: B1.

Fig. 9 Examples are displayed to show the cases that our method may fail when important object contents are not efficiently represented by local geometric features, where the B2 views are selected by ours, but they are not good. Fortunately, the second or third best views may be more like canonical views, as our method tries to give higher entropy values to the views that can watch more contents.

contents perceptible in good views, except that interesting information is measured instead of geometric information. This will be studied in the future.

ACKNOWLEDGMENT

We would like to thank the reviewers for their valuable comments. This work is partially supported by Natural Science Foundation of China (60773026, 60833007).

REFERENCES

[1] M. J. Tarr, and D. J. Kriegman, "What defines a view?" *Vision Research, vol.41, no.*15, pp.1981 – 2004, 2001.

[2] V. Blanz, M. J. Tarr, and H. H. Bülthoff , "What object attributes determine canonical views?" *PERCEPTION –LONDON, vol.28, no.*5, pp.575 – 600, 1999.

[3] O. Polonsky, G. Patan, S. Biasotti, C. Gotsman, and M. Spagnuolo, "What's in an image?" *The Visual Computer, vol.21*, pp.840–847, 2005.

[4] C. H. Lee, A. Varshney, and D. W. Jacobs, "Mesh saliency," *Proc. of ACM SIGGRAPH'05*, pp.659–666, 2005.

[5] F. Gonzalez, M. Sbert, and M. Feixas, "Viewpoint-based ambient occlusion," *IEEE Computer Graphics and Applications, vol. 28*, pp.44–51, 2008.

[6] P.-P. V´azquez, "Automatic view selection through depth-based view stability analysis," *The Visual Computer, vol.25*, pp.441–449, 2009.

[7] D. Page, A. Koschan, S. Sukumar, B. Roui-Abidi, and M. Abidi, "Shape analysis algorithm based on information theory," Proc. of *IEEE International Conference on Image Processing, Vol.1, pp.229-232, 2003.*

[8] T. C. Biedl, M. Hasan, and A. Lpez-Ortiz, " Efficient view point selection for silhouettes of convex polyhedral," *Computational Geometry, vol.44, no.*8, pp.399 – 408, *2011*.

[9] M. Feixas, M. Sbert, and F. Gonz´alez, "A unified information-theoretic framework for viewpoint selection and mesh saliency, " *ACM Trans. Appl. Percept, vol.6*, no.1, pp.1-23, *2009.*

[10] D. Sokolov, and D. Plemenos, "Virtual world explorations by using topological and semantic knowledge," *The Visual Computer, vol.24*, pp.173–185, 2008.

[11] M. Mortara, and M. Spagnuolo, "Semantic-driven best view of 3D shapes," Computers & Graphics, vol.33, no.3, pp.280-290, 2009.

[12] H. Fu, D. Cohen-Or, G. Dror, and A. Sheffer, "Upright orientation of man-made objects," *ACM Trans. Graph., vol. 27*, article 42, 2008.

[13] T. Vieira, A. Bordignon, A. Peixoto, G. Tavares, H. Lopes, L. Velho, and T. Lewiner, "Learning good views through intelligent galleries," *Computer Graphics Forum, vol. 28, no.*2, pp.717–726, 2009.

[14] A. Secord, J. Lu, A. Finkelstein, M. Singh, and A. Nealen, "Perceptual Models of viewpoint preference,: *ACM Transactions on Graphics*, vol. 30, no.5, article 109, 2011.

[15] H. Liu, L. Zhang and H. Huang, "Web-image driven best views of 3D shapes," The Visual Computer, vol.28, pp.279-287, 2012.

[16] Y. Ohtake, A. Belyaev, and H.-P. Seidel, "Ridge-valley lines on meshes via implicit surface fitting," Proc. of *ACM SIGGRAPH '04*, pp.609–612, 2004.

[17] D. Decarlo, A. Finkelstein, S. Rusinkiewicz, and A. Santella, "Suggestive contours for conveying shape," *ACM Trans. Graph., vol.22*, pp.848–855, *2003*.

[18] T. Judd, F. Durand, and E. Adelson, "Apparent ridges for line drawing," *Proc. of ACM SIGGRAPH '07*, 2007

[19] X. Xie, Y. He, F. Tian, H.-S. Seah, X. Gu, and H. QIN, "An effective illustrative visualization framework based on photic extremum lines (pels)," *IEEE Transactions on Visualization and Computer Graphics, vol.13*, pp.1328–1335, 2007.

[20] S. Rusinkiewicz, "Estimating curvatures and their derivatives on triangle meshes," *Proc. of International Symposium on 3D Data Processing Visualization and Transmission 2004, pp.*486–493, 2004.

[21] H. Johan, B. Li, Y. Wei and Iskandarsyah, "3D model alignment based on minimum projection area," The Visual Computer, vol.27, pp.565-574, 2011.

[22] D. Sokolov, and D. Plemenos, "Viewpoint quality and scene understanding," *Proc. of The 6th EG Symposium on Virtual Reality, Archaeology and Cultural Heritage*, pp.67–73, 2005.

[23] S. Fleishman, D. Cohen-Or, and D. Lischinski, "Automatic camera placement for image-based modeling," *Computer Graphics Forum*, vol.19, no.2, pp. 101-110, 2010.