

Vis Comput (2009) 25: 441–449
DOI 10.1007/s00371-009-0326-4

ORIGINAL ARTICLE

Automatic view selection through depth-based view stability analysis

Pere-Pau Vázquez

Published online: 3 March 2009
© Springer-Verlag 2009

Abstract Although the real world is composed of three-dimensional objects, we communicate information using two-dimensional media. The initial 2D view we see of an object has great importance on how we perceive it. Deciding which of the all possible 2D representations of 3D objects communicates the maximum information to the user is still challenging, and it may be highly dependent on the addressed task. Psychophysical experiments have shown that three-quarter views (oblique views between frontal view and profile view) are often preferred as representative views for 3D objects; however, for most models, no knowledge of its proper orientation is provided. Our goal is the selection of informative views without any user intervention. In order to do so, we analyze some stability-based view descriptors and present a new one that computes view stability through the use of depth maps, without prior knowledge on the geometry or orientation of the object. We will show that it produces good views that, in most of the analyzed cases, are close to three-quarter views.

Keywords Best view · Automatic view selection · View stability

1 Introduction

In Computer Graphics, the automatic selection of a representative view of an object is useful for many applications, such as the selection of thumbnails for objects data-

bases, automatic camera positioning in CAD systems, automatic scene composition, surgery planing or training, and so on. Furthermore, it is related with other composition problems such as virtual cinematography or camera placement in games.

Although very related with human visual perception, view selection is often addressed under strong conditions such as no prior knowledge on the geometry or orientation of the model. Despite the fact that this widens the potential applications, it poses some problems to the use of some of the well-known results from psychophysical studies.

Visual perception has been studied since the eighteenth century, and, although several theories have been developed, and many experiments have been carried out since then, we do not have definite answers to many problems [26]. For instance, the way humans understand three-dimensional objects remains unclear. In spite of this, several results may be useful for good view selection. Psychophysical experiments showed that for many models, a subset of views are preferred (canonical views), and these often correspond to three-quarter views [3]. The reason why we do prefer such kinds of views is not clear, but it seems that we like to see all the three dimensions of the object at the same time. Moreover, the preferred views are usually stable too. A view is stable when small moves away from it do not substantially change the information we see.

Since good view selection algorithms often deal with models without semantic information, these results are difficult to apply. The automatic orientation of objects has often been addressed with the use of Principal Components Analysis, but, in several cases, these algorithms still do not get the right orientation for, for instance, man-made objects ([11]). However, it would be very useful to find a view descriptor that automatically gives a view that is stable and whose direction is closer to three-quarter views. This kind

P.-P. Vázquez (✉)
Departament de Llenguatges i Sistemes Informàtics (LSI),
Universitat Politècnica de Catalunya, Barcelona, Spain
e-mail: ppau@lsi.upc.edu

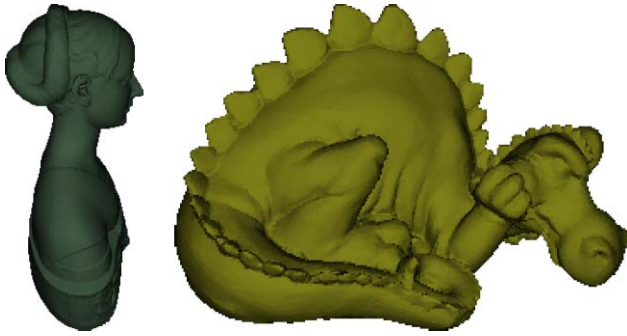


Fig. 1 Two examples of best views as selected by our algorithm

of view would be probably acceptable by most users. Our approach consists on seeking visually stable views. We will see that these lead in many cases to informative views. As a second result, in a high number of the studied objects, obtained views are quite close to three-quarter views.

The notions of stable view and the aspect graph have been studied for a long time [14]. Aspect graphs are a means of representing the complete three-dimensional geometry of complex objects as a set of prototypical two-dimensional views. It has been applied to several problems, such as object recognition, but, to the author's knowledge, view stability has not been used in order to search for good views without prior knowledge of the object (such as orientation). It has been argued recently that this might be a goal to seek for ([21]), as currently, no view descriptor gives the best results for all kinds of models.

In this paper we address the general problem of finding good representation for 3D models, useful for thumbnails creation of 3D models databases. Our goal is the selection of good views of three-dimensional objects with no user intervention. In contrast to previous approaches, we seek for a stable view, as stability is one of the common features for the preferred (i.e. canonical) views by people ([3]). Our view descriptor does not use geometric features explicitly. We use an image-based similarity metric over the depth maps to evaluate visual stability. As an assessment of our view descriptor, we will check if the viewing directions selected by our algorithm are close to three-quarter views and compare the results to the ones obtained in psychophysical experiments carried out by other researchers. We do not focus on the problem of finding the correct upright orientation of an object, which is, somewhat complementary to ours [11].

We have analyzed view stability under two parameters: visible polygons and depth images. We will show that this last approach yields better results, because it is not dependent on object tessellation, and that, in most of the analyzed cases, the resulting best views are close to three-quarter views. In Fig. 1 we show some images of the results we obtain with our algorithm.

The rest of the paper is organized as follows: Sect. 2 presents previous work. In Sect. 3 we present our two ap-

proaches for view stability measurement and search. It also shows the results obtained with our algorithm. Section 4 discusses the advantages and disadvantages of our method and compares to some special cases not handled properly by *viewpoint entropy* metric. Finally, Sect. 5 makes some concluding remarks and points some lines for future research.

2 Previous work

In recent years, a number of task-based view descriptors have been successfully introduced. However, none of those can be coined as general, and none of them *guarantees* obtaining views close to three-quarter views. Classical psychophysical experiments showed that three-quarter views are often preferred by humans as representative views for most objects. Some authors think that the main reason is because these are often stable [3].

2.1 Good viewpoint selection in computer graphics

In Computer Graphics, the problem of obtaining a representative view of an object is often addressed automatically, without prior knowledge on the geometry or orientation of the model. Facing the problem under these conditions is useful for many applications, such as the selection of thumbnails for objects databases, automatic camera positioning in CAD systems, automatic scene composition, and so on.

For surface-based models, view descriptors usually use the number of visible faces, its projection area, or its silhouette as parameters. View quality is measured using heuristic functions such as in [20] or information theory-based approaches [27]. Sbert et al. have recently published a state-of-the-art that presents many of those descriptors [23]. Polonsky et al. [21] also analyzed several relevant view descriptors and concluded that none of them can be coined as *universal*. They show that one can always find some object whose best view is *missed* by the developed metrics.

In volume rendering, view descriptors are slightly different, due to the nature of the data analyzed (voxels instead of faces, transparencies...), but they essentially use the same kind of formulations [4, 13, 17, 24]. In order to refine the search, other information such as the focus of attention might be used [29].

2.2 View stability and the aspect graph

Several studies have been carried out on human perception of 3D objects for different purposes such as object recognition.

The aspect graph is a tool that can be used to compute the viewpoints from which the maximum number of faces of a solid polyhedron are visible. The aspect graph [19] is

a representation of the different aspects an object can have from different views. The nodes of the graph correspond to a set of viewpoints from which the same topological entities are visible, and arcs correspond to transitions from one aspect to another caused by movement in viewpoint that results in a change in the visible topology of a viewed object or scene. There are some drawbacks to this representation. The most relevant is the complexity in generating aspect graphs and the potentially huge search spaces that may result, even for moderately simple objects. Another disadvantage is that the aspect graph deals with the visibility of object features and, hence, further constraints that may affect the selection of the viewpoint are not easily incorporated into the representation [22].

Weinshall and Werman [31] give a theoretical proof of equivalence between view stability and *view likelihood* (*view likelihood* measures the probability that a certain view of a given 3D object is observed) for a given aspect and show that this view can be computed from the aspect's autocorrelation matrix using Principal Components Analysis (PCA).

Tarr and Kriegman [25] have conducted psychophysical experiments investigating the influence of the aspect of an object on the quality of recognition. The experiments reveal that humans are indeed sensitive to certain types of visual events captured by the aspect graphs. One interesting consequence of other psychophysical experiments is that for many models, there exist a small number of views which seem to be preferred by most people. Palmer et al. [18] and Blanz et al. [3] call these views “canonical views” and show that they often correspond to the classical “three-quarter view” of the object. According to Blanz et al. [3], canonical views are stable, and expose as many salient and significant features as possible. The best interpretations in object recognition or reconstruction tasks are also coined canonical views. These canonical views serve to evaluate how *characteristic* or *general* a view is [30].

Freeman has exploited the *generic viewpoint assumption* to address shape from shading problems. The *generic viewpoint assumption* states that an observer is not in a special position relative to the scene [9, 10]. It is commonly used to disqualify scene interpretations that assume special viewpoints, thus, it can be used to avoid ambiguities [32]. However, human visual system does a good job at object recognition under ambiguities [26].

3 Stable views search

As we have already mentioned, several studies have shown that humans usually prefer three-quarter views for representing 3D objects. Moreover, several researchers claim that one of the main reasons why we prefer such views is because

they are stable. With this in mind, our first approach is looking for the most stable view, and check whether this can be useful as an object descriptor or not.

There are many ways of measuring view stability. The first one is the aspect graph [14]. However, for complex objects, the aspect graph is not practical, because the number of aspects that arise is very high (order of N^6 and N^9 for an N -faced polygon under orthogonal and perspective projections, respectively). Other approaches measure stability by grouping similar views. However, they work with bitmap images, instead of color ones. Although this is useful for object recognition, as they will compare with other bitmap images [8], it is clear that the rendered view is not the same if seen from the two different directions that might yield the same silhouette. Moreover, such methods usually have some information on object orientation. For the selection of a good view in computer graphics we think that extra information, such as the depth map, must be taken into account.

As we have seen, different ways of computing the stability of a view have been proposed in literature. Like in [12] we sample a set of views and group the similar ones. What we capture in such views and how we determine similarity will be key in how our view stability descriptor behaves. We have essayed two approaches: The first one consists of determining the visible polygons for each view and group views that *see* similar sets of polygons. The second, samples depth maps of the model at different positions and groups views that have similar depth maps. We will see that the second one gives better results, as the most stable view is often quite close to any of the three-quarter views of the object.

3.1 Polygon-based view stability

A stable view is the one that *sees* the same (or very similar) information than its neighborhood. One way of measuring such stability for complex objects would be sampling the viewing sphere at regular spacing (5 or 7.5 degrees have been previously used in literature [7]), and analyzing the set of polygons visible from each of these points. We build a bitmap of length N (N is the number of polygons of the model), where the i th bit is 1 if the polygon number i was visible from that view and 0 otherwise.

We can then group views with similar features, an approach followed by Ikeuchi and Kanade [12]. Given a complex object of several thousands of polygons, views that *see* similar sets of polygons are likely to show similar information. We may measure polygon set similarity by simply counting the number of differences between each descriptor or using some more sophisticated distance metric. However, this approach is highly sensitive to the number of polygons visible from each view, and this, in its turn, depends on the mesh tessellation, which might be irregularly subdivided for

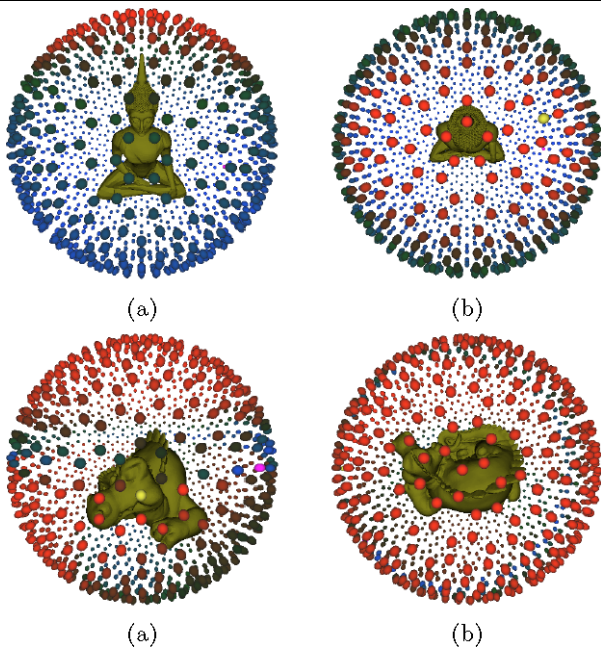


Fig. 2 Temperature map of polygon stability for two complex models. Warmer colors indicate more stable views and colder ones less stable ones. The *yellow spheres* (only clearly visible for the dragon) indicate the best viewing directions (most stable ones) while the *pink* ones show the worst ones

many different reasons. As a consequence, we will find stable views on regions that might not be very interesting, such as the bottom part of a teapot.

In Fig. 2 we show the temperature maps' result of the analysis of two different complex objects with our polygon stability metric. We may see that the more stable parts (warmer colors) appear near to the regions with a higher number of small polygons, and that symmetry is not taken into account. For the concrete case of the dragon (2), we may see the maximum stability view as one facing the head of the model, that gives little information on the remainder of the object.

There is a second reason why this approach might be not suitable. For symmetric objects, users may not distinguish which of the directions is more relevant. For example for a teapot, the users will select views that show the handle and reduce occlusions, but there is no preferred direction for the handle [3]. This is quite important because one would expect that for such a functional object users would choose different if they are hand-handed or left-handed. Moreover, the same experiments [3] also show that people will choose right or left views for different objects indistinctly. This is the main reason why we think a visual measure should not take into account the orientation to evaluate stability. Opposite to this, completely symmetric views should be considered as equally stable if the object rendered is symmetric, but not if only the silhouette is equal and the contents of the

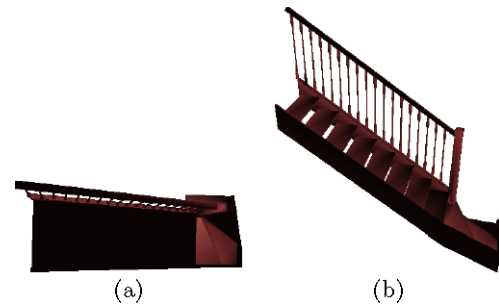


Fig. 3 Left shows a top view of a stair model; with this image, neither the color map nor the normal map will help the user to understand that the object is a stair (see on the right)

image differs. This leads to our second stability measure we propose, dubbed *Depth-Based Visual Stability*.

3.2 Depth-based visual stability

As found by Cleju and Saupe [7], in order to search for view similarity, image-based metrics behave better than pure geometry-based ones. This overcomes the problem of having potentially irregularly tessellated objects. The next question is then what images could encode the geometry of an object. Three obvious possibilities arise:

- *Shaded images*: If we use a constant color per object and a light placed at the observer or slightly above yields views that can be used to compare to each other. However, this has the problem that faces with the same normal but placed at different depths will be lit equally and therefore *represent* the same information (see Fig. 3).
- *Normal maps*: A normal map encodes information on the shape of the object; however, like the previous case, parallel faces placed at different distances will look like the same. Moreover, symmetric normal vectors with respect to the viewing direction would look different. Thus, a planar object with a normal not aligned with the view direction will look different if rotated with respect to the viewing direction (like in Fig. 3).
- *Depth maps*: It provides information of the shape of the object, due to the variation of the depth for the curved faces. It hides color, but we are more concerned to the shape than the color. Thus, this is a good candidate.

Depth maps seem to be the best option, and thus it is the approach we took.

3.2.1 Algorithm

Our method first samples the bounding sphere of an object at different positions. In this case, we use as a basis the subdivision of an icosahedron. This leads to positions placed at a minimum distance of 8.9 degrees between each other. Although we could further subdivide and get a finer sampling

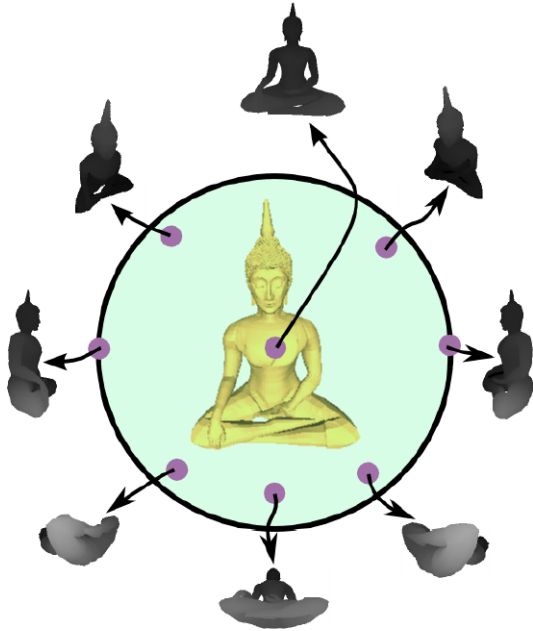


Fig. 4 Depth images of the buddha model

```

vector<int> function stability(float minDist, set views)
{
    vector<int> viewsStab
    for each view i
        int numViews ← 0
        for each view j
            if i ≠ j
                if (distance(views(i), views(j)) < minDist)
                    numViews++
        viewsStab.push_back(numViews)
    return viewsStab
}

```

Fig. 5 Stability determination algorithm

rate, this would generate too much views and the current subdivision gives good results. In Fig. 4 we illustrate this process. In [28] the authors showed that best views were accurately found with images of 256×256 and even 128×128 pixels wide. For the experiments here, we used images of 256×256 , although if speed is a must, we can perform the comparisons with smaller images.

Once we have the set of views, our algorithm selects the most stable one. Like in [12], we group similar views, by comparing each view to the others (as depicted in Fig. 5). The most stable one will be the one that has a larger number of views similar up to a threshold to it.

The key point is the similarity measurement. Classical image-based metrics such as Mean Square Error will fail to compare depth maps because they are not sensitive to rotation or translation of the images. We need a measure of

similarity that can cope with differently oriented images, especially for handling the case of symmetric objects. We are going to measure the distance between depth maps using a Kolmogorov complexity result, named Normalized Compression Distance, which has been recently used for color image comparison for medical images [28]. The following section introduces the background.

3.3 Normalized compression distance

Normalized Compression Distance is a universal metric of distance between sequences, that is, it minorizes every computable similarity distance up to an error that depends on the quality of the compressor's approximation of the true Kolmogorov complexities of the files concerned. We introduce here some concepts on Kolmogorov complexity (Li and Vitányi's book [15] is a good reference).

The *Kolmogorov complexity* ($K(x)$) of a string x is the length of the shortest binary program to compute x on a universal computer (such as a universal Turing Machine). Thus, $K(x)$ denotes the number of bits of information from which x can be computationally retrieved. Hence, $K(x)$ is the lower-bound of what a real-world compressor can possibly achieve. The *conditional Kolmogorov complexity* $K(x|y)$ of x relative to y is the length of a shortest program to compute x if y is provided as an auxiliary input. Both Kolmogorov complexity and conditional Kolmogorov complexity are machine independent up to an additive constant.

Bennet et al. [2] define the *information distance* between two, not necessarily equal length binary strings as the length of the shortest program that can transform either string into the other one, both ways. The information distance is a metric. Li et al. [16] present a normalized version of information distance dubbed *similarity metric*, defined as:

$$d(x, y) = \frac{\max\{K(y|x), K(x|y)\}}{\max\{K(y), K(x)\}}. \quad (1)$$

The authors also prove that this metric is universal (two files of whatever type similar with respect to a certain metric are also similar with respect to the similarity metric). Being Kolmogorov complexity not computable, it may be approximated with the use of a real-world compressor, leading to the *Normalized Compression Distance* (NCD):

$$\text{NCD}(x, y) = \frac{C(xy) - \min\{C(x), C(y)\}}{\max\{C(x), C(y)\}}, \quad (2)$$

where function $C(F)$ is the size of the compression of a certain file F , and xy is the concatenation of files x and y . Although the similarity metric has values in $[0..1]$, NCD values are usually in the range of $[0..1.1]$, due to compressor imperfections. NCD has been used for applications such as language classification and handwriting recognition [6].

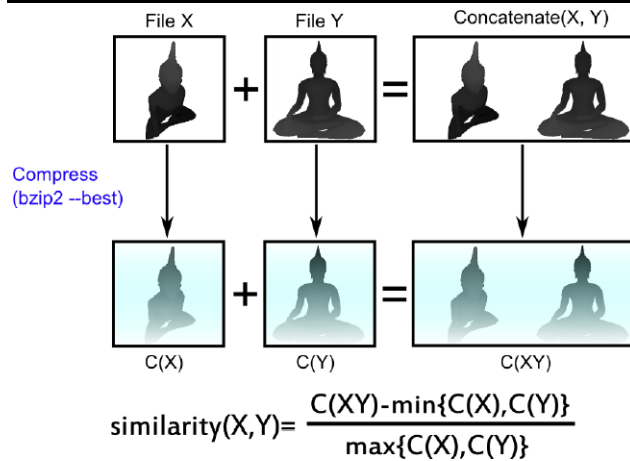


Fig. 6 Comparison of two files using Normalized Compression Distance

As demonstrated by Cilibrasi and Vitányi [6], compressors must fulfill some conditions in order to be used for computing the Normalized Compression Distance. Fortunately, most of them do, such as stream-based (*zlib*), block-based (*bzip*), and statistical (*PPMZ*) compressors. Cebrián et al. [5] also show that the size of files matters, and may influence the efficiency of the comparison processes. In the case of *bzip2*, the *best* option works properly for files up to 900 KB before being compressed. Larger sizes make the comparison processes less effective. Based on this, we use *bzip2* compressor on our PGM (gray scale) generated images. The images we analyze are of 256×256 , as they have been shown to be large enough for comparison [28] purposes. Thus, the size of the concatenated file never exceeds 900 KB.

The similarity between two depth maps is computed as depicted in Fig. 6: First the two depth maps are concatenated, and then, we run *bzip2* with the `--best` option to the files. Then the distance is computed by applying (2).

Best view selection is quite fast, but the concatenation and compression processes are costly because we generate N^2 images, where N is the number of analyzed views. For 320 views, we require roughly 10 minutes for the calculation if we use images of 256×256 . For images of 128×128 the computing time drops to less than three minutes. If we require a very quick process, we should implement some kind of hierarchical or adaptive approach. On the positive side, the most costly process is largely independent of the complexity of the model, as it is computed on the generated images only.

3.4 Results

We have run our program on several models. Some of them are shown in Fig. 7. We analyze the results by comparing to the ones obtained in [3]. A quick look shows that none of the

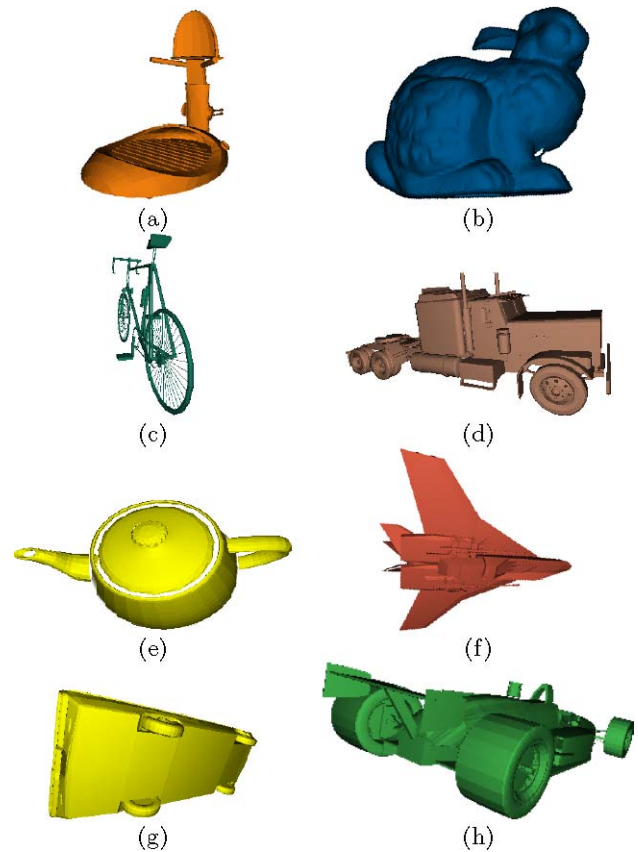


Fig. 7 Several examples of best views for different models, as selected by our algorithm. Except for image e, the selected views are acceptable although in some cases, such as b or f the camera should move a little far away from the ground plane

selected views is accidental (with low probability, see also the following section and Fig. 8). Moreover, most of them are quite close to three-quarter views. Now we proceed to the detailed analysis.

In [3] one of the models is an animal (a cow) and the selected views are in front of it, with a low displacement over (majority) or under the ground, although further to the Z axis than the bunny (Fig. 8b). Vehicles are usually observed from over the ground and in front of them (except for planes, that can be both). Therefore d and f would be correct, but not g (which is however quite close to a three-quarter view). Figures 8, c and h, might be not acceptable, but they are quite common for enthusiasts of bicycles or Formula 1 races. For human faces, we prefer a range of front directions, up to the side of the face, at least for non-textured models, which is again coherent with the results of the Laurana model in Fig. 1. Finally, the teapot result in Fig. 7e is acceptable too, as people usually select side views where no occlusion appears.

4 Discussion

We developed a view descriptor based on a metric that evaluates stability over depth maps. In all the models we tested, it completely avoids accidental views. Furthermore, the results obtained are, in many cases, consistent with psychophysical experiments' results. Compared to other view descriptors presented in Computer Graphics, our proposal is less sensitive to polygon count, and therefore solves some cases where viewpoint entropy would yield accidental views such as the ones shown in Fig. 8. These images show best views for certain models where *viewpoint entropy* gives views aligned with the axis that show a higher number of polygons but hides some of the dimensions of the object and would therefore not been chosen by an observer. Figures 8a and 8b show the torus model. Of course in Fig. 8a the object projects to a larger area, and shows a higher number of polygons, but in Fig. 8b our algorithm gives a view close to a three-quarter and would be probably preferred by most users. In Fig. 8c and 8d the dragon is shown from above and front, respectively. In the second case our algorithm slightly shows the

dragon over the Z axis. Finally, Figs. 8e and 8f show the martini glass. Our algorithm shows it from slightly over the ground, which is consistent on what users select for familiar objects, but the top view is avoided.

We have presented a new approach that seeks for a good view by analyzing the visual stability of a set of views of an object. We think our proposal is useful for building thumbnails of objects for 3D databases or as initial camera placement for navigation applications or CAD systems. The main disadvantage of our approach is the cost of the selection process which requires several minutes for the files concatenation process, as our implementation currently compares each view with the remaining ones. An adaptive or hierarchical approach with times of seconds would be enough for most applications.

The following table analyzes some of the most relevant applications and discusses the possible utility of our new view descriptor:

Thumbnails of 3D models databases: In order to reduce the browsing time and the avoiding downloading wrong models, it is very useful to add thumbnails to 3D models databases. Usually, the representative image is selected by the modeler/maintainer of the database. For large databases, an automatic selection of such views is desirable. Our visual stability descriptor may do a good job for this purpose, as other previously presented metrics could be less descriptive.

Automatic camera placement in CAD systems: We think our view descriptor yields informative views for objects, so it will be useful as initial camera for CAD systems. If on-the-fly camera selection is required, other approaches that can be accelerated through an adaptive algorithm (like in [28]) are more suitable.

Image-Based Modeling: Building an image-based representation of a 3D model requires not only the selection of an informative set of views, but also the completeness of the representation. Our view descriptor could be an starting point. Andujar et al. [1] address this problem for an impostor set construction by starting from a regular set of 20 views and further refinement on the initial directions (this refinement could also be handled by our descriptor).

Medical models: Several measures have been presented for good view selection of medical models [4, 13, 24, 28]. For medical applications it seems very useful a previous knowledge on the information desired, as this may allow its use for surgery preparation or training [17] or to perform cut-away views [29].

Object recognition: For the purpose of object recognition, the initial data is not only a model to be rendered, but a database of models in order to compare. Here, like for Image-Based Modeling approaches, it is not only necessary to obtain a good view, but a set of representative views to be compared. In general, the initial orientation of the

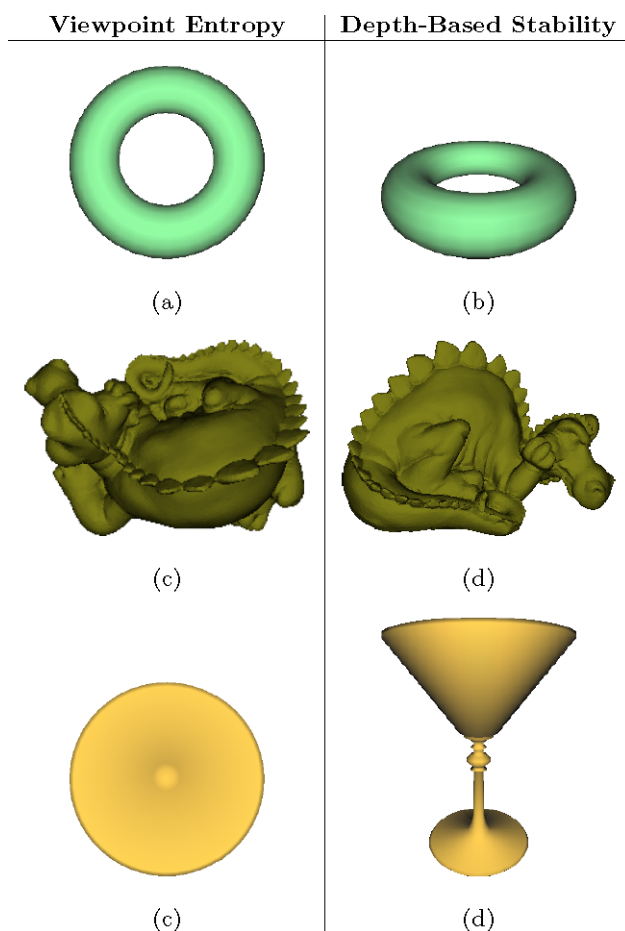


Fig. 8 Comparison of best views as selected by viewpoint entropy (left) and our depth-based stability measure (right). Note that in both cases our metric behaves better than viewpoint entropy

model is required, although some information may be extracted using PCA analysis. One may use stable views for the comparison process, but these need not include depth information, and the silhouette (a bitmask of the object) is usually enough. Our visual stability measure might be useful, but we did not try it in this context.

5 Conclusions and future work

We have analyzed different ways of computing stable views for three-dimensional objects and came out with a proposal of view descriptor that produces informative views that in most cases are close to three-quarter views. Our process is completely automatic, not requiring user intervention and no knowledge of the object being inspected.

Although this might not be the *definitive* view descriptor, it yields informative views and its behavior is often coherent with previous psychophysical experiments.

In the future, we would like to investigate the incorporation of other features such as occlusion contours.

Concerning the efficiency, it would also be interesting to have a faster algorithm. Some issues that would improve performance are the reduction of image size (maybe up to 64×64) and the implementation of some adaptive or hierarchical approach. This would reduce the number of views to be further processed. For thumbnail selection this might not be important, but other applications such as CAD applications would benefit from an automatically chosen initial camera configuration at loading time. We have also studied the use of faster compressors, such as LZO (*overhumer.com*), but the results we obtained with the fast compression version of the algorithm are not as good as with *bzip2* with the *-best* option.

Acknowledgements Supported by TIN2007-67982-C02-01 (Spanish Government). Thanks to the models providers.

References

- Andujar, C., Boo, J., Brunet, P., Gonzalez, M.F., Navazo, I., Vázquez, P.P., Vinacua, A.: Omnidirectional relief impostors. *Comput. Graph. Forum* **26**(3), 553–560 (2007)
- Bennett, C., Gacs, P., Li, M., Vitanyi, P., Zurek, W.: Information distance. *IEEE Trans. Inf. Theory* **44**, 1407–1423 (1998)
- Blanz, V., Tarr, M., Bülthoff, H.: What object attributes determine canonical views? *Perception* **28**, 575–599 (1999)
- Bordoloi, U., Shen, H.W.: View selection for volume rendering. In: *IEEE Visualization*, pp. 487–494 (2005)
- Cebrián, M., Alfonsaca, M., Ortega, A.: The normalized compression distance is resistant to noise. *IEEE Trans. Inf. Theory* **53**(5), 1895–1900 (2007)
- Cilibrasi, R., Vitanyi, P.: Clustering by compression. *IEEE Trans. Inf. Theory* **51**(4), 1523–1545 (2005)
- Cleju, I., Saupe, D.: Evaluation of supra-threshold perceptual metrics for 3d models. In: *Proc. 3rd Symposium on Applied Perception in Graphics and Visualization*, pp. 41–44. ACM, New York (2006)
- Cyr, C.M., Kimia, B.B.: A similarity-based aspect-graph approach to 3d object recognition. *Int. J. Comput. Vis.* **57**(1), 5–22 (2004)
- Freeman, W.T.: Exploiting the generic view assumption to estimate scene parameters. In: *Proc. 4th Intl. Conference on Computer Vision*, pp. 347–356. IEEE, Berlin (1993)
- Freeman, W.T.: Exploiting the generic viewpoint assumption. *Int. J. Comput. Vis.* **20**(3), 243–261 (1996)
- Fu, H., Cohen-Or, D., Dror, G., Sheffer, A.: Upright orientation of man-made objects. *ACM Trans. Graph.* **27**(3), 1–7 (2008)
- Ikeuchi, K., Kanade, T.: Automatic generation of object recognition programs. *Proc. IEEE* **76**(8), 1016–1035 (1988)
- Ji, G., Shen, H.W.: Dynamic view selection for time-varying volumes. *IEEE Trans. Vis. Comput. Graph.* **12**(5), 1109–1116 (2006)
- Koenderink, J.J., Doorn, A.J.V.: Photometric invariants related to solid shape. *Opt. Acta* **27**(7), 981–996 (1980)
- Li, M., Vitanyi, P.M.: *An Introduction to Kolmogorov Complexity and Its Applications*. Springer, Berlin (1993)
- Li, M., Chen, X., Li, X., Ma, B., Vitanyi, P.: The similarity metric. *IEEE Trans. Inf. Theory* **50**(12), 3250–3264 (2004)
- Mühler, K., Neugebauer, M., Tietjen, C., Preim, B.: Viewpoint selection for intervention planning. In: *EG/IEEE-VGTC Symposium on Visualization*, pp. 267–274 (2007)
- Palmer, S., Rosch, E., Chase, P.: Canonical perspective and the perception of objects. In: *Attention and Performance IX*, pp. 135–151 (1981)
- Plantinga, H., Dyer, C.R.: Visibility, occlusion, and the aspect graph. *Int. J. Comput. Vis.* **5**, 137–160 (1990)
- Plemenos, D., Benayada, M.: Intelligent display in scene modeling. New techniques to automatically compute good views. In: *International Conference GRAPHICON'96*
- Polonsky, O., Patanè, G., Biasotti, S., Gotsman, C., Spagnuolo, M.: What's in an image? *Vis. Comput.* **21**(8–10), 840–847 (2005)
- Roberts, D., Marshall, A.: Viewpoint selection for complete surface coverage of three-dimensional objects. In: *Proc. of the British Machine Vision Conference* (1998)
- Sbert, M., Plemenos, D., Feixas, M., Gonzalez, F.: Viewpoint quality: Measures and applications. In: Neumann, L., Sbert, M., Gooch, B., Purgathofer, W. (eds.) *Computational Aesthetics in Graphics, Visualization and Imaging* (2005)
- Takahashi, S., Fujishiro, I., Takeshima, Y., Nishita, T.: A feature-driven approach to locating optimal viewpoints for volume visualization. In: *IEEE Visualization*, pp. 495–502 (2005)
- Tarr, M., Kriegman, D.: What defines a view? *Vis. Res.* **41**(15), 1981–2004 (2001)
- Todd, J.T.: The visual perception of 3d shape. *TRENDS Cogn. Sci.* **8**(3), 115–121 (2004)
- Vázquez, P.P., Feixas, M., Sbert, M., Heidrich, W.: Automatic view selection using viewpoint entropy and its application to image-based modeling. *Comput. Graph. Forum* **22**(4), 689–700 (2003)
- Vázquez, P.P., Monclús, E., Navazo, I.: Representative views and paths for volume models. In: *Smart Graphics*, pp. 106–117 (2008)
- Viola, I., Feixas, M., Sbert, M., Gröller, M.E.: Importance-driven focus of attention. *IEEE Trans. Vis. Comput. Graph.* **12**(5), 933–940 (2006)
- Weinshall, D., Werman, M.: A computational theory of canonical views. In: *Proc. APRA IU Workshop* (1996)
- Weinshall, D., Werman, M.: On view likelihood and stability. *IEEE Trans. Pattern Anal. Mach. Intell.* **19**(2), 97–108 (1997)
- Yuille, A.L., Coughlan, J.M., Konishi, S.: The generic viewpoint constraint resolves the generalized bas relief ambiguity. In: *Conference on Information Sciences and Systems*. Princeton University Press, Princeton (2000)



Pere-Pau Vázquez is an Associate Professor at the Universitat Politècnica de Catalunya in Barcelona, the Department of Computer Science (Llenguatges i Sistemes Informàtics). He has the Computer Engineer degree since 1999 and obtained a Ph.D. in Computer Science in 2003. His main interests are the application of information theory and Kolmogorov Complexity to computer graphics, illustrative visualization, and GPU-based rendering.