Assembly Retrieval in Top-down Product Design

Xiang Chen, Song Guo, Jing Bai, Shuming Gao*
State Key Laboratory of CAD&CG, Zhejiang University
Hangzhou, P.R.China

Abstract

Nowadays, there are abundant product models in existence that contain plenty of design knowledge in various disciplines. As an approach to taking full advantages of the existing design knowledge, model reuse plays a more and more important role in complex product design due to the enormous time and cost saving it could bring. To help designers find the right models for reuse quickly and exactly, the content-based assembly retrieval is a promising but still immature way at present. In this paper we present a novel assembly interface representation which captures much more design knowledge in assembly interfaces than plain geometric-matings used most in current CAD models. Then the hierarchical assembly descriptor based on the presented assembly interface representation is described and compared to assembly descriptors in other work. Furthermore, the corresponding similarity assessment method is given to evaluate the distances among assemblies and a complete assembly retrieval method is also described. Finally, an assembly retrieval prototype system is implemented, based on which assembly search samples are analyzed to demonstrate the improved effectiveness brought into the assembly retrieval.

Key words: CAD model, design reuse, assembly retrieval, multi-level assembly interface

1. Introduction

As the rapid evolution of both computer industry and mechanical engineering, the requirement of computer aided tools for supporting large and complex product design have become urgent. Among the various promising design methods in mechanical engineering, the top-down assembly design method [1] is quite suitable for large or complex product design. The kernel ideas of this design method are decomposition and inheritance, while the product design starts from abstract concepts and evolves into concrete geometric models gradually.

Interestingly, a fact is found that in most of real-world top-down assembly design, engineering designers do not always generate everything in the product from scratch. By contrast, they would often like to reuse the existing components (component is the general term used to mention part or sub-assembly in this paper) which are suitable for their design objects and behave maturely before. Naturally, model retrieval is the way to find out what designers want, and the retrieved ones could then be adjusted parametrically for reuse according to design constraints and specifications.

However, choosing the suitable query for assembly retrieval is not a trivial problem. We find that during the design process, there are many incomplete intermediate models called "skeletons" for controlling the product layouts and component shapes [2, 3]. These models hold important design parameters which are the well-defined common results at some milestones during the whole design process and take the role of leading the following design activities. In other words, only through the help of the skeleton model, the geometric model of the product is able to evolve from coarse and vague design spaces to detailed and clear assemblies as a complete integration. Therefore, being a model which contains incomplete but crucial design knowledge, the skeleton is quite an appropriate query for searching assemblies.

* Corresponding author email: smgao@cad.zju.edu.cn

In fact, the retrieval for component reuse in top-down assembly design described above is one of the most important requirements in assembly retrieval. Although there are some previous work about assembly retrieval in mechanical engineering, shape descriptors and search algorithms which fit top-down assembly design well are still absent.

In the paper, a retrieval framework which is suitable for assembly search in top-down assembly design is presented. In our ideal scene, a designer gets the high level result of conceptual design and figures out the corresponding skeleton of the component he is responsible for, then he uses the simple skeleton as the query to find detailed assemblies which coincide with his design more or less, and finally he modifies the retrieved models to make them fit exactly to the other components or specific constraints in the whole product (Fig. 1).

The rest of the paper is organized as follows. In the next section we give a brief overview of the related work on CAD assembly retrieval. In section 3 we overview our assembly retrieval framework. Section 4 talks about the query for assembly retrieval in top-down assembly design, and Section 5 defines the assembly descriptor and illustrates the reasons behind. In section 6 we describe the details of graph matching and similarity assessment, while section 7 explains the implementation and analyzes the experimental results. Finally, we conclude the paper and present further work directions.

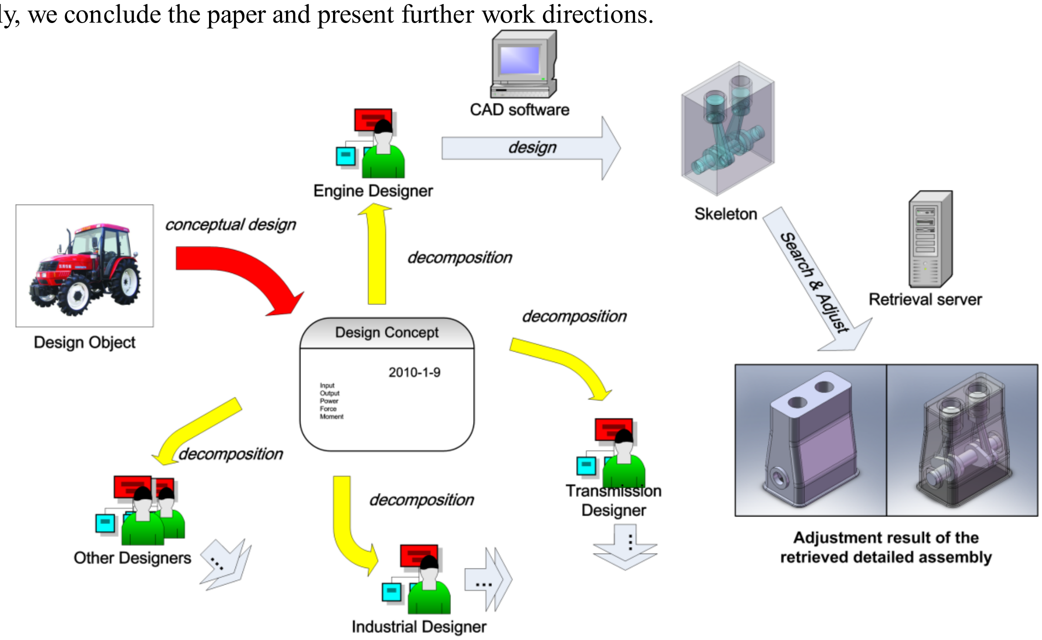

Fig. 1. The ideal reuse scene in top-down assembly design based on retrieval.

2. Related work

As Llewelyn et al. [4] point out, the design cost takes up about 15-20% of the whole product cost, but decides 70-80% of it. And from Gunn et al. [5] we learn that only 20% of parts require completely new designs, while 40% of them are obtained by directly reusing existing designs and the other 40% are achieved through editing on the existing designs. Therefore, the model retrieval technology in CAD domain attracts more and more attention because of its vital effect in model reuse during design activities.

Iyer et al. [6] have given an exhaustive review on the 3D model retrieval in CAD domain. They list several categories to which the reviewed works belong: Global feature-based; manufacturing feature recognition-based; graph-based; histogram-based; product information-based; and 3D object recognition-based. Actually, nowadays requirements in CAD retrieval has remarkable trends on multi-level and semantic modes, but most of the works mentioned in [6] do not support the multi-level and semantic retrievals well.

In these years, many researchers notice the multi-level and semantic trends and try to address the corresponding problems. Li et al. [7] present an FDAG (feature dependency directed acyclic graph) based retrieval method which can support both global and partial retrieval of CAD models. Cardone et al. [8] give a new similarity assessing method based on machining features, thereby the models with more similar machining cost would have higher similarity. In work [9], Gao et al. describe each CAD model as a DBMS graph whose nodes correspond to sub-parts of the model, and the sub-parts matching and similarity assessment are based on sub-graph isomorphism. Bai et al. [10] extract local reusable regions from feature-based models automatically based on design semantics and heuristics, and then multi-mode retrieval is carried on between the query and the reusable regions effectively. The same authors also present a general hierarchical graph representation and corresponding similarity assessment for CAD model retrieval in [11].

Unfortunately, the works mentioned above do not refer to the assembly retrieval problem. Deshmukh et al. [12-14] have presented comprehensive work on assembly retrieval. The main idea is the mixture of multiple assembly retrieval methods in a unique framework. A typical method among these is the use of mating graph in assembly. Users could input a segment of mating graph as query and find the assemblies whose mating graph contains the query, hence the Ullmann [15] algorithm is adopted to handle the sub-graph isomorphism problem. However, this method requests the users to remember some segments of the most detailed-level relations in the desired model. This cannot satisfy the specific retrieval requirements of top-down assembly design well, while the design activities often work on abstract components and the high-level relations among these components. In fact, the absence of multi-level and semantic information in the assembly descriptor is one of the main reasons why the work [12-14] could not support top-down assembly design well, about which we will discuss more in later sections.

In summary, multi-level and semantic based retrieval will play a key role in future CAD design activities, and our work in the paper considers these two elements as the main criteria for assembly retrieval.

3. Overview of the assembly retrieval framework

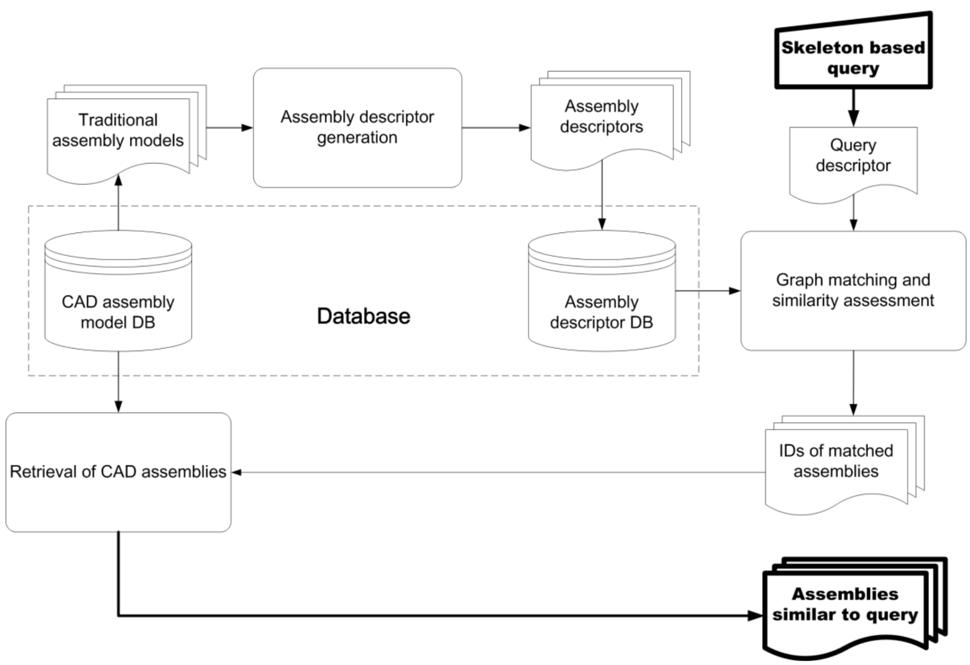

Fig. 2. The assembly retrieval framework.

To help designers reuse existing models effectively in top-down assembly design, we present an assembly retrieval approach that could fulfill the specific requirements of reuse in high-level design and locate exactly the assembly models wanted by designers. An overview of the assembly retrieval framework is shown in Fig.2.

The assembly descriptors of the assembly models in database are constructed in advance; then the assembly search starts from the skeleton-based query model which contains simplified geometric information and high-level design knowledge; based on the extracted descriptors of the query model and the assembly models in database, graph matching and similarity assessment are executed to find the assembly models matching with query and give the corresponding similarities; finally the matched models are retrieved from database and sent back to designers in a sorted manner based on the calculated similarities. The following items are the key aspects of the assembly retrieval approach which we will discuss at length in next sections:

- The skeleton based query.
- Descriptor of assembly model.
- Graph matching and similarity assessment.

4. The skeleton based query

The query needed in assembly retrieval for top-down assembly design should capture the high-level design knowledge in a relatively simple representation. In our opinion, this query model could not be found in the traditional assembly structure, which is not appropriate to be used in top-down assembly design (although it is quite suitable for representing the detailed assembly model). The reason is that the traditional structure is static while the top-down design is a dynamic process and the model evolution information during design should also be captured. Actually, we have presented a multi-level assembly model for top-down assembly design [16, 17] which stores the information used in different design phases. Each component in this model contains three-level representations, i.e. abstract model, skeleton model and detailed model (Fig. 3 shows a class diagram section of our top-down assembly model). The abstract model is used for conceptual design, while the detailed model is the final representation possessing complete design results which includes all the low-level details. Here what obtains most attention is the skeleton model:

- The skeleton model contains simplified and incomplete geometric representation which is often generated by the main designer to control the overall shape and structure of a component.
- Besides the simplified shapes, the assembly skeleton especially contains assembly interfaces among its child components, and these interfaces may be much more abstract and high-level than the detailed geometric mating relations used in traditional assembly model.

In other words, the skeleton model of a component stores the high-level design knowledge of the component and is always used as a contract for the following low-level designs.

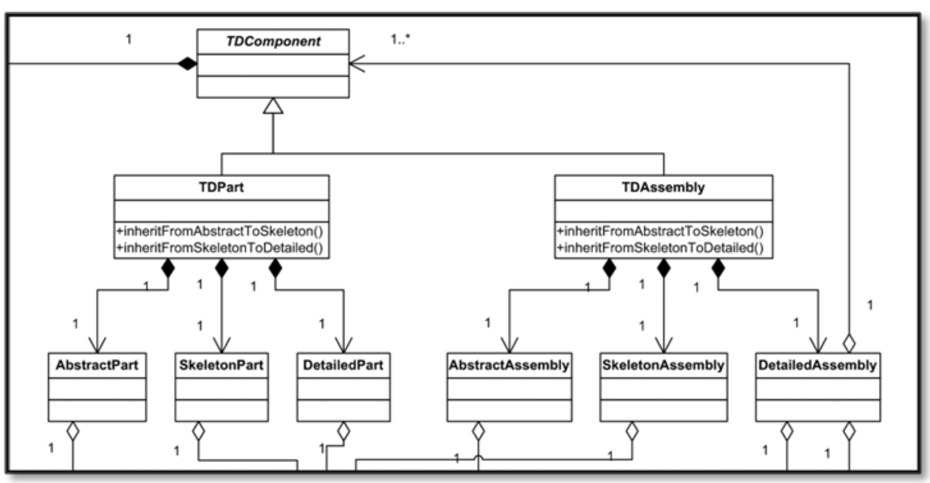

Fig. 3. A section of the multi-level assembly model for top-down assembly design.

As the skeleton model has the characteristics of being simple and high-level, we found that using assembly skeleton models as queries in assembly retrieval is the natural choice for designers in top-down assembly design. Achieving this kind of assembly retrieval will be a promising way of helping product knowledge reuse.

5. Descriptor of assembly model

In order to achieve effective assembly retrieval, a reasonable assembly descriptor with comprehensive information involving design knowledge and geometric shape is a prerequisite.

A key method in [14] is the use of detailed geometric mating graph as the descriptor for assembly retrieval. This method could search the detailed assemblies well through hint of a mating graph section. However, there are two shortages when this method is applied to assembly retrieval in top-down assembly design.

- The geometric mating graph is flat, i.e., all parts of an assembly are connected through mating relationships in a single graph. However in high-level design, designers usually consider several parts as a whole component for clear and pure thinking. When this happens, only those relationships among high-level components appear and the inner relationships are ignored. As the sample shown in Fig. 4, when designers input query graph during design, assembly with mating graph in the right cannot be found out using sub-graph isomorphism algorithm. Actually in designers' opinions, this assembly should be a matched one to the query, because:
    - Component C is split into part C1 and part C2 in detailed design.
    - The mating relationship (A', B'), (A', C1) and (B', C2) are the same as (A, B), (A, C) and (B, C) respectively.
    - (C1, C2) is just an inner relationship.
- The geometric mating relationship itself is too low-level for design. Different composition of geometric matings can implement the same high-level design objective, e.g. the kinematic relationship between components. Hence graph matching based on this low-level relationship will lose many assemblies with similar high-level assembly interfaces to query during retrieval. However, these assemblies similar in high-level knowledge are often desired by designers in top-down assembly design.

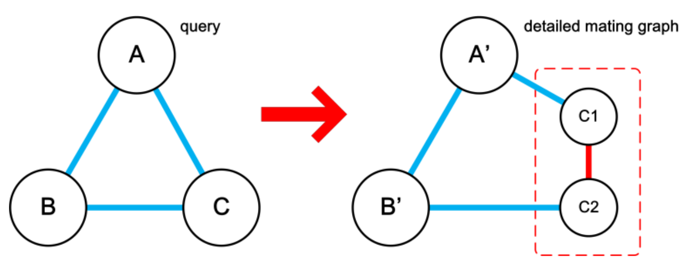

Fig. 4. Illustration of the mating graph affected by component splitting.

To solve these two problems, we present a multi-level assembly interface representation and adopt it accompanying with the hierarchical assembly structure as the assembly descriptor. Contents below give the details.

5.1 Hierarchical assembly structure

In modern mechanical design the assembly structure used most is a hierarchical tree structure [18] in which all nodes are parts (leaf nods) or sub-assemblies (non-leaf nodes). There are child components under a sub-assembly (part-of relationships), and these child components are assembled together through some geometric matings (assembling relationships), which form a graph structure implicitly. A sample (from the detailed engine model in Fig.1.) of this mixed structure is shown in Fig. 5. Considering the hierarchical characteristic in this model, we use it to address the flat mating graph problem mentioned above.

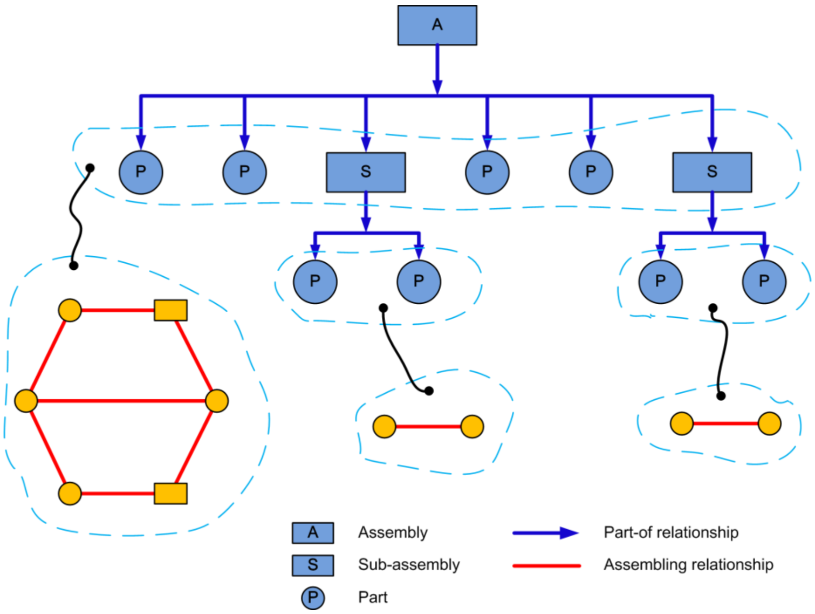

Fig. 5. The traditional hierarchical assembly structure.

5.2 Multi-level assembly interface

As discussed above, in assembly retrieval for top-down assembly design, the assembly interface representation in assembly model should also provide high-level design knowledge accommodated to designers besides the low-level geometric matings used in [14]. Hence we present a multi-level assembly interface descriptor (Table 1) here based on the assembly interface representation used in our top-down assembly model [16].

Function Layer	Degree of Freedom						
	Translation		Rotation		Composition		
Implementation Layer	Kinematic Pair						Interface Part
	Revolute	Prismatic	Screw	Cylindrical	Spherical	Planar	Essential
	Gear	Universal	Point on Surface	Point on Planar Curve	Surface	Fixed	Accessorial
Geometry Layer	Geometric Mating						
	Coincident	Concentric	Distance	Tangent	Perpendicular	Parallel	
	Point on Line	Point on Surface	Edge on Surface	Lock			

Table 1. The multi-level assembly interface descriptor.

There are three major levels in this descriptor: function layer, implementation layer, and geometry layer. Information in each layer is more abstract and intensive than layers below.

Function layer stores information about the design abstraction of the assembly interface. What we put here is information about the independent relative degree of freedom (DOF) between connected components of the assembly interface, i.e. counts of translational, rotational and composite DOFs. Composite DOF is the specific DOF as a result of constraining multiple DOFs together, e.g. the 1-DOF in screw joint is the effect of imposing a fixed ratio between 1-translational DOF and 1-rotational DOF.

Implementation layer describes ways selected by designers for implementing the design objective of the interface. The counts of various kinds of kinematic pairs and interface parts are stored in this layer. Generally, kinematic characteristics in most of assembly interfaces are composed of several elemental kinematic pairs from the ones we selected. Consequently six lower pairs (Fig. 6 shows sample implementations) and some typical higher pairs are chosen here. Interface parts are those specific parts which only help in implementing assembly interfaces and do not participate in any other main functions, and we divide them into two sub-types further, i.e. essential and accessorial. Essential ones are crucial for kinematic implementation (black parts of the swivel joint in Fig. 7a), while accessorial ones do not affect essential kinematic property of the assembly interface and could thus be ignored (e.g. bearing in Fig. 7b).

Geometry layer contains various kinds of geometric matings used most in assembly modeling nowadays. Counts of these geometric matings expose information about the detailed design of assembly interface based on the knowledge from confirmed high-level design.

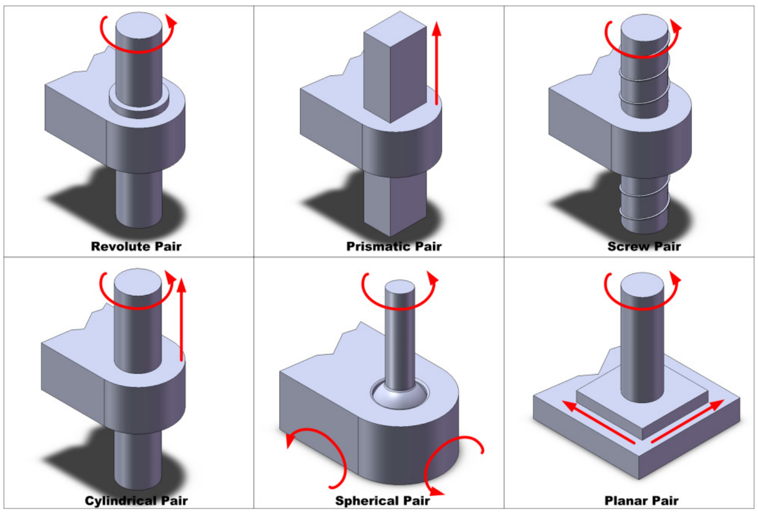

Fig. 6. Samples of the six lower kinematic pairs.

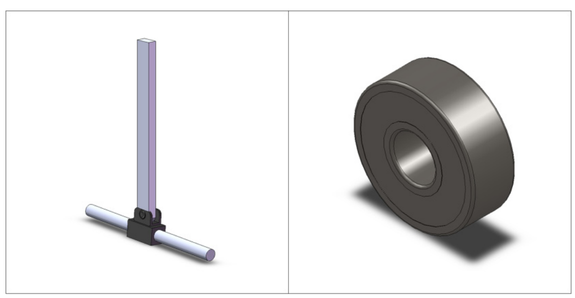

Fig. 7. Samples of the interface parts. (a) swivel joint, (b) bearing.

5.3 Definition of the assembly descriptor

In order to support the assembly retrieval mode we proposed which uses the skeleton model as query, the assembly descriptor should be capable of capturing high-level knowledge stored in skeletons. Actually, the presented multi-level assembly interface descriptor together with the hierarchical characteristics of traditional assembly model could fulfill the requirement of the skeleton based assembly retrieval. Therefore, an extended hierarchical structure based assembly descriptor is given below:

1. For an assembly, the extended hierarchical structure is based on the traditional hierarchical structure. In

this structure, tree nodes are components, while tree edges between parent and child nodes are part-of relationships, and sibling edges among child nodes of the same parent are assembly interfaces. Each non-leaf node N (an assembly or sub-assembly) implicitly contains an assembly interfaces graph in it, while the nodes in this graph are child components of N and edges are assembly interfaces among these child components.

2. The assembly interfaces are represented as the multi-level assembly interface descriptors to support not only the low-level geometric matings used in traditional assembly models, but also high-level design knowledge which are crucial in top-down assembly design.

This assembly descriptor is then used to depict both query model and target models in database, and the execution of graph matching and the similarity assessment are also based on it.

6. Graph matching and similarity assessment

6.1 Graph matching

In order to compare the query assembly interfaces graph with the assembly interfaces graphs of the assemblies in Database, an effective and efficient graph matching algorithm is necessary. There are two mainstreams of graph matching algorithms, i.e. exact matching and inexact matching [19]. In our opinion, the inexact matching algorithms may not be very suitable here because minor changes in assembly interfaces graph often exhibit major differences in functions. In other words, the topologies in assembly interfaces graph have high sensibilities with respect to function. Therefore, what we choose here is the remarkable VF2 sub-graph isomorphism algorithm [20] for its exactness and high performance.

A successful matching M (the result of graph matching) is a mapping from graph $G1$ to graph $G2$, while each node n in $G1$ is mapped to a node $M(n)$ in $G2$ and each edge e in $G1$ is also mapped to an edge $M(e)$ in $G2$. This mapping is not bidirectional, since it is validate that the query graph is subpart of the assembly graphs in database but not the vice versa. Here we let $G1'$ be the sub-graph in $G2$ which is mapped by $G1$, then in our opinion, the proportion taken up by $G1'$ in $G2$ depicts the degree of correlation between $G1$ and $G2$.

6.2 Similarity assessment

After the matching M is found, calculation of the similarity between two assembly interfaces graphs is carried on according to the following expressions:

$$\boldsymbol{Similarity}(\boldsymbol{M}) = \boldsymbol{Attenuation}(\boldsymbol{M}) * \sum_{(\boldsymbol{e},\boldsymbol{M}(\boldsymbol{e})) \in \boldsymbol{M}} \omega_e * \boldsymbol{Similarity}(\boldsymbol{e},\boldsymbol{M}(\boldsymbol{e}))$$

$$\boldsymbol{Attenuation}(\boldsymbol{M}) = (1 - \boldsymbol{k_a} * (1 - \frac{\boldsymbol{Cardinality}(\{\text{edges in source graph}\})}{\boldsymbol{Cardinality}(\{\text{edges in target graph}\})}))$$

$$\omega_e = \frac{1}{\boldsymbol{Cardinality}(\{\text{edge pairs in M}\})}$$

$$\boldsymbol{k_a} \in [0,1]$$

Here w_e is the weight of edge pair $(e, M(e))$ and we choose uniform distribution among all the weights of edge pairs, *Attenuation(M)* is used to depict the correlation degree of the source graph and the target graph under M (while M is often a non-perfect matching between the two graphs as described above). The k_a is the attenuation factor which controls the variation speed of the *Attenuation(M)*. The similarity of an edge pair is then calculated as follows:

$$\boldsymbol{Similarity}(\boldsymbol{e}, \boldsymbol{M}(\boldsymbol{e})) = \sum_{\substack{c \in \boldsymbol{Category}(\text{assembly} \\ \text{interface descriptor})}} \omega_c * \boldsymbol{Similarity}(\boldsymbol{e}_c, \boldsymbol{M}(\boldsymbol{e})_c)$$

$$\boldsymbol{Category}(\text{assembly interface descriptor}) = \{\boldsymbol{dof}, \boldsymbol{kp}, \boldsymbol{ip}, \boldsymbol{gm}\}$$

$$\sum_{\substack{c \in \boldsymbol{Category}(\text{assembly} \\ \text{interface descriptor})}} \omega_c = 1$$

This calculation is based on the multi-level assembly interface descriptor. There are four categories of values in an assembly interface descriptor as shown in Table 1. Each category has its own weight w_c and generally the category in higher layer has larger weight than the category in lower layer. The similarity of an edge pair is the weighted sum of the similarity in each category which is calculated as below:

$$\boldsymbol{Similarity}(\boldsymbol{e}_c, \boldsymbol{M}(\boldsymbol{e})_c) = 1 - \boldsymbol{Dist}(\boldsymbol{c})$$

$$\boldsymbol{Dist}(\boldsymbol{c}) = \omega_t * \sum_{t \in \boldsymbol{Type}(\boldsymbol{c})} \boldsymbol{Dist}(\boldsymbol{t})$$

$$\boldsymbol{Type}(\boldsymbol{c}) = \{\text{all types in category c}\}$$

$$\omega_t = \frac{1}{\boldsymbol{Cardinality}(\boldsymbol{Type}(\boldsymbol{c}))}$$

$$\boldsymbol{Dist}(\boldsymbol{t}) = \frac{|(\boldsymbol{e}_t - \boldsymbol{M}(\boldsymbol{e})_t)|}{\max(\boldsymbol{e}_t, \boldsymbol{M}(\boldsymbol{e})_t)}$$

Here the similarity of category c is calculated as one minus the *Dist(c)* which is the distance of values in category c. The *Dist(c)* is then the weighted sum of *Dist(t)* which is the distance of values in type t, while t is a slot in category c (e.g. type "prismatic pair" in category "kinematic pairs"). The weight w_t is defined uniformly on all types in category c. Finally e_t is the value of type t which is just a count corresponding to t, e.g. counts of prismatic pairs presented in an assembly interface (defined in section 5.2).

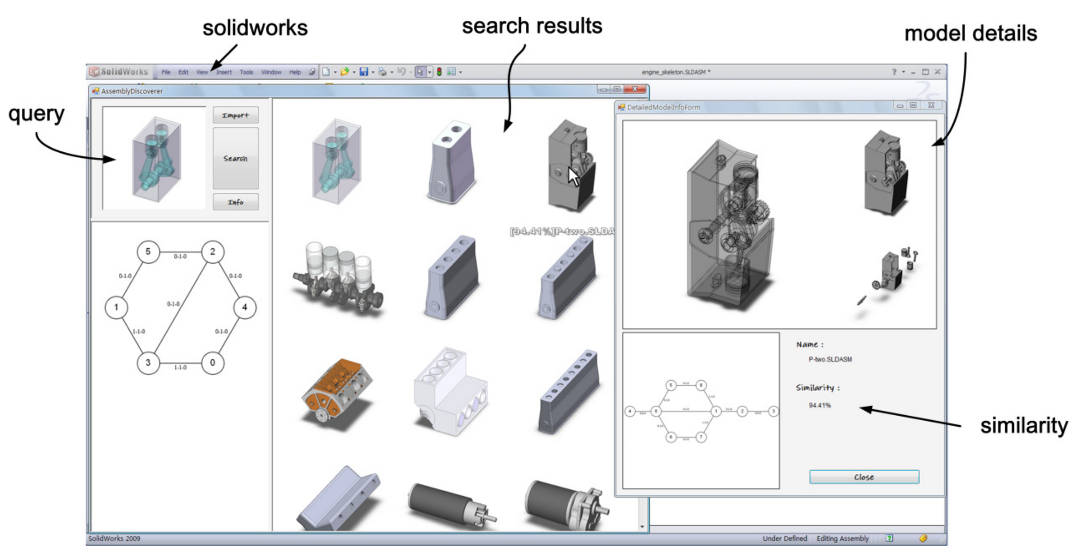

Fig. 8. The UI of the assembly retrieval system.

7. Implementation and results

The proposed assembly retrieval approach has been implemented as a multi-modules prototype system. The UI module (Fig. 8) is developed using Microsoft Visual C# 2008 and built as a plug-in of SolidWorks 2009 to interact with designers; the graph matching (implementing with VFLIB [21]) and similarity assessment module is developed using Microsoft Visual C++ 2008 and built as a win32 library which is invoked by the UI module during retrieval. Besides the two main modules, a C++/CLR module is developed as the translator to deal with the interoperability between them.

7.1 Query input

As assembly skeleton is used for query input, the high-level assembly interface knowledge could be provided by designers via the top-down assembly model based CAD system [16]. Here, different from the treating of assembly models in database, we do not consider a hierarchical structure as input because mostly designers only decompose a component one level down before considering reuse. Hence the query is a single assembly interfaces graph in which nodes are simplified shapes and edges are the multi-level assembly interfaces. Besides that, designers could decide how detailed the knowledge they provide in the assembly interface, e.g. only DOFs provided is allowed.

7.2 Database construction

Currently, our assembly library contains 106 SolidWorks assembly models (*.sldasm) downloaded from the Internet mechanical repository [22]. These assembly models are all based on the traditional representation and we need to extract the assembly descriptor from these assemblies. The key point here is the extraction of multi-level assembly interfaces. Unfortunately, the extraction of complex kinematic pairs from geometric matings and recognition of the interface parts is not a trivial problem. Currently we choose to build an assistant tool to label the multi-level assembly interfaces in the existing assemblies currently (Fig. 9).

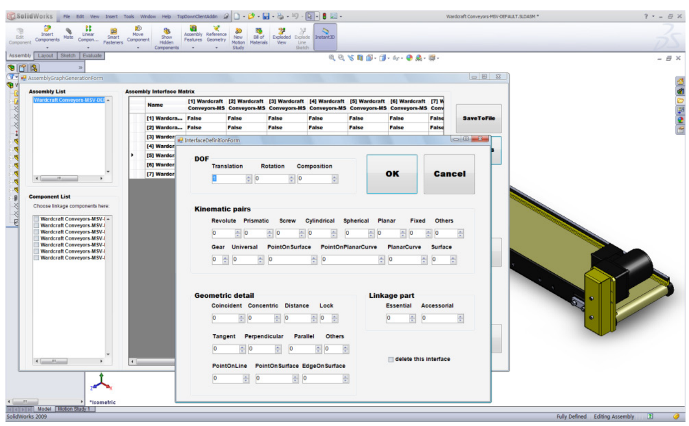

Fig. 9. The assistant tool for labeling the multi-level assembly interfaces.

As described above, the query we use has only one assembly interfaces graph, so we decompose and reorganize the extended hierarchical tree structure based assembly descriptors in database to a list of assembly interfaces graphs for convenience of graph matching during retrieval. Here each graph corresponds to one assembly or sub-assembly and represents the multi-level assembly interfaces among the child components.

7.3 Settings of the retrieval system

In order to test the performance of graph matching and similarity assessment on the assembly descriptors based on the multi-level assembly interfaces, no indices are used in the retrieval system. On the other hand, to improve the effect and accelerate the pruning of the exact graph matching (VF2), the DOFs defined in the multi-level assembly interfaces descriptors are used as the edge labels, and then the similarities are calculated by means of the values in the other three categories (kinematic pair, interface part and geometric mating).

7.4 Case studies

Here, the alterable parameters for similarity assessment in the prototype system are set as below:

$$\boldsymbol{k_a} = 0.15, \omega_{kp} = 0.7, \omega_{lp} = 0.2, \omega_{gm} = 0.1$$

Sample 1

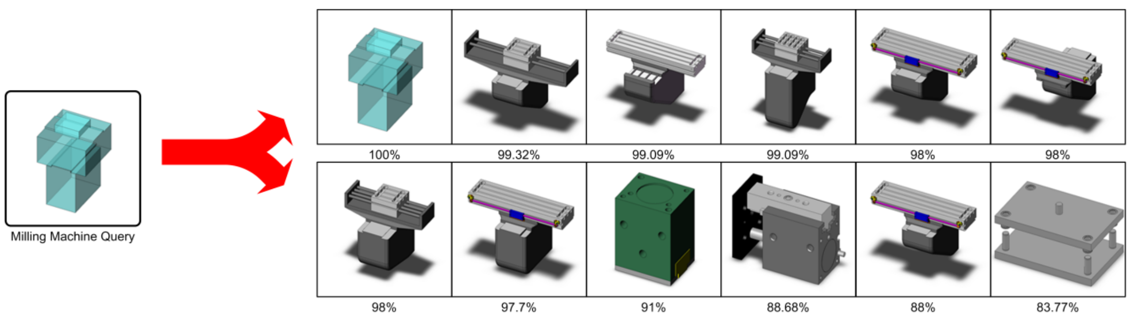

Fig. 10. Assembly search sample 1.

The first sample is the search of milling machines. The skeleton-based query is shown as the left model in Fig.10, and the 12 retrieved models with similarities higher than 80% are given in the right, including the query model itself (the queries used are also put into the assembly library for testing the identity properties of the assembly retrieval). The average search time (100 runnings) for this query is 38.75ms.

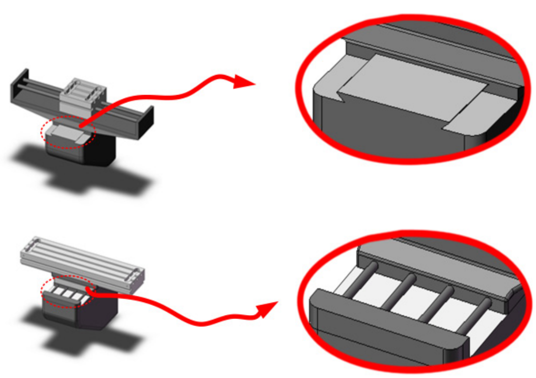

Fig. 11. Details of the search results in sample 1.

Looking into the details of the search results, the retrieved assemblies have different geometric details in assembly interfaces (Fig. 11). The upper one is a dove-tail connection while the bottom one is a compound pin-hole connection. However, the two assembly interfaces both possess one translational DOF and one prismatic pair (sliding). In other words, these two assembly interfaces have the same kinematic characteristics (both coincide with query) but different geometric matings. This case can demonstrate how the multi-level assembly interfaces descriptors act on the assembly retrieval, while low-level differences in assemblies do not exclude them from retrieval results but affect the similarities instead.

On the other hand, shapes of the three retrieved assemblies with calculated similarities 91%, 88.68% and 83.77% in Fig. 10 are not much like the query model. The reason is that we currently use only assembly interfaces information in graph matching, and this makes the assemblies with similar assembly interfaces but dissimilar component shapes be retrieved. In general, simpler topology in query graph has higher probabilities for the mentioned phenomena. Query graph with relatively complex topology in sample 2 generating almost no odd results could demonstrate this.

Sample 2

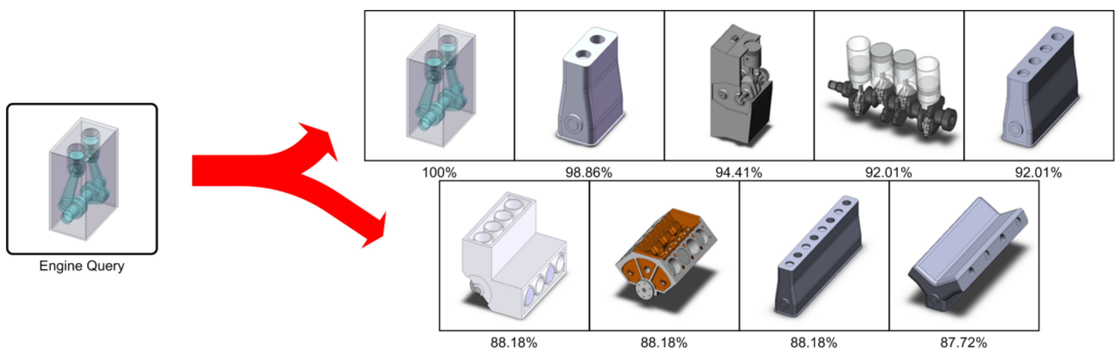

Fig. 12. Assembly search sample 2.

In this sample the skeleton of an engine model is input as a query. Fig. 12 shows the 9 search results with similarities higher than 80%. All these retrieved assemblies contain the classic piston-rod-crank mechanism. The average search time for this query (100 runnings) is 42.12ms.

Fig. 13 shows the search details that demonstrate the effect of using hierarchical assembly structure. The connecting rod in skeleton-based query is a single component while the corresponding one owned by the retrieved assembly is a sub-assembly. If hierarchical structure is not used, and then the external geometric matings for the rod sub-assembly may be defined on the “upper-rod” and the “end-cap” respectively, which would make this assembly be abandoned during retrieval.

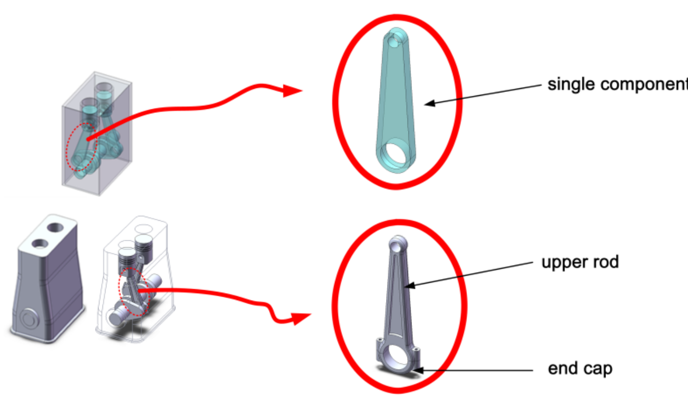

Fig. 13. Details of the search results in sample 2 (a).

Furthermore, the model in Fig. 14 indicates another benefit of the presented multi-level assembly interfaces descriptor. The tiny yellow part “pin” shown in the right is regarded as an interface part which defined in Table 1. This means that the “pin” is an inner part for implementing the assembly interface between “piston” and “rod”. Hence the disturbing geometric matings incurred by “pin” are eliminated from the assembly interfaces graph, which the assembly search method in [14] could not achieve.

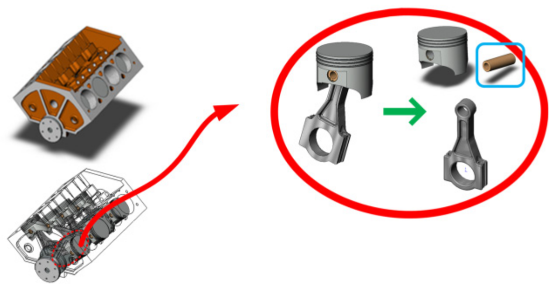

Fig. 14. Details of the search results in sample 2 (b).

8. Conclusion

In the paper a content-based assembly search method is described and implemented to retrieve reusable CAD assemblies during top-down assembly design, and an extended hierarchical assembly structure with multi-level assembly interface embedded in is also presented to capture high-level knowledge in product design better. Therefore, the skeleton query based assembly retrieval could be supported well, and similarities of models in different levels of detail are able to be differentiated and evaluated appropriately. These abilities are very important for discovering and digging out the high-level design knowledge, which is crucial to early design, from enormous existing CAD models. In all, we believe that the presented assembly search method could be a promising way to meet the requirement of knowledge-reuse in complex product design.

Several aspects of the work will be extended: a) shape-based information could be embedded in the nodes of the assembly interfaces graph to get better matching result; b) multi-level and complex query should be supported; c) a reliable and automatic approach for the extraction of high-level information in assembly interfaces from legacy models is needed.

References

[1] Mäntylä, M., A modeling system for top-down design of assembled products. *IBM J. Res. Dev.*, 1990. 34(5): p. 636-659.

[2] Csabai, A., I. Stroud, and P.C. Xirouchakis, Container spaces and functional features for top-down 3D layout design. *Computer-Aided Design*, 2002. **34**(13): p. 1011-1035.

[3] Mun, D., J. Hwang, and S. Han, Protection of intellectual property based on a skeleton model in product design collaboration. *Computer-Aided Design*, 2009. **41**(9): p. 641-648.

[4] Llewelyn, A., Review of CAD/CAM. *COMP. AIDED DES.*, 1989. 21(5): p. 297-302.

[5] Gunn, T., The Mechanization of Design and Manufacturing. *Scientific American*, 1982. 247(3): p. 114-30.

[6] Iyer, N., et al., Three-dimensional shape searching: state-of-the-art review and future trends. *Computer-Aided Design*, 2005. 37(5): p. 509-530.

[7] Li, M., et al., Toward Effective Mechanical Design Reuse: CAD Model Retrieval Based on General and Partial Shapes. *Journal of Mechanical Design*, 2009. 131(12): p. 8.

[8] Cardone, A., et al., Machining feature-based similarity assessment algorithms for prismatic machined parts. *Computer-Aided Design*, 2006. 38(9): p. 954-972.

[9] Gao, W., et al., Multiresolutional similarity assessment and retrieval of solid models based on DBMS. *Computer-Aided Design*, 2006. 38(9): p. 985-1001.

[10] Bai, J., et al. Semantic-based partial retrieval of CAD models for design reuse. in 2009 *SIAM/ACM Joint Conference on Geometric and Physical Modeling*. 2009. San Francisco, California: ACM.
[11] Bai, J., et al., Hierarchical graph generation and efficient matching for solid model similarity assessment. *Journal of Computer-Aided Design and Computer Graphics*, 2009. 21(7): p. 869-879.
[12] Deshmukh, A., et al. A system for performing content-based searches on a database of mechanical assemblies. in *ASME International Mechanical Engineering Congress & Exposition*. 2005. Orlando, FL.
[13] Gupta, S.K., A. Cardone, and A. Deshmukh, Content-based search techniques for searching CAD databases. *Computer-Aided Design and Applications*, 2006. 3(6): p. 811-819.
[14] Deshmukh, A.S., et al., Content-based assembly search: A step towards assembly reuse. *Computer-Aided Design*, 2008. 40(2): p. 244-261.
[15] Ullmann, J., An algorithm for subgraph isomorphism. *Journal of the ACM (JACM)*, 1976. 23(1): p. 31-42.
[16] Chen, X., et al. The skeleton in the multi-level assembly model for top-down innovation design of mechanical product. in 2009 *International Conference on Product Lifecycle Management (PLM09)*. 2009. Bath, England.
[17] Zhang, S.T., et al., A framework for collaborative top-down assembly design. *Proceedings of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* 2007, Vol 6, Pts a and B, 2008: p. 139-149.
[18] Lee, K. and D.C. Gossard, A hierarchical data structure for representing assemblies: part 1. *Computer-Aided Design*, 1985. 17(1): p. 15-19.
[19] Conte, D., et al., Thirty years of graph matching in pattern recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 2004. 18(3): p. 265-298.
[20] Cordella, L., et al., A (sub) graph isomorphism algorithm for matching large graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2004: p. 1367-1372.
[21] http://amalfi.dis.unina.it/graph/.
[22] http://www.3dcontentcentral.com.