On Visual Complexity of 3D Shapes

Waqar Saleem,a, Alexander Belyaevb, Danyi Wangc, Hans-Peter Seideld

aTheoretische Informatik II, Friedrich-Schiller-Universität Jena, Ernst-Abbe-Platz 2, 07743 Jena, Germany
bSchool of Engineering and Physical Sciences, Heriot-Watt University, Edinburgh EH14 4AS, United Kingdom
cPITERION GmbH, Hanns-Klemm-Str. 5, 71034 Böblingen, Germany
dMax-Planck-Institut Informatik, 66123 Saarbrücken, Germany

Abstract

We present an approach to compute the perceived complexity of a given 3D shape using the similarity between its views. Previous studies on 3D shape complexity relied on geometric and/or topological properties of the shape and are not appropriate for incorporating results from human shape perception which claim that humans perceive 3D shapes as organizations of 2D views. Therefore, we base our approach to computing 3D shape complexity on the (dis)similarity matrix of the shape’s 2D views. To illustrate the application of our approach, we note that simple shapes lead to similar views whereas complex ones result in different, dissimilar views. This reflected in the View Similarity Graph (VSG) of a shape as tight clusters of points if the shape is simple and increasingly dispersed points as it gets more complex. To get a visual intuition of the VSG, we project it to 2D using Multi-Dimensional Scaling (MDS) and introduce measures to compute shape complexity through point dispersion in the resulting MDS plot. Experiments show that results obtained using our measures alleviate some of the drawbacks present in previous approaches.

Keywords: 3D shape complexity, shape perception, view similarity, MDS

1. Introduction

Which is the more complex shape among two given shapes? How about three shapes, or a whole collection of shapes? What are their relative complexities? These questions might be easily answered by a human but are hard to answer computationally and have received relatively little attention in digital geometry research. However, they take on importance in view of the increasing occurrence of large shape collections available both publicly online and as in house catalogs of parts. “Complexity” in this paper refers to a shape’s visual or perceived, not mathematical, complexity.

Results from psychology on human perception of shape [6, 9] claim that 3D objects are perceived as arrangements of 2D views. This observation has been exploited for object recognition [8], to compute shape similarities [7] and to extract static [25] and dynamic [19] representations of 3D shapes.

In this paper, we study the application of these results to computationally solve the problems we posed earlier. We propose that a shape’s perceived complexity derives from the dissimilarity among its views. This is trivially exhibited by the sphere, the canonical simplest shape [15, 17, 21], which has identical views, up to scale, from all view points. We therefore claim that a shape is complex if its views are dissimilar from each other and simple if its views are similar. The higher the dissimilarity, the more complex the shape.

We thus propose to measure shape complexity using view similarity. Given a shape, we obtain N views of it uniformly from its view sphere and perform pairwise view comparisons leading to a N × N similarity matrix, S. The corresponding View Similarity Graph (VSG) is a graph containing a node for each view and edge weights equal to the (dis)similarity between the views at endpoints. Here, we consider edge weights to represent Euclidean distance. Thus, similar views lead to tight clusters of points in the VSG and dissimilar views result in dispersed points. To allow visual inspection, we project the VSG to a 2D MDS plot using a distance preserving projection and propose two measures of point dispersion in the MDS plot as estimates of its complexity. Figure 1 provides a pictorial overview of our method.

We tested our shape complexity measures on a small
collection of shapes collected from an online shape repository by ranking it according to our measures along with some existing methods. While one of our proposed measures leads to some irregularities in the ranking, the other gives a better ranking than existing methods.

Our contribution in this paper is to introduce a view similarity based approach to computation of perceived shape complexity. More specifically, we do so using the shape’s VSG which can be thought of as a generalized aspect graph. The approach is illustrated using two simple measures of point dispersion in a 2D projection of the VSG. The unique insight into the relation between a shape’s views that its VSG provides can be used to compute other shape properties as well.

Section 2 reviews related work and Sections 3 and 4 respectively describe how to obtain $S$ and an MDS plot from a shape. Section 5 outlines our two measures for computing complexity from an MDS plot. Rankings of a sample collection using the measures are presented in Section 6. Section 7 closes with a discussion on our measures, the VSG and its applications, and improvements of our approach.

2. Previous Work

A comprehensive study of various geometric and topological 3D shape complexity measures by J. Rossignac in [18] describes five measures that could constitute the notion of shape complexity. Two of these measures are qualitative and represent the redundancy and topology of the shape. The remaining three quantitative measures represent the smoothness of the shape and complexity of the continuous or discrete representation used for the shape. As the study was motivated by shape compression applications, the suggested measures do not carry over simply to reflect visual shape complexity. While each measure captures a distinct aspect of shape complexity, it is unclear how a single quantitative combination can be obtained. Secondly, some of the quantitative measures are too general to the extent that largely varying shapes can have the same value for them. One of the measures is sensitive to shape tessellation and some others are applicable only to specialized shape representations.

More direct approaches make use of information theory [20]. The canonical simplest shape in the literature is the sphere. There is no variation in its surface and its appearance is constant, up to scale, from all directions. Page et al. [15, 21] attribute this to the constant curvature of the sphere throughout its surface and compute the shape’s complexity as the entropy of its curvature distribution. Rigau et al. [17] observe that every point on the sphere’s surface is visible from every other surface point. They define a shape’s inner and outer complexities in terms of the mutual information (based on visibility) between shape regions and, respectively, other shape regions and regions of a bounding sphere. These approaches are sensitive to noise; small amounts of noise on the surface of the shape can cause variations in the curvature distribution or mutual visibilities leading to significant perturbation in the shape’s complexity rating.

3. Obtaining the similarity matrix

We approximate a shape’s view sphere with a Loop subdivided [11] icosahedron that is centered at the shape’s circumcenter and whose vertices are projected to lie on a sphere, e.g. Figure 1(a). This leads to 42 views of the shape, e.g. in Figure 1(b), obtained from view sphere vertices. These provide a sufficiently dense sampling of the view sphere for the similarity matrix, $S$, to be invariant to shape orientation. As the view sphere scales and translates with the shape, $S$ is also invariant to shape scaling and translation. Following earlier methods [14, 25], we compute view similarity in terms of similarity between shape boundary contours, e.g. Figure 1(c), which are obtained as the Crust [1] of shape boundary pixels in the shape views. Similarity between boundary contours is computed using the
Contour to Centroid Triangulation (CCT) algorithm [2] which has been reported to outperform other shape similarity measures [24]. The similarity matrix, $S$, of the $N = 42$ shape views, $\{v_1, v_2, \ldots, v_N\}$, is the $N \times N$ matrix whose $(i, j)$-th entry, $s_{ij}$, is the similarity between $v_i$ and $v_j$.

4. Applying MDS to the Similarity Matrix

A shape’s MDS plot contains $N$ coplanar points whose pair-wise distances are determined by the entries in $S$. The starting positions, $P_0$, of the points are modified iteratively. In each iteration $m, m \geq 0$, the point configuration, $P_m$, is tested for an MDS solution of $S$.

If a solution or a close approximation has been reached, iteration terminates. Otherwise, point positions are updated to $P_{m+1}$ and iteration continues. An example for $N = 4$ is illustrated in Figures 2(b-d).

4.1. Starting Positions

In order to obtain a distance preserving projection, $P_0 = \{p_{i0} | i \in \{1, \ldots, N\}\}$ may be assigned arbitrarily [4, 5]. For meaningful comparison of MDS plots obtained from different shapes, it is important that the plots derive from the same $P_0$. The exact choice of this $P_0$ is unimportant and we set it to uniform samples on a sinusoidal function, i.e. the $x$ and $y$ coordinates of $p_{i0}$ are given by

$$x_{i0} = \frac{i - \frac{1}{N}}{N - 1}, \quad y_{i0} = \frac{1}{2}(1 - \sin 2\pi x_{i0}).$$

Examples for $N = 42$ and $N = 4$ are shown in Figures 2(a) and (b) respectively.

4.2. Checking for an MDS Solution

Let $D_m$ be the distance matrix of $P_m$, i.e. the $(i, j)$-th entry of $D_m$ is the Euclidean distance between points $p_{im}$ and $p_{jm}$ in $P_m$. Then, $D_m$ is an MDS solution of $S$ if the ranking number matrices of $D_m$ and $S$ are the same,

$$R(D_m) = R(S).$$  (1)

For a matrix, $A$, $R(A)$ contains entries of $A$ replaced by their sorted ranks. The largest entry has a rank of one, the second largest a rank of two, and so on. Equal entries have consecutive ranks.

4.3. Updating Point Positions

The rank image matrix of a matrix, $A$, with respect to another, $B$, denoted as $R_B(A)$, satisfies the property

$$R(R_B(A)) = R(B).$$

$R_B(A)$ is obtained by permuting the entries of $A$ to satisfy the above property. Note that $R(R_B(D_m)) = R(S)$ for any $m$ and so $R_B(D_m)$ is always an MDS solution of $S$ as per Equation 1. Therefore, if a certain $D_m$ is not an MDS solution of $S$, positions of points in $P_m$ are updated to $P_{m+1}$ such that $D_{m+1}$ may be the same as $R_B(D_m)$. To achieve this, a correction factor, $f_{ij,m}$, is computed for each point pair $(p_{jm}, p_{im})$ in $P_m$ as

$$f_{ij,m} = \frac{d_{ij,m} - d_{ij,m}'}{2d_{ij,m}}.$$

where $d_{ij,m}$ and $d_{ij,m}'$ are entries in $D_m$ and $R_B(D_m)$ respectively. The correction factor for a point pair can be thought of as the force the points exert on each other. A positive value of $f_{ij,m}$ indicates that the current distance between the point pair is an underestimate and should be increased, whereas a negative value implies a shortening of the distance. The displacement of $p_{im}$ with respect to $p_{jm}$ is then given as

$$\vec{d}_{ij,m} = f_{ij,m} \cdot (p_{jm} - p_{im}).$$

Averaging over displacements for $p_{im}$ with respect to all other points gives its total displacement,

$$\vec{d}_{im} = \frac{1}{N - 1} \sum_{j=1 \neq i}^{N} \vec{d}_{ij,m}.$$

Finally, the new point position is given by

$$p_{im+1} = p_{im} + \vec{d}_{im}.$$
4.4. Stopping Condition

The total displacement for a point aggregates forces exerted on it by all other points, causing its new position to partly satisfy each of the component forces while fully satisfying none. Thus, point displacement does not lead to an exact MDS solution. However, progressive updates bring the points’ distance matrix closer to the solution. This is reflected by progressively smaller point displacements as iteration continues, e.g. in Figure 2(c). When the magnitudes of all point displacements in an iteration fall below a threshold, iteration is terminated. An excessively low threshold leads to wasteful computation of iterations that produce minute alterations in point positions which have no impact on the eventual shape complexity value.

5. Estimating Shape Complexity

The MDS method outlined above considers ranks and not magnitudes of matrix entries. As all shapes start with the same $P_0$, shown in Figure 2(a), shapes whose similarity matrices differ only in scale lead to the same MDS plot. We mitigate this by rescaling the point configuration, $P_M$, from the final MDS iteration, according to the shape’s similarity matrix, $S$, to obtain $Q = \{q_i\}$, the final set of points. A rescaling factor is computed,

$$F = \frac{\text{largest entry in } S}{\text{largest entry in } D_M},$$

and points are rescaled about their centroid to obtain $Q$.

$$c_M = \frac{1}{N} \sum_{i=1}^{N} p_{iM},$$

$$q_i = c_M + F \cdot (p_{iM} - c_M).$$

We propose two shape complexity measures based on dispersion of points in $Q$. The first, $C_{\sigma}$, relies on one dimensional information,

$$C_{\sigma} = \sqrt{\sigma^2_x + \sigma^2_y},$$

where $\sigma_x$ and $\sigma_y$ are standard deviations in the $x$ and $y$ coordinates of $q_i$. The second, $C_H$, is based on two dimensional information and is defined in terms of the convex hull, $H$, of $Q$. More specifically,

$$C_H = \text{area of } H.$$

6. Results

Figure 3 shows a few sample shapes alongside their corresponding MDS plots and values for $C_H$ and $C_{\sigma}$. For a uniform visualization across shapes, points in the MDS plots shown have been rescaled to fit the unit square while preserving aspect ratio. Actual computation of $C_H$ and $C_{\sigma}$ does not involve this rescaling. Table 1 shows results for our full test set, along with relative complexities according to each measure, e.g. according
to \( C_\sigma \), the Camel is 2.6 times as complex as the Bumpy sphere. The result of sorting our test set according to relative complexities using \( C_H \), \( C_\sigma \) and existing shape complexity methods [15, 21] is shown in Figure 4.

7. Discussion and Closing Remarks

\( C_H \) and \( C_\sigma \) lead to different rankings. \( C_\sigma \) rates relatively simple shapes like the Cone and Torus as highly complex. This is because their MDS plots (Figure 3) contain tight clusters that are separated along one of the axes. As \( C_\sigma \) relies on deviation in one dimension only (along the \( x \) and \( y \) axes separately), the computed value comes out to be large. This is corrected when we incorporate two dimensional information in \( C_H \). The \( C_\sigma \) measure can be improved by measuring deviation along the principal axes of the points instead of the coordinate axes.

As expected, the curvature based methods of [15, 21] are unable to deal with noise; they rank the Bumpy sphere as one of the most complex shapes, whereas our view based measures ignore the noise and rank the Bumpy sphere as the least complex.

Both our measures rank the Bunny less complex than simpler shapes like the Torus. We believe this is because of inadequate representation by our measures of the information contained in shapes’ MDS plots. The use of more sophisticated dispersion measures [12] could better represent the contained information. Another remedy would be to analyze dispersion directly in the view similarity graph (VSG) instead of its 2D projection. The quality of our measures could ultimately be tested by comparing complexity rankings of a sample set of shapes by users, e.g. in a user study, and by our measures. An added benefit of such a study would be to establish a benchmark database for future shape
complexity measures. It is also worth noting that our 3D shape complexity measures are not isometry-invariant. Indeed we should not require perceptual complexity measures to be invariant with respect to isometries and other geometric transforms: we perceive a straight and a knotted cylinder to have different visual complexities.

The VSG can be viewed as a generalized aspect graph. While aspect graphs are applicable to a limited set of shapes, a VSG can be obtained for any shape. Also, aspect graphs are defined in terms of distinct visual events, whereas VSG’s pose no such condition and neighboring views in a VSG may be very similar. The aspect graph of an applicable shape is unique. However, the VSG of a shape may vary depending on the used approximation of the view sphere.

The VSG of a shape can be used to compute other properties as well. A shape’s symmetries [13, 16] are a measure of its self-similarities. The VSG of a shape with high symmetry would therefore contain tight clusters, e.g. the Star in Figure 3. Each point in a cluster contributes a “vote” for a view. Different parts of a shape voting for the same view is indicative of symmetry in the shape. Similar voting schemes have been employed in previous works on symmetry [13, 16]. A large number of votes for a view could also be used as a cue for the shape’s “best view” [3, 10, 14, 22, 23, 25].

7.1. Conclusion

Our main contribution in this paper is our general approach to shape complexity using view similarity. The two measures introduced and tested in the paper are far from perfect. View similarity based measures in general can be improved by analyzing the shape’s VSG directly and by combining several geometric measures on the VSG where the combination weights are learned from appropriate examples. A deeper analysis of the VSG can also yield information pertinent to other shape properties. The VSG itself can be enriched by using a more elaborate view similarity method than the silhouette based one used in this paper.

References
