

# Surface Regions of Interest for Viewpoint Selection

George Leifman  
Technion

gleifman@tx.technion.ac.il

Elizabeth Shtrom  
Technion

lizas@tx.technion.ac.il

Ayellet Tal  
Technion

ayellet@ee.technion.ac.il

## Abstract

While the detection of the interesting regions in images has been extensively studied, relatively few papers have addressed surfaces. This paper proposes an algorithm for detecting the regions of interest of surfaces. It looks for regions that are distinct both locally and globally and accounts for the distance to the foci of attention. Many applications can utilize these regions. In this paper we explore one such application—viewpoint selection. The most informative views are those that collectively provide the most descriptive presentation of the surface. We show that our results compete favorably with the state-of-the-art results.

## 1. Introduction

Detecting the interesting regions of an object has attracted a lot of attention in computer vision. Most of the work has concentrated on images and videos [7, 9, 11, 27]. Fewer attempts were made to define the interesting regions of surfaces, which is the focus of this paper.

Many problems in computer vision and computer graphics benefit from the detection of the most interesting regions of surfaces. Examples include face recognition [18], similarity [4], alignment [4], simplification [24], icon generation [24], abstraction [29], and viewpoint selection [17].

What are these interesting regions? Lee et al. [17] define a measure of surface saliency using a center-surround operator on Gaussian-weighted mean curvatures. Measures of importance have been defined also by [4, 6, 19] and others. These algorithms detect regions where the curvature of a surface is inconsistent with its immediate surroundings. Thus, they take the human vision’s tendency to be drawn to differences into account. In [24] surface distinctiveness is based on the similarity between a given surface and similar objects in its class.

We also look for region distinctness (Figure 1). However, unlike prior approaches, which focus on local distinctness, we take into consideration also global distinctness. This accounts for 3D textures, where local distinctness is high, while global distinctness is low. Additionally, we con-



Figure 1. The left image shows the regions of interest, where red is the most interesting and blue is the least. The other images present the surface from the two most descriptive viewpoints, as calculated by our algorithm.

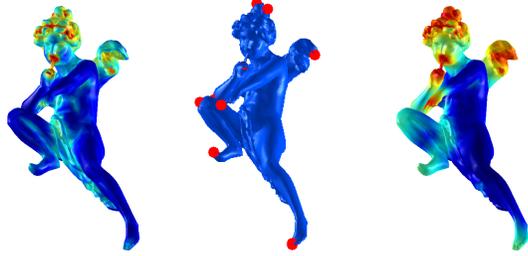
sider the fact that visual forms may possess one or several centers of gravity about which the form is organized. Therefore, regions that are close to the foci of attention are more interesting than faraway regions.

We propose a novel algorithm that detects regions of interest on surfaces by realizing the considerations above. To capture distinctness, we discuss a vertex descriptor that characterizes the geometry in the neighborhood of a vertex. Moreover, we introduce an algorithm that detects surface’s extremities, which are typically distinct. To take distance to foci into account, we show how to adjust the distinctness by computing *patch association*.

We demonstrate the utility of our regions of interest in viewpoint selection. The goal is to automatically select the camera position from which the most informative and intuitive view of the shape is seen. In many applications, such as the creation of thumbnails for huge repositories or catalogs of 3D models, it is necessary to automatically capture an informative image of an object. Good images of 3D objects can also be used for various computer vision problems, such as shape recognition or classification.

The problem of defining representative 2D views of 3D objects received attention both in computer vision [8, 18, 22] and in computer graphics [16, 17, 23, 28]. We show that our scheme outperforms the state-of-the-art methods.

Our contributions are hence twofold. First, we propose



(a) Distinctness (b) Extremities (c) Patch association  
Figure 2. Detection of regions of interest: algorithm outline

a novel algorithm for detecting the regions of interest of a surface (Sections 2-3). Second, we present an algorithm for viewpoint selection, which utilizes our regions of interest (Sections 4-5).

## 2. Detection of regions of interest

Given a surface, our goal is to compute its regions of interest. We assume that the surface is given as a triangulated mesh that consists of vertices and faces.

Since people are drawn to differences, we say that a region is *interesting* if it differs from other regions of the mesh. Therefore, we look for vertices that are distinct in their appearance.

In addition, in [14] it is found that *extremities* are considered salient by humans. Indeed, extremities are distinct. Hence, we also look for extreme vertices of meshes.

Finally, our goal is to look for regions of interest, rather than for isolated vertices. This consideration follows the human tendency to group close items together. Therefore, we introduce *patch association*, which regards the regions near the foci of attention as more interesting than faraway regions.

Hereafter, we present algorithms for realizing each of the above considerations: (1) vertex distinctness (2) shape extremities and (3) patch association. See Figure 2.

### 2.1. Vertex distinctness

We look for vertices whose geometry (i.e., appearance) is unique. This is done by computing, for each vertex, a *descriptor* that characterizes its shape and comparing the descriptors. A vertex is distinct if its descriptor is dissimilar to all other vertex descriptors of the mesh.

**Vertex descriptor:** We seek a descriptor that has good expressive power of the local shape geometry and is invariant to rigid transformations. Various local descriptors were recently proposed for shape-based retrieval [2, 25, 26], each with its benefits and drawbacks. We have examined some of them [1, 10, 15] and found that the best results were achieved using *spin images* [10], briefly described below.

The spin image is a 2D histogram that encodes the density of oriented points, in our case mesh vertices. Two cylindrical coordinates are defined for each vertex: the radial coordinate  $r$ , which is the perpendicular distance to the normal, and the elevation coordinate  $h$ , which is the distance to the tangent plane. A spin image is created for a vertex by quantizing  $h$  and  $r$ , creating bins, and counting the number of vertices in each bin. In our implementation, the geometric width of the bins, the *bin size*, is set to the median of the length of the mesh edges. We use a  $16 \times 16$  histogram.

**Dissimilarity measure:** We seek a dissimilarity measure, which is robust to small changes in the mesh, such as noise or different triangulations. In addition, the computation should be fast enough, since a quadratic number of comparisons need to be performed.

We use the *diffusion distance*, which models the difference between two histograms as a temperature field and considers the diffusion process on the field [20]. The integration of a norm on the diffusion field over time is used as a dissimilarity measure between the histograms. For computational efficiency, a Gaussian pyramid is used to discretize the continuous diffusion process.

The diffusion distance  $D(h_1, h_2)$  is defined as:

$$D(h_1, h_2) = \sum_{l=0}^L k(|d_l|), \quad (1)$$

where

$$d_0 = h_1 - h_2$$

$$d_l = [d_{l-1} * \phi(\sigma)] \downarrow_2, l = 1, \dots, L$$

are different layers of the pyramid. The notation  $\downarrow_2$  denotes half size down-sampling.  $L$  is the number of pyramid layers and  $\sigma$  is a constant standard deviation for the Gaussian filter  $\phi$  (we use  $L = 5$  and  $\sigma = 0.5$ ). In our implementation  $k(\cdot)$  is the  $L_1$  norm, which makes the diffusion distance a true metric.

Note that there are other well-known distances used in the literature. We experimented with  $L_2$ ,  $\chi^2$ , Jeffrey distance function, and the Earth Mover’s Distance (EMD). We found that the first three are more prone to noise than the diffusion distance, since they are not cross-bin distances, whereas EMD is computationally more expensive.

**Distinctness computation:** Finally, we need to compute a distinctness value for each vertex, given the dissimilarity values calculated above.

1. SINGLE-SCALE COMPUTATION: As stated above, a vertex  $v_i$  is distinct when the distance between the descriptors  $D(h(v_i), h(v_j))$  is high  $\forall j$ . This consideration, however, is insufficient, since the geodesic distances between the vertices are important as well. This is so, since similar vertices

that are far away indicate a 3D texture. Thus, a vertex is distinct when the vertices similar to it are nearby and less distinct when the resembling vertices are far away. Hence, the dissimilarity measure should be proportional to the difference in appearance and inverse proportional to the geodesic distance.

Let  $GeodDist(v_i, v_j)$  be the geodesic distance between vertices  $v_i$  and  $v_j$ , normalized by the largest geodesic distance on the mesh. Inspired by [7], the dissimilarity measure between  $v_i$  and  $v_j$  is defined as:

$$d(v_i, v_j) = \frac{D(h(v_i), h(v_j))}{1 + c \cdot GeodDist(v_i, v_j)}, \quad (2)$$

where  $c = 3$  in our implementation.

Vertex  $v_i$  is considered distinct when it is highly dissimilar to all other vertices. In practice, it suffices to consider the  $K$  most similar vertices, since if they are highly different from  $v_i$ , then clearly all vertices are highly different from  $v_i$  as well. Therefore, for every vertex  $v_i$ , we search for the  $K$  most similar vertices  $\{v_k\}_{k=1}^K$ . Vertex  $v_i$  is distinct when  $d(v_i, v_k)$  is high  $\forall k \in [1, K]$ . The single-scale distinctness value of vertex  $v_i$  is defined as:

$$\mathcal{D}(v_i) = 1 - \exp\left\{-\frac{1}{K} \sum_{k=1}^K d(v_i, v_k)\right\}, \quad (3)$$

where  $K$  is 5% of the number of mesh vertices.

**2. MULTI-SCALE COMPUTATION:** Vertices that belong to 3D textures are likely to have similar vertices at multiple scales. Conversely, distinct vertices may have similar vertices at a few scales, but not at all of them. Therefore, we incorporate multiple scales to further decrease the importance of vertices that belong to 3D textures.

We simplify the given mesh of  $F$  faces to meshes having  $\{\frac{F}{2}, \frac{F}{4}\}$  faces [5]. We then calculate the distinctness at these three scales  $(F, \frac{F}{2}, \frac{F}{4})$ .

The multi-scale distinctness value  $\mathcal{D}$  is the average of the values of distinctness at all three scales, where the values at the different scales are mapped back to the input mesh. See Figure 2(a) for the final distinctness map.

## 2.2. Shape extremities

We aim at detecting the extremities of limb-like objects. Given an object, we need to first determine whether it has a limb-like structure, and then find its extremities, if need be (Figure 2(b)). This is done in three steps, as follows.

**1. MDS TRANSFORMATION:** To determine the structure of an object and its extremities, we need to ignore the object’s pose. We do it by transforming the mesh using *multi-dimensional scaling (MDS)* [3], such that the Euclidean distances between points on the transformed mesh become



(a) The MDS-transformed mesh (b) Mesh extremities  
Figure 3. Computation of the shape extremities

similar to the geodesic distances between their corresponding points on the input mesh. Hence, the folded parts of the objects are “straightened” (Figure 3(a)).

**2. DETERMINING THE OBJECT’S STRUCTURE:** To decide whether an object has a limb-like structure, we note that the volume of a “round” shape is similar to that of its convex hull, whereas the volume of a (straightened) limb-like object and that of its convex hull, differ. Therefore, we utilize a simple procedure, which works well in practice. We compute the volume of the object  $V_O$  and the volume of the convex hull of the MDS-transformed object  $V_{CH}$ . If  $\frac{V_{CH}}{V_O} > 1.5$ , we conclude that the object is limb-like.

**3. DETECTING EXTREMITIES:** Intuitively, an extreme vertex is a vertex that resides on the “tips” of the object. Specifically, we say that a vertex is extreme if it satisfies two conditions: It resides on the convex-hull of the MDS-transformed mesh and it is a local maximum of the sum of the geodesic distance functional [12]. The latter condition can be formally expressed as follows. Given a vertex  $v$ , let  $N_v$  be its set of neighboring vertices. The local condition that an extreme vertex  $v$  should satisfy is that  $\forall v_n \in N_v$ :

$$\sum_{v_j \in S} GeodDist(v, v_j) > \sum_{v_j \in S} GeodDist(v_n, v_j). \quad (4)$$

This definition derives an algorithm for computing the extreme vertices (Figure 3). Given a mesh  $S$ , the algorithm first computes the convex hull of its MDS-transformed mesh and then finds among the vertices of the convex hull those that satisfy Equation (4).

## 2.3. Patch association

Visual forms may possess one or several centers of gravity about which the form is organized. Therefore, regions that are close to the foci of attention should be more interesting than faraway regions.

We model this effect as follows. We define a fraction (20% in our implementation) of vertices with the highest distinctness values as *focus points*. Let  $GeodFoci(v_i)$  be the geodesic distance between vertex  $v_i$  and its closest focus point, normalized to the range  $[0,1]$ . Let  $\mathcal{D}_{foci}(v_i)$  be the distinctness value of this focus point. The association of vertex  $v_i$  is defined as:

$$\mathcal{A}(v_i) = \mathcal{D}_{foci}(v_i) e^{-\frac{GeodFoci^2(v_i)}{2\sigma^2}}, \quad (5)$$

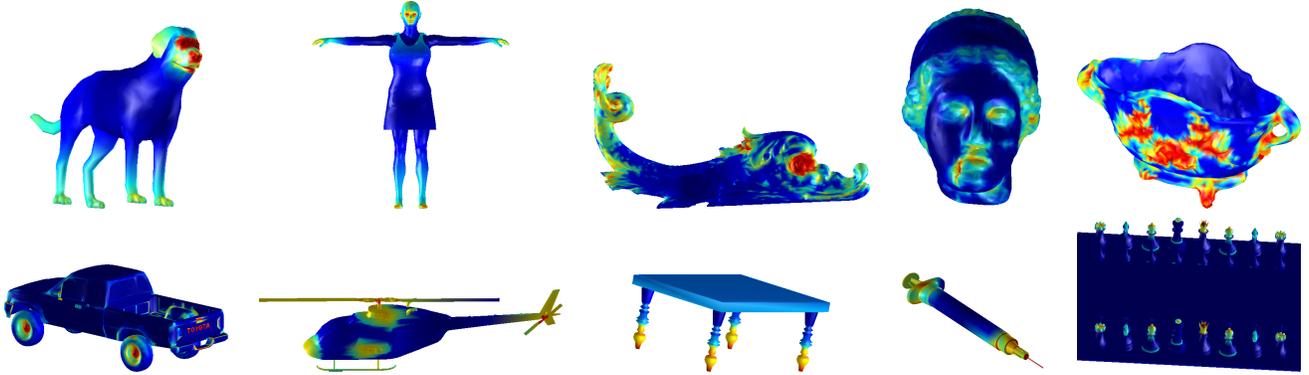


Figure 4. Regions of interest of representative objects, as computed by our algorithm

where  $\sigma = 0.05$ .

Similarly, an extreme vertex is considered a focus point. Therefore, for each mesh vertex  $v_i$ , we compute its geodesic distance to the closest extreme vertex  $GeodExt(v_i)$ , normalized to  $[0,1]$ . The extremity of  $v_i$  is defined as:

$$E(v_i) = e^{-\frac{GeodExt^2(v_i)}{2\sigma^2}}. \quad (6)$$

Naturally, for non limb-like objects,  $E(v_i) = 0 \forall i$ .

Finally, we integrate the results obtained by the different phases of the algorithm (Figure 2(c)). The degree of interest  $\mathcal{I}(v_i)$  of vertex  $v_i$  is defined as the maximum of the distinctness and the extremity of the vertex, taking into account patch association:

$$\mathcal{I}(v_i) = \max\left(\frac{\mathcal{D}(v_i) + \mathcal{A}(v_i)}{2}, E(v_i)\right). \quad (7)$$

### 3. Regions of interest: results

We ran our algorithm on a broad set of object categories, including animals (10 models), humans (6), creatures (7), sculptures (8), ancient artifacts (5), cars (4), other vehicles (7), furniture (8), tools (7) and miscellaneous accessories and instruments (17). The number of objects per class differs, depending on the object variance in the class.

Figure 4 shows the results of our algorithm for a representative model per class. It can be seen that our algorithm usually detects the “expected” regions of interest. For example, for the dog, our algorithm finds the facial features, the feet, and the tail interesting, where the facial features are the most interesting. Similarly, for the chess set, the chess pieces are more interesting than the board, yet the unique pieces, like the kings and the queens, are more interesting than the regular pawns, since there are many of them.

We also compare our results to works that explore saliency or class-distinctiveness of surfaces. Since their implementations are unavailable, we ran our code on the same models given in these papers and show the results side by side. Figure 5 compares a couple of results from [17] with

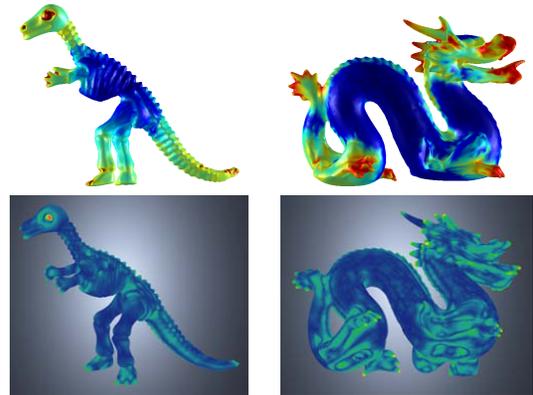


Figure 5. A comparison of our results (top) with those of [17] (bottom). Our results are less influenced by local changes of the curvature, when they happen frequently. Moreover, thanks to patch association, our algorithm detects large salient regions.

ours. Our algorithm is less influenced by local changes of the curvature, when they happen frequently. For instance, the long body of the dragon is not considered interesting by our algorithm, whereas many small regions on it are salient according to [17], since the local changes are large. In addition, the lack of patch association in [17] is noticeable. Conversely, our algorithm detects large regions, rather than small, more isolated ones. For instance, the whole dinosaur head, and not only its facial features, are interesting (yet, the facial features are more interesting).

Figure 6 compares our results with those of [4]. While in [4], the saliency is more distributed over the surface, as it detects local changes of curvature, our results are more focused on several regions of interest. For instance, the distinct facial features are emphasized in all three examples. Moreover, the shape extremity consideration is noticeable in the animals’ feet and hump.

Figure 7 compares our results with those of [24]. It is important to note that the goals are different. There, the goal is to find regions that distinguish a shape from objects in a different class, while we aim at locating the regions of interest

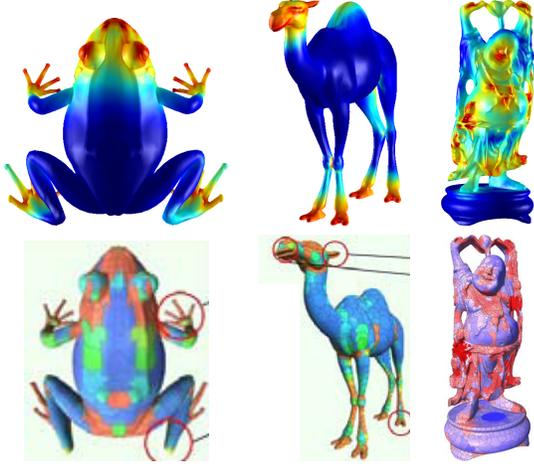


Figure 6. A comparison of our results (top) with those of [4] (bottom). The regions of interest detected by our algorithm are more focused on the head, feet, and hump.

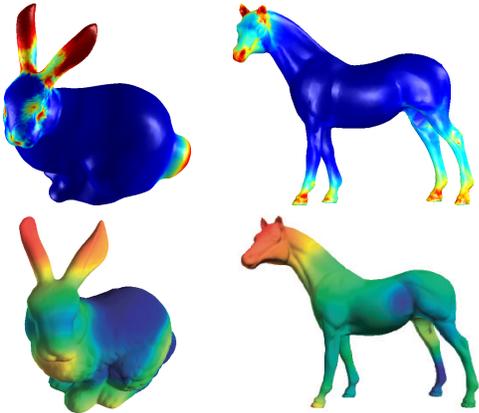


Figure 7. A comparison of our results (top) with those of [24] (bottom). Notice the differences in the facial features, the legs and the tail.

regardless of the class it belongs to. Consequentially, while in [24] the whole head of the horse is marked, we detect mostly the facial features, and similarly for the horse’s legs. In the bunny, both algorithms mark the ears as important. However, while the bunny’s tail was found to be interesting by our algorithm, it is not distinctive according to [24].

**Complexity analysis:** The complexity of the distinctness computation depends on the computation of the K-nearest neighbors and on the geodesic distance computation, which is  $O(n^2 \log n)$ , where  $n$  is the number of vertices. Finding the extremities depends on the MDS computation, which is  $O(n^2)$ . The other operations are convex hull construction  $O(n \log n)$  and volume computation  $O(n)$ . In the actual implementation, the time-consuming steps are performed on simplified meshes. The running time, for instance for the Buddha that has 540,000 vertices, is 102 seconds.

## 4. Viewpoint selection

Given a surface, our aim is to automatically determine the set of the most informative views, which jointly describe the surface well. The key idea of the algorithm is to maximize the area of the viewed regions of interest (ROI). The algorithm consists of four steps, outlined in Figure 8 and explained thereafter.

1. Generate candidate viewpoints.
2. Compute the viewed ROI from each candidate.
- Iterate on:
  3. Select an informative viewpoint.
  4. Improve the selection in the local neighborhood.

Figure 8. Viewpoint selection algorithm

Initially, we generate a set of candidate viewpoints  $P_s$ . This is done by uniformly sampling a sphere that bounds the object [21]. We use a sphere whose radius is twice as large as that of the tight bounding sphere. In practice, we sample the sphere with 200 points.

Next, we evaluate the quality of each viewpoint  $p_i \in P_s$  according to the regions of interest it views:

$$\bar{I}(p_i) = \sum_{v_j \in V} \mathcal{I}(v_j) w_i(v_j), \quad (8)$$

where  $V$  is the set of vertices of mesh  $S$ ,  $\mathcal{I}(v_j)$  is defined in Equation (7), and  $w_i(v_j)$  is a weight defined as follows. The weight should be high if the area the region occupies in the surface’s projection is large. Let  $\beta_{ij}$  be the angle between the surface normal at vertex  $v_j$  and the viewing direction  $\overrightarrow{p_i - v_j}$ . If  $v_j$  is visible from  $p_i$ , then  $w_i(v_j) = \sqrt{\cos \beta_{ij}}$ , otherwise  $w_i(v_j) = 0$ .

In Stage 3, a few descriptive viewpoints are selected. Since it is expected that points close to each other will view similar regions, it is insufficient to simply choose the best ones. Instead, we select viewpoints that collectively describe different regions of interest of the mesh.

Specifically, the first viewpoint selected is the one having the maximal  $\bar{I}(p_i)$ . Then, we iteratively add a new viewpoint, which jointly with the previously-selected viewpoints, maximize the viewed regions of interest. Let  $w_{max}(v_j)$  be the highest weight assigned to  $v_j$  by one of the viewpoints selected so far. And, let  $\delta(p_i)$  be the added degree of interest contributed by  $p_i$ , defined as:

$$\delta(p_i) = \sum_{v_j \in V} \mathcal{I}(v_j) \max(w_i(v_j) - w_{max}(v_j), 0). \quad (9)$$

We add to the set of the selected viewpoints the candidate viewpoint that maximizes  $\delta$ .

The number of viewpoints is dynamic and depends on the object’s geometry. We keep adding viewpoints to the



Figure 9. The most informative viewpoints. For each model, the selected viewpoints are shown in descending order of informativeness, from left to right.

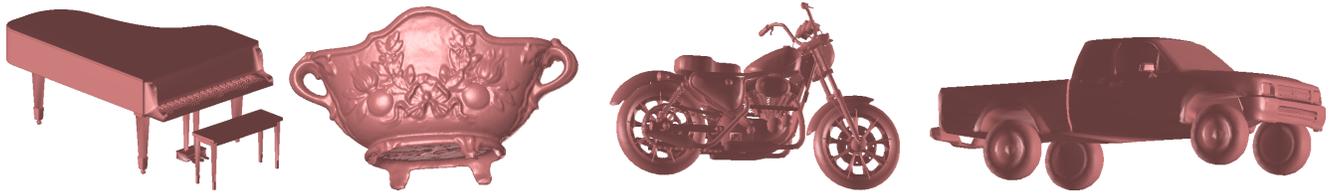


Figure 10. The most informative viewpoint

set until one of the following conditions is satisfied. First, the computed interest of the viewed vertices is at least 60% of the total computed interest over the whole mesh:  $0.6 \cdot \sum_v \mathcal{I}(v) < \bar{I}(p_0) + \sum_{p_i \in \text{set}} \delta(p_i)$ . Second, no new viewpoint  $p$  adds large-enough viewed region of interest:  $\delta(p) \leq 0.1 \cdot \sum_{v_j \in V} \mathcal{I}(v_j)$ .

When a new viewpoint is selected, Stage 4 attempts to refine its location by searching its neighborhood for a better viewpoint. The neighborhood’s size is the initial distance between the sampled candidate viewpoints. This neighborhood is uniformly sampled, every three degrees. The point that maximizes the weighted viewed region of interest (Equations (8),(9)) is chosen.

In order to avoid generating two symmetric views, as a pre-processing step we apply an algorithm that detects reflective symmetries [13]. The regions of interest on one side of the symmetric plane are zeroed.

## 5. Viewpoint selection: results

We ran our algorithm on the 79 meshes from Section 3. For 42 models our algorithm generated a single viewpoint, for 32 models two viewpoints, and for 5 models three viewpoints.

Figure 9 shows all the viewpoints generated for some models, each drawn in descending order, from the most informative view to the least informative view. For instance, our algorithm generated three viewpoints for the Buddha, which are indeed appropriate for this case, since every view reveals additional interesting details. Figure 10 shows only the most informative view for additional models. For the pi-



Figure 11. Comparison of our results (top) with some of the results of [16] (bottom)

ano and the pickup truck, our algorithm resulted in a single view. All the results are given in the supplemental material.

**Comparison to other methods:** The results of our algorithm competes favorably with those of the state-of-the-art methods. Figure 11 compares our results with [16]’s, where saliency is based on class distinctiveness. In [16], the number of viewpoints is static and set to 4. We compare only the first viewpoint. As can be seen, our selected viewpoints are often more “natural”.

Figure 12 compares our most informative viewpoint with those of [17, 23, 28]. Our selected viewpoint of David is the best according to our user study (described next). Moreover, it is close to the classic three-quarter frontal view (i.e., the oblique view between frontal view and profile view), which is considered good in human vision.

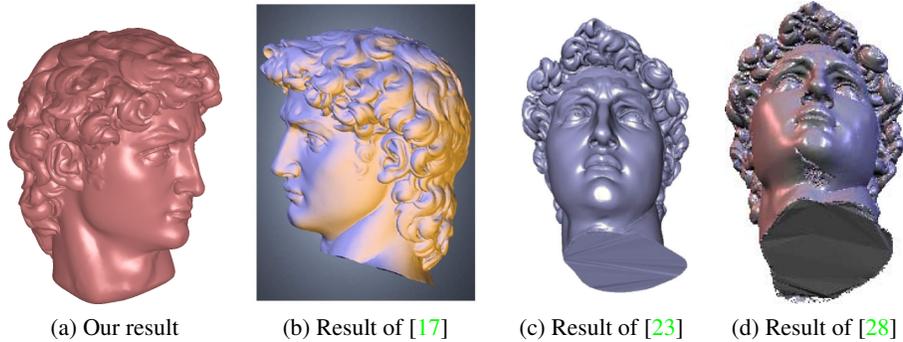


Figure 12. Comparison of the best views of David. Our view is close to the classic three-quarter frontal view.

**Evaluation using a user study:** Since there is no available ground truth for the best viewpoints, we conducted a user study. The goal was to learn which views are considered the most informative.

For each of our 79 models, we produced 12 images, each taken from a different viewpoint. The viewpoints were created by uniformly sampling the bounding sphere. We decided to use 12 images as a compromise between the accuracy of the survey (requiring a large number of viewpoints) and our wish to avoid overloading the evaluators (requiring a small number of viewpoints).

We asked the evaluators to mark the most informative views of the object—those that let the observer understand its shape. Each screen included 12 randomly-ordered viewpoints of a single object. The number of informative views that could be marked was unlimited.

We collected results for over two months, from 195 evaluators, 57% men and 43% women, at the age 15-65. We obtained 68 evaluations per model on average.

Figure 13(a) shows a typical distribution of the evaluation. In this example, there are three views considered informative by the evaluators. Therefore, rather than defining our ground truth to be a single view, we define it as a set of views, consisting of the highest-ranked views before the largest decrease in the histogram. Figure 13(c) shows the ground truth—a set of three informative views. Our best view is given in Figure 13(b).

To assess the results of our algorithm, we compared the view(s) selected by our algorithm to the ground truth. Since the evaluators could choose between 12 viewpoints only, our result is considered correct if it is closer (angularly) to a view of the ground truth than to any other view.

For 75 out of 79 models (94.9%), the most informative view selected by our algorithm matched the ground truth. If we take into account also the other informative views generated by our algorithm, the views of 78 out of 79 models (98.7%) matched the ground truth.

**Limitations:** The failure case of our algorithm is the tank in Figure 14, for which our algorithm generated a single

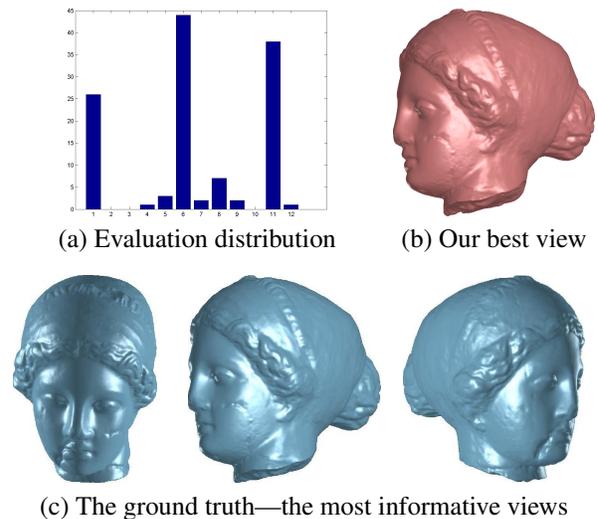


Figure 13. Informative views. Three views are considered informative by the evaluators (a). Our computed most-informative view (b) is one of those that belong to the ground-truth (c).

view. This failure can be explained by the lack of high level factors. People tend to prefer “natural” positioning of objects. Therefore, our evaluators preferred a side view. However, most of the details of the tank (turret, cannon) are on the top and therefore our algorithm computed a top view.

Similarly, for the lamp in Figure 14, our algorithm chose the view of the ground truth as the second-most informative view. As the most informative view our algorithm chose a back view, which contains many distinct details, such as wires and screws. In contrast, the evaluators preferred the front view, where the light bulb is visible.

## 6. Conclusion

This paper has studied the detection of surface regions of interest. We discuss two considerations—vertex distinctness and patch association—and methods that realize them.

The regions of interest can benefit many computer vision and geometry processing applications. We explore one such application—the selection of the most informative view-

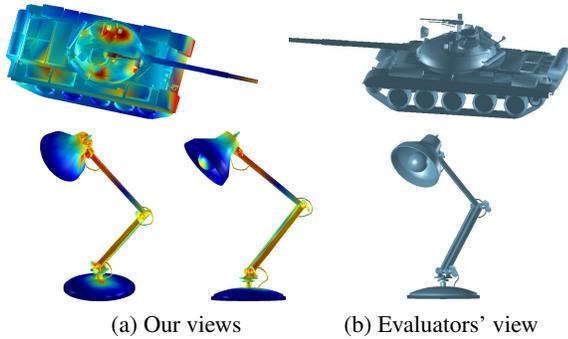


Figure 14. Limitations. High-level factors are not considered by our algorithm. Therefore, it may choose views with many details, rather than more “natural” views.

points of objects. We show state-of-the-art results, which are reinforced by a user study.

In the future, high-level factors can be added to our algorithm. A notable example is the class distinctness of [24], which detects regions that distinguish a shape from shapes in other classes.

**Acknowledgements:** The models were generously provided by the AIM@SHAPE Shape Repository, The Technion’s CG&M Lab, Stanford 3D Scanning Repository and the Princeton Benchmark. This research was supported in part by the Ollendorff foundation. We thank Naama Cohen, Sarel Ben Arye, and Eli Levi for their work on related issues, which were not included in this paper. We thank Michael Kazhdan for his symmetry code.

## References

- [1] A. Bobenko and P. Schröder. Discrete willmore flow. In *SGP*, pages 101–110, 2005. 2
- [2] B. Bustos, D. Keim, D. Saupe, T. Schreck, and D. Vranić. Feature-based similarity search in 3D object databases. *ACM Computing Surveys*, 37(4):345–387, 2005. 2
- [3] M. Cox and T. Cox. “*Multidimensional Scaling*”. Chapman and Hall, 1994. 3
- [4] R. Gal and D. Cohen-Or. Salient geometric features for partial shape matching and similarity. *ACM Transactions on Graphics*, 25(1):150, 2006. 1, 4, 5
- [5] M. Garland and P. Heckbert. Surface simplification using quadric error metrics. In *SIGGRAPH*, pages 209–216, 1997. 3
- [6] N. Gelfand, N. Mitra, L. Guibas, and H. Pottmann. Robust global registration. In *SGP*, page 197, 2005. 1
- [7] S. Goferman, L. Zelnik-Manor, and A. Tal. Context-Aware Saliency Detection. In *CVPR*, pages 2376–2383, 2010. 1, 3
- [8] P. Hall and M. Owen. Simple canonical views. In *The British Machine Vision Conf.*, volume 1, pages 7–16, 2005. 1
- [9] L. Itti, C. Koch, and E. Niebur. A Model of Saliency-Based Visual Attention for Rapid Scene Analysis. *PAMI*, pages 1254–1259, 1998. 1
- [10] A. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3D scenes. *PAMI*, 21(5):433–449, 1999. 2
- [11] T. Judd, K. Ehinger, F. Durand, and A. Torralba. Learning to predict where humans look. In *ICCV*, pages 2106–2113, 2010. 1
- [12] S. Katz, G. Leifman, and A. Tal. Mesh segmentation using feature point and core extraction. *The Visual Computer*, 21(8):649–658, 2005. 3
- [13] M. Kazhdan, B. Chazelle, D. Dobkin, A. Finkelstein, and T. Funkhouser. A reflective symmetry descriptor. In *ECCV*, pages 642–656, 2002. 6
- [14] Y. Kim, A. Varshney, D. Jacobs, and F. Guimbretière. Mesh saliency and human eye fixations. *ACM Transactions on Applied Perception*, 7(2):1–13, 2010. 2
- [15] J. Koenderink and A. van Doorn. Surface shape and curvature scales. *Image and Vision Comp.*, 10:557–565, 1992. 2
- [16] H. Laga. Semantics-driven approach for automatic selection of best views of 3D shapes. In *3DOR*, 2010. 1, 6
- [17] C. Lee, A. Varshney, and D. Jacobs. Mesh saliency. *ACM Transactions on Graphics*, 24(3):659–666, 2005. 1, 4, 6, 7
- [18] J. Lee, B. Moghaddam, H. Pfister, and R. Machiraju. Finding optimal views for 3D face shape modeling. In *Face and Gesture Recognition*, pages 31–36, 2004. 1
- [19] X. Li and I. Guskov. Multi-scale features for approximate alignment of point-based surfaces. In *SGP*, page 217, 2005. 1
- [20] H. Ling and K. Okada. Diffusion distance for histogram comparison. *CVPR*, 1:246–253, 2006. 2
- [21] J. Mitchell. Discrete uniform sampling of rotation groups using orthogonal images. *SIAM Journal of Scientific Computing*, 30:525–547, 2007. 5
- [22] F. Mokhtarian and S. Abbasi. Automatic selection of optimal views in multi-view object recognition. In *The British Machine Vision Conf.*, pages 272–281, 2000. 1
- [23] M. Mortara and M. Spagnuolo. Semantics-driven best view of 3D shapes. *Computers & Graphics*, 33(3):280–290, 2009. 1, 6, 7
- [24] P. Shilane and T. Funkhouser. Distinctive regions of 3D surfaces. *ACM Transactions on Graphics*, 26(2):7, 2007. 1, 4, 5, 8
- [25] J. Tangelder and R. Veltkamp. A survey of content based 3D shape retrieval methods. *Multimedia Tools and Applications*, 39(3):441–471, 2008. 2
- [26] O. van Kaick, H. Zhang, G. Hamarneh, and D. Cohen-Or. A survey on shape correspondence. In *Eurographics State-of-the-art Report*, 2010. 2
- [27] D. Walther and C. Koch. Modeling attention to salient proto-objects. *Neural Networks*, 19(9):1395–1407, 2006. 1
- [28] 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 *SMI*, page 40, 2006. 1, 6, 7
- [29] Y. Yang, T. Lu, and J. Lin. Saliency regions for 3D mesh abstraction. *Advances in Multimedia Information Processing*, pages 292–299, 2009. 1