

Efficient 3D Reflection Symmetry Detection: a View-Based Approach

Bo Li^{a,*}, Henry Johan^b, Yuxiang Ye^a, Yijuan Lu^a

^a Department of Computer Science, Texas State University, San Marcos, USA

^b Visual Computing, Fraunhofer IDM@NTU, Singapore

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. Experimental results and two related applications demonstrate better accuracy, efficiency, robustness and versatility of our algorithm than state-of-the-art approaches.

Keywords:

symmetry detection, reflection symmetry, view-based approach, 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 [1]. Symmetry has been used in many applications such as: 3D alignment [2], shape matching [3], remeshing [4], 3D model segmentation [5] and retrieval [6].

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. Though sampling and simplification can be used to reduce the number of vertices, they also decrease the shape accuracy and cause deviations in geometry. Therefore, a symmetry detection algorithm often directly uses original models as its input, as can be found in many existing related papers.

Motivated by the symmetric patterns existing in the viewpoint entropy [7] 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) [8]. 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. In Section 5, we show two interesting applications of our symmetry detection idea in 3D model alignment and best view selection. Section 6 concludes the paper and lists several future research directions. This paper is an extension of our prior publication [9].

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 [2] 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 symmetric [10].

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 [11]. We classify current

*Corresponding author at: 601 University Drive, Department of Computer Science, Texas State University, San Marcos, Texas 78666; E-mail: B.L58@txstate.edu, li.bo.ntu0@gmail.com; Tel: +001 512 245 6580; Fax: +001 512 245 8750.

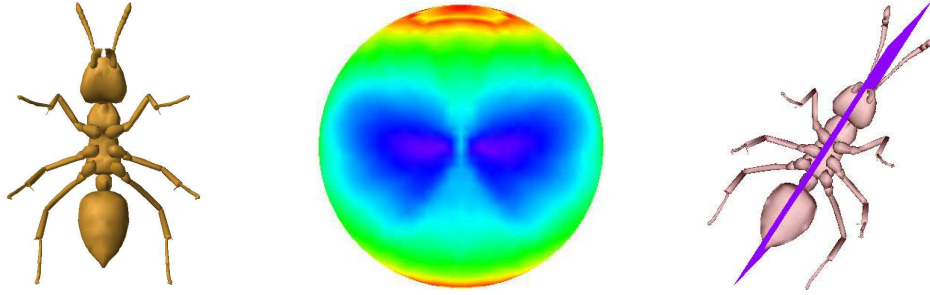


Figure 1: An overview of our view-based symmetry detection algorithm: an example of an ant model, its viewpoint entropy distribution, and the detected symmetry plane by matching the viewpoints.

57 symmetry detection techniques into the following four groups
58 according to the features employed.

59 **Symmetry detection based on pairing point features.** This
60 type of approach first samples points on the surface of a 3D
61 model and then extracts their features. After that, it finds point
62 pairs by matching the points. Based on the point pairs, symme-
63 try evidences are accumulated to decide the symmetry plane.
64 Two typical algorithms are [12] and [13]. To decide the symme-
65 try plane, Mitra et al. [12] adopted a stochastic clustering and
66 region-growing approach, while Calliere et al. [13] followed
67 the same framework of pairing and clustering, but utilized 3D
68 Hough transform to extract significant symmetries. In fact, the
69 initial idea of this approach can be traced back to the symmetry
70 distance defined in [14]. Podolak et al. [15] proposed a planar-
71 reflective symmetry transform and based on the transform they
72 defined two 3D features named center of symmetry and prin-
73 cipal symmetry axes, which are useful for related applications
74 such as 3D model alignment, segmentation, and viewpoint se-
75 lection.

76 **Symmetry detection based on pairing line features.**
77 Bokeloh et al. [16] targeted on the so-called rigid symmetries
78 by matching feature lines. Rigid symmetries are the reoccur-
79 ing components with differences only in rigid transformations
80 (translation, rotation and mirror). They first extracted feature
81 lines of a 3D model, then performed feature line matching, and
82 finally validated the symmetry based on the feature correspon-
83 dence information by adopting a region growing approach, as
84 well.

85 **Symmetry detection based on 2D image features.** Sawada
86 and Pizlo [10] [17] performed symmetry detection based on a
87 single 2D image of a volumetric shape. First, a polyhedron
88 is recovered from the single 2D image based on a set of con-
89 straints including 3D shape symmetry, minimum surface area,
90 maximum 3D compactness and maximum planarity of con-
91 tours. Then, they directly compared the two halves of the poly-
92 hedron to decide its symmetry degree. From a psychological
93 perspective, Zou and Lee [18] [19] proposed one method to
94 detect the skewed rotational and mirror symmetry respectively
95 from a CAD line drawing based on a topological analysis of the
96 edge connections.

97 **Other symmetry detection approaches.** Martinet et al. [20]
98 proposed a 3D feature named generalized moments for symme-
99 try detection. Rather than directly computing original moments

100 features, they mapped them into another feature space by spher-
101 ical harmonics transform and then searched for the global sym-
102 metry in the new feature space. Xu et al. [21] developed an al-
103 gorithm to detect partial intrinsic reflectional symmetry based
104 on an intrinsic reflectional symmetry axis transform. After that,
105 a multi-scale partial intrinsic symmetry detection algorithm was
106 proposed in [22]. There are also techniques to detect some other
107 specific symmetries, such as curved symmetry [23] and symme-
108 tries of non-rigid models [24] [25], as well as symmetry hier-
109 archy of a man-made 3D model [26]. Kim et al. [27] detected
110 global intrinsic symmetries of a 3D model based on Möbius
111 Transformations [28], a stereographic projection approach in
112 geometry. Recently, Wang et al. [29] proposed Spectral Global
113 Intrinsic Symmetry Invariant Functions (GISIFs), which are ro-
114 bust to local topological changes compared to the GISIFs ob-
115 tained from geodesic distances. Their generality and flexibil-
116 ity outperform the two classical GISIFs: Heat Kernel Signature
117 (HKS) [30] and Wave Kernel Signature (WKS) [31].

118 All above and existing symmetry detection techniques can
119 be categorized into geometry-based approach. However, distin-
120 ctively different from them, we adopt a view-based approach
121 to accumulate the geometrical information of many vertices to-
122 gether into a view in order to more efficiently detect the reflec-
123 tion symmetry of a 3D model, which also serves as the novelty
124 and main contribution of our method.

125 *Symmetry Applications.* As an important shape feature, sym-
126 metry is useful for many related applications. For example,
127 they include symmetry plane detection for 3D MRI image [32],
128 shape matching [3] [15], 3D model alignment [33] [6], shape
129 processing and analysis [34] including remeshing [4], sym-
130 metrization [12], viewpoint selection [15], and subspace shape
131 analysis [35], 3D segmentation [15] [5] [29], and curve skeleton
132 extraction [36] [37].

133 3. Symmetry Detection Algorithm

134 3.1. 3D Model Normalization Based on CPCA

135 Properly normalizing a 3D model before symmetry detec-
136 tion can help us to minimize the searching space for symmetry
137 planes to be some 2D planes that have certain common specific
138 properties, i.e., passing the same 3D point. The process of 3D
139 normalization includes three steps: 3D alignment (orientation

140 normalization), translation (position normalization), and scal-
 141 ing (size normalization).

142 3D model alignment is to transform a model into a canonical
 143 coordinate frame, where the representation of the model is inde-
 144 pendent of its scale, orientation, and position. Two commonly
 145 used 3D model alignment methods are Principal Component
 146 Analysis (PCA) [38] and its descendant Continuous Principal
 147 Component Analysis (CPCA) [8] which considers the area of
 148 each face. They utilize the statistical information of vertex co-
 149 ordinates and extract three orthogonal components with largest
 150 extent to depict the principal axes of a 3D model. CPCA is gen-
 151 erally regarded as a more stable PCA-based method. In addition,
 152 Johan et al. [39] proposed a 3D alignment algorithm based
 153 on Minimum Projection Area (MPA) motivated by the fact that
 154 many objects have normalized poses with minimum projection
 155 areas. That is, for many objects, one of their canonical views
 156 has a minimum projection area compared to the other arbitrary
 157 views of the objects. Therefore, they align a 3D model by suc-
 158 cessively selecting two perpendicular axes with minimum pro-
 159 jection areas while the third axis is the cross product of the
 160 first two axes. It is shown in [39] that MPA can align most
 161 3D models in terms of axes accuracy (the axes are parallel to
 162 the ideal canonical coordinate frame: front-back, left-right, or
 163 top-bottom view). It is also robust to model variations, noise,
 164 and initial poses. However, compared with the PCA-based ap-
 165 proaches, MPA takes a longer time to align 3D models while
 166 for this research we want to detect symmetry fast.

167 After a comparison (see Section 4.3 for more details) of the
 168 influences of different 3D model alignment algorithms on the
 169 efficiency, accuracy and robustness of our view-based symme-
 170 try detection approach, we choose CPCA to align a model be-
 171 fore performing symmetry detection. After the alignment with
 172 CPCA, we translate the model such that the center of its bound-
 173 ing sphere locates at the origin and scale the model such that
 174 its bounding sphere has a radius of 1. After this normalization,
 175 the symmetry plane(s) will pass the origin, which helps us to
 176 significantly reduce the searching space.

177 3.2. View Sampling and Viewpoint Entropy Distribution Gen- 178 eration

179 Vázquez et al. [7] proposed an information theory-related
 180 measurement named viewpoint entropy to depict the amount
 181 of information a view contains. It is formulated based on the
 182 Shannon entropy and incorporates both the projection area of
 183 each visible face and the number of visible faces into the defi-
 184 nition. However, the original definition was developed based on
 185 perspective projection, thus we use its extended version defined
 186 in [40] for orthogonal projection.

187 For each model, we sample a set of viewpoints based on the
 188 Loop subdivision [41] on a regular icosahedron, denoted as L_0 .
 189 We subdivide L_0 n times and denote the resulting mesh as L_n .
 190 Then, we set the cameras on its vertices, make them look at
 191 the origin (also the center of the model) and apply orthogonal
 192 projection for rendering. For a 3D model, to differentiate its
 193 different faces, we assign different color to each face during
 194 rendering. One example is shown in Fig. 2.

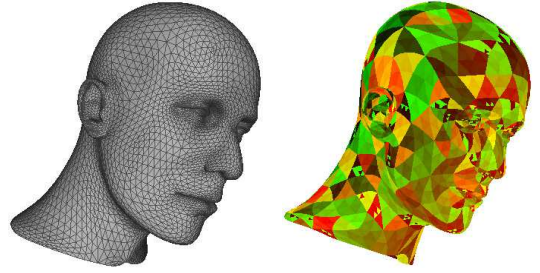


Figure 2: Face color coding example.

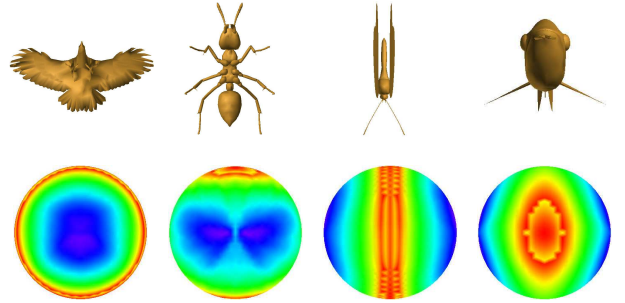


Figure 3: Viewpoint entropy distribution examples: 1st row shows the models after alignment with CPCA; 2nd row demonstrates their respective viewpoint entropy distribution. blue: large entropy; green: mid-size entropy; red: small entropy.

The viewpoint entropy [40] of a view with m visible faces is defined as follows.

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

195 where, A_j is the visible projection area of the j^{th} ($j=1, 2, \dots,$
 196 m) face of a 3D model and A_0 is the background area. S
 197 is the total area of the window where the model is rendered:
 198 $S=A_0+\sum_{j=1}^m A_j$. Projection area is computed by counting the
 199 total number of pixels inside a projected face.

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

219 3.3. Symmetry Detection Based on Iterative Feature Pairing

220 Even only using L_1 (42 viewpoints) for view sampling, if
 221 based on a naive matching approach by first directly selecting
 222 half of the total viewpoints and then matching them with the
 223 remaining half, it will result in $P(42, 21)=2.75 \times 10^{31}$ combina-
 224 tions. Thus, we develop a much more efficient symmetry de-
 225 tection method based on the following idea: iteratively select a
 226 matching pair of viewpoints to generate a symmetry plane and
 227 then verify all the rest matching pairs to see whether they are
 228 symmetric as well w.r.t the symmetry plane or at least in the
 229 symmetry plane. The method is listed in Algorithm 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;
 $\delta=0.015$: entropy difference threshold;
 $\epsilon=1e-5$: 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
  Pu ← Pos[u];
  for v ← u + 1 to N - 1 do
    if |E[u] - E[v]| > δ * min{E[u], E[v]} then
      continue;
    Pv ← Pos[v], T1 ← normalize(Pu - Pv);
    matches ← 2;
    // verify other matching pairs
    for i ← 0 to N - 2 do
      if i == u OR i == v then
        continue;
      Pi ← 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;
        Pj ← Pos[j], Pm ←  $\frac{P_i + P_j}{2}$ ;
        T2 = normalize(Pi - Pj);
        CT = T1 × T2, DT = T1 · T2;
        if ||CT|| > ε AND |DT| ≠ 0 then
          continue;
        if |T1 · Pm| > ε then
          continue;
        matches=matches+2;
        break;
    // output the symmetry plane
    if matches ≥ N - 2n+2 then
      Output and visualize the symmetry plane:
      T1[0] * x + T1[1] * y + T1[2] * z = 0
  
```

230 We need to mention the followings for the algorithm. The
 231 views corresponding to the viewpoints that are located on the
 232 symmetry plane do not need to match each other. While, ac-
 233 cording to the Loop rule [41], at most 2^{n+2} vertices of L_n are
 234 coplanar in a plane w.r.t a great circle. That is to say, at most
 235 2^{n+2} viewpoints could be in the real symmetry plane. An ideal
 236 algorithm is to perfectly match w.r.t the symmetry plane all
 237 the viewpoint pairs that are not in the symmetry plane. How-
 238 ever, we have found that usually there are numerical accuracy
 239 problems related to 3D model rendering (e.g. aliasing), view-
 240 point entropy computation (usually the entropy values of two
 241 symmetric viewpoints are not completely the same), as well as
 242 possible (either big or minor) differences in mesh triangulation.
 243 Therefore, we propose to partially solve this issue by relaxing
 244 some of the conditions though it sometimes causes certain false
 245 positive detections: if the total number (*matches*) of matched
 246 viewpoints w.r.t a candidate symmetry plane is at least $N - 2^{n+2}$,
 247 then it is confirmed as a symmetry plane. δ is a threshold which
 248 can control the strictness of symmetry definition. For example,
 249 using a small threshold we detect more strictly defined sym-
 250 metries while using a bigger threshold, we allow some minor
 251 differences and detect rough symmetry properties. T_1 and T_2
 252 are the normals of the planes w.r.t two correspondence points
 253 (P_u and P_v ; P_i and P_j). The condition $\|CT\| > \epsilon$ AND $|DT| \neq 0$
 254 means T_1 and T_2 is neither parallel nor perpendicular to each
 255 other. In another word, the line between P_i and P_j is not per-
 256 pendicular to the candidate symmetry plane since T_1 and T_2 are
 257 not parallel (otherwise, $\|CT\| = 0$); and P_i and P_j are also not
 258 in the symmetry plane (otherwise, $|DT| = 0$). P_m is the mid-
 259 point of the line segment connecting points P_i and P_j . It is
 260 used to further assert the vertical symmetry property of P_i and
 261 P_j about the candidate symmetry plane by finding out whether
 262 the midpoint is in the plane, that is $|T_1 \cdot P_m| = 0$. The compu-
 263 tational complexity of the algorithm is $O(N^4)$, which is much
 264 faster than the combinatorial matching approach: e.g. there are
 265 only $N^2 \cdot (N-1)^2 / 4 = 741,321$ combinations based on L_1 ($N=42$),
 266 which is 3.71×10^{25} faster than the naive method. In experi-
 267 ments, we select n to be 1.

268 4. Experiments and Discussions

269 4.1. Evaluation w.r.t to Dataset-Level Performance

270 We have tested our algorithm on the NIST benchmark [42]
 271 and selected models from the AIM@SHAPE Shape Reposi-
 272 tory [43] to compare with state-of-the-art approaches like the
 273 Mean shift [12] and 3D Hough transform [13] based methods,
 274 which are among the few papers that deal with global symmetry
 275 detection and at the same time provide a quantitative evalua-
 276 tion based on a common set of 3D models. 3D Hough trans-
 277 form [13] can only deal with global symmetry, while Mean
 278 shift [12] can deal with partial and approximate symmetry as
 279 well. Experiments show that our approach can stably detect the
 280 symmetry planes of diverse symmetric models and it also can
 281 detect a symmetry plane for a rough symmetric model with a
 282 bigger threshold δ .

283 Figure 4 demonstrates several examples while Table 1 com-
 284 pares their timing information. We need to mention that due

285 to the difference in the specifications of the CPUs used in the
 286 experiments, we do not directly compare the absolute running
 287 time, but rather we focus on the change of the running time with
 288 respect to the increase in the number of vertices of the 3D mod-
 289 els. As can be seen, our method shows better computational
 290 efficiency property in terms of scalability to the number of ver-
 291 tices. This is mainly because the computational time does not
 292 increase linearly with the increase in the number of vertices of
 293 a 3D model since we just render the 3D model first and detect
 294 its symmetry only based on the rendered views. However, for
 295 the other two geometry-based approaches Mean shift and 3D
 296 Hough, their computational time is highly affected by the num-
 297 ber of vertices of the model. This is because the computational
 298 complexity of Mean Shift (in the best case) and 3D Hough is
 299 $O(N \log N)$, where N is the number of pairs when only one it-
 300 eration is needed [13]. Since both of them are geometry-based
 301 approach, the value of N as well as their complexity is highly
 302 dependent on the number of vertices that a 3D model has. For
 303 our case, though the computational complexity of the viewpoint
 304 matching step (Section 3.3) is $O(N^4)$, the number of viewpoints
 305 N ($N=42$ in our experiments) is a constant number. Therefore,
 306 this matching step has a constant running cost, that is, it is not
 307 dependent on the number of vertices.

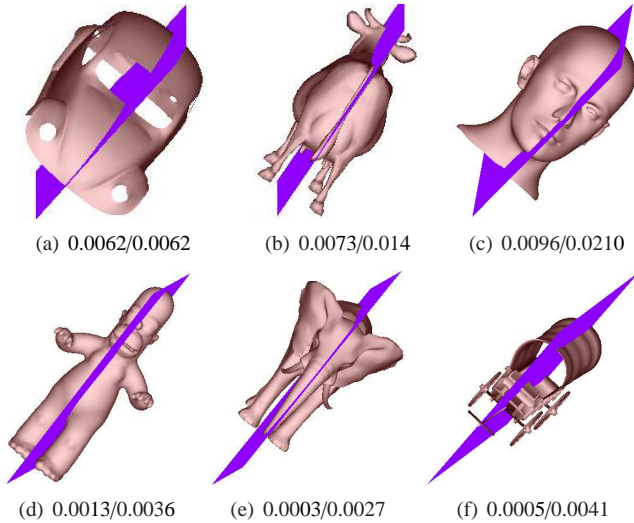


Figure 4: Example symmetry detection results with mean/max error mea-
 sures [44].

Table 1: Timing information (in seconds) comparison of our methods and other
 two state-of-the-art approaches: Mean shift [12] and 3D Hough [13] are based
 on a Pentium M 1.7 GHz CPU according to [13]; 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

308 To measure the accuracy of the detected symmetry planes,
 309 we adopt the mean (normalized by the surface area) and max-

imum (w.r.t the bounding box diagonal) distance errors devel-
 310 oped in Metro [44] which is based on surface sampling and
 311 point-to-surface distance computation. Table 2 compares the
 312 mean and max errors of the four models in Table 1 (see Fig. 4
 313 for the errors of other models) with the Mean shift [12] and
 314 3D Hough transform [13] based methods. The errors are com-
 315 puted based on the original mesh and its reflected 3D model
 316 w.r.t the detected symmetry plane. As can be seen, our approach
 317 achieves much (4~6 times w.r.t 3D Hough transform and 11~44
 318 times w.r.t Mean shift) better overall accuracy (see the mean er-
 319 rors), in spite that a few points may not be the most accurate but
 320 they still maintain a moderate accuracy (indicated by the max
 321 errors).

322 In addition, it is also very convenient to detect different de-
 323 grees of symmetries via control of the entropy difference thresh-
 324 old δ . As shown in Fig. 4, there is a minor asymmetry on the
 325 tail part of the cow, while other parts are symmetric. If we want
 326 to obtain strict symmetry, a smaller threshold δ (e.g. by reduc-
 327 ing it by half: 0.0075) will give the result that it is asymmetric.
 328 We also find that our approach can simultaneously detect mul-
 329 tiple symmetry planes for certain types of meshes, such as the
 330 Eight, Skyscraper, Bottle, Cup, Desk Lamp, and Sword in [43]
 331 and [42], as shown in Fig. 5. But we need to mention due to
 332 the limitation of CPCA and the sensitivity property to minor
 333 changes of the viewpoint entropy feature, there are a few fail
 334 cases or certain cases where the proposed method can only par-
 335 tially determine a set of reflection planes. Examples of such
 336 models are non-uniform cubes, butterflies, tori, and pears, as
 337 demonstrated in Fig. 6: (a) because of non-uniform triangula-
 338 tion, the cube model cannot be perfectly aligned with CPCA,
 339 resulting in the unsuccessful symmetry plane detection. How-
 340 ever, we have found that for most symmetric models (e.g. Mug,
 341 NonWheelChair, and WheelChair classes) that cannot be per-
 342 fectly aligned with CPCA [8], our approach can still success-
 343 fully detect their symmetry planes (e.g. the detection rates of
 344 Algorithm 1 for those types of models mentioned above are as
 345 follow: Mug: 7/8, NonWheelChair: 18/19, and WheelChair:
 346 6/7). Three examples can be found in Fig. 7; (b) the symmetry
 347 plane of the butterfly cannot be detected if based on the default
 348 threshold $\delta=0.015$, and only after increasing it till 0.0166 we
 349 can detect the plane; (c) only the red symmetry plane of the
 350 torus is detected based on the default threshold value, while
 351 both the red and green planes will be detected if we increase
 352 the threshold δ to 0.02 and all the three symmetry planes can
 353 be detected if we further increase it till 0.0215; (d) a false posi-
 354 tive (blue) symmetry plane of the pear model will appear under
 355 the condition of the default threshold, however the error will be
 356 corrected with a little smaller threshold of 0.0133. An adaptive
 357 strategy of threshold selection is among our next work plan.

358 Finally, we evaluate the overall performance of our view-
 359 point entropy distribution-based symmetry detection algorithm
 360 based on the NIST benchmark [42]. In total, we have de-
 361 tected 647 symmetry planes for all the 800 models (some of
 362 them are asymmetric). To know the general performance of
 363 our algorithm, we manually observe the symmetry property of
 364 each of the first 200/300/400 models and label its symmetry
 365 plane(s)/degree(s) to form the ground truth. Then, we exam-

Table 2: Mean/max errors [44] 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 [12]	N.A.	N.A.	N.A.	N.A.	0.059	0.018	0.111	0.037
3D Hough [13]	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

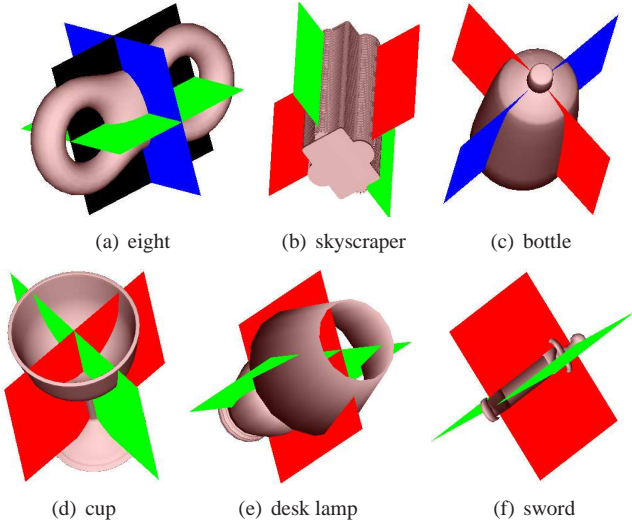


Figure 5: Multiple detected symmetry planes examples.

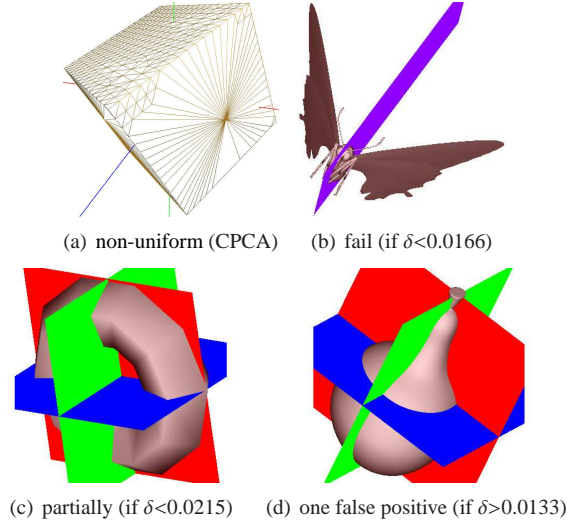


Figure 6: Failed or partially failed examples.

367 ine each detected symmetry plane to see whether it is a True
368 Positive (TP) or False Positive (FP). Similarly, we set the True
369 Negative (TN) value of a model to be 1 if it is asymmetric
370 and our algorithm also does not detect any symmetry plane.
371 While, if a symmetry plane of a symmetric model is not de-
372 tected, we increase its False Negative (FN) by 1. Table 3
373 gives the evaluation results (177/277/386 detected symmetry
374 planes) on the first 200/300/400 models (having 191/278/388
375 symmetry planes in total), which are uniformly divided into
376 10/15/20 classes. Here, for later analysis we successively list
377 the names of the 20 classes: Bird, Fish, NonFlyingInsect, Flyin-
378 gInsect, Biped, Quadruped, ApartmentHouse, Skyscraper, Sing-
379 leHouse, Bottle, Cup, Glasses, HandGun, SubmachineGun,
380 MusicalInstrument, Mug, FloorLamp, DeskLamp, Sword, and
381 Cellphone.

Table 3: Overall symmetry detection performance of our algorithm based on the first 200/300/400 models of the NIST benchmark.

# models	TP	FP	TN	FN
200	141	36	37	32
300	216	61	60	45
400	292	94	77	77

382 Based on the TP, FP, TN and FN values, we compute the
383 following nine detection evaluation metrics [45], as listed in
384 Table 4: Tracker Detection Rate (TRDR, $\frac{TP}{TG}$), False Alarm

385 Rate (FAR, $\frac{FP}{TP+FP}$), Detection Rate (DR, $\frac{TP}{TP+FN}$), Speci-
386 ficity (SP, $\frac{TN}{FP+TN}$), Accuracy (AC, $\frac{TP+TN}{TF}$), Positive Prediction
387 (PP, $\frac{TP}{TP+FP}$), Negative Prediction (NP, $\frac{TN}{FN+TN}$), False Nega-
388 tive Rate (FNR or Miss Rate, $\frac{FN}{FN+TP}$), and False Positive Rate
389 (FPR, $\frac{FP}{FP+TN}$), where the total number of symmetry planes
390 in the 200/300/400 Ground Truth models $TG=191/278/388$
391 and the total number of our detections (including both
392 trues and falses) $TF=TP+FP+TN+FN=246/382/540$. As can
393 be seen, besides the better accuracy in the detected sym-
394 metry planes as mentioned before, our detection perfor-
395 mance (e.g., for the first 200/300/400 models, Detection
396 Rate $DR=81.50\%/82.76\%/79.13\%$, and Tracker Detection Rate
397 $TRDR=73.82\%/77.70\%/75.26\%$) is also good enough. What's
398 more, the minor difference among the detection perfor-
399 mance of our algorithm on the 200, 300 and 400 models shows that
400 the overall performance of our algorithm is stable and robust in
401 terms of model type diversity and number of models evaluated.

402 In a word, as demonstrated by all the above evaluation re-
403 sults, better accuracy and efficiency than state-of-the-art ap-
404 proaches have been achieved by our simple but effective sym-
405 metry detection method. It also has good stability in dealing
406 with various model types.

407 4.2. Evaluation w.r.t to Robustness

408 *Robustness to View Sampling.* First, we also test our algorithm
409 with different levels of subdivided icosahedron for the view

Table 4: Overall symmetry detection accuracy of our algorithm based on the first 200/300/400 models of the NIST benchmark.

# models	TRDR	FAR	DR	SP	AC	PP	NP	FNR	FPR
200	73.82%	20.34%	81.50%	50.68%	72.36%	79.66%	53.62%	18.50%	49.32%
300	77.70%	22.02%	82.76%	49.59%	72.25%	77.98%	57.14%	17.24%	50.41%
400	75.26%	24.35%	79.13%	45.03%	68.33%	75.65%	50.00%	20.87%	54.97%

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 also 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 [46] 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)	

Robustness to Noise. Finally, we want to test the versatility as well as sensitivity of our algorithm when processing a modified

version of a symmetric model by adding a certain amount of noise. Due to certain factors such as creation, storage, transmission, and modification, 3D models can be noisy. A symmetry detection algorithm should be robust, thus still applicable in the case of small amounts of noise. We test the robustness of our symmetry detection algorithm against noise by randomly adding a small amount of displacement to the vertices of a 3D model.

Figure 8 demonstrates the detected symmetry planes of three example models. Table 7 shows a comparative evaluation on the detection results w.r.t the mean/max errors and the minimum entropy difference threshold value, denoted by $\min \delta$, for a successful detection of the symmetry plane(s) of a model. The results show that our algorithm has a good robustness property against a small amount of noise: by choosing different levels of entropy difference threshold values δ , we will have different tolerant levels of noise to detect symmetry planes. That is, a symmetry detection will be possible if we choose a bigger threshold if there exists a bigger amount of noise. This is contributed to our utilization of the accurate viewpoint entropy feature with a threshold for the feature paring process, since in general viewpoint entropy is stable under small changes in the vertices' coordinates of a 3D model.

4.3. Evaluation w.r.t Different 3D Alignment Algorithms

Considering the apparent advantages of the Minimum Projection Area (MPA)-based 3D alignment algorithm in finding the ideal canonical coordinate frame of a model, besides CPCA, we also evaluate the performance of a variation of our algorithm by only replacing the CPCA algorithm module with MPA. However, we found that the results are not as stable as those of the original CPCA-based version in terms of the percentage of either true or false positives based on the same threshold (δ). Choosing the threshold is also more difficult and sensitive when employing MPA since bigger threshold usually results in more false positives.

An initial analysis based on the experimental results is as follows. Due to the viewpoint sampling precision in MPA, especially for the search of the second principle axis of a 3D model which is based on a step of 1 degree, the axes found by MPA is not precise enough for this viewpoint entropy-based symmetry detection purpose, though for the 3D model retrieval application, as mentioned in the paper, the accuracy is enough. However, since our algorithm directly uses the cameras' locations to compute the symmetry plane(s) by just utilizing their correspondence relationships, it requires that the 3D model is as accurately as possible aligned w.r.t the three standard axes in order to reduce the search space and the number of viewpoints to achieve better efficiency.

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

Table 7: Comparison of the mean/max errors and the minimum entropy difference threshold values $\min \delta$ of our algorithm for successful symmetry detections of the variations of three example models after we add different levels of noise.

Noise level (%)	Beetle			Homer			Mannequin		
	mean	max	$\min \delta$	mean	max	$\min \delta$	mean	max	$\min \delta$
0.0	0.006	0.006	0.003	0.001	0.004	0.002	0.010	0.021	0.012
0.1	0.010	0.010	0.003	0.004	0.006	0.002	0.010	0.022	0.011
0.5	0.019	0.022	0.008	0.005	0.011	0.003	0.012	0.022	0.009
1.0	0.010	0.022	0.013	0.008	0.019	0.007	0.012	0.026	0.012

484 What’s more, to align a 3D model, MPA usually takes around
485 30 seconds if based on 40 Particle Swarm Optimization (PSO)
486 iterations while CPCA needs less than 1 second, which demon-
487 strates another advantage of CPCA over MPA. In addition, we
488 also have found that if based on CPCA, using bounding sphere
489 for the 3D normalization can achieve more accurate results than
490 the case of using bounding box. This should be due to the
491 fact that we also sample the viewpoints on the same bound-
492 ing sphere. However, if based on MPA, either using bounding
493 sphere or bounding box has only trivial influence on the sym-
494 metry detection performance. The reason is that the accuracy
495 of the found axes has much more direct and decisive influence
496 on the symmetry detection performance. In conclusion, using
497 CPCA is more stable, accurate and efficient than MPA, but we
498 believe an improved MPA algorithm should be more promising
499 in thoroughly solving existing errors in CPCA and achieving
500 even better results, which is among our future work.

501 5. Applications

502 Finally, we also explore two interesting applications of our
503 symmetry detection algorithm: 3D model alignment and best
504 view selection.

505 5.1. 3D Model Alignment

506 As we know, the main shortcoming of PCA-based approach
507 is that the directions of the largest extent found based on the
508 purely numerical PCA analysis are not necessarily parallel to
509 the axes of the ideal canonical coordinate frame of a 3D model.
510 This is because during the alignment process it lacks semantic
511 analysis in a 3D model’s symmetry [2] [15], or its stability [47]
512 after the alignment.

513 Based on the detected symmetry planes and the basic idea of
514 PCA, it is straightforward to apply our algorithm to 3D align-
515 ment: the first principal axis gives the maximum symmetry de-

516 gree (that is, it has the smallest total matching cost in terms
517 of viewpoint entropy for the symmetric viewpoint pairs w.r.t
518 the axis) and the second principal axis is both perpendicular
519 to the first axis and also has the maximum symmetry degree
520 among all the possible locations within the perpendicular plane.
521 Finally, we assign the orientations of each axis. This align-
522 ment algorithm is promising to achieve similar results as those
523 in [15] which is based on a planar-reflective symmetry trans-
524 form, while outperforms either PCA or CPCA for certain mod-
525 els with symmetry plane(s). However, our algorithm has better
526 efficiency than [15], thus will be more promising for related
527 real-time applications including 3D model retrieval.

528 Now we present some experimental results of the above
529 alignment algorithm. As mentioned in Section 2, there are
530 four reflection symmetry types: cyclic, dihedral, rotational,
531 and unique. In fact, some of our previous experiments already
532 demonstrate the main alignment results of several models which
533 fall into three of the above four types. For instance, Fig. 5
534 shows the two/three principal planes (thus axes) of six models
535 that have a cyclic reflection symmetry (see (c) bottle, (d) cup,
536 and (e) desk lamp), or dihedral reflection symmetry (see (a)
537 eight, (b) skyscraper, and (f) sword). Fig. 4 and Fig. 7 demon-
538 strate the first principle planes/axes of several example models
539 with a unique symmetry based on our idea. It is a trivial task
540 to continue to find other principle axes. For completeness, for
541 example, in Fig. 9, we demonstrate the complete alignment re-
542 sults of three models that have a rotational symmetry, or do
543 not have any reflection symmetry (zero symmetry), or have an
544 approximate symmetry. In a word, the alignment algorithm is
545 promising to be used in dealing with diverse types of models
546 with different degrees of symmetries.

547 5.2. Best View Selection

548 Here, we provide another option to define and search for the
549 best view of a 3D model based on our algorithm. Our definition

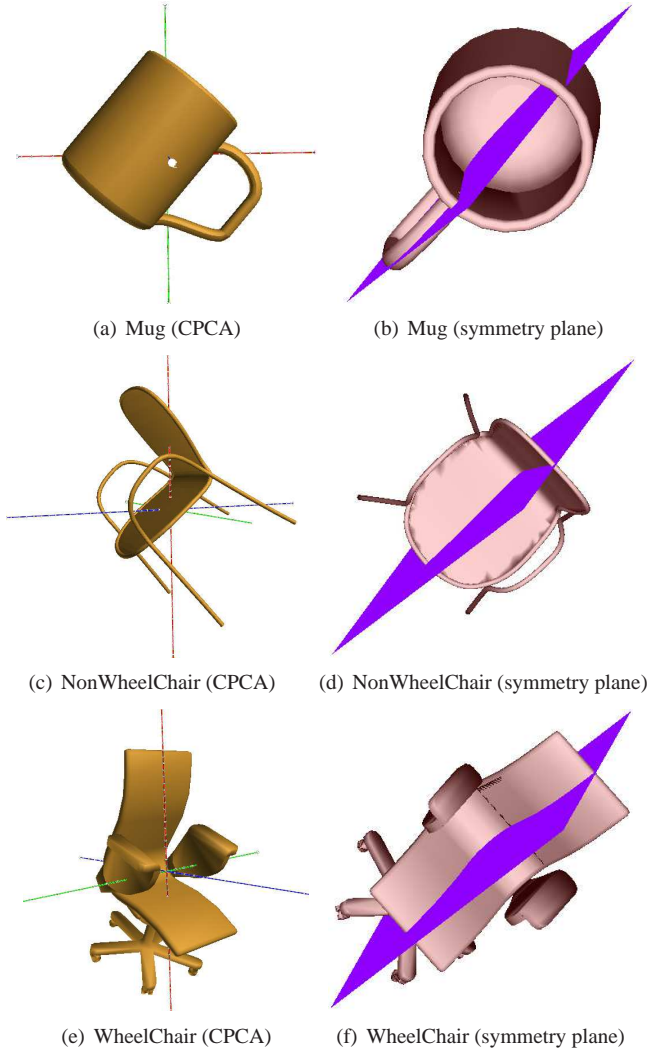


Figure 7: Examples to demonstrate that our algorithm can successfully detect the symmetry planes for most symmetric models that are not perfectly aligned with CPCA: first column shows the CPCA alignment results; second column demonstrates the detected symmetry planes.

of symmetry is related to viewpoint entropy which indicates the amount of information that a view contains. In an analogy to 3D model alignment, we use the total viewpoint entropy matching cost, that is an indicator of asymmetry, to indicate the goodness of a candidate best view corresponding to a viewpoint: the bigger the summed matching cost is, the better (more asymmetry) the viewpoint is, since it indicates that there is less redundant information in the view. When we compute the viewpoint matching cost of a candidate view, we only consider visible viewpoints as seen from the candidate view, for instance, within 180 degrees. Algorithm 1 targets finding the minimum viewpoint matching cost in terms of entropy, while we now want to find the viewpoint that gives a maximum viewpoint entropy matching cost. Thus, we develop our algorithm for this application by modifying Algorithm 1, including changing the “>”s or “≥”s to their inverses and setting a bigger threshold δ (e.g., 0.2 in our experiments). The complete best view selection algorithm is given in Algorithm 2. Fig. 10 demonstrates several promising

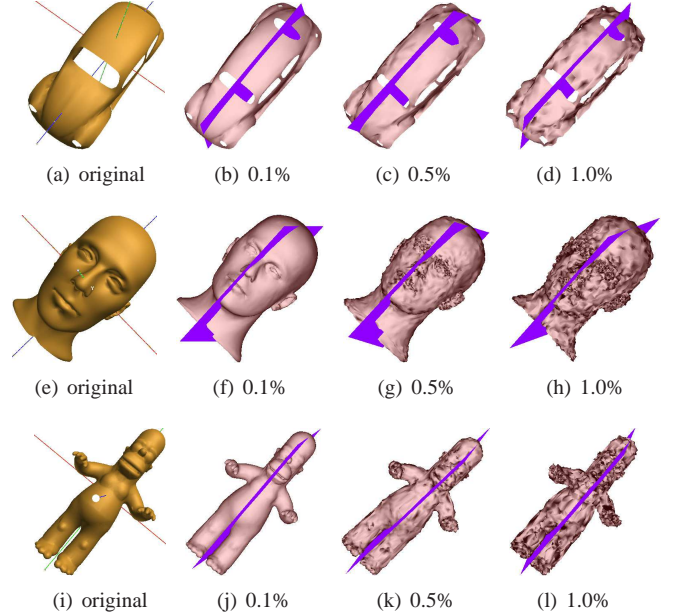


Figure 8: Examples indicating our algorithm’s robustness to noise: symmetry detection results of our algorithm in dealing with model variations with different levels of noise. The first column: original 3D models without adding any noise; The second to the fourth columns: detection results of the models when we add noise by randomly moving each vertex with a small displacement vector whose norm is bounded by 0.1%, 0.5%, and 1% of the diameter of each model’s bounding box, respectively.

informative example results based on the Algorithm 2 (using L_1 for view sampling).

6. 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 400 models and the promising results of two related applications also validate 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 of 3D models. One promising approach to achieve further improvement in terms of alignment accuracy is an improved version of the Minimum Projection Area (MPA) [39] alignment method. We can improve its accuracy to meet our precision requirement by applying the PSO-based method used in the first principle axis search in the search for the second principle axis as well. We are also 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

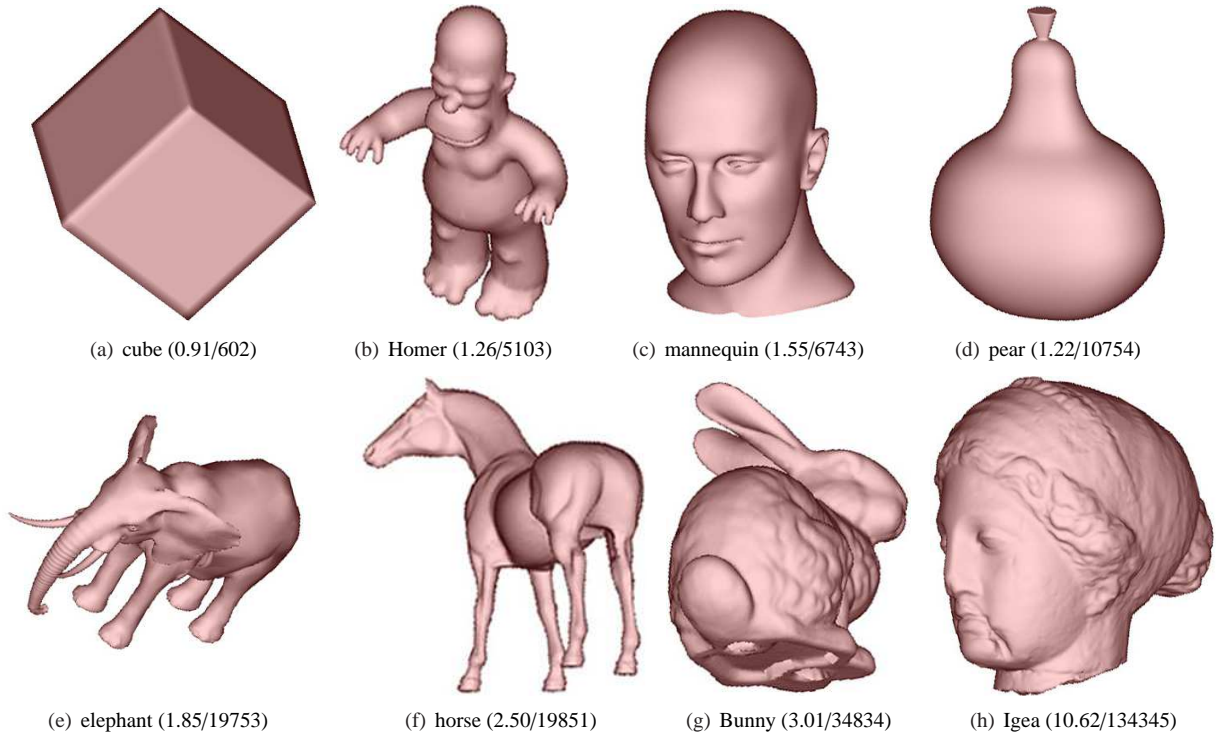


Figure 10: Best views (in terms of asymmetry property) of eight example models. The two numbers in each parenthesis are the running time (in seconds) for the model based on a computer with an Intel(R) Xeon(R) X5675 @ 3.07GHz CPU and the number of vertices the model has.

595 performance, especially for decreasing the percentage of False
 596 Negative (FN) since more symmetry planes can be successfully
 597 detected, thus avoiding the fail case like Fig. 6 (a) because of
 598 the limitation of CPCA.

599 Secondly, to further improve the efficiency of our algorithm,
 600 we could consider Hough transform for symmetry evidence vot-
 601 ing. For example, each pair of matched viewpoints casts a vote
 602 for their bisecting plane, while the peaks of the voting distri-
 603 bution correspond to prominent symmetry planes. We need to
 604 mention that directly applying Hough voting may not work be-
 605 cause rather like geometric values, symmetric viewpoints do
 606 not perfectly match each other based on their viewpoint entropy
 607 values, which has been explained in Section 3.3.

608 Finally, an automatic and adaptive strategy to select an ap-
 609 propriate threshold δ for respective models or classes is another
 610 interesting research direction and deserves our further explo-
 611 ration.

612 Acknowledgments

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

617 Henry Johan is supported by Fraunhofer IDM@NTU, which
 618 is funded by the National Research Foundation (NRF) and man-
 619 aged through the multi-agency Interactive & Digital Media Pro-
 620 gramme Office (IDMPO) hosted by the Media Development
 621 Authority of Singapore (MDA).

622 References

- 623 [1] Y. Liu, H. Hel-Or, C. S. Kaplan, L. J. V. Gool, Computational symmetry
 624 in computer vision and computer graphics, *Foundations and Trends in*
 625 *Computer Graphics and Vision* 5 (2010) 1–195.
 626 [2] M. Chaouch, A. Verroust-Blondet, Alignment of 3D models, *Graphical*
 627 *Models* 71 (2009) 63–76.
 628 [3] M. M. Kazhdan, T. A. Funkhouser, S. Rusinkiewicz, Symmetry descrip-
 629 tors and 3D shape matching, in: J.-D. Boissonnat, P. Alliez (Eds.), *Symp.*
 630 *on Geom. Process.*, volume 71 of *ACM International Conference Pro-*
 631 *ceeding Series*, Eurographics Association, 2004, pp. 115–23.
 632 [4] J. Podolak, A. Golovinskiy, S. Rusinkiewicz, Symmetry-enhanced
 633 remeshing of surfaces, in: A. G. Belyaev, M. Garland (Eds.), *Symp.*
 634 *on Geom. Process.*, volume 257 of *ACM International Conference Pro-*
 635 *ceeding Series*, Eurographics Association, 2007, pp. 235–42.
 636 [5] P. D. Simari, D. Nowrouzezahrai, E. Kalogerakis, K. Singh, Multi-
 637 objective shape segmentation and labeling, *Comput. Graph. Forum* 28
 638 (2009) 1415–25.
 639 [6] K. Sfikas, T. Theoharis, I. Pratikakis, Rosy+: 3D object pose normaliza-
 640 tion based on PCA and reflective object symmetry with application in 3D
 641 object retrieval, *Int. J. Comput. Vis.* 91 (2011) 262–79.
 642 [7] P.-P. Vázquez, M. Feixas, M. Sbert, W. Heidrich, Viewpoint selection
 643 using viewpoint entropy, in: T. Ertl, B. Girod, H. Niemann, H.-P. Seidel
 644 (Eds.), *VMV, Aka GmbH*, 2001, pp. 273–80.
 645 [8] D. Vranic, 3D Model Retrieval, Ph.D. thesis, University of Leipzig, 2004.
 646 [9] B. Li, H. Johan, Y. Ye, Y. Lu, Efficient view-based 3D reflection sym-
 647 metry detection, in: *SIGGRAPH Asia 2014 Creative Shape Modeling*
 648 *and Design*, Shenzhen, China, December 03 - 06, 2014, 2014, p. 2.
 649 URL: <http://doi.acm.org/10.1145/2669043.2669045>. doi:10.1
 650 145/2669043.2669045.
 651 [10] T. Sawada, Z. Pizlo, Detecting mirror-symmetry of a volumetric shape
 652 from its single 2D image, in: *CVPR Workshops*, CVPRW '08, 2008, pp.
 653 1–8. doi:10.1109/CVPRW.2008.4562976.
 654 [11] N. J. Mitra, M. Pauly, M. Wand, D. Ceylan, Symmetry in 3D geome-
 655 try: Extraction and applications, in: *EUROGRAPHICS State-of-the-art*
 656 *Report*, 2012.

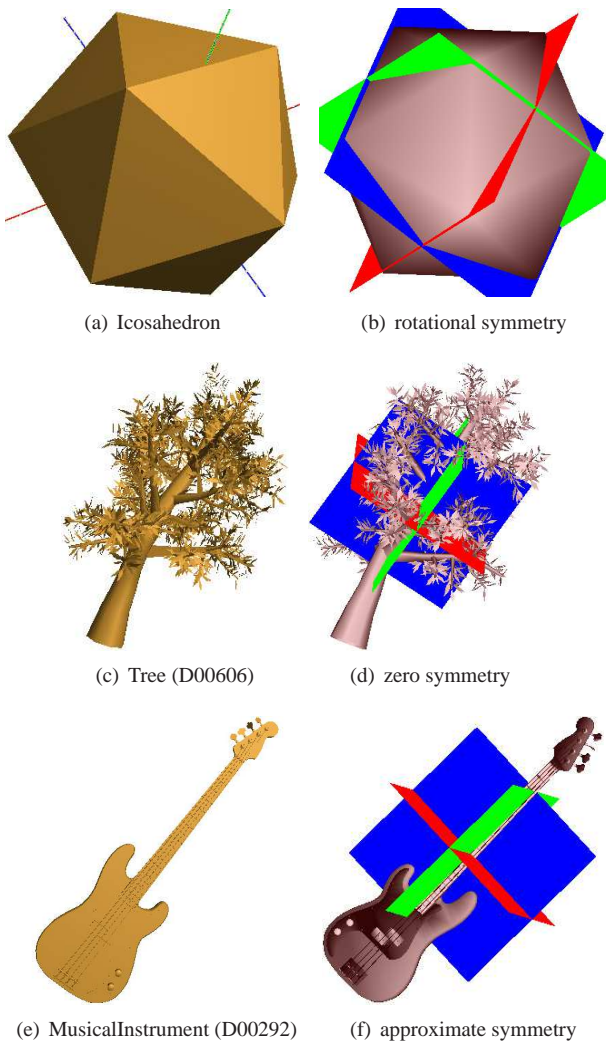


Figure 9: Alignment results of Icosahedron and two other example models from the NIST dataset. Icosahedron has 15 symmetry planes, Tree type model D00606.off has no symmetry plane, while MusicalInstrument type model D00292.off has a roughly symmetry plane.

657 [12] N. J. Mitra, L. J. Guibas, M. Pauly, Partial and approximate symmetry
658 detection for 3D geometry, *ACM Trans. Graph.* 25 (2006) 560–8.
659 [13] D. Cailliere, F. Denis, D. Pele, A. Baskurt, 3D mirror symmetry detection
660 using hough transform, in: *ICIP, IEEE*, 2008, pp. 1772–5.
661 [14] H. Zabrodsky, S. Peleg, D. Avnir, Symmetry as a continuous feature,
662 *IEEE Trans. Pattern Anal. Mach. Intell.* 17 (1995) 1154–66.
663 [15] J. Podolak, P. Shilane, A. Golovinskiy, S. Rusinkiewicz, T. A.
664 Funkhouser, A planar-reflective symmetry transform for 3D shapes, *ACM*
665 *Trans. Graph.* 25 (2006) 549–59.
666 [16] M. Bokeloh, A. Berner, M. Wand, H.-P. Seidel, A. Schilling, Symmetry
667 detection using feature lines, *Comput. Graph. Forum* 28 (2009) 697–706.
668 [17] T. Sawada, Visual detection of symmetry of 3D shapes, *Journal of Vision*
669 10 (2010) 4:1–4:22.
670 [18] H. L. Zou, Y. T. Lee, Skewed rotational symmetry detection from a
671 2D line drawing of a 3D polyhedral object, *Computer-Aided Design* 38
672 (2006) 1224–32.
673 [19] H. L. Zou, Y. T. Lee, Skewed mirror symmetry detection from a 2D
674 sketch of a 3D model, in: S. N. Spencer (Ed.), *GRAPHITE, ACM*, 2005,
675 pp. 69–76.
676 [20] A. Martinet, C. Soler, N. Holzschuch, F. X. Sillion, Accurate detection of
677 symmetries in 3D shapes, *ACM Trans. Graph.* 25 (2006) 439–64.
678 [21] K. Xu, H. Zhang, A. Tagliasacchi, L. Liu, G. Li, M. Meng, Y. Xiong,
679 Partial intrinsic reflectional symmetry of 3D shapes, *ACM Trans. Graph.*

680 28 (2009).
681 [22] K. Xu, H. Zhang, W. Jiang, R. Dyer, Z. Cheng, L. Liu, B. Chen, Multi-
682 scale partial intrinsic symmetry detection, *ACM Transactions on Graphics*
683 (Special Issue of SIGGRAPH Asia) 31 (2012) 181:1–181:10.
684 [23] J. Liu, Y. Liu, Curved reflection symmetry detection with self-validation,
685 in: R. Kimmel, R. Klette, A. Sugimoto (Eds.), *ACCV* (4), volume 6495
686 of *Lecture Notes in Computer Science*, Springer, 2010, pp. 102–14.
687 [24] D. Raviv, A. M. Bronstein, M. M. Bronstein, R. Kimmel, Full and partial
688 symmetries of non-rigid shapes, *International Journal of Computer Vision*
689 89 (2010) 18–39.
690 [25] C. Li, A. B. Hamza, Symmetry discovery and retrieval of nonrigid 3D
691 shapes using geodesic skeleton paths, *Multimedia Tools Appl.* 72 (2014)
692 1027–47.
693 [26] Y. Wang, K. Xu, J. Li, H. Zhang, A. Shamir, L. Liu, Z.-Q. Cheng,
694 Y. Xiong, Symmetry hierarchy of man-made objects, *Comput. Graph.*
695 *Forum* 30 (2011) 287–96.
696 [27] V. G. Kim, Y. Lipman, X. Chen, T. A. Funkhouser, Möbius transforma-
697 tions for global intrinsic symmetry analysis, *Comput. Graph. Forum* 29
698 (2010) 1689–700.
699 [28] Y. Lipman, T. A. Funkhouser, Möbius voting for surface correspondence,
700 *ACM Trans. Graph.* 28 (2009).
701 [29] H. Wang, P. Simari, Z. Su, H. Zhang, Spectral global intrinsic symmetry
702 invariant functions, *Proc. of Graphics Interface* (2014).
703 [30] J. Sun, M. Ovsjanikov, L. J. Guibas, A concise and provably informative
704 multi-scale signature based on heat diffusion, *Comput. Graph. Forum* 28
705 (2009) 1383–92.
706 [31] M. Aubry, U. Schlickewei, D. Cremers, The wave kernel signature: A
707 quantum mechanical approach to shape analysis, in: *ICCV Workshops*,
708 *IEEE*, 2011, pp. 1626–33.
709 [32] A. V. Tuzikov, O. Colliot, I. Bloch, Evaluation of the symmetry plane in
710 3D MR brain images, *Pattern Recogn. Lett.* 24 (2003) 2219–33.
711 [33] J. Tedjokusumo, W. K. Leow, Normalization and alignment of 3D ob-
712 jects based on bilateral symmetry planes, in: T.-J. Cham, J. Cai, C. Dorai,
713 D. Rajan, T.-S. Chua, L.-T. Chia (Eds.), *MMM* (1), volume 4351 of *Lec-
714 ture Notes in Computer Science*, Springer, 2007, pp. 74–85.
715 [34] A. Golovinskiy, J. Podolak, T. A. Funkhouser, Symmetry-aware mesh
716 processing, in: E. R. Hancock, R. R. Martin, M. A. Sabin (Eds.), *IMA
717 Conference on the Mathematics of Surfaces*, volume 5654 of *Lecture
718 Notes in Computer Science*, Springer, 2009, pp. 170–88.
719 [35] A. Berner, M. Wand, N. J. Mitra, D. Mewes, H.-P. Seidel, Shape analysis
720 with subspace symmetries, *Comput. Graph. Forum* 30 (2011) 277–86.
721 [36] A. Tagliasacchi, H. Zhang, D. Cohen-Or, Curve skeleton extraction from
722 incomplete point cloud, *ACM Transactions on Graphics* (Special Issue of
723 SIGGRAPH) 28 (2009) Article 71, 9 pages.
724 [37] J. Cao, A. Tagliasacchi, M. Olson, H. Zhang, Z. Su, Point cloud skele-
725 tons via laplacian-based contraction, in: *Proc. of IEEE Conf. on Shape
726 Modeling and Applications*, 2010, pp. 187–97.
727 [38] I. Jolliffe, *Principal Component Analysis* (2nd edition), Springer, Heidel-
728 berg, 2002.
729 [39] H. Johan, B. Li, Y. Wei, Iskandarsyah, 3D model alignment based on
730 minimum projection area, *The Visual Computer* 27 (2011) 565–74.
731 [40] S. Takahashi, I. Fujishiro, Y. Takeshima, T. Nishita, A feature-driven
732 approach to locating optimal viewpoints for volume visualization, in:
733 *IEEE Visualization*, IEEE Computer Society, 2005, pp. 495–502.
734 [41] C. Loop, *Smooth Subdivision Surfaces Based on Triangles*, Master’s the-
735 sis, University of Utah, 1987.
736 [42] R. Fang, A. Godil, X. Li, A. Wagan, A new shape benchmark for 3D
737 object retrieval, in: G. Bebis, et al. (Eds.), *ISVC* (1), volume 5358 of
738 *Lecture Notes in Computer Science*, Springer, 2008, pp. 381–92.
739 [43] AIM@SHAPE, <http://shapes.aimatshape.net/>, 2014.
740 [44] P. Cignoni, C. Rocchini, R. Scopigno, Metro: Measuring error on simpli-
741 fied surfaces, *Comput. Graph. Forum* 17 (1998) 167–74.
742 [45] V. Manohar, P. Soundararajan, H. Raju, D. B. Goldgof, R. Kasturi, J. S.
743 Garofolo, Performance evaluation of object detection and tracking in
744 video, in: P. J. Narayanan, S. K. Nayar, H.-Y. Shum (Eds.), *ACCV* (2),
745 volume 3852 of *Lecture Notes in Computer Science*, Springer, 2006, pp.
746 151–61.
747 [46] MeshLab, <http://meshlab.sourceforge.net/>, 2014.
748 [47] H. Fu, D. Cohen-Or, G. Dror, A. Sheffer, Upright orientation of man-
749 made objects, *ACM Trans. Graph.* 27 (2008).

Algorithm 2: Best view selection based on maximum viewpoint entropy matching cost.

Input : N : number of viewpoints;
 $Pos[N]$: positions of N viewpoints;
 $E[N]$: entropy values of N viewpoints;
 n : icosahedron subdivision level;
 $\delta=0.2$: entropy difference threshold;
 $\epsilon=1e-5$: small difference in double values

Output: Symmetry planes' equations, if applicable

begin

```

// initialize maximum viewpoint entropy
  matching cost
max_cost ← 0.0;
// loop viewpoint pairs (u, v)
for u ← 0 to N - 1 do
  P_u ← Pos[u];
  for v ← 0 to N - 1 do
    if u == v then
      continue;
    P_v ← Pos[v], T_1 ← normalize(P_u - P_v);
    // initialize the viewpoint entropy
    matching cost for the current
    view
    cur_cost ← 0;
    // matching other viewpoint 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;
        P_j ← Pos[j], P_m ←  $\frac{P_i + P_j}{2}$ ;
        T_2 = normalize(P_i - P_j);
        CT = T_1 × T_2, DT = T_1 · T_2;
        if |DT| < 0 then
          continue;
        if ||CT|| > ε AND |DT| ≠ 0 then
          continue;
        if |T_1 · P_m| > ε then
          continue;
        cur_cost = cur_cost + |E[i] - E[j]|;
        break;
      if cur_cost > max_cost then
        max_cost = cur_cost;
        T ← T_1;
// output the best view
T[0] * x + T[1] * y + T[2] * z = 0

```
