Variance and Symmetrical-based Approach for Optimal Alignment of 3D Model

Luh Putu Ayu Prapitasari, Parth Rawal and Rolf-Rainer Grigat
Vision Systems Institute
Hamburg University of Technology
Hamburg, Germany
Contact e-mail: putu.prapitasari@tuhh.de

***Abstract*—The concept of building 3D models, known as 3D reconstruction, already exists since the last few decades. However, by manually aligning the objects during acquisition phase does not guarantee that the output, the 3D models, will be perfectly aligned with the computer's world coordinate system. It mainly happens because in real world it is quite challenging to get perfect measurements, especially for the irregular objects. In this paper we address this problem by proposing a method to be used on the post processing phase of the 3D reconstruction process. The method is based on the variance and symmetricity of the object's point cloud which is acquired during acquisition. For the evaluation, we applied and evaluated the proposed method to both synthetic and reconstructed 3D models. The results are significant and show that the method capable of aligning the models to a fine resolution of 1' (one minute) angle.**

Keywords—3D models, Alignment, Variance, Symmetry, Reconstruction

I. INTRODUCTION

Alignment of a 3D model in 3D reconstruction is an important task and usually done during acquisition or registration phase and later on, for the better output, another alignment procedure could also be done as part of the post processing task [1][2][3][4]. There may be several requirements or problems which arises to the need of aligning a 3D model. One such requirement is optimal dimension measurement from a 3D reconstructed model. In reality, it is almost impossible to align a 3D model to perfection during its acquisition in hardware-wise and some mathematical approaches are definitely required to align it to perfection. Our task is to align the 3D model, software-wise, after acquisition. The method in [5] describes dealing with Continuous Principal Component Analysis (CPCA) for models having two orthogonal symmetry planes passing through an axis and another tool called Local Translational Invariance Cost (LTIC) used when there exists only one or no plane of symmetry. However, our work uses another approach which is quite different from the one proposed in that paper. The unique feature which makes the approach proposed here special is that, there is no featured data such as normal, surface, faces or edges are required to execute the algorithm, which was necessary for the approach listed above. Moreover, unlike some other methods, such as Iterative Closest Point (ICP) [6], which require a reference to align a misaligned model, it is as well not the case in our work.

In the later sections, the working all the four methods in the entire process namely, translation, alignment by variance, alignment by symmetry and classification are described. Testing and evaluation are done in section 4 and in section 5 all the works are then summarized.

II. CONDITIONS AND ASSUMPTIONS

Apart from the input datasets, there are a few important conditions, criteria and assumptions in the beginning which are important to know before starting the process.

Figure 1. Different objects to be aligned after 3D reconstruction. From top left to bottom right: cube, cylinder, triangle, owl, frog and two owls.

2.1. Object Lies on a Plane

The assumption made here initially is that the object lies on a rotating table. It means that there can be only 3 degrees of freedom possible theoretically for the misalignment cases, which are translation in X-axis and Y-axis and rotation in XY plane. In practice, it is possible that the table is not aligned with respect to ground or camera axis and small alignment along other 3 degrees of freedom are also possible. This means that the algorithm can align a model in 2 dimensions in a single implementation. If the object is not lying on a plane, the same algorithm could align it in 3 dimensions on being executed twice, i.e. first aligning along XY plane and then along XZ or YZ plane.

2.2. Model is Free from Outliers and Concentrated Noise

Since no information is available about the features of a reconstructed model, it is very important that all outliers and concentrated noise points are removed. If they exist, then it might affect the final alignment in respect to all 3 degrees of freedom.

2.3. Criteria and Assumption for Better Human Perception

The algorithms proposed here take care of all ambiguities during alignment, and hence it is necessary to state, how is the model defined for human perception [5]. The human perception model is defined such as: for a specified misalignment range, which is 45° in default case, the model will be aligned to its nearest alignment axis found. This assumption is made considering the fact that humans normally set the object close to its alignment axis even though it is not possible to align perfectly. As a result, if more than one solutions for alignment have been detected, the algorithm aligns it to the one nearest to the misaligned axis. This solves the problem of ambiguity in alignment.

III. Implemented Algorithms

Following sections show the working of each of the methods pointed in the flow.

3.1. Translation to World Origin

It is possible that the 3D reconstructed point cloud model is lying somewhere in the world frame but not at world origin. The first step is to move the model at world zero such that the center of mass of the object is at world zero. The Z coordinate of all the data points is to be ignored as the object is lying on a plane.

Algorithm 1 Method of Variance

INPUT: point cloud, misaligned range ▷ misaligned range is 45 by default
Translate model to world origin
procedure VarianceAlignInXY(point cloud)
  for i:=$i-n$ to $i+n$, $i=i+5$ **do** ▷ n =misaligned range
    Euclidean transformation of point cloud by i degrees
    Find variance in X direction from YZ Plane
  end for
  if global maxima exists **then**
    deg =value of angle with maximum variance; ▷ resolution is 5°
  else if global minima exists **then**
    deg =value of angle with minimum variance;
  end if ▷ deg =misalignment in degrees
  repeat above with smaller values of $n=1$ and $n=1/60$ each
  until resolution is of deg is $1'$
  $result$ =Euclidean transformation of point cloud by deg degrees
end procedure ▷ $result$ =aligned point cloud
OUTPUT: result, deg

The center of mass of object is modeled as the arithmetic means values of X and Y coordinates of all points in the point cloud. As a result, 2 out of 3 degrees of freedom have been found. The next step is to execute translation of point cloud in X and Y coordinate equal to the negative of the mean values of point cloud already found.

3.2. Alignment by Method of Variance

Variance, denoted as s^2, is a measure of the spread of data from its mean value, which can be defined as follows [7]:

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2 \quad (1)$$

where in equation 1, n is the total number of data points, X_i is the i^{th} data point whereas $\bar{X}$ is the mean of all data points. Note that if the entire dataset of points is transformed to origin from its mean value, then $\bar{X}=0$ and the variance in equation1 1 changes to equation 2.

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}X_i^2 \quad (2)$$

Equation 2 gives variance along both coordinate axes. If variance is calculated only along the x-axis, then only x component of the dataset X is considered and the equation 2 changes to equation 3.

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}x_i^2 \quad (3)$$

Equation 3 is simple, the sum of squares of x components of all the data points divided by $(n-1)$ number of data. This equation is the basis in the approach to align a 3D model by the method of variance. The idea is to rotate the translated model over its entire misaligned range and calculate variance for all the points along X direction from YZ plane. A graph of variance versus angle is formed. The global minima and global maxima values of variance will satisfy the solution for alignment problem. The next step is to resolve ambiguities.

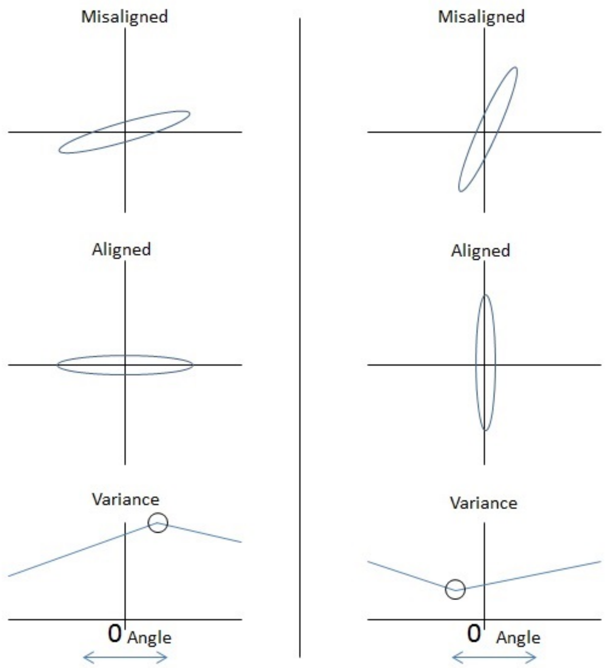

Figure 2. Comparison of two aligned model (resolving ambiguities).

Figure 2 shows an example on resolving ambiguities. In the first case (left), point cloud is more misaligned along X direction where as in second case (right), it is more aligned along Y direction. In the first case (left), alignment will lead to align itself such that variance in X direction is maximum at that angle. However, in second case (right), the variance along Xwill be minimum. This is also represented in the graph below for understanding. To resolve these ambiguities, both cases have to be dealt with. One of the successful ways to do that is to find whether it has global minima or maxima nearby. Now using this approach, we have resolved the ambiguity and the model is aligned to suit human perception.

Algorithm 1 shows the flow and approach of this method. The first step is to translate the model to world origin as explained in previous section. Next, the translated model is rotated over its entire misaligned range and variance is calculated for all the points along X direction from YZ plane. A graph of variance versus angle is formed. If it has a value of variance as global maxima, the angle at which global maxima is formed is found out and if it has a global minimum, the angle at which global minima is formed is found out. The reason to do so will be explained in the next section.

For a misaligned range over m° along both sides, clockwise and counter clockwise, the total number of values of variance obtained are equal to $\frac{2m}{r}+1$ where r is the resolution in degrees. Therefore, the first iteration with $45^{\circ}$and $r = 5^{\circ}$ give 19 values of variance. Next, the process is iterated again in steps of $r = 1^{\circ}$ from a range equal to angular step size of the iteration before, which is $m = 5°$ and 11 values of variance with a resolution of 1° are calculated. Once again, a check is made to find out at which angle is global maxima or minima is found. Finally, last iteration is made and end up with a resolution of 1'. The last iteration process will result in 121 values of variance, with $m = 1^{\circ}$ and $r = 1'$. Now, the value of the global maxima or minima can be found out, and the value of angle at the given value of variance is the misaligned angularity. The point cloud is then transformed with the help of Euclidean Transformation to align itself and form the best possible solution.

The method of variance basically is similar to Principal Component Analysis [8]. It is in our case, however, preferred over PCA because it is easier to resolve ambiguities with variance method. It is to be noted that neither variance nor PCA will work for 3D objects having more than 2 symmetry planes since the variance for such 3D objects will not change on being misaligned. Also, the error points in the point cloud away from the centroid will contribute to erroneous alignment by variance method. For avoiding such failure, a more reliable method of symmetry is used.

3. 3. Alignment by method of Symmetry

This approach is totally different from the previous approach of variance. Here, all the points on one side of XY plane are mirrored onto other side and correspondence is established between the mirrored points and points lying in the same side. For each point, a nearest neighbor is found in the mirrored point set. If the object has at least one symmetry plane, the mirrored half will fall exactly on the points lying on other side and distance between nearest neighbor will be ideally zero. The sum of the square of the distances from their respective nearest neighbor gives the squared error. The 3D object is rotated and this procedure is checked, so as the symmetry is found, the 3D object can be transformed to the orientation leading to alignment. The mirroring is explained below with equations in set notations where, $p_i(x_i, y_i, z_i) \in P$, and P is $n \times 3$ data matrix.

$\{P_{x+} | p_i(x_i, y_i, z_i) \in P, i = 1, \dots, n;\ x_i > 0\}$,

$$P' = P_{x+} \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

$\{P_{x-} | p_i(x_i, y_i, z_i) \in P, i = 1, \dots, n;\ x_i \leq 0\}$ and by assigning

$$P'' = P_{x-}, \tag{5}$$

For each point in P", search for a nearest neighbor in P'.

The squared error (e^2) obtained from the algorithm can be formulated in equation as follows:

$$e^2 = \sum_1^n (p_i - p'_i)^2 \tag{6}$$

In equation **Error! Reference source not found.**, p' is the nearest neighbor of p. A summation of such error along all query points give us squared error (e^2).

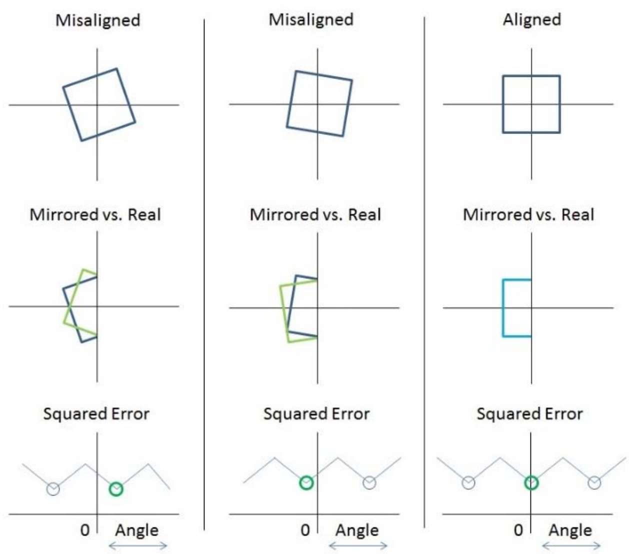

Figure 3. Mirroring, matching and resolving ambiguities.

The algorithm 2 shows the implementation of this approach. The overall procedure is almost same as the one in method of variance. First, we translate the model to world origin. Next, we iterate the method by rotating the model in smaller steps like 5°, 1° and 1'. Now, after rotating the translated model in definite steps, all the points in the right-hand side of YZ plane i.e. each point $P_i(x, y, z)$ with $x > 0$ is mirrored such that $P_i(x', y', z')$ is $x' = -x, y' = y$ and $z' = z$. In the form of equation, it can be written as: $\{\forall P_i(x, y, z) \in P, i \in 1,2, \dots, n | x > 0\}$,

$$P' = P \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{7}$$

Here P and P' are $n \times 3$ matrices with each row containing 3D points. Take a note that the equation 4 is nothing but simple mirroring of all points about YZ plane. After mirroring, all the points in right side of YZ plane are mapped onto left side. For every point in P', a nearest neighbor in left side of YZ plane is searched for P. This is carried out by the nearest neighbor search (NNS) approach [9]. Upon finding the nearest neighbor, error in symmetry is measured, which is nothing but Euclidean distance between those two points. On completion of the process for all the points, square of the error is summed up for all the points, which we term as *squared error*. The squared error so calculated is zero for an ideally symmetrical model. Next, ambiguity is resolved for nearest alignment axis which explained in detail in the next section 3.4. In the end, point cloud is transformed with the help of

Euclidean transformation to align itself and form the best possible solution within a fine resolution of 1'.

3.4. Symmetry Detection and Resolving Ambiguities

Figure 3 shows the flow of the algorithm visually. At first, a misaligned input model is given, which is compared with its mirror along YZ plane or Y axis in 2D case. The squared error is minimum when model is symmetric about YZ plane and maximum when it cannot match its mirror model. At the end, nearest angle where the squared error is minimum is found out and the point cloud model is transformed to that angle of rotation. Local minima is considered in this case to resolve ambiguity unlike global minima in the former approach. A suitable explanation to that is because of multiple possibility of existing symmetry planes. The algorithm should be able to align it to any one of them which is found to be nearest. For reconstructed point cloud models, it is very much possible that the squared error at one symmetry plane is much lesser than that to others due to noise and erroneous points, but in order to make the model less dependent on noise, nearest value of local minima is selected even though it is not a global minima. Hence, the alignment axis will change with respect to the orientation of input model which is required to encompass human perception factors. This special feature of this method makes it unique from previous method.

Algorithm 2 Method of Symmetry

```
INPUT: point cloud, misaligned range          ▷ misaligned range is 45 by default
Translate model to world origin
procedure SYMMETRYALIGNINXY(point cloud)
    for i:=i − n to i + n, i = i + 5 do       ▷ n =misaligned range
        Euclidean transformation of point cloud by i degrees
        Find mirror points for all points in point cloud in right hand side of YZ plane
        Find all the points in original point cloud in the left hand side of YZ plane
        Find a nearest neighbor for each mirrored point, a point in left hand of YZ plane in original point cloud
        Find error as Euclidean distance between two points
        Find squarederror as sum of squares of Euclidean distances between each pair of nearest neighbor
    end for
    Find nearest local minima of squarederror
    deg = value of angle at local minima
    repeat find minimum squarederror for n = 1 and n = 1/60 each
    until resolution is of deg is 1′
    result =Euclidean transformation of point cloud by deg degrees
end procedure                                 ▷ result =aligned point cloud
OUTPUT: result, deg, squared error
```

3.5. Classification of Reconstructed Models

The value of squared error (e^2), even after aligning, in the method of symmetry indicates the error which could be composed of two parts:

i. Model is symmetric and the squared error (e^2) is due to noisy points.

ii. Model is asymmetric and the squared error (e^2) is due to asymmetrical points and noisy points together.

Even though the value is influenced by noisy points, it can be still used to measure symmetricity of a reconstructed point cloud model. However, it has to be slightly modified in order to compare the values between different 3D objects.

$$S = \frac{\overline{P}_x}{e^2 \times n^{\frac{1}{10}}} \quad (8)$$

Equation **Error! Reference source not found.** has been derived based on various observations about relative size of the 3D objects as well as number of points in the point cloud. It is now independent of different models and unique in itself. Basically, the solution to the equation yield a number (for example: symmetrical factor). The symmetrical factor (S) values observed from 1 to 10 for reconstructed models and even more for synthetic models. The values 2 and 5 below are calibrated from the testing models. However, they are purely perception based and such parameters can be tuned fine in order to achieve correct classification. For this model, following are the values proposed for classifying reconstructed point cloud model into three different categories:

1) $S \geq 5$: Reconstructed model is symmetric in one or more number of symmetry planes and alignment is excellent and uses symmetry approach to align.
2) $2 < S < 5$: Reconstructed model is not symmetric but regular and alignment is very good and uses variance approach to align.
3) $S \leq 2$: Reconstructed model is irregular and alignment is good and uses again variance approach.

In a way, symmetricity factor was a byproduct of method of symmetry alignment. As a result, the same equation can be used to compute symmetricity of different models and serve itself as a tool to automatically decide on which method to use for better alignment.

IV. TESTING AND EVALUATION

The entire evaluation is split into two parts namely, testing and performance. Testing is an important part of evaluation initially and provides a green signal to implement it on real specimens. On successful testing, performance is measured on actual datasets.

4.1. Testing

The 3D models from the synthetic testers as shown in figure 4, include a teapot with one plane of symmetry, a perfect cube with four planes of symmetry and a couch again with one plane of symmetry. All the models have different sizes. All these models are aligned by default. The idea is to misalign them with a known angle and apply the algorithms mentioned here to realign them again. In the following table 1, recorded results for 5 different testers by both methods of alignment are shown along with their symmetric values. The term *Size* in the table shows the dimension in centimeter of minimum bounding box in XY plane, n is the total number of points in the model, *Misalign* is known misalignment applied on test model in degrees, *Align* is result from respective methods, *t* is time taken by algorithm is seconds, *S* is symmetricity factor and *Acc* is accuracy defined as $\left(1 - \frac{Misalign+Align}{Misalign}\right) \times 100\%$.

It is to be noted that the model 'perfect cube' fails the alignment algorithm based on variance as expected. Cube has 4 symmetry planes and therefore the variance of all the points along one direction will be constant. This problem is solved when aligned using symmetry approach.

4.2. Performance

The evaluation of six different types of 3D reconstructed models of six real objects, as depicted in figure 1, is done here. Table 1 and 2 show the performance of our algorithm on the reconstructed models. It can be noted again that the method of variance fails to align 'cube'. But then alignment by symmetry aligned it correctly. In other models, the results from both the methods are comparable but the ones by symmetry are found to be better: especially in cases where symmetricity factor is more, such as cube, triangle, cylinder and owl.

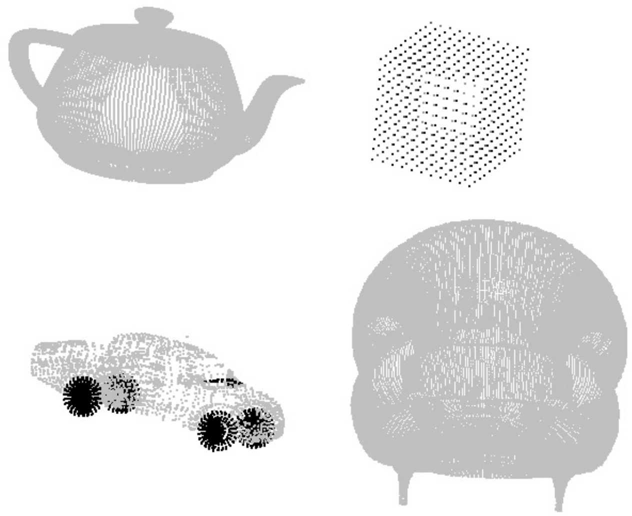

Figure 4. 3D tester models. From top left to bottom right: teapot, perfect cube, car and couch. *Source:* [10], [11].

As per classification of models on the basis of symmetricity factor (S) mentioned in Section 3.5, the six reconstructed models are classified as shown in table 3.

Table 3. Classificatiob of reconstructed models.

Symmetric	**Regular**	**Irregular**
$S \geq 5$	$2 < S < 5$	$1 < S < 2$
Cylinder	Frog	Two Owls
Owl		
Triangle		
Cube		

V. CONCLUSION

The entire algorithm of alignment of a 3D reconstructed model is implemented in Matlab R2015b and is successfully tested for misalignments within 90° in XY plane. The results of the reconstructed model are also very satisfying, aligning models to a fine resolution of 1' as shown in Table 1 for the second case, the teapot model, when it is aligned by using the Symmetry method. Also at the end, classification of reconstructed models also value adding. However, it is also observed that the calculation time got increased slowly for models with more than 200,000 points. Important features, limitations and future works are explained in sections below.

5.1. Important Features and Key Points

On a concluding note, the noteworthy features and keypoints of the work are highlighted here:

a) Features: (1) robust performance on different misalignment incorporated by human perception factors, (2) does not require featuring information in a model about faces, edges, or their normal, (3) does not require a reference aligned model for aligning procedures, (4) fields good results even with noisy data points.
b) Key points: (1) able to align the model by two different possible approaches, symmetry and variance, (2) calculates value of symmetricity factor to give a good comparison between different models, (3) classifies the model, which gives information about how good is alignment axis defined in that case.
c) Limitations: (1) involves lot of calculations due to handling of a large number of data points, since down-sampling the point cloud is not a good choice in this case, (2) takes fair amount of time in performing alignment by method of symmetry, (3) uses trial and error approach. It runs a loop and calculates the parameters rather than solving them directly.

5.2. Future Works

The method proposed here to align model in 3 degrees of freedom can also be used to align in 6 degrees of freedom in same steps and the time taken to align in 6 degrees of freedom as a result will be only 2 times to that in 3 degrees of freedom:

a) Translate model to world origin in X , Y and Z coordinates (which was only X and Y previously).
b) Align model in XY plane.
c) Align model in XZ or YZ plane.

One of the limitation is more computation time required, that needs to be considered if the algorithm would be used in real time.

REFERENCES

[1] Aubry, Mathieu, et al. "Seeing 3D Chairs: Exemplar Part-Based 2D-3D Alignment Using A Large Dataset Of Cad Models." Proceedings of the IEEE conference on computer vision and pattern recognition. 2014.

[2] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg et al., "The Digital Michelangelo Project: 3D Scanning of Large Statues," in Proceedings of the 27th annual conference on Computer graphics and interactive techniques. ACM Press/Addison-Wesley Publishing Co., 2000, pp. 131–144.

[3] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy, "Real-time 3D Model Acquisition," ACM Transactions on Graphics (TOG),vol. 21, no. 3, pp. 438–446, 2002.

[4] Van Lierde, Carl, et al. "Method and System for Registration/Alignment Of 3D Digital Models." U.S. Patent No. 9,740,960. 22 Aug. 2017.

[5] M. Chaouch and A. Verroust-Blondet, "Alignment of 3D Models," Graphical Models, vol. 71, no. 2, pp. 63–76, 2009.

[6] K.-L. Low, "Linear Least-Squares Optimization for Point-to-Plane ICP Surface Registration," Chapel Hill, University of North Carolina, vol. 4, 2004.

[7] M. Loeve, "Probability Theory, vol. II," Graduate texts in mathematics, vol. 46, pp. 0–387, 1978.

[8] I. Jolliffe, Principal Component Analysis. Wiley Online Library, 2002.

[9] J. L. Bentley, "Multidimensional Binary Search Trees Used for Associative Searching," Communications of the ACM, vol. 18, no. 9, pp. 509–517, 1975.

[10] T. Squid, “3D Models for Professionals, ” http://www.turbosquid.com/, [Online; accessed 27-July-2018].

[11] The MathWorks, Inc., “Documentation, Matlab,” 2018, http://de.mathworks.com/help/matlab/, [Online; accessed 27-July-2018].

Table 1. Evaluation results of Testers

No.	Misaligned Model		By Variance		By Symmetry		*S*
1	Model Size (cm) Points (n) Misalign (°)	: teapot : 6.5 × 4 : 41,472 : 30	t (sec) Align Acc (%)	: 0.59 : -29°55' : 99.72	t (sec) Align Acc (%)	: 14 : -30° : 100	4.8319
2	Model Size (cm) Points (n) Misalign (°)	: teapot : 6.5 × 4 : 41,472 : 15	t (sec) Align Acc (%)	: 0.59 : -15° : 100	t (sec) Align Acc (%)	: 12 : -15°1' : 99.89	21.6290
3	Model Size (cm) Points (n) Misalign (°)	: car : 90 × 37 : 9,218 : -20	t (sec) Align Acc (%)	: 0.51 : 19°53' : 99.42	t (sec) Align Acc (%)	: 3 : 19°47' : 98.92	10.1858
4	Model Size (cm) Points (n) Misalign (°)	: couch : 90 × 87 : 146,794 : -30	t (sec) Align Acc (%)	: 1.15 : 29°41' : 98.94	t (sec) Align Acc (%)	: 65 : 29°58' : 99.89	35.4026
5	Model Size (cm) Points (n) Misalign (°)	: perfect cube : 2 × 2 : 726 : 15	t (sec) Align Acc (%)	: 0.45 : -12°21' : 82.33 **failed**	t (sec) Align Acc (%)	: 1 : -15°13' : 98.56	7.2817

Table 2. Evaluation results of reconstructed models.

No.	Input Model		By Variance		By Symmetry		*S*
1	Model Size (cm) Points (n)	: cube : 10.75 × 10.75 : 130,765	t (sec) Align	: 1.03 : 34° **failed**	t (sec) Align	: 67 : -0°28'	5.0545
2	Model Size (cm) Points (n)	: triangle : 9.5 × 4.5 : 93,490	t (sec) Align	: 0.84 : -0°9'	t (sec) Align	: 50 : 2°5'	5.1537
3	Model Size (cm) Points (n)	: cylinder : 7 × 7 : 138,792	t (sec) Align	: 1.05 : -2°50'	t (sec) Align	: 60 : -6°58'	8.8050
4	Model Size (cm) Points (n)	: owl : 16 × 11.5 : 111,061	t (sec) Align	: 0.92 : -1°41'	t (sec) Align	: 52 : -0°12'	6.7110
5	Model Size (cm) Points (n)	: frog : 14 × 11 : 102,148	t (sec) Align	: 0.90 : 9°25'	t (sec) Align	: 49 : 11°31'	4.8235
6	Model Size (cm) Points (n)	: two owls : 9 × 8 : 12,728	t (sec) Align	: 0.44 : -24°18'	t (sec) Align	: 6 : -21°40'	1.8896