

Regular Paper

A Matching Method for 3D CAD Models with Different Assembly Structures Using Projections of Weighted Components

KAORU KATAYAMA^{1,a)} TAKUMI SATO²

Received: August 11, 2016, Accepted: February 9, 2017

Abstract: We present a matching method for 3D CAD assembly models consisting of multiple components. Our method discriminates not only the global shapes of the models, the numbers and kinds of their components but also the geometric layouts of the components. In order to identify the components constituting an assembly model, different numerical values such as positive integers are assigned initially to them. The same value is assigned to the same kind of components in an assembly model. However, these initially assigned values to the components vary with assembly models as often happens in practical applications. We represent an assembly model as a set of feature quantities which are computed using projections for each of the components from various angles. The similarity between two assembly models is computed from the similarities between their feature quantities. In order to make the projections reflect the layout of components in the whole assembly structure, we propose a way of reassigning numerical values to the components. This reassignment also makes the feature quantities of assembly models independent of the initially assigned values to their components. Using 3D CAD assembly models with different layouts of components, we show the effectiveness of the proposed method experimentally.

Keywords: 3D CAD model, assembly, projection, matching

1. Introduction

Computer Aided Design (CAD) systems are used recently to manufacture products in many industries such as electronics, automobile and aerospace. New or improved products can be developed efficiently by reusing the existing CAD models. This requires a method for finding CAD models suitable for such reuse from many existing models. In this paper, we present a matching method for 3D CAD assembly models consisting of multiple components. We want to distinguish assembly models by not only the global shapes of the models and, the shapes and numbers of the components but also the geometric layouts of the components.

In this paper, we consider two kinds of differences in layouts of components, that is, layouts of components with the same shapes and layouts of *subassemblies* each of which consists of multiple components. **Figure 1** shows examples of such assembly models with different layouts of components. These models have the same shape, and consist of the same kinds and numbers of components. Here we focus on four kinds of components colored blue, red, green and yellow. The other components are common among all models. Blue components and red ones have the same shapes and constitute a subassembly of the model. Green components and yellow ones also have the same shapes and constitute

another subassembly of the model. However, blue components and green ones have similar but different shapes. The above two subassemblies also have different shapes. **Figure 2** illustrates the relation among an assembly model, subassemblies, and components. In Model A and Model B of Fig. 1, the layout of blue components and red ones is different but the layout of green ones and yellow ones is the same. In Model A and Model C, the layout of green components and yellow ones is the same. In addition, the layout of blue ones and red ones is also the same since the layout in Model C is obtained by rotating the layout in Model A by

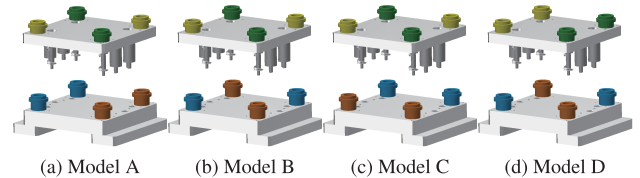


Fig. 1 3D CAD assembly models with the same shapes but different layout of components.

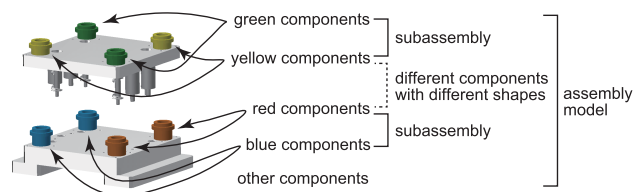


Fig. 2 3D CAD assembly model consisting of multiple subassemblies and components.

¹ Graduate School of System Design, Tokyo Metropolitan University, Hino, Tokyo 191-0065, Japan

² NEC Corporation, Minato, Tokyo 108-8001, Japan

^{a)} kaoru@tmu.ac.jp

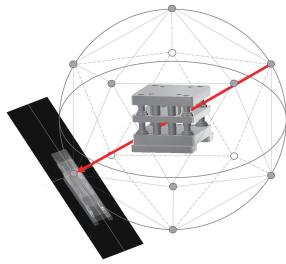


Fig. 3 Projection of 3D CAD assembly model to 2D plane.

90 degrees. However, the layout of the subassembly consisting of blue components and red ones, and the subassembly consisting of green ones and yellow ones is different. In Model A and Model D, every layout of the components colored blue, red, green and yellow is different. Although several methods for searching 3D CAD assembly models are proposed as described in Section 2, few works focus on this problem.

We represent an assembly model as a 3D array. Designers of assembly models usually give a name or an identifier to each of the components. According to the names or the identifiers, we initially assign numerical values such as positive integers to the elements of the 3D array, which correspond to each component. In an assembly model, the same kind of components have the same name or the same identifiers. However, since such names or identifiers of the components may be different depending on designers of assembly models, we assume that initially assigned numerical values to the components differ among assembly models.

For matching 3D CAD assembly models, we take a similar approach to the view-based method which is one of the most effective ways for matching 3D shape models [1]. A set of feature quantities for an assembly model is computed from a set of its projections to 2D planes as shown in Fig. 3. In order to specify the projection angles, we use vertices of a geodesic dome constructed by triangulation of a regular icosahedron. In order to obtain the feature quantities which are more tolerant to the 3D translation and rotation of the model, we apply the 2D Radon transform and the Fourier transform to the projections. However, these transformations do not make the feature set completely tolerant to 3D rotation. We call the feature quantity computed from a projection of an assembly model and a set of the feature quantities a *feature* and a *feature set*, respectively. We represent a feature as a vector in Euclidean space and an assembly model as a set of vectors. The similarity between two assembly models is computed by comparing their feature sets based on the nearest-neighbor method.

Since the initial numerical values assigned to components are different in each of the assembly models, we cannot identify the common components among assembly models by the assigned values. The feature sets computed using the initial values cannot be used to compare assembly models. In addition, our preliminary experiments show that it is difficult to discriminate distinctly differences in assembly structures of the models as described above by simple ways such as reassigning positive integers to components successively from one. In order to solve these problems, we propose a way of reassigning numerical values to components based on their “volumes” and computing the similarities

of assembly models by finding the correspondences between their components. We evaluate experimentally the proposed method using three kinds of 3D CAD assembly models with different layouts of components. Since 3D models in commonly used data set such as Princeton Shape Benchmark [2] and Engineering Shape Benchmark [3] do not contain assembly structures of the models, we cannot use them for evaluating our method. Therefore, we prepare a data set for evaluation by modifying 3D CAD models downloaded from the web [4] by ourselves. The results show the effectiveness of the proposed method.

2. Related Work

For rigid or non-rigid 3D models, a large number of methods to represent their features and retrieve them have been proposed as mentioned in Iyer et al. [5], Bustos et al. [6], Tangelder et al. [1], Lian et al. [7] and Li et al. [8]. Since 3D CAD assembly models are typical rigid 3D models, we can use the methods for rigid models, which are introduced in the above surveys, to deal with components constituting an assembly model. However, there are not many studies on the representation of assembly structures and the retrieval based on them, which is our focus.

Chen et al. [9] represent a multilevel semantical and geometrical structure of an assembly model as a hierarchical graph called “assembly descriptor.” A vertex of the graph corresponds to a component or a subassembly of the model. Two vertices are connected by an edge when there is a relation called “assembly interface” between the two corresponding components. The “part-of” relation between a (sub)assembly and its components is represented as the hierarchical structure of the graph. The similarity of two assembly models is computed by comparing their assembly descriptors using a graph matching algorithm. Then the shapes and layouts of their components are also compared. The shape distribution vector [10] is used as the feature of the shape of a component. The layout of components is represented as a spatial structure composed of line segments in the 3D space, which is called “assembly-bone.” Each line segment connects the geometric-centers of two components when there is an assembly interface between the two components. In comparing the two layouts of components, only the spatial angles between the line segments constituting their assembly-bones are used. Since differences in the lengths of the line segments are not considered in comparison and the geometric-center of a component does not change by its rotation, there are cases where the differences in layouts of components are not discriminated completely by their approach. When the assembly descriptor of the whole assembly model is built, which components constitute a subassembly is decided. Therefore, it is difficult to choose arbitrarily components whose layouts are compared.

Deshmukh et al. [11] and Miura et al. [12] propose a retrieval method for 3D assembly model using a graph matching algorithm. They represent an assembly model as a graph without hierarchy. A vertex of the graph corresponds to a component of the model and has attributes representing the type of the component and so on. Two vertices are connected by an edge when the corresponding components have a relation. The relation is represented as attributes of the edge. In Miura et al. [12], the features of the

shapes of the components are also stored in the corresponding vertices as attributes. They use the “Angle Distance” [13] as the feature of the shape of a component. It is difficult to represent the geometric layouts of the components as mentioned in Section 1 by using the edges of the graph and their attributes.

In Hu et al. [14], an assembly model is regarded as a set of components by merging the identical components. Differences in the layouts of components constituting assembly models are not considered. The similarity of two components is computed from their features based on the Light Field Descriptor [15]. The feature of an assembly model is represented as a n -dimensional vector where n is the number of different components in all assembly models. To include a component in an assembly model, a weighted value is assigned to the corresponding element of the vector. The similarity between two assembly models is computed based on the inner product of the two corresponding vectors. In Wang et al. [16], the feature of an assembly model is defined as a set of feature of each component constituting it. The feature of a component is computed based on shape distribution [10] and is represented as a vector in Euclidean space. The similarity between assembly models is computed using Earth Mover’s Distance [17]. These works do not consider the geometric layouts of components constituting an assembly model.

In our previous work on matching 3D CAD assembly models with different layouts of the components [18], we assume that, across all assembly models, a way of assigning values to the components is the same, and do not consider the case where ways of assigning values to the components are different. Therefore, the definition of the feature of an assembly model and the way to compute the similarity between two assembly models are different. In addition, we propose a way of assigning values for discriminating differences in the assembly structures more distinctly in this paper. We evaluate the proposed method using the 3D CAD models with more diverse types of assembly structures.

3. Matching 3D CAD Assembly Models

3.1 Representation of Assembly Models

We represent every assembly model as a 3D array of the same size which is enough to contain it. If a 3D CAD assembly model is represented in another format, we convert it into a 3D array. The size of the converted model in the array is proportional to the original size. In any array, the same models or the same components have the same size. In order to identify each kind of the components constituting an assembly model, a different numerical value is assigned initially to the corresponding elements of the array. When we distinguish components with the same shape by their other properties such as materials, we regard them as different kinds of components and assign different values to them. Zeros are assigned to the elements of the array, which do not correspond to any component in the model. In each assembly model, the same numerical value is assigned to the same kind of components. However, in multiple assembly models, different values may be assigned to the same kind of components which are commonly contained in all of the models. Since it is not easy to standardize ways of assigning numerical values to components across all assembly models, we assume that the values assigned

to the same components differ depending on assembly models. Therefore, it is necessary that our matching method for assembly models be robust over variation in the values assigned to the components.

3.2 Features for Discriminating Differences in Assembly Structures

We want to find 3D CAD assembly models which not only have the same shapes and are consisted of the same kinds and numbers of components but also have the same layouts of the components as a given assembly model. The location and the orientation of each assembly model in a coordinate system is usually different depending on the model. It is not easy to normalize the orientation when they have symmetric shapes. Therefore, it is desirable that features extracted from assembly models be robust to translation and rotation in 3D space. In order to compute such features, we take a similar approach to the view-based method which is one of the most effective ways for matching 3D shape models [1]. That is, we compute the feature set of an assembly model from its projections to 2D planes. The result of the projection is stored in a 2D array of a specific size. A value of an element of the 2D array is the sum of values assigned to the elements of the 3D array representing the model on a line perpendicular to the projection plane. This projection reflects not only the global shape of the model but also the shapes and the layouts of all the components. The projection is not affected by the distance between the projection plane and the model. In order to choose the projection angles near uniformly on a spherical surface, we specify them by using vertices of a geodesic dome constructed by triangulation of a regular icosahedron. The geodesic dome has a spherical structure but not a hemispherical structure since the results of the projections from the vertices which are symmetric with respect to the center of the sphere cannot be transformed only by 2D translation and rotation as two projections shown in Fig. 4.

If two assembly models are the same, we will have two sets of similar projections of them regardless of any rotations of either model. In order to make the features computed from the projections more robust to 3D rotation and translation of the models, we apply the 2D Radon transform and the Fourier transform to the results of the projections. These transformations do not make the features completely tolerant to 3D rotation and translation. In

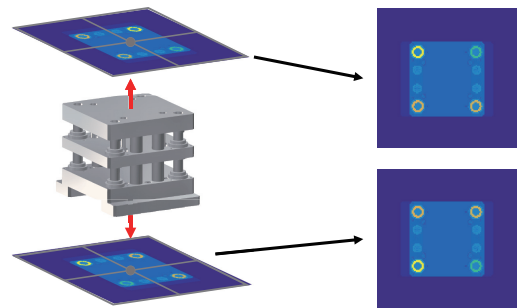


Fig. 4 Projections of 3D CAD assembly model from vertices which are symmetric with respect to center of sphere shown in Fig. 3.

order to reduce the cost of storing the feature sets and computing the similarity of the assembly models, we reduce the size of each feature by deleting high-frequency components from the power spectrum obtained by the Fourier transform.

3.3 Reassignment of Numerical Values to Components

Since ways of assigning numerical values to components may be different depending on assembly models, we cannot use the values directly to compute their feature sets and similarities. As mentioned in Section 1, we also find that differences in assembly structures of the 3D CAD models cannot often be discriminated distinctly by simple ways such as reassigning positive integers to components successively from one. In order to reflect adequately the feature of each kind of components in the whole feature of the assembly model, we reassign numerical values to the components and compute the feature set as follows.

We denote a family of sets of the same kind of components constituting an assembly model m as $C(m)$. That is, $C(m) = \{c_i : c_i \text{ is a set of the same kind of components, } i = 1, \dots, |C(m)|\}$ where $|C(m)|$ is the number of kinds of components. We divide an assembly model m into a set $c \in C(m)$ of the same kind of components and the others $m - c$ which is obtained by removing c from m . We denote the numbers of the corresponding elements to m , c and $m - c$ in the 3D array representation as $Vol(m)$, $Vol(c)$ and $Vol(m - c)$, respectively. We call these numbers *volumes* of m , c and $m - c$. We denote the 3D array representation of m where numerical values are assigned to the elements in the following way as $m[c]$.

- For a set c of the same kind of components, $1/Vol(c)$ is assigned to the corresponding elements of the 3D array
- For the components $m - c$, $1/Vol(m - c)$ is assigned to the corresponding elements of the 3D array
- Zeros are assigned to the other elements of the array

If two assembly models m_1 and m_2 are the same, we assign the same numerical values $1/Vol(c)$ and $1/Vol(m - c)$ to the same c and $m - c$ in m_1 and m_2 , respectively. For each $c \in C(m)$, we construct the 3D array $m[c]$ and compute projections from the vertices of the geodesic dome. The feature set of an assembly model is a set of the features computed from the projections of each of the constructed arrays. By this reassignment, we balance the feature of c and the feature of $m - c$ in the feature set of the whole model. If the value assigned to c is much larger than the value assigned to the other components, the feature set of the model reflects the feature of c more than the other components and the features of the components except for c may be neglected in matching assembly models. Conversely, if the values assigned to the components except for c are much larger than the value assigned to c , the feature of c may be neglected.

For an assembly model m , we compute the features from the projections of $m[c]$ for each $c \in C(m)$ as described in the previous section and make all of the features as the feature set of m . Algorithm 1 shows the procedure to compute the feature set for an assembly model m . The number of the features for m , that is, the cardinality of $F(m, V)$ in Algorithm 1, is $|C(m)| \times |V|$ where V is a set of the vertices of a geodesic dome to specify the projection angles of a 3D array $m[c]$.

Algorithm 1 ComputeFeature(m, V, A)

Require: a 3D CAD assembly model m , a set V of vertices of a geodesic dome to specify the projection angles of a model, a set A of projection angles for the 2D Radon transform

Ensure: the feature set $F(m, V)$ of m

```

1: for all  $v_j \in V$  where  $j = 1 \dots |V|$  do
2:   for all  $c_i \in C(m)$  where  $i = 1 \dots |C(m)|$  do
3:     construct the 3D array  $m[c_i]$ 
4:     compute a projection of  $m[c_i]$  to a 2D plane from the angle specified by  $v_j$ 
5:     store the result in a 2D array  $p(m[c_i], v_j)$ 
6:     for all  $a_k \in A$  where  $k = 1 \dots |A|$  do
7:       compute a projection of  $p(m[c_i], v_j)$  to a line with gradient specified by  $a_k$ 
8:       store the result in the elements of a 2D array  $r(p(m[c_i], v_j))$ , which correspond to  $a_k$ 
9:     end for
10:    compute the Fourier transform of  $r(p(m[c_i], v_j))$  along the radial coordinate
11:    store the power spectrum in a 2D array  $f(r(p(m[c_i], v_j)))$ 
12:    compute the Fourier transform of  $f(r(p(m[c_i], v_j)))$  along the angular coordinate
13:    store low-frequency components of the power spectrum in a 2D array  $f(f(r(p(m[c_i], v_j))))$ 
14:    add  $f(f(r(p(m[c_i], v_j))))$  to a feature set  $F(m, V)$  as an element  $F(m[c_i], v_j)$ 
15:  end for
16: end for
17: return  $F(m, V)$ 

```

3.4 Similarity between Assembly Models

A feature which is an element of the feature set for an assembly model is represented as a 2D array in Algorithm 1. By arranging elements of the 2D array in a row, we regard a feature as a vector in Euclidean space. Then an assembly model is represented as a set of vectors in Euclidean space. We define the dissimilarity of two features as the Euclidean distance between two vectors representing them instead of their similarity. In the following, we use the term *distance* to represent the dissimilarity between features, between feature sets, between components and between assembly models. We want to find the assembly model whose distance from another specified model is the lowest value. We compute the distance between two assembly models m_1 and m_2 using the distances between their components as follows.

Since the assigned numerical value to each of the components constituting m_1 and m_2 is based on its volume as described in Section 3.3, we cannot always identify the same components by only the values. Therefore, we check all possible correspondences between their components this time. We compute the distance between a component c^1 of m_1 and a component c^2 of m_2 using the feature sets computed from the projections of $m_1[c^1]$ and $m_2[c^2]$. For each feature in the feature set of $m_1[c^1]$, we find the closest feature in the feature set of $m_2[c^2]$ by computing their Euclidean distances. We define the distance between c^1 and c^2 by the sum of the Euclidean distances between the pair of the most closest features in the feature sets of $m_1[c^1]$ and $m_2[c^2]$. Algorithm 2 shows the detail of this procedure. $F(m_1[c^1], V)$ and $F(m_2[c^2], V)$ are subsets of $F(m_1, V)$ and $F(m_2, V)$ computed by Algorithm 1,

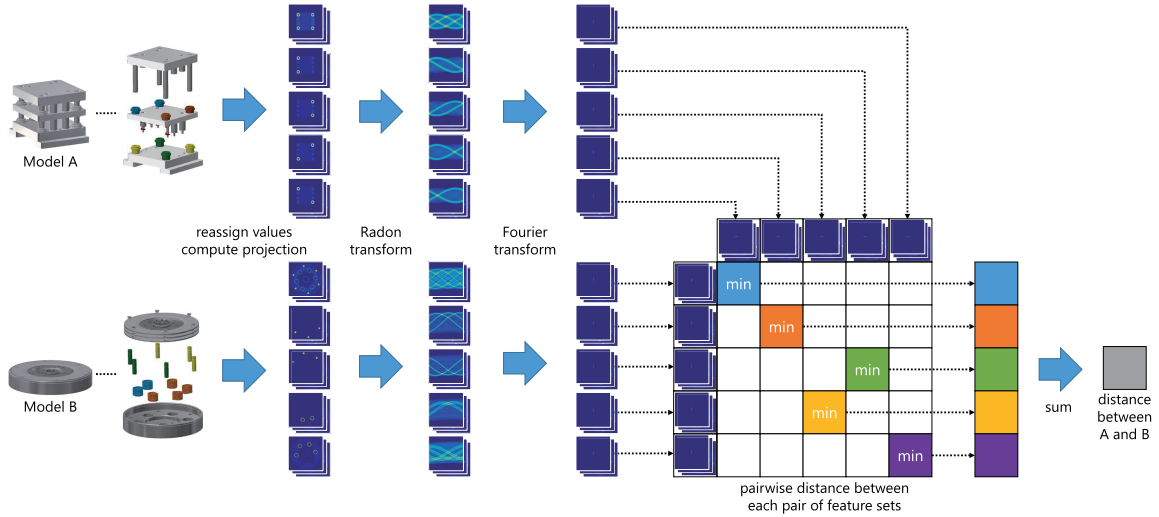


Fig. 5 Procedure to compute features and distances for 3D CAD assembly models.

Algorithm 2 $\text{ComputeComponentDistance}(F(m_1[c_i^1], V), F(m_2[c_j^2], V), V)$

Require: two subsets $F(m_1[c_i^1], V)$ and $F(m_2[c_j^2], V)$ of feature sets $F(m_1, V)$ and $F(m_2, V)$ of assembly models m_1 and m_2 , a set V of vertices of a geodesic dome to specify projection angles of a model

Ensure: distance dis between $m_1[c_i^1]$ and $m_2[c_j^2]$

```

1:  $dis \leftarrow 0$ 
2: for  $k = 1$  to  $|V|$  do
3:    $pd \leftarrow \|F(m_1[c_i^1], v_k) - F(m_2[c_j^2], v_1)\|$  for  $v_1, v_k \in V$ 
4:   for  $l = 2$  to  $|V|$  do
5:      $d \leftarrow \|F(m_1[c_i^1], v_k) - F(m_2[c_j^2], v_l)\|$  for  $v_k, v_l \in V$ 
6:     if  $d < pd$  then
7:        $pd \leftarrow d$ 
8:     end if
9:   end for
10:   $dis \leftarrow dis + pd$ 
11: end for
12: return  $dis$ 

```

which contain only features of $m_1[c_i^1]$ and $m_2[c_j^2]$, respectively. $\|F(m_1[c_i^1], v_i) - F(m_2[c_j^2], v_j)\|$ is an Euclidean distance between two vectors corresponding to $F(m_1[c_i^1], v_i)$ and $F(m_2[c_j^2], v_j)$.

For every correspondence between the components constituting each of two assembly models m_1 and m_2 , we compute the distance between the components in the same way. Then we find the closest pair of the components in all the correspondences. Next, in the correspondences between the components except for the above pair components, we find the closest pair of the components again. By repeating this procedure, we decide the correspondences between the components of m_1 and m_2 . The distance between m_1 and m_2 is the sum of the distances between the corresponding components. Algorithm 3 and Fig. 5 shows the whole procedure for computing the distance of two assembly models m_1 and m_2 . If m_1 and m_2 consist of different numbers of kinds of components, that is, $|C(m_1)| \neq |C(m_2)|$, they are clearly different. Then we do not compute the distance between m_1 and m_2 and ∞ is returned in Algorithm 3.

Algorithm 3 $\text{ComputeDistance}(m_1, m_2, V, A)$

Require: two 3D CAD models m_1, m_2 , a set V of vertices of a geodesic dome to specify the projection angles of a model, a set A of projection angles for the 2D Radon transform

Ensure: distance dis between m_1 and m_2

```

1: if  $|C(m_1)| = |C(m_2)|$  then
2:    $F(m_1, V) \leftarrow \text{ComputeFeature}(m_1, V, A)$ 
3:    $F(m_2, V) \leftarrow \text{ComputeFeature}(m_2, V, A)$ 
4:   for all  $c_i^1 \in C(m_1)$  and  $c_j^2 \in C(m_2)$  do
5:      $cd(i, j) \leftarrow \text{ComputeComponentDistance}(F(m_1[c_i^1], V), F(m_2[c_j^2], V))$ 
6:   end for
7:    $dis \leftarrow 0$ 
8:    $I \leftarrow \{1, 2, \dots, |C(m_1)|\}$ 
9:    $J \leftarrow \{1, 2, \dots, |C(m_2)|\}$ 
10:  while  $I \neq \emptyset$  do
11:    find the minimum value  $cd(i', j')$  in  $cd(i, j)$  for all  $(i, j) \in I \times J$ 
12:     $dis \leftarrow dis + cd(i', j')$ 
13:     $I \leftarrow I \setminus i'$ 
14:     $J \leftarrow J \setminus j'$ 
15:  end while
16:  return  $dis$ 
17: else
18:   return  $\infty$ 
19: end if

```

4. Experimental Evaluation

We evaluate our method to discriminate 3D CAD assembly models with different layouts of components experimentally. We prepare three kinds of 3D CAD assembly models, a clutch, a die, and a gear as shown in Fig. 6. For each kind of assembly model, we also prepare five types, A, B, C, D and E of assembly structures as shown in Fig. 7. All types of a kind of assembly model have the same shape and consist of the same kinds and numbers of components, respectively. We show the same kinds of components in the same colors in Fig. 7. We denote a clutch of type A as Clutch A, for example. For example, Clutch A, Clutch B, Clutch C, Clutch D and Clutch E of Fig. 7 have the same shapes, and consist of three green components, three yellow ones, four red ones, two blue ones and the others. The different components

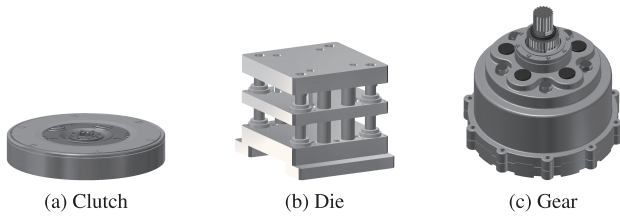


Fig. 6 3D CAD assembly models and the components.

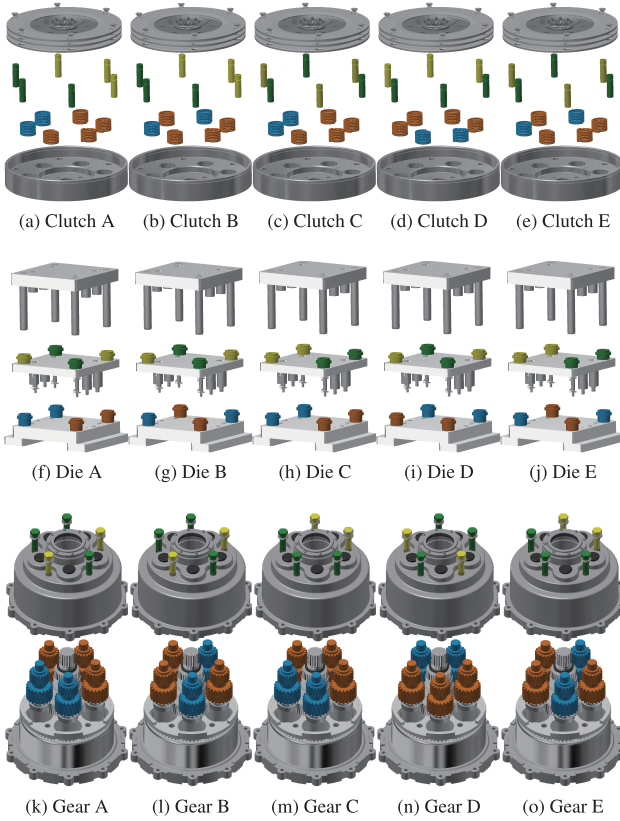


Fig. 7 Five types of assembly structures for three kinds of 3D CAD models.

shown in different colors may have the same shapes. Green components and yellow ones have the same shapes. Blue components and red ones also have the same shapes. Every assembly model consists of three subassemblies each of which consists of multiple components with the same shapes. The first subassembly consists of yellow and green components. The second one consists of blue and red components. We regard the components except these four colored components as the third subassembly.

We call an assembly model given as a query a *query model*, and call assembly models compared with a query model *database models*. We use these fifteen assembly models in Fig. 7 as database models. Each of the database models is rotated and translated randomly. For each kind of database models, we prepare 40 query models as a target for comparison by randomly rotating and translating the type A of models, that is, Clutch A, Die A or Gear A. Table 1 shows the differences in the assembly structures between type A and other types for each subassembly. “Y-G” is a subassembly consisting of yellow and green components. “B-R” is a subassembly consisting of blue and red components. “Y-G & B-R” is a “combined” subassembly consisting of Y-G and B-R. “○” means that Y-G, B-R or Y-G & B-R has

Table 1 Differences in assembly structures between type A and other types for each subassembly.

	type A	type B	type C	type D	type E
Y-G	○	○	×	○	×
B-R	○	×	○	○	×
Y-G & B-R	○	×	×	×	×

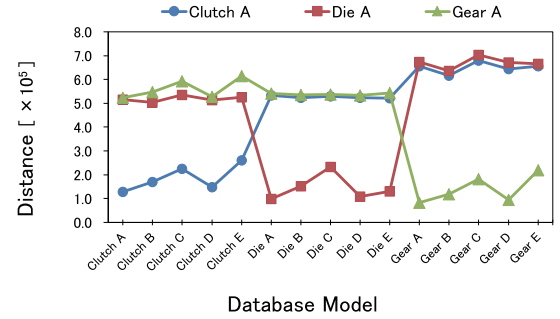


Fig. 8 Distance between fifteen database models and each kind of query model.

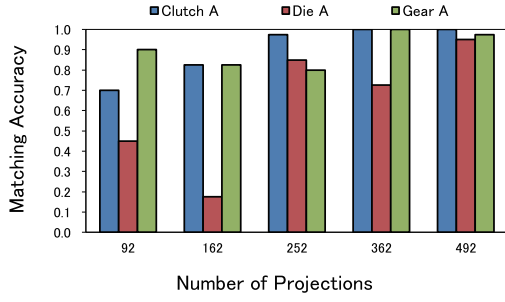
the same layout in a query model and the corresponding type of a database model, while “×” means that it has a different layout. Since query models are of type A, all columns of “type A” are ○. The columns of “type D” show that, in the type D of database models, Y-G and B-R have the same layout as query models but Y-G & B-R has a different layout. In type B of database models, Y-G has the same layout but B-R has a different layout. Therefore, Y-G & B-R also has a different layout.

In computing a feature set of an assembly model, we use the vertices of geodesic domes constructed by triangulation of a regular icosahedron to specify the projection angles of assembly models, V in Algorithm 1, Algorithm 2 and Algorithm 3. The projection angles in the Radon transform, A in Algorithm 1 and Algorithm 3, are from 0 to 179 in steps of 1 degree. Since popular benchmark data such as Princeton Shape Benchmark [2], Engineering Shape Benchmark [3] and ShapeNet [19] do not include 3D assembly models with different layouts of components, we prepare such 3D CAD assembly models for the experimental evaluation of the proposed method. All of the assembly models used in these experiments are obtained from the website [4] and simplified by removing several components from them. We develop all the programs with MATLAB 2015b on 64 bit Windows 10 Enterprise and use a PC with a 3.4 GHz Intel Core i7 processor and 32 GB RAM.

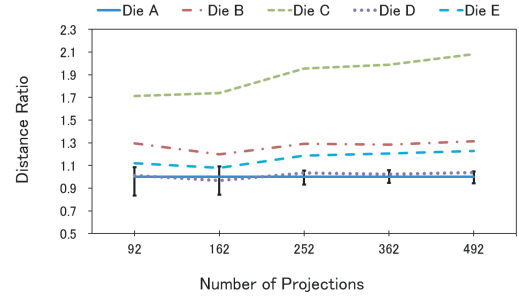
4.1 Discrimination of Differences in Assembly Structures

We evaluate the ability of our method to discriminate not only the difference of assembly models but also the difference of layouts of components in an assembly model. Figure 8 shows the distances of the fifteen database models, which are computed by Algorithm 3, from each kind of the query models, that is, Clutch A, Die A and Gear A. When a value of the distance between a database model and a query model is low, they are similar.

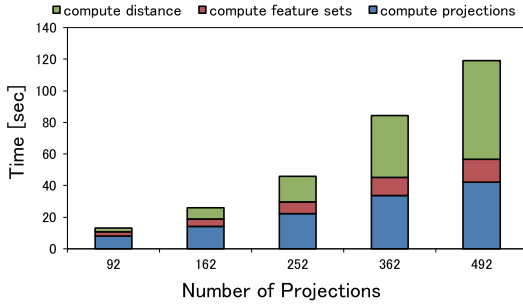
The distance of Clutch A, Die A or Gear A in the database models, which is the same type as the query model, is the lowest value in Fig. 8. Since, as shown in Table 1, type A and type D of the assembly structure are similar, the distance of type A of



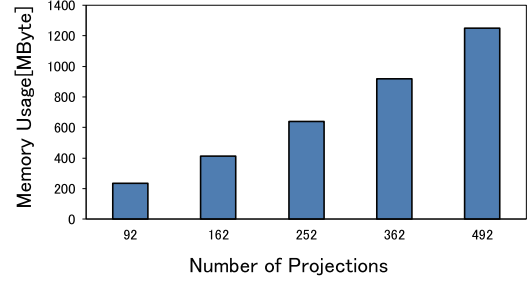
(a) Matching accuracy of all kinds of database models for each kind of query models of type A



(b) Distance ratio of each type of Dies in database models for type A of Dies in query models



(c) Average processing time for computing distance between a database model and a query model



(d) Memory usage in computing distance between a database model and a query model

Fig. 9 Effects of number of projections.

Table 2 Sizes of arrays and input parameters in Algorithm 1.

3D arrays $m, m[c_i]$	$74 \times 74 \times 74$
2D array $p(m[c_i], v_j)$	128×128
2D array $r(p(m[c_i], v_j))$	185×180
2D array $f(r(p(m[c_i], v_j)))$	185×180
2D array $f(f(r(p(m[c_i], v_j))))$ or $F(m[c_i], v_j)$	185×180
Number $ V $ of vertices of a geodesic dome	492
Number $ A $ of projection angle in the Radon transform	180

the database models from the query models is close in value to the distance of type D of the database models. The distances of database models of different kinds from the query models are much larger than those of any type of database models of the same kind as the query models.

Table 2 shows the sizes of the arrays and the input parameters of Algorithm 1 in this experiment. We set the size of the 3D arrays, m and $m[c_i]$, which have the same size, to maintain the shape of each component constituting an assembly model in converting it to the array. The size of the 2D array $p(m[c_i], v_j)$ corresponds to the size of the projection of $m[c_i]$. In the following experiments, we change these sizes or input parameters to explore their effect on the results. Since it is more difficult to discriminate the same kinds but different types of assembly models, we show only the results of such cases.

4.2 Effects of Number of Projections

The ability to discriminate the differences of assembly models and the computational cost are affected by the number of projections used to compute their feature sets. Here we explore the effects of the number of projections for an assembly model. We vary the number of projections, that is, the number of vertices V of a geodesic dome to specify the projection angles in Algo-

rithm 1 and Algorithm 3. The sizes of the arrays and the input parameters except for V in the algorithms are the same as those in Table 2.

Figure 9 (a) shows the proportion of the query models each of which is of the same kind and type as its closest database model to it with varying the number $|V|$ of projections from 92 to 492 for three kinds of the query models. That is, for each kind of database models D , the proportion is

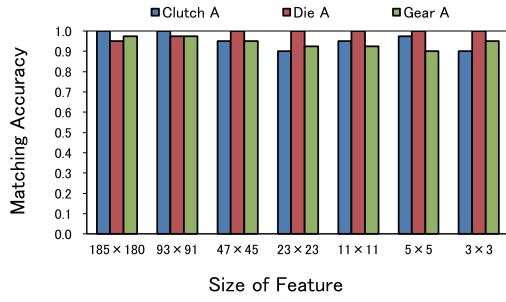
$$\frac{\left| \left\{ q \in Q : \begin{array}{l} \text{the most closest model in } D \text{ to } q \text{ is of} \\ \text{the same kind and type as } q. \end{array} \right\} \right|}{|Q|} \quad (1)$$

where Q is a set of query models of the same kind as D and $|Q| = 40$ is the cardinality of Q . We refer to this proportion as *matching accuracy*. Although there are several exceptions, the matching accuracy increases on the whole as $|V|$ increases. When $|V|$ is equal to 492, the matching accuracy is more than 0.9 for every kind of the database models.

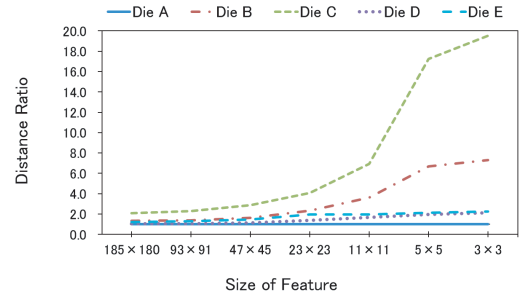
Figure 9 (b) shows one way of illustrating the change in the distances between each type of die in the database models and Die A's in the query models with varying the number of projections. We denote the distance of a database model d to a query model q as $dis(d, q)$. A value for each number of projections in the figure is the proportion based on the distances between Die A in the database models and Die A in the query models, that is,

$$\frac{\sum_{q \in Q} dis(d, q)}{\sum_{q \in Q} dis(\text{Die A}, q)} \quad (2)$$

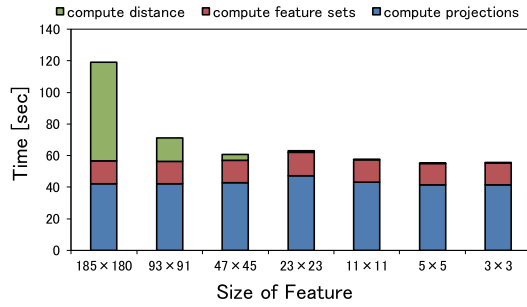
where d is a type of die in the database models and Q is the set of Die A's in the query models. We refer to this proportion as *distance ratio* for Die A. When the difference of the distance ratios



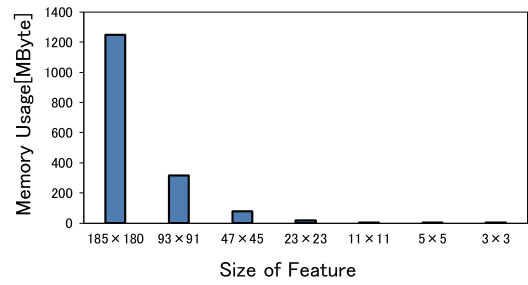
(a) Matching accuracy of all kinds of database models for each kind of query models of type A



(b) Distance ratio of each type of Dies in database models for type A of Dies in query models



(c) Average processing time for computing distance between a database model and a query model



(d) Memory usage in computing distance between a database model and a query model

Fig. 10 Effects of size of feature.

for two types of database models is large, their discrimination is not difficult. Since Die A and Die D have similar assembly structures, the distance ratio for Die D is close to the distance ratio for Die A. The error bars in the distance ratios for Die A indicate the range of the distance ratio for each of Die A's in the query models. The maximum and the minimum are

$$\max_{q \in Q} \frac{dis(d, q)}{\sum_{q \in Q} dis(Die A, q) / |Q|} \quad (3)$$

and

$$\min_{q \in Q} \frac{dis(d, q)}{\sum_{q \in Q} dis(Die A, q) / |Q|} \quad (4)$$

where the notations are the same as above, respectively. As the number of projections increases, the range of the error bars in Fig. 9 (b) becomes narrow and the matching accuracy in Fig. 9 (a) increases.

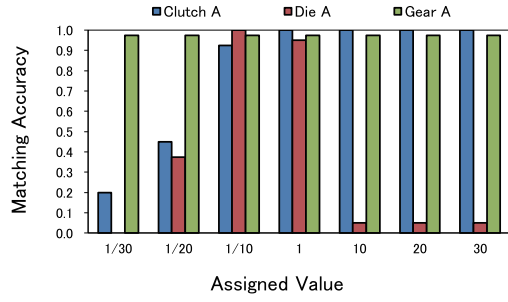
Figure 9 (c) shows the average processing time for computing the distance between two assembly models by Algorithm 3. “compute projections” in the figure corresponds to the time to compute the projections of two assembly models in step 3 of Algorithm 1. “compute feature sets” corresponds to the time to compute their feature sets in the other steps of Algorithm 1. “compute distance” corresponds to the time to compute their distance by Algorithm 3 except for steps 2 and 3 where Algorithm 1 is invoked. Both “compute projections” and “compute distance” increase largely with increase in the number of projections. Figure 9 (d) shows the memory usage in computing the distance between a database model and a query model. The memory usage increases with increase in the number of projections as expected. As shown in the next experiment, the processing time and the memory usage can be decreased without making a large effect on

the matching accuracy by reducing the size of features in a feature set.

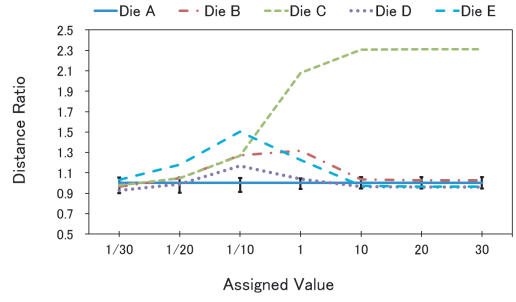
4.3 Effects of Size of Feature

We use the power spectrum computed from a projection of an assembly model as a feature in the feature set of an assembly model. By deleting low-frequency components from the power spectrum, we reduce the size of the feature which is represented by the 2D array $F(m[c_i], v_j)$ or $f(f(r(p(m[c_i], a_j))))$ in Algorithm 1. The sizes of arrays except for $F(m[c_i], v_j)$ and the input parameters in the algorithms are the same as those shown in Table 2.

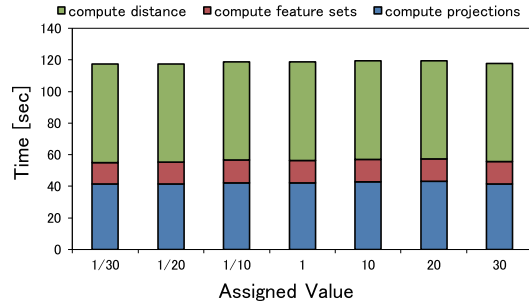
Figure 10 (a) shows the matching accuracy defined by formula (1) with decreasing the size of array $F(m[c_i], v_j)$ from 180×180 to 3×3 for three kinds of the query models. It decreases slowly as the size of the array decreases. Even when the size of the array is 3×3 , the matching accuracy is high. Figure 10 (b) shows the distance ratio defined by formula (2) with decreasing size of the array. The error bars in the distance ratios for Die A indicate the range between the maximum and the minimum which are defined by formulae (3) and (4), respectively. Regardless of the size of the array, the range of the error bar in Fig. 10 (b) is narrow. Figure 10 (c) shows the average processing time for computing the distance between two assembly models. “compute distance” decreases rapidly as the size of the array decreases. “compute projections” and “compute feature sets” are not affected by the size of feature. Figure 10 (d) shows the memory usage in computing the distance between a database model and a query model. The memory usage decreases with decrease in the size of a feature in a feature set as expected.



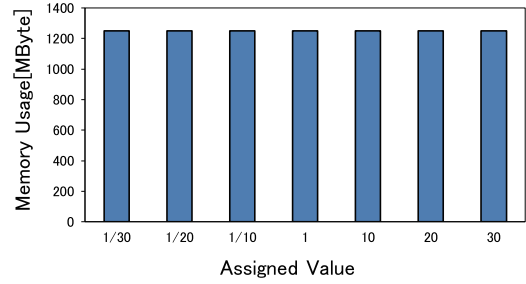
(a) Matching accuracy of all kinds of database models for each kind of query models of type A



(b) Distance ratio of each type of Dies in database models for type A of Dies in query models



(c) Average processing time for computing distance between a database model and a query model



(d) Memory usage in computing distance between a database model and a query model

Fig. 11 Effects of assigned values to components in computing features.

4.4 Effects of Assigning Values to Components

In computing a feature set of an assembly model m by Algorithm 1, we construct the 3D array $m[c]$ for each component c of m by assigning values to elements of the array as described in Section 3.3. Since the feature set is affected by the assigned values, we examine the effect on discriminating differences in assembly structures of 3D CAD models. In this experiment, we assign several values in addition to $1/V(c)$ to the corresponding elements of the array to c in $m[c]$. On the other hand, to the corresponding elements to all components except c , we always assign $1/V(m - c)$. The sizes of arrays and the input parameters in the algorithms are the same as those shown in Table 2.

Figure 11 (a) shows the matching accuracy with varying the assigned values from $1/V(c) \times 1/30$ to $1/V(c) \times 30$ to the corresponding elements in $m[c]$ to c for the three kinds of query models. “1/30” in this figure means that $1/V(c) \times 1/30$ is assigned to the elements corresponding to c , for example. Although the effect of the assigned values depends on the kinds of the assembly models, the matching accuracy is higher for all their kinds when we assign $1/V(c) \times 1/10$ or $1/V(c)$ to the elements corresponding to c in $m[c]$. As the assigned value increases, the matching accuracy increases for the clutches and it does not change largely for the gears. When the assigned value is large or small, the matching accuracy is low for the dies.

Figure 11 (b) shows the distance ratio with varying the assigned values from $1/V(c) \times 1/30$ to $1/V(c) \times 30$ to the corresponding elements in $m[c]$ to c . The error bars in the distance ratio for Die A indicate the range between the maximum and the minimum which are defined by formulae (3) and (4), respectively. The range of the error bar in Fig. 11 (b) does not change largely with varying as-

signed values. When we assign $1/V(c) \times 1/10$ or $1/V(c)$ to the corresponding elements to c in $m[c]$, the difference of the distances among each type of dies is larger. Figure 11 (c) shows the average processing time for computing the distance between two assembly models. The processing time is not influenced by the assigned values as expected. Figure 11 (d) shows the memory usage in computing the distance between a database model and a query model. The memory usage does not depend on assigned values.

5. Conclusion

We present a matching method for 3D CAD assembly models using their projections. It discriminates not only their global shapes, the numbers and kinds of their components but also the geometric layouts of the components. We also propose a way of reassigning values to the components in order to discriminate the differences in their assembly structures more clearly. Since the feature sets and similarities of assembly models do not depend on the initial values assigned to the components, the proposed method can be used in the case where identifiers of the components are different across assembly models. In order to evaluate the proposed method, we prepare the three kinds of database models and query models which are randomly rotated and translated. For each kind of database models, we also prepare the five types of layouts of the components. The results of the experiments show that the differences in the prepared assembly structures can be discriminated with a high degree of accuracy by the proposed method. It remains to reduce the processing time, to improve the discrimination power for larger or more complicated assembly models and to evaluate the proposed method using more

assembly models as future work.

Acknowledgments This work was supported by JSPS KAK-ENHI Grant Number 15K12028.

References

- [1] Tangelder, J.H. and Velthkamp, R.C.: A survey of content based 3D shape retrieval methods, *Multimedia Tools and Applications*, Vol.39, No.3, pp.441–471 (online), DOI: 10.1007/s11042-007-0181-0 (2008).
- [2] Shilane, P., Min, P., Kazhdan, M. and Funkhouser, T.: The Princeton Shape Benchmark, *Proc. Shape Modeling International*, pp.167–178 (2004).
- [3] Jayanti, S., Kalyanaraman, Y., Iyer, N. and Ramani, K.: Developing an engineering shape benchmark for CAD models, *Computer-Aided Design*, Vol.38, No.9, pp.939–953 (2006).
- [4] GrabCAD, available from (<http://grabcad.com/library>).
- [5] Iyer, N., Jayanti, S., Lou, K., Kalyanaraman, Y. and Ramani, K.: Three-dimensional shape searching: State-of-the-art review and future trends, *Computer-Aided Design*, Vol.37, No.5, pp.509–530 (online), DOI: <http://dx.doi.org/10.1016/j.cad.2004.07.002> (2005).
- [6] Bustos, B., Keim, D.A., Saupe, D. and Vranić, T.S.D.V.: Feature-based Similarity Search in 3D Object Databases, *ACM Comput. Surv.*, Vol.37, No.4, pp.345–387 (online), DOI: 10.1145/1118890.1118893 (2005).
- [7] Lian, Z., Godil, A., Bustos, B., Daoudi, M., Hermans, J., Kawamura, S., Kurita, Y., Lavoue, G., Nguyen, H.V., Ohbuchi, R., Ohkita, Y., Ohishi, Y., Porikli, F., Reuter, M., Sipiran, I., Smeets, D., Suetens, P., Tabia, H. and Vandermeulen, D.: A comparison of methods for non-rigid 3D shape retrieval, *Pattern Recogn.*, Vol.46, pp.449–461 (2013).
- [8] Li, B., Lu, Y., Li, C., Godil, A., Schreck, T., Aono, M., Burscher, M., Chen, Q., Chowdhury, N.K., Fang, B., Fu, H., Furuya, T., Li, H., Liu, J., Johan, H., Kosaka, R., Koyanagi, H., Ohbuchi, R., Tatsuma, A., Wan, Y., Zhang, C. and Zou, C.: A comparison of 3D shape retrieval methods based on a large-scale benchmark supporting multi-modal queries, *Computer Vision and Image Understanding*, Vol.131, pp.1–27 (2015).
- [9] Chen, X., Gao, S., Guo, S. and Bai, J.: A flexible assembly retrieval approach for model reuse, *Computer-Aided Design*, Vol.44, No.6, pp.554–574 (online), DOI: <http://dx.doi.org/10.1016/j.cad.2012.02.001> (2012).
- [10] Osada, R., Funkhouser, T., Chazelle, B. and Dobkin, D.: Shape Distributions, *ACM Trans. Graph.*, Vol.21, No.4, pp.807–832 (online), DOI: 10.1145/571647.571648 (2002).
- [11] Deshmukh, A.S., Banerjee, A.G., Gupta, S.K. and Sriram, R.D.: Content-based assembly search: A step towards assembly reuse, *Computer-Aided Design*, Vol.40, No.2, pp.244–261 (online), DOI: <http://dx.doi.org/10.1016/j.cad.2007.10.012> (2008).
- [12] Miura, T. and Kanai, S.: 3D Shape Retrieval considering Assembly Structure, *Asian Symposium for Precision Engineering and Nanotechnology* (2009).
- [13] Ohbuchi, R., Minamitani, T. and Takei, T.: Shape-Similarity Search of 3D Models by using Enhanced Shape Functions, *International Journal of Computer Applications in Technology (IJCAT)*, Vol.23, No.2/3/4, pp.70–85 (2005).
- [14] Hu, K.-M., Wang, B., Yong, J.-H. and Paul, J.-C.: Relaxed lightweight assembly retrieval using vector space model, *Comput. Aided Des.*, Vol.45, No.3, pp.739–750 (online), DOI: <http://dx.doi.org/10.1016/j.cad.2012.10.005> (2013).
- [15] Chen, D.-Y., Tian, X.-P., Shen, Y.-T. and Ouhyoung, M.: On Visual Similarity Based 3D Model Retrieval, *Computer Graphics Forum*, Vol.22, No.3, pp.223–232 (online), DOI: 10.1111/1467-8659.00669 (2003).
- [16] Wang, P., Li, Y., Zhang, J. and Yu, J.: An assembly retrieval approach based on shape distributions and Earth Mover's Distance, *The International Journal of Advanced Manufacturing Technology*, Vol.86, No.9, pp.2635–2651 (online), DOI: 10.1007/s00170-016-8368-z (2016).
- [17] Rubner, Y., Tomasi, C. and Guibas, L.J.: The Earth Mover's Distance as a Metric for Image Retrieval, *International Journal of Computer Vision*, Vol.40, No.2, pp.99–121 (online), DOI: 10.1023/A:1026543900054 (2000).
- [18] Katayama, K. and Sato, T.: Matching 3D CAD Assembly Models with Different Layouts of Components Using Projections, *IEICE Trans. Inf. Syst.*, Vol.E98-D, No.6, pp.1247–1250 (2015).
- [19] Chang, A.X., Funkhouser, T.A., Guibas, L.J., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L. and Yu, F.: ShapeNet: An Information-Rich 3D Model Repository, Technical Report (2015).



Kaoru Katayama received Ph.D. degree in informatics from Kyoto University in 2000. He is currently an associate professor in the Graduate School of Information and Communication Systems, Tokyo Metropolitan University. His research interests are in the areas of data engineering and data mining.



Takumi Sato received master's degree in engineering from Tokyo Metropolitan University in 2016. He currently works as an engineer at NEC Corporation. His research interests are in the areas of pattern recognition.