Identifying Different Entities for Minor Model Modification Based on Common Primary Subpart

Wanbin Pan, Shuming Gao, Hua Zhu, Ming Li and Jinming Chen

Zhejiang University, pwanbin@163.com, {smgao, zhuhua}@cad.zju.edu.cn

ABSTRACT

To identify different entities between two solid models with minor difference is a key issue in many research fields. Most of the existing approaches on this field are either not general enough for handling general CAD models, or are of low efficiency by using global comparison. In this paper, a novel different entity identification approach is proposed based on a novel concept of common primary subparts and using the associated attributed graphs with the two models. Identification efficiency is achieved by first classifying graph nodes (model faces) into the groups based on their underlying surfaces and face senses. By detecting common attributed subgraphs from the attributed graphs of the two models, all typical different entities can then be locally identified in the process of eliminating common attributed subgraphs. The left subgraphs are then further checked for finding other unusual different entities. Preliminary experiment results show that the proposed approach can not only find out all different entities, but also achieve high computational efficiency as compared with previous work.

Keywords: primary subpart, local comparison, minor difference.
DOI: 10.3722/cadaps.2011.345-356

1 INTRODUCTION

It is estimated that more than 75% of the design activities are completed by totally reusing or modifying previous design data [1]. Reusing the huge number of existing digital 3D CAD models is becoming an important way to facilitate and save time in the cycle of new design. In this circumstance, the ability to find out the difference between two given models is very useful and also a key issue pre-required in many research fields, such as similarity assessment, local processing, local remeshing, and collaborative designing.

In many research fields (e.g. local remeshing), the exact difference between two given models needs capturing particularly. The key issue in resolving this problem is the absence of prior knowledge on the location of these different entities and how big the difference zone is. Thus identifying the exact difference between two models solely based on local comparison is not easy to achieve, and the popular approaches applied in previous studies is mainly based on global comparison and of low computational efficiency.

Many research efforts have been done to identify exact difference between two given models with minor difference. However, most of these approaches are hard to handle general B-rep models or is computationally expensive. The goal of this work is to develop an approach to handle general solid models in the sense that it works directly on B-rep models and of much improved efficiency by using maximal possible local comparisons. Specifically, the approach proposed here contains the following main steps:

(1) Specify for a given model the general attributed graph form to describe every face's shape and the face's adjacency.

(2) Build the relationship between subparts of a model and their attributed subgraphs of the attributed graph of the model.

(3) Apply a novel subgraphs elimination approach to confine the exact different entities identification between the two given models to special local areas.

The remainder of the paper is organized as follows. After briefly introducing related work in Section 2, basic concepts and overview of the approach are described in Section 3. In Section 4, we build the relationships between attributed subgraphs and subparts and propose the main approach to identify different entities. The different entities identification based on subgraph elimination is presented in Section 5. In Section 6, for complementing, different entities are further detected in the left attributed graphs. Experiment results are described in Section 7. Finally, we give conclusions and describe our future work in Section 8.

2 RELATED WORK

Difference identification between two models can be categorized into two main classes: (1) Rough difference identification; (2) Exact difference identification.

2.1 Rough Difference Identification

Rough difference identification attracts much attention [8-16]. All of these works aim at getting a rough difference between two given models. The common methods of these studies are to use descriptors to represent models, and difference identification is then performed on these descriptors instead of the original whole models. There are two main descriptors applied in previous work.

Geometry-based methods: Build the discretization data of a model, and form a special descriptor, such as D2 [8]. The advantage of these methods is that they are general to handle all types of models (mesh models, solid models and so on).

Topology-based methods: A descriptor is built mainly based on the topology structure of a model (solid model as usual), such as attributed graph [13, 14]. However, such descriptors usually only maintain key features of a model while neglecting other small local features, and thus do not give all the exact different entities. This is the fundamentally different from the issue studied in this paper.

2.2 Exact Difference Identification

Getting the exact difference by comparing two models are very important in some research fields, e.g. local remeshing. There are two main classes approaches related to this work.

Space Decomposition: By decomposing the domain occupied by the model into small cells recursively [2, 3], these approaches compare leave cells of the two same space decomposition structures of the two given models, and thus avoid direct global comparison between two given models. Because the decomposition process itself is very time consuming especially for complex models, computational efficiency is generally very low.

Cellular Model: Some researches identify local modifications based on cellular model [5, 6], where the cellular model keeps the track of each feature modification in the design iteration process. However, applying these approaches directly on B-rep models is not trivial, as feature information should be recognized in advance, which however is not an easy task [16].

3 BASIC CONCEPTS AND APPROACH OVERVIEW

In this study, the two input B-rep models are *original model* and *modified model* that comes from a design process of a part. The same design process keeps the two models registered. So, in this paper, we do not take care of the register problem, and we aim at finding all the different entities (different faces, different edges, and different vertices) between the original model and the modified model.

3.1 Basic Concepts

Different Face: A different face is the face, in one model, which has no mapping face in the other model sharing the same underlying surface, having the same boundary, and of the same face sense [16, 17]. Different faces have three types: First, a ***Deleted Face*** that exists in original model but has no corresponding face in the modified model sharing the same underlying surface and having the same face sense. For example, B is a deleted face in Fig.1. Second, a ***New Face*** that exists in modified model, which has no corresponding face in original model sharing the same underlying surface and having the same face sense. For example, 2 is a New Face in Fig.1. Third, a ***Modified Face*** is a face of one model that has a face(s) in the other model sharing the same underlying surface and the same face sense, but not the same boundary. For example, A, D, 1 are modified face in Fig.1.

Different Edge: A different edge is the edge, in one model, which has no mapping edge in the other model sharing the same curve and has same two end positions, such as eAB and e12 in Fig.1.

Different Vertex: A different vertex is the vertex, in one model, which has no vertex in the other model holding the same position.

3.2 Approach Overview

A systematic overview of our approach is illustrated in Fig. 2, and it mainly consists of the following four steps.

(1) Build the corresponding attributed graphs for the original and modified B-rep models.

(2) Build specific group nodes (*VGN*s in this work) to collect together all common nodes from the above two built attributed graphs based on their associated attributes, which includes underlying surface and face sense, and mark the new faces and deleted faces from them. This step gives part of different entities that we are to identity.

(3) Detect, eliminate common subgraphs (*CPASG*s in this work) from the two original and modified attributed graphs. This step gives different entities around the faces, whose corresponding nodes are contained in VGNs linked by *connecting line*s.

(4) Further identify different entities from the left subgraphs of the two attributed graphs resulted from step (3), which together with the detection results of steps (2) and (3), give all the different entities that we aim to identity.

4 ATTRIBUTED GRAPHS FOR SOLID MODELS

A solid model is represented using an attributed graph with the nodes of the graph representing faces and the links of the graph representing edges. Geometric information is set as attributes associated with graph nodes and links. The attributed graph converts a model's topology and geometry into a compact form, facilitating model comparison. Although general graph comparison is not an easy task, comparing attributed graphs is much easier than directly comparing whole B-rep structures [16].

4.1 Construction of Attributed Graphs

In this work, the attributed graph of a model is associated with a unique graph G = (N, L), where N is the node set of attributed graph, and L is the link set of attributed graph, based on the following rules:

- Every face *f* in the model corresponds to a unique node *n* in N. The face name, its underlying surface and normal are assigned as attributes of *n*. Face sense describes the relationship between a face and its surface orientations [17].
- Every edge *e* in the model corresponds to a unique link *l* in L. The edge *e*'s name, its underlying curve, and its two endpoint positions are assigned as the attributes of *l*. A link <*ni*,

nj> represents that edge *e* connects nodes *ni* and *nj*, which also means that faces *fi* and *fj* share the same edge *e*.

Because each face f corresponds to a unique node n, so we use the same name for each face and its corresponding node (e.g. face E and node E in Fig. 1 (a)). And each edge e also corresponds to a unique link l, we use the same name for each edge and its corresponding link (e.g. edge eEF and link eEF in Fig. 1 (a)). The process of building attributed graphs is same as described in previous work [14]. Here we further assign specific attributes to meet the need of this work. Fig.1 (a) shows a simple model and its associated attributed graph. In this diagram, A-F are model faces (nodes), and eAB-eEF are model edges (links).

4.2 Common Primary Subpart

After specifying the attributes for all nodes and links in the attributed graphs of the original model and the modified model, we, here, define three terms for this paper to achieve common local shape identification by attributed subgraph comparison.

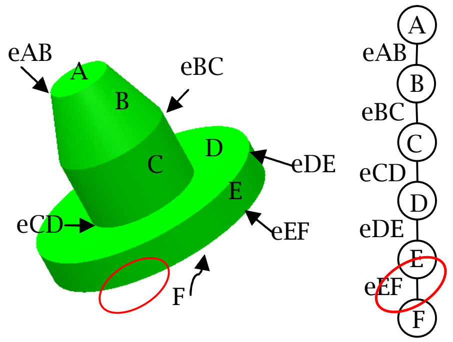
Primary Subpart: A subpart of a solid model is composed of two faces and one of their common edges is called a primary subpart. Since a face can have more than one edges, a face can belong to more than one primary subparts. For example, in Fig.1, face E, face F and edge eEF build a primary subpart, and face E, face D and edge eED build another primary subpart.

Primary Attributed Subgraph (PASG): Representing a solid model as an attributed graph, every primary subpart of the model is mapped to a unique PASG of the attributed graph. So every PASG is descripted as (*n1*,*l*,*n2*), two nodes with a connecting link *l*. For example, in Fig.1, node E, node F and link eEF build a PASG (E,eEF,F), and node E, node D and link eED build another PASG (E,eED,D).

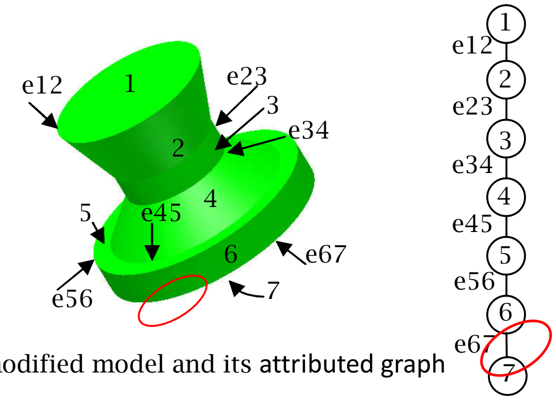
Common Primary Attributed Subgraph (CPASG): For two PASGs: psg1 = (*ni*,*l1*,*nj*) and psg2 = (*np*,*l2*,*nq*), where psg1 and psg2 belong to different attributed graphs, *l1* connects *ni* and *nj*, and *l2* connects *np* and *nq*. We say psg1 and psg2 are common if and only if *l1* and *l2* are common, (*ni*,*np*), (*nj*,*nq*) are respectively common, and the CPASG ((*ni*,*l1*,*nj*),(*np*,*l2*,*nq*) or (*ni*,*nq*), (*nj*,*np*), the CPASG ((*ni*,*l1*,*nj*),(*nq*,*l2*,*np*)) are both common. Two nodes (or links) are common if their corresponding attributes are all the same. For example, in Fig.1, the psg1 = (E,eEF,F) is common to psg2 = (6,e67,7). They together build a CPASG ((E,eEF,F), (6,e67,7)) in Fig. 2 (a).

The two primary subparts corresponding to a same CPASG are called common primary subparts. If two primary subparts in two models are common, the edges in these two primary subparts are common, and the face in one of these two primary subparts has the same boundary along the common edge as the face sharing same underlying surface and face sense in the other primary subpart.

Therefore, if all of the primary subparts in one model are common to those of the other model, they are definitely the same (no different entities exist) assuming that they are in the same modification process. Moreover, two PASGs are common also indicates that have a mapping relationship between their primary subparts. Thus, different entities must exist in the set of primary subparts that do not have common primary subparts.



(a) original model and its attributed graph



(b) modified model and its attributed graph

Fig. 1: The attributed graphs for an original model (a) and its modification (b) by performing a chamfer operation on face ‘C’ and a draft operation on face ‘B’.

However, directly seeking common primary subparts between two solid models in B-rep structure is transformed into seeking CPASGs from their associated attributed graphs. However, directly finding CPASGs from the attributed graphs of the original and the modified models is still a problem of global matching, and is very time consuming. In order to reduce the complexity of the graph matching problem, we are trying to confine this problem into some specific local regions, using the idea of associated graph used in the previous work [3][16].

5 CPASG-BASED DIFFERENT ENTITIES IDENTIFICATION

After clarifying the CPASGs and common primary subparts relationships, we propose an approach to identify different entities by eliminating the CPASGs from two attributed graphs.

5.1 Identification of New Faces and Deleted Faces through Nodes Group Operation

Efficiency of the matching between two PASGs is achieved by first grouping common nodes whose corresponding faces share the same underlying surface and face sense. This is because we have to comparing the nodes and links based on the concept of CPASG before two PASGs are determined common to each other.

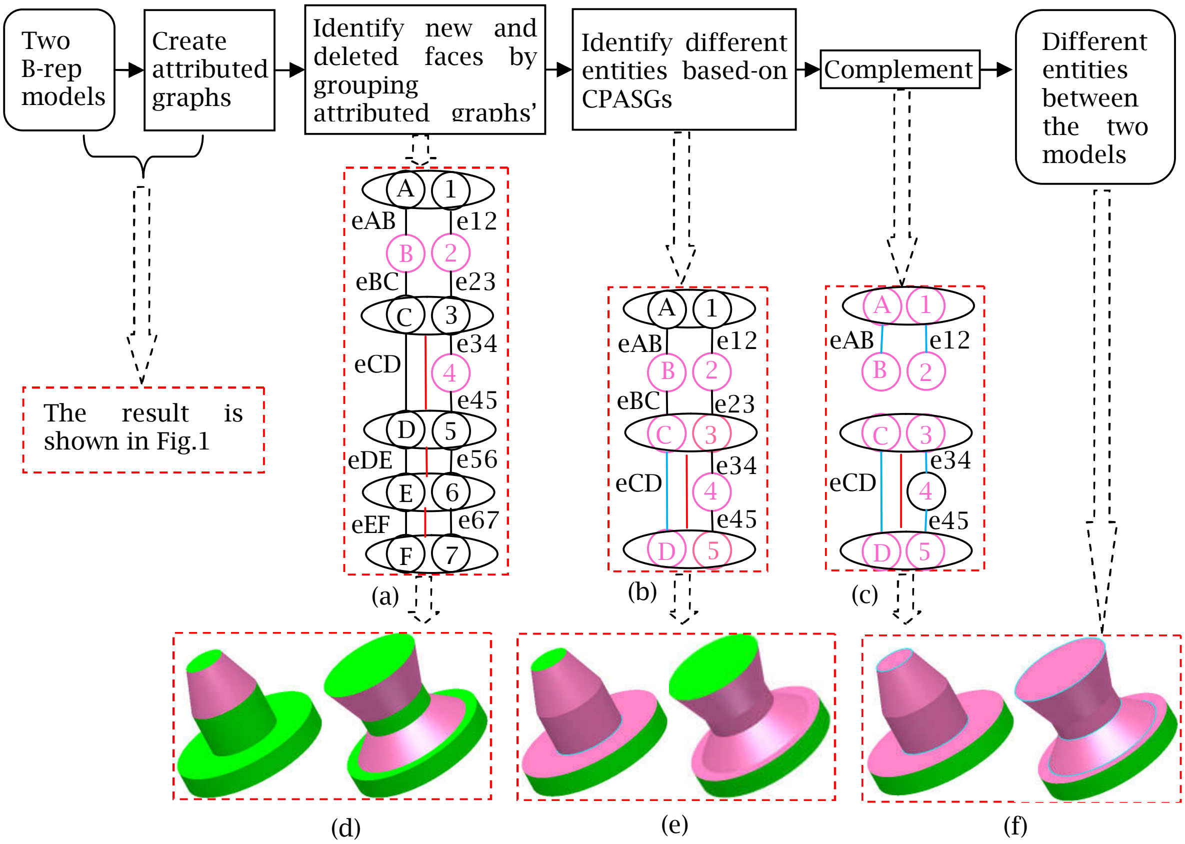


Fig. 2: Algorithm overview and the illustration with the example models in Fig.1. The resulted different faces (9 faces) are rendered in pink, and different edges (5 edges) are rendered in light blue.

5.1.1 Nodes Group Operation

Two basic concepts are used in this paper to group the two built attributed graphs in section 4.

Virtual Group Node (VGN): All the nodes, both in the original model's attributed graph and in the modified model's attributed graph, are grouped together as virtual group nodes based on their attributes. A virtual group node is marked as VGN (*n1*,*n2*,...,*nm*), *n1*,*n2*,...,*nm*, having the same attributes, are its member nodes, for example the virtual group nodes VGN (A,1) and VGN (C,3) in Fig. 2 (a).

Connecting Line: A connecting line, marked as <*VGN1*,*VGN2*> , is a line connecting two VGNs. such as the red lines in Fig.2

The realization of group strategy contains two steps:

(1) Build VGNs by collecting together common nodes, i.e. nodes sharing the same underlying surfaces and face senses.

(2) Connect every two VGNs by a *connecting line* if an attributed link exists between their members.

Fig. 2 (a) shows a simple example of the result of the group operation between the original model's attributed graph and the modified model's attributed graph, where the nodes in a common virtual group node shows that they have the same underlying surface and same face sense. The virtual group nodes (e.g. VGN (A,1), VGN (C,3), VGN (E,6) and VGN (F,7)) are constructed from the attributed graphs of the original model and the modified model. Based on the nodes connectivity in the two attributed graphs, VGN (E,6) and VGN (F,7) are connected by a connecting line <VGN (E,6),VGN (F,7)> (in red). Virtual group nodes VGN (A,1) are VGN (C,3) are not connected because nodes A and C are not connected in the attributed graph of the original model, and nodes 1 and 3 are neither connected in the attributed graph of the modified model. Other connecting lines in the Fig. 2 (a) are constructed in the same way. If a node belonging to one attributed graph has no common nodes in the other attributed graph, it will not be grouped and kept unchanged, such as the nodes B, 2 and 4. Concrete algorithm of the above process is also described in Fig. 3.

5.1.2 Identification of new Faces and Deleted Faces

After the group operation, the nodes that have not been collected into any virtual group node indicate that their corresponding faces are different. In Fig. 2 (a), the face B in the original model, or node B, has no corresponding face in the modified model sharing the same underlying surface and the same face sense, and is a deleted face. Similarly, face 2 in the modified model, or node 2, has no corresponding face in the original model sharing the same underlying surface and the same face sense, and is a new face. The different faces are marked pink in Fig. 2 (d).

```
Algorithm 1:  Void Group_Two_Attributed Graph_Nodes (AG_1, AG_2)
--------------------------------------------------------------------------------------------------------------
Classify the nodes in AG_1 and AG_2 by surface equation and face sense.
For each classification cf do
  If (all of the nodes in cf belong to AG_1) OR (all of the nodes in cf belong to AG_2) then
     // new faces or deleted faces
    Mark the faces corresponding to these nodes as different faces.
  Else
    Create a new virtual group node nVPN.
    Put all of the nodes in cf into nVPN.
  END If
End For
For each link <N, P> in AG_1 do
  If (N belongs to VGN1 AND P belongs to VGN2 ) then
     Create a connecting line n_l .
     Use n_l to connect VGN1 and VGN2, n_l = <VGN1, VGN2>.
  End If
End For
```

Fig. 3: Illustration of the algorithm of group operation.

5.2 Identification of Different Entities Based-on CPASGs

After having grouped the nodes, the problem of detecting CPASGs is simplified to identifying common links between two connected virtual group nodes. In this subsection we aim to detecting all of the common subparts between the original model and the modified model through the detection of CPASGs from their attributed graphs.

5.2.1 Detection of CPASGs

With the help of VGNs and connecting lines, we confine the CPASG detection into two connected VGNs and simplify the PASG matching problem to a link comparison problem that processing in each connecting link set.

Connecting Link Set: A connecting link set is a collection of links between two connected VGNs. For example, the connecting link set {eDE,e56}, in Fig. 2 (a), describes the links between VGN (D,5) and VGN (E,6).

Fig. 2 (a) shows a simple example of the result of the group operation between the original model's attributed graph and the modified model's attributed graph. The virtual group nodes VGN (E,6) and VGN (F,7) are connected by a connecting line <VGN (E,6),VGN (F,7)>. The connecting link set between these two VGNs is {eEF, e67}. Because the links eEF and e67 share a same curve and have same end positions, they are common. Following the concept of common PASGs in section 2, the PASGs (E,eEF,F) and (6,e67,7) are common. We then have a CPASG ((E,eEF,F),(6,e67,7)) in Fig. 2. Other CPASGs are detected in the same way. The concrete algorithm of the above process is also described in Fig.5.

5.2.2 Identification of Different Entities Through Elimination of CPASGs

The links are eliminated from their attributed graphs respectively when their corresponding CPASGs are detected, for example, in Fig. 2 (a), the links eEF and e67, belonging to the attributed graphs of the original model and the modified model respectively. When the CPASG ((E,eEF,F),(6,e67,7)) has been detected, we eliminate the link eEF from the attributed graph of the original model and the link e67 from the attributed graph of the modified model. The CPASG ((E,eEF,F),(6,e67,7)) is correspondingly eliminated from the two attributed graphs. Other links corresponding to CPASGs are eliminated in the same way.

After some links having been eliminated from the attributed graphs of the original model and the modified model, some nodes may become isolated, such as E, F, 6 and 7. This isolated status is checked in our approach using a function ***Isolated***. We simply delete these isolated nodes from their attributed graphs, because there have no PASGs containing these nodes. When all nodes in a VGN are eliminated, the VGN will also be deleted in our approach. Fig. 2 (b) shows the left attributed graphs after the elimination of CPASG.

After all links corresponding to the CPASGs are deleted from the connecting link set of two connected VGNs, the PASGs, holding the left links of the connecting link set, indicate different entities. For example, in Fig. 2 (b), the left link in connecting link set between VGN (C,3) and VGN (D,5) is {eCD}. Therefore the PASG (C,eCD,D), belonging the attributed graph of the original model, has no common PASG in the attributed graph of the modified model, and the entities corresponding to this PASG are marked as different entities, as shown in Fig. 2 (e).

According to the attributes associated with every link, the relationships, between the different entities and the left links of the connecting link set, are described in Tab. 1. For case (1), such as the link eAB in Fig. 4 (c), because eAB has no common link in the connecting link set {eAB,e12}, so we record all faces, edges and vertices associated with the edge eAB. For case (2), such as the link eCD in Fig. 4 (f), because eCD has the same curve as e34, while their end positions are not same, so we need further compare end positions of the two links to determine the different vertices. The whole process of identification of different entities through the elimination of CPASGs is described in Fig.5.

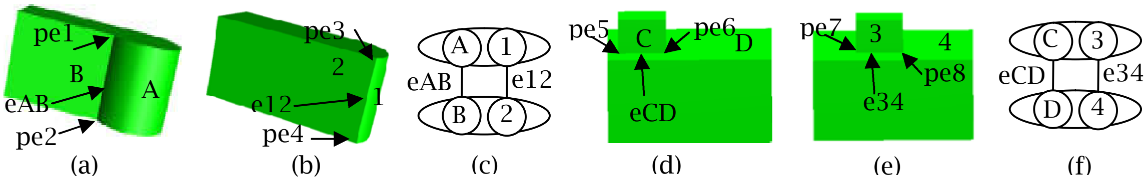


Fig. 4: Illustration of typical examples of the relationships between the different entities and the left links of the connecting link set. ((a) original model; (b) the modified model by cutting along face B of the model in (a); (d) original model; (e) the modified model by moving extruded feature along face D of the model in (d); (c) the PASG (A,eAB,B) and PASG (1,e12,2) of the attributed graphs of the models in (a) and (b) respectively after group operation; (f) the PASG (D,eCD,D) and PASG (3,e34,4) of the attributed graphs of the models in (d) and (e) respectively after group operation.)

Case	*Example*	*Different Entities*
1. Two same underlying surfaces have different intersection curves, such as (a) and (b) in Fig. 4.	In Fig. 4 (c), B has same attributes as 2, A has same attributes as 1, eAB's curve not same to e12.	Modified faces: A, B, 1, 2. Different edges: eAB, e12. Different vertices: pe1, pe2, pe3, pe4.
2. Two same underlying surfaces have same intersection curve but different end positions, such as (d) and (e) in Fig.4.	In Fig. 4 (f), C has same attributes as 3, D has same attributes as 4, eCD has same curve to e34, but pe5 is not same to pe7 and pe6 is not same to pe8.	Modified faces: C, D, 3, 4. Different edges: eCD, e34. Different vertices: need comparing.

Tab. 1: Lookup table of none matched links and their different entities.

For each connecting line <*VGN1*, *VGN2*> in connecting lines

Subdivide the connecting link set between *VGN1* and *VGN2* into two groups *G1* and *G2*, *G1* contains the links belonging to *AG1* and *G2* contains the links belonging to *AG2*.

G1.count >0 **AND** *G2*.count >0

Yes

No

For each link *l1*=<*A, B*> in *G1*

Has a link *l2* = <*C,D*> in *G2* with all attributes same to *l1*

Yes

No

Record CPASG: ((A,*l1*,B),(C,*l2*,D)).
Delete *l1* from *AG1* and *G1*.
Delete *l2* from *AG2* and *G2*.
Delete **Isolated** nodes among A, B, C and D from AG1 or AG2 respectively.

Record the different entities associating to *l1* according to Tab. 1.

For each link *l2*=<*C,D*> in *G2*

Record the different entities associating to *l2* according to Table 1.

Fig. 5: Illustration of recording different entities and eliminating CPASGs.

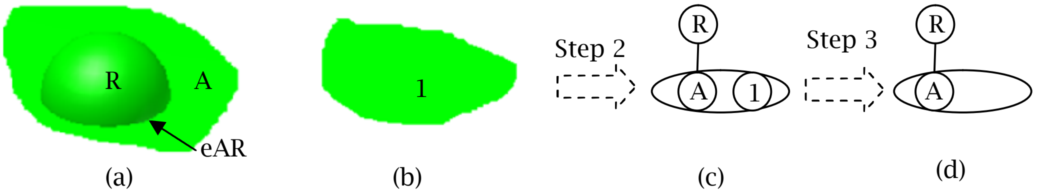


Fig. 6: Illustration of an example of additional modified face which is usually not identified.(Both (a) and (b) subparts of two models respectively; (b) the modified model by removing a half-sphere feature from the face A of the model in (a); (c) the result of grouping nodes around A and 1; (d) the result of eliminating CPASGs, where node 1 is deleted.)

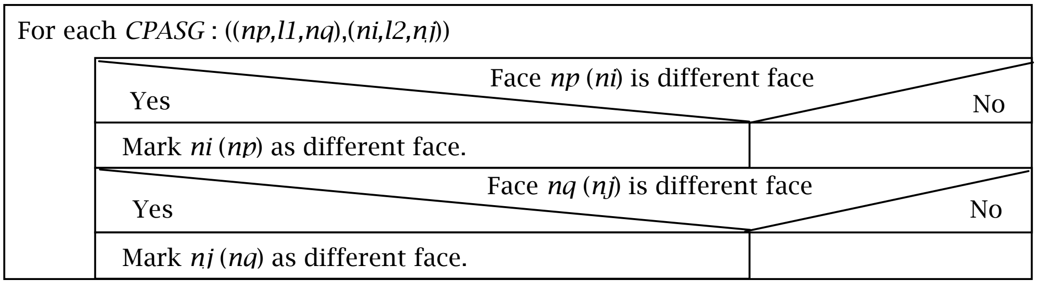


Fig. 7: Illustration of different faces identifying among CPASGs.

5.2.3 Identifying additional modified faces from CPASGs

Previous works usually are lack of the ability of identifying all modified faces [5] through only detecting the appeared and disappeared entities. For example in Fig.6 a feature R is removed from a face A, and the edge eAR and face A have been identified as different entities. But the modified face 1 (from face A) has not been identified. In our work, by checking the status of modification of each face that its corresponding node belongs to a CPASG, we handle the problem conveniently.

First, we observe a mapping relationship. If two faces f1 and f2, respectively from the original model and modified model, share the same underlying surface and face sense and belong to a CPASG, they must have a mapping relationship in a modification process. For example, in Fig. 2 (a), the faces D and 5, belong to the original model and the modified model respectively. They have the same underlying surface and face sense, and their corresponding nodes are in one CPASG ((D,eDE,E),(5,e56,6)). We thus have that face 5 is modified from face D. Thus, for any two given faces having the above relationship, if the status of one of these two faces is modified, it means the boundaries of these two faces are different, and we have that the other face is also a modified face. For example in Fig. 6 (a), nodes A and 1 belong to a CPASG and the edge eAR does not exist in face 1.

So, by an enumeration of the recorded CPASGs, we can identify all additional modified faces. The whole process is shown in Fig. 7. All typical different entities have been locally identified in the process of eliminating CPASGs in section 5. Such as all of the new faces and deleted faces, and all of the different entities corresponding to the PASGs, whose two nodes are contained in two VGNs respectively.

6 IDENTIFYING DIFFERENT ENTITIES FROM THE LEFT ATTRIBUTED GRAPHS

In most cases, adding new entities to the original model, or removing entities from the original model, is adopted in the model modification process. So, all of the entities associating to the new faces and deleted faces, respectively in the original model and modified model, should be different because of the changing of the entities' boundaries, such as the simply example in Fig. 6. However, by chance, some entities (edges or vertices) may be deleted along with faces deletion, but brought back by adding new faces subsequently. Because our approach works on the two models with minor difference, so the new faces and deleted faces are finite. And the number of the entities associating to them is also limited. So, in this work, we take an exhaustive comparison approach to find the different edges and vertices associating to new faces and deleted faces through comparing the links belonging to the left attributed graph of the original model and the links belonging to left attributed graph of the modified.

Fig. 2 (c) shows an example in the result of 'Complement'. In the exhaustive comparison process, if two links are identified as common, then we eliminate these two links from the left attributed graphs of the original model and the modified model respectively. For example, the links, associating to new faces and deleted faces in Fig.2 (b), are eAB, e12, eBC, e23, e34 and e45. After the exhaustive comparison process, the link eBC and the link e23 are identified as common in the exhaustive comparison process, while the other links are identified as different. In Fig. 2 (c), all of the common links in the left attributed graphs are deleted, while all different links are marked in pink. The relationships, between the different entities and the different links, are described in Tab. 1.

Together with the detection results of the process in section 5, all the different entities that we aim to identity between the original model and the modified model are found out.

7 IMPLEMENTATION

A prototype system DiffSeekProtype1 for different entities identification between two solid models in the design process is developed using the geometric modeling kernel ACIS R 19 on a PC with a Intel Core 2.0GHz CPU and 1GB RAM. For ease of comparison, three pairs of B-rep models studied in previous work are tested using the proposed approach. An approach of octree space decomposition is used in [3]. The result is shown in Tab. 2.

8 CONCLUSION

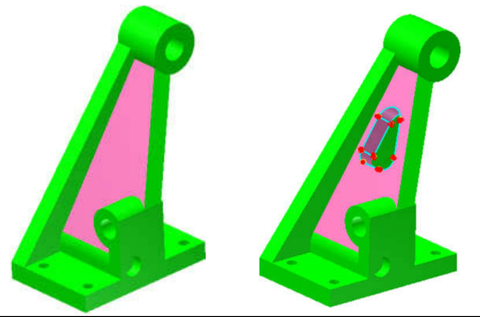
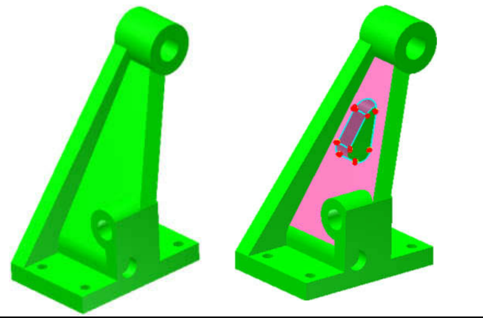
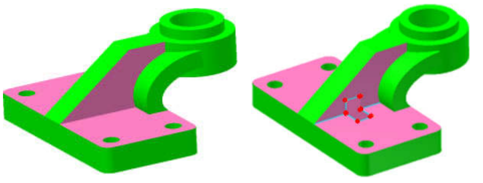
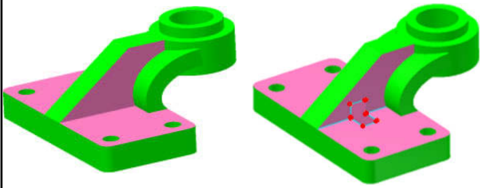
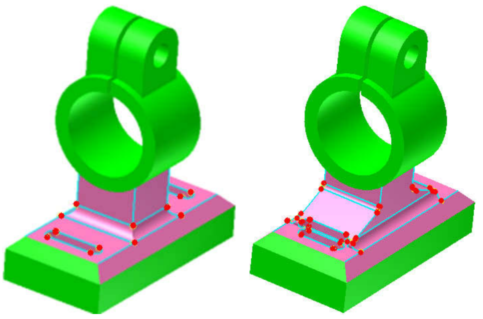
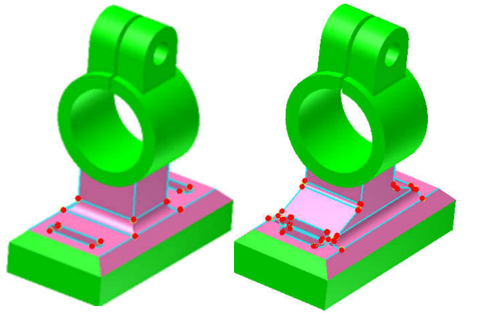
An efficient approach for identifying different entities from minor model modification is proposed in this paper. Considering that the original and the modified models only have minor difference, a strategy of local zone comparison is applied to improve algorithm efficiency. This is achieved by introducing the novel concept of primary subparts and using the attributed graph associated with a B-rep model. By eliminating CPASGs from the attributed graphs of the two models, instead of enumerating all possible cases, our approach greatly reduces the extent that the identification approach needs to consider.

The proposed approach can handle general CAD models in the sense that it works directly on B-rep models without built-in features, which may raise its potentiality in many related research topics such as remeshing, local similarity assessment.

Our future work on this topic will focus on extending the scope of primary subpart for further improvement of the algorithm efficiency.

9 ACKNOWLEDGMENTS

The authors are very grateful to the financial support from the financial support from NSF of China (No. 60736019). They also would like to thank Xiang Chen, Jianguo Tang, Rui Sun, Chuhua Xian for their valuable comments and suggestions on this work.

Case	*DiffSeekProtype1*	*Octree approach*
Case 1		
Average time	0.117 (s)	10.417 (s)
Case 2		
Average time	0.120 (s)	18.291 (s)
Case 3		
Average time	0.295 (s)	10.891 (s)

Tab. 2: Experimental result comparison between our approach and the octree approach [3]. In every cell of the table, the left one is the original model and the right one is the modified model. The obtained different faces, edges and vertices are respectively shown in pink, light blue and red.

REFERENCES

[1] Llewelyn, A. I.: Review of CAD/CAM, Computer-Aided Design, 21(5), 1989, 297-302. DOI:10.1016/0010-4485(89)90036-5

[2] Francois, V.; Cuilliere, J.C.; Gueury, M.: Automatic meshing and remeshing in the simultaneous engineering context, Research in Engineering Design, 11, 1999, 55–66. DOI:10.1007 /s00163 0050005

[3] Francois, V.; Cuilliere, J.-C.: 3D Automatic remeshing applied to model modification. Computer Aided Design, 32(7), 2000, 433–444. DOI:10.1016/S0010-4485(00)00028-2

[4] Di W,; Radha, S.: The incremental editing of faceted models in an integrated design environment, Computer-Aided Design, 36(9), 2004, 821-833. DOI:10.1016/j.cad.2003.09.009

[5] Sypkens Smit, M.; Bronsvoort, W.F.: The difference between two feature models, Computer Aided Design and Application, 4(6), 2007, 843–851. DOI: 10.3722/cadaps.2007.843-851

[6] Sypkens Smit, M.; Bronsvoort, W. F.: Efficient tetrahedral remeshing of feature models for finite element analysis. Engineering with Computers, 25, 2009, 327–344. DOI:10.1007/s00366-009-0123-8
[7] Cuillière, J.-C.; François, V.; el al.: Automatic CAD Models Comparison and Re-meshing in the Context of Mechanical Design Optimization, PROCEEDINGS OF THE 18TH INTERNATIONAL MESHING ROUNDTABLE 2009. Part 4, 231-245. DOI:10.1007/978-3-642-04319-2_14
[8] Osada, R.; Funkhouser, T.; Chazelle, B.; Dobkin, D.: Shape distributions, ACM Transactions on Graphics, 21(4), 2002, 807-832. DOI:10.1145/571647.571648
[9] Xin, H.; Xutang, Z.; Wenjian, L.: Using enhanced shape distributions to compare CAD models. Proceedings of Paeific-Rim Conference on Multimedia. 2007, City Univ Hong Kong, Hong Kong, PEOPLES R CHINA, 358-362. DOI:10.1007/978-3-540-77255-2_41
[10] Hong, T.; Lee, K.; Kim, S.: Similarity comparison of mechanical parts to reuse existing designs, Computer-Aided Design, 38(9), 2006, 973-984. DOI:10.1016/j.cad.2006.05.004
[11] Hu, K.-M.; Wang, B.; et al: A Face-based Shape Matching Method for IGES Surface Model, Shape Modeling International Conference (SMI), 2010,216 – 220. DOI: 10.1109/SMI.2010.15
[12] Li, M.; Fuh, J. Y. H.; Y.F. Zhang, Y. F.; Qiu. Z. M.: General and partial shape matching approaches on feature-based CAD models to support efficient part retrieval. In Proc. of ASME DETC 2008 Computers and Information in Engineering (CIE). 2008. Brooklyn, New York, United States, 121-130. DOI:10.1115/DETC2008-49159
[13] El-Mehalawi, M.; Miller. R. A.: A database system of mechanical components based on geometric and topological similarity Part I: representation, Computer-Aided Design, 35(1), 2003, 83-94. DOI:10.1016/S0010-4485(01)00177-4
[14] El-Mehalawi, M.; Miller. R. A.: A database system of mechanical components based on geometric and topological similarity Part II: indexing, retrieval, matching, and similarity assessment, Computer-Aided Design, 35(1), 2003, 95-105. DOI:10.1016/S0010-4485(01)00178-6
[15] Bespalov, D.; Regli, W. C.; Shokoufandeh, A.: Local feature extraction and matching partial objects, Computer-Aided Design, 38, 2006, 1020–1037. DOI:10.1016/j.cad.2006.07.005
[16] You, C.-F.; Tsai, Y.-L.: 3D solid model retrieval for engineering reuse based on local feature correspondence, Int J Adv Manuf Technol, 46, 2010, 649–661. DOI:10.1007/s00170-009-2113-9
[17] ACIS, http://doc.spatial.com/index.php, Spatial Corp.