Normalization and Alignment of 3D Objects Based on Bilateral Symmetry Planes

Jefry Tedjokusumo and Wee Kheng Leow

Dept. of Computer Science, National University of Singapore,
3 Science Drive 2, Singapore 117543
{jefryted, leowwk}@comp.nus.edu.sg
www.comp.nus.edu.sg/∼leowwk

Abstract. Recent advancements in 3D scanning technologies have inspired the development of effective methods for matching and retrieving 3D objects. A common pre-processing stage of these retrieval methods is to normalize the position, size, and orientation of the objects based on PCA. It aligns an object's orientation based on PCA eigenvectors, and normalizes its size uniformly in all 3 spatial dimensions based on the variance of the object points. However, orientation alignment by PCA is not robust, and objects with similar shape can be misaligned. Uniform scaling of the objects is not ideal because it does not take into account the differences in the objects' 3D aspect ratios, resulting in misalignment that can exaggerate the shape difference between the objects. This paper presents a method for computing 3D objects' bilateral symmetry planes (BSPs) and BSP axes and extents, and a method for normalizing 3D objects based on BSP axes and extents. Compared to normalization methods based on PCA and minimum volume bounding box, our BSP-based method can normalize and align similar objects in the same category in a semantically more meaningful manner, such as aligning the objects' heads, bodies, legs, etc.

1 Introduction

Recent advancements in 3D scanning technologies have led to an increased accumulation of 3D models in databases and the Internet, and inspired the development of effective techniques for retrieving 3D objects that are similar in shape to a query model (e.g., [1,2,3,4,5,6]). 3D object matching and retrieval typically involve three basic stages: (1) object normalization, (2) feature extraction and object representation, and (3) object comparison. The first stage typically normalizes objects' positions, sizes, and orientations by translating the objects' centroids to the origin of the 3D coordinate frame, normalizing the variances of the points on the objects, and aligning their principal axes obtained using Principal Component Analysis (PCA) [1,7]. The second stage extracts various features from the objects and represents the objects in various forms such as histograms, 2D spherical maps, 3D grids, and abstract representations in terms of the extracted features [7]. The third stage typically uses very simple distance measures such as the Euclidean distance to perform efficient comparison.

T.-J. Cham et al. (Eds.): MMM 2007, LNCS 4351, Part I, pp. 74–85, 2007.

The standard normalization method described above is not ideal. Orientation alignment based on PCA is not robust because PCA is sensitive to point distributions of the objects. Objects with similar shape may be misaligned [1] (Fig. 1). Moreover, this method does not take into account the difference in the objects' 3D aspect ratios. Normalization of objects with different 3D aspect ratios by the same scaling factors in all 3 spatial dimensions causes misalignments of their corresponding parts (Figs. 2, 7(a)). All these misalignments can result in an exaggeration of the difference between objects with similar shapes. Consequently, relevant objects (i.e., objects in the same category as the query) may be regarded by the matching algorithm as different from the query and are not retrieved. Therefore, it is important to normalize and align the objects properly.

A straightforward improvement over the standard normalization method is to scale the objects according to their 3D aspect ratios. This brings out a question: In which coordinate system should the objects' 3D aspect ratios be measured? A possibility is to measure 3D aspect ratios along the PCA axes. This method is not robust because, as discussed above, orientation alignment based on PCA is not robust. An alternative method is to compute the objects' minimum volume bounding boxes (MBB) [8], and normalize the objects based on MBB axes and widths. Our studies show that this method is even less robust than the PCA method, as will be discussed further in Sections 2 and 4.

It is observed that many natural and man-made objects exhibit bilateral (i.e., left-right) symmetry. It is a kind of reflectional symmetry that has an interesting semantic meaning: the bilateral symmetry plane (BSP) divides an object into a left and a right half, each is a mirror reflection of the other about the BSP. Moreover, the major axis that defines the object's top and bottom lies in the BSP. Therefore, by normalizing objects according to the principal axes and 3D aspect ratios defined on BSP, the objects' semantically corresponding parts such as head, body, legs can be aligned. Consequently, shape matching of objects aligned in this manner would be semantically more meaningful.

Note that PCA or MBB alone is insufficient for computing an object's BSP. The PCA and MBB planes (i.e., the planes normal to the PCA/MBB axes) may not be aligned to the BSP plane in terms of position and 3D orientation (Figs. 1 and 2). Furthermore, an object has three PCA planes and three MBB planes. Using only PCA and MBB algorithms, it is impossible to determine which of the three planes is nearest to the object's BSP. For objects that are not exactly bilaterally symmetric, the best fitting BSP may not pass through the objects' centroid. So, to determine an object's BSP, the algorithm needs to compute the correct 3D orientation and position of a plane that separates the object into two bilaterally symmetric parts.

This paper presents a method for (1) computing 3D Objects' BSPs and BSP axes and extents, and (2) normalizing and aligning 3D objects based on BSP axes and extents. Test results show that the algorithm can compute the exact BSPs of exactly bilaterally symmetric objects. For objects that are roughly bilaterally symmetric, the algorithm can compute the best fitting BSPs. Normalization of objects according to BSPs yields better normalization and alignment between 3D objects in the same category compared to those using PCA and MBB.

2 Related Work

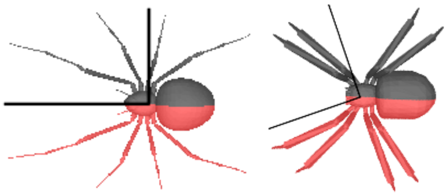
There is a substantial amount of work on 3D symmetry detection. Alt el al. [9] described algorithms for computing exact and approximate congruences and symmetries of geometric objects represented by point sets. Wolter et al. [10] presented exact algorithms for detecting all rotational and involutional symmetries in point sets, polygons and polyhedra. Jiang et al. [11,12] presented methods for determining rotational and involutional symmetries of polyhedra. Brass and Knauer [13,14] further developed methods for computing and testing symmetries of non-convex polyhedra and general 3D objects. Zabrodsky et al. [15] defined a Continuous Symmetry Measure to quantify the symmetries of objects. Minovic et al. [16] described an algorithm for identifying symmetry of 3D objects represented by octrees.

Sun and Sherrah [17] proposed algorithms for determining reflectional and rotational symmetries of 3D objects using orientation histograms. To reduce the search space, their algorithms search for the symmetries of an object along its principal axes and small orientation neighborhoods around them. The principal axes are obtained from a method similar to PCA. Our studies show that this approach is not robust because the reflectional symmetry plane of an object can be quite far from the PCA planes normal to the PCA axes (Section 4).

The above research work has focused on symmetry detection or quantification. On the other hand, Jiang and Bunke [18] applied symmetry detection in polyhedra for object recognition. Kazhdan et al. developed methods of matching 3D shape using reflectively symmetric feature descriptors [19] and rotationally symmetric descriptors [20].

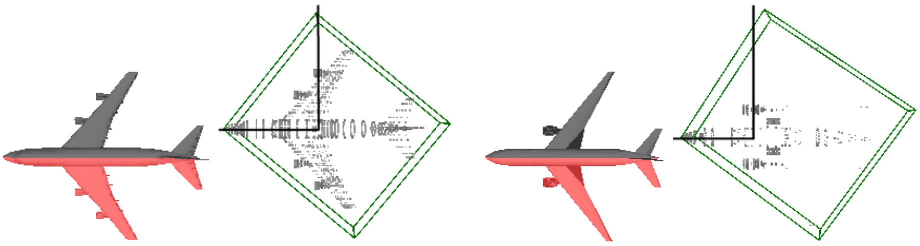
In this paper, we focus on determining bilateral symmetry planes (BSPs), BSP axes, and 3D aspect ratios for more robust and semantically meaningful normalization and alignment of 3D objects. The objects are represented as point-and-mesh models, and the object points need not be uniformly distributed over their surfaces. Indeed, many of our test objects are composed of highly non-uniformly distributed points.

Principal Component Analysis (PCA) is a well-known method for computing the principal axes of an object and the spread of points along the axes. It is the standard method for normalizing 3D objects' orientation. However, the principal axes obtained by PCA are sensitive to the distributions of points on the objects. Differences in point distributions between two objects of similar shape can cause their orientations to be misaligned [1]. This problem is most serious for objects that are not exactly bilaterally symmetric (Fig. 1). Moreover, the variances of the points along the PCA axes (i.e., the eigenvalues) are sensitive to non-uniformity of point distributions. Two objects with the same extents but different point distributions can have different variances (Fig. 2) As a result, the eigenvalues cannot be used as good estimates of the object's 3D aspect ratios.



(a) (b)

Fig. 1. PCA misalignment. Spider *b* is not exactly bilaterally symmetric, causing its PCA axes (black lines) to be misaligned with those of spider *a*. However, their computed BSPs are well aligned. Left and right sides of BSPs are denoted by different colors.



(a) (b) (c) (d)

Fig. 2. MBB misalignment. Airplanes *a* and *c* have different 3D aspect ratios. (b, d) Their first PCA axes (with the largest eigenvalues, horizontal black lines) are aligned with the BSPs but their MBBs (green boxes) are not.

The minimum volume bounding box (MBB) algorithm developed in [8] is less sensitive to the overall distribution of the points on the objects but is very sensitive to the positions of the points furthest from the objects' centroid. Typically, the objects' MBBs are not aligned with their BSPs (Fig. 2). However, it can compute the extents of the objects even if the point distributions are not uniform. So, MBB widths can be good estimates of the objects' 3D aspect ratios if MBB axes are aligned with the BSPs.

3 Bilateral Symmetry Plane

3.1 Computing BSP

For objects that are rotationally symmetric, such as a ball and an orange, each of the multiple rotational symmetry planes is a bilateral symmetry plane (BSP). However, for most natural and man-made objects with bilateral symmetry, they have only one BSP each (Figs. 1, 2, 5). To compute an object's BSP, we use the fact that each point on the object's surface has a mirror reflection with respect to the BSP.

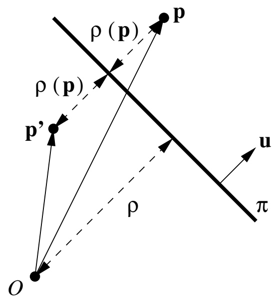


Fig. 3. Object point $\mathbf{p}$ and its mirror reflection $\mathbf{p}'$ with respect to the plane π

A plane π in 3D space can be parameterized by the equation

$$\mathbf{w} \cdot \mathbf{x} + w_0 = 0 \tag{1}$$

where w_0 and $\mathbf{w} = (w_1, w_2, w_3)$ are the parameters of the plane, and $\mathbf{x}$ is any 3D point on the plane. Consider any two points $\mathbf{x}_1$ and $\mathbf{x}_2$ lying on the plane. From Eq. 1, we obtain

$$\mathbf{w} \cdot (\mathbf{x}_2 - \mathbf{x}_1) = 0 \tag{2}$$

which means that $\mathbf{w}$ is normal to the plane. Thus, the plane's unit normal vector $\mathbf{u}$ is given by $\mathbf{w}/\|\mathbf{w}\|$. The perpendicular distance ρ of the plane from the origin is given by $\mathbf{u} \cdot \mathbf{x}$ for any point $\mathbf{x}$ on the plane. That is, $\rho = -w_0/\|\mathbf{w}\|$.

Now, consider a point $\mathbf{p}$ on the object's surface. From Fig. 3, it is obvious that the perpendicular distance of $\mathbf{p}$ from a plane π, denoted as $\rho(\mathbf{p})$, is

$$\rho(\mathbf{p}) = \mathbf{p} \cdot \mathbf{u} - \rho \, . \tag{3}$$

Then, the ideal mirror reflection $\mathbf{p}'$ of $\mathbf{p}$ with respect to the plane π is (Fig. 3):

$$\mathbf{p}' = \mathbf{p} - 2\,\rho(\mathbf{p})\,\mathbf{u} \, . \tag{4}$$

In practice, a 3D object is typically represented as a point-and-mesh model, which consists of a sparse set S of points on the 3D object's surface. So, for a point $\mathbf{p}_i \in S$, its ideal mirror reflection $\mathbf{p}'_i$ may not be in S. Let f denote the closest-point function and $f(\mathbf{p}'_i)$ denote a point in S closest to $\mathbf{p}'_i$. That is, $f(\mathbf{p}'_i)$ is the closest approximation to $\mathbf{p}'_i$. Then, the mean-squared error E between all $\mathbf{p}'_i \in S$ and its closest approximation $f(\mathbf{p}'_i)$ is:

$$E(\boldsymbol{\theta}) = \sum_{\mathbf{p}_i \in S} \| \, \mathbf{p}'_i - f(\mathbf{p}'_i)\|^2 = \sum_{\mathbf{p}_i \in S} \| \, \mathbf{p}_i - 2\,\rho(\mathbf{p}_i)\,\mathbf{u} - f(\mathbf{p}'_i)\|^2 \tag{5}$$

where the vector $\boldsymbol{\theta} = (w_0, w_1, w_2, \, w_3)$. Therefore, the problem of computing the bilateral symmetry plane is to find the plane π, parameterized by $\boldsymbol{\theta}$, that minimizes the error $E(\boldsymbol{\theta})$.

The algorithm for computing an object's BSP can be summarized as follows:

Compute BSP

1. Compute the three PCA axes (i.e., eigenvectors) of the object, and the three PCA planes normal to these axes.
2. Set the PCA plane with the smallest error E as the seed plane.
3. Rotate the seed plane in all three rotation angles by increments of δ to generate initial planes ω_j.
4. For each initial plane ω_j,
   perform gradient descent to obtain locally optimal BSP π_j and its error E_j.
5. Return the plane π_k with the smallest error E_k.

Given a sufficiently small δ, the above algorithm can find the globally optimal estimate of the object's BSP. In the tests, $\delta = 22.5°$ is used. This algorithm can also use MBB axes to obtain the seed plane. However, our tests show that initializing with PCA is more robust than initializing with MBB because the objects' BSPs tend to be closer to PCA planes than MBB planes (Section 4).

3.2 BSP-Based Object Normalization

Orientation alignment based on BSP offers an approach that can take into account the semantics of the object parts, such as head, body, legs, etc. We define the first BSP axis as the vector in BSP with the largest dispersion of points. This definition is analogous to that of PCA axis. The second BSP axis is the vector in BSP perpendicular to the first BSP axis. The third BSP axis is naturally the vector normal to BSP.

Based on the above definition, we can compute the BSP axes as follows:

Compute BSP Axes and Extents

1. Project 3D points on an object to its BSP.
2. Apply 2D PCA on the projected points and obtain principal axes in BSP.
3. Measure the extents (i.e., the distances between the furthest points) of the object along the two principal axes in BSP. The axis with a larger extent is defined as the first BSP axis, and the other one is the second BSP axis.
4. BSP's normal vector is defined as the third BSP axis. The extent along this axis is also computed.
5. The extents along the three BSP axes define the object's 3D aspect ratio.

In the third step, PCA eigenvalues should not be used as measures of the object's extents because they are sensitive to non-uniform distribution of points.

Similar to PCA axes, the BSP axes for different objects may be pointing in opposite directions even though their orientations are the same. A common technique of handling this problem is to reflect the principal axes before matching the objects [4]. With three principal axes, there are altogether eight reflected versions to be compared. The reflection with the smallest matching error would be the one with the semantically matching axis directions.

BSP-based normalization is performed by translating the objects centroids' to the origin of the 3D coordinate frame, aligning the objects' BSP axes, and normalizing their 3D aspect ratios to a standardized 3D aspect ratio according to their BSP extents. In case this method distorts the shapes of some objects too significantly, an alternative is to group objects into categories according to some criterion such as difference in aspect ratios, semantic class, etc., and scale the objects in each category to a different standardized 3D aspect ratio that minimizes shape distortion.

4 Experiments

The test set contains 1602 objects some of which are exactly bilaterally symmetric while the others are roughly bilaterally symmetric. This test set is compiled by combining 512 aircrafts in the Utrecht database [21] and 1090 objects in the Princeton database [22]. The Utrecht database contains 6 categories of aircrafts whereas the Princeton database contains about 50 categories of objects. Highly non-bilaterally symmetric objects in the Princeton database are excluded.

Two sets of tests were conducted to assess the performance of the algorithm for computing BSPs and BSP-based object normalization and alignment. The implementation of the MBB algorithm was downloaded from the web site valis.cs.uiuc.edu/˜sariel/research/papers/98/bbox.html.

4.1 Test on BSP Computation

For this test, the following normalized error E' was computed for the estimated BSP $\boldsymbol{\theta}$ of each object S:

$$E'(\boldsymbol{\theta}) = \frac{1}{|S|\, v} \sum_{\mathbf{p}_i \in S} \| \mathbf{p}'_i - f(\mathbf{p}'_i) \| \tag{6}$$

where v is the variance of the points $\mathbf{p}_i$ from the object's centroid. This normalized error is independent of the number of points and the size of the objects, and so can be compared among the objects.

The algorithm for computing BSP was performed on all 1602 objects. It successfully computed the BSPs of 1589 (99.2%) objects. Among the successful cases, the computed BSPs of 487 (30.7%) bilaterally symmetric objects are practically exact, with $E' \leq 0.00001$ (Fig. 4). A total of 1348 objects (84.8%) with bilateral symmetry and approximate bilateral symmetry have errors $E' \leq 0.03$. For the other 241 (15.2%) successful cases, the computed BSPs have various amounts of error ranging from 0.03 to greater than 0.1. Sample results are shown in Fig. 5. For objects that are bilaterally symmetric (rows 1, 2), exact BSPs are found. For objects that are roughly bilaterally symmetric (row 3), the best fitting BSPs are computed. Therefore, the error E' is well correlated to the degree of bilateral symmetry of the test objects.

For the 13 (0.8%) failure cases (Fig. 5, row 4), all their errors are greater than 0.03, and 53.9% of them are greater than 0.1. The computed BSPs are all larger

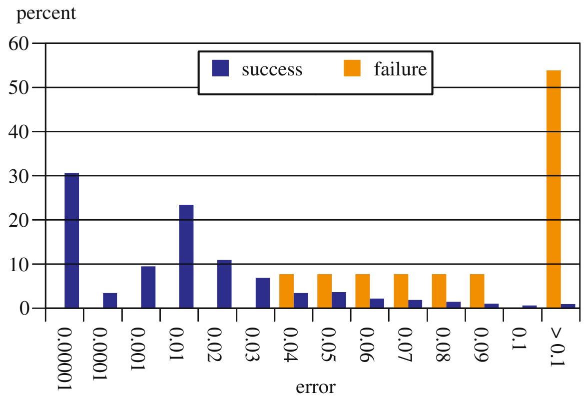


Fig. 4. Frequency distribution of the errors of computed BSPs of test objects

Table 1. Percentages of PCA and MBB planes nearest to test objects' BSPs

	1st	2nd	3rd
PCA	13.4%	61.7%	19.9%
MBB	31.1%	46.1%	22.8%

than 30° from the desired BSPs. The main reason of the failure is that these objects are not exactly bilaterally symmetric and there are very few points on them. In some cases, one or two outliers (i.e., points without mirror reflections and lying at large distances from the objects' centroids) are enough to severely tilt the orientation of the computed BSP. One method of solving this problem is to apply a robust method to exclude outliers while computing the BSP.

As discussed in Section 1, an object has three PCA planes and three MBB planes. Table 1 tabulates the percentage of PCA/MBB planes that are nearest, in terms of 3D orientation, to the computed BSPs of the test objects. It shows that most of the objects' BSPs are nearest to the second PCA plane (the plane normal to the second PCA axis). This is expected because most objects' second PCA axes run across their bodies in the left-right direction normal to their BSPs. Nevertheless, there are many other objects whose BSPs are nearest to other PCA/MBB planes. These results show that PCA and MBB, by themselves, are not able to determine the correct BSPs in general.

Figure 6 plots the frequency distribution of the angular difference between an object's BSP and its nearest PCA/MBB plane. 69.6% of the objects have BSPs exactly aligned with their nearest PCA planes (i.e., 0° difference). On the other hand, only 1.2% of the objects have BSPs exactly aligned with their nearest MBB planes. Most (30.7%) of the objects' BSPs are, in fact, more than 20° off the nearest MBB planes. This test result shows that it is better to use PCA planes to initialize the algorithm for computing BSP (Section 3.1).

Fig. 5. Sample BSP results. (Rows 1–3) Successful cases: (Rows 1, 2) Bilaterally symmetric objects, (Row 3) Approximately bilaterally symmetric objects. (Row 4) Failure cases. Left and right sides of BSPs are denoted by different colors.

4.2 Test on BSP-Based Normalization

Four types of normalization methods were compared:

1. PCA with uniform scaling (PCA):
   Normalize objects' centroids, PCA axes, and variance of points. This is the standard normalization method and serves as the base case.
2. PCA with 3D aspect ratio normalization (PCA3):
   Normalize objects' centroids, PCA axes, and 3D aspect ratio estimated by PCA eigenvalues.
3. MBB:
   Normalize MBB centroid, MBB axes, and MBB extents.
4. BSP:
   Normalize object's centroid, BSP axes, and BSP extents.

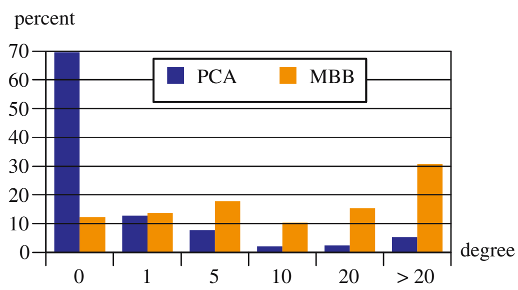


Fig. 6. Frequency distribution of the angular difference between objects' BSPs and their nearest PCA/MBB planes

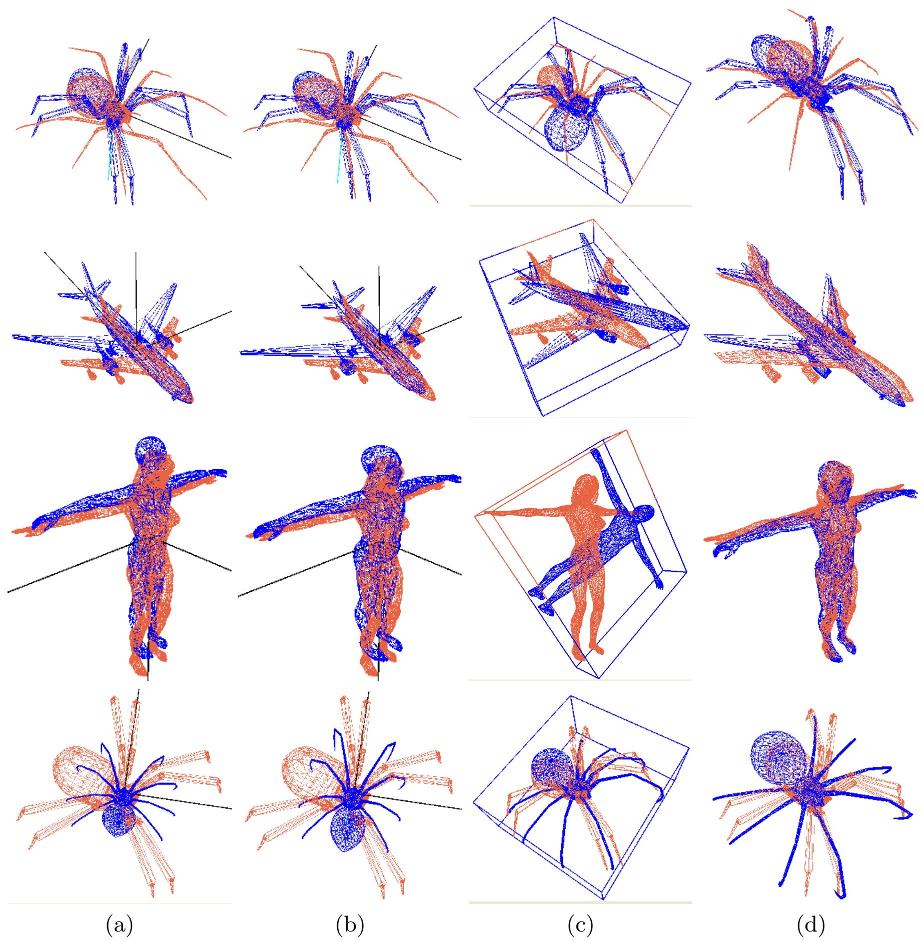


Fig. 7. Comparison of normalization methods. (a) PCA with uniform scaling, (b) PCA with normalization of 3D aspect ratio, (c) MBB, (d) BSP.

Figure 7 illustrates the difference between the various normalization methods. In many cases, both PCA and PCA3 can align the objects' principal axes well (Fig. 7(a, b), rows 1–3). But, sometimes, they give the wrong orientation alignment (Fig. 7(a, b), row 4). They are unable to normalize the 3D aspect ratios well enough to achieve semantically meaningful alignment of the object parts due to PCA's sensitivity to point distributions (Fig. 7(a, b), rows 2–4).

MBB can normalize the 3D aspect ratios relatively well. But, a slight difference in the lengths and widths of the objects can cause the orientation alignment to be off by as much as 90° (Fig. 7(c), rows 1–3). On the other hand, our BSP-based method consistently normalizes and aligns the objects well (Fig. 7(d)). In particular, semantically equivalent parts, such as heads, bodies, legs, wings, and tails, of different objects are correctly aligned.

5 Conclusions

This paper presented a method for computing 3D objects' bilateral symmetry planes (BSPs) and BSP axes and extents, and a method for normalizing and aligning 3D objects based on BSP axes and extents. The algorithm successfully computed the BSPs of 99.2% of the test objects. For exactly bilaterally symmetric objects, the exact BSPs are found. For roughly bilaterally symmetric objects, the best fitting BSPs are computed. Compared with normalization methods based on PCA and minimum volume bounding box, our method based on BSP can normalize and align similar objects in the same category in a semantically meaningful manner, such as aligning the objects' heads, bodies, legs, etc. Better normalization and alignment of objects are expected to improve the performance of shape matching and retrieval algorithms of 3D objects.

References

1. Kazhdan, M., Funkhouser, T., Rusinkiewicz, S.: Rotation invariant spherical harmonic representation of 3D shape descriptors. In: Proc. Eurographics Symp. on Geometry Proc. (2003)
2. Osada, R., Funkhouser, T., Chazelle, B., Dobkin, D.: Matching 3D models with shape distribution. In: Proc. SMI. (2001)
3. Paquet, E., Rioux, M., Murching, A., Naveen, T., Tabatabai, A.: Description of shape information for 2-D and 3-D objects. Signal Processing: Image Communication **16** (2000) 103–122
4. Tangelder, J.W.H., Veltkamp, R.C.: Polyhedral model retrieval using weighted point sets. In: Proc. SMI. (2003)
5. Vranic, D.V., Saupe, D.: 3D shape descriptor based on 3D Fourier transform. In: Proc. Conf. Digital Signal Proc. Multimedia Comm. and Services. (2001) 271–274
6. Yu, M., Atmosukarto, I., Leow, W.K., Huang, Z., Xu, R.: 3D model retrieval with morphing-based geometric and topological feature maps. In: Proc. IEEE CVPR. (2003) II–656–II–661
7. Atmosukarto, I., Leow, W.K., Huang, Z.: Feature combination and relevance feedback for 3D model retrieval. In: Proc. MMM. (2005) 334–339

8. Barequet, G., Har-Peled, S.: Efficiently approximating the minimum-volume bounding box of a point set in three dimensions. J. Algorithms **38** (2001) 91–109
9. Alt, H., Mehlhorn, K., Wagener, H., Welzl, E.: Congruence, similarity and symmetries of geometric objects. Discrete Computational Geometry (1988) 237–256
10. Wolter, J.D., Woo, T.C., Volz, R.A.: Optimal algorithms for symmetry detection in two and three dimensions. Visual Computer **1** (1985) 37–48
11. Jiang, X.Y., Bunke, H.: A simple and efficient algorithm for determining the symmetries of polyhedra. CVGIP: Graphical Models & Image Proc. **54** (1992) 91–95
12. Jiang, X., Yu, K., Bunke, H.: Detection of rotational and involutional symmetries and congruity of polyhedra. Visual Computer **12** (1996) 193–201
13. Brass, P., Knauer, C.: Computing the symmetries of non-convex polyhedral objects in 3-space. In: Proc. European Workshop on Comp. Geometry. (2002)
14. Brass, P., Knauer, C.: Testing congruence and symmetry for general 3-dimensional objects. Comp. Geometry: Theory and Applications **27** (2004) 3–11
15. Zabrodsky, H., Peleg, S., Avnir, D.: Symmetry as a continuous feature. IEEE Trans. PAMI **17** (1995) 1154–1165
16. Minovic, P., Ishikawa, S., Kato, K.: Symmetry identification of a 3d object represented by octree. IEEE Trans. PAMI **15** (1993) 507–154
17. Sun, C., Sherrah, J.: 3D symmetry detection using the extended gaussian image. IEEE Trans. PAMI **19** (1997) 164–169
18. Jiang, X.Y., Bunke, H.: Determination of the symmetries of polyhedra and an application to object recognition. In: Proc. Comp. Geometry: Methods, Algorithms and Applications (LNCS 553). (1991) 113–121
19. Kazhdan, M., Chazelle, B., Dobkin, D., Finkelstein, A., Funkhouser, T.: A reflective symmetry descriptor. In: Proc. ECCV. (2002) 642–656
20. Kazhdan, M., Funkhouser, T., Rusinkiewicz, S.: Symmetry descriptors and 3D shape matching. In: Proc. Eurographics Symp. on Geometry Proc. (2004) 115–123
21. :(Utrecht Univeristy Object Database, www.cs.uu.nl/centers/give/imaging/3drecog/3dmatching.html)
22. : (Princeton University Object Database, shape.cs.princeton.edu/benchmark)