ANN-based 3D part search with different levels of detail (LOD) in negative feature decomposition

Chih-Hsing Chu a,*, Han-Chung Cheng a, Eric Wang b, Yong-Se Kim b

a Department of Industrial Engineering and Engineering Management, National Tsing Hua University, 101 Kuang Fu Rd., Sec 2, Hsinchu 300, Taiwan
b School of Mechanical Engineering, Sungkyunkwan University, Suwon, Republic of Korea

ARTICLE INFO

Keywords:
Similarity assessment
Levels of detail (LOD)
Negative feature
Feature recognition
Part search
Design retrieval

ABSTRACT

Duplicate designs consume a large amount of enterprise resources during product development. Automatic search for similar parts is an effective solution for design reuse. Previous studies have only concerned similarity assessment based on complete 3D models, which may produce unsatisfactory result in practice. This paper proposes a novel scheme which incorporates the concept of LOD (levels of detail) into 3D part search. The scheme allows searching with different LOD variants created from the negative feature tree (NFT) of a solid model. A back-propagation artificial neural network is established to combine the D2-based similarity evaluation at each level of NFT. A human cognition model (HCM) is obtained by training the network with a set of data generated from a human experiment of similarity ranking. Search examples based on HCM show that the proposed scheme provides a practical tool for retrieval of similar part models.

1. Introduction

Product development is a crucial activity for most modern enterprises. Lowering costs, improving quality, and shortening time-to-market have been identified as key factors in the development process (Chu, Chang, & Cheng, 2006). Design is a critical stage in most new product development. In the automobile industry, it determines 80% of the total development costs, which can only be decreased by less than 25% after this stage (Regli & Cicirillo, 2000). Over 75% of design activities belong to design modification, variant design, or case-based design (Wood & Agogino, 1996). Engineers usually adapt existing drawings to fulfill new requirements, enhance functionalities, and/or lower costs. How to choose a good product design to start with is a critical decision in these cases.

Many companies have introduced PDM (product data management) technologies for managing product design information, but effective tools for searching similar parts according to design intent are not yet available. Text-based search on file names and keywords does not always produce useful results (Elinson, Nau, & Regli, 1999). Unfortunately, this is the approach (probably the only approach) commonly used in companies. It is not surprising that parts are frequently recreated in different forms yet for the same function, making multiple copies of the same data. Duplicate designs thus consume a large amount of enterprise resources and lengthen the product development time (Chu & Hsu, 2006). They influence not only data management, but also production utilization, inventories, tooling, and other tasks of a product life cycle. Taking on a wider context, strategic sourcing to appropriate suppliers becomes an imperative in distributed product development (Iyer, Jayanti, Lou, Kalyanaraman, & Ramani, 2005). Broadening the supplier base helps strengthen the core competence of most manufacturers (Yang & Lai, 2006). Similarity assessment for 3D components is recognized as an enabling technology that facilitates supplier selection and evaluation (Tanskanen, 1997). Good part search that permits design knowledge reuse would rely on content-based similarity comparison. This requires adoption of advanced CAD technologies.

Most similarity assessment schemes generate a Shape Signature from 3D models and distinguish different models based on the dissimilarity of the signature defined by a distance function (Cardone & Karnik, 2003; Natraj, Subramaniam, Kuiyang, Yagnanarayanan, & Karthik, 2005; Yang, Lin, & Zhang, 2007). Shape Signature is a high-level abstraction of 3D geometry that is unique to each model in theory. Most previous studies concerning part search can be classified into six categories based on the information contained in the shape signature: feature-based, histogram-based (shape distribution, shape statistics), graph-based (B-Rep topology graph, skeletal graph, reeb graph), product information-based (section image, group technology), 3D object recognition (aspect graph, extended Gaussian images), and global attributes-based methods (spherical harmonics, invariant) (Cardone & Karnik, 2003; Natraj et al., 2005;...
Yang et al., 2007; Ramesh, Yip-Hoi, & Dutta, 2001; Cardone, Gupta, Deshmukh, & Karnik, 2006). The choice of shape signature and similarity function largely determines the discrimination capability of similarity assessment.

In practice, two 3D parts often do not have to be similar in every detail from the perspective of design reuse. Partial similarity, rather than the similarity based on complete models, may better satisfy the purpose of searching for similar parts. Previous studies, however, were lack of support to such partial similarity search. To overcome this deficiency, this paper proposes a 3D part search scheme that integrates the concept of LOD (levels of detail). This scheme creates different LOD variants of a solid model from different levels of negative feature decomposition, i.e. the negative feature tree of the model. The similarity assessment incorporates the comparison result of each level in the tree based on the D2 distribution of the corresponding LOD models. A back-propagation artificial neural network (ANN) is established to characterize human cognition in ranking similar parts. A human experiment is conducted and the experimental data determines the weights of the neural network that combine the similarity values of different levels. The trained ANN works as the kernel of a human cognition model (HCM) for part search. Test examples show that the proposed scheme performs well in discriminating 3D mechanical parts. It facilitates design reuse by offering a novel capability of searching with different LODs.

2. 3D part search based on feature tree

2.1. Overall framework

This work develops a 3D part similarity assessment scheme integrated with the concept of LOD (levels of detail). Fig. 1 shows the search procedure adopted by the scheme. First, the user uploads a query part or classified sets of negative features that may represent the current design intention. Different LOD models are generated from different levels of the negative feature tree. A D2-based shape signature is then created at each level from the distance sets of all the features belonging to it. Every part model

![Diagram](image-url)
to be assessed is retrieved from the database, along with its shape signature of each level, and compared with the query model. We then compute the dissimilarity value between the query/candidate parts at each level and combine them using an artificial neural network. Or, the user is allowed to specify a set of attributes for combining these values.

2.2. NFD-based LOD model

Previous studies have addressed the importance of manufacturing feature recognition in automatic process planning (Han, Pratt, & Regli, 2000). Volumetric decomposition (Sakurai, 1995) provides the capability of hierarchical approximation based on the features of a part. Convex decomposition was proposed for representing polyhedral objects with form features and negative features (Kim, 1992; Kim & Wilde, 1992; Waco & Kim, 1994). This work adopts the similar approach, i.e. alternating sum of volumes with partitioning (ASVP) decomposition is applied for establishing a feature tree of a part. ASVP is a convex volume decomposition using convex hull, set difference, and cutting operations (Kim & Wilde, 1992). It organizes the boundary faces of a part in an outside-in hierarchy, while associating volumetric components with these faces. The ASVP decomposition is converted into the form feature decomposition (FFD) by systematically combining components according to the hierarchical structure of the decomposition result and the face dependency information obtained during the decomposition process (Kim, 1992). Positive-to negative conversion is then applied to convert the FFD into negative feature decomposition (NFD) (Waco & Kim, 1994). Fig. 2 shows the conversion process from ASVP to NFD. The result of NFD refers to as negative feature tree (NFT).

As shown in Fig. 3, LOD model is the simplified representation of a 3D part with different levels of features in its NFT. It corresponds to different groupings of the hierarchy in negative feature decomposition, which consists of one positive base feature (stock) and other negative removal volumes (machining volumes). Coarse models (or models of lower LOD) are produced by pruning the nodes of lower levels and performing Boolean operations on the pruned tree. This provides a systematic way of controlling levels of detail in a 3D model.

2.3. D2 similarity assessment

Histogram-based similarity methods offer many shape functions to be chosen such as A3, D1, D2, D3 and D4. The D2 function is considered as a better shape discrimination criterion among them (Osada, Funkhouser, Chazelle, & Dobkin, 2002). Thus, we adopt D2 as the major shape signature in similarity assessment. However, previous studies have found that regular D2 cannot distinguish certain geometries, e.g. Fig. 4 shows that two different parts mainly consisting of negative features have a similar D2 distribution (IP, Lapadat, Sieger, & Regli, 2002; Rea, Sung, Corney, Clark, & Taylor, 2005). The reason is that the regular D2 function does not differentiate among in-distance, out-distance, or mixed distance when taking the sample distance between random points on the part surface. The method loses its discrimination capability in minor shape discrepancies, or when the part model contains severe feature interactions. We try to overcome this problem by integrating D2 with NFT. In this case, the D2 assessment is applied to the negative feature volumes of NFT. These volumes are guaranteed to be convex, and more importantly without intersections with other features. These characteristics inherited by the feature decomposition process help improve the discerning capability of the method. Fig. 5 shows the resultant D2 distributions.

We have developed a three-step algorithm for similarity assessment complying with the above ideas. Instead of comparing the similarity of two complete parts, it compares the feature decomposition result at each level using the D2 function.

2.3.1. Generate D2 distribution at each level

Generate a set of distances between two random points for every feature of each level; then compute D2 distributions by the distance sets. The total number of random points is proportioned to the feature size. This step is described as follows:

1. Suppose \( L = \{\text{LOD}_1, \ldots, \text{LOD}_n\} \) denotes a set of LOD models at \( n \) levels and the feature set of LOD\(_i\) is given by \( F^i = \{f_1^i, \ldots, f_m^i\} \).
2. Triangulate each \( f_j \) in \( F \) into facets. Assume \( d_j \) represents a set of distances between two random points in these facets. Then, the sets of distances of \( \text{LOD}_i \) are given by \( D^i = \{d_1^i, \ldots, d_j^i, \ldots, d_m^i\} \) and generated as

\[ D^i = \{d_1^i, \ldots, d_j^i, \ldots, d_m^i\} \]

Fig. 2. The conversion process of negative feature recognition (Waco & Kim, 1994).

Fig. 3. Different LOD models generated from NFT.
The triangular facets of $f_j$ are expressed as $T = \{t_1, t_2, \ldots, t_q\}$. Calculate the surface area of each triangular facet; estimate the total area ($TA$) by adding the facet area, and log the accumulated $TA$ with respect to each facet.

- Generate a random value within $[0, TA]$ and identify the facet corresponding to the value in the log.
- Generate two random values $r_1, r_2$ within $[0,1]$; then calculate a random point $p$ with the three vertices $P_1, P_2,$ and $P_3$ of the facet and $r_1, r_2$ according to

$$p = (1 - \sqrt{r_1})p_1 + \sqrt{r_1}(1 - r_2)p_2 + \sqrt{r_1r_2}p_3$$

(1)

- Calculate $d_p$, which consists of a set of distances between two points randomly generated from $f_j$.

(3) Generate $D2$ distribution $h_i(bin, \delta)$ of LOD, as the follows:

- Derive the maximum distance ($D_{MAX}$) from $D'$ and the bin-width is given by $D_{MAX}/\text{bin\_number}$.
- Set the corresponding $\text{bin\_number}$ for every distance value of $D'$, $j = 1, \ldots, \text{bin\_number}$.
- Generate a shape distribution $h_i(bin, \delta)$ based on occurrence probability $\delta_j$ of distance in each $\text{bin\_number}$, where $\text{bin\_number}$ is the user input parameter.

(4) Define the $D2$ set of each level as $H = \{h_1, \ldots, h_j, \ldots, h_n\}$.

![Fig. 4. The D2 shape histograms for different LOD models are very similar.](image)

2.3.2. Calculate the dissimilarity value of each level

The number of levels in the NFT of the query model may be different from that of candidate models. To compare models of this case is problematic. We propose the following method for solving this difficulty:

(1) Determine the maximal number of levels ($\text{max}_n$) in the query model and all candidate models.

(2) Obtain $H^1 = \{h_1, \ldots, h\}$ for the query model, and $H^c = \{h_1, \ldots, h\}^c$ for a given candidate model. Both NFTs have $q$ and $c$ levels, respectively.

(3) For each $h_i$, we need to distinguish four conditions in calculating the similarity value between the query and candidate parts.

- $\text{max}_n > q$ and $\text{max}_n > c$:
  - If $h_i$ exist in both query and candidate models, the dissimilarity value ($DS_i$) is defined as
    $$DS(H^1, H^c) = \sum_{i=1}^{\text{bin}} |\delta_i^1 - \delta_i^c|$$
    (2)
  - If $h_i$ does not exist in either of them, then $DS_i = 1$.
  - If $h_i$ does not exist in both, then $DS_i = 0$.

- $\text{max}_n = q$, $\text{max}_n > c$:
  - If $h_i$ exists in both query and candidate models, then calculate $DS_i$ based on Eq. (2).
  - If $h_i$ does not exist in the candidate model, then $DS_i = 1$.

- $\text{max}_n > q$, $\text{max}_n = c$:
  - If $h_i$ exists in both query and candidate models, then calculate $DS_i$ based on Eq. (2).
  - If $h_i$ does not exist in the candidate model, then $DS_i = 1$.

- $\text{max}_n = q = c$, then calculate $DS_i$ according to Eq. (2).

Assume the dissimilarity values of a candidate model is denoted as $L = \{DS_1, \ldots, DS, \ldots, DS_{\text{max}_n}\}$; the values of all candidate models is written as $\text{DisSet} = \{L^1, \ldots, L^p\}$, where $P$ denotes the number of candidate parts.

2.3.3. Combine the dissimilarity values of all LOD levels

There are two ways of determining the final similarity value from the ones of each level. A simpler method is to let the user specify the weights of each LOD in searching similar parts. The next section will describe a human cognition model based on artificial neural network in detail.

(1) Specify the number of LODs ($L_n$) in similarity assessment and select between complete/partial comparison, i.e. the user intends to assess the similarity value based on the complete model or the model of certain LOD.

(2) Input weights for each level within and transform the normalized weight ($w_i$) according to

$$w_i = \frac{R_i}{\sum_{i=1}^{p} R_i}$$

(3) Retrieve $L^k = \{DS_1, \ldots, DS, \ldots, DS_{\text{max}_n}\}$ from $\text{DisSe}$ and combine them into one single value by

$$\text{DisValue}_k = \sum_{i=1}^{p} w_i DS_i$$

(4)

For a given set of $p$ candidate parts, we compute the corresponding $\text{DisValue}_k$ and determine their similarity ranks with respect to the query part. $\text{DisValue}_k$ in the human cognition model (HCM) is determined by the neural network as
DisValue\(^k\) = HCM(DS\(_1\), …, DS\(_n\), …, DS\(_{\text{max}_n}\)) (5)

3. Human cognition model (HCM)

Artificial neural networks (ANN) construct complex system models without the need of explicit descriptions of the system characteristics or rules. A network consists of processing elements and connections. Each processing element has a single output signal that fans out along connections to every other processing element. An MLFF (multi-layer feed forward) network using arbitrary squashing functions can approximate most functions of interest to a desired degree of accuracy (Hornick, Stinchcombe, & White, 1989). Multi-layer back-propagation networks with sigmoid output functions are usually used to approximately represent most continuous mappings (Funahashi, 1989). We adopt this computing mechanism for characterizing the human cognition in ranking similar parts. Our assumption is that humans determine the similarity (or dissimilarity) between two 3D models by incorporating the discrepancies at different levels of detail. Personal differences are supposed to be reflected on how these discrepancies are combined into the final similarity. Artificial neural networks provide a proper tool for capturing such human cognition. To be specific, they can be used to learn, at least approximately, how the final similarity is constructed from the result at different levels. This is our main idea.

3.1. Human similarity cognition experiment

We design a human cognition experiment to generate a set of training data for ANN. The experiment consists of the following steps.

- Produce test 3D parts
  - Thirty-two test models are generated, subject to the limitations of negative feature decomposition (Kim & Wilde, 1992). One is randomly selected as the query part.
- Select subjects
  - It is necessary that the people to be tested have experience of part design and some knowledge of 3D CAD. There are eight subjects in the experiment and all of them are graduate students in engineering.
- Experiment rules
  - The subjects can freely rotate, translate, and zoom all test 3D models in computer. The experiment time is 30 min. The subjects are suggested to separate the test models into several groups, i.e. to place similar parts into a group; then rank the query part within each group.
- Determine the rank of test parts
  - Compute the similarity value among the groups as

\[
\text{Group Similarity} = \frac{n + 1 - G_i}{m(n + 1)/2}
\]

where \(G_i\) denotes the similarity rank of group \(i\) and \(n\) is the number of groups. The next step is to determine the part similarity value by

\[
\text{Part Similarity} = \frac{m + 1 - P_j}{m(m + 1)/2}
\]

where \(P_j\) denotes the similarity rank of part \(j\) and \(m\) is number of parts in the group to which the part belongs. The similar similarity value of a test part is expressed as

\[
\text{Part Similarity} \times \text{Group Similarity}
\]

- Average the similarity values of the eight subjects.
- Rank all test parts by the similarity value in a descending order.

\[
\text{DisValue}^k = \text{HCM}(\text{DS}_1, \ldots, \text{DS}_n, \ldots, \text{DS}_{\text{max}_n})
\]

3.2. Training back-propagation network

The human cognition method adopts a back-propagation network (BPN) for combining the dissimilarity value of each level. The network is trained with the experimental data to learn the way the human subjects determine the similarity rank. The following procedure describes the training process.

- Obtain input data to BPN
  - Obtain the input data to the network with the similarity assessment algorithm described previously, i.e. \(\text{DisSet} = \{L^1, \ldots, L^{32}\}\). The BPN consist of six input nodes equivalent to the maximum level number in the NFT’s of the test parts. The dissimilarity values of each candidate part is written as \(L^1 = \{\text{DS}_1, \ldots, \text{DS}_n\}\).
- Determine the target value of BPN
  - The target value of the BPN is determined by the result of the human experiment. The rank number in the sorted list, generated in the last step of the experiment, is converted into a dissimilarity value as

\[
\text{Dissimilarity value} = 1 - \left(\frac{32 - \text{Rank}}{32}\right)
\]

where the value 1 indicates that the query and candidate parts are most similar. The target value is derived as the average of eight dissimilarity values for each candidate part. The BPN has only single output, corresponding to the average dissimilarity value.
- Train BPN to characterize human similarity cognition
  - We use Matlab\textsuperscript{\textregistered} http://www.mathworks.com/ to build the network. There are a number of parameters (e.g. learning rate, momentum, network architecture, training/testing ratio, epochs, and input sequence) to be determined via the training process. The procedure for determining these parameter values is as follows:

### Table 1

<table>
<thead>
<tr>
<th>Network architecture</th>
<th>LR</th>
<th>(\mu)</th>
<th>MSE (a) Training</th>
<th>MSE (b) Testing</th>
<th>MSE [(a)-(b)]</th>
</tr>
</thead>
<tbody>
<tr>
<td>6-4-1</td>
<td>0.01</td>
<td>0.8</td>
<td>0.028759</td>
<td>0.031011</td>
<td>0.002252</td>
</tr>
<tr>
<td>6-5-1</td>
<td>0.01</td>
<td>0.8</td>
<td>0.020675</td>
<td>0.023186</td>
<td>0.002511</td>
</tr>
<tr>
<td>6-6-1</td>
<td>0.01</td>
<td>0.8</td>
<td>0.014901</td>
<td>0.017329</td>
<td>0.002428</td>
</tr>
<tr>
<td>6-7-1</td>
<td>0.01</td>
<td>0.8</td>
<td>0.014036</td>
<td>0.018563</td>
<td>0.004527</td>
</tr>
<tr>
<td>6-8-1</td>
<td>0.01</td>
<td>0.8</td>
<td>0.013671</td>
<td>0.019076</td>
<td>0.006637</td>
</tr>
<tr>
<td>6-10-1</td>
<td>0.01</td>
<td>0.8</td>
<td>0.012687</td>
<td>0.019324</td>
<td>0.013929</td>
</tr>
<tr>
<td>6-13-1</td>
<td>0.01</td>
<td>0.8</td>
<td>0.011151</td>
<td>0.023543</td>
<td>0.005392</td>
</tr>
</tbody>
</table>

Epochs is 20,000 and ratio is 1:1.
(1) Select the error function
The function estimates the error between the output and the target value. The mean square error (MSE) is used as the performance function during the training process.

(2) Choose the activation function
Since the similarity assessment involves continuous mapping with a value within $[0,1]$, a binary sigmoid function is chosen as the activation function of the BPN. The input transforms the output based on
$$f(x) = \frac{1}{1 + e^{-x}}$$

(3) Determine the momentum ($\mu$)
Generate five $\mu$ values by decreasing from 0.9 to 0.5 in a step of 0.1, with other parameters fixed. The MSE of the testing set becomes minimal at $\mu = 0.9$. We next obtain ten values by decreasing from 0.95 to 0.9 in a step of 0.01 to determine whether MSE begins to rise when $\mu > 0.9$ or when $\mu < 0.9$.

(4) Decide the learning rate (LR)
Identify the condition when the MSE starts to increase when $LR > 0.1$, and again obtain nine values with an increment of 0.01 from 0.01 to 0.1. The MSE of the testing set becomes minimal in the case of $LR = 0.05$.

(5) Confirm training/testing number
As stated above, the number of test parts is 32. The training vs. testing ratio is chosen as 5:3 for multifold cross-validation. The testing set contains J5, J11, J12, J14, J15, J16, and J26 (see the implementation section). The validation set consists of five parts J6, J13, J17, J22, and J23, which are randomly chosen from the testing set. The validation set verifies the test result of unknown parts.

(6) Decide the epochs
A MSE figure helps identify that epochs = 30,000 is the stop point during the training under the parameter combination as $\mu = 0.9$, $LR = 0.05$ and 6-4-1 network architecture. The MSE of the testing set becomes minimal at this point.
Construct the network architecture
The hidden layer of the BPN is chosen as one. The maximal and minimal numbers of the hidden nodes are 13 and 4, respectively, according to the following equations:

\[ H = \frac{M + N}{2} \]  
\[ H = 2N + 1 \]

where \( H \) denotes the number of hidden nodes; \( N \) represents the number of input nodes and \( M \) is the number of output nodes. We test out seven different network architectures and the result is shown in Table 1. The MSE of the testing set becomes lowest in the 6-6-1 architecture. A large number of hidden nodes can lower the MSE of the training set, but they may increases the MSE of the testing set. This condition takes place when \( H > 6 \). Therefore, the network parameters are determined as \( \mu = 0.9 \), LR = 0.05, epochs = 30,000, and 6-6-1 architecture. Fig. 6 shows the change of the MSE.

Integrate BPN into the similarity assessment algorithm
The trained BPN serves as the underlying mechanism of the human cognition method. The network is integrated into the algorithm to reflect the characteristic of human similarity cognition in 3D part search.

4. Implementation result
The proposed search scheme is implemented with C++ in ACIS® http://www.spatial.com/ for testing its effectiveness. All part models (32 in total) have been converted into the SAT format prior to
the test. Negative feature decomposition is applied to the converted models using the method developed in the previous work (Kim, Jung, Kang, & Rho, 2003). Fig. 7 shows the similarity assessment results based on the complete models and three different LOD models (level 4, 3 and 2). The maximal number of levels is six in the NFT’s of the models. The weights are chosen as 1, 0.8, 0.5, 0.3, 0.2, and 0.1 for each level (from top down). The first column shows the search result based on the complete models with six levels in NFT. The most similar part is the same in all four cases despite that the query model (J02) is of a different level of detail in each case. The parts of ranks 2 and 3 are not identical. J07 appears as the second part in the 4-level search. J04 and J07 are second and third ones in the 3-level case, respectively. Their ranks swap in the 2-level search. The corresponding NFT is also shown under each part model.

Fig. 8 shows the search result produced by the human cognition method based on the complete models. The query model is J09. The part number, the similarity rank determined by human beings, and the similarity rank computed by HCM are also listed respectively next to each model. We randomly chose five validation models in the human experiment (they are marked with squares in the figure). These models are held unknown to HCM prior to search. In the search result, only one similarity rank among them (J06-J22-J17-J13-J23) is different from that of the human comparison (J06-J22-J13-J17-J23). Table 2 shows the precision rate of the cognition model as the number of searched parts varies. The rate is calculated as the ratio between the number of parts found by HCM and the total part number. For example, all the first five parts have been successfully identified. The precision rate variation illustrates that the network has captured the human discerning capability to a large extent. In addition, a modified NAP (non-interpolated average precision rate is used for characterizing the similarity assessment capability of the proposed model. The definition of the metric is

Fig. 9. Five new models created from deleting the NFT of J11.

Fig. 10. Search for five similar parts generated from J11.
\[
NAP = \frac{\sum_{i=1}^{N} \text{min}(\text{Rank}_i)}{\text{max}(\text{Rank})}
\]

where \( N \) is the total part number; \( i \) is the similarity rank predicted by HCM and \( \text{Rank} \) is the ideal rank, i.e., the rank determined by the human experiment. The NAP value of the HCM search is 0.800763, which indicates a good ranking performance.

In addition, we select the part J11 as a new query model. Five new test models are generated by deleting some features from the NFT of the part. Fig. 9 shows these models and their corresponding NFT’s. These models are supposed to appear in high similarity ranks in the part search based on J11, due to the way they are constructed. Fig. 10 shows the eight most similar parts recognized by HCM. The five test models rank 1, 2, 4, 5, and 6 in the search result. Only J12 occupies one place among the top five. This test validates the shape discerning ability of the method.

5. Conclusions

Automatic 3D part search facilitates design reuse in new product development. In practice, a search method allowing part comparison with partial model information may capture the user’s design intent better. Previous studies have provided a variety of methods for assessing shape similarity based on complete 3D models, but have not yet addressed partial similarity assessment. This study proposes a novel scheme for this need which incorporates the concept of LOD (level of detail). Similarity comparison can be conducted on LOD variants that correspond to different subtrees in negative feature decomposition of a solid model. The similarity value combines the assessment result of each level computed with a D2 shape distribution of the corresponding LOD model and weighted by user inputs. In addition, a human experiment was conducted to determine the similarity rank of a set of test models with respect to a query part. A back-propagation artificial neural network was trained with the experimental result. The network learns and captures human preference for combining the similarity of each LOD in part comparison. It works as the shape assessment mechanism in a human cognition model (HCM). We then test on a set of 3D part models using the method in various manners for verifying its discriminating capability. The similarity ranking of the test models indicates a good performance level with a 0.8 NAP value compared to that of the human experiment. Besides, the precision rate is high in ranking unknown models. The test result shows that the proposed scheme provides a novel and practical tool for retrieval of similar part models. This work can be further improved in several aspects. First, a part search engine of practical use should provide Google-like user interfaces for more complex query with Boolean operations applied to feature attributes (e.g., type, size, parameters). In addition, a fuzzy-logic approach can be included for enhancing the effectiveness of part search.

References


