

Automatic pre-mesh CAD data repairing

Robert Bronsart, Desta M. Edessa, Lutz Kleinsorge

Department of Mechanical Engineering and Marine Technology, Rostock, Germany

Email address:

robert.bronsart@uni-rostock.de (R. Bronsart), desta.edessa2@uni-rostock.de (D. M. Edessa),

lutz.kleinsorge@uni-rostock.de (L. Kleinsorge)

To cite this article:

Robert Bronsart, Desta M. Edessa, Lutz Kleinsorge. Automatic Pre-Mesh CAD Data Repairing, *International Journal of Mechanical Engineering and Applications*, Vol. 1, No. 1, 2013, pp. 1-9. doi: 10.11648/j.ijmea.20130101.11

Abstract: Computer aided design (CAD) models are the starting point for many downstream applications such as mesh generation, structural/fluid/thermal analysis, rapid prototyping, numerical controlled machining, casting, computer graphics. Each of these downstream applications are strongly dependent on the accuracy and consistency of the input geometry, but due to numerical problems, imprecise design, software idiosyncrasies, or data exchange issues, the surface patches produced at the CAD step may abut within unpredictable tolerances, resulting in gaps, cracks, holes, overlaps, T-connections, invalid topology and inconsistent orientation which resulted in elusive automatic grid generation. In this paper, an automatic CAD processing tool based on a powerful software development platform (Open CASCADE Technology) is presented to reduce the amount of time and cost associated with cleaning/repairing CAD geometric data for grid generation. Geometries are read from IGES file format, and translated to an internal representation for processing (shape fix followed by sewing) and finally exported in STL, STEP and IGES file formats. The tool automatically detects and heals commonly found geometrical and topological errors. As the main target is automatically repairing CAD data format errors prior to mesh generation, many test cases are performed for different kind of geometries to check the algorithm consistency and robustness. Different CAD inconsistencies are considered to check the accuracy and efficiency of the tool. It is shown that improvement in terms of time and cost can be achieved.

Keywords: Geometry, CAD, Repairing, Shape fix, Sewing

1. Introduction

The use of computer-aided design, computational analysis, and optimization in any industries with regards to complex high-performance product is inevitable. Today's commercial and open-source software tools and libraries are basics for high-performance computational fluid dynamics (CFD), computational solid dynamics (CSD), computational electromagnetic analysis, specification and verification of any product design. In addition to services as virtual laboratory (CFD, CSD, etc), they play a very important role in automatic manufacturing, inspection and assembly operations as well, which ends with comprehensive archival design database for reference; which otherwise require expensive and time consuming physical experimentation and documentation.

Modern computer aided design (CAD) systems have attained certain degree of maturity; however their efficiency, reliability, and compatibility with subsequent analysis tools remained an active research topic to-date. At the heart of this problem lie some mathematical issues, concerned with the

computation, representation, and manipulation of complex geometries that have stubbornly resisted the best efforts of the research community to formulate rigorous and efficient solution procedures [1].

CAD systems allow designers to build a shape by modeling the surface patches or polygons that comprise its boundaries. These boundaries are often represented as composite parametric surfaces, or employ a discrete representation in terms of triangular facets. Ideally the CAD software generates at least watertight (C^0 continuity) which is then stored in a format that maintains both a geometric description of each patch as well as topological connectivity between patches [2]. In addition to generating and storing geometrical and topological data, the capability of exchanging information with other systems in a large range of format; is also one the most important requirement for CAD system.

Virtually all computer-based design tasks commence with the use of CAD systems to create detailed geometrical models. These models serve as the base for diverse analysis tools, such as computational fluid dynamics (CFD), stress

analysis, geophysical data exploration, and computational electromagnetic, etc. The models are also employed in many manufacturing processes, such as numerical-control machining, injection molding, and casting. The success of such downstream applications is, of course, predicated on the receipt of geometrical models that are accurate, self-consistent, and economical in data volume [1]. The representation of CAD model includes feature based data and a resulting Boundary-Representation (B-Rep) model. The B-Rep model consists of more than just geometry and indeed one of the major problems in accessing CAD geometry has been due to an oversimplification of what constitutes a valid B-Rep model. B-Rep models contain geometry (shapes), topology (how objects are connected), and tolerances (how closely do they actually fit together). This combination of model data is then accessed by the CAD systems methods to define a valid B-Rep model. Therefore, a valid B-Rep model should be considered to consist of geometry, topology, tolerances, and methods used by the CAD system it was defined within [3, 4].

Therefore, in order to perform downstream application without problem, these CAD model constitutes should be valid for downstream application system algorithms.

2. CAD Data Inconsistencies and its Origins

Despite much work and major advances in geometric and solid modeling, practical implementations of geometric modeling operations remain error prone, and the goal of implementing correct, efficient, and robust systems for carrying them out has not yet been attained. There is agreement that the problem is serious, but what strategy has the best chance of solving it is not agreed on [5]. This statement has been made almost 20 years back, but the problem remains; as the computational capability advanced a lot for the last two decades because the problem difficulty seems to be rooted in the interaction of approximate numerical and exact symbolic data. Geometric objects belong conceptually to a continuous domain, yet they are almost always analyzed by algorithms doing discrete computation. Due to numerical problems, imprecise design, software idiosyncrasies, or data exchange issues, the surface patches produced at the CAD step may abut within unpredictable tolerances, resulting in gaps, overlaps, or intersections [6, 7].

The sources of geometrical and topological inconsistencies can be further explained looking at the two CAD data development main categories: nature and approach.

Nature happens when physical object (real-world data) is interchanged between concepts (virtual). This can be further categorized into designed (virtual) which related to basic concept abstraction and digitized (physical) which related to measurement of real-world phenomenon. Virtual design is the source for inaccuracies in the modeling process and description or representation, while digitized processes for

measurement inaccuracies and limitations.

Approach is also another source of inconsistencies happened when one form of data is changed to another (even to mesh). Some of the approaches are tessellation (which may lead to gaps, intersections, and degeneracy), reconstruction from points (topological noise, holes, gaps, and etc), solid model boundary extraction (singularities), translation from one CAD data file to another, and others more.

It is useful to categorize the inconsistencies in CAD geometries based on application specific, for example numerical simulation. Andrey [7] categorized CAD geometric inconsistencies as dependent and independent CAD errors. Independent CAD errors refers to global continuity of geometric model and further classified as geometric errors (such as non-abutting patches of a surfaces; curve self intersections; surfaces self intersections; inconsistent normal vectors between surface patches) and topological errors (such as unshared vertices of neighboring edges; unshared adjacent edges; vertices non-matching no underlying curve; unclosed loops of the face; non-lying of the boundary curve on a surface; incorrectness of face loops; overlapping faces; intersecting faces) [7]. Geoffrey and Clive [8] listed more topological and geometrical inconsistencies collectively.

On the other hand, dependent CAD error refers to badly meshed geometries and largely resulted from meshing algorithm.

This paper is mainly concerned with pre-mesh geometric inconsistencies (independent CAD errors) and their automatic healing process to sufficiently prepare for meshing algorithm.

3. Need and Challenge of CAD Repair

Many downstream applications such as mesh generation, structural/fluid/thermal analysis, rapid prototyping, numerical controlled machining, casting, computer graphics, and real-time rendering have specific requirements for the input geometric definition and representation. Hence, the successes of the downstream applications are strongly dependent on accuracy and consistency of the input geometry [9].

In most cases the problems of CAD model errors do not affect the efficiency of graphical applications, as these errors are too small to be observed visually. But, the major problems are encountered during downstream applications [7]. For instance, quality mesh is essential for engineering analysis such as CFD, FEM. Therefore, geometric data models that have been created by mesh generation software or imported from external CAD systems have to satisfy quality constraints which usually impede automatic mesh generation [10]. These quality constraints required by mesh generators usually include globally continues (C^0 continuity) and consistent representation of geometric models [2, 8]. Therefore it is necessary to adapt/repair the geometrical entities by changing their mathematical description while maintaining the same geometrical shape [10].

The research community in this field addresses the importance and difficulty of CAD repair over the years. Ken

Morgan [1] notes that, despite meshing algorithms have attained a high degree of sophistication and reliability, they need error-free geometrical input. He also showed detailed meshes (with 50 million elements) for a fighter aircraft; and characterized the meshing algorithm as essentially 100% reliable, provided the input CAD model is error-free. He presented the following "typical" breakdown of the effort in a realistic CFD analysis: 1-4 weeks for geometry repair and preparation, 10-20 minutes for surface meshing, 3-4 hours for volume meshing, and about 1 hour for the actual flow analysis [1].

Vincent et al. [9] discuss with regard to computational fluid dynamics simulation process stages (preprocessing, flow solution and post- processing of the results) . Pre-processing includes geometry cleanup and mesh generation to discretize the computational domain. They discuss the inconsistencies with regards to Initial Graphics Exchange Specification (IGES) and StereoLithography Interface Specification (STL) file formats and the difficulty to generate automatic mesh generation directly from these file formats. Therefore, they express the situation as the analyst has to manually clean the geometry to make it suitable for grid generation and this cleanup (pre-meshing) process is very time consuming, expensive and tedious task for a design/analysis engineer. They finally note that for realistic simulations, this is the single most labor-intensive task in the process, preventing true auto-meshing.

Bronsart et al. [11] discussed the situation with regards to generation of panel mesh around ship hull to calculate the wave resistance. They revealed that creating panels on the hull surface is a labor intensive none or semi automatic task requiring a profound experience based knowledge and estimated the time required to prepare geometry and generate panel mesh about 30 to 90% of the total time to calculate the wave resistance.

4. Related Works

Over the years, there are many techniques emerged by CAD, meshing and computer graphics communities to solve CAD model inconsistencies. Some have suggested generic means to represent the geometry, sometimes interfacing directly with the CAD packages themselves [12, 13]. Others attempt to deal with the representations provided by neutral file formats generated by CAD software [7, 8, 13-15]. The present work falls into the latter category.

On the other hand two approaches for healing geometries have been proposed. The one which acts on the CAD model and the other which acts on the mesh [7, 16]. Again the current work falls into the first approach which attempts to insure global continuity of geometrical data model produced by CAD systems. The latter approach can be further categorized into two: surface-based methods and volume-based methods. The first, operate directly on the input mesh, while the second, convert the mesh into a set of voxels before repair [2, 17, 18]. Bischoff and Kobbelt [18] combine the advantage of surface oriented and volumetric algo-

rithms to exploit the topological simplicity of a voxel grid to reconstruct a cleaned up surface in the vicinity of intersections and cracks, but keep the input tessellation in regions that are away from these inconsistencies. They are thus able to preserve any characteristic structure (i.e. iso-parameter or curvature lines) that might be present in the input tessellation. Their algorithm closes gaps up to a user-defined maximum diameter, resolves intersections, handles incompatible patch orientations and produces a feature-sensitive, manifold output that stays within a prescribed error-tolerance to the input model.

Many algorithms developed to clean geometric inconsistencies were tolerance driven and require users' interaction which takes significant time. Busaryev et al. [2], develop an algorithm that simultaneously repairs imperfect geometry and topology while generating Delaunay meshes. They were able to fix many errors in their user input tolerance driven algorithm.

Patel et al. [19] also develop a CAD repairing/healing algorithm, which can detect commonly found geometrical and topological issues like cracks, gaps, overlaps, intersections, T-connections, and no/invalid topology in the model, process them and build correct topological information. Their algorithm is based on iterative vertex pair contraction and expansion operations called stitching and filling, respectively, to process the model accurately. The algorithm closes small gaps/overlaps via the stitching operation and fills larger gaps by adding new faces through the filling operation.

Petersson and Chand [15] develop a tool for preparation of CAD geometries imported from IGES files and maintained in boundary representation consisting of a patchwork of trimmed and untrimmed surfaces for mesh generation. They claim that the algorithm can identify gross errors and remove automatically while a user interface provided for manipulation of geometries such as correcting invalid trimming curves or removing unwanted details.

Petersson [14] demonstrates an application code called "rap", which can suit the needs of mesh generation by cleaning up CAD geometries imported from IGES file. The topology of the model is computed and watertight surface triangulation is created on the CAD model which simplifies and speeds up the mesh generation operation.

In this paper, an automatic CAD processing tool is presented to reduce the amount of time and cost associated with geometric cleaning. The algorithm is based on the software development platform Open Computer Aided Software for Computer Aided Design and Engineering (Open CASCADE/OCC) Technology. It is an open source C++ library, consisting of thousands of classes and providing solutions in the area of 3D surface and solid modeling, visualization, data exchange and rapid application development [20]. The tool detects and heals commonly found geometrical and topological errors automatically. Geometries are read from Initial Graphics Exchange Specification (IGES) files, and translated to OpenCASCADE shape object on which geometric healing performed and finally exported to STereo-

Lithography (STL), S**T**andard for the Exchange of Product model data (STEP) and IGES file formats. As the healing processes are independent of input file format, the tool can be extended to other commonly used file formats.

5. Characteristics of IGES and STL File Formats

Most boundary representations for engineering simulations such as computational fluid dynamics (CFD) originate directly from computer-aided design (CAD) systems. One of the most important characteristics of CAD systems are their capability of exchanging information with other systems in a large range of formats. IGES and STL are the two common types of output format from CAD systems [21]. The fundamental units of data in IGES file format are entities such as points, curves, surfaces, solids and their combination. Using these entities, IGES is a mechanism for the digital exchange of database information among CAD systems. It is also common format for exchanging information with computation fluid dynamics (CFD) and finite element method (FEM) programs [22]. But, it creates a problem for the downstream application as it does not provide topological connectivity information. Therefore, in this work, IGES file format is taken as primary input data format.

STL is a triangular representation of a 3D surface and solid entities. The surfaces or solid entities are tessellated into a set of oriented triangles (facets). STL is a widely used data exchange format in the rapid prototyping industry. The generation of an STL triangulation is very efficient and can approximate most geometry very precisely [21]. There are various work addressing grid generation from STL surface representation [21, 23-27] and also many commercial and open source tools claiming to consistently generate meshes based on STL file format [28]. Though, this work targets watertight (C^0 continues) and consistent STL file as an output even though IGES and STEP files can also be generated.

6. Repair Algorithm

6.1. Open CASCADE Technology Information Model

It is essential to precisely understand the difference between geometry and topology to work with Open CASCADE Technology. Geometry is a representation of simple shapes which have mathematical descriptions (such as points, curves, and surfaces) and is used to define the actual dimensions of entities. While, topology defines the connectivity and associativity of the entities. It consists of entities as shown in Fig. 1, and are called “shapes” in Open CASCADE.

In this library, objects are topologically represented by entities such as Vertex (a point in 3D space), Edge (a linear or curved segment), Wire (a set of consequently connected edges), Face (a surfaces limited by wire(s)), Shell (a set of faces connected by their edges), and other more, see Fig. 1.

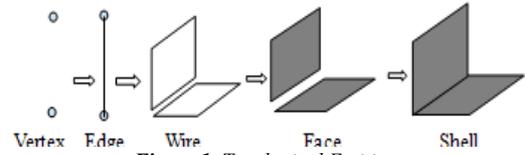


Figure 1. Topological Entities.

Therefore, one can construct, explore, visualize, transform and manipulate all geometrical and topological entities in Open CASCADE Technology. In this work, the advantages of this platform have been taken to develop a tool to translate data from one form to another, to detect and heal commonly found geometrical inconsistencies, and to finally write watertight STL file formats.

Fig. 2 shows the work flow of the algorithm.

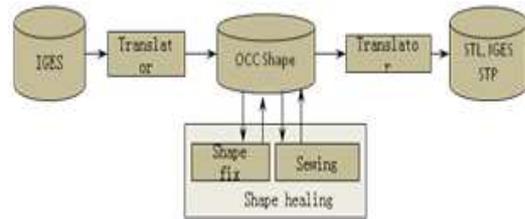


Figure 2. Algorithm main Activities.

6.2. Import and Export

Import and Export module of the algorithm includes the exchange of CAD data between different standards and OCC shape as intermediate data. In this work, CAD data are read from IGES file format and translated to OCC shape. The healed geometries are exported to STL, STP and IGES file formats. Before shape healing (shape fix and sewing), some useful parameters are extracted from the input model. These parameters help us to have a general knowledge about the input model and consists of the number of entities, the type of entities, over all dimension of the model, IGES global section information (unit, color, version, source, scale, precision), the minimum edge length in the model, shape analysis (Number of free edges and wires). Some of the above parameters are used for shape healing to perform shape fix and sewing processes automatically.

6.3. Shape Healing

Once the geometries are imported and translated to OCC shapes. The next important work package is shape healing which intended to solve commonly found geometrical and topological inconsistencies in imported CAD format. In this tool the shape healing package consists of two parts: shape fixing module which deals with intra entity errors and sewing module which intended to resolve errors between entities and used to ensure the topological connectivity of the model.

6.3.1. Shape Fixing

Due to various reasons mentioned in previous sections, many geometrical or topological inconsistencies might exist in CAD models. Intra entity inconsistencies imported from

IGES file format are repaired using shape fix module of the developed tool. In this tool different shape fixing package organized under ShapeFix classes are used to solve the problem in shapes violating Open CASCADE requirements. It is not necessary for user to detect problems before using ShapeFix because all components of ShapeFix package make an analysis of existing problems before fixing them by a corresponding tool from package of *ShapeAnalysis* and then fix the discovered problems. Shape fixing is intended to solve inconsistencies in individual topological entities (edges, wires, faces) or geometries entities (curves, surfaces). Problems with regard to faces such as: disorder of wires on the faces, face with two wires, and face with no wire are solved with *ShapeFix_Face* class.

The inconsistencies in wires are resolved with *ShapeFix_Wire* class which includes the geometric filling of gaps. This fixing and geometrical filling of gaps includes:

- Fix disordered edges in the wire (reorder).
- Fix small edges (remove edges with a length less than the given tolerance value), (degeneracy) same point with different vertices.
- Fix disconnected edges (adjacent edges having different vertices), ensure whether the end vertex of the preceding edge coincides with the start vertex of the following.
- Fix intersections of 2D curves of the edges (self-intersection of 2D curves of individual edges, intersection of 2D curves of each of the two adjacent edges, intersection of 2D curves of non-adjacent edges).
- Fix lacking edges to fill gaps in the parametrical space of a surface, or checks whether a wire is not closed in the parametrical space of the surface. The algorithm computes the gap between the edges, analyses positional relationship of the ends of these edges and (if possible) tries to insert a new edge into the gap or increases the tolerance.
- Fix gaps in 2D and 3D wires by means of geometrical filling (check gaps between the ends of 2D or 3D curves of adjacent edges).

Another shape fix class that has been integrated in our tool is *ShapeFix_FixSmallFace* which developed to drop small faces from the shape. These small faces might be spot or strip faces. If the size in one dimension of a face (strip face) is less than the given precision or if the size of a face (spot face) is less than the given precision; the algorithm remove this face.

All the above shape fixing processes are performed in one or either of the following ways.

- Increasing the tolerance of an edge or a vertex.
- Changing topology (adding/removing/replacing an edge in the wire and/or replacing the vertex in the edge).
- Changing geometry (shifting a vertex or adjusting ends of an edge curve to vertices, or re-computing a 3D curve or 2D curves of the edge).

In the tool, the tolerance values are set to IGES precision

values with some coefficients depending on the type of fixing to ensure fully automated processes. But the coefficients are also open for the user to change based on their specific use. The above shape fixing detail has been summarized from Open CASCADE documentation and interested could look for more detail [20].

6.3.2. Sewing

Usually IGES file format is preferred to convey CAD model information when it comes to complicated shapes made of NURBS surfaces (such as ship hull, car body, etc). This file might have plenty of topological problems with regards to gaps and overlaps which is a major problem for downstream mesh generator and then for numerical computation (CFD, CSD). Here comes the need to repair this inter entities topological problem to ensure downstream application easier and possible. In this work, most of the problems are solved fully automatically and depending on the fact that how corrupted IGES file is, with regards to setting tolerance value. The sewing algorithm is tolerance driven and allows the creation of connected topology (shells and wires) from (faces and edges) respectively. The algorithm does not change the geometrical representation of the shapes, but adds the information of topological connectivity. Three user defined parameters drive the sewing algorithm in this module: maximum tolerance (maximum distance between topological elements which can be sewed), minimum tolerance (size of the smallest element in shape (edge)) and manifold/non manifold. Manifold sewing of faces merges only two nearest edges belonging to different faces or one closed face with each other and already connected elements are left untouched. Non manifold sewing of faces merges all edges at a distance less than the specified tolerance. The tolerance management and manifold/non manifold setting will be discussed in the next section. In general the algorithm searches each free boundary in the shape, and then identifies the set of candidates to be merged, and selects those candidates which can be merged according to given criteria, and finally merges selected candidates to build the resulting sewed shape.

7. Algorithm Overview

The tool consists of two translator modules (IGES to OCC shape and OCC shape to STL, IGES, STEP) and two shape processing modules (shape fix and sewing) as shown in Fig. 3. The input file format is read and translated to OCC shape on which the shape processing or healing are carried out. It is important to mention that the shape healing module is independent of input file format, say IGES file, so it is easy to extend for other format as well. After geometries are translated to OCC shape, shape fixing module is the subsequent step to be performed to repair the inconsistencies exist in intra topological and geometrical entities. After commonly found inconsistencies are repaired or removed the resulting shape is kept again in OCC shape format. From which the sewing algorithm designed to repair inconsistencies between

topological elements will be performed. The sewing step of the tool is split into two parts namely first sewing (fully automated) and the second sewing (fully automated for closed shell and automated for open shell). After inconsistencies and unnecessary geometric element are removed in the shape fix step, the first sewing algorithm which attempt to repair the inconsistencies (gaps and overlaps) between patches is the next process. The algorithm considers minimum edge length greater than IGES resolution as maximum tolerance and IGES resolution as minimum tolerance. The parameters are determined immediately after IGES file format is imported. The first sewing solves problems related to gaps and overlaps between patches totally or partially without user interaction. Simple CAD models or models with big minimum edge length greater than IGES resolution are mostly repaired at this stage therefore; one can simply export the repaired geometry from this step. But, this is not always true and it needs further treatment in case of complicated models or model with very small minimum edge length. The inconsistencies that are not repaired by first sewing needs more treatment, the reason is that the gap or overlaps in that CAD model is more than the minimum edge greater than IGES resolution. To solve this, the output shapes from first sewing step pass through second sewing algorithm which is technically similar to first sewing but needs user interaction. This sewing step is categorized into two to ease and automate the sewing step, based on the input model type: second sewing for open shell and closed shell. To send the shapes to open or closed shell sewing algorithm, the user should know the input model whether it is open or closed.

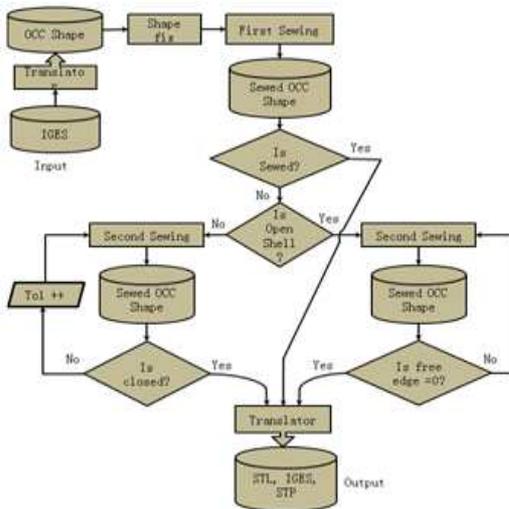


Figure 3. Algorithm Overview

7.1. Sewing for Closed Shell

For closed shells, the program works by iterating the tolerance value till the shape will be fully closed (no free edges) without any user interaction. Once every free edge is connected and no free edges remain, the iteration will stop and the algorithm will export the watertight (C^0 -continuity) and

consistent shape representation to the specified output file formats. Fig. 4 shows a box defined by six separate and disconnected faces. This corrupted box is imported and healed using the algorithm, and solved completely in the first sewing step. No action needed from second sewing step as no free edge found. The second example is a ship hull with deck, which is read in from an IGES file and healed fully automatically using the algorithm for closed shell as shown in Fig. 5. In all figures, the blue color shows the initial free edge.

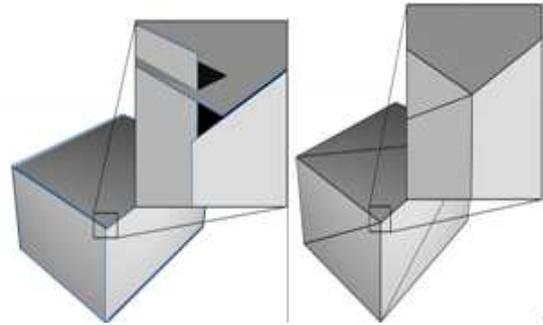


Figure 4. Box with six separate faces before and after repair

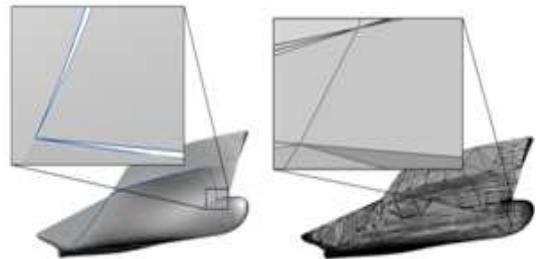


Figure 5. Ship hull form before and after repair

7.2. Sewing for Open Shell

The same algorithm and steps are followed for open shells as for closed shells. But open shells, need few user interaction if not totally healed by first sewing step. The user has to define the maximum tolerance based on the maximum tolerance used in first sewing step. So, once the tolerance is defined by user, the program repairs the gaps and overlaps automatically.

Open shells from simple surfaces to complex model, with different inconsistencies are considered to investigate the repairing accuracy and consistency of the tool.

7.2.1. Simple Surfaces

Simple surfaces with different gaps and overlaps are illustrated before and after repairing algorithm. Fig. 6-a, shows a gap between two surfaces with same gap size, and its corresponding repaired surface. Fig. 6-b illustrates the two surfaces having varying overlap and its corresponding repaired surface. Fig. 6-c, shows overlaps and gaps between three surfaces and the corresponding sewed surface. All simple surfaces with inconsistencies are considered to illustrate the working principle of the algorithm behind our tool. In the examples it is easy to understand how the algorithm

affects the original shape to repair topological inconsistencies (overlaps, gaps). The sewing algorithm first identifies the necked edges in the model, and based on the defined tolerances, select candidate to be merged. The algorithm find the midpoint between the two candidates edge and stretch the edges to midpoint to construct one common edge as shown in all Figures.

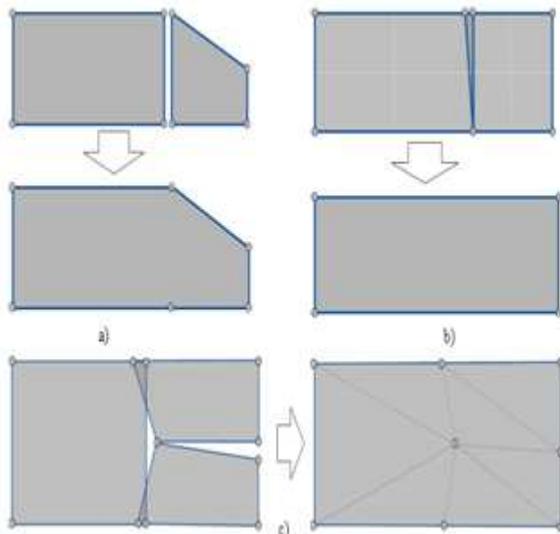


Figure 6. Different kind of gaps between surfaces before and after repairing

7.2.2. Flipped Surfaces

These examples are considered to examine how the algorithm treats if the gaps and overlaps are out of plane. Like for simple surfaces with in plane topological inconsistencies, the algorithm find the midpoint to connect the neighboring necked edges as shown in Fig. 7.

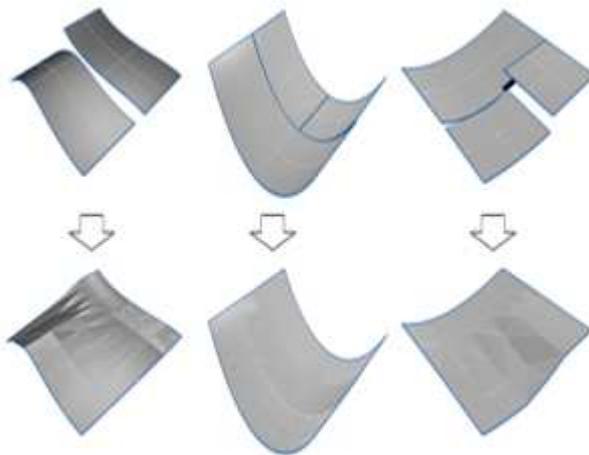


Figure 7. Flipped Surface before and after repairing

7.2.3. Complex Objects

The above simple and flipped surfaces are all repaired fully automatically as the minimum edge length greater than IGES resolution in the model is greater than the size of gaps and overlaps in the models. This might be not true in general cases, especially when it comes to complex models

as the inconsistencies in the model might be greater than the minimum edge length greater than IGES resolution. Therefore, the maximum tolerance is free for users to define till the user found the model consistently repaired and accurate. Two objects with more complexity are tested as shown in Fig.-8-a and 8-b.

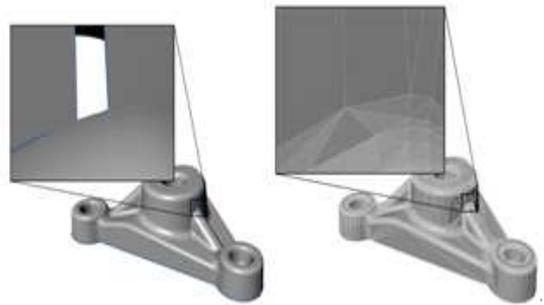


Figure 8. a. Bearing before and after repair

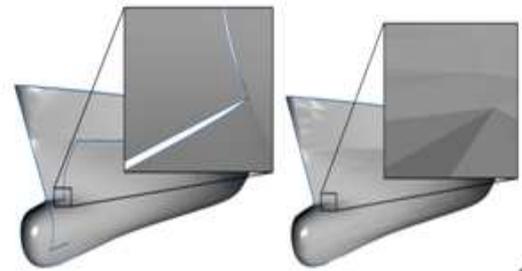


Figure 8. b. KCS Ship hull form before and after repair

8. Tolerance Management

Tolerance definition and management is very essential in any CAD systems. Topological and geometrical inconsistencies are by far related to tolerance of the model. Open CASCADE treats tolerance as a local property of the model. The three main local tolerances are vertex, edge, and face. The geometrical meaning of vertex tolerance is a sphere with radius centered in vertex's point. This sphere must encompass curves end of all edges connected at that point. Edge's tolerance is a maximum deviation between its 3D curve and any other representation. Thus it is a radius of a pipe that goes along its 3D curve and encompass curve restored from all representations. Likewise face tolerance is a thickness of a pie surrounding the surfaces. These individual tolerances setting allow specifying local inconsistencies while leaving the rest of the model well defined.

In some cases, it is not trivial to fully automate the procedure to completely ride off all consistencies without any user interaction. Because, defining too small maximum tolerance would leave too many disconnected faces, and specifying too big upfront would connect too distant faces which might ended with missing faces, holes or even completely damaged model. Therefore it is essential to define the tolerance value carefully. In this work, only complex objects with large inconsistencies need special treatment to achieve the desired output.

9. Discussion

This work proposes an approach which repair intra and inter entity errors of a CAD model automatically. Intra entity errors implies for inconsistencies in each entities of a CAD model while inter entity errors implies for inconsistencies between entities of the CAD model. The package shape fix of the tool deals with intra entity inconsistencies and used to ensure if geometrical and topological entities inside CAD model are valid and error free. The algorithm automatically resolves errors or invalidity inside the model. The sewing package of the tool mainly deals with inter entity errors. These inter entity errors such as gaps, overlaps, T-joints and small holes are mostly the reason for unsuccessful practice of downstream application. Therefore, this work proposes to repair those errors automatically.

In addition to generating repaired, consistent and valid geometry, developing very few user interactions tool which further reduce the amount time and cost related to shape healing is the target of this work. To more reduce the user interactions and to keep the original shape of the model at the same time, the sewing package is divided into two parts. First sewing which does not need user interaction, where small gaps, overlaps, T-connections, and holes repaired fully automatically. In case not completely repaired by first sewing, the user could use the second sewing option, which again categorized into two depending on the input CAD model. This is also again to reduce the user interaction. For closed shells, the user again does not need to interact, while for open shells, the user should define the maximum tolerance accordingly, and to heal big topological inconsistencies that could not sewed in first sewing.

It is also important to mention that; user can define maximum tolerance as larger as he/she wants. But, the shape of the original geometry might be lost unnecessarily in case of too big tolerance value definition. Therefore, users should be reasonable enough for the definition of maximum tolerance value. The maximum tolerance should be equal to or slightly more than the size of gaps, overlaps or holes to repair the inconsistencies. Therefore it might require the user to define the desired tolerance values to obtain completely repaired result. The repair algorithm does not affect the model part which is already repaired while the user increases tolerance values. The first sewing result, the maximum gap/overlap/hole and the maximum tolerance for first sewing give some clue about the maximum tolerance for second sewing for open shell.

10. Conclusion

This paper presents a CAD Data repairing tool which deals with independent CAD errors to sufficiently prepare the model for downstream application such as mesh generation.

Fully automatic and automatic detection and healing of commonly found topological and geometrical inconsistencies in inter and intra entity level is achieved for 2D and 3D

pre-mesh CAD models. The tool could handle small errors fully automatically and large gaps, overlaps, T-connections and holes for open shells automatically and fully automatically for closed shells. The repaired CAD model is successfully exported to STL file format preserving the curvature of the original shape. The exported STL file is checked with grid generation algorithm and mesh is generated successful. It is also exported to IGES and STP file format, but in some cases the result contains gaps and overlaps when imported to other CAD systems. The developed tool is highly automated with very few user interactions and would be one of the best alternative tool for CAD model repair prior to downstream applications such as grid generation.

References

- [1] Ken Morgen, "Closing the gap between cad model and downstream application", SIAM News, Vol. 32, no. 5, Pp. 303-319, 1999.
- [2] Busaryev Oleksiy, Dey Tamal K., Levine Joshua A., "Repairing and meshing imperfect shapes with Delaunay refinement," Symposium on Solid and Physical Modeling, Pp. 25-33, 2009.
- [3] Braid, I., "A history of geometric modeling," Spatial Tech-Ex, Pp. 1-1 – 1-17, 1991.
- [4] Beall M.W., Walsh J., Shephard M.S., "Accessing CAD geometry for mesh generation," in: 12th International Meshing Roundtable, Sandia National Laboratories, SAND-2003-3030P, Pp. 33-42, 2003.
- [5] Christoph M. Hoffmann, "Geometric and solid modeling: an introduction," The Morgan Kaufmann Series in Computer Graphics and Geometric Modeling, July 1989.
- [6] Gerteisen E.A., Hilbrink N., Mezentsev A. A., Woehler T., "CAD repair for finite element type simulation methods," Swiss CAD/CAM conference, Neuchatel University, Switzerland, Pp. 266 – 276, February 1999.
- [7] Butlin Geoffrey and Clive Stops, "CAD data repair," Proceedings of the 5th International Meshing Roundtable, Sandia National Laboratories, Pp.7-12, October 1996.
- [8] Vincent Francois, Jean Christophe Cuilliere and Michel Gueury, "Automatic meshing and remeshing in the simultaneous engineering context," Research in Engineering Design Vol. 11, Pp. 55–66, 1999.
- [9] Wang D., Hassan O., Morgan K., Weatheril N., "Enhanced remeshing from stl with applications to surface grid generation," Commun. Numer. Meth. Engng, Vol. 23, Pp. 227-239, 2007.
- [10] Bronsart R., Knieling G., Zimmermann M., "Automatic generation of a panel-based representation of ship hulls for wave resistance calculations," Proceedings, PRADS 2004, Schiffbautechnische Gesellschaft, Hamburg, 2004.
- [11] TAUTGES T. J., "The common geometry module (CGM): A generic extensible geometry interface," in Proceedings, 9th International Meshing Round Table, 2000.
- [12] Merazzi S., Gerteisen E. A., and Mezentsev A., "A generic

- CAD mesh interface,” In Proceedings, 9th International Meshing Round Table, 2000.
- [13] Stienbrenner J. P., Wymann N. J., and Chawner J. R., “Fast surface meshing on imperfect CAD models,” in Proceedings, 9th International Meshing Round Table, 2000.
- [14] Petersson, N. A., “A software demonstration of 'rap': preparing cad geometries for overlapping grid generation.” Proceedings of the 8th International Conference on Numerical Grid Generation in Computational Field Simulations, UCRL-JC-147260, 2002.
- [15] Petersson, N. Anders and Kyle K. Chand, “Detecting translation errors in cad surfaces and preparing geometries for mesh generation,” Proceedings, 10th International Meshing Roundtable, Sandia National Laboratories, pp.363-371, October 7-10, 2001.
- [16] Piret C., Remacle J.-F. and Marchandise E., “Mesh and CAD repair based on parameterizations with radial basis functions,” Proceedings of the 20th International Meshing Roundtable 2012, Part 6, 419-436, DOI: 10.1007/978-3-642-24734-7_23, 2012.
- [17] Hétroy Franck, Rey Stéphanie, Andújar Carlos, Brunet Pere, Vinacua Alvar, “Mesh repair with topology control,” Rapport de recherche, INRIA, numéro6535.France,may 2008.
- [18] Bischoff Stephan and Kobbelt Leif, “Structure preserving cad model repair,” Comput. Graph. Forum, 24(3):527–536, 2005.
- [19] Patel Paresh S., Marcum David L. and Remotigue Michael G., “Automatic CAD model topology generation,” International Journal for Numerical Methods in Fluids, 52:823–841, 2006.
- [20] OpenCASCADE Website- <http://www.opencascade.org/>
- [21] Desheng Wang, Oubay Hassan, Kenneth Morgan and Nigel Weatherill, “Enhanced remeshing from STL files with applications to surface grid generation,” Communication in Numerical Methods in Engineering, 23:227–239, 2007.
- [22] Perez-Arribas F., Suarez-Suarez J.A., Fernandez-Jambrina L., “Automatic surface modelling of a ship hull,” Journal of Computer-Aided Design, Volume 38 Issue 6, pp 584-594, June 2006.
- [23] Ito Y, Nakahashi K., “Surface triangulation for polygonal models based on CAD data,” International Journal for Numerical Methods in Fluids, 39:75–96, 2002.
- [24] “Stereolithography Interface Specification” 3D Systems, Inc.: CA, 1989.
- [25] Nakahashi K, Sharov D., “Direct surface triangulation using the advancing front method,” AIAA Paper 95-1686-CP, 1995.
- [26] Ito Y, Nakahashi K. “Direct surface triangulation using stereolithography (STL) data,” AIAA Paper 2000-0924, 2000.
- [27] Bechet C, Cuilliere JC, Trochu F., “Generation of a finite element MESH from stereolithography (STL) files” Computer Aided Design; 34:1–17, 2002.
- [28] Mesh Repair Website- <http://www.meshrepair.org/>