Proceedings of the ASME 2008 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2008
August 3-6, 2008, Brooklyn, New York, USA

DETC2008-49698

EUCLIDEAN SYMMETRY DETECTION FROM SCANNED MESHES BASED ON A CONBINATION OF ICP AND REGION GROWING ALGORITHMS

Tomohiro Mizoguchi* **Satoshi Kanai***

Division of Systems Science and Informatics, Graduate School of Information Science and Technology,
Hokkaido University

ABSTRACT

Recently, meshes of engineering objects have been easily acquired by 3D laser or high-energy industrial X-ray CT scanning systems and they are widely used in product developments. For the effective use of scanned meshes in inspection, re-design, and simulation of the objects, it is important to reconstruct CAD models from the meshes. Engineering objects often exhibit Euclidean symmetries for their functionalities. Therefore, it is essential to detect such symmetries when reconstructing CAD models with compact data representations which are similar to the ones already defined in CAD systems. However, existing methods for reconstructing CAD models have not focused on detecting such symmetries. In this paper, we propose a new method that detects partial or global Euclidean symmetries, including translation, rotation, and reflection, from scanned meshes of engineering objects based on the combination of the ICP and the region growing algorithms. Our method can robustly and efficiently extract pairs of symmetric regions and their transformations under which the pair can be closely matched to each other. We demonstrate the effectiveness of the proposed method from experiments on various scanned meshes.

1. INTRODUCTION

3D laser scanning systems are widely used in product developments for acquiring geometric point cloud data from real-world engineering objects. More recently, high-energy industrial X-ray CT scanning systems, which have been rapidly developed, have enabled users to quickly and non-destructively obtain 3D image data of complex objects containing internal structures [1]. The acquired data is easily converted into a 3D mesh by well-known surface reconstruction algorithms, such as marching cube [2]. For inspection, re-design, and simulation of engineering objects, reconstructing CAD models from the scanned meshes is very important.

(a) Translation and rotation (b) Reflection

Fig.1: Examples of symmetries in engineering objects

Engineering objects often exhibit Euclidean symmetries for their functionalities. We define "*Euclidean symmetries*" as an arbitrary set of translations, rotations, and reflections. For example, two hexagonal portions exist in the object in Fig.1(a). They are estimated to be defined in the CAD system by first creating a solid corresponding to the hexagonal portion and then by translating it along the vectors and rotating it around the axis. On the other hand, when the right and left sides of the object seem symmetric, as with the object in Fig.1(b), they are estimated to be defined by first creating a solid corresponding to one side and then reflecting it about the plane. Therefore, it is essential to detect such Euclidean symmetries to reconstruct CAD models with compact data representations which are similar to the ones already defined in CAD systems. However, existing methods for CAD model reconstruction have not focused on detecting such symmetries [3,4,5,6].

Fig.2 shows an example of our approach for reconstructing a CAD model from a scanned mesh based on Euclidean symmetry detection. In this example, a reflective symmetry is considered. First, the reflective symmetry is detected, and the 3D partial mesh and the reflective plane are obtained. Then, the

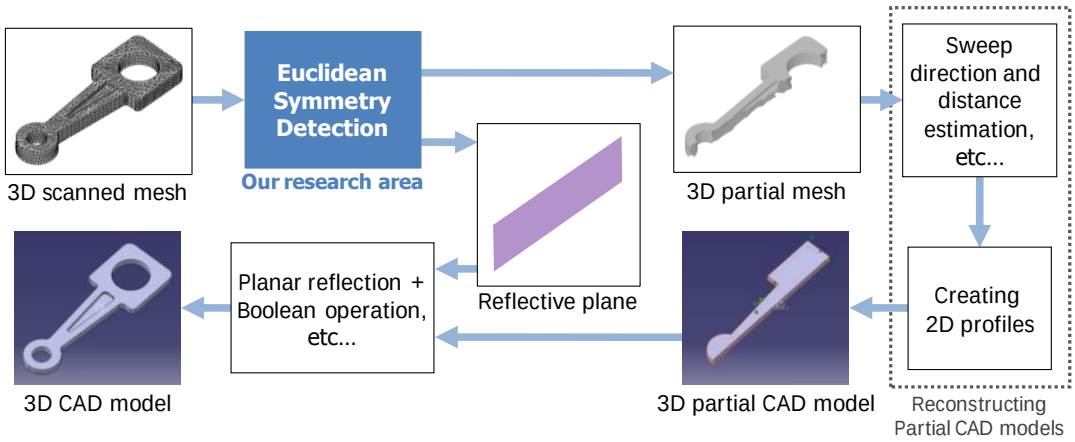


Fig.2: An example of our CAD models reconstruction approach (a case of reflective symmetry)

3D partial CAD model is reconstructed from the 3D partial mesh by estimating a sweeping direction and distance and by creating 2D profiles. Finally, the 3D CAD model is reconstructed by reflecting the partial CAD model about the extracted reflective plane and performing some Boolean operations.

From the example in Fig.1, for reconstructing a CAD model from the scanned meshes, it is necessary to detect partial or global Euclidean symmetries, including translations, rotations, and reflections.

1.1. Related works

Symmetry detection has been widely studied in computer vision and computer graphics fields, and many algorithms have been proposed.

Zabrodsky *et al.* [7] proposed an algorithm to detect approximate symmetries from 2D images by expressing symmetry as a continuous feature. Loy *et al.* [8] proposed the algorithm that detects rotational and reflective symmetries from 2D images based on Hough transform. However, these only aim at detecting symmetries from 2D images.

Sun and Sherrah [9] proposed the algorithm that uses the Gaussian image for detecting global symmetries from a 3D model. Martinet *et al.* [10] also proposed an algorithm to detect global symmetries from 3D meshes by examining the extrema and spherical harmonic coefficients of the object's generalized moments. However, these methods cannot detect partial symmetries.

Simari *et al.* [11] proposed the algorithm that detects planar reflective symmetries based on robust M-estimation and then hierarchically segments the mesh using detected symmetries. Podolak *et al.* [12] proposed to detect all of the possible planar reflective symmetries from a 3D mesh based on the voting scheme. However, these methods can detect only planar reflective symmetries and cannot do translational and rotational symmetries.

To our knowledge, only Mitra *et al.* [13] proposed the algorithm that detects partial or global symmetries, including translation, rotation, and reflection, from 3D meshes. This algorithm first computes principal curvatures and directions at each vertex. Then, for each pair of vertices, it computes a transformation under which the pair can be matched using estimated principal curvatures and directions, and then the algorithm maps each transformation into a particular point in the transformation space. Next, it finds the large clusters of mapped points in the transformation space through the mean-shift clustering algorithm. Finally, it detects symmetries as sets of vertex pairs that constitute the large clusters in the space and thus can be matched under the approximately same transformation. However, since this algorithm computes principal curvatures and directions at each vertex, it may fail to robustly detect symmetries from noisy scanned meshes. Moreover, it cannot detect symmetries from meshes that include many planar regions where principal curvatures and directions cannot be uniquely defined, which is a critical drawback for our purpose. Engineering objects often include many planar regions, and Mitra's algorithm cannot be used for scanned meshes of such objects.

1.2. Our purpose and an overview of our algorithm

In this paper, we propose a new method that detects partial or global Euclidean symmetries from scanned meshes of engineering objects that include many planar regions. Our method is based on the combination of *iterative closest points* (*ICP*) [14,15] and *region growing* algorithms [16]. We define "*Euclidean symmetry detection*" as extracting a pair of partial or global symmetric regions on meshes and the transformation under which the pair can be closely matched to each other. In our case, a transformation includes an arbitrary set of translations, rotations, and reflections.

Since the ICP algorithm can accurately extract a transformation that matches a pair of regions on the mesh, it is reasonable to use the ICP algorithm for symmetry detections. However, it may fail to extract an accurate transformation if a pair of regions is sampled from planes [17]. Therefore, to solve this problem, we propose a method that starts symmetry

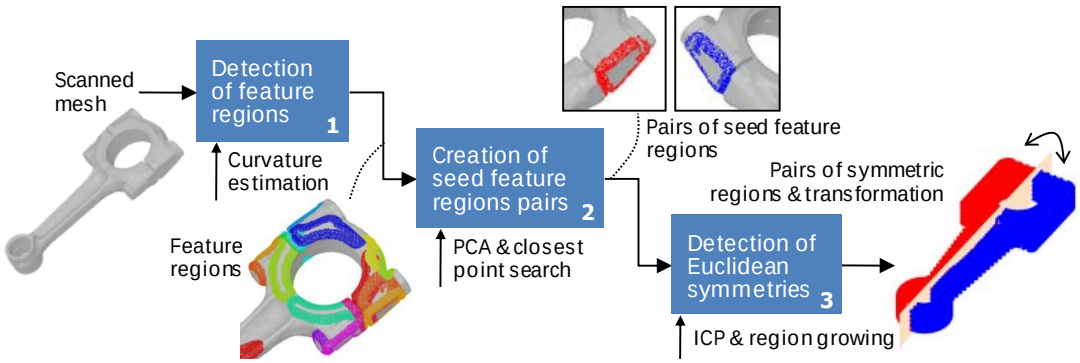


Fig.3: An overview of our algorithm

detection from the feature regions lying in the vicinity of planar ones as pairs of regions to be matched. Our method consists of following three steps as shown in Fig.3.

Step1: Detection of feature regions (Section 2)

Our method first estimates mesh principal curvatures at each vertex based on local quadratic polynomial surface fittings and detects sharp vertices [18,19]. Then, using detected sharp vertices, it extracts feature regions that lie in the vicinity of the planar regions and do not lie in the interior of them.

Step2: Creation of seed feature regions pairs (Section 3)

Next, our algorithm performs principal component analysis for each detected feature region. Then, for each pair of feature regions, it apply the three step processes, (1) comparing the number of vertices, (2) PCA matching, and (3) ICP matching. As a result of this process, seed feature regions pairs can be created as pairs of feature regions that have approximately the same number of vertices and can be closely matched under certain transformations.

Step3: Detection of Euclidean symmetries (Section 4)

Finally, our algorithm detects Euclidean symmetries based on the combination of ICP [14,15] and region growing [16] algorithms. It grows each pair of seed feature regions by iteratively adding a pair of neighboring vertices that can be matched under the current transformation to the seed feature regions and by computing a transformation using the ICP algorithm that matches the current pair of enlarged regions. As a result of this process, final pairs of Euclidean symmetric regions and their transformations are extracted.

Advantages of our algorithm:

The advantages of our proposed algorithm are summarized as follows:

1. Detection of Euclidean symmetries from meshes including many planar regions: Since our algorithm starts symmetry detection from a pair of feature regions that lie in the vicinity of the boundaries of planar regions and do not lie in the interior of them, it enables the user to detect symmetries even from a mesh that includes many planar regions.

2. Robustness for noisy meshes: Since our algorithm evaluates a set of vertices and essentially uses only vertex coordinates and does not require any additional information such as normals or curvatures in the symmetry detection steps (steps 2 and 3), it can robustly detect Euclidean symmetries from noisy scanned meshes.

3. Efficiency for large meshes: It is known that most of the computational time in the ICP algorithm is used to search for closest vertices [17]. Since our algorithm can find the new closest vertices during the iteration of the ICP algorithm by recursively searching the new pair from the current one using mesh connectivity, the efficiency in searching for closest points is increased, which makes our algorithm for Euclidean symmetry detection faster.

4. Class of detected symmetries: Our algorithm enables the detection of the large class of partial or global symmetries, including translations, rotations, and reflections.

In this paper, we deal with noisy scanned meshes that are reconstructed by marching cube based algorithms from the X-ray CT scanned data. In general, the edge lengths of such meshes become identical compared with those of laser scanned meshes. Our algorithm also works for open mesh that represents 3D surface data.

2. DETECTION OF FEATURE REGIONS

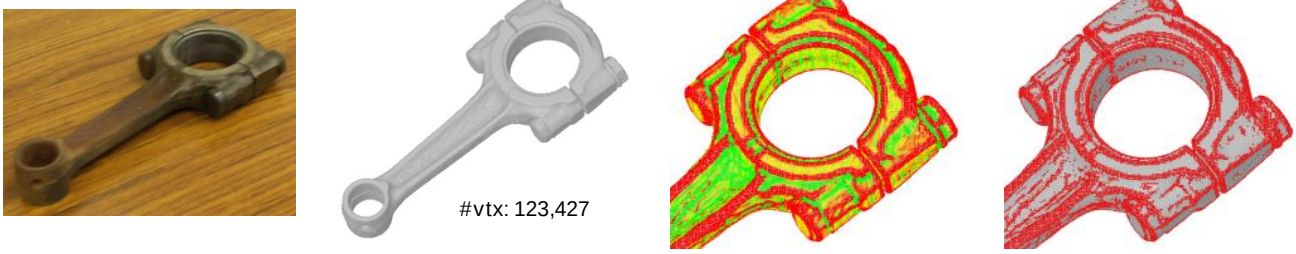
2.1. Curvature estimation and sharp vertices extraction

To estimate mesh principal curvatures robustly on scanned meshes, our algorithm first fits a quadratic polynomial surface $h(u,v)$ in Eq. (1) for a set of vertices $N(i)$ around each vertex x_i.

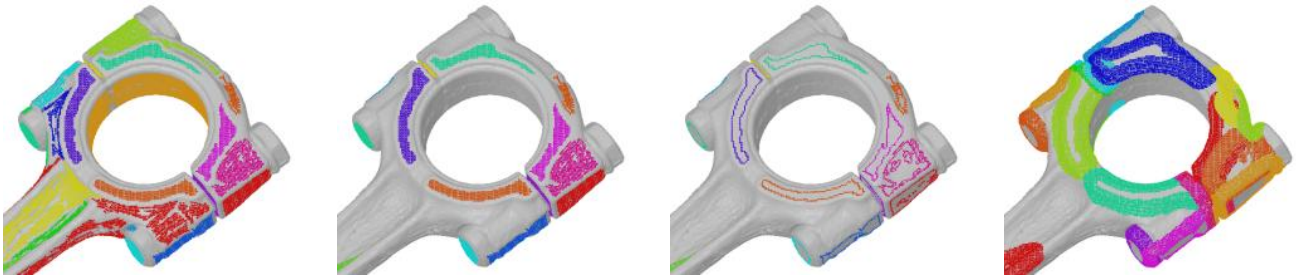
$$h(u,v) = a_0u^2 + a_1v^2 + a_2uv + a_3u + a_4v + a_5 \qquad (1)$$

$N(i)$ is defined as a set of vertices which are topologically connected to the center vertex x_i, including x_i, within the specified Euclidean distance satisfying Eq. (2)

$$\|\mathbf{x}_j - \mathbf{x}_i\| < W \cdot l_{i,avg}, \qquad (2)$$



(a) Mechanical parts (b) Scanned mesh (c) Max. curvature (d) Sharp vertices



(e) Initial regions $\{IR_j\}$ (f) Planar regions $\{PR_k\}$ (g) Boundary vertices $\{BR_l\}$ (h) Feature regions $\{FR_m\}$

Fig.4: Detection of feature regions

where $l_{i,avg}$ is the average length of edges connected to x_i and W is the parameter that specifies the neighboring size. From our previous experiments, we set W=3.0 for all meshes in this paper. Then, the algorithm calculates principal curvatures $\kappa_{i,\max}$ and $\kappa_{i,\min}$ at the corresponding point on $h(u,v)$ of x_i. Detailed algorithms of these processes are precisely described in [18,19].

Next, the algorithm classifies the vertex as a *sharp vertex* if the maximum curvature $\kappa_{i,\max}$ of x_i satisfies Eq. (3) [18,19].

$$\frac{1}{|\kappa_{i,\max}|} < th_{sharp} \cdot l_{i,avg} \tag{3}$$

The threshold th_{sharp} must be set depending on the geometry and the resolutions of the mesh. Intuitively, the larger th_{sharp} classifies the smaller number of vertices as sharp vertices, and vice versa. In our previous work [19], we found that this curvature estimation still worked well even for both noisy X-ray CT and laser scanned meshes.

Fig.4 shows an example of this process. The object in Fig.4(a) was scanned by the X-ray CT scanning system and the mesh in Fig.4(b) is reconstructed. Fig.4(c) shows estimated maximum principal curvatures and Fig.4(d) shows extracted sharp vertices.

2.2. Detection of feature regions

Next, our algorithm extracts a set of topologically connected non sharp vertices as initial regions IR_j as shown in Fig.4(e). Then, it fits a least-squares plane to each IR_j and evaluates the averaged fitting error $\bar{e}_j$. If $\bar{e}_j$ is less than threshold $\tau_{pla} l_{avg}$, our algorithm extracts IR_j as planar regions PR_k, as shown in Fig.4(f). Here l_{avg} is the averaged edge length of the mesh and we usually set $\tau_{pla} = 0.5$. Next, the algorithm extracts a set of topologically connected boundary vertices BR_l that lie on the boundaries in each PR_k as shown in Fig.4(g). A PR_k may contain multiple BR_l. Finally, the algorithm extracts a set of vertices $\{v_{m,s}\}$ that lie in the specified Euclidean distance Wl_{avg} from each BR_l as feature regions FR_m, as shown in Fig.4(h). The distance is calculated using Eq. (2), and we usually set $W = 10.0$.

3. CREATION OF SEED FEATURE REGIONS PAIRS

In this step, our algorithm creates pairs of seed feature regions that can be pairs of symmetric regions when grown by performing principal component analysis for each FR_i and then evaluating the matching error of a pair of feature regions $\langle FR_i, FR_j \rangle$.

3.1. Principal component analysis

Our method first calculates the barycenter $\mathbf{b}_i$ of each FR_i and then performs principal component analysis (PCA) to compute principal axes $\mathbf{A}_i = \langle \mathbf{a}_{i,1}, \mathbf{a}_{i,2}, \mathbf{a}_{i,3} \rangle$ corresponding to eigenvalues $\lambda_{i,1} \geq \lambda_{i,2} \geq \lambda_{i,3}$.

3.2. Initial matching and error evaluation

After performing the PCA, for each pair of feature regions $\langle FR_i, FR_j \rangle$, our method creates seed feature regions pairs based on following 3-step processes. We assume that if the regions represent the same geometry, the numbers of vertices in them are also the same.

Step1. Compare the number of vertices:

If $1.0-\alpha < |FR_i|/|FR_j| < 1.0+\alpha$, proceed to the next step. Otherwise, stop the process. We set $\alpha = 0.1$ for all meshes in this paper.

Step2. Perform PCA matching and evaluate error:

Compute a translational vector $\mathbf{t}_{ij} = \mathbf{b}_j - \mathbf{b}_i$ that matches $\mathbf{b}_i$ to $\mathbf{b}_j$ and then translate each vertex $v_{i,k} \in FR_i$ such that $\mathbf{v}'_{i,k} = \mathbf{t}_{ij} + \mathbf{v}_{i,k}$. Then compute a rotational matrix $\mathbf{R}_{ij}$ that matches $\mathbf{A}_i$ to $\mathbf{A}_j$ and then rotate each vertex around $\mathbf{b}_j$ such that $\mathbf{v}''_{i,k} = \mathbf{R}_{ij}\mathbf{v}'_{i,k}$. Next randomly select $\alpha(\%)$ of vertices $\{v''_i\}$, and then, for each selected $v''_{i,k}$, find the closest vertex $\mathbf{v}_{j,c(k)}$ from FR_j so that the distance between them becomes the smallest, and then calculate the distance $e_{ij,k}$ between them. Here $c(k)$ is the index of the closest vertex $\mathbf{v}_{j,c(k)}$ in FR_j corresponding to $v''_{i,k}$. We usually set $\alpha = 5.0$. Finally, calculate matching error $\bar{e}_{ij}^{pca}$ as the averaged distance. If $\bar{e}_{ij}^{pca}$ is less than threshold $\tau^{pca} l_{avg}$, proceed to the next step, otherwise, stop the process. We usually set $\tau^{pca} = 3.0$.

Step3. Perform ICP matching and evaluate error:

Apply the ICP algorithm to match $\langle FR_i, FR_j \rangle$ each other and calculate matching error $\bar{e}_{ij}^{icp}$ using the same manner mentioned in step2. (The detail of the ICP algorithm is mentioned in section 4.1.) If $\bar{e}_{ij}^{icp}$ is less than the threshold $\tau^{icp} l_{avg}$, create the seed create feature regions pair as $\langle FR_i^{seed}, FR_j^{seed} \rangle$, otherwise, stop the process. We usually set $\tau^{icp} = 1.0$.

In cases of detecting reflective symmetries, we deal with pairs $\langle FR_i^{ref}, FR_j \rangle$, where FR_i^{ref} is a mirror image of FR_i. Since they can be processed in the same method as $\langle FR_i, FR_j \rangle$, we only discuss $\langle FR_i, FR_j \rangle$ in the next section.

4. DETECTION OF EUCLIDEAN SYMMETRIES

The final step of our method detects Euclidean symmetries using the combinatorial algorithm of the ICP [14,15] and the region growing algorithms [16].

4.1. Iterative closest point algorithm

The *iterative closest point* (*ICP*) *algorithm* was first proposed by Besl and McKay [14] and Chen and Medioni [15] for matching a pair of scanned data $\langle X, Y \rangle$, and many variants have been proposed [17]. In the ICP algorithm, the optimal transformation can be found where the matching error between corresponding points is minimized. If we denote such a transformation $\mathbf{x}'_i = \mathbf{t}_{XY} + \mathbf{R}_{XY}\mathbf{x}_i$ as $\mathbf{T}_{XY} = \langle \mathbf{t}_{XY}, \mathbf{R}_{XY} \rangle$, where $\mathbf{x}_i \in X$, it can be calculated by the following procedures:

1. **Initialize:** Compute the initial transformation $\mathbf{T}_{XY}^{init}$, and set $itr = 0$.
2. **Find closest points:** For each point $\mathbf{x}_i^{itr}$ in the current position of the data *X*, find the point $\mathbf{y}_{c(i)}^0$ in *Y* that is closest to $\mathbf{x}_i^{itr}$.
3. **Compute a transformation:** Compute the transformation $\mathbf{T}_{XY}^{itr}$ so that the sum of matching errors between corresponding points is minimized. The objective function to be minimized is:

$$F^{itr} = \sum_{i=1}^{N} \left\| \mathbf{y}_{c(i)}^0 - \mathbf{x}_i^{itr+1} \right\|^2. \quad (4)$$

4. **Update points and calculate an error:** Update points using the current transformation $\mathbf{T}_{XY}^{itr}$ such that $\mathbf{x}_i^{itr+1} = \mathbf{t}_{XY}^{itr} + \mathbf{R}_{XY}^{itr}\mathbf{x}_i^0$, then calculate the current average matching error as $E^{itr} = (F^{itr})^{1/2}/N$.
5. **Termination condition:** If $E^{itr} - E^{itr+1} > \varepsilon$, update $itr \leftarrow itr + 1$ and repeat the process from Step 2. Otherwise, output the current transformation $\mathbf{T}_{XY}^{itr}$ as the optimal one $\mathbf{T}_{XY}^{opt}$, and then stop the process.

4.2. Region growing algorithm for segmentation

The *region growing algorithm* for segmentation was first proposed by Besl and Jain [16], and many variants have been proposed [18-23]. This algorithm extracts a set of regions, each of which can be approximated by a surface. This algorithm first creates a set of seed regions $\{R_i^{seed}\}$, and then, for each seed region R_i^{seed}, iterates the following processes:

1. **Initialize:** Initialize $R_i^0 = R_i^{seed}$ and fit the initial surface f_i^0 to R_i^0. Set $itr = 0$.
2. **Fit a surface:** Fit a surface f_i^{itr} to R_i^{itr}.
3. **Add vertices:** Calculate the distance between the surface f_i^{itr} and the vertex x_j that is topologically connected to R_i^{seed} and does not belong to any region. Add x_j to R_i^{seed} if the distance is below the threshold. Continue this process until no more vertices satisfy the condition, and finally extract the grown region R_i^{itr+1}.
4. **Terminate condition:** If $|R^{itr+1}| \geq |R^{itr}|$, update $itr \leftarrow itr + 1$ and repeat the process from Step 2. Otherwise, stop the process and extract the final region R_i^{itr} and the surface f_i^{itr}. Here $|R^{itr}|$ is the number of the vertices in R^{itr}.

4.3. Our region growing for Euclidean symmetry detection

We propose a new combinatorial algorithm of the ICP and region growing algorithms for detecting Euclidean symmetries. Compared with the region growing algorithm for segmentation, our region growing algorithm iterates computing a transformation by the ICP algorithm and performing matching, and adding pairs of vertices.

Our combinatorial algorithm uses extracted pairs of seed feature regions $SFRP_i = \langle FR_{i1}^{seed}, FR_{i2}^{seed} \rangle$ as initial Euclidean symmetric pairs $\langle SR_{i1}^0 = \{v_{i1,k}^0\}, SR_{i2}^0 = \{v_{i2,l}^0\} \rangle$. Then, for each pair $\langle SR_{i1}^0, SR_{i2}^0 \rangle$, it iteratively grows SR_{i1}^0 and SR_{i2}^0 and extracts the final Euclidean symmetric pairs and their transformation according to the following processes:

1. **Initialize:** For each pair $\langle SR_{i1}^0, SR_{i2}^0 \rangle$, compute an initial transformation $\langle \mathbf{t}_i^{init}, \mathbf{R}_i^{init} \rangle$ using PCA matching as mentioned in section 3.2 and then perform initial matching

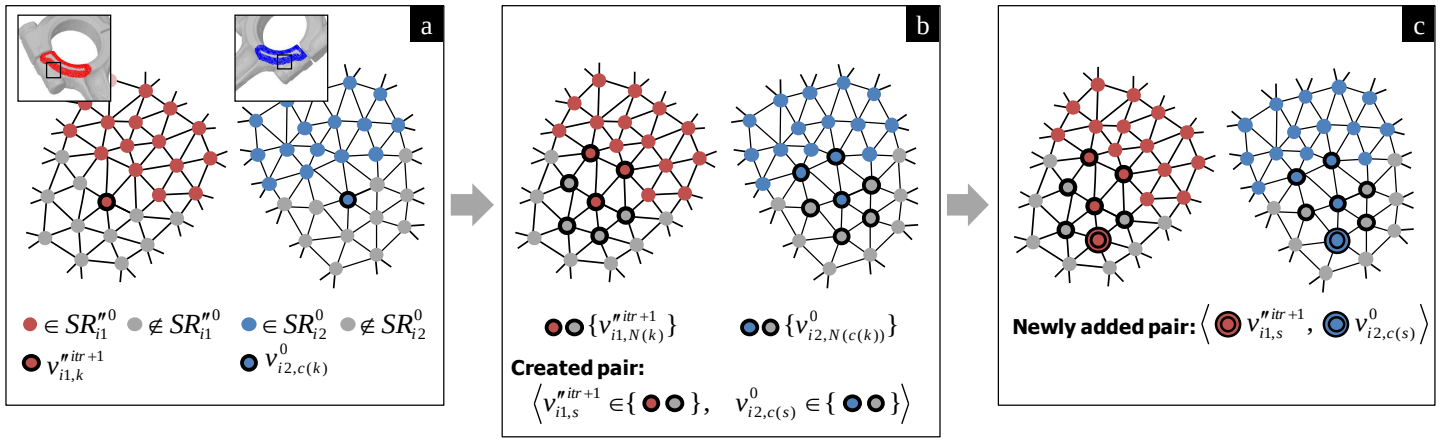


Fig.5: An example of our region growing

by transforming $\mathbf{v}^{0}_{i1,k}$ to $\mathbf{v}''^{0}_{i1,k}$. Set $itr = 0$.

2. **Compute a transformation by the ICP algorithm and perform matching:** Compute a transformation $\langle \mathbf{t}_i^{itr}, \mathbf{R}_i^{itr} \rangle$ that matches SR''^{itr}_{i1} to SR^{0}_{i2} based on the ICP algorithm, and then perform matching by transforming $\mathbf{v}''^{0}_{i1,k} \in SR^{0}_{i1}$ such that $\mathbf{v}''^{itr+1}_{i1,k} = \mathbf{t}_i^{itr} + \mathbf{R}_i^{itr}\mathbf{v}''^{0}_{i1,k}$.
3. **Add pairs of vertices:** An example of this step is shown in Fig.5. Here $\mathbf{v}''^{itr+1}_{i1,k}$ and $\mathbf{v}^{0}_{i2,c(k)}$ are in the seed feature regions, and $\mathbf{v}''^{itr+1}_{i1,k}$ are matched to $\mathbf{v}^{0}_{i2,c(k)}$ under the transformation $\langle \mathbf{t}_i^{itr}, \mathbf{R}_i^{itr} \rangle$ (Fig.5-a). Then create new all possible pairs $\langle \mathbf{v}''^{itr+1}_{i1,s}, \mathbf{v}^{0}_{i2,c(s)} \rangle$ each of which is selected from their 1-ring vertices $\{\mathbf{v}''^{itr+1}_{i1,N(k)}\}$ and $\{\mathbf{v}^{0}_{i2,N(c(k))}\}$, including $\mathbf{v}''^{itr+1}_{i1,k}$ and $\mathbf{v}^{0}_{i2,c(k)}$ respectively (Fig.5-b). (In the case of Fig.5, 48 pairs are created.) Compute the distance $e_{i,s}$ between the pair of vertices. If $e_{i,s}$ is below the threshold $\tau_{add} l_{avg}$, add $\mathbf{v}''^{itr+1}_{i1,s}$ to SR''^{0}_{i1} and $\mathbf{v}^{0}_{i2,c(s)}$ to SR^{0}_{i2} respectively (Fig.5-c). Continue this process until no more pairs satisfy the condition. Finally, extract the grown regions SR''^{itr+1}_{i1} and SR^{itr+1}_{i2}. We usually set $\tau_{add} = 1.0$.
4. **Terminate condition:** If $\left|SR''^{itr+1}_{i1}\right| > \left|SR''^{itr}_{i1}\right|$, update $itr \leftarrow itr + 1$ and repeat the process from step 2. Otherwise, extract $\langle SR^{itr}_{i1}, SR^{itr}_{i2} \rangle$ as the final pair of Euclidean symmetric regions and their transformation $\langle \mathbf{t}_i^{itr}, \mathbf{R}_i^{itr} \rangle$, and then stop the process.

4.4. Efficient search of closest points

Our algorithm repeats the ICP algorithm in the iterations of region growing. The ICP algorithm itself is a repetitive process. Therefore, it seems computationally expensive to repeat the ICP algorithm. In this ICP algorithm, most of the computational time is consumed finding closest points [17].

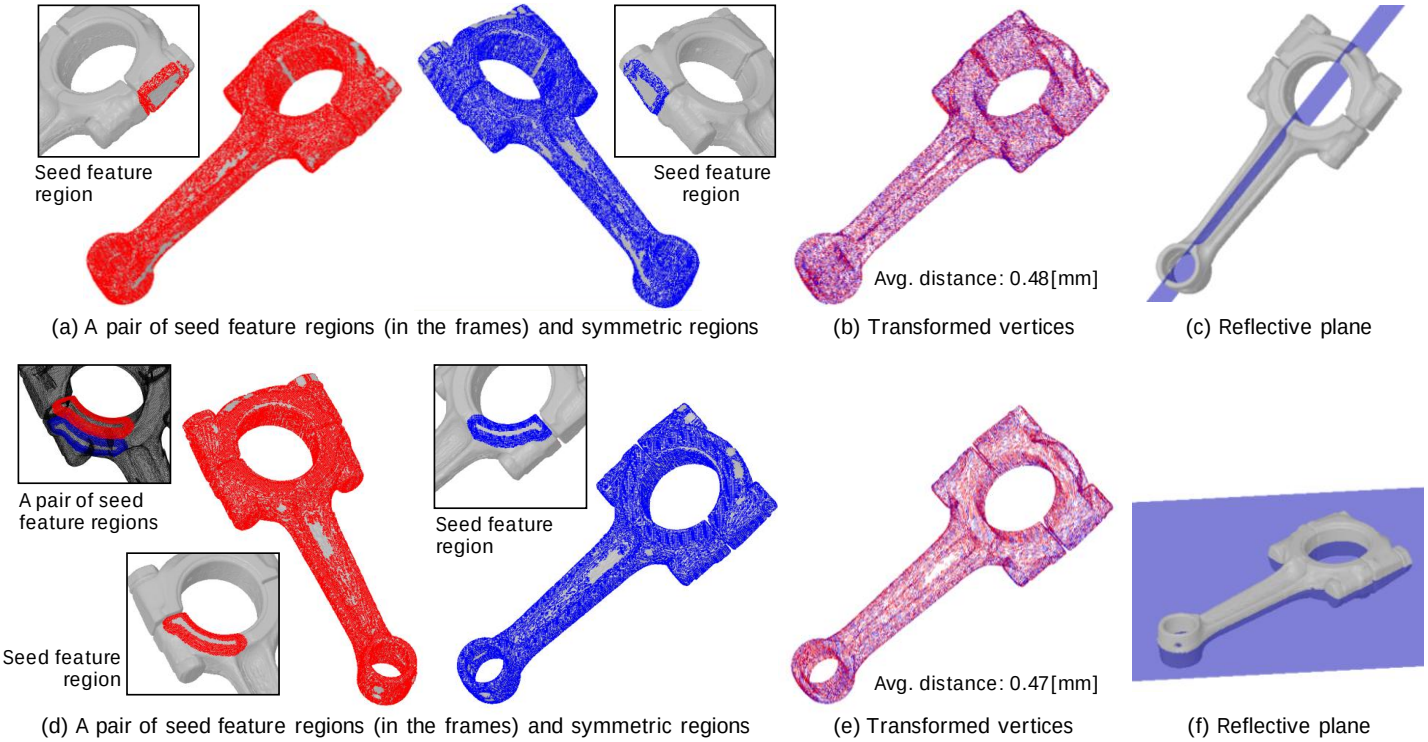
However, in our method, it can be efficiently performed. Since each vertex $\mathbf{x}_i$ keeps its closest vertex $\mathbf{y}_{c(i)}$ during the iteration of the region growing, the new closest vertex $\mathbf{y}^{new}_{c(i)}$ can be searched from $\mathbf{y}_{c(i)}$ using the steepest-descent algorithm. This algorithm recursively searches the vertex $\mathbf{y}^{curr}_{k}$ among the 1-ring vertices $\{\mathbf{y}_{k \in N(j)}\}$ of $\mathbf{y}_j$ where the distance between $\mathbf{x}_i$ and $\mathbf{y}^{curr}_{k}$ becomes the smallest. Moreover, our algorithm can use mesh connectivity for this search, so it does not need to construct the additional data structures, such as *k-d* tree, which are commonly used to accelerate the search.

Our algorithm needs full searches to find the closest vertices only at the first iteration of the ICP algorithm and at the first iteration of the region growing algorithm. In this iteration, our method performs the full searches in order to assure that it accurately finds the globally closest vertices. However, pairs of seed feature regions include only about 10,000 vertices at maximum, as in the case of the large scanned mesh in Fig.7. Therefore, even the full searches for these pairs do not require a long computational time.

5. RESULTS

We applied our proposed algorithm to the X-ray CT scanned meshes on real-world engineering objects. All the experiments were processed on a 2.4 GHz Core 2 Duo CPU.

Fig.6 shows some examples of results for the scanned mesh of the connecting rod, which is also shown in Fig.1 and Fig.4. The mesh contains 123,427 vertices and the averaged edge length is 0.98mm. Our method detected 15 feature regions from the mesh. It also detected 32 pairs of symmetric regions and two global reflective symmetries were included in them. Fig.6(a) and Fig.6(d) show that pairs of seed feature regions are appropriately grown and that the pairs of reflective symmetric regions are detected. Fig.6(b) and Fig.6(e) show the transformation result according to the extracted transformation. We evaluated the averaged distances E^{itr} between closest vertices at the end of the region growing, and they were 0.48mm and 0.47mm respectively. Fig.6(c) and Fig.6(f) show the detected reflective planes, which are calculated by fitting the least-squares planes for sets of mid-points between the vertices and their closest points. We evaluated the angle between the normal vectors of the two planes, and the angle was about 89.7 degree. This shows that our method can accurately and robustly extract a transformation under which a region is closely matched to the other. The running time was about 98 seconds.



(a) A pair of seed feature regions (in the frames) and symmetric regions (b) Transformed vertices (c) Reflective plane

(d) A pair of seed feature regions (in the frames) and symmetric regions (e) Transformed vertices (f) Reflective plane

Fig.6: Result for the X-ray CT scanned mesh of the connecting rod

Fig.7 shows some examples of the results for the scanned mesh in Fig.7(a) of the crankshaft. The mesh contains 487,357 vertices and the averaged edge length is 1.02mm. Our method detected 100 feature regions and 114 pairs of symmetric regions from the mesh. It shows that the pairs of seed feature regions shown in Fig.7(a) and Fig.7(d) are created. Fig.7(b), Fig.7(c), and Fig.7(e) show detected reflective planes and reflective symmetric regions about the planes. The averaged distance E^{itr} was 0.56mm and 0.51mm respectively. Our method could extract symmetries in about 19 minutes from this mesh. This shows that our method can efficiently detect symmetries from large meshes.

Limitations: The limitation of our method is that it fails to detect symmetries when the appropriate pairs of feature regions cannot be detected in step2. An example of this limitation is shown in Fig.8. The mesh seems to include two global reflective symmetries. Here we denote them by S_1 and S_2. From this mesh, our method could detect only one of them S_1 as in Fig.8(a). Since our method cannot detect appropriate feature regions and could not create the pair of seed feature regions for S_2, it failed to detect S_2 illustrated in Fig.8(b). This limitation should be solved in our future work.

6. CONCLUSION AND FUTURE WORKS

In this paper, we proposed an algorithm that detects Euclidean symmetries from scanned meshes of engineering objects. Our method is based on the combination of the ICP and the region growing algorithms. Various experimental results show that our method can robustly and efficiently detect Euclidean symmetries, including translation, rotation, and reflection, from large and noisy scanned meshes. Extracted symmetries then enable to estimate CAD modeling commands and user-defined parameters. And finally, CAD models can be reconstructed that contain compact data representations using detected symmetries.

In future work, we will develop a method to extract pairs of partial regions that do not overlap each other in space from detected pairs of symmetric regions. Then, we will develop a method to estimate CAD commands and their parameters that are originally defined in CAD systems from the detected pairs of regions and transformations.

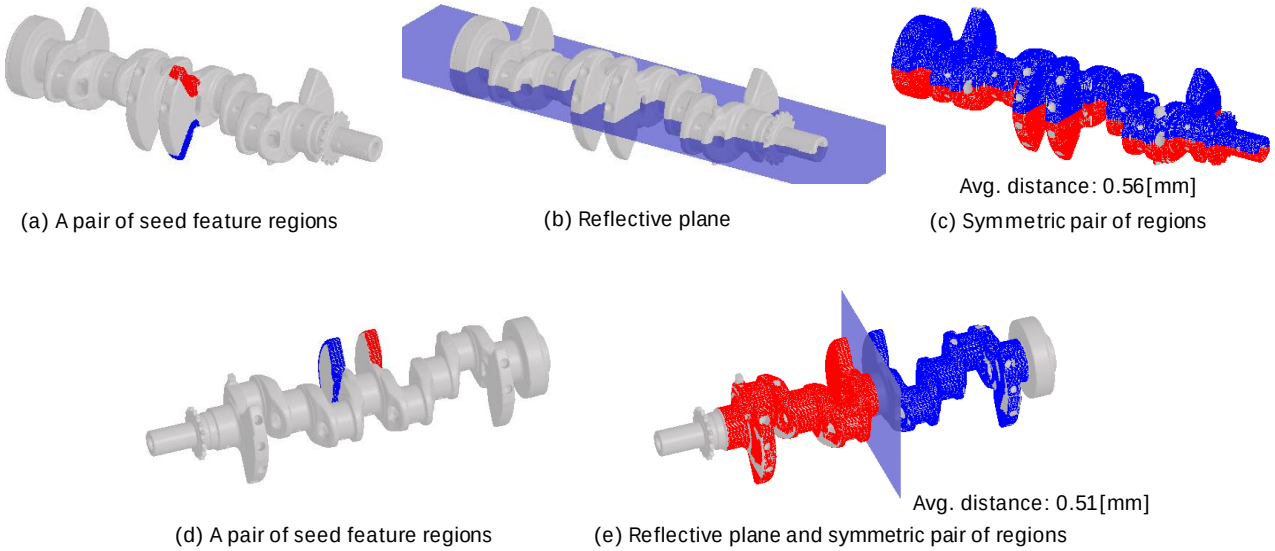
ACKNOWLEDGMENTS

This work was financially supported by the Grant-in-Aid for Scientific Research under the Project No. 18004488-00. We thank Ichiro Nishigaki, Tatsuro Yashiki and Noriyuki Sadaoka in HITACHI Co.Ltd. for providing the X-ray CT scanned meshes in Fig.4, Fig.6, Fig.7 and Fig.8.

REFERENCES

[1] Suzuki, H., 2005, "Convergence engineering based on X-ray CT scanning technologies," Proc. of JSME Digital Engineering Workshop, pp.74-77.

[2] Lorensen, W.E. and Harvey, E.C., 1987, "Marching cubes: A high resolution 3D surface construction algorithm," Proc. of ACM SIGGRAPH, 21(4), pp.163-169.



(a) A pair of seed feature regions (b) Reflective plane (c) Symmetric pair of regions

(d) A pair of seed feature regions (e) Reflective plane and symmetric pair of regions

Fig.7: Result for the X-ray CT scanned mesh of the crankshaft

[3] Benkó, P., Martin, R.R., and Várady, T., 2001, "Algorithms for reverse engineering boundary representation models," Computer-Aided Design, 33(11), pp.839-851.

[4] Várady, T., Facello, M.A., and Terék, Z., 2007, "Automatic extraction of surface structures in digital shape reconstruction," Computer-Aided Design, 39(5), pp.379-388.

[5] Thompson, W.B., Owen, J.C. de St. Germain, H.J., Stark, S.R., and Jr. Henderson, T.C., 1999, "Feature-based reverse engineering of mechanical parts," IEEE Transactions on Robotics and Automation, 15(1), pp.57-66.

[6] Ke, Y., Fan, S., Zhu, W., Li, A., Liu, F., and Shi, X., 2006, "Feature-based reverse modeling strategies," Computer-Aided Design, 38(5), pp.485-506.

[7] Zabrodsky, H., Peleg, S., and Avnir, D., 1995, "Symmetry as a continuous feature," IEEE Transactions on Pattern Analysis and Machine Intelligence, 17(12), 1154–1166.

[8] Loy, G. and Eklundh, J., 2006, "Detecting symmetry and symmetric constellations of features," Proc. of 9th European Conference on Computer Vision, 2, pp. 508-521.

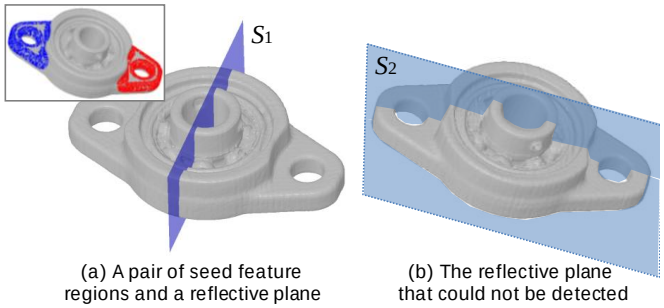
[9] Sun, C. and Sherrah, J., 1997, "3D symmetry detection using the extended Gaussian image," IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(2), pp.164-169.

[10] Martinet, A., Soler, C., Holzschuch, N., and Sillion, F., 2006, "Accurate detection of symmetries in 3D shapes," ACM Transactions on Graphics, 25(2), pp.439-464.

[11] Simari, P., Kalogerakis, E., and Singh, K., 2006, "Folding meshes: Hierarchical mesh segmentation based on planar symmetry," Proc. of the fourth Eurographics symposium on Geometry processing, pp.111-119.

[12] Podolak, J., Shilane, P., Golovinskiy, A., Rusinkiewicz, S., and Funkhouser, T., 2006, "A planar-reflective symmetry transform for 3D shapes," ACM Transactions on Graphics, 25(3), pp.549-559.

[13] Mitra, N.J., Guibas, L.J., and Pauly, M., 2006, "Partial and approximate symmetry detection for 3D geometry," ACM Transactions on Graphics, 25(3), pp.560-568.



(a) A pair of seed feature regions and a reflective plane (b) The reflective plane that could not be detected

Fig.8: Result for the X-ray CT scanned mesh of the bearing

[14] Besl, P. and McKay, N., 1992, "A method for registration of 3-D shapes," IEEE Transactions on Pattern Analysis and Machine Intelligence, 14(2), pp.239-256.

[15] Chen, Y. and Medioni, G., 1992, "Object modeling by registration of multiple range images," Image and Vision Computing, 10(3), pp.145-155.

[16] Besl, P.J. and Jain, R.C., 1988, "Segmentation through variable-order surface fitting," IEEE Transactions on Pattern Analysis and Machine Intelligence, 10(2), pp.167-192.

[17] Rusinkiewicz, S. and Levoy, M., 2001, "Efficient variants of the ICP algorithm," Proc. of International Conference on 3D Digital Imaging and Modeling, pp.145-152.

[18] Vieira, M. and Shimada, K., 2005, "Surface mesh segmentation and smooth surface extraction through region growing," Computer-Aided Geometric Design, 22(8), pp.771-792.

[19] Mizoguchi, T., Date, H., Kanai, S., and Kishinami, T., 2007, "Quasi-optimal mesh segmentation via region growing/merging," Proc. of ASME/DETC, No.35171.

[20] Sapidis, N.S., and Besl, P.J., 1995, "Direct construction of polynomial surfaces from dense range images through region growing," ACM Transactions on Graphics, 14(2), pp.171-200.

[21] Fitzgibbon, A.W., Eggert, D.W., and Fisher, R.B., 1997, "High-level CAD model acquisition from range images," Computer-Aided Design, 29(4), pp.321-330.
[22] Vieira, M. and Shimada, K., 2005, "Surface extraction from point-sampled data through region growing," International Journal of CAD/CAM, 5(1), pp.61-75.
[23] Mizoguchi, T., Date, H., Kanai, S., and Kishinami, T., 2006, "Segmentation of scanned mesh into analytic surfaces based on robust curvature estimation and region growing," Proc. of Geometric Modeling and Processing, pp.644-654.