## Science Arts & Métiers (SAM)

is an open access repository that collects the work of Arts et Métiers ParisTech researchers and makes it freely available over the web where possible.

This is an author-deposited version published in: http://sam.ensam.eu
Handle ID: .http://hdl.handle.net/10985/8366

**To cite this version :**

RYM CHOUADRIA, Philippe VERON - Identifying and remeshing contact interfaces in a polyhedral assembly for digital mock-up applications - Engineering With Computers - Vol. 22, n°1, p.47-58 - 2006

Any correspondence concerning this service should be sent to the repository
Administrator : archiveouverte@ensam.eu

R. Chouadria · P. Véron

# Identifying and re-meshing contact interfaces in a polyhedral assembly for digital mock-up

**Abstract** Polyhedral models are widely used for applications such as manufacturing, digital simulation or visualization. They are discrete models; easy to store, to manipulate, allowing levels of resolution for visualization. They can be easily exchanged between CAD systems without loss of data. Previous works (Comput Aided Des 29(4):287–298, 1997, Comput Graphics 22(5):565–585, 1998) have focused on simplification process applied to polyhedral part models. The goal of the proposed approach is to extend these processes to polyhedral assembly models, describing the digital mock-up of a future manufacturing product. To apply simplification techniques or other processes on polyhedral assemblies, contact surfaces between interacting objects have to be identified and specific constraints must be applied for processing. The approach proposed allows checking and maintaining a global consistency of the assembly model to ensure the reliability of the future processes. Thus, contacts between objects are detected using an approach that works for a static configuration of the assembly. Finally, a precise detection of the faces involved in each contact area is made and the resulting input domains identified are processed using a local Frontal Delaunay re-meshing technique to produce an identical tessellation on both objects involved in the processed contact. The quality of the triangulation produced is also checked.

**Keywords** Assembly · Polyhedral model · Contact detection · Triangulation

## 1 Introduction

Nowadays, the assembly models, widely used in CAD systems, are based on a parametric description of parts associated with relationship information between the parts. Such assembly information can be controlled and checked using geometrical constraints defining the different connections between the parts. In CAD systems, more specific information can also be associated to the assembly model such as part materials or the kinematics description of joints.

Nevertheless, these models are widely dependent on the CAD software, sharing and exchanging them during the overall design process is often difficult and inefficient. Indeed, a fraction of data is lost during the transfer and conversion processes. Moreover, assembly models based on parametric description of shape do not fit for specific applications such as manufacturing or finite element analyses for example. Polyhedral models are more adapted and used in these cases [11]. Rapid prototyping and manufacturing applications are two fields where the geometric models need to be precise [1, 8]. Digital mock-up visualization and manipulation are other applications where the models need to be easy and fast to handle [8, 13]. A polyhedral model is also simple to store and it can be produced by all CAD systems. Its discrete property makes it easy to translate and convert with no loss of data. Multi resolutions techniques can be applied, which allows its handling under multiple levels of details.

In this context, the overall goal is to introduce polyhedral models as a standard for all numerical simulation processes. Previously, decimation and adaptation techniques have been developed for a single polyhedron (part model) [15, 16]. The scope of this paper lies in extending and applying these decimation tools to assemblies of polyhedral models. Specific tools required to process assembly specifications are presented. These tools are able to identify and characterize the contact areas of the parts of the assembly for their future processing. A preliminary phase is necessary to check the consistency of the contact areas identified and to repair them before applying specific decimation processes such as envelope generation or constraint decimation respectful of the contact areas. Indeed, the main difficulty is that the contact areas have to be meshed (facetted) identically on both interacting objects of the assembly, to maintain the overall consistency during the future decimation process. This is also necessary if

R. Chouadria (✉) · P. Véron
Laboratoire des Sciences de l'Information et des Systèmes (LSIS),
UMR CNRS 6168-Equipe Ingénierie, Mécanique, Systèmes,
Ecole Nationale Supérieure d'Arts et Métiers, 2, cours des arts
et métiers, 13627 Aix en Provence, France
E-mail: rym.chouadria@aix.ensam.fr
E-mail: philippe.veron@aix.ensam.fr

the contact areas have to be deleted to produce assembly envelope or exploited for finite elements purposes.

From a general point of view, a polyhedral assembly is processed as a set of polyhedral part models with specific additional assembly constraints. Finally, the processing of polyhedral assembly models including decimation, envelope generation or kinematics joint extraction provides more adapted models for virtual mock-up or simulation applications. All these treatments can be applied to assemblies provided by reverse engineering processes characterized by their error measurements and their inconsistencies produced during the reconstruction step.

Section 2 shows a general overview of the process. A first coarse contact detection step and an exact contact area detection step are described in Sect. 3. Then, the identification of the boundaries of the contact for re-meshing is detailed, and the meshing techniques adopted are described in Sect. 4. Section 5 describes the pre-processing step that checks consistency of the input polyhedral assembly. Finally, an example of assembly envelope generation is given in Sect. 6.

## 2 Overview of the overall process

For polyhedral assembly simplification, as for assembly envelope creation purposes, the assembly needs first to be read correctly and repaired to remove inconsistency in the part models usually resulting from CAD data transfer tessellation algorithms. Then, specific information has to be extracted for future processing. This information is known as contact areas and sets of faces involved in each contact. The last step of the assembly processing consists in re-meshing the contact areas identified. Here, the aim is to produce a local mesh similar on both interacting objects.

The different assembly processing steps are described in Fig. 1. Input data can be either tessellated CAD parts or polyhedral parts generated by a reverse engineering process. The output of the assembly processing is a consistent polyhedral assembly that is appropriate for efficient decimation or envelope generation purposes.

Some terms need to be defined for better understanding. A contact area is defined as a set of faces of contact and a domain is defined as an area containing only edges and vertices for meshing purposes. A contact interface is defined as a contact area delimited by edges along the exact contact boundary. Entities of the mesh are edges and vertices. Polyhedral models are referred as parts or objects.

## 3 Contact detection between parts

The contact detection step consists in seeking for contacts in the assembly. It is important to have the pairwise parts in contact, as it is necessary to get the contact

interface (the exact contact area) for every pair of parts. Many works focus on collision detection [4, 6] and some approaches use static configurations that can be used in contact detection, but handling static contact, from the mechanical point of view has not been treated. An approach inspired from a collision detection technique [5] based on an octree space partitioning has been developed. This approach has been adapted to handle static contacts. Tolerances in the contact detection and tangency case treatments have been introduced [3].

The method is based on two sub-steps. The first one detects coarsely the pairs of parts in contact. An aligned axis bounding box surrounding every part is constructed. Intersections and contacts (tangencies) between bounding boxes are checked using simple, robust and fast tests on the bounding boxes. The boxes are projected on the $x$, $y$ and $z$-axes, and they are then defined by intervals on the axes. If $x$, $y$, $z$ intervals of two boxes overlap or are tangent (relatively to the tolerance), then the boxes are potentially in contact. Figure 4 shows two boxes overlapping defining a contact between a shaft and a part with a hole. For each potential contact identified, an "intersection zone" (volume, surface, line or point) is computed and the presence of at least one face of each part of the pair considered is checked (Figs. 2, 5). If this last condition is satisfied, the parts are really in contact. In Fig. 2 the objects Object A and B have their bounding boxes Box A and box B built, and an intersection zone exists. The result of this step is a list of pairs of parts recording all the existing connections between the parts in the assembly.

The second sub-step of the contact detection procedure collects the sets $F_A$ and $F_B$ of faces lying inside or tangent relatively to a tolerance to the "intersection zone" (Figs. 3, 4). Aligned axis bounding boxes are constructed for all the faces $F_A$ and $F_B$, if the intervals of the intersection zone and the box of a face overlap or are tangent relatively to a tolerance on the three axes then the face is candidate for contact (Fig. 5). The list of candidate faces $F_A$, $F_B$ are checked for contact. First, the previous overlapping or tangent intervals on bounding boxes of faces is used. Then if $F_A$ and $F_B$ intersect [3] they are in contact, if they are parallel the distance between them must be lower than the clearance distance. The resulting faces in contact are listed by pair $(F_A, F_B)$ Fig. 6. This defines the contact area on every part (Fig. 3).

Every contact defined between two parts implies the definition of a joint. At this stage, the joint contains the two interacting parts, the faces involved in the contact and a joint type. The type only characterizes planar and non-planar joints, depending on the intersection zone's geometry.

If the intersection zone is a volume, the joint is non-planar. It can either define non-planar contacts, or complex contacts consisting of disconnected contact areas, even if the contact areas are planar. An intersection zone that is a rectangular surface characterizes a planar joint, but also linear or punctual joints. Linear

**Fig. 1** General overview of the polyhedral assembly processing proposed
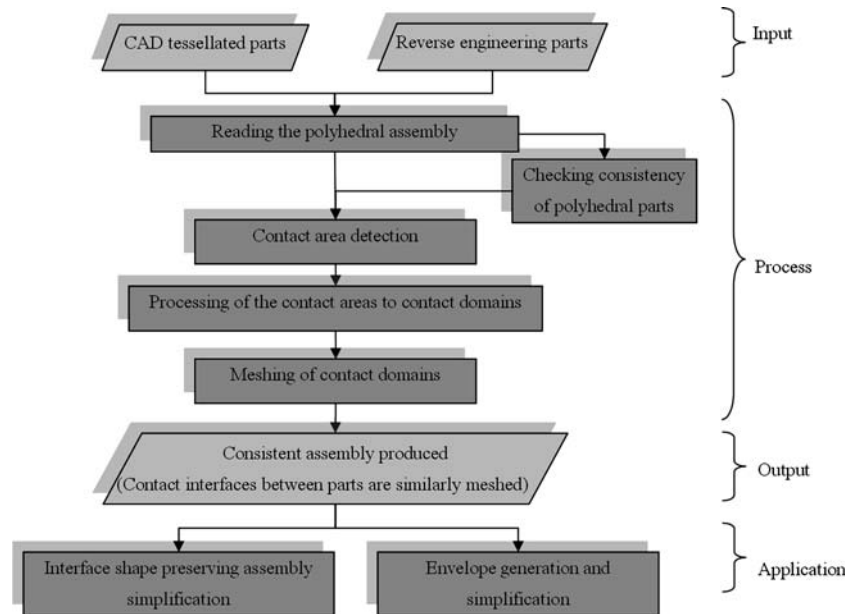
joints can be retrieved if the mesh tessellation is modified, and the boundary of the contact line described. For punctual joints this can be done only if the contact point exists in the tessellation.

The joint type can be translated into real one (i.e., pin, slider, ball etc.) in a further process, by analyzing the geometry and the curvature of the contact interface, only if the exact boundary of the contact is defined. It is first necessary to modify the tessellation of the contacts on both objects.

The result of the contact detection process is, for each contact identified between two parts, a list of pairs of faces in contact. Figure 6 shows light grey faces for the part with a hole and dark grey faces for the shaft. The list of pairs of faces $(F_A, F_B)$ (Fig. 6) that needs to be analyzed and processed to produce the exact contact interfaces.

## 4 Re-meshing contact areas

The input data for this step is composed of the sets $(F_A, F_B)$ of faces in contact of the two respective parts (Fig. 7) which define the contact areas. Some of these faces are partially involved in the real contact area, because of their tessellation. The boundary of the shared
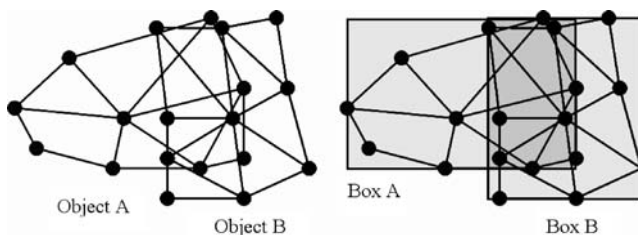


**Fig. 2** Polyhedral model assembly and its corresponding bounding boxes

contact area is not strictly identified, and it is not the same on the respective pairs of objects. This must be corrected if the exact contact interface, and boundary edges of the contact are required for stress calculation or simply for quality visualization purposes.

The object of this re-meshing process is to set the same polyhedral representation on both parts in contact. A pre-processing of the contact areas defines a new domain to re-mesh taking into account all the common entities of the contact.

Then, edges of the contact areas need to be identified as entities at the contact's boundary, and these edges must be present on both contact interfaces. Local meshing modifications must be done. The contact areas have to be re-meshed with edges in common as constraints.

### 4.1 Definition of the domains to mesh

The contact areas are two discrete representations of the same contact on both objects. They can be parallel and the distance between the objects corresponds to a tolerance introduced by the CAD system, or the 3D scanning system. It is fundamental to maintain the assembly's characteristics while the mesh is modified. In consequence, the contact areas are not moved, and the contact tolerances are maintained. These constraints on the meshing domain determination problem increase its complexity.

The determination of the domain to re-mesh is divided in three sub-steps (Fig. 8):

#### 4.1.1 Removal of non-significant edges

This step extracts the edges at the boundary of each contact area, and gives to these edges a ''boundary''
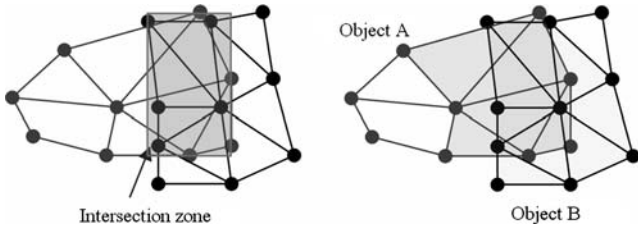
**Fig. 3 a** Intersection zone between the parts A and B, **b** the sets of faces $F_A$ and $F_B$ defining the initial contact areas identified between the parts A and B

label. The internal edges of the contact interface with significant curvature are also extracted. These edges identification is based on a mean curvature criterion computed on each internal edge. The discrete mean curvature $H_e$ [16] is defined for an edge (Fig. 9) by:

$$H_e = \frac{1}{2} \times \beta \times l_e \tag{1}$$

with $\beta$ angle between adjacent faces, $l_e$ edge length.

Edges with "low" mean curvature (close to zero) are removable without loss of information on the curvature. They are simply removed from the domain. Edges kept are considered "sharp" as they define curvature variations of the surface. All the vertices of the contact area are kept for re-meshing because they define the domain as they are contained by the initial surface before tessellation.

It is interesting to note that the use of a simple angle criterion is sufficient to define edges that shall be kept because the purpose is not shape recognition at this stage but only definition of entities of domains to re-mesh.

We want to keep the faces of the contact so non-significant edges are not removed. Edges are not removed. The boundary edges and the edges with significant curvature are labeled non-removable.

### 4.1.2 Subdivision of "edges in contact"

The sets of faces of the interacting contact areas are intersecting or distant. For distant faces, edges of each
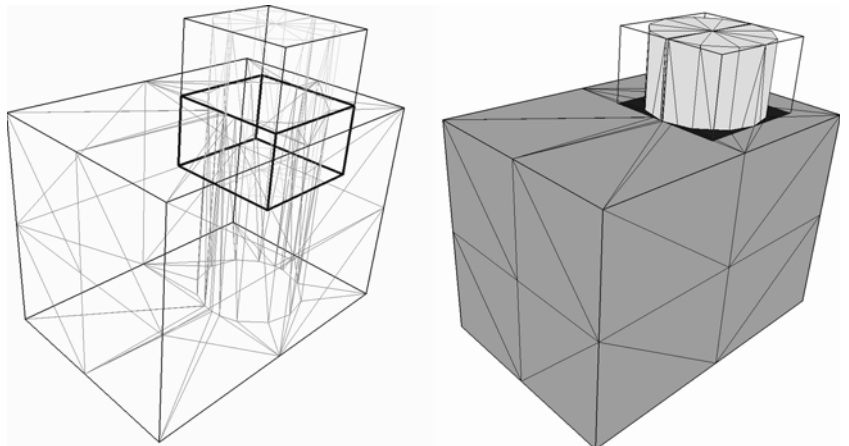
face $F_A$ must be projected on the opposite face $F_B$ to check for edges' intersections. Only edges that have been previously labeled non-removable are subdivided, and the contact is dynamically updated. For intersecting faces $F_A$ and $F_B$ no projection is needed. The intersection point is computed for both cases. To take into account this intersection point in the contact areas, the intersecting edges are subdivided on both interfaces. For every pair of intersecting faces ($F_A$, $F_B$), intersecting edges are detected and then subdivided with different schemes depending on the configuration of the computed intersection point (Figs. 10, 11). Figure 10 a shows the case of two faces in contact with respective intersecting edges AB and A′B′, the two edges are subdivided and a common vertex is created for both contact areas. Figure 10b shows the case of intersecting edges with a subdivision point close to the extremity A′, only AB is subdivided and the subdivision point is moved to A′. A circle of tolerance around every intersection point is defined to determine if any vertex is close or coincident. Figure 10c shows the case of two intersecting edges, with an intersection close to respective extremities A′ and B. None of the edges are subdivided A′ is moved to coincide with B.

Such schemes avoid the creation of too small edges because only non-removable edges are subdivided. The subdivision is propagated to adjacent faces if necessary, and the mesh is dynamically updated during the process.

### 4.1.3 Insertion of shared entities

This sensitive step projects significant edges that are in common on both objects and inserts them into each domain $D_A$ and $D_B$ to re-mesh. The edges are internal edges and edges of the boundary of the contact that are in common. These edges have been previously labeled, only those who are shared by the contact area are inserted. This ensures that the contact areas are similarly re-meshed on both objects.

The faces of each interface can be distant of a given value or not. This distance, which represents a clearance

**Fig. 4** Overlapping bounding boxes, the contact between shaft and part with a hole is within the intersection zone

**Fig. 5** Faces candidate for contact are inside and tangent to the intersection zone
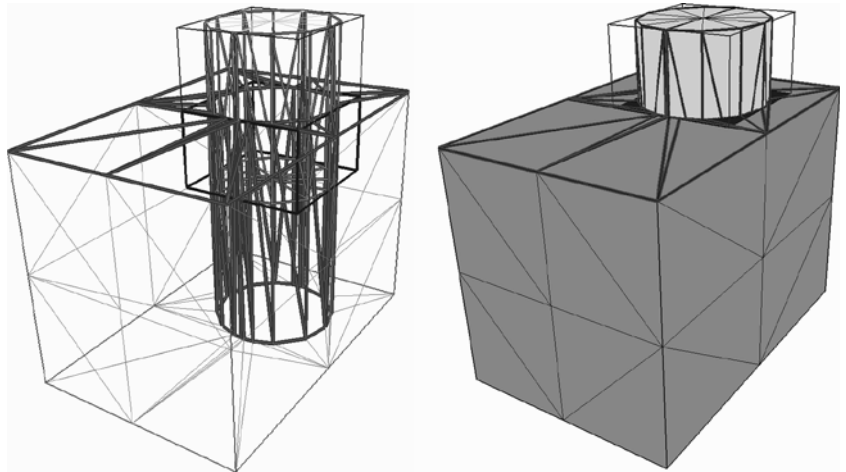
defined in the assembly model to ensure the kinematics mobility between the parts, must be preserved. The contact interfaces processed are a combination of planar, cylindrical and spherical areas, and they do not have, a priori, the same tessellation.

For every contact area A the shared edges are edges contained in the intersection area, and they are inserted into the opposite contact area B (Fig. 7). For planar contact areas the inserted edges are simply projected on the facing contact area. For non-planar contact areas edges are inserted with mean curvature checking. As every edge is inserted it is subdivided to fit the local curvature of the contact area Fig. 13. Inserting edges

without checking the local mean curvature generates edges that do not lie on the initial surface.

This step is sensitive because inserting edges that lie on the initial object affects the re-meshing and envelope creation step. An edge that lies on the initial mesh generates a face that lies on the initial mesh, and does not modify the geometry and the shape of the parts. The clearance distance is not modified nor information resulting from the tessellation algorithm lost (Figs. 12, 13). For envelope creation it is less easy to create envelopes, because as the objects are kept distant, vertices are distant and when merging objects this should be carefully handled.

The edges are only added to the mesh but the contact is not updated. This step allows us to establish constraints, the inserted edges, for the meshing process.

The domains are appropriate for the re-meshing process which will produce similar meshes of both contact areas because of the inserted edges.
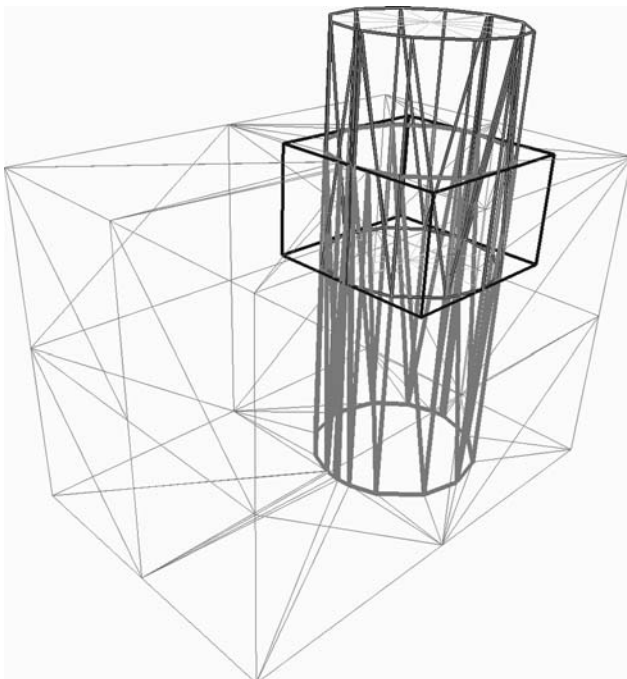
### 4.2 The frontal re-meshing technique proposed

In a first time, we have tried to apply an adaptive refinement approach to our problem. Edges are subdivided dynamically, and the faces of the contact area are updated. Shared edges are then inserted in the contact faces, and the faces can be split or the edges swapped depending on the edge location, and the contact type.
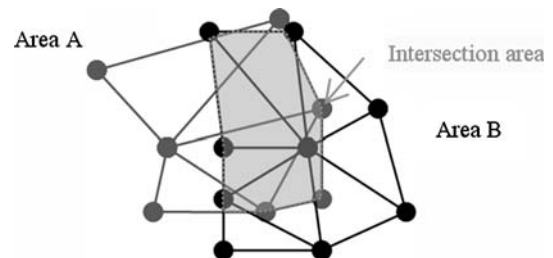


**Fig. 6** Faces in contact for every part defining two contact areas of every part



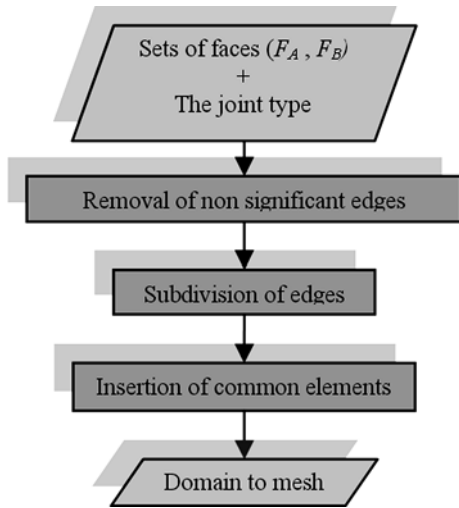**Fig. 7** Contact areas: sets of faces $F_A$ and $F_B$

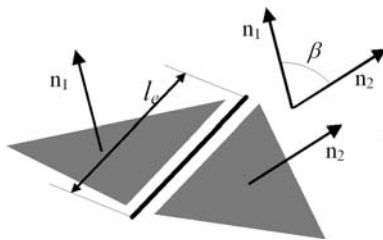**Fig. 8** Pre-processing algorithm of the area to re-mesh



**Fig. 9** Discrete mean curvature of an edge

We must then ensure the contact are similarly meshed on the interacting objects. This approach results in creating a lot of faces even for planar contacts, and if the tessellation is fine the edges are very small in the contact area. It is clear that the contact area can be decimated, but in case of large assemblies such an approach is not efficient. Ensuring a minimum of faces created can only be done using a re-meshing technique on significant edges defining the domain to re-mesh. The domain lies on contact faces to respect the curvature of the contact area.

The domains $D_A$ and $D_B$ previously identified for each contact are the input data of the re-meshing step. They are constituted of a set of vertices and edges as constraints. The edges are at the boundary of the contact areas, internal edges with significant curvature, and inserted "shared" edges.

The technique adopted must generate two-dimensional elements (triangular faces) in a three-dimensional context. Moreover, edges "shared" by the contact areas of both object must be respected as constraints to ensure a similar meshing of the two domains for FE applications, and assembly decimation. The meshes produced must be optimal related to at least a local Delaunay criterion.

The main mesh generation methods [2, 9, 10] for unstructured triangular meshing problems are spatial partitioning methods, Delaunay-type methods and advancing front methods.

Spatial partitioning methods also called quadtree/octree have been introduced by Sheppard [14]. They work by recursive subdivision of a cube containing the geometric model. The subdivision stops when a chosen precision is reached. The resulting mesh cannot match a given surface. If the geometric model's orientation changes then the whole mesh is modified. Additional smoothing of the mesh is necessary to make the elements' shape improved.

The Delaunay-type methods are very popular. They are based on the Delaunay criterion, also called empty sphere criterion, applied by Watson [17] for mesh generation. The criterion is used for connecting the nodes so that each node of the mesh is outside the circumscribed circles of the triangles of the mesh in 2D. The domain is made convex, so the boundaries of the domain must be retrieved after meshing. Thus the boundary elements of the domain can be constrained. The respect of edges as constraints is forced during or after the meshing process.

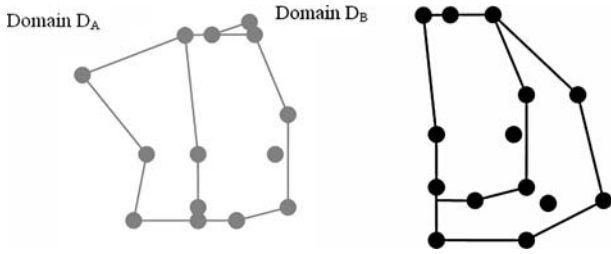**Fig. 10** Edge subdivision cases
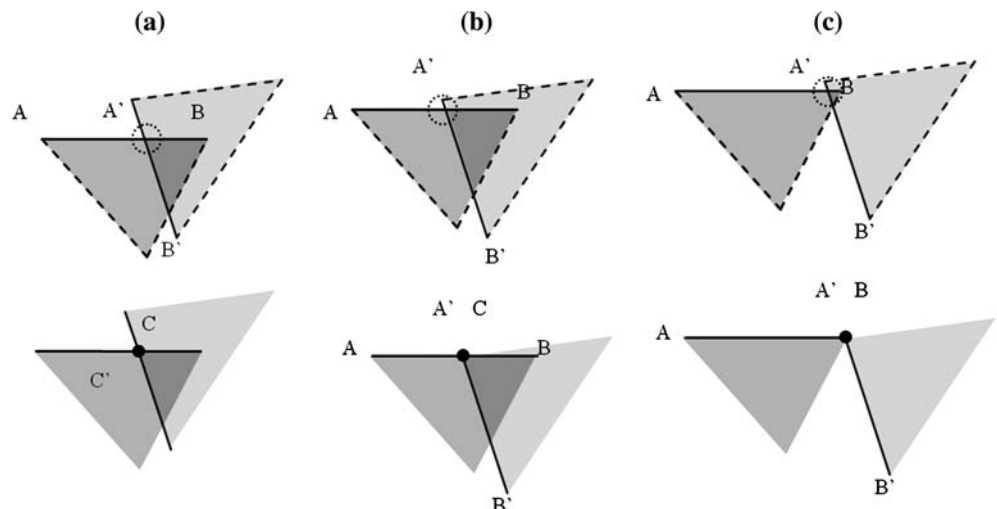
**Fig. 11** Domains to re-mesh

Then the mesh is not strictly a Delaunay mesh. The case of surface meshing is similar as no Delaunay criterion can be applied successfully to the whole surface; the choice of a local criterion allows a Delaunay-admissible mesh [12]. The advantage of using Delaunay type methods is a guaranteed optimal quality of the resulting mesh for a given domain.

Advancing front, or frontal methods [7] are also very popular. In these methods the mesh is built progressively by propagating the mesh from an initial contour called front. An active front is maintained for the generation of new faces. The front is spread until the surface is completely covered. Additional tests are necessary to avoid overlapping faces on the domain. Quality criterion must be integrated to optimize the obtained mesh. The edges of the domains are respected (preserved) during the process.

Delaunay-type and frontal methods are appropriate for our domain meshing problem. The Delaunay type methods have to be adapted to the "boundary" and internal edges, this will add constraints to the process. While frontal methods must be improved with a mesh quality criterion. The two techniques must include a procedure to project "shared" edges, edges of the intersection area, because the contact areas A and B are meshed one by one. The new "shared" edges of contact area A resulting from the meshing procedure are projected on the contact area B, and then contact area B is re-meshed with these new edges as constraints.

As there are many constraints, we adopted a frontal approach of the domains' meshing, as forcing constraints with a Delaunay-type method is complex and time consuming. The Delaunay criterion is used locally to ensure a Delaunay-admissible surface mesh, the



**Fig. 12** Insertion of edges, with respect of curvature of the contact and the gap resulting from the tessellation
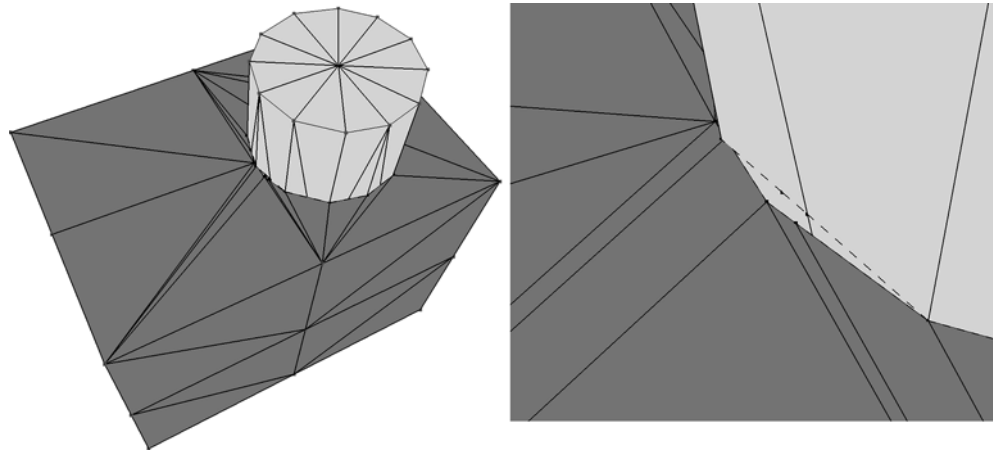


**Fig. 13** Insertion of edges and nodes, on the hole's faces and shaft's faces, the contacts are not updated
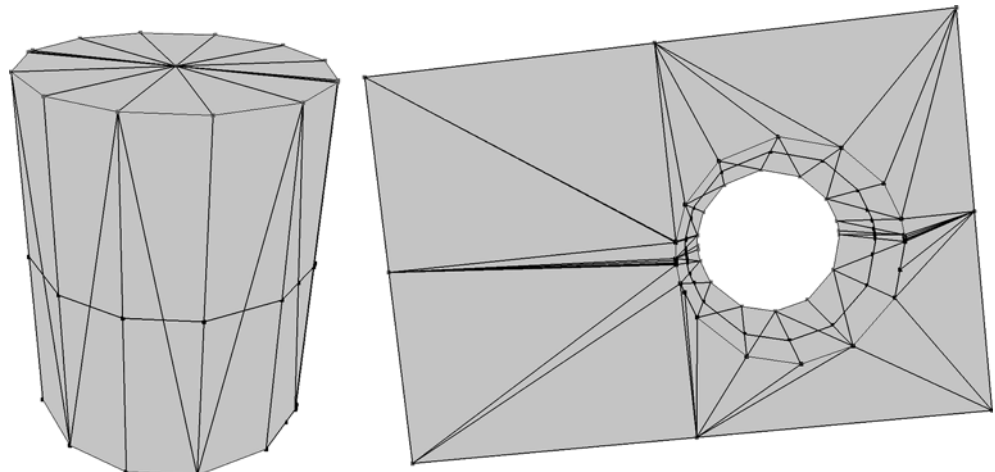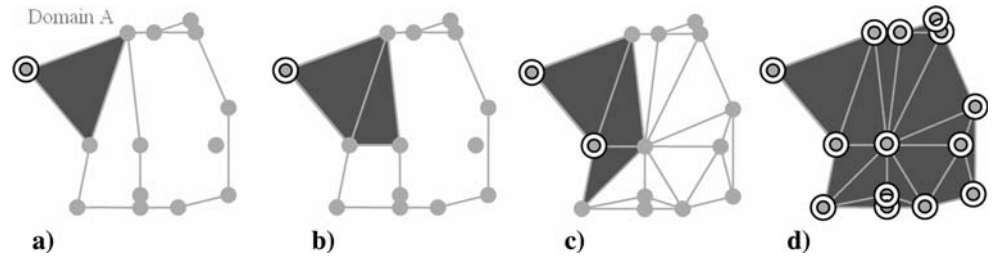
Fig. 14 Front propagation from a first edge

method is a frontal-Delaunay method. The joint type gives information on planar and non-planar contacts, we use planar process on planar domain.

### 4.3 Frontal delaunay method

To start, the front is an edge of the "boundary" edges of the interface, or an internal edge. From this starting arbitrary edge a first face is created with the closest node (Fig. 14a). The new front is then made up from the first face's edges. The process is repeated for each edge of the new front (Fig. 14b, c). The front is updated iteratively on the surface until the domain is completely covered (Fig. 14d).

Defining an "eligible" node is the key of the process. This concept encloses all the constraints of the meshing process, so that the mesh is optimal.

First, an "eligible" node does not generate intersections with the edges of the existing domain. The edges of the domain are the edges of the boundary; the internal inserted edges and edges created during previous iterations of the meshing process.
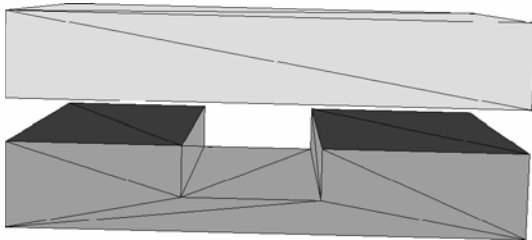


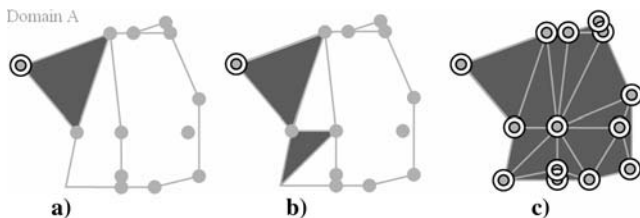Fig. 15 Assembly with two contact areas for each part, with a gap not to re-mesh



Fig. 16 Front propagation from all the front edges

An "eligible" node always generates edges lying on the initial mesh. This avoids meshing existing holes or cavities of the objects (Fig. 15). In Fig. 14a, after creating the first face a first node is considered non-eligible, the node surrounded by two circles is not eligible any more; this avoids creating an overlapping face. Figure 14c shows two non-eligible nodes.

Overlapping faces are avoided by checking the propagation direction, and by eliminating nodes, only connected to surface edges during the process, from the eligible node list.

The local Delaunay criterion is applied during the advancing front process. As a triangle is created it is locally checked for the Delaunay criterion, because the process handles non-planar contact areas. Non-planar contacts are already defined by the joint pre-detection step. The front's connectivity is checked during the process as an edge is connected to at most two faces. An edge connected to two faces is called a surface edge. If the front's edges are all connected to two faces the domain is completely meshed.

The meshing process described previously is applied to the first domain $D_A$. The shared nodes are marked, and the new shared edges of the contact area are projected. The new shared edges are new constraints for the meshing process of the second domain $D_B$. This is sufficient to have identically meshed interfaces.

This meshing process fails in meshing interfaces with multiple separate closed contours. That's a single interface formed by multiple domains to re-mesh. Only one domain would be meshed in this case because the initial front will be within one of the closed contours (Fig. 15).

To improve the front propagation process the approach has been slightly modified and it is now appropriate for the case of meshing separate contact areas of a part (Fig. 15). The front contains all the edges of the domains. The edges are boundary edges and internal edges.

An arbitrary edge of the front is chosen and a first face is created with the closest "eligible" node (Fig. 16a, b). The process is iterated for all the front edges. The edges of the new faces are added to the front if they are connected to only one face. The surface edges are removed from the new front. The process stops when there are no more edges connected to only one face in the front. The set of faces produced covers the entire domain (Fig. 16c).
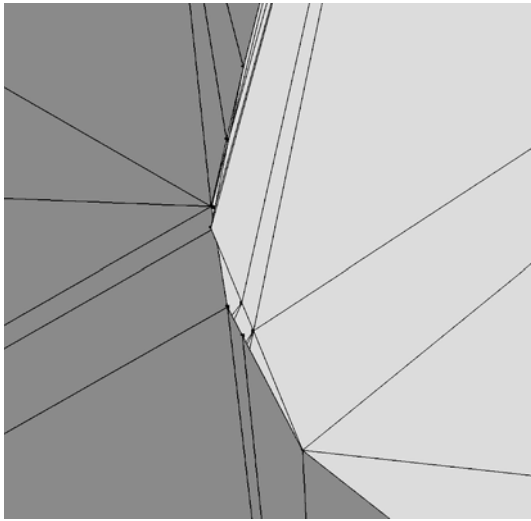
**Fig. 17** Maintaining geometry of the parts after re-meshing, gap created by tessellations and maintained after re-meshing
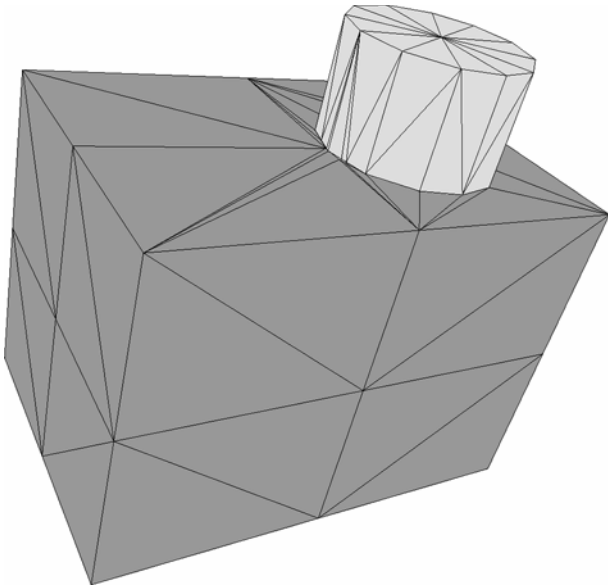


**Fig. 18** Re-meshed assembly

The "shared" edges as constraints for the meshing process ensure that the meshes are similar in the contact areas.

This extension of the front propagation process solves the problem of distinct contact areas and handles all the cases of domain re-meshing. The geometry is preserved after processing the assembly, as the clearance distance. Even gaps created by tessellation of the CAD models are not changed during the process Fig. 17. The resulting re-meshed assembly (Fig. 18) has the same contact interfaces on both parts (Fig. 19), and a contact line can be easily extracted.

## 5 Polyhedral parts checking and fixing

The goal of this step is to eliminate mesh entities that do not give further information on the shape of the part. A part produced by tessellation of a parametric CAD part model is often inconsistent. Indeed, the size of faces is often not uniformly distributed and small or thin faces appear on cylindrical shapes, or on some "virtual edges" of the parts because of the patch decomposition of initial B-Rep parametric models. "Slivers" also appear in complex shaped parts. Figure 20 illustrates these degenerate faces on the cylinder; they are useless for local geometry description. Calculating the normal, the angles or the surface of a small or very thin face is the cause of numerical instability and that should be avoided.

A very simple approach for inconsistent face elimination has been used. The edges length is checked relatively to a given limit length value. This limit value is computed as the weighted average edge length of a polyhedral part. In this way, the limit value is always dependant of the object sizes. The weight ensures that no major modification of the shape is done, a value of 10% has been chosen. An edge collapse operator is then used to delete inconsistent (i.e., relatively small) edges that are not significant for the description of the smallest shape detail of the part. A candidate edge can be collapsed to one of its extremity vertices, or to an intermediate arbitrary vertex. Figure 21 illustrates the edge collapse operator used. An edge AB is candidate for collapsing. It can be collapsed to a midpoint C or to the extremity vertices A or B.

Because of the shortness of the candidate edge, collapsing to an endpoint or to a midpoint produces very slightly different meshes. Thus we have chosen to collapse to an arbitrary endpoint.

In the illustration Fig. 22, multiple edge collapse to midpoint lead to intersecting faces between the parts of the assembly. These intersecting faces can already exist or not in the initial assembly. The contact detection process handles intersecting faces in the assembly.

This pre-processing step produces an appropriate assembly and ensures the numerical robustness of the further contact detection step.

The user can choose to improve the parts before or after re-meshing contacts. The process can be skipped in case of parts with small geometry. If the modifications generate intersections between parts the contact detection process will handle it, because the contacts are determined from pairs of faces in contact or intersection.

## 6 Envelope creation

A direct application of our interface re-meshing process is the envelope generation. For visualization purposes, it is necessary to remove as many faces as possible to allow the interactive handling of a large virtual mock up. The

**Fig. 19** Parts before and after Re-meshing, new tessellations take the boundary of the contact into account
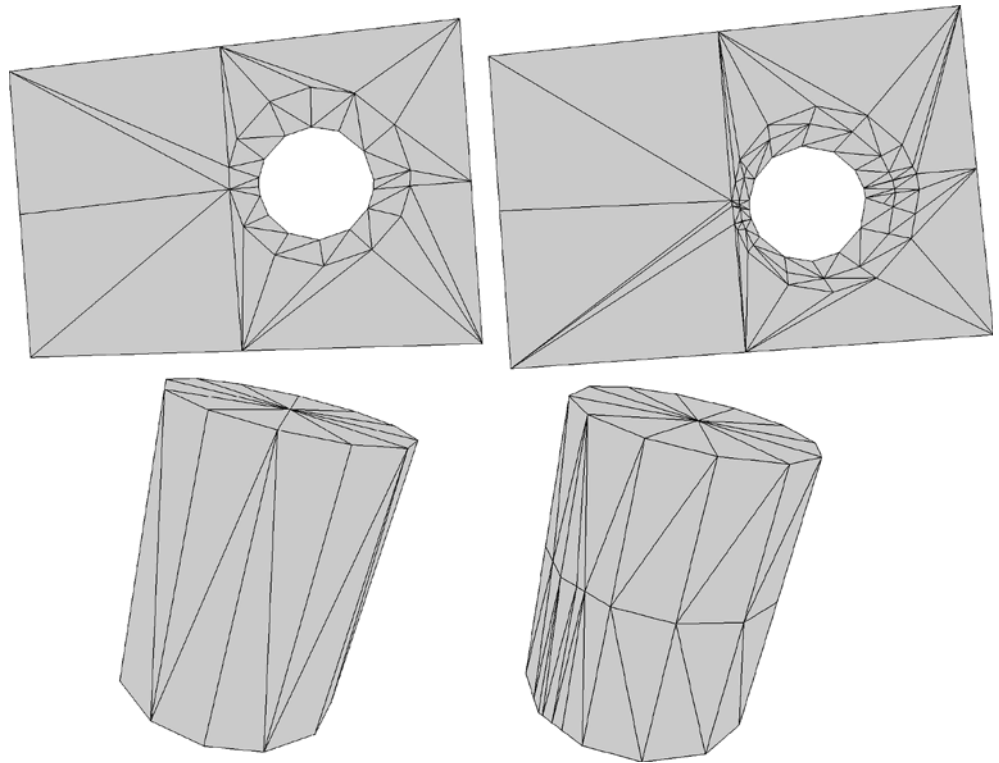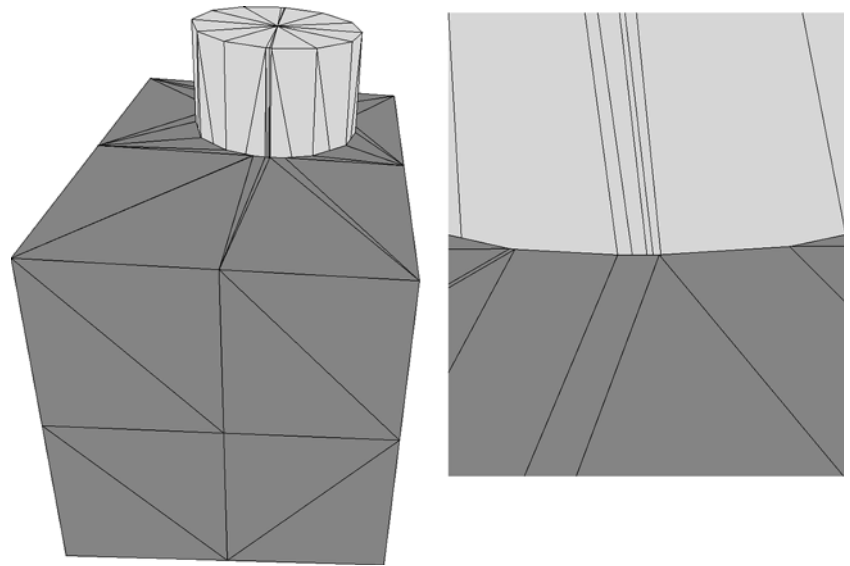


**Fig. 20** Example of a polyhedral assembly translated by the I-DEAS CAD software

removal of faces, that are not visible, helps saving time and memory space for the visualization, analysis and handling of complex assemblies containing a large number of parts. That's envelope creation's purpose.

Envelope creation is done by merging assemblies, but as we have extracted contacts interfaces, they are not visible and they can be removed. The envelope creation process first removes contact interfaces as shown in Fig. 23. Then nodes of the contact lines can be at a tolerance distance from each other, this creates gaps in the envelope (Fig. 24). For every pair of nodes in

contact, a midpoint is constructed, and the nodes are merged to this midpoint. The envelope is then created by merging the modified parts: this creates a new envelope mesh (Fig. 25) without gaps.

## 7 Conclusion

The extraction of specific information from a polyhedral assembly is our purpose. A treatment of the assembly is necessary; the need of setting up conformity in the
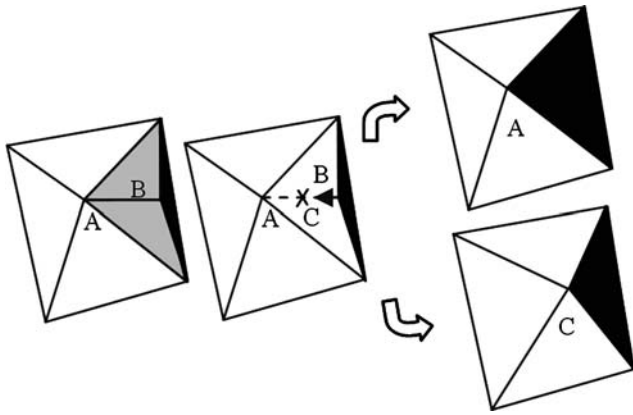
**Fig. 21** The edge collapse operator used on small length edges

assembly meshing is the first step. The second step is the contact detection, introducing tangency handling and clearance handling. Contact detection is here handled in

a static environment, and from a mechanical point of view. A joint identification is enabled between the linked parts. The type of joint is partially defined at this step, as a difference is made between linear, planar, or non-planar joints. The last step was to modify contact areas to get the same triangulation of the contact on the respective objects. The process can be applied to assemblies of convex and concave parts. The processed assembly is used for various applications. The first application we explore is creating assemblies' envelopes, useful for virtual mock-up manipulation, especially complex assemblies. This application is the first experimented in a large range of future works. It is also a contribution to data exchange of assemblies from polyhedral models transmitted to calculation software with respect to contacts.

Future works are focused on mesh decimation of assemblies or envelopes, with constraints on contact areas or contact nodes. This will contribute to assemblies' simplification problems and data reduction

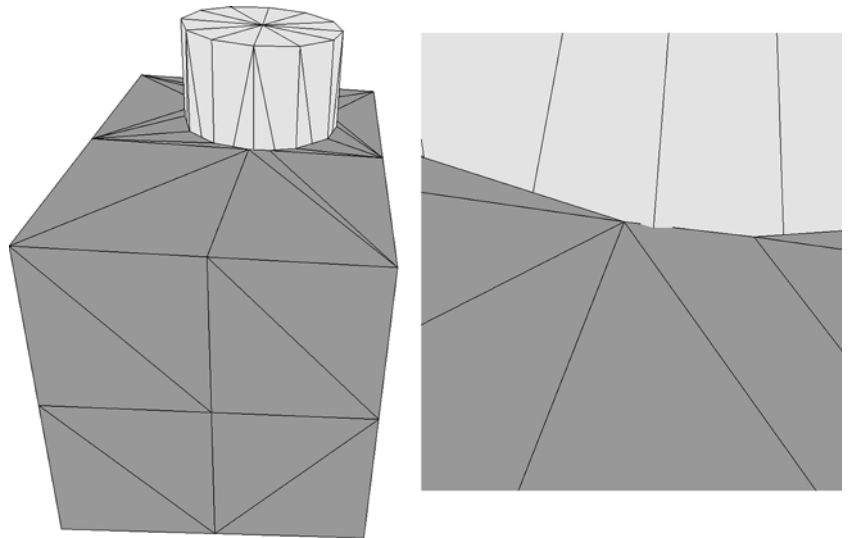**Fig. 22** Example of polyhedral assembly after small length edge removal



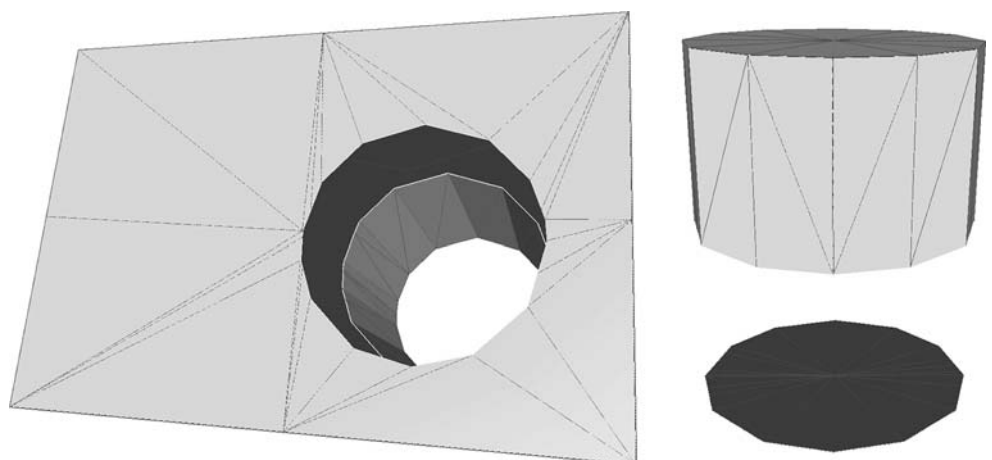**Fig. 23** Parts of the assembly with interfaces deleted before the envelope creation

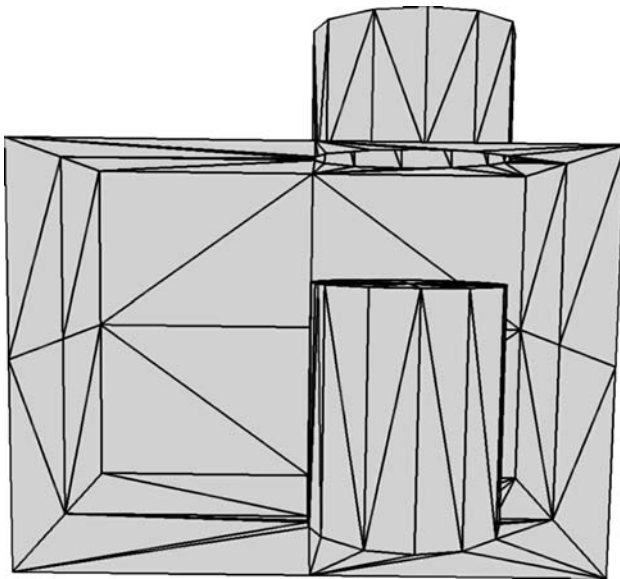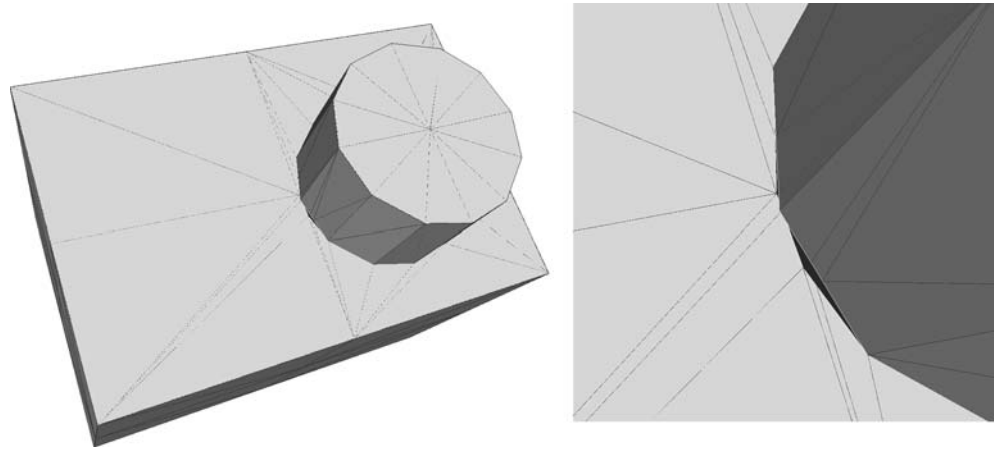**Fig. 24** Gaps in parts before envelope creation



**Fig. 25** Section of the envelope of the assembly

encountered in virtual mock-up manipulation or structure dimensioning software.

## References

1. Beall MW, Walsh J, Shephard MS (2003) Accessing CAD geometry for mesh generation. In: Proceedings of 12th international meshing roundtable sandia national laboratories, pp 33–42
2. Bern M, Plassman P (2000) Mesh generation. In: Joerg and Sack (eds) Elsevier, Amsterdam
3. Chouadria R, Veron P (2003) Identification des contacts dans un assemblage de modèles polyédriques. In: Proceedings of CPI 2003
4. Jimenez P, Thomas F, Torras C (2001) 3D collision: a survey. Comput Graphics 25:269–285
5. Kitamura Y, Smith A, Takemura H, Kishino F (1998) A real-time algorithm for accurate collision detection for deformable polyhedral objects. Presence 7(1):36–52
6. Lin MC (1998) Collision detection geometric models: a survey. In: Proceedings of IMA conference on Mathematics of surfaces
7. Lohner R, Parikh P, Gumbert C (1988) Interactive generation of unstructured grid for three dimensional problems. Numerical grid generation in computational fluid mechanics '88, Pineridge, Swansea, pp 687–697
8. Mc Leod P (2001) The availability and capabilities of low-end virtual modelling (prototyping) products to enable designers and engineers to prove concept early in the design cycle'', Prime Faraday Watch, November 2001
9. Owen S (1998) A survey of unstructured mesh generation technology. In: Proceedings of 7th international meshing roundtable, Dearborn, MI
10. George PL, Frey PJ (1999) Maillages application aux éléments finis. Ed Hermès
11. Goldstein BLM, Kemmerer SJ, Parks CH (1998) A brief history of early product data exchange standards. National Institute of Technology
12. Pebay PP, Frey PJ (1998) Delaunay-admissibilité des triangulations de surfaces. Analyse numérique, CR Acad Sciences de Paris, pp 313–318
13. Rock SJ, Wozny MJ (1991) A flexible file format for solid freeform fabrication. In: Proceedings of solid freeform fabrication symposium, pp 1–12
14. Shephard MS, Georges MK (1991) Three-dimensional mesh generation by finite octree technique. Int J Numerical Methods Eng 32:709–749
15. Veron P, Leon JC (1997) Static polyhedron simplification using an error measurement criterion. Comput Aided Des 29(4):287–298
16. Veron P, Leon JC (1998) Shape preserving polyhedral simplification with bounded error. Comput Graphics 22(5):565–585
17. Watson DF (1981) Computing the delaunay tessellation with application to voronoi polytopes. Comput J 24(2):167–172