Mécanique & Industries 11, 85–92 (2010)
© AFM, EDP Sciences 2010
DOI: 10.1051/meca/2010020

www.mecanique-industries.org

Mécanique
& Industries

Automatic modification retrieval between CAD parts

Khaled Souaissa[1,2], Jean-Christophe Cuillière[1], Vincent François[1,a], Abdelmajid Benamara[2] and Hedi BelHadjSalah[2]

[1] ERICCA, Département de Génie Mécanique UQTR, Trois Rivières, Canada, G9A 5H7
[2] LGM, ENIM, Monastir, Tunisia

Received 30 June 2009, Accepted 11 March 2010

Abstract – Several research works have been focused on integrating FEA (finite-elements analysis) with CAD (Computer Aided Design) over the last decade. In spite of the improvements brought by this integration, research work remains to be done in order to better integrate all the operations led during the design process. Until now, the communication between CAD modules remains static. The design process involves several modifications of an initial design solution. Consequently, there is a need for more flexible communications between CAD modules through the design cycle. Some approaches have been developed in order to reduce the design process length when using FEA, and to automate the transfer of part's data from one step of the process to the next one. Automatic re-meshing is one of these approaches and it is based on automatically updating the part's mesh around modifications zones, in the case of a minor change in the part's design, without the need to re-mesh the entire part. The purpose of this paper is presenting a new tool, aiming at the improvement of automatic re-meshing procedures. It basically consists in automatically identifying and locating modifications between two CAD parts (typically an initial design and a modified design). A major benefit of the approach presented here is that it is completely independent of the description frame, which is made possible with the use of vector-based geometric representations.

Key words: Geometric comparison / BREP / vectorial space / remesh / NURBS

Résumé – Recherche des différences entre des modèles CAO solides. Plusieurs travaux ont été effectués à propos de l'intégration MEF (méthode des éléments-finis) – CAO depuis une décennie. Bien que le processus se soit amélioré, il reste du travail à faire pour intégrer l'ensemble des modules de la CAO. Jusqu'à maintenant, les systèmes de CAO présentent des communications statiques entre leurs modules. Le processus itératif de conception nécessite une communication plus flexible afin d'être plus efficace. Pour réduire le temps de conception lors d'une étude éléments-finis, plusieurs approches ont permis d'optimiser la phase de maillage en n'effectuant, en cas de changement mineur de design, qu'un remaillage autour des modifications de design sans devoir remailler le domaine au complet. L'objectif de cet article est de présenter un nouvel outil, permettant d'identifier et de localiser des modifications entre deux modèles de pièces (une solution initiale de conception et une solution modifiée). Cet outil utilise une formulation vectorielle de la géométrie, ce qui lui permet d'opérer une comparaison complètement indépendante du repère.

Mots clés : Comparaison de géométrie / BREP / vectorisation / remaillage / NURBS

1 Introduction

The design process of mechanical part, usually involves several modifications of an initial design solution. This means that during the design and manufacturing cycle of a given part, the geometry can change several times. When geometry has to be analyzed using FEA (finite-element analysis), the mesh has to be completely rebuilt. This is obviously a great waste of time if we consider that the geometry changes very little when the design is updated. It is even a greater waste of time when the

[a] Corresponding author: francois@uqtr.ca

Article published by EDP Sciences

analysis process requires significant adaptive refinement. Therefore, repeating these tasks at every design modification makes the design process very expensive with regard to processing time. In order to reduce the time of the design phase in product development, automatic updating of models and processes (such as mesh generation and FEA) could induce very important gains with regard to processing time. For example, instead of re-meshing entirely a modified model, it can be re-meshed only around modification zones, while partially preserving the former mesh. Also, when performing a FEA on a modified design, instead of solving the entire modified model, it would be very powerful to be able to solve the problem only in modifications zones and to retrieve, where the design isn't modified, results of previous analyses.

In the perspective to automate these updates, we should primarily know the location of modification zones between an initial design and a modified design. For that, it's necessary to develop automatic processes to detect these modifications.

The purpose of this paper is to present and discuss the different approaches on which the tool is established. The tool has to meet three major requirements. The first one is that the comparison must be independent from the part's description frame. The second requirement is that it must be independent from a specific CAD solid kernel. The third requirement is that the process should be exclusively based on geometric data analysis. This paper is organized as follows. The first section contains a review of related work. The second section presents an overview of vector-based representations, which will be used in the development of the tool presented here. In the third section, we detail four algorithms involved in our method, and illustrate them with examples. Finally, we end this paper with a brief conclusion and perspectives.

2 Related work

Reducing time and cost of product development is one of the primary challenges of today's manufacturing sector. More than 75% of design activity involves reusing previous design knowledge to address a new design problem: about 40% of design activity is based on building from an existing design and 40% is based on modifying an existing design. In aeronautics and automotive industry, where thousands of CAD parts models are involved in product databases, easily re-using design/manufacturing information stored would result in a much faster and more efficient design process. Consequently, it's necessary to provide the design activity with a tool to search for similar model parts inside very large parts databases. Several approaches aiming at the retrieval of a desired model from a large selection of 3D shape models have already been proposed. The different approaches commonly depend of the way of the model part is represented. In [1–3], we find an abounded survey of the methods used in 3D shape similar matching. Hilaga [4] proposes a technique to calculate the similarity between polyhedral models by comparing skeletal and topologic structures. Osada proposes and analyses a method to compute shape signatures with a shape distribution sampled from a shape function measuring global geometric properties of a 3D shape model [5,6]. Ohbuchi [7] improved Osada's method by introducing mutual orientation of surfaces, and used a 2D histogram having distance and angle. Vranic and Saupe [8,9] developed a method based on spherical harmonics. Cicirello and Regli [10, 11] used a graph-based approach to assess the similarity of solid models. Ramesh et al. [12] developed a feature recognition based technique to identify similarities between parts. All these methods try to abstract the shape representation by a vector descriptor in order to calculate a signature that is used to compare two shapes. Furthermore, these methods compare shapes in global way. They do not intend to find the location of modifications, but to estimate a similarity rate between two models. François et al. [13,14] introduce a so called additional format aiming at locating modification zones. The additional format consists in an octree structure for which cells contain information (topologic and geometric) relative to the B-REP (Boundary Representation) model. In this context, the comparison between models is performed by comparing their associated additional format. The weakness of this method is that it is time expensive and closely depends of the two parts description frame. As mentioned earlier, there is a need, in order to achieve CAD/FEA integration, for effective procedures to compare models and locate modifications zones. These procedures could be useful to identify similarities between parts in large databases but would also allow designers to take advantage of previous analyses by updating meshes and analyses only around modified shapes.

Our approach to comparisons between models is inspired by a geometrical analysis based on vector approach [15,16]. It enables existing tolerancing relations to be written between the design parameters. The objective of this method is to analyze dependency relations that may exist between several specifications of a geometric object. The object is modeled on the basis of the TTRS concept [17, 18] and vector modeling synthesized by the metric tensor of a sheaf of vectors. The sheaf of vectors is constructed from the part's surface normal vectors.

In our approach, we will use this metric tensor concept to compare two parts. As we will see later, in our method, the metric tensor is computed from a sheaf of vectors which are constructed from control points of the NURBS (Non-Uniform Rational B-Splines) [19] representation of entities of the CAD model. It is very important to outline that the major benefit of using an approach based on a vectorial space derived into a metric tensor makes that the process is fully independent from the frame considered in both models. Consequently, with our approach, we can identify modifications between two models even if one model is affected by a rigid motion transformation with regard to the other model. In fact, the transformation matrix corresponding to this rigid motion is computed at the end of the process.

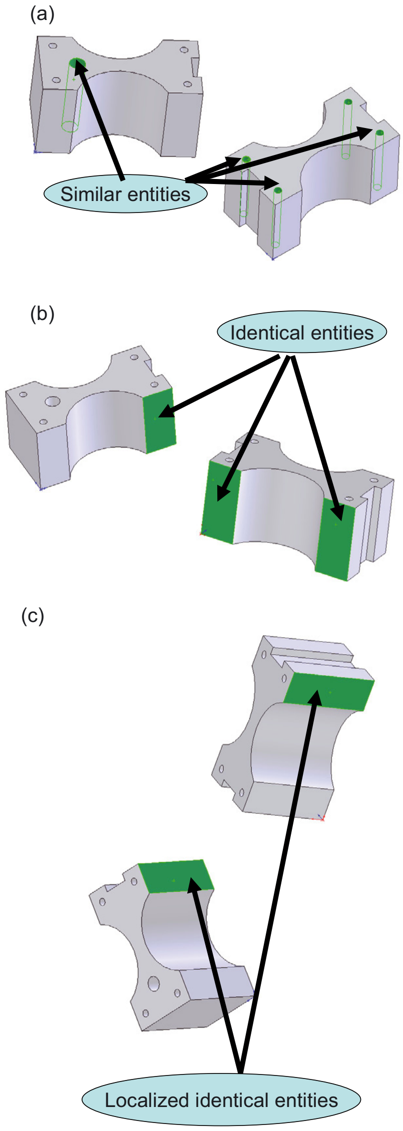

Fig. 1. Types of comparison between entities.

3 Comparison algorithm

In this work we have adopted the following terminology (Fig. 1) with regard to the comparison between two entities of a BREP model (edges faces, etc.) [20]:

- Two geometric entities are similar if they have the same shape regardless of their size.
- Two geometrical entities are identical if they are similar and if their dimensions are the same.
- Two entities are identically localized if they are identical and if their location (relatively to the part's solid model) is the same.

The comparison between two CAD models consists in comparing each entity of the first model with those of the second model. This comparison is practically achieved using four consecutive algorithms (Fig. 2). The first algorithm aims at finding similarities between entities and makes use of the vectorial space concept. The objective of the second algorithm is to find among similar entities listed as a result of the first algorithm, those which are identical. The purpose of the third algorithm to find among identical entities listed as a result of the second algorithm, those which are identically localized. Finally, the last algorithm identifies partial modifications performed on a BREP entity.

In the following sections we will discuss and detail these four algorithms.

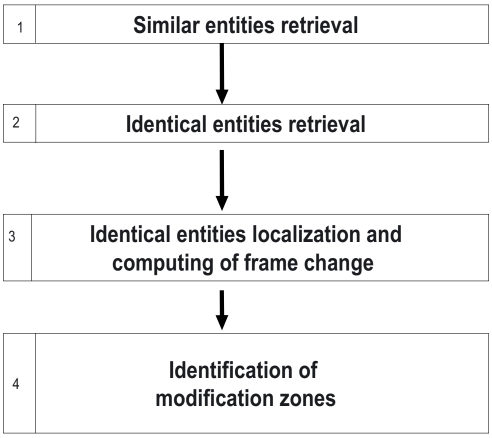
1
Similar entities retrieval
2
Identical entities retrieval
3
Identical entities localization and computing of frame change
4
Identification of modification zones

Fig. 2. Comparison algorithm.

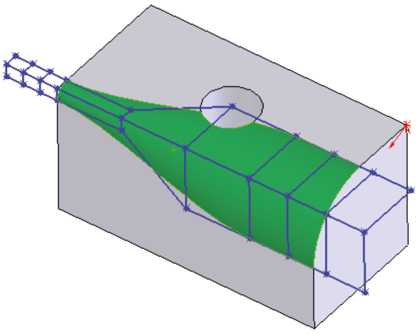

Fig. 3. Surface vectorial space.

4 The vectorial space

We start the process by defining a point cloud from the BREP curves and surfaces control points (Fig. 3). Then, a vectorial space is derived from this point cloud. It consists in a vectors sheaf where each vector is defined by a pair of control points. The coordinates of control points are defined in the homogeneous space. The last component of homogeneous vectors corresponds to the weight associated with control points.

In the case of a vectorial space $\overline{\overline{\overline{T}}}_{\alpha}$ associated with a curve $\vec{c}_{\alpha}(u)$, control points are derived into vectors as follows:

$$\vec{V}_{i+1} = \frac{\vec{P}_{i+1} - \vec{P}_i}{\left\|\vec{P}_{i+1} - \vec{P}_i\right\|}$$

where $\vec{P}_i$ are the control points of $\vec{c}_\alpha(u)$ NURBS definition; and i varies from 0 to $n-1$. This set of vectors is referred to as $\overline{\overline{\Gamma}}_\alpha = \left\{\vec{V}_1, \vec{V}_2, \ldots, \vec{V}_n\right\}$.

In the case of a vectorial space associated with a surface, vectors $\overline{\overline{\Gamma}}_\beta$ are defined using:

$$\vec{V}_{i,j}^u = \frac{\vec{P}_{i+1,j} - \vec{P}_{i,j}}{\left\|\vec{P}_{i+1,j} - \vec{P}_{i,j}\right\|} \text{ and } \vec{V}_{i,j}^v = \frac{\vec{P}_{i,j+1} - \vec{P}_{i,j}}{\left\|\vec{P}_{i,j+1} - \vec{P}_{i,j}\right\|}$$

where $\vec{P}_{i,j}$ are the control points of $\vec{s}_\alpha(u,v)$ NURBS definition; and i varies from 0 to $n-1$ and j varies from 0 to $m-1$, and a list of $\vec{V}_{i,m}^u$ and $\vec{V}_{n,j}^v$ is added.

This set of vectors is referred to as

$$\overline{\overline{\Gamma}}_{S_\beta} = \left\{\vec{V}_{0,0}^u, \vec{V}_{0,0}^v, \ldots, \vec{V}_{n-1,m-1}^u, \vec{V}_{n-1,m-1}^v, \vec{V}_{0,m}^u, \ldots, \vec{V}_{n-1,m}^u, \vec{V}_{n,0}^v, \ldots, \vec{V}_{n,m-1}^v\right\}$$

In Figure 3, we present the vectorial space of the underlying surface of a BREP face.

Once these vectors are computed, a corresponding metric tensor is derived. Using homogeneous coordinates, any vector $\vec{V}$ can be written in a unique way as a linear combination of the four basis vectors $B = (\vec{e_1}, \vec{e_2}, \vec{e_3}, \vec{e_4})$. Given a vectors field $\overline{\overline{\Gamma}}_\alpha = \left\{\vec{V}_0, \ldots, \vec{V}_p\right\}$ defined in this basis. It can be expressed as:

$$\overline{\overline{\Gamma}}_\alpha = \begin{Bmatrix} \vec{V}_0 \\ \vec{V}_1 \\ . \\ . \\ \vec{V}_p \end{Bmatrix} = \begin{bmatrix} V_0^x & V_0^y & V_0^z & V_0^w \\ V_1^x & V_1^y & V_1^z & V_1^w \\ . & . & . & . \\ . & . & . & . \\ V_p^x & V_p^x & V_p^z & V_p^w \end{bmatrix} \begin{Bmatrix} \vec{e}_1 \\ \vec{e}_2 \\ \vec{e}_3 \\ \vec{e}_4 \end{Bmatrix}$$

The metric tensor $\overline{\overline{G}}(\overline{\overline{\Gamma}}_\alpha)$ is defined as the tensor product of the two vector sheaf, through the following expression:

$$\overline{\overline{G}}(\overline{\overline{\Gamma}}_\alpha) = \overline{\overline{\Gamma}}_\alpha \otimes \overline{\overline{\Gamma}}_\alpha = \begin{Bmatrix} \vec{V}_0 \\ \vec{V}_1 \\ . \\ . \\ \vec{V}_p \end{Bmatrix} \otimes \begin{Bmatrix} \vec{V}_0 \\ \vec{V}_1 \\ . \\ . \\ \vec{V}_p \end{Bmatrix}$$

$$= \begin{bmatrix} V_0^x & V_0^y & V_0^z & V_0^w \\ V_1^x & V_1^y & V_1^z & V_1^w \\ . & . & . & . \\ . & . & . & . \\ V_p^x & V_p^x & V_p^z & V_p^w \end{bmatrix} \cdot \begin{bmatrix} V_0^x & V_0^y & V_0^z & V_0^w \\ V_1^x & V_1^y & V_1^z & V_1^w \\ . & . & . & . \\ . & . & . & . \\ V_p^x & V_p^x & V_p^z & V_p^w \end{bmatrix}^T$$

can be written

$$\overline{\overline{G}}(\overline{\overline{\Gamma}}_\alpha) = \overline{\overline{\Gamma}}_\alpha \otimes \overline{\overline{\Gamma}}_\alpha = \begin{bmatrix} \vec{V}_0 \cdot \vec{V}_0 & \vec{V}_0 \cdot \vec{V}_1 & . & . & \vec{V}_0 \cdot \vec{V}_p \\ \vec{V}_1 \cdot \vec{V}_0 & \vec{V}_1 \cdot \vec{V}_1 & . & . & \vec{V}_1 \cdot \vec{V}_p \\ . & . & . & . & . \\ . & . & . & . & . \\ \vec{V}_p \cdot \vec{V}_0 & \vec{V}_p \cdot \vec{V}_1 & . & . & \vec{V}_p \cdot \vec{V}_p \end{bmatrix}$$

5 First algorithm: retrieval of similar entities

As we mentioned above, the goal of the first algorithm is to match similar geometric entities (as defined earlier) between the initial and the modified CAD part. As detailed in the following section, the comparison is based on the metric tensors (as defined just above) of both geometric entities.

5.1 Two types of modifications

In CAD design, modifications that can be performed on an existing part are classified into two categories (Fig. 4):

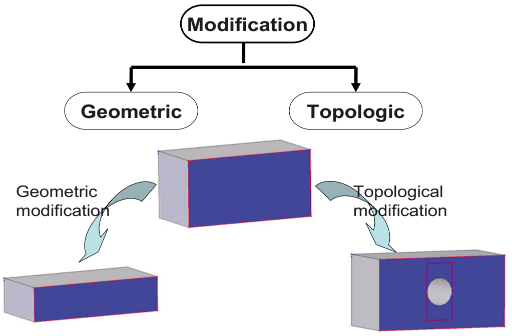

Fig. 4. Geometric and topologic modifications.

(a) Geometric modifications: the initial model is modified by changing dimensions only, without affecting the topology as we define it just below (modifying the diameter of a hole without interacting with other feature for example).

(b) Topologic modifications: the topology of the solid is modified (adding a hole, or changing a chamfer to a fillet, for example). This is the most general type of modification and, by the way, the most difficult to handle. It is important to underline that the definition of topology that is used in this work is wider than those based on the parameters of Euler's rule. In fact, in the terminology that we use, we consider that modifying a fillet to a chamfer induces a topologic change in the BREP even if this does not change its topologic structure. As soon as the type of a BREP geometric entity (a curve or a surface) underlying a given BREP topologic entity, is modified (a cylinder to a plane in the

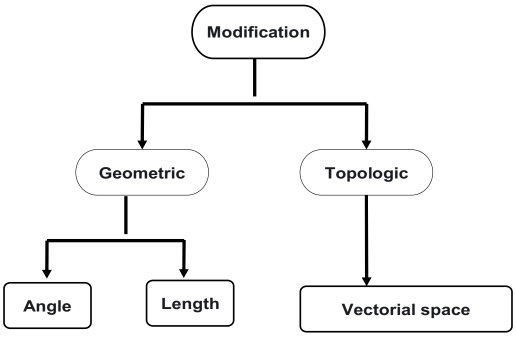

Fig. 5. Types of model modification and relative impact on the vectorial space.

case of a fillet changed to a chamfer) we consider it as a topologic modification in the BREP.

5.2 Comparison criteria

Verifying if two geometric (surfaces, curves, etc.) or topologic entities (faces, edges, etc.) are similar or not can be performed by verifying if their metric tensors are equivalent or not. Before detailing further this comparison, we present criteria on which it is based.

In Figure 5, we present the relation between the vectorial space and the two types of modification introduced above (geometric and topologic). A geometric modification (increasing the length of a face for example) induces modifications of angles and lengths in the vectorial space vectors. On the other hand, a topologic modification (like the addition of a hole in a face) induces modifications of the vectorial space's topology (changing by the way the metric tensor's size). Thus, the abstraction of a CAD geometry into a vectorial space is a powerful tool to identify both types of model modifications.

In the following paragraph, we present the various steps which lead to compare two faces belonging respectively to the initial and modified model (Fig. 6). Overall, this algorithm is applied to all faces of the part:

1. Construct the vectorial space of each face for both CAD models (initial and modified parts). The vectorial space of a given face includes the vectorial space of its underlying surface and the vectorial space of the underlying curves of its boundary edges. Compute associated metric tensors (for both underlying surfaces and curves).
2. Compare the size of the two metric tensors. At this stage, if sizes are different then, the two faces are identified as different.
3. Compare metric tensors of the two underlying surfaces. If these metric tensors are different then the two entities are different. If these metric tensors are equivalent (lines or columns tensors can be exchange), we cannot confirm that the two entities are similar because it is possible that the two underlying surfaces are similar whereas their underlying boundary curves are different. Consequently, metric tensors of underlying curves of the face's boundary edges have to be compared also.
4. Compare the face's boundary edges metric tensors. If they are equivalent then the two entities are similar. If the metric tensors are different then the two entities are identified as different.

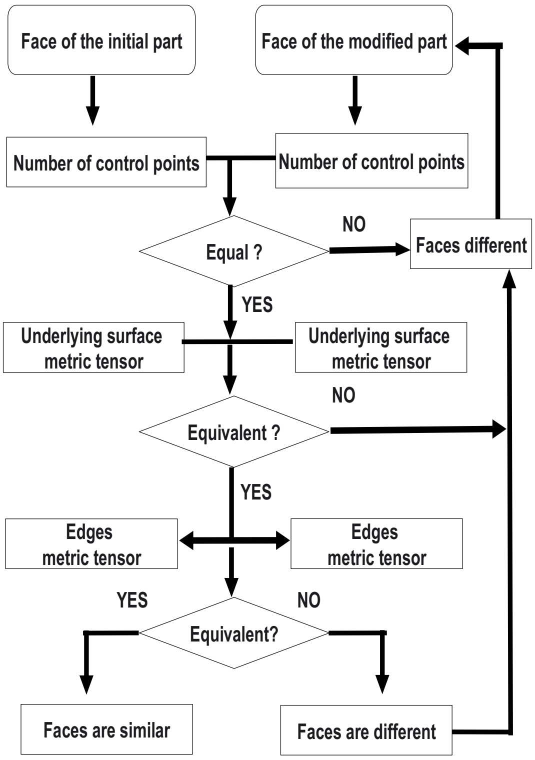

Fig. 6. Vectorial space based similarity comparison.

6 Second algorithm: identical entities retrieval

In this section we present the algorithm used to retrieve identical entities between the initial and modified model. The aim of the second algorithm is matching identical entities among sets of similar entities identified through the first algorithm. This second algorithm is based on inertial properties of sets of control points associated with a unit mass. These inertial properties are referred to as the inertia tensor. The inertia tensor of a set of control points is defined by associating a unit mass to each control points.

In the following paragraph, we present the various steps of the second algorithm (matching identical faces among similar faces) for the comparison of two similar faces belonging respectively to the initial and modified part (Fig. 7). This algorithm has to be applied to all similar faces:

1. Construct a local frame system from the face vectorial space.

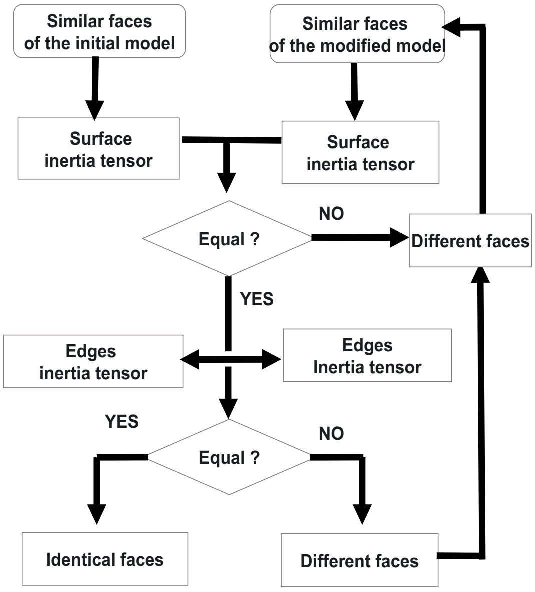

Fig. 7. Inertia-based identity comparison.

2. Compute and compare the inertia tensor of underlying surfaces and their associated curves. If the inertia tensors are different then the two entities are different. If they are equivalent, another step has to be considered because, like in the case of the first algorithm, this condition isn't sufficient to assert that the two faces are identical.
3. Compute and compare inertia tensors of the face's boundary edges. If they are equal then, we can conclude that the two faces are identical.

7 Third algorithm: identical entities localization

The second algorithm presented above finds out, for each face of the initial model, sets of identical entities in the modified part. After this, the aim of the third algorithm is matching localized identical entities among sets of identical entities identified through the first and second algorithms. The third algorithm starts with assuming an initial reference face in each of the two models. These initial reference faces are chosen within sets of faces that have been identified as identical between the two models. Ideally, these initial reference faces should be faces of both models that we are sure to be the same. The problem is that, at this stage, no faces between the two models are known to be the same. Consequently, at the beginning of the process, a pair of identical faces has to be assumed as initial reference faces. For both initial reference faces a local frame system is built using its vectorial space. Then, barycentric vectors (the barycentre of the set of control points defining the inertia tensor) are computed for both models. For a given face of a model its associated barycentric vector is defined as the vector based on the following two points: the face's barycentre and the barycentre of the reference face. The barycentric vector coordinates are calculated with regard to the local frame system of the reference face. Once computed, the comparison method consists in comparing barycentric vectors between two identical faces. In fact, if two faces are identified as identical, feature equal barycentric vectors, they are necessarily at the same location with regard to reference faces. Steps of this algorithm are presented in Figure 8. This algorithm is first applied for an initial pair of reference faces. This results in a list of localized identical faces between the two models. Then, other alternatives in the choice of reference faces are tested and, at the end, the pair of reference faces inducing the maximum number of localized identical faces is kept as the base for the computation of the transformation matrix between the frames associated with the initial and modified models.

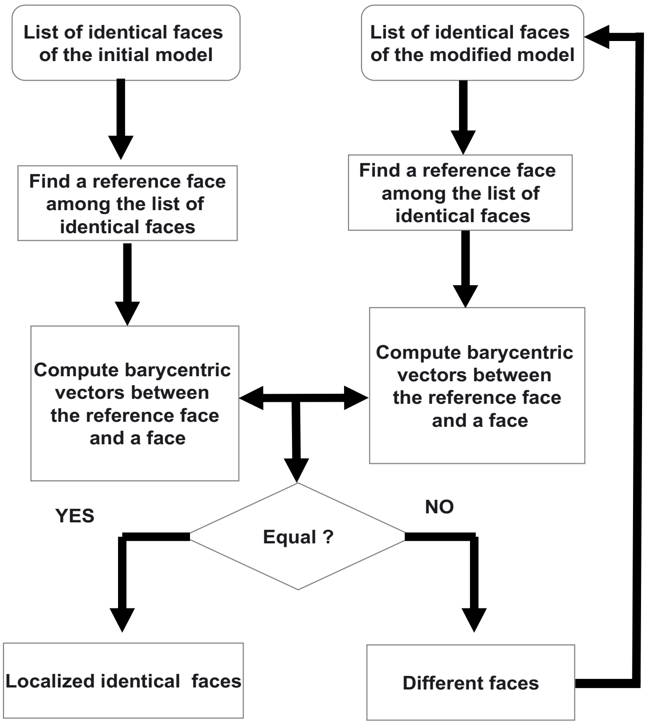
List of identical faces of the initial model
List of identical faces of the modified model
Find a reference face among the list of identical faces
Find a reference face among the list of identical faces
Compute barycentric vectors between the reference face and a face
Compute barycentric vectors between the reference face and a face
YES
Equal ?
NO
Localized identical faces
Different faces

Fig. 8. Localization algorithm for identical faces.

8 Fourth algorithm: identification of the modified zones

Once the identical faces are identified and localized, it remains to identify partially modified entities and, by the way, sub-entities (edges of a face) which are identical between the initial and modified models. Before detailing this process, a terminology has to be specified with regard to entities which have (or have not) been identified as localized identical. These entities are referred to as:

1. Retained entity: if it's identically localized or geometrically modified.
2. Partially modified entity: if one or more sub-entities of the entity are retained (some edges of a face are

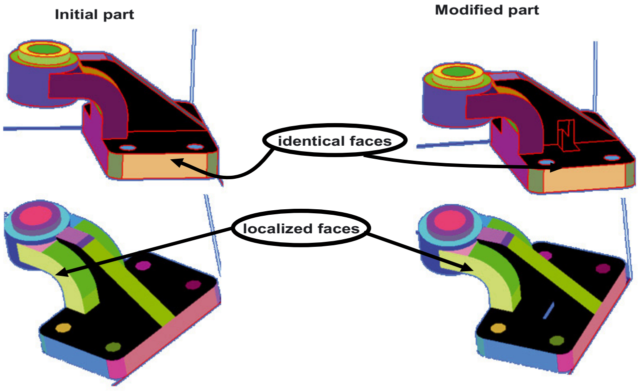

Fig. 9. Example 1.

retained for example) and the underlying geometric entities (curves) are retained.
3. New entity: if the underlying geometric entity is modified.

The three algorithms presented above are able to identify and localize identical entities. To be really efficient, the process needs to find out if entities that have been identified as modified by the first three algorithms, are in fact new entities or partially modified entities. For this, the same concepts as for the third algorithm are used (using a reference frame and barycentric vectors). These concepts are now applied to underlying boundary edges of each modified face.

The algorithm used to find retained entities in a modified face, stands as follows:

1. Use the reference surface and its local frame.
2. Compute inertia curve tensors and compare them.
3. Compute barycentric vectors which separate the curve and the reference surface barycentre. We denote that the barycentric vector is defined in the local frame system of the reference surface.

The final output of the four algorithms mentioned above is constituted of three lists of entities. The first list contains retained and partially modified entities. The second list contains entities of the initial model that have disappeared and the third list contains new entities that have been created in the modified model.

9 Results

Two examples are presented in Figures 9 and 10 to illustrate the algorithms presented in previous sections. Results are identified using color conventions. Entities identified as identical or localized identical are represented using the same color. Entities identified as modified are coloured black. At this step partial modifications cannot be shown. Indeed, the actual definition of partial modification zones is a result of the re-meshing algorithm. Tables 1 and 2 summarize relevant data related to these two examples.

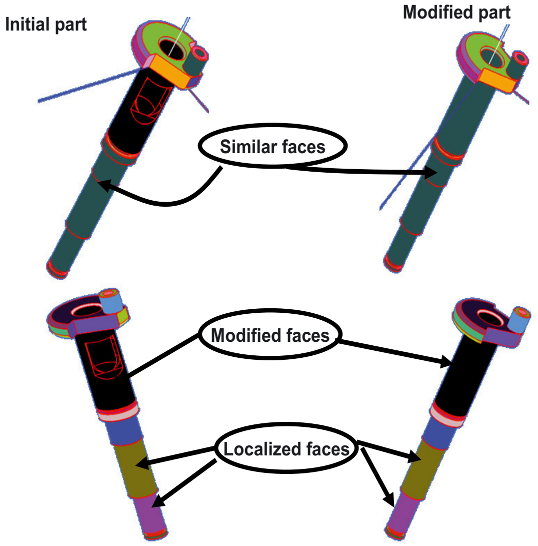

Fig. 10. Example 2.

10 Conclusion

In this work, we implemented an automatic tool for finding modifications between CAD models. This tool is intended for being used in a more general research project aiming at the automatic re-meshing and re-analysis of a part that has been modified during the design process. This paper focuses on the way model comparisons are achieved. The whole process is actually performed through four consecutive algorithms. The first algorithm retrieves similarities between parts through the use of a vectorial space and derived metric tensors. The second algorithm retrieves identical entities among similar entities and is based on the use of inertia tensors. The third algorithm retrieves localized identical entities by searching for reference faces through an iterative process involving barycentric vectors. The purpose of fourth algorithm is to retrieve partial modifications by using barycentric vectors and inertia tensors for underlying curves of the BREP structure.

The use of the vectorial space and metric tensors is crucial in this process as it guarantees that it is fully independent of frames in which the initial and modified parts are defined. In fact, at the end of the process, the transformation matrix between the frames associated with the initial and modified models is computed. The process is also independent of the CAD system used and it is purely based on geometric concepts. Indeed, the vectorial space as well as metric and inertia tensors are basically computed from the BREP's NURBS descriptions. However, the definition of the vectorial space is closely linked with the internal representation of NURBS geometry in the CAD system used. In the context of the work presented in this paper, the intent is to compare two versions of the same model defined in the same CAD system, which means that the same internal representation of NURBS geometry is used for both models. The generalization of the process to comparisons between models coming from

Table 1. Numerical data for example 1.

	Initial part	Modified part
Number of BREP surface patches	37	42
Number of BREP curves	89	113
Total number of control points	864	980
Total number of vectors in the BREP vectorial spaces	756	824
Minimum size of a BREP entity metric tensor	4	4
Maximum size of a BREP entity metric tensor	40	44
CPU time required for identical entities identification (Intel Pentium 4.3 GHz)		6 s
CPU time required for localized identical entities identification		2 s

Table 2. Numerical data for example 2.

	Initial part	Modified part
Number of BREP surface patches	75	92
Number of BREP curves	159	208
Total number of control points	1751	2173
Total number of vectors in the BREP vectorial spaces	1642	1973
Minimum size of a BREP entity metric tensor	4	4
Maximum size of a BREP entity metric tensor	28	40
CPU time required for identical entities identification (Intel Pentium 4.3 GHz)		37
CPU time required for localized identical entities identification		2

different CAD systems, requires to solve the problem of approximating a given shape with a unique NURBS representation.

References

[1] A. Cardone, S.K. Gupta, M. Karnik, A Survey of Shape Similarity Assessment Algorithms for Product Design and Manufacturing Applications, J. Compu. Inf. Sci. Eng. 3 (2003) 109–118

[2] N. Iyer, S. Jayanti, K. Lou, Y. Kalyanaraman, K. Ramani, Three-dimensional shape searching: state-of-the-art review and future trends, Comput. Aided Des. 37 (2005) 509–530

[3] T. Hong, K. Lee, S. Kim, C. Chu, H. Lee, Similarity comparison of mechanical parts, Comput. Aided Des. Appl. 2 (2005) 759–768

[4] M. Hilaga, Y. Shinagawa, T. Kohmura, T.L. Kunii, Topology Matching for Fully Automatic Similarity Estimation of 3D Shapes SIGGRAPH. Los Angeles, 2001, pp. 203–212

[5] R. Osada, T. Funkhouser, B. Chazelle, D. Dobkin, Shape distributions, ACM Transactions on Graphics 21 (2002) 807–832

[6] R. Osada, T.A. Funkhouser, B. Chazelle, D.P. Dobkin, Matching 3D Models with Shape Distributions, Shape Modeling International, IEEE Computer Society, 2001, pp. 154–166

[7] R. Ohbuchi, T. Minamitani, T. Takei, Shape-Similarity Search of 3D Models by using Enhanced Shape Functions, TPCG, 2003, pp. 97–104

[8] D. Vranic, D. Saupe, J. Richter, Tools for 3D-Object Retrieval: Karhunen-Loeve Transform and Spherical Harmonics, IEEE Fourth Workshop on multimedia signal processing, Cannes, France, 2001, pp. 293–298

[9] D. Saupe, D.V. Vranic, 3D model retrieval with spherical harmonics and moments, DAGM, 2001, pp. 392–397

[10] V.A. Cicirello, Intelligent Retrieval of Solid Models, Drexel University, Philadelphia, PA, 1999

[11] V.A. Cicirello, W.C. Regli, Machining Feature-Based Comparisons of Mechanical Parts, Shape Modeling International, 2001, p. 176

[12] M. Ramesh, D. Yip-Hoi, D. Dutta, Feature Based Shape Similarity Measurement for Retrieval of Mechanical Parts, J. Comput. Inf. Sci. Eng. 1 (2001) 245–257

[13] V. Francois, J.-C. Cuilliere, 3D Automatic remeshing applied to model modification, Comput. Aided Des. 32 (2000) 433–444

[14] V. Francois, J.-C. Cuilliere, Automatic mesh pre-optimization based on the geometric discretization error, Adv. Eng. Softw. 31 (2000) 763–774

[15] P. Serre, A. Riviere, A. Clement, Geometric Product Specification and Verification: Integration of Functionality: Analysis of fonctional geometrical specification, Kluwer, 2003, pp. 115–125

[16] P. Serre, A. Riviere, A. Clement, The clearence effect for assemblability of over-contrained mechanism, 8th CIRP International Seminar on Computer Aided Tolerancing, Charlotte (USA), 2003

[17] A. Clement, A. Riviere, P. Serre, Geometry and topology declarative: a new paradigme for CAD/CAM systems, IDMME. Compiègne, France, 1998, pp. 587–595

[18] A. Clement, A. Riviere, P. Serre, C. Valade, The TTRS: 13 Constraints for Dimensionning and Tolerancing, 5th CIRPP International Seminar on Computer Aided Tolerancing, Toronto, Canada, 1997

[19] L.A. Piegl, W. Tiller, The NURBS book, 1997

[20] M.E. Mortenson, Geometric Modeling, 1985