Computer-Aided Design 62 (2015) 78–97

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

Automatic shape adaptation for parametric solid models ☆

Wanbin Pan [a,b], Xiang Chen [a], Shuming Gao [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

HIGHLIGHTS

- Automatic shape adaptation is achieved based on corresponding faces and dimensions.
- A new method of determining corresponding faces is put forward.
- An algorithm to identify corresponding dimensions by constraint graph is proposed.

ARTICLE INFO

Article history:
Received 3 November 2012
Accepted 4 November 2014

Keywords:
Automatic shape adaptation
Shape frame
Corresponding face
Dimension promotion

ABSTRACT

Adaptation, as is well known, plays a fundamental role in Case-Based Design. However, after decades of efforts, automatic adaptation approach is still rare. In common design works, the first thing one will usually do is choosing a start-up model (a candidate model) of moderate complexity based on a simple query model possessing primary design constraints. To enable the candidate model to smartly adapt its shape to that of the query model according to the embedded constraints, a novel automatic shape adaptation approach is proposed in this paper. First, to determine the corresponding faces between two non-preregistered models as relevant elements, a shape frame concept and its quantitative descriptor are defined. Second, to unify the representation of seemingly different but inherently consistent dimensions, a promotion method is adopted. Third, based on the corresponding faces and the promoted dimension representation, the corresponding dimensions between the two parametric solid models are identified. Finally, the parametric information is smoothly transferred from the query model to the candidate model as design constraints, and the shape of the candidate model is automatically adapted to the query model. Besides that, a prototype system is also implemented to verify the effectiveness of the automatic shape adaptation approach.

1. Introduction

It is well known that designers use their experience of design along with combinations and/or adaptations of previous cases or parts of cases in creating a new design [1]. Watson et al. [2] term this as Case-Based Design (CBD). Along with the computer-aided design (CAD) systems joining modern industries, a vast number of design cases are generated and stored in the internet or enterprise repositories. These cases always contain plenty of embedded knowledge worthy of utilization. Subsequently, CBD has got more and more attention in the past few decades [2], while the advancement of CBD approaches makes it easier and faster to produce impressive and innovative designs in a widespread field [3,4].

CBD mainly consists of two processes: retrieving a suitable case and adapting it to satisfy new design requirements [2,4–6]. Modern retrieval technologies [7–23], capable of automatically searching out the cases similar to a user input query from a huge case library, are extensively used and well studied. On the other hand, adaptation is essential for CBD since no two design problems are ever identical. According to the researches mentioned in [11,24], only a small part of products requires completely new designs, while more parts of them are obtained through an adaptation of the existing cases. However, manual adaptation process is usually tedious in some design tasks, especially during routine design. To free the designers from these unnecessary burdens and improve the efficiency of CBD, it is very important and necessary making the case adaptation automatic.

Unfortunately, although the importance of automatic case adaptation in CBD is obvious, the automatic adaptation approach is

☆ This paper has been recommended for acceptance by Vadim.

* Corresponding author. Tel.: +86 0571 88206681 514.
E-mail addresses: panwanbin@hdu.edu.cn (W. Pan), xchen@cad.zju.edu.cn (X. Chen), smgao@cad.zju.edu.cn (S. Gao).

http://dx.doi.org/10.1016/j.cad.2014.11.001
0010-4485/

rare [2,3,5,25–30]. While some good works referred in [3,4,31,32] show that case adaptation falls broadly into two categories: adaptation in conceptual design, and adaptation in shape (geometric) design. Despite of focusing on different design categories, two major common challenges have to be overcome in order to achieve automatic adaptation:

(a) *How to automatically identify which design elements in the retrieved case are relevant to new design requirements*.
(b) *How to automatically transfer the new design requirements to the retrieved case via their relevant design elements*.

It should be pointed out that, up to now, these two challenges are still the obstacles of achieving automatic case adaptation [3]. On one hand, the automatic case adaptation for concepts, ranging from engineering to architecture to software design, shows much more feasible and prospective when the prerequisite of knowledge around domain area can be captured sufficiently and represented properly [3,4]. However, satisfying the above prerequisite is domain dependent and is not a trivial work [33,34]. On the other hand, the automatic case adaptation for geomantic shapes seems much more difficult without the aid of semantic information support (and/or domain specification), such as (a) the works of shape deformation [35–38] (establishing a meaningful correspondence between two shapes is often difficult and semantically dependent [39]), and (b) the works of shape optimization [6,40–44] (domain specifications are required, such as materials, and pressures). So, when the semantic information support or domain specification is absent, shape adaptation is often carried out by humans instead of using the existing works [3].

Although a general automatic shape adaptation (geometric adaptation) is difficult and the related works are rare, it plays an important role in modern product design, such as the designs related to product shape optimization and structure optimization [42,44–47]. Modern product design usually contains distinct design phases, such as preliminary design, conceptual design, geometric modeling, and so on. Furthermore, product design is an iterative design process among the distinct design phases, and the specifications of the required function of a product get more refined only as the design process moves toward its goal [4,34]. Accordingly, the geometric shape of a product usually needs modification at each iterative step since the function of a product is often represented by its geometric shape (especially in mechanical engineering). Thus, in order to reduce the burden of manual geometric modification at each iterative step, automatic shape adaptation is necessary and very important.

In this paper, we propose an automatic shape adaptation approach to enable one parametric solid model to smartly adapt its shape to that of the other one according to the embedded constraints. Because the embedded constraints in each parametric solid model are usually adopted by designers to represent their new design requirements (intent), transferring shape according to the embedded constraints is actually transferring design intent [48,49]. As we know, with the aid of embedded constrains, each parametric solid model can update its shape elements (faces, edges, and vertices) in a predicted manner after any dimension constraint modification. Apparently, an effective and direct method to transfer shape between two parametric solid models is to establish relationships between their embedded dimension constraints. However, without semantic information support, it is a difficult issue to establish relationships between two models' embedded dimension constraints automatically since the parametric way for a parametric solid model can be very flexible (different parametric ways lead to different constraint configurations).

Our approach is presented with a view to overcome the above mentioned issue in feature-based parametric design. The inputs of the approach are two parametric solid models without having been pre-registered: a query model and a candidate model. The query model, indicating the gross new shape design requirements through its roughly B-rep shape and its dimension constraints, is a simple parametric solid model. The candidate model is a parametric solid model searched from a parametric solid model library according to the query model, whose overall shape is similar to that of the query model but having more details. According to the two challenges for automatic shape adaptation (i.e. how to automatically identify which design elements are corresponding (relevant) between the two given models' shapes, and how to automatically transfer the shape of the query model to the candidate model via their corresponding (relevant) design elements), our solution consists of the following parts: (1) we select face as the shape relevant design element, and in spite of the absence of semantic support, a novel method is presented to determine the corresponding shape frames and corresponding faces between the two models automatically; (2) we adopt a promotion method that makes face as the intermediate element that relates each embedded dimension to its model's shape; (3) based on 3D dimension constraint graphs and corresponding faces, we identify all corresponding dimensions and corresponding dimension chains between the corresponding shape frames effectively; (4) we establish the dimension relationships among the corresponding dimensions and corresponding dimension chains to make the candidate model adapt its shape to that of the query model.

The rest of the paper is organized as follows. In Section 2, we give a brief review of the related works. In Section 3, some concepts and an overview of our approach is provided. Section 4 gives a detailed process of determining corresponding faces and corresponding shape frames. In Section 5, a dimension promotion method is adopted to identify corresponding dimensions and corresponding dimension chains between the two given models based on the result of Section 4. Afterward, the dimension relationship between the two given models is established. Section 6 introduces the implementation details of the prototype system and shows some experiments and their efficiencies. Finally, we discuss our approach in Section 7, and present the conclusion of the paper and our further works in Section 8.

2. Related works

Nowadays, there are few works dedicated to the automatic shape adaptation. Therefore, the works that we have surveyed are mainly relevant to the general case adaptation. According to different design levels, the existing works related to the first challenge (i.e. how to automatically establish correspondences between a source case and a target problem or between two cases) can be divided into two types: conceptual correspondence research and geometric correspondence research.

Conceptual correspondence research: Establishing a conceptual correspondence, between a library case and an input target problem (or new requirements), is often performed in the process of case retrieval based on a set of specific metrics. Meaningful labels are early used to index and retrieve cases or case features. Besides, the similarity between two labels measures the degree of the correspondence between the two cases (or two case features) that correspond to the two labels respectively. Kolodner [50,51] presents an algorithm for knowledge-based memory reorganization, which can decide which case (or case features) is corresponding to the new inputs with the aid of previous knowledge. In analogy-based design area, Falkenhainer et al. [52] build a structure mapping (corresponding) engine to explore the computational aspects of the structure-mapping theory proposed in Gentner [53]. The mechanism of structure-mapping mentioned above is essentially independent of task, domain, or the context knowledge. In computer-aided functional synthesis area, after classifying each case under a specific component name according to its function, potential design solutions based upon previously known cases can be presented [13,54,55], since each new requirement corresponds to a specific function component. However, the identified relevant conceptual design elements are often yet to provide direct correspondence between the geometric design elements of the two cases, and which is the focus of this work.

Geometric correspondence research: The design requirements are usually represented as shapes in CBDs focusing on geomantic design. Then, a geometric correspondence identification in geometric design is to identify a shape correspondence between two models. Lots of model retrieval approaches have been proposed in the past decades, such as (a) the approaches that get the candidate models by inputting a query model [11,17,21–23,56,57], and (b) the works that discover the candidate model by deforming a template model (such as the work presented by Ovsjanikov et al. [58]). Usually, model retrieval approaches can determine whether two models' shapes are similar or not on globally or partially, but they are difficult to obtain more matching details especially when the two models have different resolutions [9,59]. Shape correspondence, being a fundamental problem for 3D surface shape retrieving in the computer graphics communities [39], is mainly related to discrete shapes represented by triangle meshes, contours or point sets. Geometric correspondence establishment can be carried out by identifying portions of the shapes that are geometrically similar. Sumner et al. [35] build the correspondence between the source and target triangles models using an iterated closest point algorithm with regularization, aided by user selecting marker points. Hu et al. [60] describe a shape with a set of salient features extracted from surfaces that are represented by triangle meshes, and achieve both global and partial shape correspondence with these salient feature points. Establishing geometric correspondence can also be processed by identifying portions of the shapes that serve the same function. Kaick et al. [61] use prior knowledge to establish a correspondence between shapes based on similar functions while ignoring significant differences in geometry or even topology. According to whether the approach explores the whole solution space in search for a better correspondence result, the recent geometric correspondence approaches can be divided into global search approaches and local search approaches. For example, Zhang et al. [36] present a global feature correspondence search approach by leveraging the power of the state-of-the-art mesh deformation techniques and relying on a combinatorial tree traversal for correspondence search. While the most prominent example of the local search type is the ICP algorithm [62], which replaces the computation of correspondences among the feature points of two models with the computation of alignment transformations between them. Ovsjanikov et al. [63] present an approach that works well to establish shape correspondences amongst near-isometrically deformed shapes. Generally speaking, establishing a meaningful correspondence between two discrete shapes is often difficult since it often requires an understanding of the structures of the two shapes respectively (at both the local and global levels) [10,39,60,64–66]. Shape matching is a character of reverse engineering that the geometric elements (such as surface) in the physical model should have corresponding geometric elements in its re-designed digital model. For example, Goyal et al. [67] present a re-design 3D modeling approach toward locally and globally shape-aware. Using the re-design approaches to establish shape correspondences between two solid models require the two models having similar resolutions while having been normalized and registered (aligned). Our approach aims to determine face correspondences between two non-preregistered B-rep models with different resolutions automatically, so the recent state-of-the-art approaches for shape correspondence are not suitable here.

Regarding the second challenge (i.e. how to automatically transfer the new design requirements to the retrieved case via their relevant design elements), the related works are as follows.

Watson et al. [2] describe a survey of adaptation in CBD where they summarize that adaption 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. In addition, Kolodner [68] classifies the knowledge-based adaptation approaches into four categories: substitution approaches, transformation approaches, special purpose approaches and derivational replay approaches. Especially, most of the current case-based design works focus on concept design, where the knowledge capture and its representation [34] are the preconditions for an effective adaptation in product design [33]. Goel et al. [4] present a review of many researches on development of case-based reasoning in design and their corresponding case-based design systems. Pearce et al. [69] present a CBD approach, which is an intelligent case browsing system and aims at providing architects with a design library for the conceptual architectural design of office buildings. After selecting a candidate case with the above approach, users have to make the adaptation themselves. Maher et al. [41] present a CBD approach focusing on case transformation and regarding case adaptation as a constraint satisfaction problem, where a candidate case provides a starting point for a new design problem and constraints that are used to revise the case for consistency with the new context. Hua et al. [6] 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. Hunt [29] presents an evolutionary adaptation approach for structural engineering design which extends the basic framework of CBR to include an evolutionary approach to adaptation. This is the adaptation approach that integrates case combination and knowledge-based adaptation to solve design problems. Perera et al. [30] propose an interactive prototype CBD system NIRMANI which aims at automating design either fully or partially. Adaptation in NIRMANI is carried out in two phases: (1) using an index elaboration and index revision mechanism to refine the design problem specification, and (2) converting the problem specification into a solution through a case modification and combination process based on a set of heuristic rules and domain knowledge. Bose et al. [40] present a case-based design system for the solution of mechanism design problems which can be solved by using four-bar linkages. Some modern 3D cad systems have provided a similar adaptation mechanism in geometric design stage [70], where a part is set to be under-constrained, based on domain knowledge, and then the automatic shape adaptation of the part can be carried out in an assembly circumstance based on its mating relationships determined by human intervention. Vong et al. [71] employ case-based reasoning (CBR) to adapt an existing and effective ECU setup to fit another similar class of engine. The adaptation procedure is done through a more sophisticated step called case-based adaptation which can interactively learn the expert adaptation knowledge. In a word, whatever adaptation approach is used, extensive domain specific knowledge is required as mentioned by Raphael et al. [25], such that case adaptation is often carried out by humans [3].

In engineering field, the shape optimization [32,72] is similar as the shape adaptation in some aspects. In the shape optimization, values of the shape variables have to be determined which result in an optimal value of a target parameter. Perez et al. [42] present the implementation of a particle swarm optimization algorithm suitable for constraint structural optimization tasks. Recently, Yildiz [44] proposes a hybrid optimization approach based on differential evolution algorithm for solving structural design, accompanied with two design problems: a welded beam design problem and the optimal design of a vehicle component. Meanwhile, there are also many approaches for topological structure optimization. Allaire et al. [45] and Xia et al. [46] present topology optimization approaches based on the level-set approach.

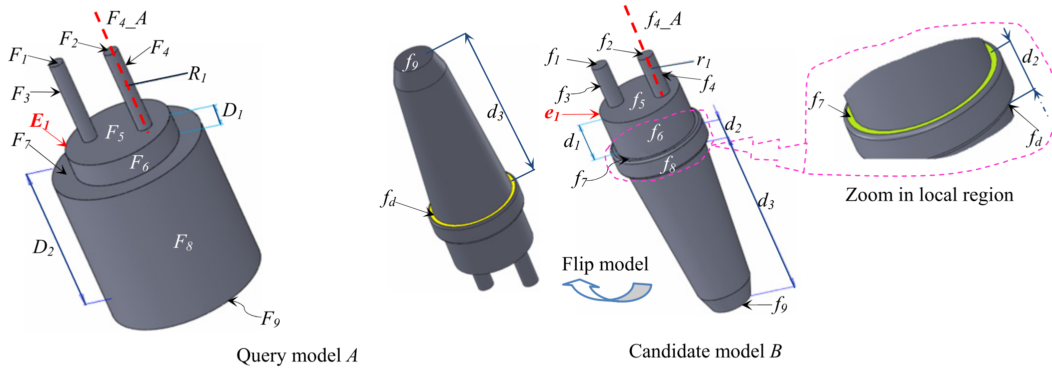


Fig. 1. Illustration of correspondences related to the concepts (shape frame $\{F_i|1 \le i \le 9\}$ and shape frame $\{f_i|1 \le i \le 9\}$ comprise a pair of corresponding shape frames; each pair of face F_i and f_i corresponds to a pair of corresponding faces ($1 \le i \le 9$); dimension D_1 constrains the distance between face F_7 and F_5; dimension d_1 constrains the distance between face f_7 (the green plane face) and f_5; D_1 and d_1 comprise a pair of corresponding dimensions; dimension D_2 constrains the distance between face F_7 and F_9; dimension d_2 constrains the distance between face f_7 and f_d (the yellow plane face); dimension d_3 constrains the distance between face f_d and f_9; D_2 corresponds to the dimension chain (d_2, d_3); F_4_A and f_4_A are the axes of face F_4 and f_4 respectively). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Richardson et al. [43] address single and multi-objective topology optimization of truss-like structures using genetic algorithms and present an approach based on kinematic stability repair to improve the performance of the genetic algorithms. Wang et al. [47] present a material perturbation approach (MPM) using a fixed mesh for stress sensitivity analysis and shape optimization. Although the adapted shape can be obtained by optimizing the parameters of an existing and successful design, the parameters (including the goal function) used in an optimization approach are domain-related (such as materials, and pressures), furthermore, shape optimization approaches are often designed for assemblies not for a single part.

In computer graphics field, recently, the main idea of the state-of-the-art approaches on shape deformation is that the shape of a given model will be deformed according to some correspondences (built online or predefined) between a source model (or reference model) and a target model (or reference model). Sumner et al. [35] present a deformation approach that transfers the induced deformation from a source model to a target model through a correspondence map. Brouet et al. [38] present a fully automatic approach that transfers a garment from a source wearer to a target wearer while retaining the garment's specific patterns as much as possible. Meng et al. [73] present an approach to control the shape of a cloth during the automatic resizing. Lee et al. [74] propose an automatic virtual garment transfer system for human models with various shapes and poses, focusing on the realistic fitting result while preserving the original garment size. Besides, there are also some state-of-the-art approaches (such as the structure-aware shape processing approaches [75]) that try to do shape deformation for each model according to its own inner constraints. Kraevoy et al. [76] present a non-homogeneous resizing approach that resizes each model respecting to the model's vulnerability map, and creates new models that preserve the structures and features of the original ones. Gal et al. [77] introduce iWIRES to do structure-preserving shape deformation for the man-made models. Zheng et al. [78] introduce component-wise controllers to do structure-aware shape deformation. Bokeloh et al. [79] present a high-level shape editing approach by extracting all possibly linear patterns and their relationships from a model (each pattern is represented by a small number of variables). Although the above structure-aware shape processing approaches analyze and process shapes at a high level, it is still a challenging task presently to extract the high-level information (such as semantic local shapes and their relations) from discrete models [75,80].

3. BASIC concepts and approach overview

3.1. Basic concepts

Face layout: The **face layout** is defined as the relative positions and relative orientations among the faces belonging to the same face set. The face layout describes the rotation invariant characters of a face set.

Shape frame: Given a face set *FS* of a solid model, if the overall shape of the model can be characterized by the surface shapes of the faces in *FS* and/or the face layout of *FS*, then *FS* is defined as a **shape frame** of the solid model. For example, the face set $\{F_i|1 \le i \le 9\}$ in Fig. 1 forms a shape frame of model *A*.

Usually, in view that the query model in this work is very simple, the shape frame of the query model is considered consisting of all the faces of the query model and is denoted by $\mathbf{SF}_Q$. Additionally, in terms of a query model and its SF_Q, the shape frame of the candidate model is obtained; such shape frame is called the **corresponding shape frame** of SF_Q. Considering that the overall shape of the candidate model is similar to that of the query model, in this work we assume that SF_Q and its corresponding shape frame have the same count of faces. Fig. 1 shows a pair of corresponding shape frames respectively belonging to model *A* and model *B*.

Directed surface judgment: The **directed surface judgment** as a Boolean property is attached to a face in this work. Such property indicates whether the face contains a directed surface (whose normal always points to the outside of body) or not.

Corresponding face: Let FS_Q in the query model and FS_Q in the candidate model be a pair of corresponding shape frames. If a pair of the faces respectively belonging to FS_Q and FS_Q has the same surface type and directed surface judgment, as well as similar relative orientations and relative positions (relative to other faces in their own face sets), then such pair of the faces can be defined as a pair of **corresponding faces**. For example, face F_1 and f_1 in Fig. 1 comprise a pair of corresponding faces.

Corresponding edge: Let E_i in the query model and e_i in the candidate model be a pair of edges; E_i is shared by face F_i and F_j while e_i is shared by face f_i and f_j. If F_i corresponds to f_i while F_j corresponds to f_j, or F_i corresponds to f_j while F_j corresponds to f_i, then such pair of edges is defined as a pair of **corresponding edges**. For example, in Fig. 1, face F_5 and F_6 respectively correspond to face f_5 and f_6; edge E_1 is shared by F_5 and F_6; edge e_1 is shared by f_5 and f_6. Thus, edge E_1 and e_1 comprise a pair of **corresponding edges**.

Corresponding axis: Let F_Q in the query model and F_C in the candidate model be a pair of corresponding faces. If F_Q and F_C are quadratic faces containing only one axis respectively, then their axes are called **corresponding axes**. For example, in Fig. 1, F_4_A and f_4_A are the axes of face F_4 and f_4 respectively. Because face F_4 and f_4 are corresponding faces, F_4_A and f_4_A comprise a pair of **corresponding axes**.

To establish dimension relationships between two parametric solid models, we adopt a dimension promotion method (see Section 5) to promote each dimension to be defined on/between entity(s) constrained by that dimension. Here each entity can be an edge, an axis of a quadratic face or a face.

Constrained entity: The **constrained entity** is an entity (not a sketch element) which is constrained by a promoted dimension. According to our dimension promotion mechanism (see Section 5), a constrained entity can be a face, an edge or an axis of a quadratic face in this work. Accordingly, each pair of corresponding faces, corresponding edges, or corresponding axes is also a pair of **corresponding constrained entities**.

Corresponding dimension: Let D and d be two dimensions with the same dimension type (distance or angle) belonging to the query model and the candidate model respectively, they are called **corresponding dimension**s of each other if the constrained entities constrained by them are corresponding constrained entities. For example, in Fig. 1, D_1 constrains the relative position between face F_5 and face F_7, and d_1 constrains the relative position between face f_5 and face f_7. Since F_5 and F_7 correspond to f_5 and f_7 respectively, D_1 is the corresponding dimension of d_1. Similarly, R_1 and r_1 are the two radiuses of the two faces' surfaces respectively, and face F_4 corresponds to face f_4. Thus, R_1 and r_1 comprise a pair of the corresponding dimensions.

Corresponding dimension chain: Let D be a dimension in the query model, if there is a dimension chain in the candidate model whose first constrained entity and last constrained entity (i.e. the two constrained entities are constrained by the dimension chain) are the corresponding constrained entities of those constrained by D, the dimension chain is defined as the **corresponding dimension chain** of D. For example, in Fig. 1, D_2 constrains the relative position between face F_7 and face F_9 in model A. Meanwhile, F_7 and F_9 correspond to f_7 and f_9 respectively. Since there is no corresponding dimension for D but having the only one dimension chain (d_2, d_3) that constrains the relative position between face f_7 and face f_9. Thus, D_2 corresponds to the dimension chain ($d_{2,} d_3$).

3.2. Overview of approach

In order to effectively improve the reuse of parametric solid models in Case-Based Design, we propose an automatic approach for shape adaptation of parametric solid models. The inputs of the approach are two parametric solid models: a query model and a candidate model. The query model, used to indicate the new shape design requirements through its boundary representation and its dimension constraints, is a simple parametric solid model created by the designer. The candidate model is a parametric solid model searched from a parametric solid model library according to the query model, whose overall shape is similar to that of the query model but having more details. The goal of our approach is to automatically adapt the shape of the candidate model to that of the query model, making the candidate model more suitable for reuse. To properly reduce the complexity, at present, we hypothesize that the two input models are composed of plane faces and/or quadratic faces.

Regarding the first challenge for automatic shape adaptation, i.e. how to effectively determine the corresponding faces between the query model and the candidate model, we solve it using a global method. We determine the corresponding shape frames between the two models first, and then determine the corresponding faces based on the corresponding shape frames. By mainly adopting face layout as the descriptor of a set of faces, our method does not require that the query model and the candidate model are pre-registered (or pre-aligned).

For example, in Fig. 2, after determining corresponding shape frames, the faces with the same color (except gray), belonging to the two given models A and B respectively, are determined as the corresponding faces. Although the relative sizes of face 11 and 17′ are vastly different, they are corresponding faces since both of them intrinsically constrain the shapes of their own model in the same direction.

As for the second challenge for automatic shape adaptation, i.e. how to automatically transfer the shape of the query model to the candidate model, considering that the shape of the query model is determined by its SF_Q and dimension constraints, we first identify all the corresponding dimensions and corresponding dimension chains between SF_Q and its corresponding shape frame based on their corresponding faces (other corresponding constrained entities can be deduced by corresponding faces according to their definitions) and the 3D dimension constraint graphs of the query model and the candidate model, then make the corresponding dimensions and corresponding dimension chains have the same values as those of their counterparts to achieve the automatic shape transfer from the query model to the candidate model.

For example, in Fig. 2, after establishing the dimension relationship between the radiuses of face 11 and 17′($r = R$), the gross radiuses of the candidate model and the query model are the same.

Fig. 2 shows the systematic overview of the approach which consists of the following three parts:

(1) Determination of the corresponding shape frames and corresponding faces between the query model and the candidate model;
(2) Establishment of dimension relationship between SF_Q and its corresponding shape frame;
(3) Update of the candidate model. In this step, the candidate model is updated by making the corresponding dimensions and dimension chains have the same values as those of their counterparts in the query model.

In view that the third part of our approach is simple, we just describe the part 1 and part 2 in detail below.

4. Determination of the corresponding shape frames and corresponding faces

According to our knowledge, there is no existing method that can determine face correspondences between two non-preregistered B-rep models automatically and accurately, as a result, a similarity comparison method is used to achieve above targets in this work. Since each shape frame is a face set, for the sake of generality, we first adopt a face set descriptor to describe the rotation invariant characters of each face set. Next, we propose a face sets distance measurement to evaluate the similarities among face sets by comparing their descriptors. Afterward, based on the face sets distance measurement and according to SF_Q, we apply a heuristic filtering process to determine a face set (being a shape frame of the candidate model and having the greatest similarity with SF_Q) as the corresponding shape frame of SF_Q. Subsequently, the preliminary corresponding faces between the two models are determined. Finally, corresponding faces between the two models are completed based on the feature information of the two models, respectively.

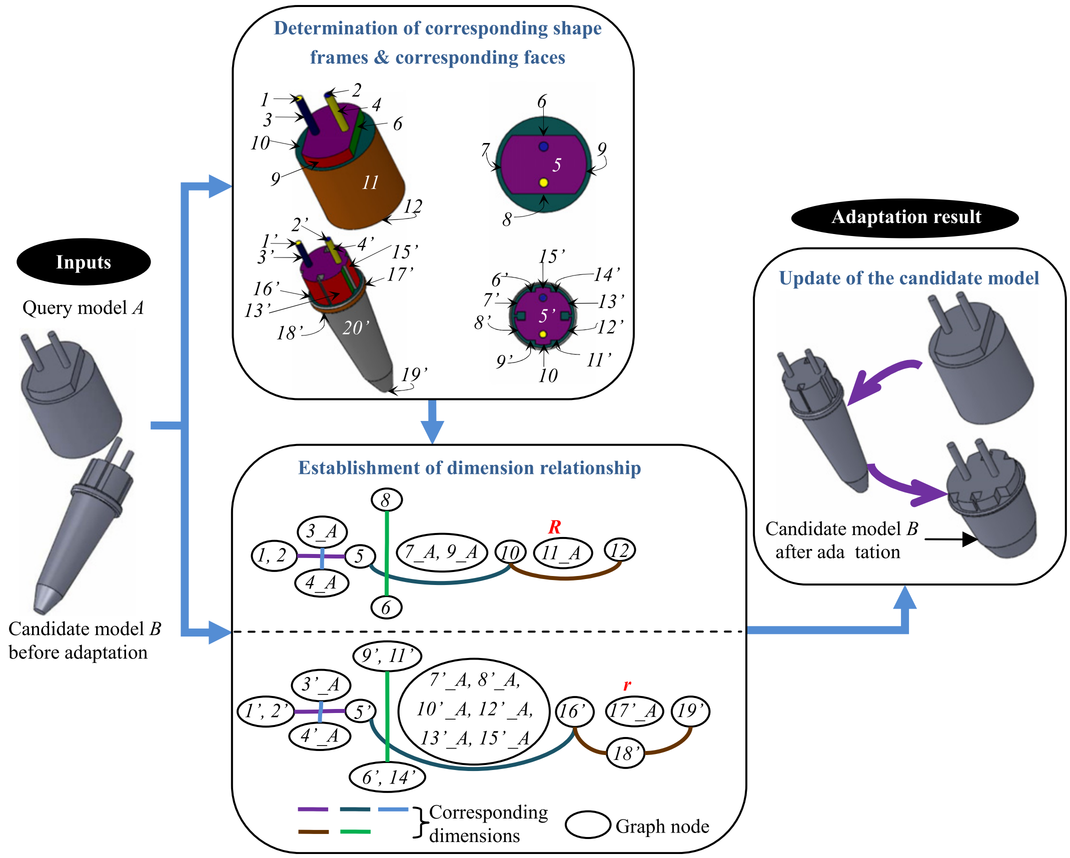


Fig. 2. Overview of automatic shape adaptation approach: model B automatically adapts its shape to model A (N_A represents the axis of the quadratic surface containing face N; each graph node, attached with a property(s), represents a constrained entity(s); R & r are the radiuses of face 11 and 17′ respectively). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

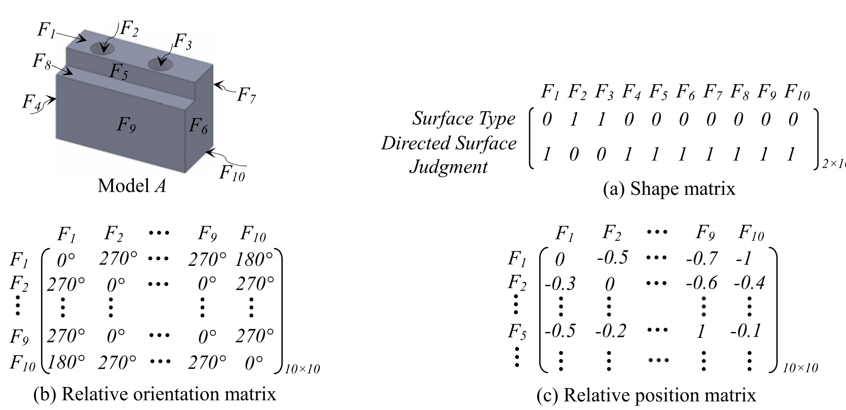
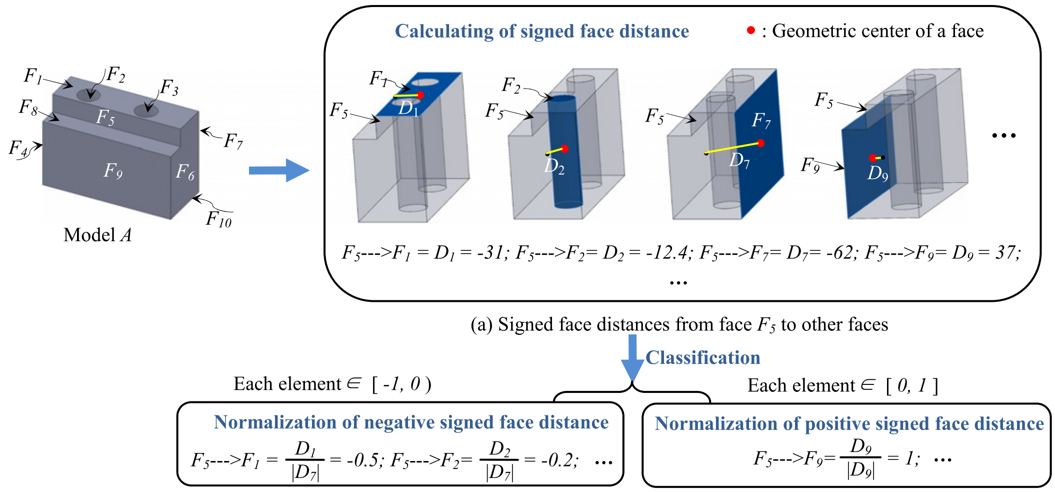


Fig. 3. Illustration of the ordered face set $\{F_1, F_2, F_3, F_4, F_5, F_6, F_7, F_8, F_9, F_{10}\}$ and its descriptor.

4.1. Descriptor of ordered face set

As the two given models have different resolutions [9,59], the faces in a shape frame of the candidate model may not always be adjacent to each other. Therefore, the approaches (such as graphs, shape distributions, etc. presented in [7,17,19,81]) frequently used to describe a geometric model are not suitable to describe an arbitrary face set here. Consequently, to facilitate face sets similarity assessment and inspired by [11], we adopt a quantitative descriptor composed of three matrixes (**shape matrix**, **relative orientation matrix** and **relative position matrix**) to describe the rotation invariant characters of a face set.

The shape matrix, describing the rotation invariant characters of each face in a face set FS, is a $2 \times M$ matrix where M is the face count of FS. The value in position $[1, i]$ integrating with the value in position $[2, i]$ describes the rotation invariant characters of the ith face. The relative orientation matrix, describing the relative orientations between each pair of faces in FS, is an $M \times M$ matrix. The value in position $[i, j]$ is the relative orientation value between the ith face and the jth face. Meanwhile, the relative position matrix, describing the relative positions between each pair of faces in FS, is also an $M \times M$ matrix. The value in position $[i, j]$ is the value that represents the distance from the ith face to the jth one semantically. For example, the shape matrix (Fig. 3(a)) and the relative orientation matrix (Fig. 3(b)) integrating with the relative



(a) Signed face distances from face F_5 to other faces

(b) Normalization of the signed face distances from face F_5 to other faces

Fig. 4. The process of calculating signed face distance.

position matrix (Fig. 3(c)) comprise the descriptor of the ordered face set $\{F_1, F_2, F_3, F_4, F_5, F_6, F_7, F_8, F_9, F_{10}\}$.

Since each matrix is order-related and each face set should be corresponding to a unique descriptor, each face set descriptor is uniquely given accompanied with a face permutation in this work, i.e. different face permutations correspond to different face set descriptors for the same face set. In this case, each face set descriptor in this work is actually the descriptor of an **ordered face set**.

4.1.1. Construction of shape matrix

The directed surface judgment and surface type are used to describe the rotation invariant characters of each face. Especially, the surface type property of a face is set to 0, 1, 2, 3 or 4 when that face is a plane, cylinder, cone, sphere or any other kind respectively. The directed surface judgment of a face is set to 1 or 0 when the surface containing that face is a directed surface or not respectively. Such rule is applied to each kind of face in this work. Fig. 3(a) shows the constructed shape matrix for the ordered face set $\{F_1, F_2, F_3, F_4, F_5, F_6, F_7, F_8, F_9, F_{10}\}$.

4.1.2. Construction of relative orientation matrix

In this paper we use angle to measure the relative orientation between two faces. However, it is not enough to unambiguously describe the relative orientation between two faces only depending on their intersection angle. For example, in Fig. 3, the intersection angle between face F_5 and face F_8 is 90° which is equal to the intersection angle between face F_5 and face F_1, while the relative orientations between face F_5 & face F_8 and between face F_5 & face F_1 are totally different. As a result, to properly describe the relative orientation between two faces by using angle, we not only include the intersection angle between the two faces, but also consider the concavity and convexity of their intersection. For example, in Fig. 3 the face intersection angle between face F_5 and face F_8 is 90°, after considering the intersection convexity, the angle describing the relative orientation between face F_5 and face F_1 becomes 270°. To calculate the intersection angle between two faces, we adopt the methods described in [82]. Fig. 3(b) shows the constructed relative orientation matrix for the ordered face set $\{F_1, F_2, F_3, F_4, F_5, F_6, F_7, F_8, F_9, F_{10}\}$.

4.1.3. Construction of relative position matrix

As any two faces in a shape frame are not always parallel to each other, thus, we adopt a signed face distance to assess the relative position from one face to the other uniformly. The formulized definition of **signed face distance** from face F_i to face F_j is as follows function (4.1):

$$\textit{Signed Face Distance}(F_i, F_j) = \begin{cases} sign * E(cp_j, vp\_f_i); & S = Dot(cp_j - vp\_f_i, NF_i), \\ \quad \text{if } F_i \text{ is a plane face} & \\ sign * E(cp_j, vp\_x_i); & S = Dot(cp_j - vp\_x_i, CF_i), \\ \quad \text{else.} & \end{cases} \tag{4.1}$$

If $S < 0$, $sign = -1$, else $sign = 1$;

E is the Euclidean distance.

Here, F_i and F_j represent two faces; surface f_i containing face F_i; cp_j is the geometric center of F_j. If F_i is a plane face, then vp_f_i is the vertical projection point of cp_j onto f_i and NF_i is the normal of f_i on vp_f_i. If F_i is a quadratic face having one axis (such as cylinder face), then vp_x_i is the vertical projection point of cp_j onto that axis, otherwise, vp_x_i is the geometric center of F_i. If F_i is a quadratic face, then we multiply the normal of f_i on fip (fip is the first intersection point between f_i and the ray starting from cp_j while passing through vp_x_i) by dc (dc is a direction coefficient: if the directed surface judgment of F_i is $= 1$, then $dc = 1$, otherwise, $dc = -1$) to get the vector CF_i. If F_i is a plane face, then S is the dot product between vector $(cp_j - vp\_f_i)$ and NF_i, otherwise, S is the dot product between vector $(cp_j - vp\_x_i)$ and CF_i.

Especially, each signed face distance not only represents the distance between two faces geometrically but also reflects their spatial position relationship semantically. As shown in Fig. 4, aided with the methods described in [82], the signed face distances from face F_5 to other ones can be calculated (shown in Fig. 4(a)). Because the geometric center of F_1 is behind F_5 in Fig. 4(a), thus $D_1 = -31$. Distinctively, $D_9 = 37$ since the geometric center of F_9 is in front of F_5.

Since the query model and the candidate model have different sizes, it is natural to normalize each signed face distance before carrying out meaningful comparisons among the relative positions of the faces respectively belonging to the two models. After calculating all the signed face distances from one face F_i to the other faces belonging to the same model, we do the normalization as follows:

(1) *Classifying all the signed face distances into two groups based on their signs*;
(2) *Normalizing each signed face distance in its own group while keeping its sign.*

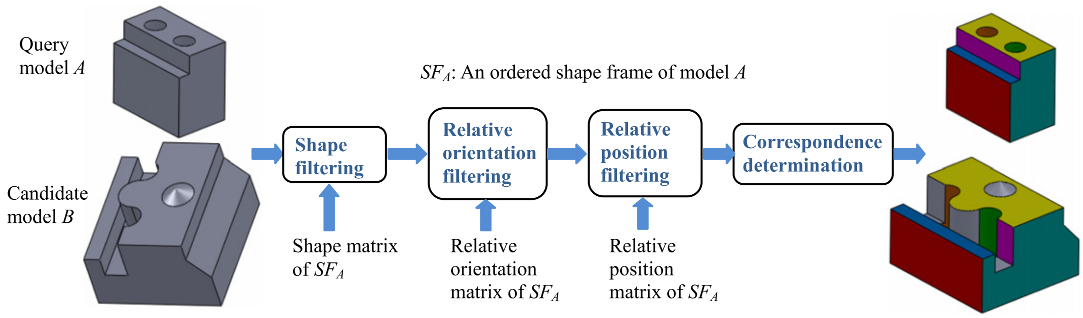


Fig. 5. The process of determining corresponding shape frames and corresponding faces. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Especially, normalizing signed face distances group by group while preserving their signs can keep two faces' semantically relative positions. After normalizing, each signed face distance $\in [-1, 0) \cup [0, 1]$. For example, in Fig. 4(b), the signed face distance from face F_5 to F_1 becomes $D_1/|D_7| = -0.5$, since D_7 possesses the longest absolute value among all the negative signed face distances originating from F_5. On the other hand, the signed face distance from face F_5 to F_9 becomes $D_9/|D_9| = 1$, since D_9 possesses the longest absolute value among all the positive signed face distance originating from F_5 as well.

When all of the signed face distances in a model are calculated, the relative position matrix for an ordered face set can be easily constructed. Fig. 3(c) shows the constructed relative position matrix for the ordered face set $\{F_1, F_2, F_3, F_4, F_5, F_6, F_7, F_8, F_9, F_{10}\}$.

4.2. Similarity assessment of ordered face sets

After constructing the ordered face set descriptors for two ordered face sets, the similarity between them can be measured by comparing their corresponding matrixes. To this end, we define the following distance measurement to assess the similarity between two ordered face sets:

Ordered face sets distance: Let us assume SM_1 & SM_2 (two shape matrixes), OM_1 & OM_2 (two relative orientation matrixes) and PM_1 & PM_2 (two relative position matrixes) respectively constructed for two ordered face sets FS_1 & FS_2. The face counts of the two face sets are equal to K. Then,

$$d_{shape}(FS_1, FS_2) = \begin{cases} +\infty, & SM_1(1, i) \neq SM_2(1, i) \\ \quad \text{and/or} & FSM_1(2, i) \neq FSM_2(2, i), \quad i = 1, 2, \ldots, K \\ 0, & \text{else} \end{cases} \tag{4.2}$$

$$d_{OM}(FS_1, FS_2) = \sum_{i=1}^{K} \sum_{j=1}^{K} |OM_1(i, j) - OM_2(i, j)| \tag{4.3}$$

$$d_{PM}(FS_1, FS_2) = \sum_{i=1}^{K} \sum_{j=1}^{K} |PM_1(i, j) - PM_2(i, j)| \tag{4.4}$$

$$d(FS_1, FS_2) = d_{shape}(FS_1, FS_2) + d_{OM}(FL_1, FL_2) + d_{PM}(FL_1, FL_2). \tag{4.5}$$

$d_{shape}(FS_1, FS_2)$ denotes the shape distance between FS_1 and FS_2. Especially, when two faces, at the same permutation place respectively belonging to FS_1 and FS_2, have different surface types and/or different directed surface judgments, we assign $+\infty$ to $d_{shape}(FS_1, FS_2)$. $d_{PM}(FS_1, FS_2)$ and $d_{OM}(FS_1, FS_2)$ respectively denote the relative position distance (between FS_1 and FS_2) and the relative orientation distance (between FS_1 and FS_2). $d(FS_1, FS_2)$ denotes the ordered face sets distance between FS_1 and FS_2. Obviously, the smaller the ordered face sets distance between two ordered face sets, the greater the similarity between them.

4.3. The process of corresponding shape frames and corresponding faces determination

According to the definition of corresponding shape frame, determining the corresponding shape frame of the candidate model (corresponding to SF_Q) is to find the shape frame SF_C (in the candidate model) that is the most similar to SF_Q. Besides, based on the method adopted to assess the similarity between two ordered face sets, the SF_C should have a special face permutation matching along with that in SF_Q. In other words, to find the corresponding shape frame of the candidate model (corresponding to SF_Q) is to find the ordered SF_C (a shape frame accompanied with a face permutation is called an **ordered shape frame**) that has the minimum ordered face sets distance to the **ordered SF_Q** (the face permutation of SF_Q is random). To this end, a natural way is to employ an exhaustive combination search for the target ordered shape frame (having the same face count as SF_Q) from the candidate model. Note that this naive brute-force search method is aimless and has a factorial time complexity. Furthermore, there is no pre-existing solution for identifying shape frames from the candidate model based on the query model according to our review of literature.

Regarding the above problems, we first apply a heuristic filtering process (Fig. 5) (based on the definition of ordered face sets distance) to find out all the possible ordered shape frames of the candidate model (from all of its ordered face sets) according to the ordered SF_Q. We then calculate the ordered face sets distances between the ordered SF_Q and each found possible ordered shape frame. Finally, the possible ordered shape frame SF_C owning the minimum ordered face sets distance is determined as the corresponding shape frame of SF_Q. Meanwhile, according to the definition of corresponding faces and the definition of ordered face sets distance, each pair of corresponding faces is just the pair of faces respectively belonging to the ordered SF_C and the ordered SF_Q at the same permutation place.

4.3.1. Ordered shape frame finding based on the process of heuristic filtering

Based on the definition of ordered face sets distance, an ordered face set, to be a possible ordered shape frame of the candidate model according to the ordered SF_Q, should have small distances to the ordered SF_Q respectively in aspects of each face's rotation invariant characters and face layout. According to the above sense, by composing of shape filtering, relative orientation filtering and relative position filtering, we design the heuristic filtering process (shown in Fig. 5) to find out all the possible ordered shape frames of the candidate model (from all the ordered face sets of the candidate model) according to the ordered SF_Q.

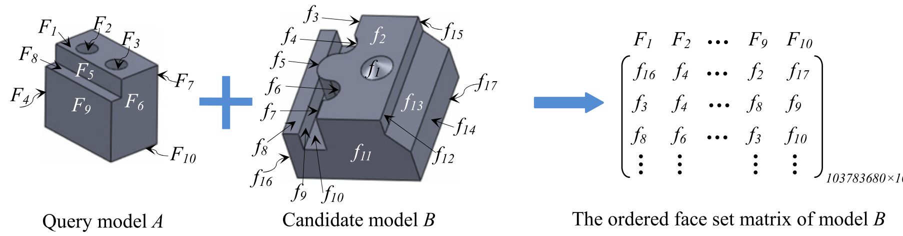


Fig. 6. Illustration of ordered face set matrix for the ordered shape frame $\{F_1, F_2, F_3, F_4, F_5, F_6, F_7, F_8, F_9, F_{10}\}$.

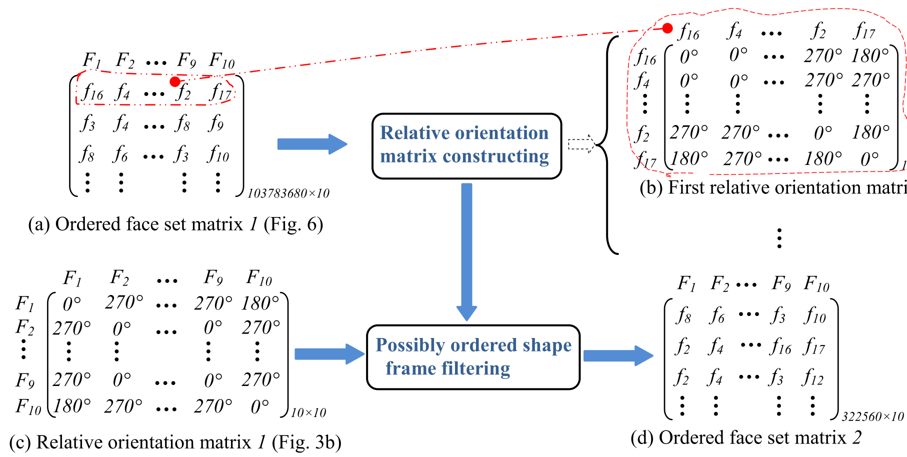


Fig. 7. The process of filtering out possible ordered shape frames based on relative orientation.

4.3.1.1. Shape filtering. According to function (4.2), an ordered face set, to be a possible ordered shape frame of the candidate model according to the ordered SF_Q, should have the same shape matrix as the ordered SF_Q. In this work, a shape filtering, as the first step of our heuristic filtering process (Fig. 5), is carried out to preliminarily find out all the possible ordered shape frames of the candidate model according to the ordered SF_Q. Then, an "ordered face set matrix" is used to save all the possible ordered shape frames. For example, in Fig. 6, there are totally 103 783 680 possible ordered shape frames of model *B* according to the ordered shape frame $\{F_1, F_2, F_3, F_4, F_5, F_6, F_7, F_8, F_9, F_{10}\}$.

4.3.1.2. Relative orientation filtering. According to function (4.3), an ordered face set, to be a possible ordered shape frame of the candidate model according to the ordered SF_Q, should have a small distance from its own relative orientation matrix to that of the ordered SF_Q. Here, if the distance is larger than ξ ($\xi = 5°$), then the ordered face set is not a possible ordered shape frame and eliminated from the ordered face set matrix directly. In this work, a relative orientation filtering, as the second step of our heuristic filtering process (Fig. 5), is carried out to further filtering out possible ordered shape frames from the ordered face set matrix.

For example, the first relative orientation matrix (Fig. 7(b)) is constructed for the possible ordered shape frame $SF_T\{f_{16}, f_4, \ldots, f_2, f_{17}\}$ (Fig. 7(a)). The relative orientation matrix (Fig. 7(c)) is constructed for the ordered $SF_Q\{F_1, F_2, F_3, F_4, F_5, F_6, F_7, F_8, F_9, F_{10}\}$. Since the distance between the two values in the [1, 2] positions respectively belonging to the above is 270°, which is larger than ξ. Thus, SF_T is eliminated from the matrix 1 (Fig. 7(a)), and the ordered face set matrix 2 (Fig. 7(d)) is the output of this filtering step.

4.3.1.3. Relative position filtering. According to function (4.4), an ordered face set, to be a possible ordered shape frame of the candidate model according to the ordered SF_Q, should have a small distance from its own relative position matrix to that of the ordered SF_Q. Here, if the distance is larger than δ ($\delta = 0.1$), then the ordered face set is not deemed as a possible ordered shape frame and eliminated from the ordered face set matrix directly. In this work, a relative position filtering, as the third step of our heuristic filtering process (Fig. 5), is carried out to further filtering out possible ordered shape frames from the ordered face set matrix.

For example, the first relative position matrix (Fig. 8(b)) is constructed for the possible ordered shape frame $SF_T\{f_8, f_6, \ldots, f_3, f_{10}\}$ (Fig. 8(a)). The relative position matrix (Fig. 8(c)) is constructed for the ordered $SF_Q\{F_1, F_2, F_3, F_4, F_5, F_6, F_7, F_8, F_9, F_{10}\}$. Since the distance between the two values in the [1, 10] positions respectively belonging to the above matrixes is 0.8, which is larger than δ. Thus, SF_T is eliminated from the matrix 2 (Fig. 8(a)), and the ordered face set matrix 3 (Fig. 8(d)) is the output of this filtering step.

After the shape filtering and relative orientation filtering, any possible ordered shape frame in the ordered face set matrix has the following character: the *i*th face in that possible ordered shape frame has the same face rotation invariant characters as the *i*th face in the ordered SF_Q, and the two faces have similar relative orientations relative to other faces in their own ordered face sets ($i = 0, 1, \ldots, K$, K is the face count of SF_Q). The relative position filtering, according to the ordered SF_Q, will eliminate all the possible ordered shape frames from the ordered face set matrix when they do not characterize the overall shape of the candidate model. Such case will be right if we consider that the overall shape of the candidate model is similar to that of the query model.

4.3.2. Correspondence determination

After the above heuristic filtering process, based on the function (4.5), the possible ordered shape frame SF_C in the ordered face set matrix, owning the greatest similarity with the ordered SF_Q, can be easily found (SF_C is just the corresponding shape frame of SF_Q). Meanwhile, according to the definition of corresponding faces and the definition of ordered face sets distance, each pair of corresponding faces is just the pair of faces respectively belonging to the ordered SF_C and the ordered SF_Q at the same permutation place.

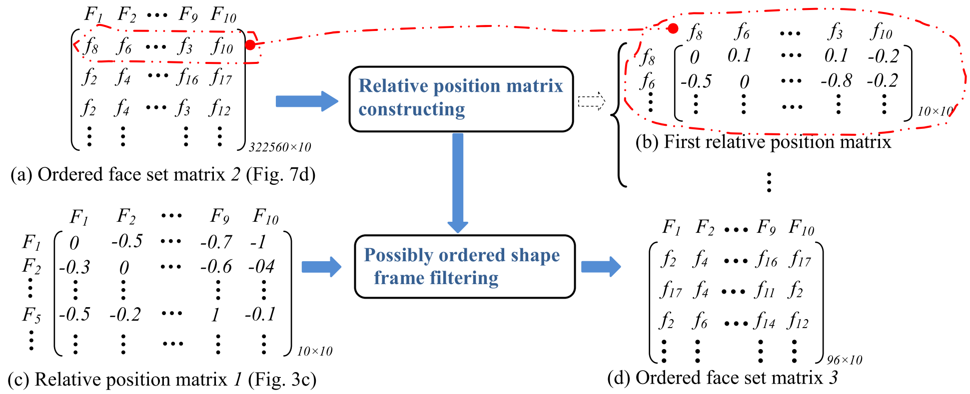


Fig. 8. The process of filtering out possible ordered shape frames based on relative position.

For example, before heuristic filtering, there are 103 783 680 possible ordered shape frames in model *B* according to the ordered $SF_Q\{F_1, F_2, F_3, F_4, F_5, F_6, F_7, F_8, F_9, F_{10}\}$ (Fig. 6). After heuristic filtering, the count of possible ordered shape frames is reduced to 96 (Fig. 8(d)). Finally, the ordered shape frame $SF_C\{f_2, f_4, f_6, f_{15}, f_7, f_{11}, f_{14}, f_8, f_{16}, f_{17}\}$ is determined as the corresponding shape frame of SF_Q. Meanwhile, the faces in the same permutation place respectively belonging to the ordered SF_Q and the ordered SF_C, with the same color (except gray) in Fig. 5, are determined as the corresponding faces between the query model and the candidate model.

4.3.3. Completing of corresponding faces

Although some faces in the candidate model have no corresponding faces in the query model based on the method mentioned in the above sections, they will also change along with the shape adaptation. Here, for the convenience of dimension relationship establishment in the following sections, if two faces in the candidate model belong to the same feature and share the same surface, then we entitle them to have the same corresponding face(s) in the query model. For example, in Fig. 2, face 7 and face 7′ can be determined as a pair of corresponding faces in the above sections. As the cylinder faces 7′, 8′, 10′, 12′, 13′ and 15′ belong to the same feature and share the same cylinder surface, then each of them is corresponding to face 7. Similarly, if two faces in the query model belong to the same feature and share the same surface, then we entitle them to have the same corresponding face(s) in the candidate model. For example, because the cylinder faces 7 and 9 belong to the same feature and share the same cylinder surface, both face 7 and face 9 are corresponding to the faces 7′, 8′, 10′, 12′, 13′ and 15′.

5. Establishment of dimension relationship

In the first place, the natural way to transfer the query model's shape to the candidate model is to establish dimension relationships (algebraic relations) between each pair of corresponding dimensions & between each dimension and its corresponding dimension chain. Next, we make the corresponding dimensions and corresponding dimension chains in the candidate model have the same values as those of their counterparts in the query model. However, identifying the corresponding dimensions and dimension chains among the two models is not a trivial work since the parametric way for a parametric solid model can be very flexible.

To solve the above issue, firstly, we adopt a promotion method to promote each dimension in a model to be defined on/between its constrained entity(s) (edge, axis of a quadratic face and face) since dimensions mainly exist in 2D sketches and features, such as the promotion examples related to dimension D_1 and R_1 in Fig. 9. Then, we identify corresponding dimensions and the corresponding dimension chains between the two given models based on their corresponding faces (since corresponding edges and corresponding axes among the two models can be deduced from corresponding faces easily according to their definition (Section 3)).

5.1. Dimension promotion

Generally speaking, the mainstream CAD systems for constructing a parametric solid model adopt attaching new features to the previous features [83,84]. Furthermore, the extrusion features (each of them is formed by sweeping the 2D sketch profile along the direction perpendicular to the sketch plane containing the profile) and revolution features (each of them is constructed by revolving a 2D sketch profile about an axis by a position angle) [83] are the most common and fundamental features in current CAD systems. Thus, we hypothesize that the query model and the candidate model are constructed mainly by using extrusion features and/or revolution features.

Without losing generality, we consider two types of dimension constraints on a 2D sketch: distance dimension and angle dimension. Especially, each distance dimension is defined between two sketch points, between sketch point & sketch line or between two sketch lines; and each angle dimension is defined between two sketch lines while using interior angle as its value. We do not consider dimensions among curves since such sketch elements can be positioned with respect to each other by specifying constraints between their reference sketch points or sketch lines [48].

As we know, each sketch point and each sketch segment (line or curve), after sweeping, correspond to an edge (or axis) and a face respectively. For example, in Fig. 10(a) and (b), sketch point sp_2 corresponds to edge E_1 (swept from sp_2); sketch point cp_1 corresponds to the axis F_3_A (swept from cp_1); sketch line sp_1sp_2 corresponds to face F_1 (swept from sp_1sp_2). Because each edge is always deemed as the intersection of two faces in a manifold model, we adopt $F_i \cap F_j$ to represent an edge E_i (face F_i and face F_j are respectively swept from two segments that make a point p_i as their intersection point corresponding to edge E_i after sweeping). For example, edge E_1 is represented as $F_1 \cap F_2$.

Because each sketch element (point, line or curve) corresponds to an entity (edge, axis or face) after sweeping, a dimension constraining on/between a sketch element(s) (by being defined on/between it/them) is actually constraining the distance or angle on/between the entity(s) corresponding to that sketch element(s). Accordingly, promoting each dimension in a parametric solid model is to find the entity(s) constrained by that dimension. Table 1 shows the method of dimension promotion for each type of dimension.

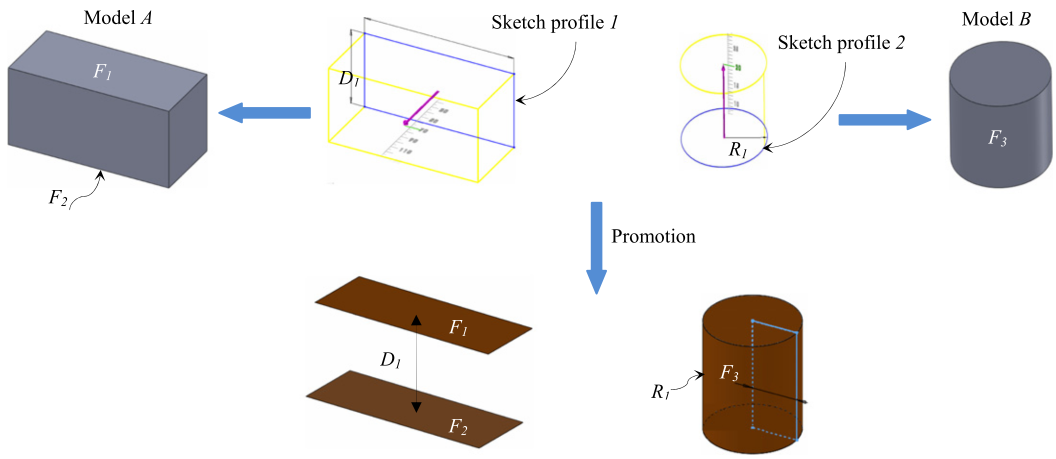
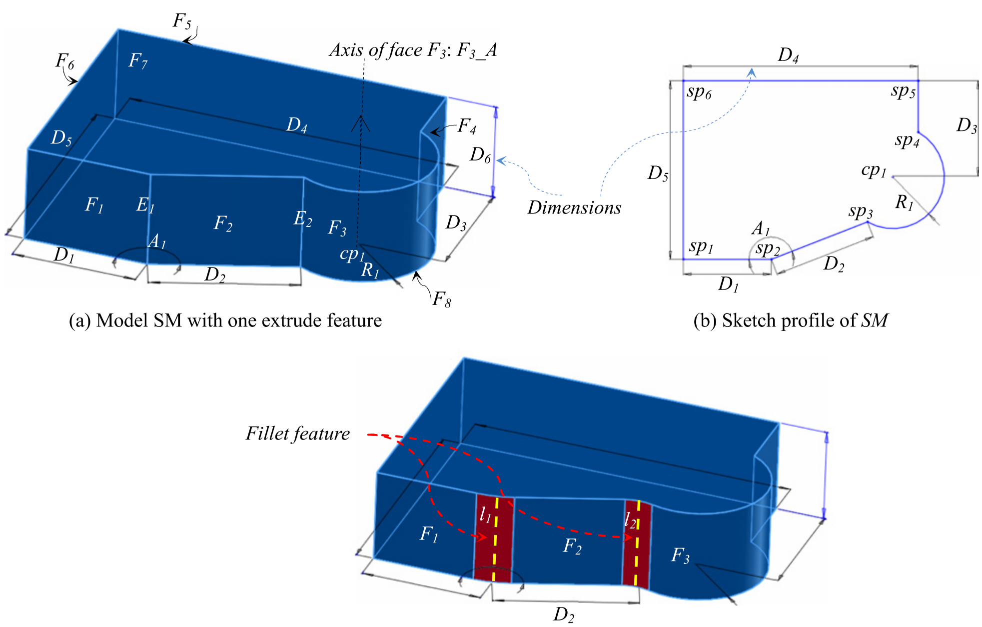


Fig. 9. Illustration of dimension promotions (D_1 is promoted to be defined between face F_1 and F_2 while R_1 is promoted to be defined on F_3).



(a) Model SM with one extrude feature

(b) Sketch profile of *SM*

(c) Model SM adds a fillet feature along the edge E_1 and E_2

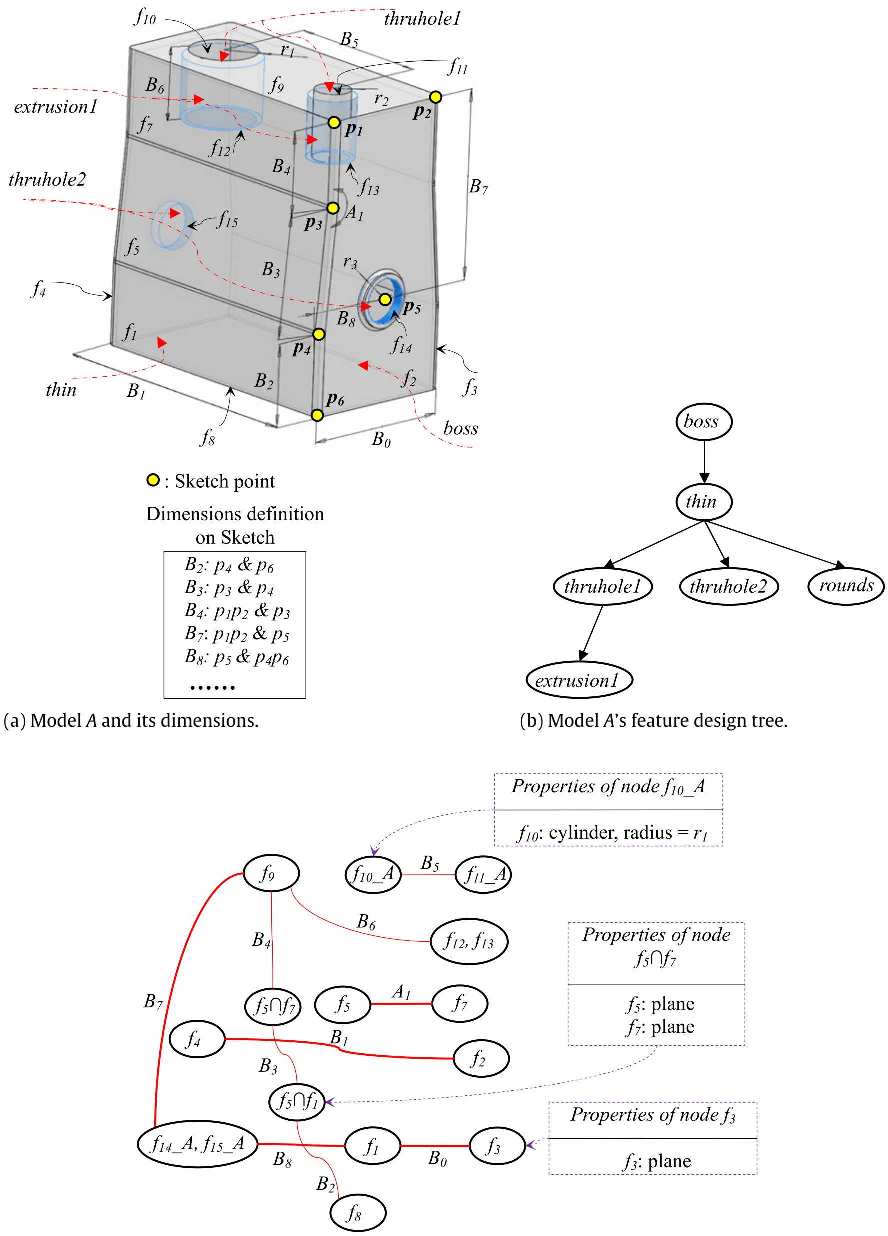
Fig. 10. Model SM and its sketch profile (A_1 is an angle dimension defined between sketch line sp_1sp_2 and sp_2sp_3; D_1, D_2, D_3 and D_5 are the distance dimensions defined between sketch point sp_2 & sketch line sp_1sp_6, between sketch point sp_2 & sp_3, between sketch point cp_1 & sketch line sp_5sp_6 and between sketch line sp_5sp_6 & sp_1sp_2 respectively).

Table 1
Lookup table for dimension promotion (P_1 and P_2 represent two sketch points; L_1 and L_2 represent two sketch lines; F_1 and F_2 represent two faces; E_1 and E_2 represent two edges; each item 'X: Y & Z' represents that X is defined between Y & Z; each item 'X: Y' represents that X is defined on Y).

Definition before promotion		Definition after promotion	Description
Sketch distance dimension D	$D: P_1$ & P_2	$D: E_1/F_1\_A$ & E_2/F_2_A	E_1/F_1_A & E_2/F_2_A are respectively swept from P_1 & P_2.
	$D: P_1$ & L_1	$D: E_1/F_1\_A$ & F_2	E_1/F_1_A & F_2 are respectively swept from P_1 & L_1.
	$D: L_1$ & L_2	$D: F_1$ & F_2	F_1& F_2 are respectively swept from L_1 & L_2.
Sketch angle dimension A	$A: L_1$ & L_2	$A: F_1$&F_2	
Parametric dimension D of sketch curve C	All definition types	$D: F_1$	F_1 is swept from curve C.
Extrusion dimension D of feature EF		$D: F_1$&F_2	F_1&F_2 are the starting face & end face of feature EF respectively.

For example, in Fig. 10(b), D_1 constrains the distance between sp_2 & sp_1sp_6. Edge E_1 and face F_6 are respectively swept from sp_2 and sp_1sp_6. Meanwhile, $E_1 = F_1 \cap F_2$ (face F_1 and F_2 are swept from sp_1sp_6 and sp_2sp_3 respectively). Thus, the distance between edge $F_1 \cap F_2$ and face F_6 is constrained by D_1. So, we promote D_1 to be defined between $F_1 \cap F_2$ and F_6. Similarly, D_2 is promoted to be defined between $F_1 \cap F_2$ and $F_2 \cap F_3$ (F_3 is swept from the arc sp_3sp_4) while D_5 is promoted to be defined between F_1 and F_5 (swept from sketch line sp_5sp_6). Because center-point cp_1 of sketch arc sp_3sp_4 is swept to the axis F_3_A of face F_3, thus, D_3 is promoted to be defined between F_3_A and F_5.

Since changing the value of R_1, as the radius of sketch arc sp_3sp_4, is changing the radius of the cylinder face F_3, thus, we promote R_1 to be defined on F_3. Changing the value of A_1 is changing the angle between sketch line sp_1sp_2 and sp_2sp_3, which also affect the relative angle between face F_1 and F_2, thus, A_1 is promoted to be defined between F_1 and F_2. For the extrusion dimension D_6, in Fig. 10, constrains the distance between face F_8 and F_7 (respectively



(a) Model A and its dimensions.

(b) Model A's feature design tree.

(c) 3D dimension constraint graph *fdg*.

Fig. 11. A 3D dimension constraint graph (dimensions in model A (a), with a feature design tree shown in (b), are promoted and represented in *fdg* (c)).

corresponding to the starting face and the end face of model SM). So, we promote D_6 to be defined between F_8 and F_7.

Especially, using two faces' intersection to represent an edge (explicit or implicit), not only keeps a consistent definition form for all promoted dimensions, but also explicitly reflects the intrinsically geometric constraint among faces. For example, in Fig. 10(c), edge E_1 and E_2 disappear from model SM after adding a fillet feature. However, the distance between the implicit edge E_1 (represented as l_1) and the implicit edge E_2 (represented as line l_2) is still constrained by D_2.

5.2. Construction of 3D dimension constraint graph

After promoting all the dimensions in a parametric solid model to be defined on/between constrained entity(s), a **3D dimension constraint graph** $3DDCG = (V, E)$ is adopted to uniformly represent them. V is the node set in the graph. Each node represents a constrained entity(s). We make each dimension that constrains the shape of the surface containing a face F_i as a property of the node representing a constrained entity(s) that belongs to F_i. For example, as shown in Fig. 11(c), the node f_3 represents face f_3; node f_{11}_A represents the axis of face f_{11}; node $f_5 \cap f_7$ represents the edge between face f_5 & f_7, and the radius of f_{10} is set as the property of node f_{10}_A since f_{10}_A is the axis of f_{10}. E is the graph edge set in the graph. Each graph edge represents a dimension.

Additionally, since any two constrained entities, belonging to the same feature and sharing the same geometric element (surface, line or axis), always have the same distances and orientations relative to other geometric elements in the same model. So, the constrained entities satisfying the above conditions are represented by the same graph node. For example, in Fig. 11(a), both face f_{12} and f_{13} belong to *extrusion*1 feature and share the same plane surface according to the feature design tree (Fig. 11(b)). Because face f_{12} is referred to the promoted dimension B_6 (Fig. 11(a) and (c)), f_{12} and f_{13} are represented by the same node $\{f_{12}, f_{13}\}$. Similarly, both the axis f_{14}_A of face f_{14} and the axis f_{15}_A of f_{15} belong to *thruhole*2

Table 2
Lookup table for corresponding dimensions (the nodes in each pair of (QN_i, QN_j) and (CN_i, CN_j) are unordered; F_i, F_j, F_k and F_h represent four faces in the query model while f_i, f_j, f_k and f_h represent four faces in the candidate model; each item '$X = Y$' represents that constrained entity Y is represented by graph node X).

Case	Node pair (QN_i, QN_j)	Node pair (CN_i, CN_j)	Description
1	$QN_i = F_i, QN_j = F_j$	$CN_i = f_i, CN_j = f_j$	(F_i corresponds to f_i & F_j corresponds to f_j) or (F_i corresponds to f_j & F_j corresponds to f_i).
2	$QN_i = F_i\_A, QN_j = F_j\_A$	$CN_i = f_i\_A, CN_j = f_j\_A$	(F_i_A corresponds to f_i_A & F_j_A corresponds to f_j_A) or (F_i_A corresponds to f_j_A & F_j_A corresponds to f_i_A).
3	$QN_i = F_i\_A, QN_j = F_j$	$CN_i = f_i\_A, CN_j = f_j$	F_i_A corresponds to f_i_A & F_j corresponds to f_j.
4	$QN_i = F_i \cap F_j, QN_j = F_k$	$CN_i = f_i \cap f_j, CN_j = f_k$	$F_i \cap F_j$ corresponds to $f_i \cap f_j$ and F_k corresponds to f_k.
5	$QN_i = F_i \cap F_j, QN_j = F_k\_A$	$CN_i = f_i \cap f_j, CN_j = f_k\_A$	$F_i \cap F_j$ corresponds to $f_i \cap f_j$ and F_k_A corresponds to f_k_A.
6	$QN_i = F_i \cap F_j, QN_j = F_k \cap F_h$	$CN_i = f_i \cap f_j, CN_j = f_k \cap f_h$	$F_i \cap F_j$ corresponds to $f_i \cap f_j$ and $F_k \cap F_h$ corresponds to $f_k \cap f_h$ or $F_i \cap F_j$ corresponds to $f_k \cap f_h$ and $F_k \cap F_h$ corresponds to $f_i \cap f_j$

feature and share the same axis. Because B_7 is promoted to be defined between f_{14}_A and face f_9, we use node $\{f_{14}\_A, f_{15}\_A\}$ to represent the constrained entity f_{14}_A and f_{15}_A.

Furthermore, using *3DDCG* to represent all promoted dimensions belonging to a parametric solid model, not only reflects the direct dimension constraints between two constrained entities, but also indicates the indirect dimension constraints between two constrained entities which are connected by a sequence of dimensions (each pair of the adjacent dimensions in that sequence is connected by the same constrained entity). For example, in Fig. 11(a), although there is no dimension that constrains the distance between face f_9 and f_8, a dimension path (dimension chain) (B_4, B_3, B_2) can be found between node f_9 and f_8 in Fig. 11(c). We can see that the relative distance between face f_9 and f_8 is controlled by that path.

5.3. Identification of corresponding dimensions and corresponding dimension chains

After building the two 3D dimension constraint graphs FDG_Q and FDG_C respectively for the query model and the candidate model, based on their corresponding faces, we can identify their corresponding dimensions and corresponding dimension chains conveniently.

5.3.1. Identification of corresponding dimensions

Obviously, according to the dimension promotion method, given two corresponding faces F_i and f_i, it is easy to establish corresponding dimension relationships among the dimensions respectively constraining the surface shape of F_i and the surface shape of f_i. For example, in Fig. 12(c) and (d), R_2 as the dimension that constrains the radius of the surface containing face F_7 is promoted to be defined on face F_7. Similarly, r_2 as the dimension that constrains the radius of the surface containing face f_{11} is promoted to be defined on face f_{11}. Because face F_7 corresponds to face f_{11}, R_2 and r_2 are corresponding dimensions.

Given two dimensions D_i and B_i, as two graph edges connecting two pairs of nodes (QN_i, QN_j) and (CN_i, CN_j) which respectively belong to FDG_Q and FDG_C, if the two pairs of nodes contain the characters that follow into one of the 6 cases (shown in Table 2) while D_i and B_i have the same dimension type (distance or angle), then D_i and B_i are corresponding dimensions:

For example, in Fig. 12(c) and (d), D_6 is promoted to be defined between the axis of face F_6 and the axis of face F_7, and B_5 is promoted to be defined between the axis of face f_{10} and the axis of face f_{11}; meanwhile f_{10} & f_{11} respectively correspond to F_6 & F_7, thus f_{10}_A & f_{11}_A respectively correspond to F_6_A & F_7_A according to the definition of corresponding axis. Finally, based on Table 2(2) and the definition of corresponding dimension, D_6 and B_5 are corresponding dimensions.

Table 3
Lookup table of searching priorities for an optimal dimension path (each item 'Node $= X$' in the table represents that the constrained entity 'X' is represented by graph node 'Node').

Start node	End node		
	Node = Face	Node = Edge	Node = Axis
Node = Face	High	Middle	Low
Node = Edge	High	Middle	Low
Node = Axis	High	Middle	Low

5.3.2. Identification of corresponding dimension chains

Given two pairs of graph nodes (QCE_i, QCE_j) and (CCE_i, CCE_j) respectively belonging to FDG_Q and FDG_C, and dimension D as the graph edge connects QCE_i and QCE_j, we have to identify the corresponding dimension chain in the candidate model corresponding to D when the following conditions are satisfied: (1) the two pairs of graph nodes contain the characters that follow into one of the 6 cases listed in Table 2; (2) there is no graph edge (having the same dimension type as D) connecting CCE_i & CCE_j directly. In other words, in the above situation, we have to find a dimension chain constraining the constrained entities represented by node CCE_i and CCE_j respectively and having the same effect as D does on the constrained entities represented by node QCE_i and QCE_j respectively. Such case has been mentioned in Section 5.2 that the dimension chain (B_4, B_3, B_2) constrains the distance between face f_9 and f_8. With the help of 3D dimension constraint graph, to identify the corresponding dimension chain in the candidate model is to find a dimension path from FDG_C.

5.3.2.1. A heuristic method for the optimal dimension path identifying. Adopting depth-first search method [85] accompanied with the requirement of no graph edge repeating, as we can see that more than one dimension path, starting from any node i while ending with any node j, may be found from FDG_C. For example, in Fig. 13(b), (B_1, B_2, B_3) and (A_1, A_2, A_3, A_4) are the two none graph edge repeated dimension paths starting from node f_1 while ending with node f_5. To find the **optimal dimension path** from FDG_C that corresponds to D, we combine the following searching priorities (Table 3) into the depth-first searching method.

Optimal dimension path: After embedding the priorities (Table 3) into the depth-first searching method, the first found dimension path, connecting the two nodes CCE_i and CCE_j, is deemed as the **optimal dimension path** (corresponding dimension chain) for D.

Here, the searching priorities are designed according to the common automatic dimensioning mechanism, adopted by some mainstream CAD systems (such as SolidWorks), which automatically defines each distance dimension on a sketch from between two paralleled lines, to between one line and one point, to between two points in a descending priority for facilitating machining and measuring. Since each line will be swept to a

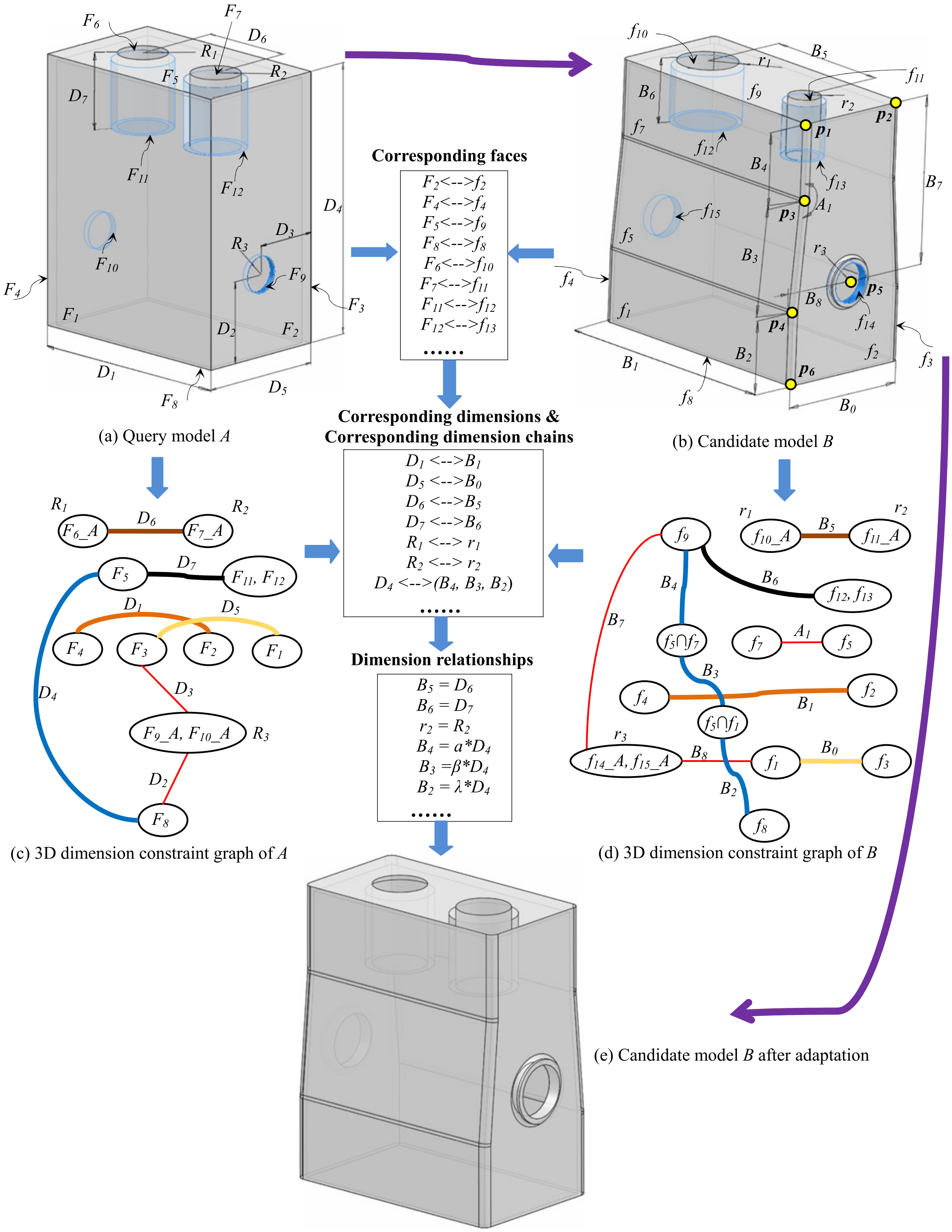


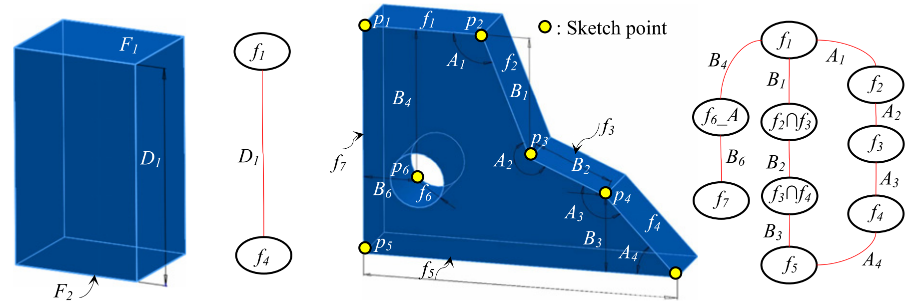
Fig. 12. Illustration of the process to establish dimension relationship (different colors (except red) are added to the graph edges in (c) and (d) to represent corresponding dimensions and corresponding dimension chains). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

face while each point is swept to an edge or an axis, thus each graph node representing a face(s) has a higher priority than each graph node representing an axis or an edge in each searching step. Furthermore, as edges and faces usually describe the shape of a solid model, while an axis is often used to determine the position of an embedded feature (such as the hole in Fig. 13(b)), thus each graph node representing an edge(s) has a higher priority than each graph node representing an axis in each searching step.

Moreover, because each angle dimension and each distance dimension are usually considered to respectively control the relative orientation and the relative distance between two constrained entities [86,87], we assume that all the dimensions in an optimal dimension path should have the same type (distance or angle).

For example, in Fig. 13(a) and (b), if $D = D_1$, then the optimal dimension path searching process starts from node f_1 firstly, then the next searching node is $f_2 \cap f_3$ according to Table 3 since it has a higher priority than node f_6_A (the axis of face f_6). After that the next node is $f_3 \cap f_4$, and the searching process ends with node f_5. Finally, the optimal dimension path (B_1, B_2, B_3) is found. Similarly, in Fig. 12, the dimension path (B_4, B_3, B_2) will be identified as the optimal dimension path when $D = D_4$.

However, if there is no corresponding dimension or corresponding dimension chain in the candidate model that corresponds to D, we have to modify the parametric way of the candidate model manually to support automatic shape adaptation at present.



(a) Query model M_1 and its partial *3DDCG fdg_m*$_1$.

(b) Candidate model M_2 and its partial *3DDCG fdg_m*$_2$.

Fig. 13. An example of different dimension paths for the same pair of nodes.

5.4. Establishment of dimension relationship according to correspondence

After identifying the corresponding dimension B or corresponding dimension chain BC in the candidate model for a given dimension D in the query model, the establishment of dimension relationship between them is given as follows.

5.4.1. Establishment of dimension relationship for corresponding dimensions

If dimension B has no parametric constraint with any other dimension in the candidate model, then we establish the relationship between them as $B = D$; otherwise, use the method of imposing constraints sequentially [48] to recalculate all the dimensions that have parametric constraints with B while ensuring $B = D$. For example, in Fig. 12(c) and (d), two corresponding dimensions D_6 and B_5 representing with dark brown graph edges have a relationship as $B_5 = D_6$.

5.4.2. Establishment of dimension relationship for corresponding dimension chains

Let $BC = (B_1, B_2, \ldots)$ be the dimension chain that constrains the relative distance or the relative orientation between two constrained entities N_i and N_j in the candidate model. Furthermore, BC corresponds to D which is a dimension belonging to the query model. Then, according to the parametric constraints in the candidate model, two different methods are adopted to establish dimension relationships between D and each dimension in BC:

(1) If D is a distance dimension,
  (1.1) If each dimension in BC has no parametric constraint with other dimensions in the candidate model, then $B_i = (B_i/m\_d) * D$ where m_d is the sum value of all dimensions in BC. For example, in Fig. 12(c) and (d), (B_4, B_3, B_2) is the corresponding dimension chain of dimension D_4 representing with blue graph edges. Meanwhile, B_4, B_3 and B_2 have no parametric constraint with other dimensions in the candidate model. Thus, $B_4 = (B_4/m\_d) * D_4$, $B_3 = (B_4/m\_d) * D_4$ and $B_2 = (B_2/m\_d) * D_4$.
  (1.2) Else, we take a global solution recalculation by adding the constraint (the distance between N_i and N_j is equal to the value of D) to the global parametric constraints equations of the candidate model.
(2) Else if D is an angle dimension, then we take a global solution recalculation by adding the constraint (the angle between N_i and N_j is equal to the value of D) to the global parametric constraints equations of the candidate model.

After establishing the relationships among the dimensions of two models, the parametric information transfers from the query model to the candidate model. Then, the candidate model updates its shape by updating each dimension with the new incoming value, and the process of automatic shape adaptation is finished. For example, in Fig. 12(e), the shape of candidate model B in Fig. 12(e) is the result of adapting its shape in Fig. 12(b) to the query model in Fig. 12(a).

6. Implementation and results

The proposed automatic shape adaptation approach has been implemented in OSAD prototype. The UI module (Fig. 14) is developed by using Microsoft Visual C# 2008 and built as a plug-in of SolidWorks 2009 to interact with designers [88]; the matrix calculations (Matlab C++ Math Library [89]) and shape frame similarity assessment module are developed by using Microsoft Visual C++ 2008 and built as a win32 library which is invoked by the UI module during the process of the most similar shape frame identification in the candidate model.

6.1. Input models of OSAD

Our approach is relatively independent on 3D model retrieval approaches. For identifying dimension chains conveniently and feasibly in this work, we choose to build an assistant tool to take the following work: transforming each distance dimension to be defined from between two paralleled lines, to between one line and one point, to between two points in a descending priority. This is designed according to the common automatic dimensioning mechanism adopted by some mainstream CAD systems (such as SolidWorks).

6.2. Typical examples

Here, we provide seven additional pairs of models to demonstrate the efficiency and the main process of our automatic shape adaptation.

In each case of 1 & 2 & 3, in Fig. 15, both of the query model and the candidate model have simple shapes. When the corresponding faces between the two models are determined, the colored faces (except the gray ones) in the candidate model comprise a roughly overall shape of the model itself. After automatically transferring the shape of the query model to the candidate model, the changes (in the candidate model) occur in the corresponding faces and their adjacent faces.

In case 4 & 5 of Fig. 15, each query model contains more detailed shape design requirements than the query model in case 1, and

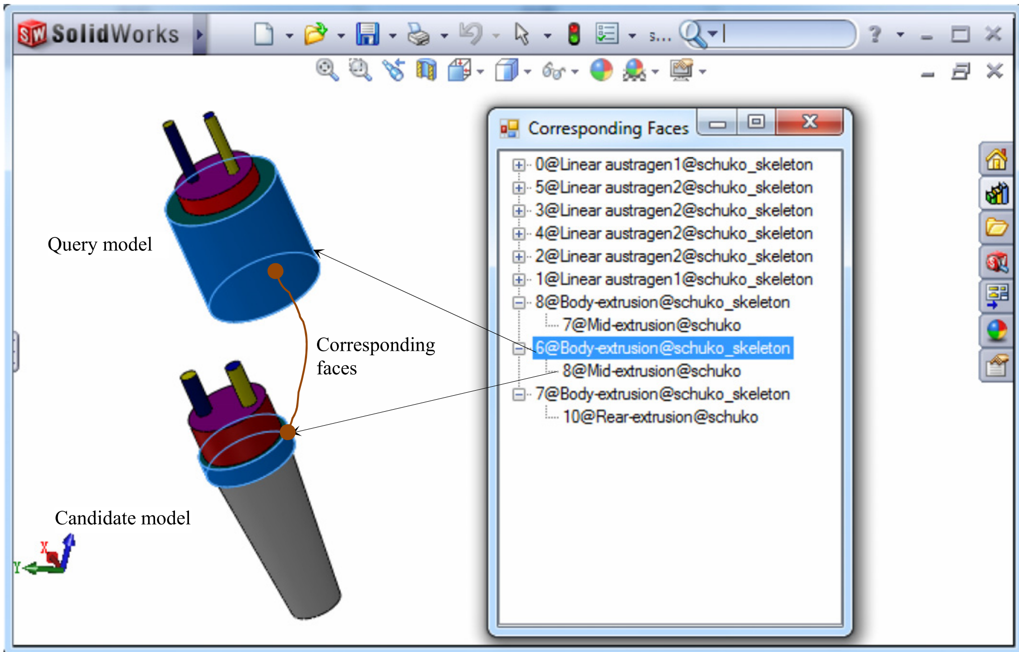


Fig. 14. User interface of the OSAD system.

has a close shape complexity to its corresponding candidate model. When the corresponding faces between each query model and its corresponding candidate model are determined, the colored faces (except the gray ones) in each candidate model comprise a compactly overall shape of the model itself. After automatically shape transferring, the shape of each candidate model is updated according to its corresponding query model.

Although this shape adaption approach is developed mainly for mechanical models, our approach can also be applied to parametric solid models of other fields (such as furniture models, and building models) when the models are designed conforming to our assumptions described in Section 7. In case 6 & 7 of Fig. 16, we illustrate automatic shape adaptation for the man-made models (each of them should be designed as a parametric part): one lantern and one chair.

Using our OSAD prototype, the efficiencies of the above 7 cases of automatic shape adaptation, shown in Table 4, are run on a laptop computer with an Intel Core 2 Duo CPU 2.10 GHz, 2 GB RAM and Windows 7 Operating System.

7. Discussion

Compared with the state-of-the-art approaches on geometric adaptation, our approach has several advantages and is more suitable for engineering applications. Here, we conduct the comparisons (Table 5) between the state-of-the-art approaches on geometric adaptation and our approach according to the following items: (1) what is the **input(s)**; (2) whether the target shape can be **driven by overall size** (such as global height, and global width); (3) whether the shape adaptation process needs **using domain-specific knowledge**; (4) whether the shape adaptation is a **preserving constraints** [75] adaptation; (5) what about the **adaptation means**; (6) what about the **application field(s)** and (7) whether the result shape needs **achieving engineering accuracy**.

7.1. Limitations

Although our approach can effectively adapt the shape of the candidate model to that of the input primary model (query model), there still exists some assumptions described as follows.

(1) The two input models are composed of plane faces and/or quadratic faces. The free-form surfaces currently have not yet considered since they are usually not related to dimension constraints directly.
(2) The input parametric solid models are constructed mainly by extrusion features and revolution features since they are the most common and fundamental features in current CAD systems.
(3) The corresponding faces have the same surface type and the same directed surface judgment, guaranteeing the correspondence between two corresponding faces is accurate.
(4) The shape frames of two input models have similar face layouts. Under this assumption, the corresponding faces can be effectively determined. In Fig. 17, there is no shape frame in the candidate model that has the similar face layout to that of the query model since the faces (contained in the red area) respectively belonging to the two models have different relative orientations based on the function (4.3). Thus, the corresponding faces determination between the two models is failed.
(5) Two corresponding pairs of constrained entities respectively belonging to the two given models have corresponding dimensions or corresponding dimension chains. This assumption is used to ensure that dimension-driven adaptation can be achieved [48,90,91].

Fig. 18 shows a failed example of establishing all dimension relationships between the two given models. Because the horizontal positions of the axes of face F_2 and face f_2 are respectively constrained by distance dimension D_2 and the geometric constraint (point–line coincident: sketch circle center *cp* and the sketch symmetry red dash line), there is no corresponding dimension or corresponding dimension chain that corresponds to D_2 in the candidate model (face F_2 corresponds to face f_2 and face F_1 corresponds to face f_1). Thus, after shape transferring, the axes of the corresponding faces F_2 and f_2 have different positions respectively relative to face F_1 and face f_1 while all of the other corresponding faces respectively belonging to the two given models have the same face layouts.

In addition, our approach also has the following limitation: The establishment of corresponding shape frames is inefficient

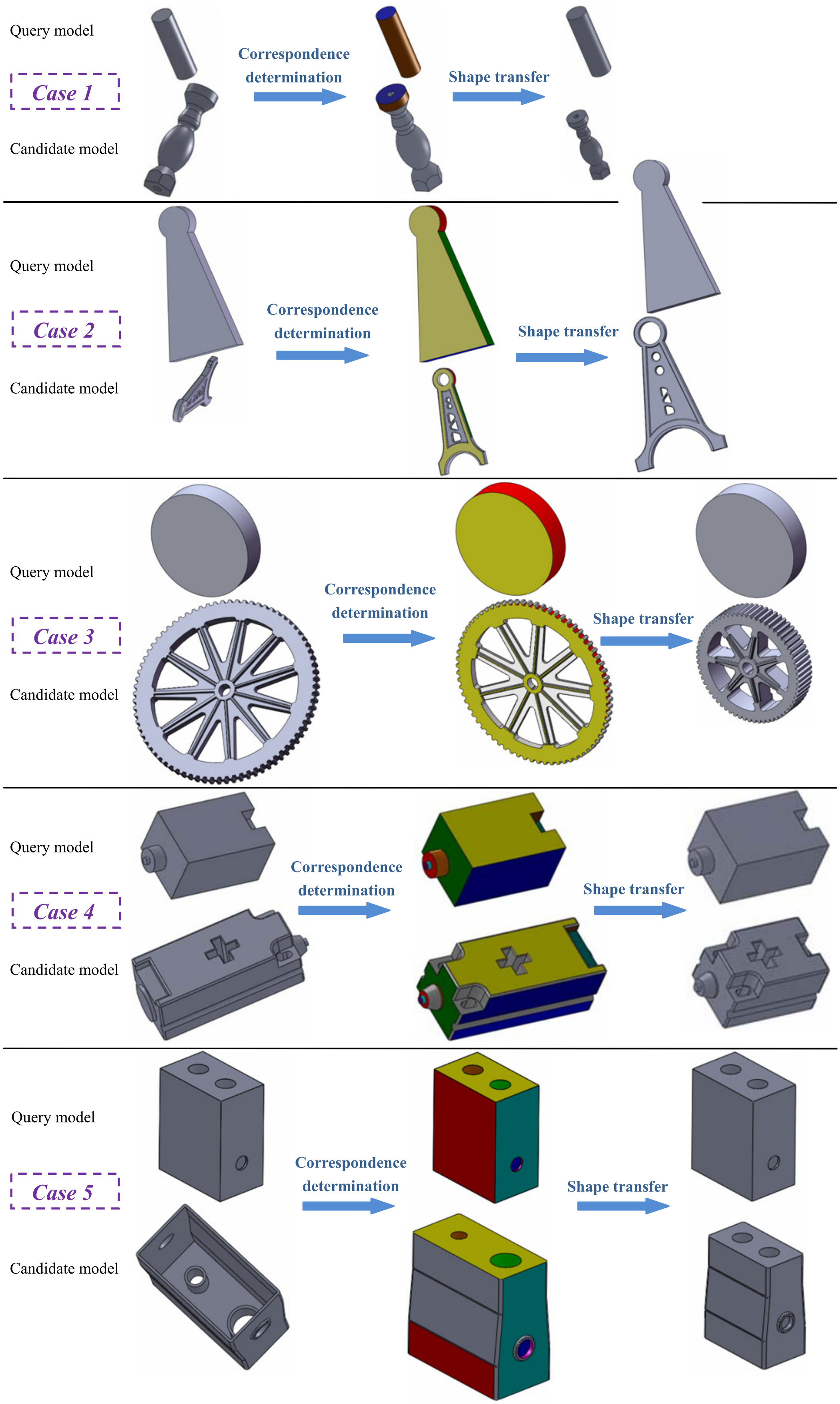


Fig. 15. Illustration of automatic shape adaptation for mechanical models. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

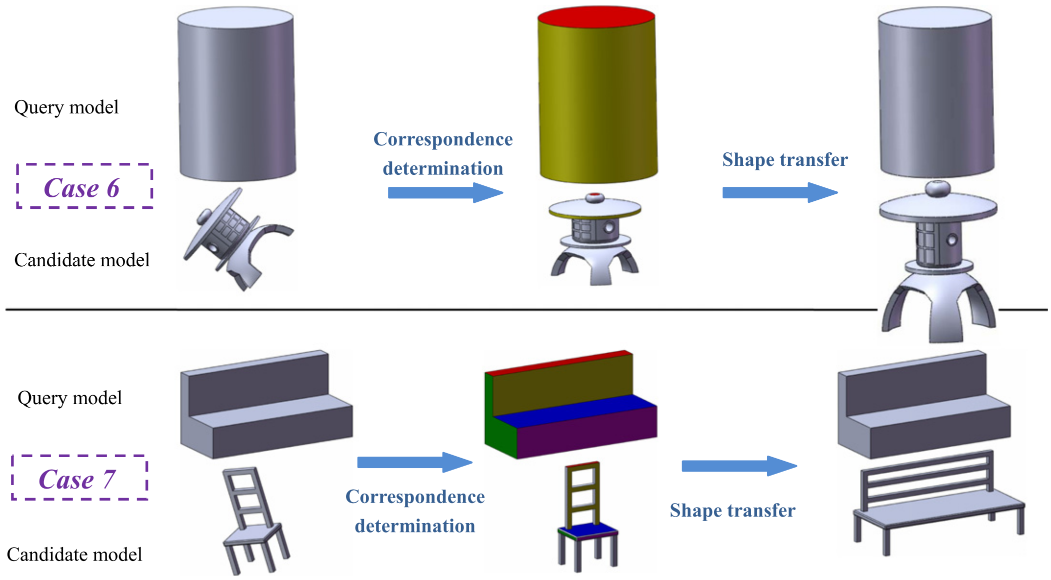


Fig. 16. Automatic shape adaptation for the man-made models.

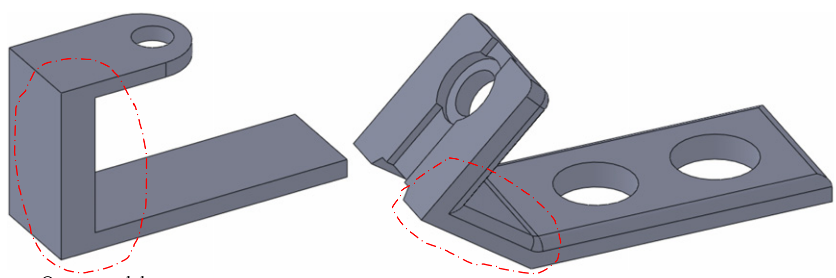


Fig. 17. Two models with similar overall shape while having no corresponding shape frames. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 4
The efficiencies of the automatic shape adaptation for the examples.

Time (s)	Case						
	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7
Correspondence determination	4.76	7.89	349.12	890.76	780.89	4.89	5.02
Shape transfer	0.23	1.03	1.45	2.12	1.98	0.22	0.56

Table 5
Comparisons between the state-of-the-art approaches on geometric adaptation and our approach.

Reference	Sumner et al. [35]	Lee et al. [74]	Kraevoy et al. [76]	Gal et al. [77]	Wang et al. [47]	Our approach
Input(s)	Mesh model	Mesh model	Mesh model	Mesh model	Parametric model with physical properties	Parametric model
Driven by overall size	No	No	Yes	Yes	No	Yes
Using domain-specific knowledge	No	No	No	No	Yes	No
Preserving constraints	No	No	No	Yes	Yes	Yes
Adaptation means	Deformation transferring	Pose-independent fitting	Non-homogeneous resizing	Structure-aware deforming	Analysis satisfying adjustment	Overall shape transferring
Application field(s)	Visualization; Prototyping design for art	Visualization; Prototyping design for garment; Online clothing stores	Visualization; Prototyping design for man-made objects	Visualization; Prototyping design for man-made objects	Shape optimization	Parametric design, such as mechanical design
Achieving engineering accuracy	No	No	No	No	Yes	Yes

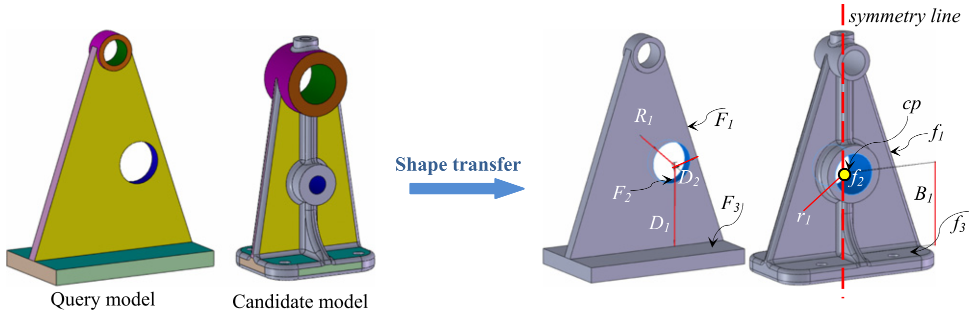


Fig. 18. Partial shape transfer. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

currently (such as the examples shown in Table 4) while this approach focuses on the geometric adaptation for mechanical engineering application and the face correspondence should be precise.

8. Conclusion and future works

Although the automatic shape adaptation is essential and important for Case-Based Design, presently, there is no effective solution that achieves automatic shape adaptation. In this paper, according to the requirements of case-based geometric design, we have proposed an approach to achieve automatic shape adaptation for parametric solid models. Our approach has the following contributions and characteristics:

(1) The proposed approach can effectively achieve the automatic and high quality shape adaptation of a candidate parametric solid model to its corresponding query model based on the corresponding faces and corresponding dimensions between the two models.
(2) A new method for determining the corresponding faces between two shape-similar solid models is put forward. By adopting the concept of shape frame and its quantitative descriptor, the method can automatically and accurately determine the corresponding faces between the two solid models in a global way without needing to pre-register (or pre-align) the two models.
(3) An algorithm for identifying the dimension correspondence between two parametric solid models is set forward. Based on the dimension promotion method, 3D dimension constraint graph and corresponding faces, the algorithm can effectively identify the corresponding dimensions and corresponding dimension chains between two parametric solid models, which enable the automatic and accurate dimension driven shape adaptation.

Considering that the automatic shape adaptation is a well-recognized challenging problem, we choose to solve the problem step by step. There are several works that could be conducted to make our approach more general in the future according to the limitations described in Section 7. For example, (a) to further study heuristic filtering methods to make the process of determining corresponding faces and corresponding shape frames more efficiently, (b) to remove the assumption that each dimension in the query model should have a corresponding dimension (or a corresponding dimension chain) in the candidate model since it is too strong to satisfy for some cases, and so on. Although our approach mainly focuses on the parametric solid models that are deemed as rigid, it will be a very interesting work to extend our research to large non-rigid deformable models.

Acknowledgment

The authors are very grateful to the financial support from NSF of China (61173125 and 61472111).

References

[1] Akin O. Psychology of architectural design. London: Pion; 1986.
[2] Watson I, Perera S. Case-based design: a review and analysis of building design applications. Artif Intell Eng Des Anal Manuf 1997;11:59–87.
[3] Chakrabarti A, Shea K, Stone R, Cagan J, Campbell M, Hernandez NV, et al. Computer-based design synthesis research: an overview. J Comput Inf Sci Eng 2011;11: 021003-1–021003-10.
[4] Goel AK, Vattam S, Wiltgen B, Helms M. Cognitive, collaborative, conceptual and creative—four characteristics of the next generation of knowledge-based CAD systems: A study in biologically inspired design. Comput-Aided Des 2012; 44:879–900.
[5] Chandrasekaran B. Design problem solving: a task analysis. In: AI Magazine. 1990. p. 59–71.
[6] Hua K, Fairings B, Smith I. CADRE: case-based geometric design. Artif Intell Eng 1996;10:171–83.
[7] El-Mehalawi M, Millerb RA. A database system of mechanical components based on geometric and topological similarity. Part I: representation. Comput-Aided Des 2003;35:83–94.
[8] Sundar H, Silver D, Gagvani N, Dickinson S. Skeleton based shape matching and retrieval. In: 2003 international conference on shape modeling and applications. Seoul (Korea): IEEE Computer Society; 2003. p. 130–9.
[9] Gao W, Gao SM, Liu YS, Bai J, Hu BK. Multiresolutional similarity assessment and retrieval of solid models based on DBMS. Comput-Aided Des 2006;38: 985–1001.
[10] Campbell RJ, Flynn PJ. A survey of free-form object representation and recognition techniques. Comput Vis Image Underst 2001;81:166–210.
[11] Chen X, Gao S, Guo S, Bai J. A flexible assembly retrieval approach for model reuse. Comput-Aided Des 2012;44:554–74.
[12] Chiou S-J, Sridhar K. Automated conceptual design of mechanisms. Mech Mach Theory 1999;34:467–95.
[13] Han Y-H, Lee K. A case-based framework for reuse of previous design concepts in conceptual synthesis of mechanisms. Comput Ind 2006;57:305–18.
[14] Bespalov D, Regli WC, Shokoufandeh A. Local feature extraction and matching partial objects. Comput-Aided Des 2006;38:1020–37.
[15] Hong T, Lee K, Kim S. Similarity comparison of mechanical parts to reuse existing designs. Comput-Aided Des 2006;38:973–84.
[16] Zhu K, Wong YS, Loh HT, Lu WF. 3D CAD model retrieval with perturbed Laplacian spectra. Comput Ind 2012;63:1–11.
[17] Iyer N, Jayanti S, Lou K, Kalyanaraman Y, Ramani K. Three-dimensional shape searching: state-of-the-art review and future trends. Comput-Aided Des 2005; 37:509–30.
[18] Hilaga M, Shinagawa Y, Kohmura T, Kunii TL. Topology matching for fully automatic similarity estimation of 3D shapes. In: SIGGRAPH 2001, Proceedings of the 28th annual conference on computer graphics. 2001, p. 203–12.
[19] El-Mehalawi M, Miller RA. A database system of mechanical components based on geometric and topological similarity. Part II: indexing, retrieval, matching and similarity assessment. Comput-Aided Des 2003;35:95–105.
[20] Osada R, Funkhouser T, Chazelle B, Dobkin D. Shape distributions. ACM Trans Graph 2002;21:807–32.
[21] Ma L, Huang Z, Wang Y. Automatic discovery of common design structures in CAD models. Comput Graph 2010;34:545–55.
[22] Tao S, Huang Z, Zuo B, Peng Y, Kang W. Partial retrieval of CAD models based on the gradient flows in Lie group. Pattern Recognit 2012;45:1721–38.
[23] Ma L, Huang Z, Wu Q. Extracting common design patterns from a set of solid models. Comput-Aided Des 2009;41:952–70.
[24] Gunn TG. The mechanization of design and manufacturing. Sci Am 1982;247: 114–30.
[25] Raphael B, Kumar B, McLeod IA. Representing design cases based on methods. In: Computing in civil engineering of American society of civil engineers. ASCE. New York, 1994, p. 285–92.
[26] Goel AK, Craw S. Design, innovation and case-based reasoning. Knowl Eng Rev 2005;30:271–6.
[27] Hanney K, Keane MT, Cunningham P, Smyth B. What kind of adaptation do CBR systems need? A review of current practice. In: Proceedings of the fall symposium on adaptation of knowledge for reuse. AAAI Press; 1995.
[28] Heylighen A, Neuckermans H. A case base of case-based design tools for architecture. Comput-Aided Des 2001;33:1111–22.

[29] Hunt J. Evolutionary case based design. In: Progress in case-based reasoning. Vol. 1020. 1995. p. 17–31.
[30] Perera S, Watson I. Nirmani: an integrated case-based system for strategic design and estimating. In: Progress in case-based reasoning. Vol. 1020. 1995. p. 185–200.
[31] Schmitt G. Case-based design and creativity. Autom Constr 1993;2:11–9.
[32] Renner G, Ekárt A. Genetic algorithms in computer aided design. Comput-Aided Des 2003;35:709–26.
[33] Rocca GL. Knowledge based engineering: between AI and CAD. Review of a language based technology to support engineering design. Adv Eng Inf 2012; 26:159–79.
[34] Chandrasegaran SK, Ramani K, Sriram RD, Horváth I, Bernard A, Harik RF, et al. The evolution, challenges, and future of knowledge representation in product design systems. Comput-Aided Des 2013;45:204–28.
[35] Sumner RW, Popović J. Deformation transfer for triangle meshes. ACM Trans Graph (TOG) 2004;23:399–405.
[36] Zhang H, Sheffer A, Cohen-Or D, Zhou Q, Kaick Ov, Tagliasacchi A. Deformation-driven shape correspondence. Comput Graph Forum 2008;27:1431–9.
[37] Botsch M, Sorkine O. On linear variational surface deformation methods. IEEE Trans Vis Comput Graphics 2008;14:213–30.
[38] Brouet R, Sheffer A, Inria LB, Cani M-P. Design preserving garment transfer. ACM Trans Graph (TOG) 2012;31:1–11.
[39] Kaick Ov, Zhang H, Hamarneh G, Cohen-Or D. A survey on shape correspondence. Comput Graph Forum 2011;30:1681–707 (extended version of Eurographics STAR).
[40] Bose A, Gini M. A case-based approach to planar linkage design. Artif Intell Eng 1997;11:107–19.
[41] Maher ML, Zhang DM. CADSYN: a case-based design process model. Artif Intell Eng Des Anal Manuf 1993;7:97–110.
[42] Perez RE, Behdinan K. Particle swarm approach for structural design optimization. Comput Struct 2007;85:1579–88.
[43] Richardson JN, Adriaenssens S, Bouillard P, Coelho RF. Multiobjective topology optimization of truss structures with kinematic stability repair. Struct Multidiscip Optim 2012;46:513–32.
[44] Yildiz AR. Comparison of evolutionary-based optimization algorithms for structural design optimization. Eng Appl Artif Intell 2013;26:327–33.
[45] Allaire G, Jouve F, Toader A-M. Structural optimization using sensitivity analysis and a level-set method. J Comput Phys 2004;194:363–93.
[46] Xia Q, Shi T, Liu S, Wang MY. A level set solution to the stress-based structural shape and topology optimization. Comput Struct 2012;90–91:55–64.
[47] Wang D, Zhang W. A general material perturbation method using fixed mesh for stress sensitivity analysis and structural shape optimization. Comput Struct 2013;129:40–53.
[48] Kumar AV, Yu L. Sequential constraint imposition for dimension-driven solid models. Comput-Aided Des 2001;33:475–86.
[49] Gossard DC, Zuffante RP, Sakurai H. Representing dimensions, tolerances, and features in MCAE systems. IEEE Comput Graph Appl 1988;8:51–9.
[50] Kolodner JL. Maintaining organization in a dynamic long-term memory. Cogn Sci 1983;7:243–80.
[51] Kolodner JL. Reconstructive memory: a computer model. Cogn Sci 1983;7: 281–328.
[52] Falkenhainer B, Forbus KD. The structure-mapping engine: algorithm and examples. Artificial Intelligence 1989;41:1–63.
[53] Gentner D. Structure-mapping: A theoretical framework for analogy. Cogn Sci 1983;7:155–70.
[54] Chen Y, Liu Z-L, Xie Y-B. A knowledge-based framework for creative conceptual design of multi-disciplinary systems. Comput-Aided Des 2012;44:146–53.
[55] Kurtoglu T, Campbell MI, Arnold CB, Stone RB, McAdams DA. A component taxonomy as a framework for computational design synthesis. J Comput Inf Sci Eng 2009;9: 011007-1–011007-10.
[56] Tangelder JWH, Veltkamp RC. A survey of content based 3D shape retrieval methods. Multimedia Tools Appl 2008;39:441–71.
[57] You C-F, Tsai Y-L. 3D solid model retrieval for engineering reuse based on local feature correspondence. Int J Adv Manuf Technol 2010;46:649–61.
[58] Ovsjanikov M, Li W, Guibas L, Mitra NJ. Exploration of continuous variability in collections of 3D shapes. ACM Trans Graph 2011;30:1–10.
[59] Sun R, Gao S, Zhao W. An approach to B-rep model simplification based on region suppression. Comput Graph 2010;34:556–64.
[60] Hu J, Hua J. Salient spectral geometric features for shape matching and retrieval. Vis Comput (CGI) 2009;25:667–75.
[61] Kaick Ov, Tagliasacchi A, Sidi O, Zhang H, Cohen-Or D, Wolf L, et al. Prior knowledge for part correspondence. Comput Graph Forum 2011;30:553–62 (Special Issue of Eurographics 2011).
[62] Rusinkiewicz S, Levoy M. Efficient variants of the ICP algorithm. In: Third international conference on 3-d digital imaging and modeling. 2001, p. 145–52.
[63] Ovsjanikov M, Ben-Chen M, Solomon J, Butscher A, Guibas L. Functional maps: a flexible representation of maps between shapes. ACM Trans Graph (TOG) 2012;31:1–11.
[64] Castellani U, Cristani M, Fantoni S, Murino V. Sparse points matching by combining 3D mesh saliency with statistical descriptors. Comput Graph Forum 2008;27:643–52 (Proc. EUROGRAPHICS).
[65] Chow CK, Tsui HT, Lee T. Surface registration using a dynamic genetic algorithm. Pattern Recognit 2004;37:105–17.
[66] Gelfand N, Mitra NJ, Guibas LJ, Pottmann H. Robust global registration. In: M. Desbrun, H. Pottmann (Eds.), SGP'05 Proceedings of the third eurographics symposium on geometry processing 2005, p. 197–206.
[67] Goyal M, Murugappan S, Piya C, Benjamin W, Fang Y, Liu M, et al. Towards locally and globally shape-aware reverse 3D modeling. Comput-Aided Des 2012;44:537–53.
[68] Kolodner JL. Case based reasoning. Morgan Kaufmann; 1993.
[69] Pearce M, Goel AK, Kolodner IL, Zimring C, Sentosa L, Billington R. Case-based design support: A case study in architectural design. IEEE Expert 1992;7: 14–20.
[70] Autodesk WikiHelp, in, The Autodesk Inc., 2012. http://wikihelp.autodesk.com/Inventor/enu.
[71] Vong C-m, Wong P-k. Case-based adaptation for automotive engine electronic control unit calibration. Expert Syst Appl 2010;37:3184–94.
[72] Sokolowski J, Zolesio J-P. Introduction to shape optimization. Springer series in computational mathematics, Berlin (Heidelberg): Springer; 1992. p. 5–12.
[73] Meng Y, Wang CCL, Jin X. Flexible shape control for automatic resizing of apparel products. Comput-Aided Des 2012;44:68–76.
[74] Lee Y, Ma J, Choi S. Automatic pose-independent 3D garment fitting. Comput Graph 2013;37:911–22.
[75] Mitra NJ, Wand M, Zhang H, Cohen-Or D, Bokeloh M. Structure-aware shape processing, Eurographics state-of-the-art report (STAR). 2013. p. 175–97.
[76] Kraevoy V, Sheffer A, Shamir A, Cohen-Or D. Non-homogeneous resizing of complex models. ACM Trans Graph (TOG) 2008;27:1–9.
[77] Gal R, Sorkine O, Mitra NJ, Cohen-Or D. iWIRES: an analyze-and-edit approach to shape manipulation. ACM Trans Graph (TOG) 2009;31:1–10.
[78] Zheng Y, Fu H, Cohen-Or D, Au OK-C, Tai C-L. Component-wise controllers for structure-preserving shape manipulation. Comput Graph Forum 2011;30: 569–72.
[79] Bokeloh M, Wand M, Seidel H-P, Koltun V. An algebraic model for parameterized shape editing. ACM Trans Graph (TOG) 2012;31:1–10.
[80] Mitra NJ, Pauly M, Wand M, Ceylan D. Symmetry in 3D geometry: extraction and applications. Comput Graph Forum 2013;32:1–23.
[81] Ip CY, Lapadat D, Sieger L, Regli WC. Using shape distributions to compare solid models. In: Proceedings of the seventh ACM symposium on solid modeling and applications. 2002, p. 273–80.
[82] Schneider PJ, Eberly DH. Geometric tools for computer graphics. 1st ed. Morgan Kaufmann; 2002.
[83] Chen X, Hoffmann CM. Towards feature attachment. Comput-Aided Des 1995; 27:695–702.
[84] Wu J, Zhang T, Zhang X, Zhou J. A face based mechanism for naming, recording and retrieving topological entities. Comput-Aided Des 2001;33:687–98.
[85] Even S. Graph algorithms. 2nd ed. Cambridge University Press; 2011.
[86] Bouma W, Fudos I, Hoffmann C, Cai J, Paige R. Geometric constraint solver. Comput-Aided Des 1995;27:487–501.
[87] Yang RD, Fan X, Wu D, Yan J. Virtual assembly technologies based on constraint and DOF analysis. Robot Comput-Integr Manuf 2007;23:447–56.
[88] SolidWorks Web Help, in, The Dassault Systèmes SolidWorks Corporation, 2012. http://help.solidworks.com/HelpProducts.aspx.
[89] MATLAB C++ Math Library reference 2.1, in, The MathWorks Inc. http://www.mathworks.com/index.html.
[90] Lin VC, Gossard DC, Light RA. Variational geometry in computer-aided design. ACM SIGGRAPH Comput Graph 1981;15:171–7.
[91] Light R, Gossard D. Modification of geometric models through variational geometry. Comput-Aided Des 1982;14:209–14.