



An approach to automatic adaptation of assembly models



Wanbin Pan^{a,b}, Shuming Gao^{a,*}, Xiang Chen^a

^a State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou, PR China

^b School of Media and Design, Hangzhou Dianzi University, Hangzhou, PR China

ARTICLE INFO

Article history:

Received 28 October 2014

Received in revised form 21 May 2015

Accepted 16 June 2015

Available online 4 July 2015

Keywords:

Assembly model adaptation

Shape adaptation

Kinematic semantics adaptation

ABSTRACT

Adaptation plays a fundamental role in case-based design. However, after decades of efforts, automatic adaptation is still an open issue. In works of case-based design, a designer usually chooses a start-up product model (a candidate model) of moderate complexity based on a query model possessing primary new design requirements (kinematic semantics and geometry), then achieves the target design by adapting the candidate model according to the new design requirements and human interventions are often indispensable. To smartly adapt the candidate model to fit the new design requirements, a novel approach to automatic adaptation of assembly models is proposed in this paper. First, in order to effectively identify the corresponding links and interfaces between two non-preregistered assembly models as relevant elements, an attributed kinematic graph is put forward and adopted. Second, based on the attributed kinematic graph, the kinematic semantics of the candidate model is automatically adapted to that of the query model. Third, through performing interface layout transferring, the geometry of the candidate model is automatically adapted to that of the query model based on the corresponding links and interfaces. A prototype system is also implemented to verify the effectiveness of the proposed approach.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Recently, in order to effectively reuse models to improve the efficiency and quality of product design, case-based design (CBD) attracts more and more attention [1–5], which views product model design as a process of retrieving one candidate model and adapts it to make it fit the new design requirements. Over the past decades, retrieval methods are well studied [2,3,5–11], while the works about adaptation are still immature and often human-dependent [3,12]. Since the adaptation process is an essential step in CBD while usually tedious and time-consuming, especially during routine designs, to free the designers from this unnecessary burden, automatic product model adaptation is very important for improving the efficiency of CBD.

At present, the mainstream approaches for automatic product model adaptation fall broadly into two types: parametric design approach and combination/substitution approach. The parametric design approach mainly adopts parametric method to embed the domain knowledge into product models, and uses parameter adjustment to satisfy new design requirements [13–15]. As for the

combination/substitution approach, elementary units/function units are employed to achieve target solution based on elementary unit combination or substitution [16,17].

As pointed out by [9,18], a designer in the early design stage often has just primary new design requirements, making it difficult to formulate a complete model. Therefore, it would be very helpful if the designer can create a rough query model indicating the requirements and use it to find some similar previous models with more details. This is because the previous models found, called candidate models hereafter, can inspire the designer to conduct further design and save the designer's time by reusing them in the detailed design stage. In order to effectively reuse the candidate model, one key technical issue is how to make the candidate model adapt to the more abstract query model indicating the requirements, i.e. how to effectively modify the more detailed candidate model to make it meet the primary new design requirements indicated by the abstract query model.

Generally, an ideal adaptation approach should be automatic, independent of domain knowledge library and low demand on shape similarity so as to make the adaptation approach more efficient, general and flexible. Thus, the mainstream automatic product model adaptation approaches are still far from what industries expect. For example, the parametric design approaches usually require the candidate model nearly to have the consistent geometry shapes with the new requirement/the query model. And

* Corresponding author.

E-mail addresses: panwanbin@hdu.edu.cn (W. Pan), smgao@cad.zju.edu.cn (S. Gao).

the combination/substitution approaches often need domain knowledge libraries. Furthermore, the domain knowledge is usually not provided for most of the models, which makes the mainstream approaches tend to be failed in automation.

In this paper, a new approach to automatic adaptation of assembly models is presented with a view to overcoming the above mentioned problems with automatic product model adaptation. The objective of the proposed approach is to make the assembly model adaptation automatic, independent of domain knowledge library and low demand on shape similarity. The inputs of the approach are two assembly models without having been pre-registered: a query model and a candidate model. The query model, indicating the primary new design requirements through its rough shape and its kinematic semantics, is a simple model. The candidate model is searched from a model library according to the query model, whose overall shape is similar to that of the query model but having more details and more complex kinematic semantics. Considering that the kinematic semantics is the intrinsic characteristic of an assembly model, the basic idea of the proposed approach is to realize automatic kinematic semantics adaptation of the candidate mode first, and then automatically achieve the geometry adaptation based on the result of the kinematic semantics adaptation.

The rest of the paper is organized as follows. After briefly introducing related works in Section 2, some concepts and the overview of the approach are described in Section 3. In Section 4, we identify the corresponding links and interfaces. Subsequently, the kinematic semantics adaptation and the geometry adaptation are respectively presented in Sections 5 and 6. Implementation and comparisons are introduced in Section 7. Finally, we present the conclusion and our further works in Section 8.

2. Related works

As there are a few works dedicated to automatic case adaptation particularly for automatic model adaptation, the works that we have surveyed are surrounding the two challenges for achieving automatic concept/geometry adaptation in mechanical engineering domain [3]: identifying correspondence between the two given cases/models and transferring new design requirements via the correspondence.

2.1. Identification of the correspondence between two given cases/models

2.1.1. Concept correspondence

Identifying the conceptual correspondence, between a library case and an input target problem/a query case, is often performed in the process of case retrieval in Case-Based Reasoning [2,3,5,6,10,19,20]. And, there are so many variations of case retrieval methods, such as weight distribution method [21], hybrid similarity measure method [22] and etc. Recently, ontology-based technology becomes a hot research topic for case retrieval in CBR. For example, Guo et al. [11] and Bejarano et al. [6] integrate ontology into CBR system, which make their proposed methods have enough semantic understanding ability on engineering design. However, concept correspondence is yet to provide direct geometry correspondence between two matched cases.

2.1.2. Geometry correspondence:

Generally, finding the common local areas between models is the focus of partial retrieval [8,23–25] and common design structure discovery [26,27] in engineering areas. For example, Bai et al. [8] present a hierarchical way to support multi-level partial retrieval by defining the criteria for reusable subpart of models. Besides, You et al. [23] and Tao et al. [25] also present two retrieval

methods respectively based on the local feature correspondence via IMC detection and local surface region decomposition. Additionally, Ma et al. [26] present a common design structure discovery method based on face adjacency graph which also can be used for finding feature correspondence. Besides, some global content-based retrieval methods also provide geometry correspondence implicitly or explicitly [28–33]. For example, both of the methods based on DBMS [31] and hierarchical representation for B-rep model retrieval [29] provide a level-of-detail geometric area correspondence between a query model and a candidate model. However, the correspondences brought by these partial retrieval methods are usually coarse and/or affected by geometric detail differences.

Currently, the assembly model retrieval methods usually provide high-level geometry correspondence. For example, Deshmukh et al. [7] present a mating graph to describe the relationship among the parts of an assembly model, and part correspondence between two assembly models can be obtained based on isomorphism sub-graph identification. Wu et al. [33] present a retrieval method based on product spatial layout/structure matching, where two parts has similar attributes and spatial position is deemed as a pair of corresponding parts. Additionally, Zhang et al. [27] present a generic face adjacency graph for discovering the common design structure from assembly models and bringing part correspondence as well. Based on the hierarchical structure of each assembly [34], Chen et al. [9] present a flexible assembly model retrieval method, which can bring geometry correspondence between two assembly models in topology. However, their corresponding result is more or less subjective [27] since the hierarchical structure is often flexible.

2.2. Transferring of new design requirements

Watson et al. [2] describe a survey of adaptation in CBD where they summarize that adaptation in design can be carried out in four types of approaches: human intervention, knowledge-based adaptation, case-combination adaptation and combinations of the above approaches. Avramenko et al. [12] summarize two approaches-structural adaptation and derivational adaptation, and various adaptation techniques ranging from no adaptation to case-based substitution. As we know, the kinematic semantics design is one of the most creative and important stages in mechanical design [35]. And the mainstream adaptation method for kinematic semantics design is function element synthesis [36–40], which provides a number of optional combinations of new mechanisms satisfying new design requirements. However, each function element is usually defined in advance based on domain knowledge and used with the support of a specific knowledge library.

Totally or partially, some works achieve automatic product model adaptation aided with domain knowledge. For example, Hua et al. [41] describe a prototype design system called case adaptation by dimensionality reasoning (CADRE), which uses dimensional and topological adaptation based on production rules and shape grammars. Liu et al. [14] present a case-based parametric design approach for test turntable, by using a knowledge library composed of parameterized 3D models and a matched case library composed of design specifications. Cheng et al. [15] present a similar parametric approach and applied for hydrostatic rotary table design based on Pro/Engineer. Zhang et al. [16] present a design reuse approach to realize fixture design knowledge retrieval and fixture model retrieval based on ontology. Their approach adopts evolutionary methods to modify the retrieved model to meet the new design requirements. Different from most of the traditional CBD approaches for architectures, Hua [17] presents a new 3D architecture design approach as similar as modeling by examples. However, geometry adaptation is often

carried out by humans when the domain knowledge is insufficient [3].

3. Basic concepts and approach overview

Before giving the overview of our approach, some basic concepts are first introduced.

3.1. Basic concepts

Although link is often used to represent an abstract kinematic unit [35], in order to distinguish from the assembly component [9,34] whose content rely on the hierarchical structure of its assembly model, one part or more members connected together such that no relative motion can occur among them is considered as one link in this paper, such as part A' and sub-assembly B' are two links in Fig. 1.

3.1.1. Link layout

The link layout, as the rotation invariant characteristics, describes how the links of an assembly model are arranged in the 3D space. Furthermore, the link layout, representing the fundamental characteristic of instantiating a mechanism, is the key geometric information of each assembly model [42–47].

3.1.2. Corresponding link

Let QM and CM be two assembly models having the same kinematic semantics and similar link layouts. If two shape-similar links respectively belonging to QM and CM have the same kinematic semantics context and similar link layouts (relative to other links in their own models), then they compose a pair of corresponding links. For example, in Fig. 1, links A , B and C respectively correspond to links A' , B' and C' .

3.1.3. Consistent geometry

If QM and CM are two shape-similar assembly models having the same kinematic semantics and the link layouts while the links between the two models are one–one mapping, then we call them having consistent geometries. For example, QM and CM in Fig. 1 have consistent geometries.

We adopt $JT \langle L_1, L_2 \rangle$ to represent a joint JT between two contacted links L_1 and L_2 having relative motion. JT can be a cylindrical joint, a revolute joint and etc.

3.1.4. Corresponding joint

Let $J_1 \langle L_1, L_2 \rangle$ and $J_2 \langle L_1', L_2' \rangle$ be two joints respectively belonging two assembly models. If L_1 corresponds to L_1' and L_2 corresponds L_2' (or L_1 corresponds to L_2' and L_2 corresponds to L_1') while J_1 and J_2 have the same joint type, then J_1 and J_2 compose a pair of corresponding joints, such as cylindrical joints $\langle C, B \rangle$ and $\langle C', B' \rangle$ compose a pair of corresponding joints in Fig. 1.

3.1.5. Interface

An interface of a link is a place where the link contacts with another link. Matching along with the state-of-the-art methods for automatic deducing kinematic semantics focusing on low joints, at present, each interface is composed of adjacent face(s) in this

work. For example, the interface on link C contacting to B is represented as $\langle I \rangle$ shown in Fig. 2. Here, face I is an interface face.

3.1.6. Interface layout

The interface layout, as the rotation invariant characteristics, describes how the interfaces of a link are arranged on the link.

3.1.7. Corresponding interface

Let $J_1 \langle L_1, L_2 \rangle$ and $J_2 \langle L_1', L_2' \rangle$ be two corresponding joints. If L_1 corresponds to L_1' and L_2 corresponds L_2' , then the interface on L_1/L_2 contacting to L_2/L_1 and the interface on L_1'/L_2' contacting to L_2'/L_1' compose a pair of corresponding interfaces. To properly reduce the complexity, we assume that each pair of corresponding interfaces has similar shapes. For example, in Fig. 2, interfaces $\langle I \rangle$ and $\langle I' \rangle$ compose a pair of corresponding interfaces.

3.1.8. Face layout

The face layout, as the rotation invariant characteristics of a face set, is defined as the relative positions and relative orientations among the faces belonging to the same face set [48].

3.1.9. Corresponding interface face

Let QI and CI be a pair of corresponding interfaces. If two interface faces respectively belonging to QI and CI has the most similar face layouts (relative to other interface faces in their own links), then we call them a pair of corresponding interface faces. For example, in Fig. 2, faces I and I' compose a pair of corresponding interface faces. To properly reduce the complexity, at present, we assume that each pair of corresponding interface faces has the same face type.

3.2. Approach overview

In order to effectively achieve the reuse of product models in CBD, we propose an approach to automatic adaptation of assembly models by overcoming the two common challenges for automatic geometry adaptation [3]: identifying correspondence between the two given models and transferring new design requirements via their correspondence. The inputs of our approach are two non-registered assembly models: a query model and a candidate model. The query model, used to indicate the primary new design requirements through its kinematic semantics and geometry, is an abstract model created by the designer. The candidate model, having a more complex kinematic semantics and holding more geometric details than the query model, is searched from a product model library according to the query model. The two models have similar kinematic semantics. Considering that the core of an assembly model is its mechanism and the kinematic semantics of the mechanism consists of the assembly's links and their contacting interfaces [9,35], we choose links and interfaces as relevant elements to build correspondence and transfer the primary new design requirements to achieve the automatic adaptation of assembly models.

Regarding the first challenge that how to automatically identify the correspondence between the two models, in view that kinematic semantics implied in each assembly model is the intrinsic characteristic of the assembly model, we first extract the

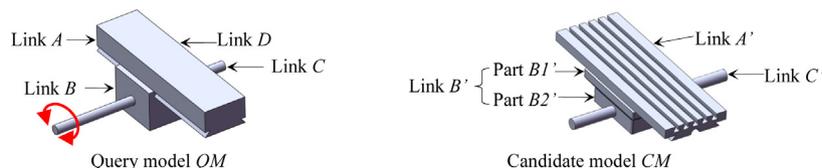


Fig. 1. Illustration for concepts related to link.

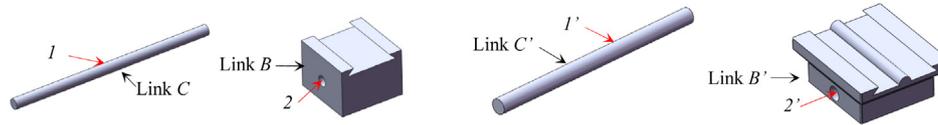


Fig. 2. Illustration for concepts related to interface (1,2, 1' and 2' represents four faces).

kinematic semantics from the query model and candidate model and set up the attributed kinematic graph for each model, then identify the corresponding links and interfaces by performing maximum isomorphic sub-graph matching between the two attributed kinematic graphs.

As for the second challenge that how to automatically transfer the primary new design requirements from the query model to the candidate model via their correspondence, we first carry out automatic kinematic semantics adaptation by adapting the kinematic semantics of the candidate model to that of the query model based on their attributed kinematic graphs and corresponding links & interfaces; then we achieve automatic geometry adaptation by automatically transferring the link layout of the query model to the candidate model based on the interface layout transferring among the corresponding links and interfaces.

The flowchart of our approach is shown in Fig. 3.

4. Identification of corresponding links and interfaces

In view that links and interfaces are the elements with semantics of the assembly model, we identify the high-level geometry correspondence (corresponding links and interfaces) between two non-registered assembly models based on their semantic models to make the identification more precise. Furthermore, in order to improve the efficiency of the identifying process, we extend the traditional semantic model of an assembly model, i.e. kinematic graph consisting of the kinematic topology of an assembly model, by incorporating the link layout of an assembly model with the kinematic graph to form an attributed kinematic graph. Before describing our method, the attributed kinematic graph is first defined below.

4.1. Attributed kinematic graph

As shown in Fig. 4, each attributed kinematic graph of an assembly model $AKG=(N, E)$ is composed of nodes N and edges E . We integrate geometric attributes to the AKG as follows:

- 1) Each node, corresponding to a link L , is assigned with the shape distribution vector of L [49].
- 2) Each edge, corresponding to a joint $JT \langle L_1, L_2 \rangle$, has two kinds of attributes: multilevel information [9] shown in Appendix A and looselink layout.

- The multilevel information is deduced from the assembly constraints between L_1 and L_2 [50,51], such as the degree-of-freedom (DOFs). This kind of attribute is used to carry out a flexible graph matching since the two input assembly models have inconsistent geometries.
- The loose link layout represents an approximate spatial arrangement among the links using the relative orientation and position between each pair of contacting link. The relative orientation is the intersection angle between the two links' motion directions while the relative position is the normalized distance between the two links' geometry centers [33].

Though the main process of generating attributed kinematic graphs is similar as described in previous work [9], we deduce each link for an assembly model without considering its hierarchical structure, which makes our high-level geometry correspondence more objective. Furthermore, all of the geometry attributes attached to each AKG are rotation invariant, which makes our attribute comparison more reasonable in graph matching between two AKG s respectively corresponding to two non-registered assembly models. For a concise visualization, we only label the DOF on each edge in this work, such as shown in Fig. 4b.

4.2. Correspondence determination based on maximum isomorphic sub-graph searching

After constructing the AKG s for two assembly models, their corresponding links and interfaces can be obtained by using a sub-graph isomorphism method. In view that it is difficult to know which part(s) of the AKG of the query model should be matched in the AKG of the candidate model in advance, we, expecting to reuse model maximization, choose the maximum isomorphic sub-graphs between the two AKG s as their optimized sub-graph matching result. Moreover, to accelerate maximum isomorphic sub-graph searching, we integrate the following heuristic rules into the common sub-graph isomorphism method [52]:

- 1) Two matched nodes must have a similar shape that the distance between the two shape distribution vectors of the two matching nodes must be less than α (α is 0.3 here), which facilitates the subsequent shape adaptation.
- 2) Two matched edges must have the same semantics (DOF and kinematic joint) while their assembly constraints have overlap; the distances respectively for their relative orientations and

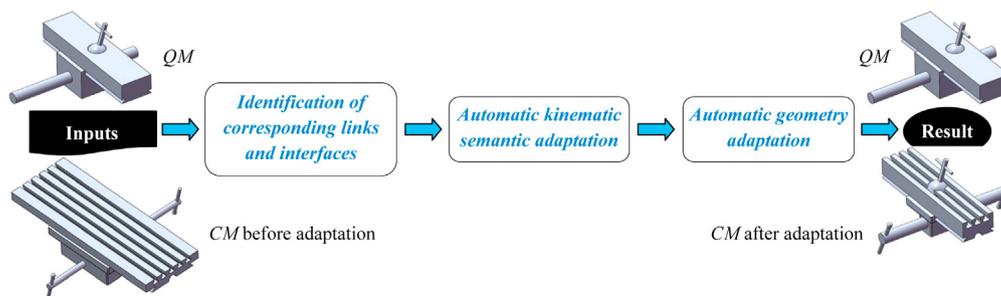


Fig. 3. Flowchart of the proposed approach.

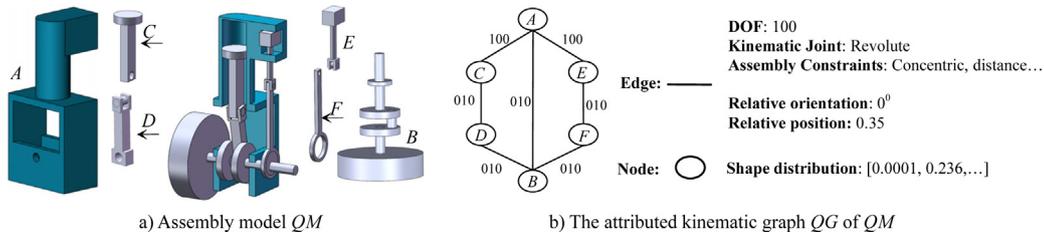


Fig. 4. An example of attributed kinematic graph.

relative positions must be less than $r1$ and $r2$ ($r1 = 5^\circ$, $r2 = 0.1$ in this work), respectively.

Based on the above heuristic rules, each pair of the found maximum isomorphic sub-graphs between two AKGs not only has the same kinematic semantics, but also possesses similar link layouts. Besides, we use the accumulation of the relative orientation distances and relative position distances between all pair of matched edges of two isomorphic sub-graphs as the two sub-graphs' link layout distance. Then, according to the definition of corresponding links, we choose the pair of the found maximum isomorphic sub-graphs having the minimum link layout distance as the optimized matching result.

According to the definitions of corresponding links and interfaces, each pair of matched nodes represents two corresponding links while each matched edge represents a pair of corresponding joints, i.e. two pairs of corresponding interfaces. For example, in Fig. 5c, link C corresponds to link C' since nodes C and C' are matched nodes while the interfaces respectively on links C and C' are two corresponding interfaces as shown in Fig. 5d.

5. Automatic kinematic semantics adaptation

Since the kinematic semantics design is usually a key design step for assembly model [9,35], it is reasonable to start processing assembly model adaptation from kinematic semantics adaptation. In order to effectively reuse the kinematic semantics of an

assembly model (by reusing its links and interfaces), we achieve automatic kinematic semantics adaptation by automatically adapting the semantic model of the candidate model to that of the query model based on their corresponding links and interfaces to make the adaptation independent of knowledge library and more general. Moreover, to make the adaptation effectively reuse the link layout of an assembly model, we adopt the attributed kinematic graph (AKG) as the semantic model.

Besides, to improve the efficiency of the adaptation process, we adopt a heuristic graph processing method by revising the AKG of the candidate model through amending of inconsistent kinematic semantics and incorporating of new required kinematic semantics. Details are now described.

5.1. Amending of inconsistent kinematic semantics

As the links and interfaces are the elements with semantics of an assembly model, the links and interfaces in the candidate model, which have no counterparts in the query model, represent the kinematic semantics of the candidate model that is inconsistent with that of the query model. In order to reuse the candidate model's semantics as much as possible, we amend the inconsistent kinematic semantics of the candidate model by using the following two different methods according to the structure of the AKG of the model. Here, let cn_1 and cn_2 be the two nodes in the AKG of the candidate model respectively matching with nodes qn_1 and qn_2 in the AKG of the query model.

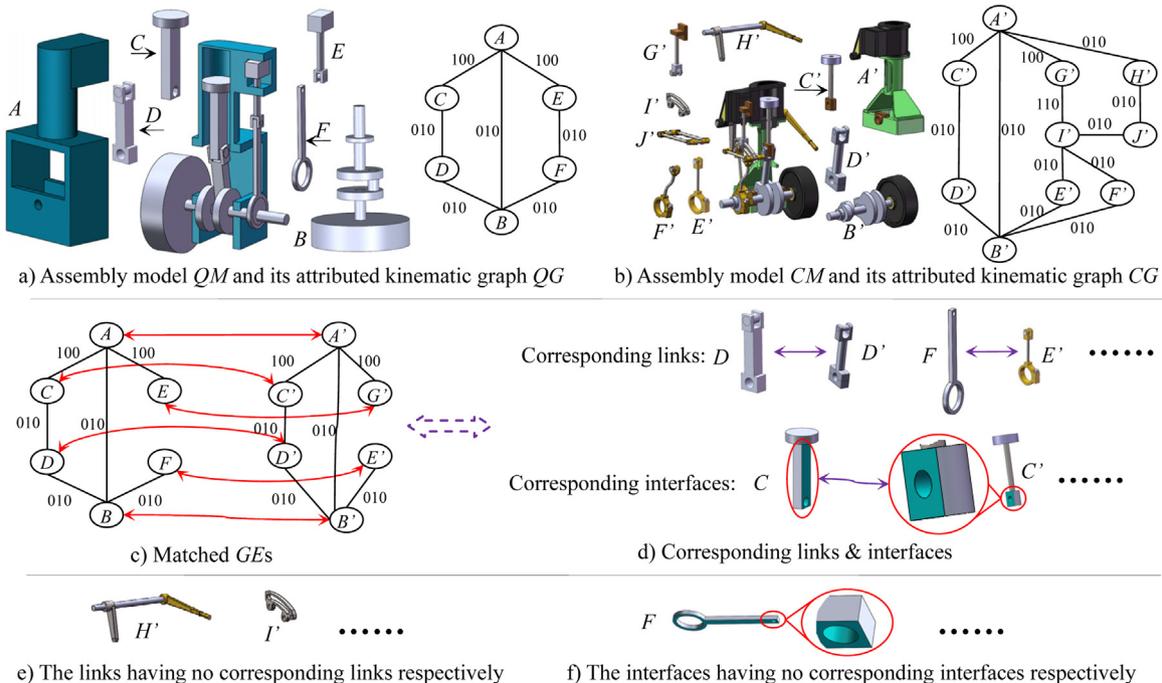


Fig. 5. Corresponding links and interfaces between QM and CM (GEs: graph elements; interface faces are in cyan).

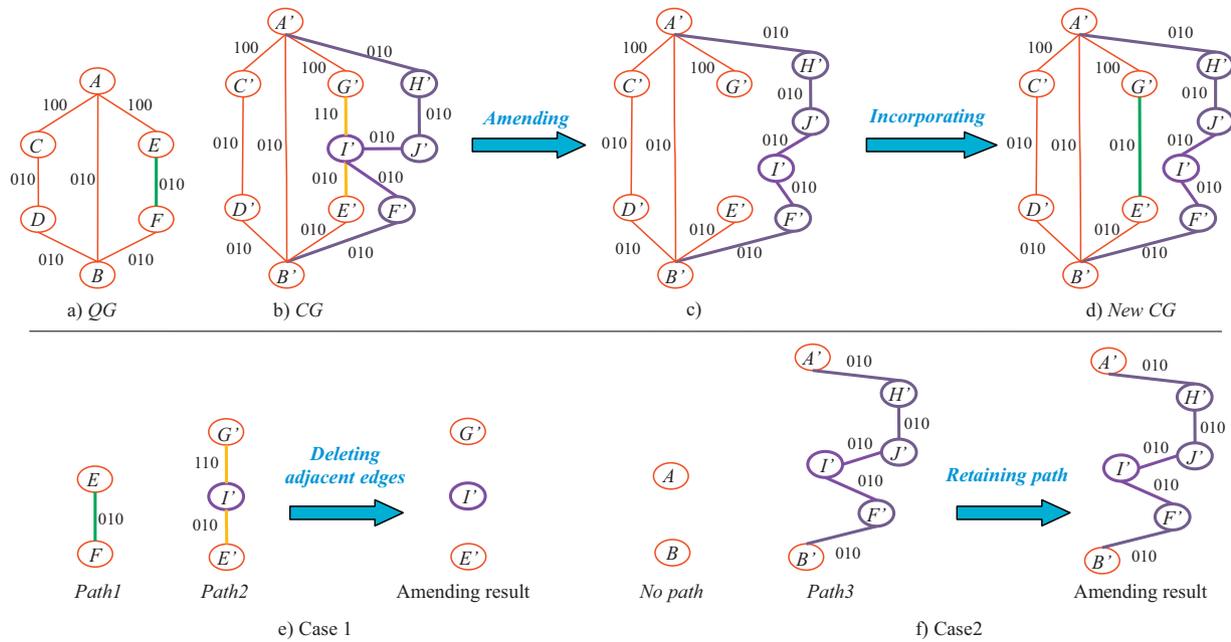


Fig. 6. An example of automatic kinematic semantics adaptation (QG and CG come from Fig. 5; each red graph element refers to a matched graph element while the other colored graph elements refer to non-matched graph elements; the red and green graph elements represent the consistent and new required kinematic semantics respectively while the orange and purple graph elements represent inconsistent kinematic semantics).

1) Deleting conflictive kinematic semantics: After ignoring all the matched edges, if there exists a path CL (none edge/node repeated) between cn_1 and cn_2 in the AKG of the candidate model, and there also exists a path QL (none edge/node repeated) between qn_1 and qn_2 in the AKG of the query model, then the non-matched edges connecting cn_1 and cn_2 in CL should be deleted from the AKG of the candidate model because this usually indicates that the two models have conflictive kinematic semantics between their corresponding link pairs. And each of these deleted non-matched edges corresponds to a deleted inconsistent kinematic semantics. In order to reuse the candidate model's semantics as much as possible, the other graph elements of the path will be retained.

For example, after ignoring all the matched edges in Fig. 6a and b (red edges), Fig. 6e shows two paths: *Path1* from node E to node F in QG and *Path2* from node G' to node E' in CG. Here, E and F are matched with G' and E' respectively. Obviously, the kinematic semantics between the two pairs of matched nodes are different. In order to conveniently replace the kinematic semantics between G' and E' with the one existing between E and F , the orange edges adjacent to G' and E' are deleted from *Path2* (also from CG) while retaining node I' in CG as shown in Fig. 6e&c.

2) Retaining certain inconsistent kinematic semantics: After ignoring all the matched edges, if there exists a path CL (none edge/node repeated) between cn_1 and cn_2 in the AKG of the candidate model, while there has no path between qn_1 and qn_2 in the AKG of the query model, then CL should be retained in the AKG of the candidate model because this usually indicates that the candidate model has richer kinematic semantics than the query model between their corresponding link pairs. And each of these retained non-matched edges/nodes corresponds to a retained inconsistent kinematic semantics.

For example, after ignoring all the matched edges in Fig. 6a and b (red edges), Fig. 6f shows that there is no path between nodes A and B in QG while there exists path *Path3* between nodes A' and B' in CG. Here, A and B are matched with A' and B' respectively. Since

there is no conflictive kinematic semantics between these two corresponding link pairs, the purple path between nodes A' and B' is retained in CG as shown in Fig. 6f and c in order to reuse the candidate model's semantics as much as possible.

Furthermore, if an edge not only corresponds to a deleted inconsistent kinematic semantics but also corresponds to a retained inconsistent kinematic semantics, then we choose to delete it by default in order to avoid confliction with the primary new design requirements, such as the case in Fig. 14.

5.2. Incorporating of new required kinematic semantics

On the other hand, the links/interfaces in the query model, which have no counterparts in the candidate model, represent the new required kinematic semantics that the candidate model should possess. In order to incorporate the above new required kinematic semantics into the semantic model of the candidate model, we incorporate each graph element which corresponds to a link/interface having no counterpart from the AKG of the query model to that of the candidate model based on the two models' corresponding links. For example, in Fig. 6a, the green edge, corresponding to two contacted interfaces respectively belonging to links E and F as shown in Fig. 5c and f, is the new required kinematic semantics. Since links G' and E' respectively correspond to links E and F as shown in Fig. 5c, the green edge is incorporated into the final CG by connecting nodes G' and E' as shown in Fig. 6d.

Obviously, the proposed automatic kinematic semantics adaptation method is a general method, which can automatically adapt kinematic semantics between two assembly models without any further design knowledge; changing the attributes of the semantic model can make it feasible for other kinematic semantics designs. Additionally, Fig. 6d demonstrates the target kinematic semantics that CM will be had after its adaptation.

6. Automatic geometry adaptation

Although there are many methods for geometry adaptation [2,3,12], the common standard of automatic geometry adaptation

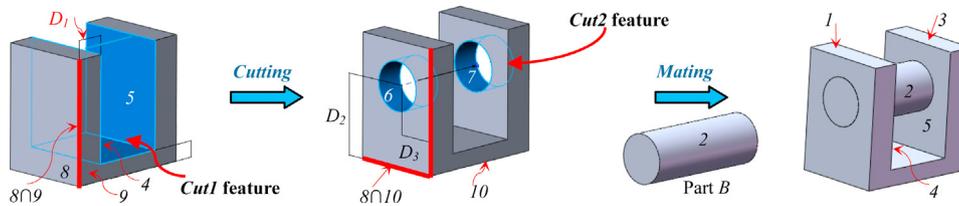


Fig. 7. An example of the construction history of an interface (the interface is composed of faces 2, 4 and 5).

is still rare especially for assembly models. As a result, the automatic geometry adaptation in this work is to automatically adapt the candidate model to have the consistent geometry with the query model around their corresponding areas (composed of corresponding links and interfaces). In order to make the adaptation automatic and low demand on shape similarity, we make full use of the result of the two models' automatic kinematic semantics adaptation. Accordingly, the automatic geometry adaptation contains the following two steps: (1) adapting the candidate model to make it include the kinematic semantics of the query model; (2) transferring the link layout from the query model to the candidate model.

According to the above sense, we carry out the first step on the candidate model by deleting its interfaces corresponding to the deleted inconsistent kinematic semantics while incorporating the links and interfaces corresponding to the new required kinematic semantics from the query model. As for the second step, considering that interfaces are the contacting constraints imposed on a link with respect to other links in an assembly model, i.e. the interface layout on each link directly corresponds to the model's link layout, we solve the link layout transferring between two assembly models based on interface layout transferring between each pair of their corresponding links to make the link layout transferring more effective. Details are now explained.

6.1. Geometry adaptation for including the query model's kinematic semantics

Since deleting interfaces from an assembly model can be simply achieved by part deleting and/or feature suppression [53], we focus on link and interface incorporation in this subsection.

In view that each link is often composed of parts, we incorporate each link by duplicating its parts along with their inner assembly constraints from the query model to the candidate model. And, the original link and the duplicated link compose a new pair of corresponding links. This incorporation is simple and not the focus of this work.

Considering that each interface is usually generated by feature attaching and/or part assembling, we carry out interface incorporation/duplication by duplicating the features/parts of an interface from one link to its counterpart. And, the original interface and the duplicated interface also compose a new pair of corresponding interfaces. Especially, if an interface is on a link that represents a new required kinematic semantics, then the interface is automatically duplicated along with the link duplication, otherwise, in order to improve the efficiency of the interface duplication, we use a corresponding geometry element identification method to automatically determine the location geometry elements on one link for precisely and automatically placing a feature/part duplicated from the other link.

6.1.1. Interface duplication based on corresponding location geometry element identification

Let QL and CL be a pair of corresponding links respectively belonging to the query model and the candidate model; QI is an

interface needed duplicating from QL to CL . The main algorithm for interface duplication is described as follows.

Step 1: Identifying all the location geometry elements in QL for QI .

The location geometry elements, which constrain the arrangement of an interface in a model, can be divided into two types roughly according to the construction history of the model [54]:

- 1) **Outline faces** which affect the arrangement of each interface on a model by affecting the model's global size. Here, we choose the faces nearest the OBB [55] of each model as its outline faces, such as the faces 1 and 2 are two outline faces in Fig. 7.
- 2) **Feature location elements** which support feature attaching for an interface generation. As shown in Fig. 7, edge $8 \cap 9$ constrains the layout of *Cut1* feature on face 9 by dimension D_1 , thus, edge $8 \cap 9$ and face 9 are two feature location elements. Similarly, since *Cut2* feature is generated on face 8 for locating part B, face 8 is also a feature location element.

Step 2: Identifying all the corresponding location geometry elements in CL for placing the duplicated QI .

Using adjacent face(s) to represent each location geometry element except face, we identify corresponding location geometry elements between QL and CL through identifying their corresponding faces [48]. For example, in Fig. 8, the location geometry elements of interface $\langle 1, 2, 3 \rangle$ on link L_1 are edge $4 \cap 5$, face 9 and etc. And, the input face permutation for the method [48] can be $\{4, 5, 9, \dots\}$, then the corresponding output face permutation is $\{4', 5', 9' \dots\}$. So faces 4, 5 and 9 respectively correspond to faces $4'$, $5'$ and $9'$. After that, all the corresponding location geometry elements can be determined. For example, edge $4' \cap 5'$ is identified as the corresponding location edge of $4 \cap 5$ to place the duplicated *Cut1* feature on link L_2 for duplicating interface $\langle 1, 2, 3 \rangle$.

Step 3: Global shape adaptation.

In order to accurately place a duplicated interface, a global shape adaptation of CL should be fulfilled in advance. For example, in Fig. 8, to make the duplicated *Cut1* feature have the same effect in L_2 as its original *Cut1* feature imposed on face 5 in L_1 , the two links should have consistent global shapes. Here, based on corresponding outline faces, we adopt a face layout transferring method described in [48] (transferring global shapes between two shape-similar models) to make the CL have a consistent global shape as QL . Such as new L_2 has a consistent global shape as L_1 in Fig. 8d and b after global shape adaptation.

Step 4: Interface duplication.

After identifying all the corresponding location geometry elements between QL and CL , QI can be automatically duplicated to CL through features/parts duplication, which is supported by most of the mainstream 3D CAD systems. Besides, if interface duplication contains parts duplication, then the assembly constraints related to these parts are also duplicated. For example, in Fig. 8e, the assembly constraints between face 2 and faces 6 and 7 are also duplicated along with the duplication of part B and used to constrain face $2'$ and faces $6'$ and $7'$.

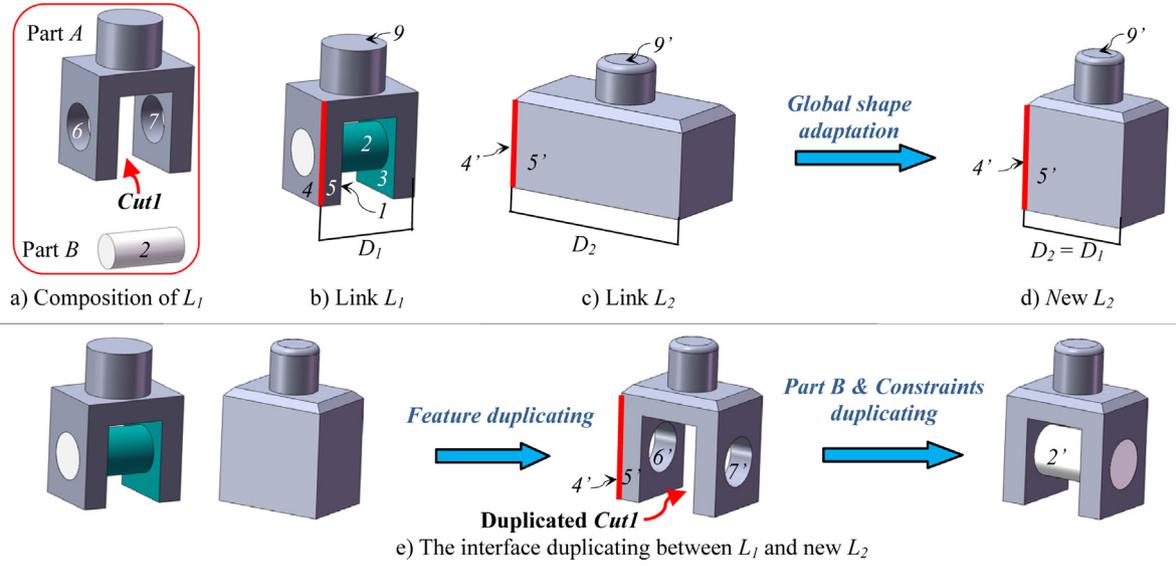


Fig. 8. An example of interface duplicating (L_1 corresponds to L_2 ; faces 1&3 on part A belong to *Cut1* feature and face 2 is on part B; faces 4, 5 and 9 on link L_1 respectively correspond to faces 4', 5' and 9' on link L_2).

Besides, since each interface is used to contact one link with another, the interface incorporation should also duplicate the assembly constraints between each pair of the contacted original interfaces to their duplicated ones. After building corresponding geometry element relationships between a duplicated interface and its original interface [56–60], the assembly constraint duplication can be achieved directly. Additionally, Fig. 8e demonstrates the process of interface <1, 2, 3> duplication from L_1 to L_2 .

6.2. Link layout transferring based on interface layout transferring

With the above geometry adaptation, each link/interface in the query model has a one–one mapping counterpart in the candidate model. In view that each promoted dimension between an interface's geometry element (*IGE*) and its location geometry elements determines the layout of an interface on a link [48], we transfer the interface layout of a link to its corresponding link by establishing relationships between the promoted dimensions of the two links to make the interface layout transferring more accurate. Furthermore, in order to effectively establish dimension relationships between two links, probably built in different parametric ways respectively, we adopt a 3D dimension constraint graph [48] to uniformly represent all promoted dimensions in each link.

6.2.1. Corresponding location geometry element identification based on temporary coordinate systems

In accordance with the kinematic semantics adaptation, each pair of corresponding links has a similar interface layout, so it is reasonable and sensible to use face layout similarity to identify the corresponding location geometry elements between each pair of corresponding links as similar as Section 6.1.1.

According to the construction history of an interface, the location geometry elements of an *IGE* include not only the outline faces and the feature location elements of its interface, but also other *IGEs* having constraints with it. In order to accelerate the efficiency of corresponding location geometry element identification between two corresponding links, we adopt temporary corresponding coordinate systems to evaluate the layout similarity between two faces. Here, we still use *QL* and *CL* as a pair of corresponding links respectively belonging to the query model and the candidate model. The main algorithm is described as follows.

Step 1: Identifying all the location geometry elements in *QL* (as similar as Section 6.1.1).

Step 2: Identifying all the corresponding *IGEs* between *QL* and *CL* based on their corresponding interface faces identification (as similar as Section 6.1.1).

Step 3: Constructing all temporary corresponding coordinate systems (*CCSs*) between *QL* and *CL*.

Each pair of corresponding coordinate systems is built in two steps: first choose two interface faces F_1 and F_2 in *QL* to build a non-repeated temporary Cartesian coordinate system $X_1Y_1Z_1$ based on the three geometry centers respectively belonging to F_1 , F_2 and *QL*; then build a corresponding coordinate system $X'_1Y'_1Z'_1$ on link *CL* based on the interface faces F'_1 and F'_2 respectively corresponding to F_1 and F_2 , such as the case shown in Fig. 9.

Moreover, each face F in a built coordinate system has a unique representation: a 4-dimension vector $(dF, \angle OFX, \angle OFY, \angle OFZ)$. dF is the normalized geometry center distance between F and its link; $\angle OFX$, $\angle OFY$ and $\angle OFZ$ respectively represent the intersection angles between vector OCF and the X axis, between OCF and the Y axis and between OCF and the Z axis. Here, OCF originates from the system origin and ends in the geometry center of F .

Step 4: Evaluating the layout similarity between two faces in each pair of *CCSs*;

In function (6-1), f'_1 and f'_2 are two interface faces in *CL* respectively corresponding interface faces F_1 and F_2 in *QL*; $\text{Position_distance}(F_i, f_j)$ and $\text{Orientation_distance}(F_i, f_j)$ are respectively used to evaluate the relative position distance and relative orientation distance between F_i and f_j (F_i belongs to *QL* and f_j belongs to *CL*); $\text{Similarity}(F_i, F_1, F_2, f_j, f'_1, f'_2)$ represents a rotation invariant layout similarity between F_i and f_j in a pair of *CCSs* that respectively built based on F_1 and F_2 and f'_1 and f'_2 .

$$\text{Position_distance}(F_i, f_j) = \left\| dF_i - df_j \right\| + \left\| \angle OF_i X_1 - \angle OF_j X'_1 \right\| + \left\| \angle OF_i Y_1 - \angle OF_j Y'_1 \right\| + \left\| \angle OF_i Z_1 - \angle OF_j Z'_1 \right\|$$

$$\text{Orientation_distance}(F_i, f_j) = \left\| \angle F_i F_1 - \angle f_j f'_1 \right\| + \left\| \angle F_i F_2 - \angle f_j f'_2 \right\|$$

$$\text{Similarity}(F_i, F_1, F_2, f_j, f'_1, f'_2) = \text{Type}(F_i == f_j) \times \left(\frac{\beta_1}{e^{\text{Position_distance}(F_i, f_j)}} + \frac{\beta_2}{e^{\text{Orientation_distance}(F_i, f_j)}} \right) \quad (6-1)$$

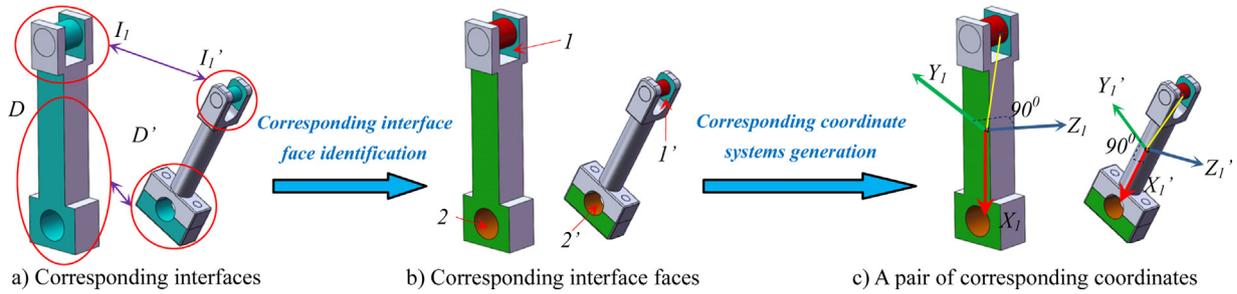


Fig. 9. An example of corresponding coordinate systems (link D corresponds to link D' in Fig. 5d; each pair of corresponding interface faces is in the same color except gray in b&c; coordinate systems $X_i Y_i Z_i$ and $X_i' Y_i' Z_i'$ are respectively built based on faces 1&2 and faces 1'&2').

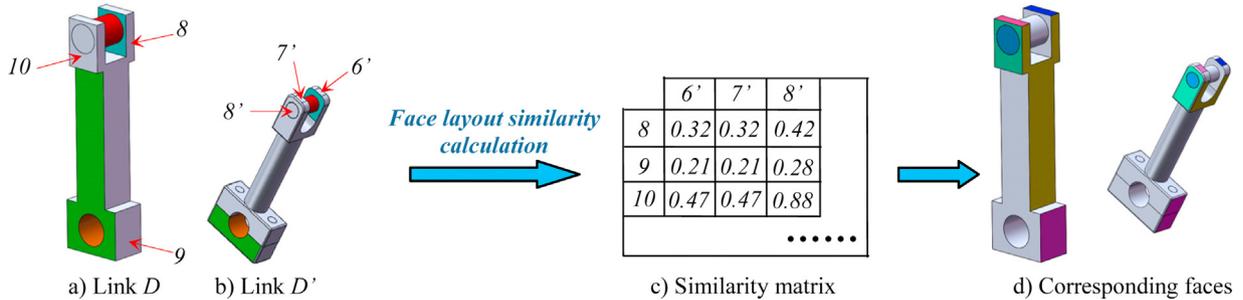


Fig. 10. The process of determining corresponding location faces between each pair of corresponding links.

Here, we use the weighted average of all the layout similarities between F_i and f_j in all pairs of CCSs between QL and CL as the two faces' final face layout similarity. Additionally, as shown in Fig. 10c, we use a similarity matrix to represent all the face layout similarities between all pairs of the faces respectively belonging to the two links.

Step 5: Determining corresponding location faces using greedy algorithm.

For example, in Fig. 10d, the faces in the same color respectively belonging to links D and D' are corresponding location faces except the gray ones. After identifying all the corresponding location faces, all the other corresponding location geometry elements can be determined as similar as the method described in Section 6.1.1.

6.2.2. Interface layout transferring based on constraint-resolving

After identifying all the corresponding location geometry elements and IGEs between QL and CL , for each pair of corresponding interface's geometry elements $QIGE$ and $CIGE$ respectively belonging to QL and CL , we establish the dimension

relationships between the promoted dimensions that respectively constrain the distance/orientation between $QIGE$ and its location geometry elements and between $CIGE$ and its location geometry element [48].

Finally, we adopt a constraint-resolving method [61] to make the parametric information transfer from QL to CL . Then, CL updates its shape by updating each dimension with the new incoming value to make CL have the same interface layout as QL , such as the case shown in Fig. 11.

After transferring the interface layout from each link in the query model to its corresponding link in the candidate model, the candidate model has the consistent geometry with the query model around their corresponding areas and the process of automatic geometry adaptation is also finished. However, if there are not enough corresponding location geometry elements or interface's geometry elements between two corresponding links, then human intervention is needed to make a desired interface transferring/interface layout transferring result, such as the case shown in Fig. 14.

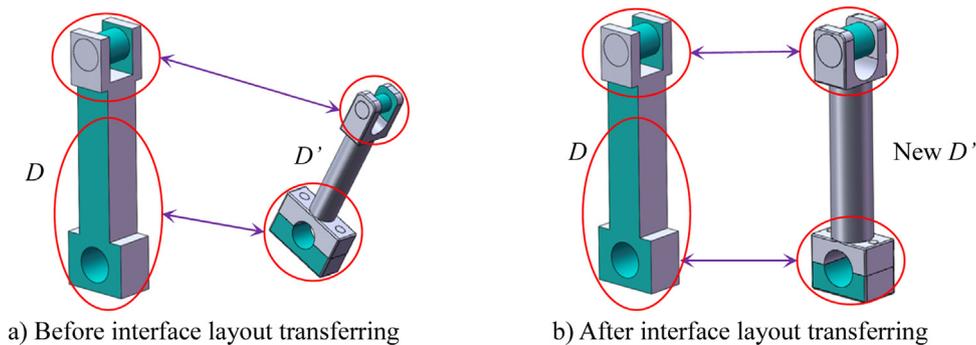


Fig. 11. An example of interface layout transferring.

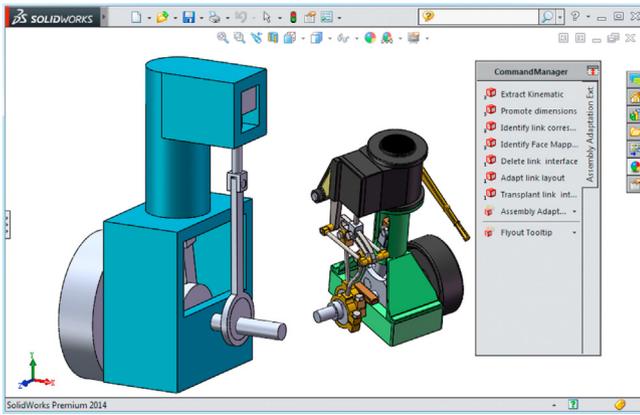


Fig. 12. User interface of our prototype OSAAD.

7. Implementation and comparisons

The proposed automatic adaptation approach has been implemented in OSAAD prototype as shown in Fig. 12, which is developed by using Microsoft Visual C# 2008 and built as a plug-in of SolidWorks 2014 to interact with designers [47]; the maximum isomorphic sub-graph searching is developed by using C++ based on the boost graph library [52] and built as a win32 library. And various assembly models have been used to test our approach.

7.1. Typical examples

7.1.1. Example 1: a regular automatic assembly model adaptation process

We use the models as shown in Fig. 12, matching along with their automatic kinematic semantics adaptation as shown in Fig. 6, to illustrate the implementation of our approach concretely.

According to the graph adaptation guidance in Fig. 6, the interfaces respectively corresponding to the deleted graph edges should be deleted from CM , such as the interface on G' contacting to I' will be deleted from CM . And the links and interfaces respectively corresponding to the incorporated graph elements should be incorporated to CM . Fig. 13d shows an example of interface transferring. Then, shape adaptation is carried out by

making each corresponding link in CM have the same interface layout as its counterpart in QM , such as link G' has the same interface layout as link E . Finally, the partial area of new CM contained in the red dashed curve of Fig. 13c has the same kinematic semantics and link layout as QM .

7.1.2. Experiment 2: an example lacking adequate corresponding location geometry elements

Here, one additional complex example, in Fig. 14, is adopted to demonstrate the effectiveness of our approach when there are lots of geometry differences between two shape-similar assembly models QM and CM .

As shown in Fig. 14g, a pair of typical corresponding links F and F' is selected to show what the adaptation result will be when it is difficult to find enough corresponding location geometry elements and/or corresponding interface faces. For example, the outline face 3, as a location geometry element in F as shown in Fig. 14h, has no corresponding location face in F' . Similarly, it is difficult to identify which interface face in I_3' corresponds to the red interface face 2 in I_3 . In such a case, our interface layout transferring will be automatically and partially achieved in the candidate model where the corresponding interface faces and/or corresponding geometry elements exist, such as the case shown in Fig. 14h and i: after adaptation, the layout among the hole faces respectively in I_1' , I_2' , I_3' and I_4' is the same as that among their corresponding hole faces respectively in I_1 , I_2 , I_3 and I_4 .

After ignoring all the matched edges, there exists one path (F' , K' , II' , H') between nodes F' and H' while there is no path between nodes F and H , so, edge $H'II'$ corresponds to a retained inconsistent kinematic semantics. Meanwhile, there exists one path (H' , II' , J') between nodes H' and J' and there also exists a path (H , J) between nodes H and J , so, edge $H'II'$ also corresponds to a deleted inconsistent kinematic semantics. Thus, $H'II'$ will be deleted from the final attributed kinematic graph (as shown in Fig. 14f) by default in the process of automatic kinematic semantics adaptation.

7.2. Comparisons

Compared with the state-of-the-art approaches on assembly model adaptation guided by semantics explicitly or implicitly for engineering applications as shown in Table 1, our approach has

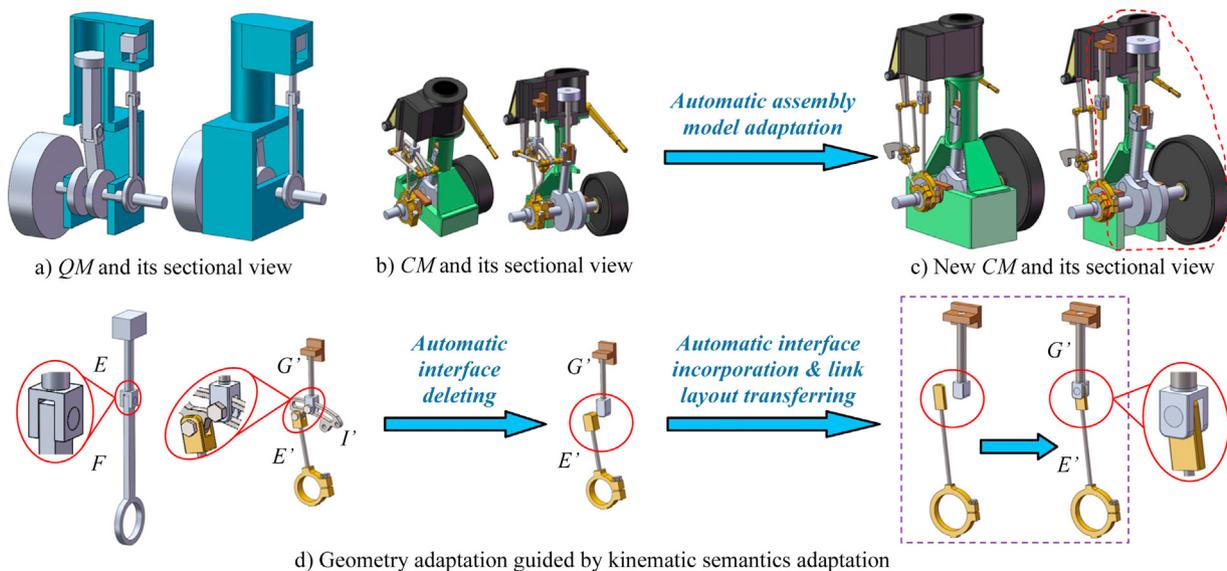


Fig. 13. An example of assembly model adaptation (links E and F respectively correspond to links G' and E').

Table 1
Comparison.

Reference	Input(s)	Semantics-driven	Specified	Supporting manner	Adapting means	Adaptation type
Liu et al. [14]	Semantics and model	Yes	Yes	Providing initial model	Modifying control parameters by human	Dimensional
Zhang et al. [16]	Semantics and models	Yes	Yes	Providing initial schemes and models and the flowing final model generation guidance	Adopting evolutionary method based on the initial models	Topological
This work	Models	Yes	No	Providing kinematic semantic adaptation and geometry adaptation guidance	Adapting the candidate model according to the query model	Topological and dimensional

several advantages according to the following items: (1) what is the input(s); (2) whether the geometry adaptation is semantics-driven; (3) whether the adopted semantics are domain-specified; (4) what about the supporting manner of semantic; (5) what about the adapting means for obtain the final model and (6) what about the main geometry adaptation type.

7.3. Sensitivity analysis

In order to find the corresponding areas having similar link layouts between the two given assembly models more efficiently based on sub-graph matching, which is in general a NP-hard problem, we not only require the two matched edges must have the same kinematic semantics and their assembly constraints have overlaps, but also use two thresholds $r1$ and $r2$ to make the

maximum isomorphic sub-graph searching process return fewer primarily matched sub-graphs. For all the test examples given in the paper, the values of $r1$ and $r2$ are set as 5° and 0.1 respectively and the effect is relatively good.

For the five cases shown in Fig. 15, the influences of the thresholds $r1$ and $r2$ to the efficiency and precision of maximum isomorphic sub-graph searching are shown in Fig. 15f and g respectively. It can be seen from Fig. 15f and g that, when both QM and CM in a case only contain translational degree of freedoms, then the matching time and the precision are only sensitive to the value of $r2$ as case 2 shows, and when both QM and CM in a case contain not only translational degree of freedoms but also rotational degree of freedoms, then the matching time and the precision are sensitive not only to the value of $r1$ but also to the value of $r2$ as cases 3–5 show. Additionally, when the values of $r1$

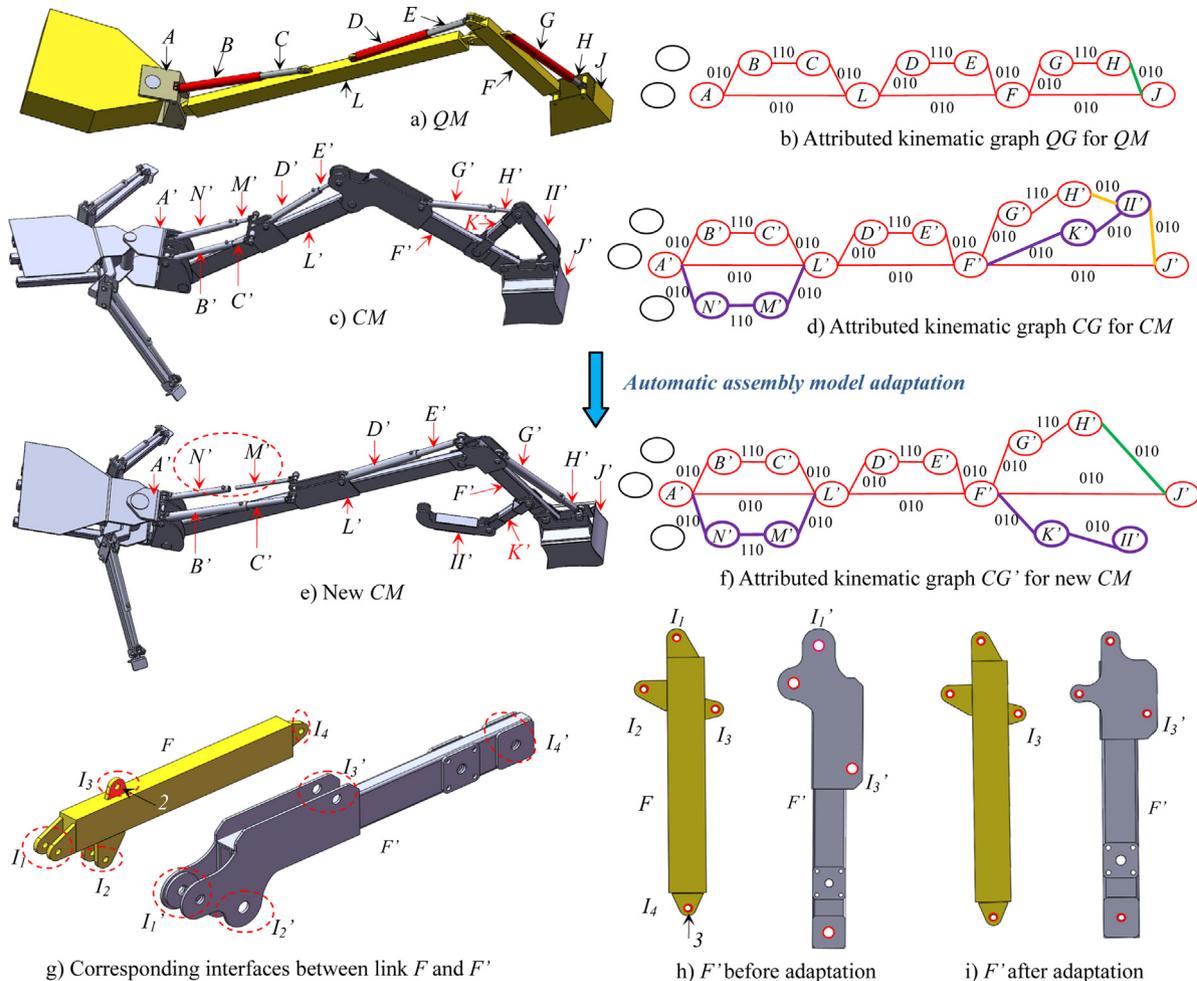


Fig. 14. Additional automatic adaptation example (QM is the query model while CM is the candidate model; $A&A'$, $B&B'$, $C&C'$, $L&L'$, $D&D'$, $E&E'$, $F&F'$, $G&G'$, $H&H'$, and $J&J'$ represent 10 pairs of corresponding links; each red graph element refers to a matched graph element while the other colored graph elements refer to non-matched graph elements; each pair of I_i and I_i' represents a pair of corresponding interfaces).

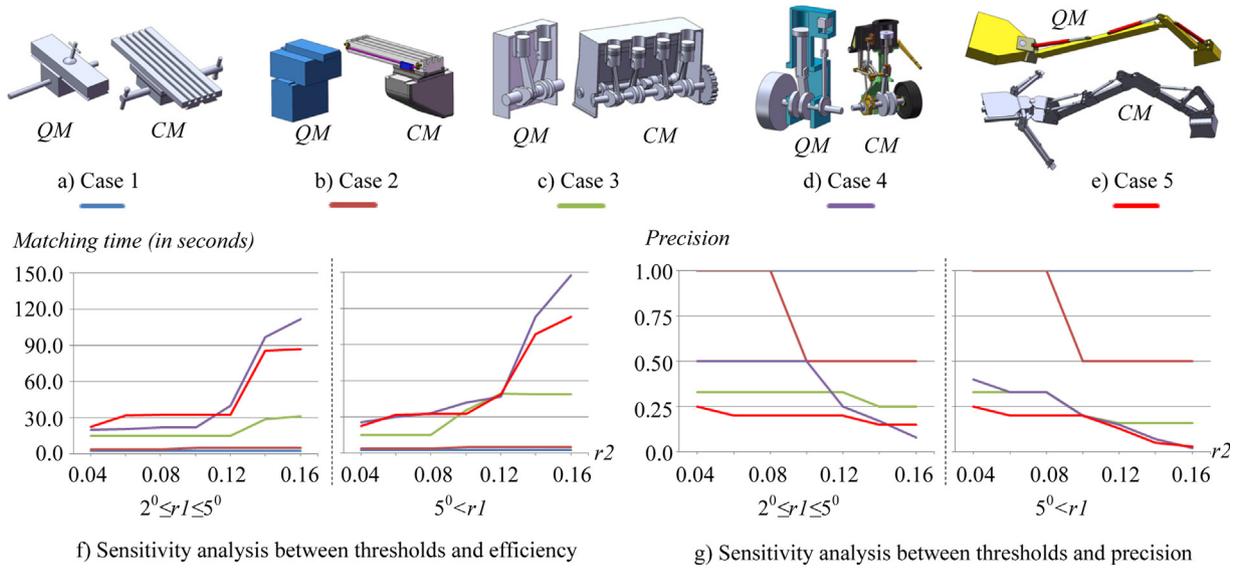


Fig. 15. Sensitivity analysis of the thresholds r_1 and r_2 used in maximum isomorphic sub-graph searching based on five cases (r_1 and r_2 refer to the thresholds of the distances between two matched edges' relative orientations and between their relative positions respectively).

Table 2
Multilevel information for each joint.

Semantics Layer	Degree of Freedom					
	Translation		Rotation		Composition	
	Kinematic Joint					
	Revolute	Prismatic	Cylindrical	Screw	Spherical	Planar
	Gear	Universal	Point on Surface	Point on Curve	Surface	
Geometry Layer	Assembly Constraint					
	Coincident	Concentric	Distance	Tangent	Perpendicular	Parallel
	Point on Line		Point on Surface		Edge on Surface	

and r_2 are set less than 5° and 0.1 respectively, our maximum isomorphic sub-graph searching algorithm returns fewer primarily matched sub-graphs in a relatively short time (usually less than 40s in our experiments) as shown in Fig. 15g.

8. Conclusion and future works

Assembly model adaptation is an important step of case-based assembly model design, but so far there is rare effective solution to automatic adaptation of assembly models. In this paper, according to the requirements of case-based assembly model design, we propose an approach to automatic adaptation of assembly models. In general, our approach has the following contributions and characteristics:

- 1) The approach enables the assembly model adaptation automatic, independent of domain knowledge library and low demand on shape similarity. This is achieved by dividing the adaptation into two steps: first carry out automatic kinematic semantics adaptation through the heuristic processing of the attributed kinematic graphs instead of knowledge library supporting; then perform automatic geometry adaptation mainly guided by the result of automatic kinematic semantics adaptation to make the geometry adaptation low demand on shape similarity.
- 2) Based on the attributed kinematic graph and heuristic graph matching, the approach can automatically and precisely identify the high-level geometry correspondence

(corresponding links and interfaces) between two assembly models without pre-registered.

It is important to note that this paper only covers partial contents of assembly model adaptation. Besides, the proposed approach itself has the following limitations:

- 1) The two input assembly models need to have similar link layouts. Under this assumption, the corresponding links and interfaces can be effectively determined.
- 2) Each interface can be incorporated relying on feature/part duplication to ensure the incorporation accurately.
- 3) Each location geometry element of one model has a counterpart in the other model. This assumption is used to ensure that constraint-driven shape adaptation and automatic interface duplication can be achieved.

Additionally, our approach also has the following limitation: using the layout to identify corresponding location geometry elements and corresponding interface faces is inefficient currently while this approach focuses on the geometry adaptation for mechanical engineering application and the geometry correspondence should be precise.

Considering that the automatic adaptation for assembly models is a well-recognized challenging problem, we choose to solve the problem step by step. And there are several works could be conducted to make our approach more general in the future according to the above limitations. For example, (a) to remove the

assumption that each location geometry element in one link should have a corresponding location geometry element in its counterpart link since it is too strong to satisfy for some cases, (b) to further study interface duplicating method to make it independent of features and so on.

Acknowledgements

The authors are very grateful to the financial supports from NSF of China (61173125) and National 863 High Technology Project (2013AA041301).

Appendix A.

See Table 2.

References

- [1] X. Chen, Assembly Search Based Top-down Product Design, College of Computer Science and Technology, Zhejiang University, Hangzhou, 2012148.
- [2] I. Watson, S. Perera, Case-based design: a review and analysis of building design applications, *Artif. Intell. Eng. Des. Anal. Manuf.* 11 (1997) 59–87.
- [3] A. Chakrabarti, K. Shea, R. Stone, J. Cagan, M. Campbell, N.V. Hernandez, K.L. Wood, Computer-based design synthesis research: an overview, *J. Comput. Inf. Sci. Eng.* 11 (2011) 021003-021001-021003-021010.
- [4] S.K. Chandrasegaran, K. Ramani, R.D. Sriram, I. Horváth, A. Bernard, R.F. Harik, W. Gao, The evolution, challenges, and future of knowledge representation in product design systems, *Comput.-Aided Des.* 45 (2013) 204–228.
- [5] D.P. Moreno, A.A. Hernández, M.C. Yang, K.N. Otto, K. Hölttä-Otto, J.S. Linsey, K. L. Wood, A. Linden, Fundamental studies in Design-by-Analogy: a focus on domain-knowledge experts and applications to transactional design problems, *Des. Stud.* 35 (2014) 232–272.
- [6] J.C.R. Bejarano, T. Coudert, E. Vareilles, L. Geneste, M. Aldanondo, J. Abeille, Case-based reasoning and system design: An integrated approach based on ontology and preference modeling, *Artif. Intell. Eng. Des. Anal. Manuf.* 28 (2014) 49–69.
- [7] A.S. Deshmukh, A.G. Banerjee, S.K. Gupta, R.D. Sriram, Content-based assembly search: a step towards assembly reuse, *Comput.-Aided Des.* 40 (2008) 244–261.
- [8] J. Bai, S. Gao, W. Tang, Y. Liu, S. Guo, Design reuse oriented partial retrieval of CAD models, *Comput.-Aided Des.* 42 (2010) 1069–1084.
- [9] X. Chen, S. Gao, S. Guo, J. Bai, A flexible assembly retrieval approach for model reuse, *Comput.-Aided Des.* 44 (2012) 554–574.
- [10] J.A. Recio-García, P.A. González-Calero, B. Díaz-Agudo, jCOLIBRI2: A framework for building Case-based reasoning systems, *Sci. Comput. Program.* 79 (2014) 126–145.
- [11] Y. Guo, Y. Peng, J. Hu, Research on high creative application of case-based reasoning system on engineering design, *Comput. Ind.* 64 (2013) 90–103.
- [12] Y. Avramenko, A. Kraslawski, Case Based Design: Applications In Process Engineering, Springer-Verlag, Berlin, 2008.
- [13] A. Heylighen, H. Neuckermans, A case base of Case-Based Design tools for architecture, *Comput.-Aided Des.* 33 (2001) 1111–1122.
- [14] Q. Liu, J. Xi, Case-based parametric design system for test turntable, *Expert Syst. Appl.* 38 (2011) 6508–6516.
- [15] Q. Cheng, C. Wu, Z. Liu, J. Wang, P. Gu, The parametric design and automatic assembly of hydrostatic rotary table based on Pro/Engineer, *Adv. Comput. Sci. Appl.* 279 (2014) 837–842.
- [16] X. Zhang, G. Peng, X. Hou, T. Zhuang, A knowledge reuse-based computer-aided fixture design framework, *Assembly Autom.* 34 (2014) 169–181.
- [17] H. Hua, A case-based design with 3D mesh models of architecture, *Comput.-Aided Des.* 57 (2014) 54–60.
- [18] C.-Y. Tsai, C.A. Chang, A two-stage fuzzy approach to feature-based design retrieval, *Comput. Ind.* 56 (2005) 493–505.
- [19] A.K. Goel, S. Craw, Design, innovation and case-based reasoning, *Knowl. Eng. Rev.* 20 (2006) 271–276.
- [20] C. Langenhan, M. Weber, M. Liwicki, F. Petzold, A. Dengel, Graph-based retrieval of building information models for supporting the early design stages, *Adv. Eng. Inf.* 27 (2013) 413–426.
- [21] B.M. Li, S.Q. Xie, Product similarity assessment for conceptual one-of-a-kind product design: a weight distribution approach, *Comput. Ind.* 64 (2013) 720–731.
- [22] Z.-P. Fan, Y.-H. Li, X. Wang, Y. Liu, Hybrid similarity measure for case retrieval in CBR and its application to emergency response towards gas explosion, *Expert Syst. Appl.* 41 (2014) 2526–2534.
- [23] C.-F. You, Y.-L. Tsai, 3D solid model retrieval for engineering reuse based on local feature correspondence, *Int. J. Adv. Manuf. Technol.* 46 (2010) 649–661.
- [24] J. Bai, Design Reuse Oriented 3d Cad Model Retrieval, College of Computer Science and Technology, Zhejiang University, Hangzhou, 2009171.
- [25] S. Tao, Z. Huang, L. Ma, S. Guo, S. Wang, Y. Xie, Partial retrieval of CAD models based on local surface region decomposition, *Comput.-Aided Des.* 45 (2013) 1239–1252.
- [26] L. Ma, Z. Huang, Y. Wang, Automatic discovery of common design structures in CAD models, *Comput. Graphics* 34 (2010) 545–555.
- [27] J. Zhang, Z. Xu, Y. Li, S. Jiang, N. Wei, Generic face adjacency graph for automatic common design structure discovery in assembly models, *Comput.-Aided Des.* 45 (2013) 1138–1151.
- [28] M. El-Mehalawi, R.A. Millerb, A database system of mechanical components based on geometric and topological similarity. Part I: representation, *Comput.-Aided Des.* 35 (2003) 83–94.
- [29] Z. Li, X. Zhou, W. Liu, A geometric reasoning approach to hierarchical representation for B-rep model retrieval, *Comput.-Aided Des.* (2014) .
- [30] R. Huang, S. Zhang, X. Bai, C. Xu, Multi-level structuralized model-based definition model based on machining features for manufacturing reuse of mechanical parts, *Int. J. Adv. Manuf. Technol.* (2014) .
- [31] W. Gao, S.M. Gao, Y.S. Liu, J. Bai, B.K. Hu, Multiresolutional similarity assessment and retrieval of solid models based on DBMS, *Comput.-Aided Des.* 38 (2006) 985–1001.
- [32] K.-M. Hu, B. Wang, J.-H. Yong, J.-C. Paul, Relaxed lightweight assembly retrieval using vector space model, *Comput.-Aided Des.* 45 (2013) 739–750.
- [33] Y. Wu, Q. Gao, Spatial structure similarity retrieval of product assembly model, *Comput.-Aided Des. Comput. Graphics* 26 (2014) 113–120.
- [34] K. Lee, D.C. Gossard, A hierarchical data structure for representing assemblies: part 1, *Comput.-Aided Des.* 17 (1985) 15–19.
- [35] L.-W. Tsai, Mechanism Design: Enumeration of Kinematic Structures According to Direction, CRC Press, Florida, 2000.
- [36] S.-J. Chiou, K. Sridhar, Automated conceptual design of mechanisms, *Mech. Mach. Theory* 34 (1999) 467–495.
- [37] G. Krishnan, C. Kim, S. Kota, Building block method: a bottom-up modular synthesis methodology for distributed compliant mechanisms, *Mech. Sci.* 3 (2012) 15–23.
- [38] Y.-H. Han, K. Lee, A case-based framework for reuse of previous design concepts in conceptual synthesis of mechanisms, *Comput. Ind.* 57 (2006) 305–318.
- [39] H.-S. Yan, A methodology for creative mechanism design, *Mech. Mach. Theory* 27 (1992) 235–242.
- [40] B. He, P. Zhang, L. Liu, Simultaneous functional synthesis of mechanisms with mechanical efficiency and cost, *Int. J. Adv. Manuf. Technol.* (2014) .
- [41] K. Hua, B. Fairings, I. Smith, CADRE: case-based geometric design, *Artif. Intell. Eng.* 10 (1996) 171–183.
- [42] G. Lashin, J. Feldhusen, A CAD-based tool for development of large layouts, *Res. Eng. Des.* 8 (1996) 217–228.
- [43] A. Csabai, I. Stroud, P.C. Xirouchakis, Container spaces and functional features for top-down 3D layout design, *Comput.-Aided Des.* 34 (2002) 1011–1035.
- [44] M. Mantyla, A modeling system for top-down design of assembled products, *IBM J. Res. Dev.* 34 (1990) 636–659.
- [45] X. Chen, S. Gao, Y. Yang, S. Zhang, Multi-level assembly model for top-down design of mechanical products, *Comput.-Aided Des.* 44 (2012) 1033–1048.
- [46] UG/NX9, Siemens Product Lifecycle Management Software Inc., 2014.
- [47] Soliworks, The Dassault Systèmes SolidWorks Corporation, 2014.
- [48] W. Pan, X. Chen, S. Gao, Automatic shape adaptation for parametric solid models, *Comput.-Aided Des.* 62 (2015) 78–97.
- [49] R. Osada, T. Funkhouser, B. Chazelle, D. Dobkin, Shape distributions, *ACM Trans. Graphics* 21 (2002) 807–832.
- [50] J.U. Turner, S. Subramaniam, S. Gupta, Constraint representation and reduction in assembly modeling and analysis, *IEEE Trans. Rob. Autom.* 8 (1992) 741–750.
- [51] G. Xiaoning, Technical Report: A kinematics analysis and simulation system for the complex virtual prototyping under CAVE, 2005.
- [52] Boost, Boost C++ Libraries, 2014.
- [53] S. Venkataraman, M. Sohoni, Reconstruction of feature volumes and feature suppression, Proceedings of the seventh ACM symposium on Solid modeling and applications, Saarbrücken, Germany, 2002, pp. 60–71.
- [54] X. Chen, C.M. Hoffmann, Towards feature attachment, *Comput.-Aided Des.* 27 (1995) 695–702.
- [55] J. O'Rourke, Finding minimal enclosing boxes, *Int. J. Comput. Inf. Sci.* 14 (1985) 183–199.
- [56] J. Wu, T. Zhang, X. Zhang, J. Zhou, A face based mechanism for naming, recording and retrieving topological entities, *Comput.-Aided Des.* 33 (2001) 687–698.
- [57] X. Li, F. He, X. Cai, D. Zhang, Y. Chen, A method for topological entity matching in the integration of heterogeneous CAD systems, *Integr. Comput.-Aided Eng.* 20 (2013) 15–30.
- [58] J. Li, B.C. Kim, S. Han, Parametric exchange of round shapes between a mechanical CAD system and a ship CAD system, *Comput.-Aided Des.* 44 (2012) 154–161.
- [59] J. Jiang, Z. Chen, K. He, A feature-based method of rapidly detecting global exact symmetries in CAD models, *Comput.-Aided Des.* 45 (2013) 1081–1094.
- [60] S.-x. Jing, F.-z. He, S.-h. Han, X.-t. Cai, H.-j. Liu, A method for topological entity correspondence in a replicated collaborative CAD system, *Comput. Ind.* 60 (2009) 467–475.
- [61] A.V. Kumar, L. Yu, Sequential constraint imposition for dimension-driven solid models, *Comput.-Aided Des.* 33 (2001) 475–486.