Efficient View-Based 3D Reflection Symmetry Detection

Bo Li* Texas State University
Henry Johan† Fraunhofer IDM@NTU
Yuxiang Ye‡ Texas State University
Yijuan Lu§ Texas State University

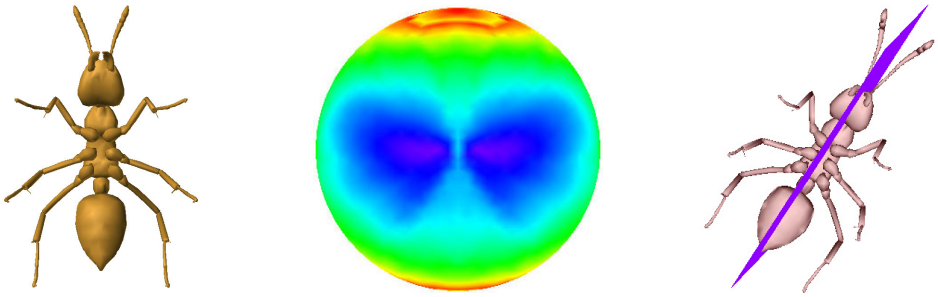


Figure 1: *An overview of our view-based symmetry detection algorithm: an example of an ant model, its viewpoint entropy distribution, and the detected symmetry plane by matching the viewpoints.*

Abstract

Symmetries exist in many 3D models while efficiently finding their symmetry planes is important and useful for many related applications. This paper presents a simple and efficient view-based reflection symmetry detection method based on the viewpoint entropy features of a set of sample views of a 3D model. Before symmetry detection, we align the 3D model based on the Continuous Principal Component Analysis (CPCA) method. To avoid the high computational load resulting from a directly combinatorial matching among the sample views, we develop a fast symmetry plane detection method by first generating a candidate symmetry plane based on a matching pair of sample views and then verifying whether the number of remaining matching pairs is within a minimum number. Experiments demonstrate better accuracy, efficiency, and flexibility of our algorithm than state-of-the-art approaches.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—[Curve, surface, solid and object representations]

Keywords: symmetry detection, viewpoint entropy, matching

1 Introduction

Symmetry is an important clue for geometry perception: it is not only in many man-made models, but also widely exists in the nature [Liu et al. 2010]. Symmetry has been used in many applications such as: 3D alignment [Chaouch and Verroust-Blondet 2009], shape matching [Kazhdan et al. 2004], remeshing [Podolak et al. 2007], 3D model segmentation [Simari et al. 2009] and retrieval [Sfikas et al. 2011].

However, existing symmetry detection algorithms still have much room for improvement in terms of both simplicity and efficiency in detecting symmetry planes, as well as the degree of freedom to find approximate symmetry planes for a roughly symmetric 3D model. In addition, most of the existing symmetry detection methods are geometry-based, thus their computational efficiency will be tremendously influenced by the number of vertices of a model. Motivated by the symmetric patterns existing in the viewpoint entropy [Vázquez et al. 2001] distribution of a symmetric model, we propose a novel and efficient view-based symmetry detection algorithm (see Fig. 1) which finds symmetry plane(s) by matching the viewpoint entropy features of a set of sample views of a 3D model aligned beforehand using Continuous Principal Component Analysis (CPCA) [Vranic 2004]. Based on experimental results, we find that our symmetry detection algorithm is more accurate (in terms of both the positions of detected symmetry planes and sensitivity to minor symmetry differences), efficient, robust (e.g. to the number of vertices and parameter settings such as view sampling), and versatile in finding symmetry planes of diverse models.

In the rest of the paper, we first review the related work in Section 2. In Section 3, we present the viewpoint entropy distribution-based symmetry detection algorithm. Section 4 describes diverse experimental evaluation and comparison results of the detection algorithm. Section 5 concludes the paper and lists several future research directions.

2 Related Work

Symmetry Types Though there are different types of symmetry, reflection symmetry is the most important and commonly studied. Chaouch and Verroust-Blondet [Chaouch and Verroust-Blondet 2009] introduced four types of reflection symmetries, which are cyclic (several mirror planes passing through a fixed axis), dihedral (several mirror planes passing through a fixed axis with one perpendicular to the axis), rotational symmetry (looks similar after rotation, e.g., different platonic solids, like tetrahedron, octahedron, icosahedron and dodecahedron) and unique symmetry (only one mirror plane, for instance, many natural and most man-made objects). Most symmetric objects are mirror rather than rotational

*e-mail: b_l58@txstate.edu
†e-mail: hjohan@fraunhofer.sg
‡e-mail: y_y13@txstate.edu
§e-mail: lu@txstate.edu

SA'14 Workshop on Creative Shape Modeling and Design, December 03, 2014, Shenzhen, China.

http://dx.doi.org/10.1145/2669043.2669045

symmetric [Sawada and Pizlo 2008].

Symmetry Detection Symmetry detection is to search the (partial or full) symmetry planes of a 3D object. The latest review on symmetry detection is available in [Mitra et al. 2012]. We classify current symmetry detection techniques into the following four groups according to the features employed.

Symmetry detection based on pairing point features. This type of approach first samples points on the surface of a 3D model and then extracts their features. After that, it finds point pairs by matching the points. Based on the point pairs, symmetry evidences are accumulated to decide the symmetry plane. Two typical algorithms are [Mitra et al. 2006] and [Cailliere et al. 2008]. To decide the symmetry plane, Mitra et al. [Mitra et al. 2006] adopted a stochastic clustering and region-growing approach, while Calliere et al. [Cailliere et al. 2008] followed the same framework of pairing and clustering, but utilized 3D Hough transform to extract significant symmetries. In fact, the initial idea of this approach can be traced back to the symmetry distance defined in [Zabrodsky et al. 1995]. Podolak et al. [Podolak et al. 2006] proposed a planar-reflective symmetry transform and based on the transform they defined two 3D features named center of symmetry and principal symmetry axes, which are useful for related applications such as 3D model alignment, segmentation, and viewpoint selection.

Symmetry detection based on pairing line features. Bokeloh et al. [Bokeloh et al. 2009] targeted on the so-called rigid symmetries by matching feature lines. Rigid symmetries are the reoccurring components with differences only in rigid transformations (translation, rotation and mirror). They first extracted feature lines of a 3D model, then performed feature line matching, and finally validated the symmetry based on the feature correspondence information by adopting a region growing approach, as well.

Symmetry detection based on 2D image features. Sawada and Pizlo [Sawada and Pizlo 2008] [Sawada 2010] performed symmetry detection based on a single 2D image of a volumetric shape. First, a polyhedron is recovered from the single 2D image based on a set of constraints including 3D shape symmetry, minimum surface area, maximum 3D compactness and maximum planarity of contours. Then, they directly compared the two halves of the polyhedron to decide its symmetry degree. From a psychological perspective, Zou and Lee [Zou and Lee 2006] [Zou and Lee 2005] proposed one method to detect the skewed rotational and mirror symmetry respectively from a CAD line drawing based on a topological analysis of the edge connections.

Other symmetry detection approaches. Martinet et al. [Martinet et al. 2006] proposed a 3D feature named generalized moments for symmetry detection. Rather than directly computing original moments features, they mapped them into another feature space by spherical harmonics transform and then searched for the global symmetry in the new feature space. Xu et al. [Xu et al. 2009] developed an algorithm to detect partial intrinsic reflectional symmetry based on an intrinsic reflectional symmetry axis transform. After that, a multi-scale partial intrinsic symmetry detection algorithm was proposed in [Xu et al. 2012]. There are also techniques to detect some other specific symmetries, such as curved symmetry [Liu and Liu 2010] and symmetries of non-rigid models [Raviv et al. 2010], as well as symmetry hierarchy of a man-made 3D model [Wang et al. 2011]. Kim et al. [Kim et al. 2010] detected global intrinsic symmetries of a 3D model based on Möbius Transformations [Lipman and Funkhouser 2009], a stereographic projection approach in geometry. Recently, Wang et al. [Wang et al. 2014] proposed Spectral Global Intrinsic Symmetry Invariant Functions (GISIFs), which are robust to local topological changes compared to the GISIFs obtained from geodesic distances. Their generality and flexibility outperform the two classical GISIFs: Heat Kernel Signature (HKS) [Sun et al. 2009] and Wave Kernel Signature (WKS) [Aubry et al. 2011].

All above and existing symmetry detection techniques can be categorized into a geometry-based approach. However, distinctively different from them, we adopt a view-based approach to accumulate the geometrical information of many vertices together into a view in order to more efficiently detect the reflection symmetry of a 3D model, which also serves as the novelty and main contribution of our method.

Symmetry Applications As an important shape feature, symmetry is useful for many related applications. For example, they include symmetry plane detection for 3D MRI image [Tuzikov et al. 2003], shape matching [Kazhdan et al. 2004] [Podolak et al. 2006], 3D model alignment [Tedjokusumo and Leow 2007] [Sfikas et al. 2011], shape processing and analysis [Golovinskiy et al. 2009] including remeshing [Podolak et al. 2007], symmetrization [Mitra et al. 2006], viewpoint selection [Podolak et al. 2006], and subspace shape analysis [Berner et al. 2011], 3D segmentation [Podolak et al. 2006] [Simari et al. 2009] [Wang et al. 2014], and curve skeleton extraction [Tagliasacchi et al. 2009] [Cao et al. 2010].

3 Symmetry Detection Algorithm

3.1 3D Model Normalization Based on CPCA

Properly normalizing a 3D model before symmetry detection can help us to minimize the searching space for symmetry planes. The process of 3D normalization includes three steps: 3D alignment (orientation normalization), translation (position normalization), and scaling (size normalization). 3D model alignment is to transform a model into a canonical coordinate frame, where the representation of the model is independent of its scale, orientation, and position. Two commonly used 3D model alignment methods are Principal Component Analysis (PCA) [Jolliffe 2002] and its descendant Continuous Principal Component Analysis (CPCA) [Vranic 2004] which considers the area of each face. They utilize the statistical information of vertex coordinates and extract three orthogonal components with largest extent to depict the principal axes of a 3D model. CPCA is generally regarded as a more stable PCA-based method. In addition, Johan et al. [Johan et al. 2011] proposed a 3D alignment algorithm based on Minimum Projection Area (MPA) which aligns a 3D model by successively selecting two perpendicular axes with minimum projection areas while the third axis is the cross product of the first two axes. However, compared with the PCA-based approaches, MPA takes a longer time to align 3D models while for this research we want to detect symmetry fast.

After a comparison of the influences of different 3D model alignment algorithms on the efficiency, accuracy and robustness of our view-based symmetry detection approach, we choose CPCA to align a model before performing symmetry detection. After the alignment with CPCA, we translate the model such that the center of its bounding sphere locates at the origin and scale the model such that its bounding sphere has a radius of 1. After this normalization, the symmetry plane(s) will pass the origin, which helps us to significantly reduce the searching space.

3.2 View Sampling and Viewpoint Entropy Distribution Generation

Vázquez et al. [Vázquez et al. 2001] proposed an information theory-related measurement named viewpoint entropy to depict the amount of information a view contains. It is formulated based on

the Shannon entropy and incorporates both the projection area of each visible face and the number of visible faces into the definition. However, the original definition was developed based on perspective projection, thus we use its extended version defined in [Takahashi et al. 2005] for orthogonal projection.

For each model, we sample a set of viewpoints based on the Loop subdivision [Loop 1987] on a regular icosahedron, denoted as L_0. We subdivide L_0 n times and denote the resulting mesh as L_n. Then, we set the cameras on its vertices, make them look at the origin (also the center of the model) and apply orthogonal projection for rendering. For a 3D model, to differentiate its different faces, we assign different color to each face during rendering. One example is shown in Fig. 2.

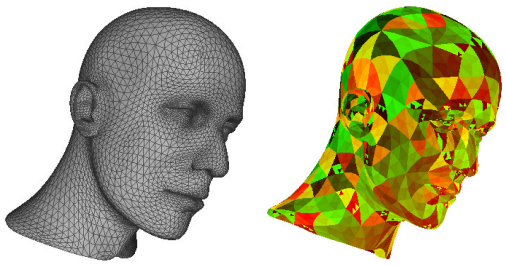


Figure 2: *Face color coding example.*

The viewpoint entropy [Takahashi et al. 2005] of a view with m visible faces is defined as follows.

$$E = -\frac{1}{log_2(m+1)} \sum_{j=0}^{m} \frac{A_j}{S} log_2 \frac{A_j}{S} \quad (1)$$

where, A_j is the visible projection area of the j^{th} (j=1, 2, $\cdots$, m) face of a 3D model and A_0 is the background area. S is the total area of the window where the model is rendered: S=A_0+$\sum_{j=1}^{m} A_j$. Projection area is computed by counting the total number of pixels inside a projected face.

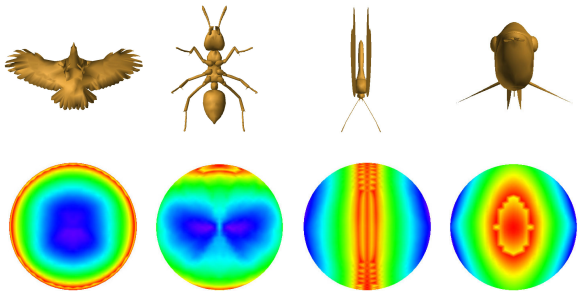


Figure 3: *Viewpoint entropy distribution examples: 1^{st} row shows the models after alignment with CPCA; 2^{nd} row demonstrates their respective viewpoint entropy distribution. Blue: large entropy; green: mid-size entropy; red: small entropy.*

Figure 3 shows the viewpoint entropy distributions of several models by using L_4 (2,562 sample viewpoints) for view sampling and mapping their entropy values as colors on the surface of the spheres based on the HSV color model. We can see there is a perfect correspondence between the symmetry of a model and that of its viewpoint entropy distribution sphere: their symmetry planes are the same. Therefore, the symmetry of a 3D model can be decided by finding the symmetry in the entropy distribution, thus avoiding the high computational cost of direct matching among its geometrical properties. What's more, since viewpoint entropy is computed based on the projection of each face, it is highly sensitive to small differences in the model. In addition, each viewpoint simultaneously captures the properties of many vertices and faces of a model as a whole, which also helps to significantly reduce the computational cost. We also find that it is already accurate enough based on a coarse view sampling, such as using L_1, as demonstrated in Section 4.2. Motivated by these findings, we propose to detect the symmetry of a 3D model based on its viewpoint entropy distribution.

3.3 Symmetry Detection Based on Iterative Feature Pairing

Even only using L_1 (42 viewpoints) for view sampling, if based on a naive matching approach by first directly selecting half of the total viewpoints and then matching them with the remaining half, it will result in $P(42, 21)$=2.75×10^{31} combinations. Thus, we develop a much more efficient symmetry detection method based on the following idea: iteratively select a matching pair of viewpoints to generate a symmetry plane and then verify all the rest matching pairs to see whether they are symmetric as well w.r.t the symmetry plane or at least in the symmetry plane. The method is listed in Algorithm 1.

We need to mention the followings for the algorithm. The views corresponding to the viewpoints that are located on the symmetry plane do not need to match each other. While, according to the Loop rule [Loop 1987], at most 2^{n+2} vertices of L_n are coplanar in a plane w.r.t a great circle. That is to say, at most 2^{n+2} viewpoints could be in the real symmetry plane. An ideal algorithm is to perfectly match w.r.t the symmetry plane all the viewpoint pairs that are not in the symmetry plane. However, we have found that usually there are numerical accuracy problems related to 3D model rendering (e.g. aliasing), viewpoint entropy computation (usually the entropy values of two symmetric viewpoints are not completely the same), as well as possible (either big or minor) differences in mesh triangulation. Therefore, we propose to partially solve this issue by relaxing some of the conditions though it sometimes causes certain false positive detections: if the total number ($matches$) of matched viewpoints w.r.t a candidate symmetry plane is at least $N - 2^{n+2}$, then it is confirmed as a symmetry plane. δ is a threshold which can control the strictness of symmetry definition. For example, using a small threshold we detect more strictly defined symmetries while using a bigger threshold, we allow some minor differences and detect rough symmetry properties. T_1 and T_2 are the normals of the planes w.r.t two correspondence points (P_u and P_v; P_i and P_j). The condition $\|CT\| > \epsilon$ AND $|DT| \neq 0$ means T_1 and T_2 is neither parallel nor perpendicular to each other. In another word, the line between P_i and P_j is not perpendicular to the candidate symmetry plane since T_1 and T_2 are not parallel (otherwise, $\|CT\| = 0$); and P_i and P_j are also not in the symmetry plane (otherwise, $|DT| = 0$). P_m is the midpoint of the line segment connecting points P_i and P_j. It is used to further assert the vertical symmetry property of P_i and P_j about the candidate symmetry plane by finding out whether the midpoint is in the plane, that is $|T_1 \cdot P_m| = 0$. The computational complexity of the algorithm is $\mathcal{O}(N^4)$, which is much faster than the combinatorial matching approach: e.g. there are only $N^2 \cdot (N-1)^2/4$=741,321 combinations based on L_1 (N=42), which is 3.71×10^{25} faster than the naive method. In experiments, we select n to be 1.

Algorithm 1: Symmetry detection by iterative pairing

Input : N: number of viewpoints;
$Pos[N]$: positions of N viewpoints;
$E[N]$: entropy values of N viewpoints;
n: icosahedron subdivision level;
δ=0.015: entropy difference threshold;
ϵ=1e-5: small difference in double values

Output: Symmetry planes' equations, if applicable

begin

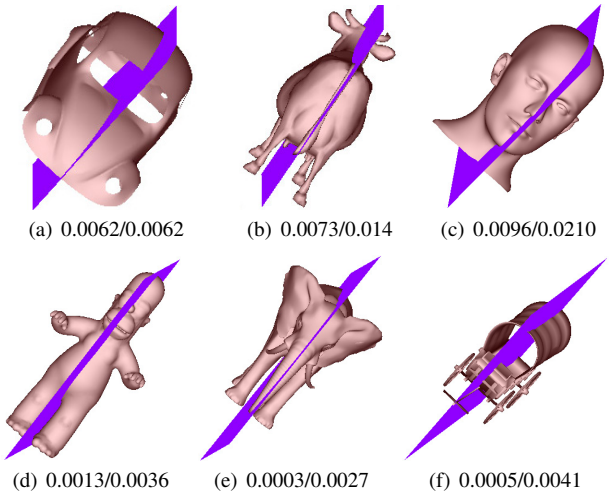
```
// loop symmetric viewpoint pairs (u, v)
for u ← 0 to N − 2 do
    P_u ⟵ Pos[u];
    for v ← u + 1 to N − 1 do
        if |E[u]−E[v]| >δ*min{E[u],E[v]} then
            continue;
        P_v ⟵ Pos[v], T_1 ⟵ normalize(P_u − P_v);
        matches ⟵ 2;
        // verify other matching pairs
        for i ← 0 to N − 2 do
            if i == u OR i == v then
                continue;
            P_i ⟵ Pos[i];
            for j ← i + 1 to N − 1 do
                if j == u OR j == v OR j == i then
                    continue;
                if |E[i]−E[j]| >δ*min{E[i],E[j]} then
                    continue;
                P_j ⟵ Pos[j], P_m ⟵ (P_i+P_j)/2;
                T_2 = normalize(P_i − P_j);
                CT = T_1 × T_2, DT = T_1 · T_2;
                if ||CT|| > ε AND |DT| ≠ 0 then
                    continue;
                if |T1 · P_m| > ε then
                    continue;
                matches=matches+2;
                break;
        // output the symmetry plane
        if matches ≥ N − 2^{n+2} then
            Output and visualize the symmetry plane:
            T_1[0] ∗ x + T_1[1] ∗ y + T_1[2] ∗ z = 0
```

4 Experiments and Discussions

4.1 Evaluation w.r.t to Dataset-Level Performance

We have tested our algorithm on the NIST benchmark [Fang et al. 2008] and selected models from the AIM@SHAPE Shape Repository [AIM@SHAPE 2014] to compare with state-of-the-art approaches like the Mean shift [Mitra et al. 2006] and 3D Hough transform [Cailliere et al. 2008] based methods, which are among the few papers that deal with global symmetry detection and at the same time provide a quantitative evaluation based on a common set of 3D models. Experiments show that our approach can stably detect the symmetry planes of diverse symmetric models and it also can detect a symmetry plane for a rough symmetric model with a bigger threshold δ. Figure 4 demonstrates several examples while Table 1 compares their timing information. We need to mention that due to the difference in the specifications of the CPUs used in the experiments, we do not directly compare the absolute running time, but rather we focus on the change of the running time with respect to the increase in the number of vertices of the 3D models. As can be seen, our method shows better computational efficiency property in terms of scalability to the number of vertices. This is mainly because the computational time does not increase linearly with the increase in the number of vertices of a 3D model since we just render the 3D model first and detect its symmetry only based on the rendered views. However, for the other two geometry-based approaches Mean shift and 3D Hough, their computational time is highly affected by the number of vertices of the model.

To measure the accuracy of the detected symmetry planes, we adopt the mean (normalized by the surface area) and maximum (w.r.t the bounding box diagonal) distance errors developed in Metro [Cignoni et al. 1998] which is based on surface sampling and point-to-surface distance computation. Table 2 compares the mean and max errors of the four models in Table 1 (see Fig. 4 for the errors of other models) with the Mean shift [Mitra et al. 2006] and 3D Hough transform [Cailliere et al. 2008] based methods. The errors are computed based on the original mesh and its reflected 3D model w.r.t the detected symmetry plane. As can be seen, our approach achieves much (4∼6 times w.r.t 3D Hough transform and 11∼44 times w.r.t Mean shift) better overall accuracy (see the mean errors), in spite that a few points may not be the most accurate but they still maintain a moderate accuracy (indicated by the max errors).



(a) 0.0062/0.0062 (b) 0.0073/0.014 (c) 0.0096/0.0210

(d) 0.0013/0.0036 (e) 0.0003/0.0027 (f) 0.0005/0.0041

Figure 4: *Example symmetry detection results with mean/max error measures [Cignoni et al. 1998].*

Table 1: *Timing information (in seconds) comparison of our methods and other two state-of-the-art approaches: Mean shift [Mitra et al. 2006] and 3D Hough [Cailliere et al. 2008] are based on a Pentium M 1.7 GHz CPU according to [Cailliere et al. 2008]; while our method is using an Intel(R) Xeon(R) X5675 @ 3.07GHz CPU.*

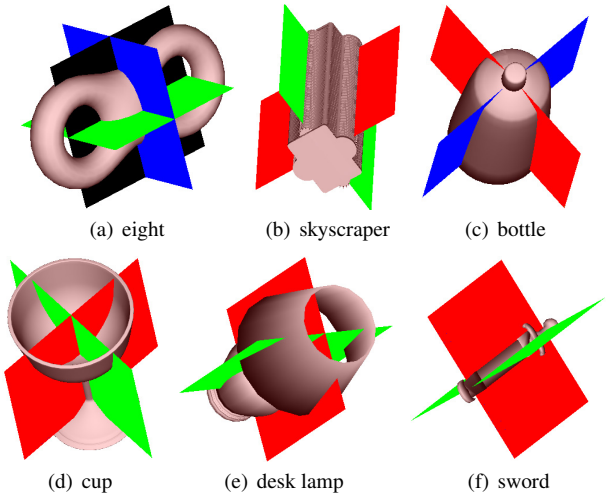
Models	Cube	Beetle	Homer	Mannequin
#Vertices	602	988	5,103	6,743
Mean shift	1.8	6.0	91.0	165.0
3D Hough	2.2	3.0	22.0	33.0
Our method	0.7	0.8	1.0	1.1

In addition, it is also very convenient to detect different degrees of symmetries via control of the entropy difference threshold δ. As shown in Fig. 4, there is a minor asymmetry on the the tail part of the cow, while other parts are symmetric. If we want to obtain strict symmetry, a smaller threshold δ (e.g. by reducing it by half:

Table 2: *Mean/max errors [Cignoni et al. 1998] comparison of our methods and other two state-of-the-art approaches. For the Cube model, since there are three detected symmetry planes, we use their normal directions (x/y/z) to differentiate them.*

Methods	Cube		Beetle		Homer		Mannequin	
	mean	max	mean	max	mean	max	mean	max
Mean shift [Mitra et al. 2006]	N.A.	N.A.	N.A.	N.A.	0.059	0.018	0.111	0.037
3D Hough [Cailliere et al. 2008]	N.A.	N.A.	N.A.	N.A.	0.007	0.001	0.046	0.009
Our method	0.0005 (x) 0.0057 (y, z)	0.0008 (x) 0.0082 (y, z)	0.0062	0.0062	0.0013	0.0036	0.0096	0.0210

0.0075) will give the result that it is asymmetric. We also find that our approach can simultaneously detect multiple symmetry planes for certain types of meshes, such as the Eight, Skyscraper, Bottle, Cup, Desk Lamp, and Sword in [AIM@SHAPE 2014] and [Fang et al. 2008], as shown in Fig. 5. But we need to mention due to the limitation of CPCA and the sensitivity property to minor changes of the viewpoint entropy feature, there are a few fail cases or certain cases where the proposed method can only partially determine a set of reflection planes. Examples of such models are non-uniform cubes, butterflies, tori, and pears, as demonstrated in Fig. 6: (a) because of non-uniform triangulation, the cube model cannot be perfectly aligned with CPCA, resulting in the unsuccessful symmetry plane detection. However, we have found that for most symmetric models (e.g. Mug, NonWheelChair, and WheelChair classes) that cannot be perfectly aligned with CPCA [Vranic 2004], our approach can still successfully detect their symmetry planes (e.g. the detection rates of Algorithm 1 for those types of models mentioned above are as follow: Mug: 7/8, NonWheelChair: 18/19, and WheelChair: 6/7). Three examples can be found in Fig. 7; (b) the symmetry plane of the butterfly cannot be detected if based on the default threshold δ=0.015, and only after increasing it till 0.0166 we can detect the plane; (c) only the red symmetry plane of the torus is detected based on the default threshold value, while both the red and green planes will be detected if we increase the threshold δ to 0.02 and all the three symmetry planes can be detected if we further increase it till 0.0215; (d) a false positive (blue) symmetry plane of the pear model will appear under the condition of the default threshold, however the error will be corrected with a little smaller threshold of 0.0133. An adaptive strategy of threshold selection is among our next work plan.

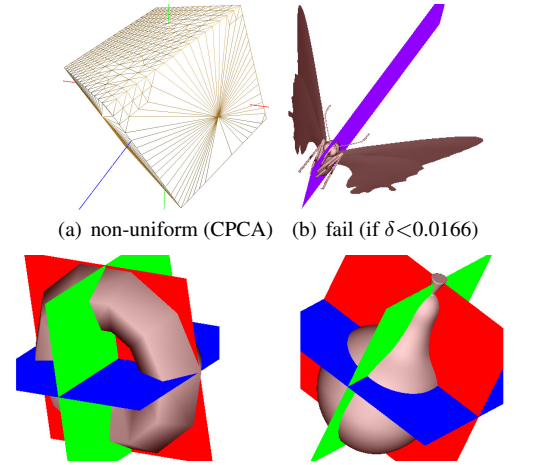


(a) eight (b) skyscraper (c) bottle

(d) cup (e) desk lamp (f) sword

Figure 5: *Multiple detected symmetry planes examples.*

Finally, we evaluate the overall performance of our viewpoint entropy distribution-based symmetry detection algorithm based on the



(a) non-uniform (CPCA) (b) fail (if δ<0.0166)

(c) partially (if δ<0.0215) (d) one false positive (if δ>0.0133)

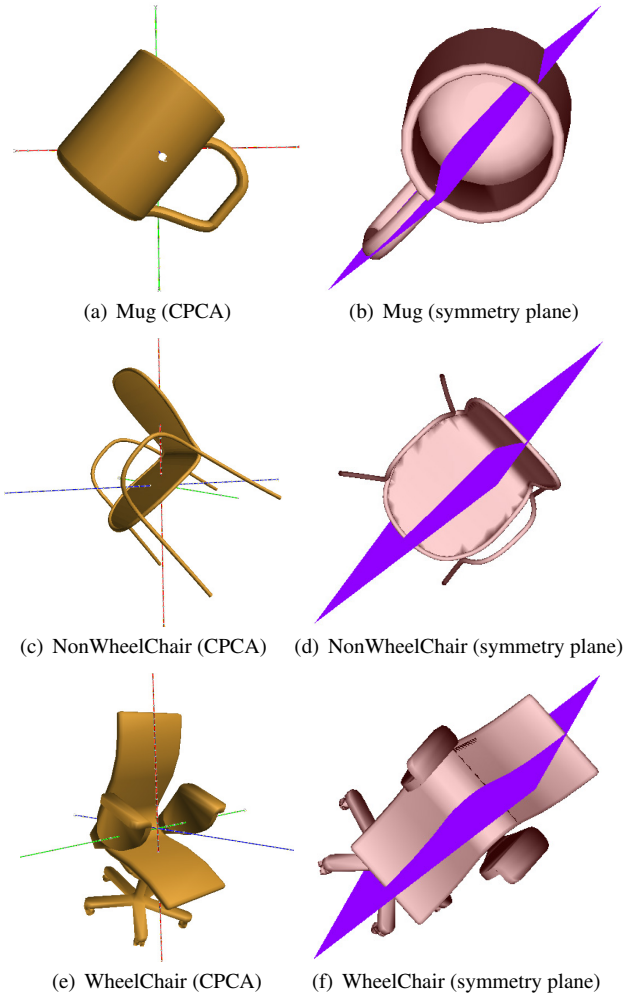
Figure 6: *Failed or partially failed examples.*

NIST benchmark [Fang et al. 2008]. In total, we have detected 647 symmetry planes for all the 800 models (some of them are asymmetric). To know the general performance of our algorithm, we manually observe the symmetry property of each of the first 200 models and label its symmetry plane(s)/degree(s) to form the ground truth. Then, we examine each detected symmetry plane to see whether it is a True Positive (TP) or False Positive (FP). Similarly, we set the True Negative (TN) value of a model to be 1 if it is asymmetric and our algorithm also does not detect any symmetry plane. While, if a symmetry plane of a symmetric model is not detected, we increase its False Negative (FN) by 1. Table 3 gives the evaluation results (177 detected symmetry planes) on the 200 models (having 194 symmetry planes in total), which are uniformly divided into 10 classes (Bird, Fish, NonFlyingInsect, FlyingInsect, Biped, Quadruped, ApartmentHouse, Skyscraper, SingleHouse and Bottle).

Table 3: *Overall symmetry detection performance of our algorithm based on the first 200 models of the NIST benchmark.*

Metrics	TP	FP	TN	FN
#	149	28	37	32

Based on the TP, FP, TN and FN values, we compute the following nine detection evaluation metrics [Manohar et al. 2006], as listed in Table 4: Tracker Detection Rate (TRDR, $\frac{TP}{TG}$), False Alarm Rate (FAR, $\frac{FP}{TP+FP}$), Detection Rate (DR, $\frac{TP}{TP+FN}$), Specificity (SP, $\frac{TN}{FP+TN}$), Accuracy (AC, $\frac{TP+TN}{TF}$), Positive Prediction (PP, $\frac{TP}{TP+FP}$), Negative Prediction (NP, $\frac{TN}{FN+TN}$), False Negative Rate (FNR or Miss Rate, $\frac{FN}{FN+TP}$), and False Positive Rate (FPR, $\frac{FP}{FP+TN}$), where the total number of symme-



(a) Mug (CPCA) (b) Mug (symmetry plane)

(c) NonWheelChair (CPCA) (d) NonWheelChair (symmetry plane)

(e) WheelChair (CPCA) (f) WheelChair (symmetry plane)

Figure 7: *Examples to demonstrate that our algorithm can successfully detect the symmetry planes for most symmetric models that are not perfectly aligned with CPCA: first column shows the CPCA aligned results; second column demonstrates the detected symmetry planes.*

try planes in the 200 Ground Truth models TG=194 and the total number of our detections (including both trues and falses) TF=TP+FP+TN+FN=246. As can be seen, besides the better accuracy in the detected symmetry planes as mentioned before, our detection performance (e.g., Detection Rate DR=82.32%, Tracker Detection Rate TRDR=76.80%) is also good enough.

In a word, as demonstrated by all the above evaluation results, better accuracy and efficiency than state-of-the-art approaches have been achieved by our simple but effective symmetry detection method.

4.2 Evaluation w.r.t to Robustness

Robustness to View Sampling We also test our algorithm with different levels of subdivided icosahedron for the view sampling, e.g., L_2, L_3, and L_4. Table 5 compares the mean/max errors and running time for the four models listed in Table 1. As can be seen, increasing the view sampling often cannot increase the accuracy while the running time will be significantly increasing. Thus, we choose to sample the views based on L_1 which gives better overall performance in both the accuracy and efficiency.

Robustness to Number of Vertices We test the robustness of our algorithm w.r.t the change of the (especially large) number of vertices (resolution) that a 3D model contains. We first subdivide a triangular mesh into its finer version based on several iterations of midpoint subdivision by utilizing the tool of MeshLab [MeshLab 2014] and then use the resulting meshes for the test and comparison. We have tested the Elephant, Mannequin and Cube models, and found that our algorithm can stably and accurately detect their symmetry planes, independent of the number of vertices. Table 6 compares their mean/max errors and timings. We can see that the increase in computational time is often significantly slower (especially for models with an extremely large number of vertices; e.g. for Mannequin (467,587 vertices) and Cube (196,610 vertices) they are about 8 and 28 times slower, respectively) than the increase in the number of vertices since rendering the sampling views to compute their viewpoint entropy dominates the running time.

Table 6: *Mean/max errors and timing comparison of our algorithm w.r.t the robustness to different number of vertices. For the Cube model, since there are three detected symmetry planes, we use their normal directions (x/y/z) to differentiate them.*

Models	#Vertices	mean	max	time
Elephant	29,285	0.0003	0.0027	3.0
	116,920	0.0003	0.0027	12.3
	467,252	0.0003	0.0027	48.4
Mannequin	17,450	0.0091	0.0210	2.6
	29,194	0.0091	0.0210	3.8
	467,587	0.0091	0.0210	48.2
Cube	6,146	0.0050 (x)	0.0077 (x)	1.5
		0.0082 (y)	0.0137 (y)	
		0.0061 (z)	0.0093 (z)	
	24,578	0.0002 (x)	0.0003 (x)	3.0
		0.0002 (y)	0.0004 (y)	
		0.0001 (z)	0.0001 (z)	
	196,610	0.0003 (x)	0.0005 (x)	5.8
		0.0003 (y)	0.0004 (y)	
		0.0001 (z)	0.0002 (z)	

5 Conclusions and Future Work

In this paper, we have proposed an efficient and novel view-based symmetry detection algorithm based on viewpoint entropy distribution. We have compared with the two latest symmetry detection approaches based on a common set of selected models and demonstrated the better performance of our method in terms of accuracy and efficiency. A detailed evaluation of our approach on a dataset of 200 models also validates its good robustness, detection rate, and flexibility.

To further improve and explore the algorithm, we list several promising directions here as our next work. Firstly, traditional PCA-based approaches cannot guarantee that the directions of the largest extent are parallel to the axes of the ideal canonical coordinate frame (front-back, left-right, or top-bottom view) of 3D models. One promising approach to achieve further improvement in terms of alignment accuracy is the Minimum Projection Area (MPA) [Johan et al. 2011] alignment method, which can align most 3D models in terms of axes accuracy (the axes are parallel to the ideal canonical coordinate frame) and is also robust to model variations, noise, and initial poses. We are interested in combining it with CPCA for the 3D alignment process: first performing CPCA for an initial alignment and then correcting possible tilt views (poses) by utilizing a similar idea as MPA. It is promising to help to achieve even better symmetry detection performance, especially for decreasing the percentage of False Negative (FN).

Table 4: *Overall symmetry detection accuracy of our algorithm based on the first 200 models of the NIST benchmark.*

Metrics	TRDR	FAR	DR	SP	AC	PP	NP	FNR	FPR
Values	76.80%	15.82%	82.32%	56.92%	75.61%	84.18%	53.62%	17.68%	43.08%

Table 5: *Mean/max errors and timing comparison of our algorithm with different view sampling. For the Cube model, since there are three detected symmetry planes, we use their normal directions (x/y/z) to differentiate them.*

View sampling	Cube			Beetle			Homer			Mannequin		
	mean	max	time	mean	max	time	mean	max	time	mean	max	time
L_1	0.0005 (x) 0.0057 (y) 0.0057 (z)	0.0008 (x) 0.0082 (y) 0.0082 (z)	0.7	0.0062	0.0062	0.8	0.0013	0.0036	1.0	0.0096	0.0210	1.1
L_2	0.0005 (x)	0.0008 (x)	3.4	0.0062	0.0062	3.6	0.0013	0.0036	3.8	0.0096	0.0210	3.7
L_3	0.0057 (y)	0.0082 (y)	22.6	0.0062	0.0062	16.9	0.0013	0.0036	19.5	0.0096	0.0210	27.3
L_4	0.0057 (z)	0.0082 (z)	2481.7	0.0062	0.0062	1048.0	0.0013	0.0036	1600.5	0.0096	0.0210	3465.1

Secondly, an automatic and adaptive strategy to select an appropriate threshold δ for respective models or classes is another interesting research direction and deserves our further exploration.

Finally, we also plan to explore several interesting applications of our symmetry detection algorithm, such as 3D model alignment and best views selection. Based on the detected symmetry planes and the basic idea of PCA, it is straightforward to apply our algorithm in 3D alignment: the first principal axis gives maximum symmetry degree (that is, it has the smallest total matching cost in terms of entropy for the symmetric viewpoint pairs w.r.t the axis) and the second principal axis is both perpendicular to the first axis and also has a maximum symmetry degree. It is promising to achieve similar results as those in [Podolak et al. 2006] while outperforms either PCA or CPCA for certain models with symmetry plane(s). However, our algorithm has better efficiency property than [Podolak et al. 2006], thus will be more promising for related real-time applications including 3D model retrieval.

Our symmetry is related to viewpoint entropy which indicates the amount of information that a view contains. In an analogy to 3D model alignment, we plan to use the total viewpoint entropy matching cost to indicate the goodness of a candidate best view w.r.t a viewpoint: the bigger the summed matching cost is, the better the viewpoint is, since it indicates that there is less redundant information in the view. Algorithm 1 targets finding the minimum viewpoint matching cost in terms of entropy, while we now need to find the viewpoint that gives a maximum viewpoint matching cost.

Acknowledgments

The work of Bo Li, Yuxiang Ye and Yijuan Lu is supported by the Texas State University Research Enhancement Program (REP), Army Research Office grant W911NF-12-1-0057, and NSF CRI 1305302 to Dr. Yijuan Lu.

Henry Johan is supported by Fraunhofer IDM@NTU, which is funded by the National Research Foundation (NRF) and managed through the multi-agency Interactive & Digital Media Programme Office (IDMPO) hosted by the Media Development Authority of Singapore (MDA).

References

AIM@SHAPE, 2014. http://shapes.aimatshape.net/.

AUBRY, M., SCHLICKEWEI, U., AND CREMERS, D. 2011. The wave kernel signature: A quantum mechanical approach to shape analysis. In *ICCV Workshops*, IEEE, 1626–1633.

BERNER, A., WAND, M., MITRA, N. J., MEWES, D., AND SEIDEL, H.-P. 2011. Shape analysis with subspace symmetries. *Comput. Graph. Forum 30*, 2, 277–286.

BOKELOH, M., BERNER, A., WAND, M., SEIDEL, H.-P., AND SCHILLING, A. 2009. Symmetry detection using feature lines. *Comput. Graph. Forum 28*, 2, 697–706.

CAILLIERE, D., DENIS, F., PELE, D., AND BASKURT, A. 2008. 3D mirror symmetry detection using hough transform. In *ICIP*, IEEE, 1772–1775.

CAO, J., TAGLIASACCHI, A., OLSON, M., ZHANG, H., AND SU, Z. 2010. Point cloud skeletons via laplacian-based contraction. In *Proc. of IEEE Conf. on Shape Modeling and Applications*, 187–197.

CHAOUCH, M., AND VERROUST-BLONDET, A. 2009. Alignment of 3D models. *Graphical Models 71*, 2, 63–76.

CIGNONI, P., ROCCHINI, C., AND SCOPIGNO, R. 1998. Metro: Measuring error on simplified surfaces. *Comput. Graph. Forum 17*, 2, 167–174.

FANG, R., GODIL, A., LI, X., AND WAGAN, A. 2008. A new shape benchmark for 3D object retrieval. In *ISVC (1)*, Springer, G. Bebis and et al., Eds., vol. 5358 of *Lecture Notes in Computer Science*, 381–392.

GOLOVINSKIY, A., PODOLAK, J., AND FUNKHOUSER, T. A. 2009. Symmetry-aware mesh processing. In *IMA Conference on the Mathematics of Surfaces*, Springer, E. R. Hancock, R. R. Martin, and M. A. Sabin, Eds., vol. 5654 of *Lecture Notes in Computer Science*, 170–188.

JOHAN, H., LI, B., WEI, Y., AND ISKANDARSYAH. 2011. 3D model alignment based on minimum projection area. *The Visual Computer 27*, 6-8, 565–574.

JOLLIFFE, I. 2002. *Principal Component Analysis (2nd edition)*. Springer, Heidelberg.

KAZHDAN, M. M., FUNKHOUSER, T. A., AND RUSINKIEWICZ, S. 2004. Symmetry descriptors and 3D shape matching. In *Symp. on Geom. Process.*, Eurographics Association, J.-D. Boissonnat and P. Alliez, Eds., vol. 71 of *ACM International Conference Proceeding Series*, 115–123.

KIM, V. G., LIPMAN, Y., CHEN, X., AND FUNKHOUSER, T. A. 2010. Möbius transformations for global intrinsic symmetry analysis. *Comput. Graph. Forum 29*, 5, 1689–1700.

LIPMAN, Y., AND FUNKHOUSER, T. A. 2009. Möbius voting for surface correspondence. *ACM Trans. Graph. 28*, 3.

LIU, J., AND LIU, Y. 2010. Curved reflection symmetry detection with self-validation. In *ACCV (4)*, Springer, R. Kimmel, R. Klette, and A. Sugimoto, Eds., vol. 6495 of *Lecture Notes in Computer Science*, 102–114.

LIU, Y., HEL-OR, H., KAPLAN, C. S., AND GOOL, L. J. V. 2010. Computational symmetry in computer vision and computer graphics. *Foundations and Trends in Computer Graphics and Vision 5*, 1-2, 1–195.

LOOP, C. 1987. *Smooth Subdivision Surfaces Based on Triangles*. Master's thesis, University of Utah.

MANOHAR, V., SOUNDARARAJAN, P., RAJU, H., GOLDGOF, D. B., KASTURI, R., AND GAROFOLO, J. S. 2006. Performance evaluation of object detection and tracking in video. In *ACCV (2)*, Springer, P. J. Narayanan, S. K. Nayar, and H.-Y. Shum, Eds., vol. 3852 of *Lecture Notes in Computer Science*, 151–161.

MARTINET, A., SOLER, C., HOLZSCHUCH, N., AND SILLION, F. X. 2006. Accurate detection of symmetries in 3D shapes. *ACM Trans. Graph. 25*, 2, 439–464.

MESHLAB, 2014. http://meshlab.sourceforge.net/.

MITRA, N. J., GUIBAS, L. J., AND PAULY, M. 2006. Partial and approximate symmetry detection for 3D geometry. *ACM Trans. Graph. 25*, 3, 560–568.

MITRA, N. J., PAULY, M., WAND, M., AND CEYLAN, D. 2012. Symmetry in 3D geometry: Extraction and applications. In *EUROGRAPHICS State-of-the-art Report*.

PODOLAK, J., SHILANE, P., GOLOVINSKIY, A., RUSINKIEWICZ, S., AND FUNKHOUSER, T. A. 2006. A planar-reflective symmetry transform for 3D shapes. *ACM Trans. Graph. 25*, 3, 549–559.

PODOLAK, J., GOLOVINSKIY, A., AND RUSINKIEWICZ, S. 2007. Symmetry-enhanced remeshing of surfaces. In *Symp. on Geom. Process.*, Eurographics Association, A. G. Belyaev and M. Garland, Eds., vol. 257 of *ACM International Conference Proceeding Series*, 235–242.

RAVIV, D., BRONSTEIN, A. M., BRONSTEIN, M. M., AND KIMMEL, R. 2010. Full and partial symmetries of non-rigid shapes. *International Journal of Computer Vision 89*, 1, 18–39.

SAWADA, T., AND PIZLO, Z. 2008. Detecting mirror-symmetry of a volumetric shape from its single 2D image. In *CVPR Workshops, CVPRW '08*, 1 –8.

SAWADA, T. 2010. Visual detection of symmetry of 3D shapes. *Journal of Vision 10*, 6, 4:1–4:22.

SFIKAS, K., THEOHARIS, T., AND PRATIKAKIS, I. 2011. Rosy+: 3D object pose normalization based on PCA and reflective object symmetry with application in 3D object retrieval. *Int. J. Comput. Vis. 91*, 3, 262–279.

SIMARI, P. D., NOWROUZEZAHRAI, D., KALOGERAKIS, E., AND SINGH, K. 2009. Multi-objective shape segmentation and labeling. *Comput. Graph. Forum 28*, 5, 1415–1425.

SUN, J., OVSJANIKOV, M., AND GUIBAS, L. J. 2009. A concise and provably informative multi-scale signature based on heat diffusion. *Comput. Graph. Forum 28*, 5, 1383–1392.

TAGLIASACCHI, A., ZHANG, H., AND COHEN-OR, D. 2009. Curve skeleton extraction from incomplete point cloud. *ACM Transactions on Graphics (Special Issue of SIGGRAPH) 28*, 3, Article 71, 9 pages.

TAKAHASHI, S., FUJISHIRO, I., TAKESHIMA, Y., AND NISHITA, T. 2005. A feature-driven approach to locating optimal viewpoints for volume visualization. In *IEEE Visualization*, IEEE Computer Society, 495–502.

TEDJOKUSUMO, J., AND LEOW, W. K. 2007. Normalization and alignment of 3D objects based on bilateral symmetry planes. In *MMM (1)*, Springer, T.-J. Cham, J. Cai, C. Dorai, D. Rajan, T.-S. Chua, and L.-T. Chia, Eds., vol. 4351 of *Lecture Notes in Computer Science*, 74–85.

TUZIKOV, A. V., COLLIOT, O., AND BLOCH, I. 2003. Evaluation of the symmetry plane in 3D MR brain images. *Pattern Recogn. Lett. 24*, 14, 2219–2233.

VÁZQUEZ, P.-P., FEIXAS, M., SBERT, M., AND HEIDRICH, W. 2001. Viewpoint selection using viewpoint entropy. In *VMV*, Aka GmbH, T. Ertl, B. Girod, H. Niemann, and H.-P. Seidel, Eds., 273–280.

VRANIC, D. 2004. *3D Model Retrieval*. PhD thesis, University of Leipzig.

WANG, Y., XU, K., LI, J., ZHANG, H., SHAMIR, A., LIU, L., CHENG, Z.-Q., AND XIONG, Y. 2011. Symmetry hierarchy of man-made objects. *Comput. Graph. Forum 30*, 2, 287–296.

WANG, H., SIMARI, P., SU, Z., AND ZHANG, H. 2014. Spectral global intrinsic symmetry invariant functions. *Proc. of Graphics Interface*.

XU, K., ZHANG, H., TAGLIASACCHI, A., LIU, L., LI, G., MENG, M., AND XIONG, Y. 2009. Partial intrinsic reflectional symmetry of 3D shapes. *ACM Trans. Graph. 28*, 5.

XU, K., ZHANG, H., JIANG, W., DYER, R., CHENG, Z., LIU, L., AND CHEN, B. 2012. Multi-scale partial intrinsic symmetry detection. *ACM Transactions on Graphics (Special Issue of SIGGRAPH Asia) 31*, 6, 181:1–181:10.

ZABRODSKY, H., PELEG, S., AND AVNIR, D. 1995. Symmetry as a continuous feature. *IEEE Trans. Pattern Anal. Mach. Intell. 17*, 12, 1154–1166.

ZOU, H. L., AND LEE, Y. T. 2005. Skewed mirror symmetry detection from a 2D sketch of a 3D model. In *GRAPHITE*, ACM, S. N. Spencer, Ed., 69–76.

ZOU, H. L., AND LEE, Y. T. 2006. Skewed rotational symmetry detection from a 2D line drawing of a 3D polyhedral object. *Computer-Aided Design 38*, 12, 1224–1232.