Vis Comput
DOI 10.1007/s00371-011-0590-y

ORIGINAL ARTICLE

3D model alignment based on minimum projection area

Henry Johan · Bo Li · Yuanmin Wei · Iskandarsyah

Abstract 3D model alignment is an important step for applications such as 3D model retrieval and 3D model recognition. In this paper, we propose a novel Minimum Projection Area-based (MPA) alignment method for pose normalization. Our method finds three principal axes to align a model: the first principal axis gives the minimum projection area when we perform an orthographic projection of the model in the direction parallel to this axis, the second axis is perpendicular to the first axis and gives the minimum projection area, and the third axis is the cross product of the first two axes. We devise an optimization method based on Particle Swarm Optimization to efficiently find the axis with minimum projection area. For application in retrieval, we further perform axis ordering and orientation in order to align similar models in similar poses. We have tested MPA on several standard databases which include rigid/non-rigid and open/watertight models. Experimental results demonstrate that MPA has a good performance in finding alignment axes which are parallel to the ideal canonical coordinate frame of models and aligning similar models in similar poses under different conditions such as model variations, noise, and initial poses. In addition, it achieves a better 3D model retrieval performance than several commonly used approaches such as CPCA, NPCA, and PCA.

Keywords 3D model alignment · Minimum projection area · 3D model retrieval

H. Johan (✉) · B. Li · Y. Wei · Iskandarsyah
School of Computer Engineering,
Nanyang Technological University, Singapore, Singapore
e-mail: henryjohan@ntu.edu.sg

Published online: 20 April 2011

1 Introduction

3D models are created in arbitrary scale, orientation, and position in 3D space. Therefore, pose normalization of 3D models is important in many computer graphics applications such as 3D model retrieval, 3D model recognition, and 3D visualization. The goal of 3D model pose normalization is to transform a model into a canonical coordinate frame, where the representation of the model is independent of its scale, orientation, and position. An ideal canonical coordinate frame of a 3D model is defined as a coordinate frame whose axes are parallel to the front-back, left-right and top-bottom directions of the model. The normalization process includes two steps: alignment and scaling. The important and difficult step is 3D model alignment, and the traditional method to deal with this is Principal Component Analysis (PCA) [7]. To improve the accuracy, various alignment algorithms based on the idea of PCA have been proposed, such as Continuous PCA (consider the area of each face) [23] and Normal PCA (consider the normal of each face) [16]. Other approaches utilize symmetry information [1], virtual contact area (VCA) [15], or projection area [9, 12].

The existing alignment algorithms, however, still have room for improvement in terms of the performance in finding alignment axes which are parallel to the ideal canonical coordinate frame and 3D model retrieval. This motivates us to propose a novel 3D alignment algorithm which finds the alignment axes based on minimum projection area (MPA). Our proposed algorithm is based on the observation that many objects have a minimum projection area when we orthogonally project them in the direction parallel to one of the axes of the ideal canonical coordinate frame. Based on experimental results, we find our MPA algorithm can align most 3D models in terms of axes accuracy (the axes are parallel to the ideal canonical coordinate frame). Our alignment

algorithm can also align similar models in similar poses, which is important for 3D model retrieval. It is also robust with respect to model variations, noise, and initial poses.

The rest of this paper is organized as follows. In Sect. 2, we review the related work in 3D model alignment and viewpoint selection. In Sect. 3, we present the details of our MPA alignment algorithm. Extensive experimental results are shown in Sect. 4. Section 5 discusses the conclusions and future work.

2 Related work

In this section, we review the related techniques in 3D model alignment and viewpoint selection.

2.1 3D model alignment

Nowadays, there are several approaches to align a 3D model. Here, we review four different approaches.

PCA-based approach Principal Component Analysis (PCA) [7] and Continuous Principal Component Analysis (CPCA) [23] are two commonly used alignment algorithms. They utilize the statistical information of vertex coordinates and extract the three orthogonal components with largest extent to depict the principal axes of a 3D model. An extension of the idea of CPCA is Normal Principal Component Analysis (NPCA) [13], which applies CPCA to the normals of the surface points of a 3D model. The shortcoming of the PCA-based approach is that the directions of the largest extent are not necessarily parallel to the axes of the ideal canonical coordinate frame of 3D models. CPCA is generally regarded as a more stable PCA-based method. However, Papadakis et al. [13] found that for some models (e.g. car, shovel, hammer, and plotted plant) CPCA outperforms NPCA, but for some other models (like plane, chair, gun, and desktop computer) NPCA has a better alignment performance.

Symmetry-based approach Chaouch and Verroust-Blondet [1] proposed an approach based on the analysis of the reflection symmetry property of a 3D model, such as cyclic, dihedral, and rotation symmetries. Podolak et al. [14] developed a symmetry transform to measure the degree of symmetry of a 3D model with respect to any candidate symmetry plane. Tedjokusumo and Kheng Leow [19] developed an alignment algorithm using bilateral symmetry planes (BSPs) by considering the 3D aspect ratio of a model. They defined three BSP axes in an analogous way as PCA [7]: the first BSP axis has the largest extend in the BSP, the second is perpendicular to the first, and the third is the normal of the BSP. However, the symmetry-based approach has a limitation in dealing with models without an apparent symmetry property or non-symmetrical models.

Optimization-based approach Fu et al. [5] proposed an upright alignment algorithm for man-made models. The algorithm first computes the convex hull of a model, then finds a set of candidate bases, and finally selects the base with the largest assessment function value as the bottom of the model. The assessment function is composed of four geometrical properties: static stability, symmetry, parallelism, and visibility. Random Forest classifier and Support Vector Machine (SVM) classifier are adopted to train the function. The upright orientation algorithm achieves around 90% prediction accuracy in terms of the vertical extent of models. Martinek and Grosso [11] proposed an optimization and GPU-based approach to align two 3D models. They constructed a model function with respect to the intersection and union of the projection results of two models.

Projection area-based approach Recently, we noticed that there are two other papers which use projection area for alignment. Lian et al. [9] proposed a method that first determines two sets of candidate axes using PCA and the rectilinearity metric. Then, the final alignment axes are decided by selecting the set of candidate axes which minimizes the sum of the projected area of silhouettes. Napoléon and Sahbi [12] presented an alignment method which selects one of three alignment results (original pose, PCA, and NPCA) that gives the minimum visual hull, that is minimizes the sum of the projected areas on the three projection planes. Unlike the above two methods, our proposed method performs alignment by successively selecting two axes with minimum projection areas. Moreover, we perform a global optimization search for finding the minimum projection area, and our algorithm does not rely on the PCA-based approach.

2.2 Viewpoint selection

Our proposed MPA alignment is based on minimum projection area, so it can be considered as a view-based approach. As such, we also review some viewpoint selection techniques. The goal of viewpoint selection is to find a set of representative views to depict a 3D model. Usually, it is used to select the best views of a 3D model.

Lee et al. [8] defined the idea of mesh saliency for 3D models in terms of Gaussian-weighted mean curvatures. Viewpoint selection, one of the applications of mesh saliency, was demonstrated based on a gradient-descent search to find the candidate views with local maximums and a random search algorithm to find the global maximum. Yamauchi et al. [24] proposed a method to find a set of representative views for a 3D model by clustering the views and using mesh saliency [8] to characterize the quality of a view. Vázquez et al. [20, 21] proposed an information theory-related measurement called viewpoint entropy to depict the amount of information a view contains, and based

on this they developed a method to automatically find a set of best views with top view entropy values. All these viewpoint selection techniques can select the best views of 3D models with respect to their view quality metrics; however, in general the direction of the best views are not parallel to the axes of the ideal canonical coordinate frame of 3D models.

3 Minimum Projection Area-based (MPA) alignment

3.1 Basic idea

Based on its ideal canonical coordinate frame, every 3D model has six canonical orthographic projection views, which are front, back, left, right, top, and bottom views, as shown in Fig. 1. If we only consider the projection area (the area of the region occupied by the object in the view images in Fig. 1), then there are only three different canonical views because under orthographic projection, the front view has the same projection area as the back view, the left view has the same area as the right view, and the top view has the same area as the bottom view. We observe that, for many objects, one of their canonical views (that is, either front-back view or left-right view or top-bottom view) has a minimum projection area compared to the other arbitrary views of the objects. Figure 2 shows two such examples. In fact, we conduct experiments on several 3D model databases and verify that the above-mentioned observation is true for a large number of 3D models.

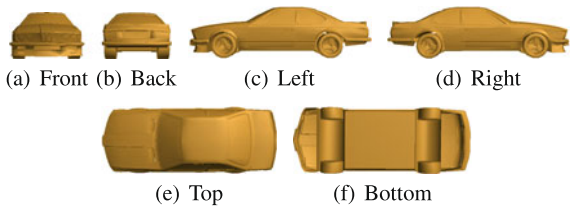


Fig. 1 Six canonical orthographic projection views of a car model based on its ideal canonical coordinate frame

Motivated by the above findings, we develop a Minimum Projection Area-based alignment algorithm (MPA). Our algorithm finds three principal axes of a 3D model which satisfy the following. The first principal axis gives the minimum projection area when we perform an orthographic projection of the model along (parallel to) this axis, the second axis is perpendicular to the first axis and gives a minimum projection area, and the third axis is the cross product of the first two axes.

3.2 MPA alignment algorithm

Given a 3D model, the set of candidate axes is generated by using a sphere. A *candidate axis* is defined as a line which connects a point on the surface of the sphere and the center of the sphere. To compute the projection area of this axis, we perform an orthographic projection of the model in the direction parallel to the axis and determine the projection area by counting the number of pixels occupied by the model in the projection image.

The steps of our MPA alignment are as follows.

Step 1: *Find the first principal axis*. We sample a set of points on the surface of the sphere, compute the candidate axes based on these points, and find the axis with minimum projection area. To find this axis, we devise an efficient search algorithm based on the Particle Swarm Optimization (PSO) [3] method (see Sect. 3.3).

Step 2: *Find the second principal axis*. We find the axis with minimum projection area by sampling on the perimeter of a circle which is perpendicular to the first principal axis. Since this is only a 1D search, we perform a brute-force search to find the second principal axis by sampling the perimeter in the range of [0°,180°) and choosing a step of 1°.

Step 3: *Compute the third principal axis*. We compute the third axis as the cross product of the first two principal axes.

For 3D model retrieval application, the following two steps are performed to align similar models in similar poses.

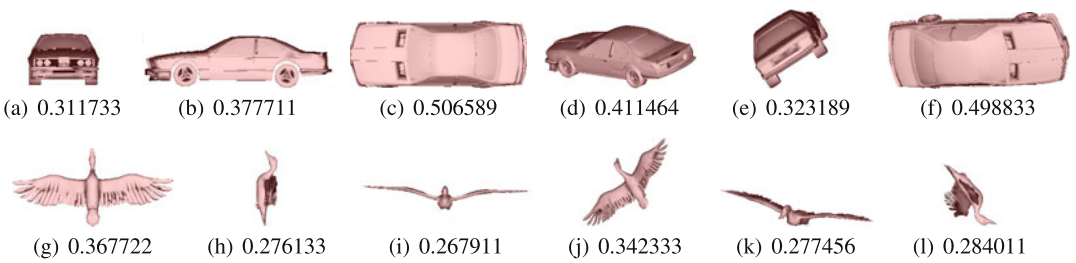
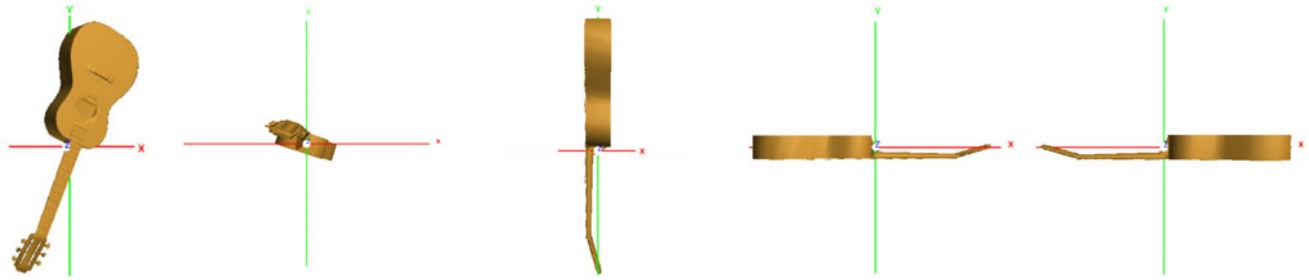


Fig. 2 Examples showing that one canonical view of a 3D model usually has the minimum projection area. In *each row*, the *first three images* are the front, left, and top views of a 3D model and the remaining three images are three arbitrary views of the same model. The number underneath each view is its normalized projection area



(a) Original pose (b) Step 1: first principle axis (c) Step 2: second principle axis (d) Step 3∼4: axis ordering (e) Step 5: axis orientation (final result)

Fig. 3 An example of the alignment process using our MPA algorithm. (**b**)–(**e**) Show the intermediate alignment results of the five steps in the algorithm, respectively

Step 4: *Axis ordering*. First, we determine the top and bottom orientations of the model by adopting the static stability metric in upright orientation [5]. We compute six static stability values of the principal axes (two for each axis, in the positive and negative directions). The direction with the largest static stability value is set as the bottom of the model (the y negative axis of the model) and the corresponding principal axis is set as the y axis. Then, we determine the x and z axes based on the variance of the remaining two principal axes. The axis with a larger variance is set as the x axis and the other as the z axis. In order to compute the variance, we employ a similar method as [23] by considering the area of each face of the model.

Step 5: *Axis orientation*. We employ the viewpoint entropy metric [21] to decide the orientations of the x and z axes. We render two views of the model from the positive and negative sides of the x axes (z axes) and select the one with a larger entropy value as the left side (front part) of the model.

Figure 3 shows the result at each step of MPA alignment for a guitar model.

3.3 PSO-based search for minimum projection area

The simplest method to find the axis with minimum projection area is by performing a brute-force search. We can uniformly sample a set of points on the surface of the sphere based on the subdivision of a regular icosahedron which is denoted as the zero level icosahedron L_0. Figure 4 shows the resulting icosahedrons at different levels of subdivision by applying the Loop subdivision rule [10] once (L_1), twice (L_2), thrice (L_3), and four times (L_4).

Figure 5 shows the distribution of projection area of two models in the NIST database [4] using the third level icosahedron L_3 for sampling the axes and mapping their projection areas as colors on the surface of the spheres. The drawback of the brute-force search is the high computational cost. Based on experimental results, we find that in order to get a result with good accuracy, we have to use at least an L_6 icosahedron (40004 sampling points). As such, the brute-force search is not the ideal method for finding the axis with minimum projection area.

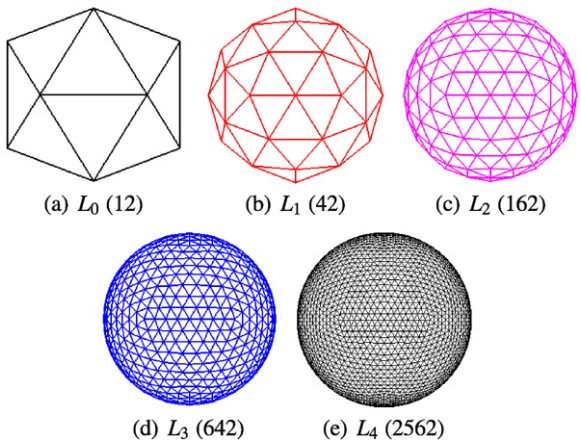
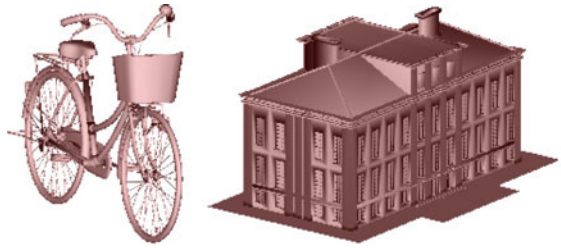


Fig. 4 Subdivision of an icosahedron. The number in each bracket is the number of sample points of the corresponding subdivided icosahedron

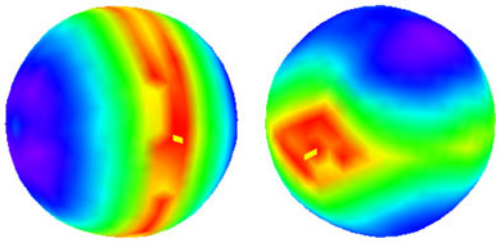
To find the axis efficiently, we develop a search method based on PSO [3] which is a global search optimization algorithm. PSO belongs to swarm intelligence optimization techniques and it imitates the random search actions of a flock of birds seeking a piece of food in a region. Each bird adopts the same strategy of searching the surrounding area of the bird that is nearest to the food, and they learn with each other and update themselves based on the obtained information. PSO has been found to be robust and fast in solving nonlinear and nondifferentiable problems [17].

The steps of our PSO-based search are as follows.

Step 1: *Initialization*. We initialize the number N_P and positions of a set of search particles and then compute the private best for each particle and current global best based on all the private bests. In practice, we use the 42 sample points in L_1 to distribute the search particles. To compute the private best of a search particle, we consider its $\lfloor N_P/3 \rfloor$



(a) A bicycle model (b) An apartment house model



(c) Area distribution of (a) (d) Area distribution of (b)

Fig. 5 Distribution of the projection area of two models. Area is coded using HSV color model and smooth shading. *Red*: small area; *green*: mid-size area; *blue*: large area. The *yellow* bar depicts the sample point with minimum area

nearest neighboring particles in terms of geodesic distance. Then, we set the value for the maximum number of search iteration N_t.

Step 2: *Update*. We compute the velocity update step s inversely proportional to the current iteration number i,

$$s = \frac{N_t - i}{N_t} + c, \tag{1}$$

where c is a constant variable. We choose c to be 0.5 in our experiment. Based on the following two equations [17], we update the new position for each particle as follows.

$$\mathbf{x}(i+1) = \mathbf{x}(i) + s \cdot \mathbf{v}(i), \tag{2}$$

$$\begin{aligned}\mathbf{v}(i+1) = {} & \omega * \mathbf{v}(i) + c_1 \cdot r_1 \cdot \big(\mathbf{x}_p(i) - \mathbf{x}(i)\big) \\ & + c_2 \cdot r_2 \cdot \big(\mathbf{x}_g(i) - \mathbf{x}(i)\big).\end{aligned} \tag{3}$$

$\mathbf{x}(i)$ and $\mathbf{v}(i)$ are the position and velocity of a particle; $\mathbf{x}_p$ and $\mathbf{x}_g$ are the positions of private and global bests. c_1 and c_2 are non-negative constant numbers, typically $c_1 = c_2 = 2$ [3]; r_1 and r_2 are random variables between 0 and 1. ω is an inertia-weight to balance the abilities of global search and local search. A larger ω means more global search power and less dependency on the initial positions of the search particles. A smaller ω corresponds to finer search in a local region. Similarly as in [17], we dynamically decrease ω from 1.4 to 0 based on an inversely proportional function with respect to the iteration number i:

$$\omega = \frac{\omega_{\min} - \omega_{\max}}{N_t} \cdot i + \omega_{\max}, \tag{4}$$

where $\omega_{\max}$ (1.4) and $\omega_{\min}$ (0) is the maximum and minimum inertia-weight values. The new position $\mathbf{x}(i+1)$ may not be located on the surface of the sphere, as such we project it to the surface of the sphere in the direction from the center to the computed $\mathbf{x}(i+1)$.

Step 3: *Evaluation*. Based on its new position, for each particle, we compute the corresponding axis, render the 3D model, compute the projection area, and update its private best. Based on all the private bests, we update the global best.

Step 4: *Verification*. If the current iteration number has exceeded N_t, we stop and output the axis which corresponds to the position of the current global best as the first principal axis; otherwise, go to *Step 2*: *Update* to continue the search.

4 Experiments and discussion

To intensively investigate the performance of our MPA alignment algorithm, we test the MPA algorithm on the following four representative standard databases:

- Princeton Shape Benchmark database (*PSB*) [18]. It contains 907 general models, classified into 92 classes.
- NIST Generic Shape Benchmark (*NIST*) [4]. This database is used to overcome several shortcomings or biases of previous benchmarks, such as different sizes of each class. It contains 800 models, classified into 40 classes, 20 models each.
- AIM@Shape Watertight Models Benchmark (*WMB*) [22]. The dataset has 400 watertight models, divided into 20 classes, 20 models each. Some are non-rigid models with different variations.
- Engineer Shape Benchmark (*ESB*) [6]. This is a CAD model database which contains 867 models, which are classified into 45 classes.

4.1 Evaluation with respect to axes accuracy

Experiments on different types of models, such as general models in PSB, CAD models in ESB, and non-rigid models in WMB, demonstrate that our MPA can align most of them accurately, robustly, and consistently. Some examples are shown in Fig. 6.

Finding three alignment axes which are parallel to the ideal canonical coordinate frame is important. Therefore, we perform axes accuracy experiments on the above-mentioned four databases and compare MPA with CPCA in terms of the percentages of the alignment results that have three axes parallel to the ideal canonical coordinate frame (allow a very small rotational difference). For a database, we calculate the average percentage over all the models as well as the percentage for each class. Table 1 compares their performances

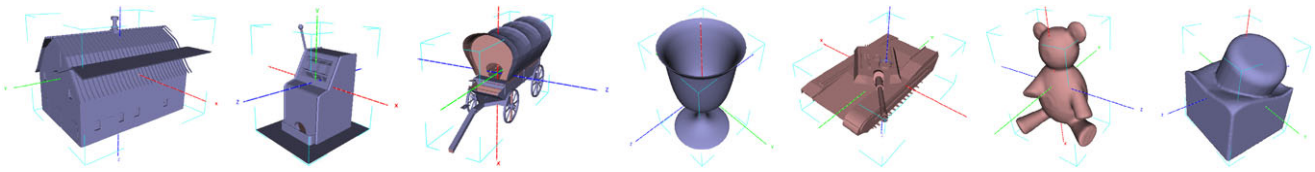
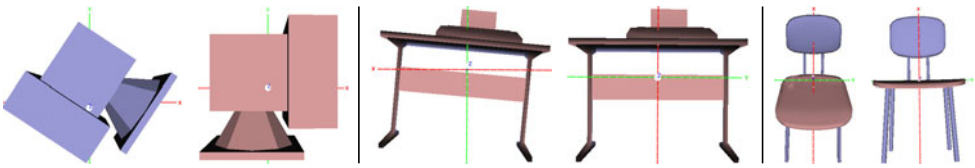


Fig. 6 Example alignment results for different types of models using our MPA alignment algorithm

Table 1 Comparison of the axes accuracy performances in terms of models and classes using MPA and CPCA on the PSB, NIST, WMB, and ESB databases. R^{CPCA} and R^{MPA} are the average performance over all the models in a database. Δ is the performance difference value of subtracting MPA's axes accuracy percentage by CPCA's for one class. $\Delta \geq 20$: MPA is much better than CPCA; $20 > \Delta > 0$: MPA is better than CPCA; $\Delta = 0$: MPA is the same as CPCA; and vice versa

Databases	#(models)	R^{CPCA}	R^{MPA}	#(classes)	$\Delta \geq 20$	$20 > \Delta > 0$	$\Delta = 0$	$0 > \Delta > -20$	$\Delta \leq -20$
PSB	907	632 (69.7%)	804 (88.6%)	92	41 (44.6%)	8 (8.7%)	29 (31.5%)	6 (6.5%)	8 (8.7%)
NIST	800	652 (81.5%)	695 (86.9%)	40	5 (12.5%)	16 (40.0%)	12 (30.0%)	6 (15.0%)	1 (2.5%)
WMB	400	270 (67.5%)	327 (81.8%)	20	6 (30.0%)	8 (40.0%)	3 (15.0%)	1 (5.0%)	2 (10%)
ESB	867	657 (76.1%)	744 (86.2%)	45	15 (33.3%)	10 (22.2%)	15 (33.3%)	4 (8.9%)	1 (2.2%)

Fig. 7 Examples indicating that our MPA algorithm achieves better alignment results than CPCA



(a) CPCA (b) MPA (c) CPCA (d) MPA (e) CPCA (f) MPA

and Table 2 lists the classes in which MPA achieves a much better performance than CPCA on the PSB database.

As shown in Table 1, our MPA approach achieves apparently better overall performance than CPCA. MPA is better than CPCA in aligning 53.3% classes for PSB, and 52.5%, 70.0%, 55.5% for NIST, WMB, and ESB, respectively. Conversely, the percentages for the cases in which CPCA outperforms MPA are much smaller (15.2%, 17.5%, 15%, and 11.1%, respectively). MPA has a much better performance (the surpassing percentage difference is more than 20) in aligning the listed 41 classes of PSB models in Table 2, especially for box-like shapes, such as desktop computer, computer monitor, school desk, and church. Figure 7 shows some examples which demonstrate that MPA can find more accurate axes than CPCA.

For certain models, MPA cannot find accurate axes, and usually there exists some small rotational differences. The reason for these differences is that a small rotation from the accurate axes will make the projection area even smaller. These types of classes include dog, desk chair, potted plant, barren tree, conical tree, handgun, and fireplace. Some alignment results for these classes are shown in Fig. 8. Nevertheless, we can see that, even if the axes found are not the perfect ones, their alignment results are still consistent among the models in the same class, which is important for applications, such as 3D model retrieval.

4.2 Evaluation with respect to robustness

In this section, we test the robustness properties of MPA with respect to model variations, noise, and initial poses as well as the convergence of PSO with respect to iteration number.

(1) *Robustness to model variations*. The basic requirement for alignment in applications such as 3D model retrieval and recognition is to align similar models in a similar way under different conditions such as variations and deformations. For this purpose, we investigate the alignment performance on non-rigid models with different variations, for example, hand, teddy, and head models in the previously mentioned four databases. Some example alignment results for these types of models in the WMB database are shown in Fig. 9. The first nine models are examples of deformable models. We can also see that the head models with different variations are aligned consistently, such as the three similar head models looking to the front and the other three similar head models looking to the left.

(2) *Robustness to noise*. 3D models may have noise due to storage, transmission, and modification. A 3D model alignment algorithm should be insensitive to small amounts of noise. We test the robustness of our MPA algorithm against noise by randomly adding a small

Table 2 List of the 41 classes of PSB in which MPA achieves a much better performance than CPCA in terms of axes accuracy percentage

Class	#	CPCA	MPA
Helicopter	18	77.8	100.0
Enterprise spaceship	11	36.4	100.0
Dog	7	00.0	57.1
Horse	6	16.7	50.0
Rabbit	4	00.0	75.0
Snake	4	25.0	75.0
Head	16	62.5	93.8
Skull	6	00.0	66.7
Barn	5	40.0	100.0
Church	4	00.0	100.0
Gazebo	4	80.0	100.0
One story building	14	35.7	100.0
Skyscraper	5	80.0	100.0
Two story building	10	10.0	100.0
Chess set	9	66.7	100.0
City	8	37.5	75.0
Desktop computer	11	00.0	100.0
Computer monitor	13	00.0	100.0
Eyeglasses	7	71.4	100.0
Fireplace	6	00.0	33.3
Cabinet	9	66.7	100.0
School desk	4	00.0	100.0
Bench seat	11	00.0	45.5
Dining chair	11	00.0	90.9
Desk chair	15	00.0	20.0
Shelves	13	76.9	100.0
Rectangular table	25	72.0	100.0
Single leg table	6	66.7	100.0
Handgun	10	00.0	40.0
Ladder	4	50.0	100.0
Streetlight lamp	8	75.0	100.0
Mailbox	7	14.3	85.7
Potted plant	26	53.8	88.5
Satellite	4	25.0	50.0
Large sail boat	6	00.0	50.0
Sink	4	25.0	100.0
Slot machine	4	25.0	100.0
Hammer	4	75.0	100.0
Covered wagon	5	00.0	100.0
Semi vehicle	7	14.3	100.0
Train car	5	40.0	100.0

amount of displacement to the vertices of a 3D model. Figure 10 shows that MPA has a good robustness property against a small amount of noise. This is contributed to our utilization of projection area for aligning a 3D model since in general projection area is stable under small changes of the vertices' coordinates.

(3) *Robustness to initial poses*. 3D models may have arbitrary initial poses. It is important for our alignment algorithm to align a model with different initial poses to the same pose. Figure 11 illustrates three sets of examples indicating MPA's robustness to initial poses. As can be seen, MPA is not dependent on the initial poses of a 3D model, and only a very small difference exists among the minimum area found. MPA is independent of initial poses because we adopt the global optimization approach PSO to find the first principal axis with minimum projection area. In the initial stage of the search, it uses a global search to avoid local minimums and then enhances the local search ability to find an as accurate as possible global minimum projection area.

(4) *Evaluation with respect to PSO's iteration number*. In PSO, the number of iterations is an important factor which influences the accuracy and search time. To test the influence of iteration number on the alignment results, we apply MPA using different iteration numbers to find the first principal axis. Figure 12 shows the results. We find that after 11 iterations the area converges to about 0.291 and we achieve the best results, which are below 0.2913 at 30–40 iterations. We also find that the convergence speed is fast. Usually after 10 iterations, MPA already finds an area which is close to the optimal one. For the same accuracy as PSO, the brute-force method needs a much longer time. For example, MPA based on 10 iterations finds smaller area than the brute-force method using an L_4 icosahedron (1281 vertices) for axis sampling; MPA needs about 8 seconds while the brute-force method takes about 43 seconds. Based on 40 iterations, MPA finds a smaller area than the brute-force method based on the L_6 icosahedron (40004 vertices); MPA averagely needs 46 seconds and the brute-force method needs 530 seconds for the PSB models.

4.3 Evaluation with respect to retrieval performance

In this section, we evaluate MPA in terms of retrieval performance improvement on a rotation-dependent shape descriptor by comparing the retrieval performances when using different alignment methods such as PCA, NPCA, CPCA, and our MPA. For the selection of a rotation-dependent shape descriptor, we choose to modify the Light Field descriptor [2], which is a famous and typical shape descriptor. The distance of two models is defined as the minimum distance between 10 corresponding views of the two models. The Light Field descriptor adopts an integrated image shape descriptor which contains 35 Zernike moments and 10 Fourier descriptors and uses an L1 distance metric to measure the differences. To find the minimum distance between

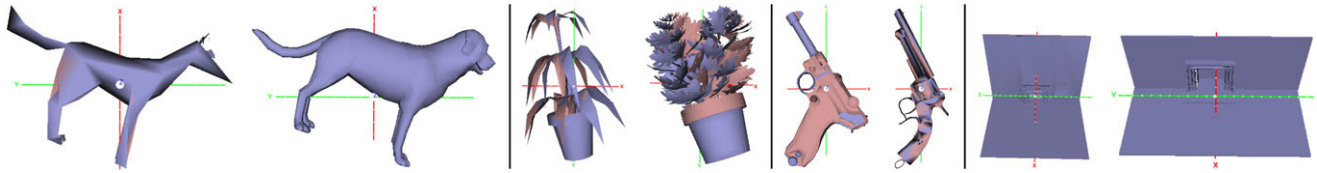


Fig. 8 Examples showing that the alignment results of our MPA algorithm are still consistent within classes even if the result axes are not the perfect ones

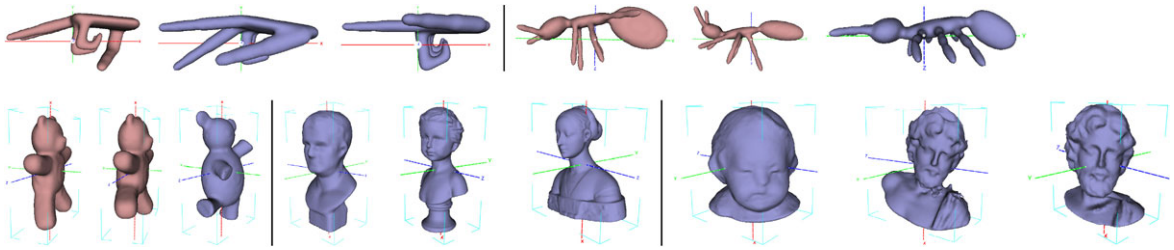


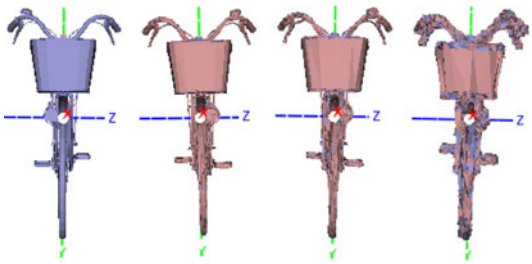
Fig. 9 Examples indicating MPA can align similar models in similar poses

Table 3 Comparison of retrieval performance among our MPA and three other alignment algorithms based on the modified-LF shape descriptor

Methods	NN	FT	ST	DCG	AP
PSB					
MPA	**60.4**	**33.5**	**43.2**	**0.603**	**50.5**
CPCA	58.7	32.8	42.6	0.597	49.8
NPCA	57.8	32.3	41.6	0.596	49.3
PCA	58.4	31.1	40.7	0.586	48.3
NIST					
MPA	**83.5**	**42.2**	**55.2**	**0.745**	**53.8**
CPCA	81.3	41.5	53.7	0.734	52.7
NPCA	81.1	38.2	49.9	0.724	49.9
PCA	77.3	39.2	50.4	0.710	49.7
WMB					
MPA	**89.5**	**46.7**	**59.8**	**0.783**	**59.7**
CPCA	84.8	44.6	58.8	0.765	57.6
NPCA	86.3	44.2	57.5	0.765	56.5
PCA	85.5	44.2	59.0	0.764	57.1

two models, the original Light Field approach performs an alignment process by rotating a camera system of 20 cameras set on the vertices of a regular dodecahedron. We modify this original Light Field descriptor by replacing its internal alignment process with an explicit alignment step using PCA, NPCA, CPCA, or our MPA, and we name it modified-LF.

To perform a comprehensive evaluation for 3D model retrieval performance, we employ five metrics including Nearest Neighbor (NN), First Tier (FT), Second Tier (ST), Discounted Cumulative Gain (DCG) [18], and Average



(a) 0.291556 (b) 0.292822 (c) 0.295589 (d) 0.308511

Fig. 10 Examples indicating MPA's robustness to noise. (**a**) The view from the first principal axis with minimum projection area for the original bicycle model, (**b**)–(**d**): the views from the first principal axis of the bicycle model when we added noise by randomly moving each vertex with a small displacement vector whose norm is bounded by 0.12%, 0.25%, and 1% of the diameter of the model's bounding box, respectively. The number underneath each view is its normalized projection area

Precision (AP). NN measures the percentage of the closest matches that are relevant models. FT/ST is the percentage of a class that has been retrieved among the top $(C-1)/2(C-1)$ list, where C is the cardinality of the relevant class of the query model. DCG measures the accuracy of the retrieval list using the summed weighted value related to the positions of the relevant models. AP is used to measure the overall performance averaged over all the models.

We tested the modified-LF retrieval algorithm on the PSB, NIST, and WMB databases using the above-mentioned different alignment algorithms. Table 3 compares their performances. Compared with PCA, NPCA, and CPCA, our MPA achieves better performances in all the five performance metrics. The main reason for the improvement is our achieving a higher percentage of consistent alignment results for models belonging to the same class.

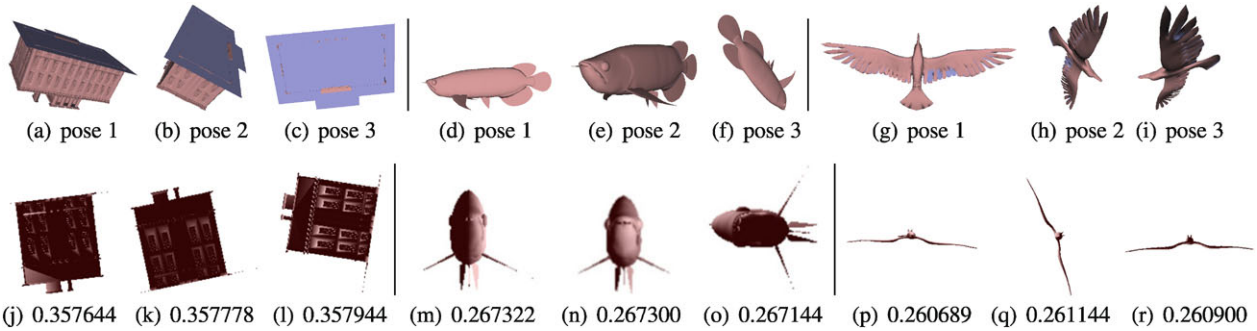


Fig. 11 Three sets of examples indicating MPA's robustness to initial poses. PSO is used to search for the first principal axis. The *second row* shows the corresponding views from the first principal axis for the models in the *first row*. The rotated views are only due to different up-vectors of the cameras during rendering. Note that we get the same final alignment results for each set of models. The number underneath each view is its normalized projection area

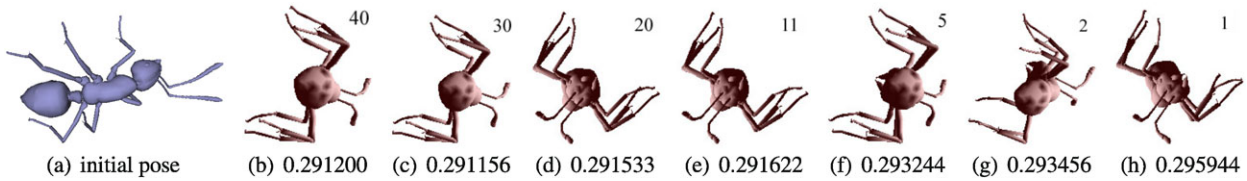


Fig. 12 Examples showing MPA's first principal axis results based on PSO's iteration number. (**b**)–(**h**): the axes with minimum projection area based on the iteration numbers displayed in the upper right corner. The rotated views are due to different up-vectors of cameras during rendering. The number underneath is the normalized projection area

4.4 Limitations of MPA

As shown in the previous experiments, MPA has a good performance in 3D model alignment. Nevertheless, it has some limitations. Firstly, it does not work well for certain types of models which do not have normalized poses with minimum projection areas. Some examples are shown in Fig. 8. Secondly, though in general the axes found are accurate, we cannot guarantee a perfect alignment for all models, that is the $z+$, $x+$ and $y+$ axes correspond to the front, left, and top parts of a model, respectively. This is because we do not consider the semantics information of models during the alignment. Although we already utilize the static stability and view entropy, our approach still lacks the semantics information for deciding the perfect axes orientations for all 3D models.

5 Conclusions and future work

A novel Minimum Projection Area-based alignment approach (MPA) for 3D model pose normalization was proposed in this paper. It is based on the idea of finding two perpendicular principal axes with minimum projection area. PSO was employed to efficiently find the axis with minimum projection area. Three evaluation experiments were conducted: (1) accuracy in terms of finding three axes which are parallel to the axes of the ideal canonical coordinate frame of a 3D model; (2) robustness of results with respect to model variations, noise, initial poses, and PSO iteration number; and (3) 3D model retrieval performance using a rotation-dependent shape descriptor. All three experiments demonstrated the ability of our MPA approach to find a consistent pose for similar models. Experimental results showed that our MPA algorithm achieves a better performance compared to PCA, CPCA, and NPCA in terms of axes accuracy and 3D model retrieval.

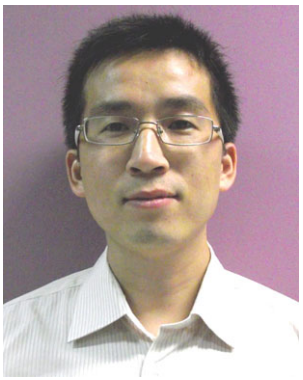
Regarding the limitations of MPA, we think it can be improved by combining other types of features, such as symmetry, with projection area when searching for the principal axes. We would like to investigate this further. Another possible future work is to perform semantics analysis for axis ordering with the ultimate goal of achieving perfect alignment.

References

1. Chaouch, M., Verroust-Blondet, A.: Alignment of 3D models. Graph. Models **71**(2), 63–76 (2009)
2. Chen, D.Y., Tian, X.P., Shen, Y.T., Ouhyoung, M.: On visual similarity based 3D model retrieval. Comput. Graph. Forum **22**(3), 223–232 (2003)
3. Eberhart, R.C., Hu, X.: Human tremor analysis using particle swarm optimization 1999. In: Proc. of the Congress on Evolutionary Computation, pp. 1927–1930 (1999)

4. Fang, R., Godil, A., Li, X., Wagan, A.: A new shape benchmark for 3D object retrieval. In: Proc. of the 4th International Symposium on Advances in Visual Computing (ISVC), pp. 381–392 (2008)
5. Fu, H.B., Cohen-Or, D., Dror, G., Sheffer, A.: Upright orientation of man-made objects. ACM Trans. Graph. **27**(3) (2008)
6. Jayanti, S., Kalyanaraman, Y., Iyer, N., Ramani, K.: Developing an engineering shape benchmark for CAD models. Comput. Aided Des. **38**(9), 939–953 (2006)
7. Jolliffe, I.: Principal Component Analysis, 2nd edn. Springer, Heidelberg (2002)
8. Lee, C.H., Varshney, A., Jacobs, D.W.: Mesh saliency. ACM Trans. Graph. **24**(3), 659–666 (2005)
9. Lian, Z., Rosin, P., Sun, X.: Rectilinearity of 3D meshes. Int. J. Comput. Vis. **89**(2), 130–151 (2010)
10. Loop, C.T.: Smooth subdivision surfaces based on triangles. Masters Thesis, The University of Utah (1987)
11. Martinek, M., Grosso, R.: Optimal rotation alignment of 3D objects using a GPU-based similarity function. Comput. Graph. **33**(3), 291–298 (2009)
12. Napoléon, T., Sahbi, H.: From 2D silhouettes to 3D object retrieval: Contributions and benchmarking. EURASIP J. Image Video Process. **2010**(1), 1–17 (2010)
13. Papadakis, P., Pratikakis, I., Perantonis, S.J., Theoharis, T.: Efficient 3D shape matching and retrieval using a concrete radialized spherical projection representation. Pattern Recognit. **40**(9), 2437–2452 (2007)
14. Podolak, J., Shilane, P., Golovinskiy, A., Rusinkiewicz, S., Funkhouser, T.: A planar-reflective symmetry transform for 3D shapes. ACM Trans. Graph. **25**(3) (2006)
15. Pu, J., Karthic, R.: An automatic drawing-like view generation method from 3D models. In: Proc. of ASME IDETC/CIE 2005, 25th Computers and Information in Engineering (CIE) Conference, pp. 301–320 (2005)
16. Pu, J., Lou, K., Ramani, K.: A 2D sketch-based user interface for 3D CAD model retrieval. Comput. Aided Des. Appl. **2**(6), 717–725 (2005)
17. Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: Proc. of IEEE International Conference on Evolutionary Computation (ICEC), pp. 69–73 (1998)
18. Shilane, P., Min, P., Kazhdan, M.M., Funkhouser, T.A.: The Princeton shape benchmark. In: Proc. of Shape Modeling International (SMI), pp. 167–178 (2004)
19. Tedjokusumo, J., Leow, W.K.: Normalization and alignment of 3D objects based on bilateral symmetry planes. In: Cham, T.J., et al. (eds.) MMM (1). Lecture Notes in Computer Science, vol. 4351, pp. 74–85. Springer, Berlin (2007)
20. Vázquez, P.P., Feixas, M., Sbert, M., Heidrich, W.: Viewpoint selection using viewpoint entropy. In: Ertl, T., Girod, B., Niemann, H., Seidel, H.P. (eds.) VMV Aka GmbH, pp. 273–280 (2001)
21. Vázquez, P.P., Feixas, M., Sbert, M., Heidrich, W.: Automatic view selection using viewpoint entropy and its applications to image-based modelling. Comput. Graph. Forum **22**(4), 689–700 (2003)
22. Veltkamp, R.C., ter Haar, F.B.: SHREC 2007 3D retrieval contest. Technical Report UU-CS-2007-015, Department of Information and Computing Sciences, Utrecht University (2007)
23. Vranic, D.: 3D Model retrieval. PhD Thesis, University of Leipzig (2004)
24. Yamauchi, H., Saleem, W., Yoshizawa, S., Karni, Z., Belyaev, A.G., Seidel, H.P.: Towards stable and salient multi-view representation of 3D shapes. In: Proc. of Shape Modeling International (SMI), p. 40 (2006)

Henry Johan received his BS, MS, and PhD degrees in Computer Science from the University of Tokyo (Japan) in 1999, 2001, and 2004, respectively. From 2004 to 2006, he was a post-doctoral fellow in the Department of Complexity Science and Engineering at the University of Tokyo. Since 2006, he has been an assistant professor in the School of Computer Engineering at Nanyang Technological University (Singapore). His research interests in computer graphics include rendering, animation, and shape analysis. He is a member of ACM SIGGRAPH.



Bo Li received his BS and MS degrees in Computer Science and Technology from Xi'an Polytechnic University (China) in 2002 and Xi'an Jiaotong University (China) in 2005, respectively. He was a teaching assistant in the School of Science at Xi'an Polytechnic University from 2002 to 2007. From 2005 to 2007, he was a senior software testing engineer in Zhong Xing Telecommunication Equipment Company Limited. Since 2007, he has been a PhD student in the School of Computer Engineering at Nanyang Technological University (Singapore). His research interests include 3D model retrieval, shape matching, and 3D modeling.

Yuanmin Wei received his BS degree in Computer Science and Technology from Jilin University (China) in 2010. Since 2010, he has been an MS student in the School of Computer Engineering at Nanyang Technological University (Singapore). His research interests include 3D model alignment and computer graphics.

Iskandarsyah received his BE degree in Electrical and Electronic Engineering and MS degree in Digital Media Technology from Nanyang Technological University (Singapore) in 2009 and 2010, respectively. Currently, he is a research associate in the School of Computer Engineering at Nanyang Technological University. His research interests include machine learning, image processing, scientific visualization, game design, and programming.