Automatic Mesh Repair and Optimisation

For

Quality Mesh Generation

CHONG CHIET SING

NATIONAL UNIVERSITY OF SINGAPORE

2007

Automatic Mesh Repair and Optimisation

For

Quality Mesh Generation

CHONG CHIET SING

(B.Eng.(Hons.), M.Eng., NUS)

A THESIS SUBMITTED FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY IN ENGINEERING

DEPARTMENT OF MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2007

Summary

It has been accepted by many researchers that modification of a model is often a necessity as a precursor to effective mesh generation. Imperfect CAD and data-scanned models are very common in model preparations and translations. However, editing the geometry directly is often found to be cumbersome, tedious and expensive. The novelty of the work presented in this thesis is the development of mesh repair and optimisation processes, which simplify the problems of the imperfect models and enables one to deal with simple polygons rather than complex surface representations. The present work describes the development of tools and algorithms which automatically turn invalid or defective models into valid meshed models. At the same time, these meshed models are optimized in term of geometrical fidelity and mesh quality so as to make them suitable for accurate analysis and visualization purposes. The work in the thesis is made up of two components: the mesh repair algorithms which ensure the validity of the mesh models generated and the mesh optimisation algorithms which promise quality meshed models.

The first contribution in this thesis is the development of mesh repair solution that automatically rectifies common geometrical and topological errors that are inherent in the processes of CAD modelling and simulation. A problem

detection and identification module is developed which helps users to automatically identify problems and errors in their models, instead of discovering these problems and errors at a later stage. The mesh repair algorithms also replace traditional complex geometry repair processes with a novel but simplistic mesh repair technique to create water-tight models that suit the meshing needs for finite element analysis. These algorithms are generic and can be applied to many types and formats of CAD/CAE models.

The second contribution is to present a novel hole-filling algorithm that fills holes of any arbitrary boundaries in an oriented manifold mesh and ensures water-tightness, due to missing surface patches in both 3D surface models and faceted models. The key feature of this algorithm is the capability to approximate the missing shape or geometry over the significantly complex and large holes. To cater for complex geometrical configurations, a Genetic Algorithm coupled with Rough Set Theory is developed for the purpose of optimal triangulation based on a global minimization of dihedral angles. A quartic Bézier surface interpolation is then performed over the optimal initial triangulation to approximate the shape over the hole.

One difficult task in performing research studies is to bridge research with applications. The third contribution is the discovery of two possible avenues to apply the developed techniques, and they are as follows:

1. Model Feature Suppression based on Hole Repair Algorithm
2. Restoration and reverse engineering of bio-models, artifacts, and the designing of implants in Cranioplasty

The Fourth contribution is the investigation of the use of a genetic algorithm (GA) to perform the large-scale triangular mesh optimization process. This optimization process consists of a combination of mesh reduction and mesh smoothing processes that will not only improve the speed for the computation of a 3D graphical or finite element model; it will also improve the quality of its mesh. The genetic algorithm (GA) is developed and implemented to replace the original mesh with a re-triangulation process. While retaining features is important to both visualization models and finite element models, this algorithm also optimizes the shape of the triangular elements, improve the smoothness of the mesh and perform mesh reduction based on the needs of the user.

Acknowledgements

The author wishes to express his heartfelt gratitude to his supervisors Associate Professor A. Senthil Kumar and Associate Professor Lee Heow Pueh for their invaluable guidance and support throughout the entire project life.

The author also thanks the Institute Of High Performance Computing, particularly Dr. Su Yi and Dr. Terence Hung for the help and support rendered.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Product design followed by finite element analysis is a standard design process in today's engineering world. Engineers work using different design systems and they eventually transfer data across the systems for design and analysis. During this process there is a possibility that the model is not transferred accurately and some data may be lost hindering the further design process. This research work addresses such issues by focusing on developing algorithms and techniques that automatically turns invalid or defective models into valid meshed models. At the same time, these meshed models are optimized in terms of geometrical fidelity and mesh quality so that they are suitable for accurate analysis and visualization purposes.

1.1 Bad Geometry/Mesh

There are five trends that influence the need for CAD data repair: ubiquitous CAD, improving CAD technology, legacy models, geometry based meshing, and increasing demand to re-use CAD geometry for engineering analysis. It has been accepted by many that modification of a model is often a necessity

as a precursor to effective mesh generation. However, editing the geometry directly is often found to be cumbersome, tedious and expensive.

Overview of problems in model generation in the areas of finite element modeling and visualization that are summarized from [4, 5, 8-10] are:

1. Geometrical and topological errors due to model translations, model acquisitions, model creation and handling.
2. Quality of mesh – triangle quality, mesh topology and mesh resolution (visualization)
3. Complexity – feature considerations, mesh adaptability.
4. Fidelity and accuracy – Good representations to the geometric models, able to assure high level of accuracy in analysis and visualization.
5. Speed – Time-saving in terms of human intervention and computational time.
6. Un-guaranteed Quality of Current Repair Tools.

In the design of complex parts involving free form or sculptured surfaces, the design is usually represented by a Boundary representation model.(B-Rep model). B-Rep models are often converted into the popular STL model (faceted models), that are common in areas such as the automotive industry that involves repaid prototyping or solid machining. Faceted models are also widely used in graphics and visualization. In preparing a CAD model for translation from system to system and for numerical simulation, one of the critical issues involves the rectification of geometrical and topological errors. Though visually insignificant, these errors hinder the creation of a valid finite

element model of good mesh quality. Translation errors can produce in-congruencies leading to the formations of gaps and overlaps of the mesh of the surfaces, as well as inconformity across the boundaries of the surfaces. Others prominent errors, such as missing surfaces which eventually contribute holes in the meshes, hinder the users from creating water-tight meshes which is essential for volume mesh generation. These errors occur even in data created with some of the best solid-modeling systems, such as Pro/Engineer or systems based on the Para-solid kernel developed by EDS.Unigraphics. Even straightforward processes, such as programs that produce the stereo-lithography (STL) format used by rapid prototyping systems, may fail because of cracks or slivers in the geometry. Interpretation of CAD geometry for commercial finite element software such as ANSYS would need something more than a translator program.

1.2 What is Mesh Repair

In preparing a CAD model for numerical simulation, one of the critical issues involves the rectification of geometrical and topological errors. Though visually insignificant, these errors hinder the creation of a valid finite element mode with good mesh quality. Typical geometric and topological errors are shown in Table 1.1.

Table 1.1 Typical geometrical and topological errors.

Typical Geometrical errors	Typical Topological errors
1. Cracks	**1. Missing parts / surfaces**
2. Gaps	**2. Trimmed surfaces with holes**
3. Overlapping of surfaces	**3. Duplication**
4. Sliver surfaces	**4. Inconsistent surface orientation**

The criterion of a good mesh repair in the areas of finite element modeling and visualization that are summarized from [4, 5,8-10] are as follows:

1. Able to rectify geometrical and topological errors due to model translations, model acquisitions, model creation and handling.
2. Able to produce good mesh quality with good triangle quality, water-tight mesh topology and desirable mesh resolution (visualization).
3. Able to handle highly complex models with considerations of features present in the models and provide mesh adaptability.
4. High fidelity and accuracy – the repair algorithms will ensure that the repaired models are good representations for the geometric models or for satisfying the users' desires, which lead to high level of accuracy in analysis and visualization.
5. Able to save the tons of effort and time wasted in repairing CAD models; in terms of human intervention and computational time.
6. Able to reduce human errors by making correct decision for the users during the repair processes.

1.2.1 Repairing Geometrical Errors: gaps, overlaps and T-joints

Figure 1.1(a) shows an example of a surface mesh of an aircraft. In Figure 1.1(b), the regions of in-congruency are shown by lines marking the edges which are one-manifold. These in-congruencies are caused by gaps and overlaps of the mesh of the surfaces, as well as inconformity (T-joints) across the boundaries of the surfaces.

(a) **(b)**

Figure 1.1 (a) Surface mesh of an aircraft, (b) Gaps, overlaps and non-conforming edges.

1.2.2 Repairing Topological Errors: Complex holes or missing surfaces

Triangulated models or mesh-based models are used widely to represent an object in many computer-aided engineering and modelling applications. In some cases, triangulated models are the only available representation, for example, in reverse engineering applications and bio-medical or geo-technical applications where natural processes are simulated. Using surface-fitting techniques to model these data with analytical surfaces, like NURBS (Non-

uniform Rational B-Splines) surfaces, can be extremely challenging. Moreover, hardware-based data acquisition techniques used to obtain triangulated models often result in missing entities and this is manifested as holes in the models. In addition to these problems, translation of models from different file formats or platforms often results in missing triangles or surfaces. In order to prepare a watertight triangulated model, the holes need to be patched up. When the hole occurs over a non-planar region, the underlying geometry needs to be approximated so that the filled hole conforms to the intended shape. Unfortunately, the hole-filling process is time consuming when performed manually and it is desirable to have an automatic hole-filling algorithm that can approximate the missing geometry with good fidelity.

In this work, the objective is to develop a robust and automatic technique for filling holes in triangulated models such that the underlying shape is approximated with good fidelity. This method uses Genetic Algorithm to obtain a valid and optimal initial triangulation even when the hole is geometrically and topologically too complex. The shape approximation capability is achieved by exploiting the geometric information provided by the mesh that surrounds the hole. This allows us to model the underlying shape by making use of as much localized information as possible, hence allowing varying curvatures to be modelled. A customized Advancing Front meshing is then performed over the approximated shape to generate an unstructured triangular mesh over the region. This method is not only well suited for the automatic repair of mesh models used in simulation-driven applications, but it

can also be used to restore incomplete or impaired biomedical models obtained from data-acquisition devices, such as in cranioplasty applications.

1.3 Mesh Optimisation using Biologically-inspired algorithms – Genetic Algorithms (GA)

Computer-aided design is a complex engineering activity. The design task can often be seen as an optimization problem in which the parameters or the structure describing the best quality design are sought. Genetic algorithms (GA) constitute a class of search algorithms especially suited to solving complex optimization problems. In addition to parameter optimization, genetic algorithms are also suggested for solving problems in creative design, such as combining components in a novel, creative manner.

There are two main motivations in this particular work:

1. To improve the mesh quality of a finite element model with poorly shaped elements and to reduce computational time by cutting down the number of elements of the original finite element model that are finely meshed.
2. To handle and reduce complex 3-D models which are difficult to render fast due to the large number of triangles present.

The goal of most finite element analyses (FEA) is to verify the suitability of an engineering design. The challenge is to build a sufficiently accurate model in the available time. One of the most time-consuming tasks in building a finite element model is the generation and optimization of the finite element mesh. The number of triangles or elements can be further increased or decreased

depending on the application requirement as long as the data in these meshes would not affect the accuracy for the simulation processes. Also, real-time graphics are becoming increasingly prevalent in our world. Computer games, training simulations and medical imaging all rely on interactive graphics. For the most part, complex geometric models comprising of meshes of triangles are the backbone of such systems. When digitizing a part, in order not to miss any detail of its geometry, a large number of measurement points are normally collected. Such models allow us to display arbitrary model geometry in real time, but there is a significant rendering cost in drawing all those triangles. Reducing the number of triangles in our models would allow us to render scenes faster and to render bigger and more complex scenes interactively. In fact, keeping lots of data points in planar or nearly planar region is rather unsophisticated.

1.4 Thesis Layout

The thesis is organized as follows:

In Chapter 2, a detailed literature survey was done in relation to the research work in the areas of model and mesh repair, mesh optimisation and genetic algorithms in meshing.

In Chapter 3, the research objectives are defined and focus on the to-date research drawbacks in the area of mesh repair and optimization.

In Chapter 4, an automatic mesh repair algorithm for triangular meshes with cubic curve approximation is discussed. This discussion includes the mesh-repair components and highlights several algorithms that handle typical errors such as gaps, overlaps, holes and slivers that are often present in geometrical and meshed models.

In Chapter 5, an automatic high fidelity hole-repair algorithm in meshes with shape prediction is introduced to handle holes of any arbitrary boundaries in an oriented manifold mesh that ensures water-tightness of the mesh after patching the missing surfaces. The key feature of this algorithm is the capability to approximate the missing shape or geometry over the significantly large hole.

In Chapter 6, various techniques and new potential applications that make use of the techniques and algorithms that are developed for mesh repair are introduced. This research leads to potential applications such as the development of a mesh feature suppression function and a design aid for cranioplasty: the process of restoring defects, usually holes, in the human skull.

In Chapter 7, a novel technique that involves the use of biologically-inspired algorithms, the Genetic Algorithms, in the process of mesh optimization is introduced.

In Chapter 8, two case-studies consisting of erroneous meshed models, undergoing the repair and optimization processes base on the algorithms developed in chapters 4-7, will be presented.

In Chapter 9, the contributions of the thesis are highlighted and the conclusions as well as some future work recommended to the thesis are also discussed.

Last but not least, the reference works in this thesis and the journal and conference publications arising from this thesis, which have been submitted or accepted for publications, are listed.

Chapter 2:

Literature Survey

Many models contain geometric elements that cause problems for programs and processes that employ CAD geometry. The errors may include cracks and sliver faces, poor accuracy, and loss of the relationships between edges and surfaces. Some solid models may enclose parts of themselves. Some edges may become disassociated from model faces or contain loops or knots.

These errors occur even in data created with some of the best commercial solid-modeling systems. They can be due to user error. However, poor user training or sloppiness is not the only cause of all bad data. In the course of developing a design, it is common to add or delete features or to move them around. Occasionally, a feature may be squeezed until it is nearly invisible, or two features may overlap and leave a sliver or crack between them. Such errors can easily be un-noticed. Setting model tolerances too loose can create unacceptably large gaps between model faces and edges, and modelers with very tight tolerances can cause errors by generating unrealistically small faces and edges needed to fill in the gaps between surfaces. If the CAD geometry is used for finite element analysis or translated to another CAD system via Initial

Graphics Exchange Standard Format (IGES) [1] or Standard for the Exchange of Product model data Format (STEP) [2], then seemingly insignificant errors in geometry can cause big problems. Cracks and sliver faces can cause problems in finite-element meshing, Stereo-lithography (STL) [3] output, and tool-path generation routines. That is because CAD/CAE programs do not see a whole part the way the human mind perceives it.

Presently there are many commercial software modules, such as TransMagic™ Plus, CAD Doctor by Elysium and CADfix by TranscenData, which claim to be able to perform automatic geometry repair. However, these third party software modules can only rectify common geometry problems encountered prior to a simulation session. A successful or unsuccessful outcome is possible. Thus there is yet no absolute solution for geometry/mesh repair of CAD models.

2.1 Current-State-of-the-Art on Gaps and Overlaps Repair

Most current-state-of-art handles the problem by repairing the geometry directly. Steinbrenner *et al.* [4] presented an edge merging algorithm trying to pair up common edges, allowing adjacent meshes to become computationally water-tight via their shared edge curves. However, the algorithm would not work on two edge curves that were adjacent near one end and diverged on the other, and it could only perform repairing for errors due to inconsistency in tolerances. Barequet *et al.* [5, 6] proposed a fault-repair algorithm which handled only polygonal model that converted an unordered collection of

polygons to a shared-vertex representation to help eliminate error. However, this algorithm was not robust enough as it only targeted at removing bulk errors (extraneous geometry) and small positional errors (erroneous geometry); and the algorithm might not detect large intersecting polygon. This algorithm also did not handle trimmed patches with intersection curve boundaries and general B-rep models involving non-regular arrangement of surface patches. Barequet [7] also proposed using geometric hashing to stitch gaps between surfaces.

Murali *et al.* [8] used a spatial partitioning method to define a watertight surface boundary of a model. However, his method did not seem to be able to handle geometric intersections. Peterson *et al.* [9] developed user-interactive tools for the efficient preparation of CAD geometries for mesh generation. Similarly, Morvan *et al.* [10-11] described a virtual environment that provided tools for model correction, controlled primarily by the user. These user-controlled environments proved to be too cumbersome and inefficient for large models. Errors were easily missed by the user and new errors might even be introduced and the algorithm was not robust enough. Turk *et al.* [12] focused on a topological construction method of removing overlaps of polygon by clipping them against each others in order to generate polygonal models from range data. Unfortunately, it did not consider geometric intersections and inconsistent topology that might be present. Yau *et al.* [13] presented a surface reconstruction algorithm for the global stitching of STL models. It was however not as efficient as all data points were involved in the process and a complex model might take a long time to get ‘stitched’. This method also could

not tackle complicated or small features, e.g. a fine comb shape model. Hu *et al.* [14] developed an algorithm which made use of an overlay grid method to fill holes and gaps, and remove overlapping areas. However, the repaired geometries might differ from the actual desired. Makela and Dolenc [15] used local technique for filling gaps and cracks on the surface models, but when a large number of these gaps and cracks were involved, this gap/hole filling method might cause the number of polygons that described the surface to increase tremendously. Sheng and Meier [16] used a zipping operation on small gaps between surface, which was slightly similar to the present proposed method, but they only focused on topological construction and did not consider geometric intersections.

The first positive result on slivers was an algorithm by Chew [17] that eliminated 3D slivers by adding new points to generate a uniformly dense mesh. In a recent breakthrough, Cheng *et al*. [18] showed the use of assigned weights to the points so that the weighted Delaunay triangulation was free of slivers without adding new points. Edelsbrunner *et al*. [19] removed slivers using Delaunay triangulation algorithm with a ratio property of the bounded circumradius of the triangles to its shortest edge length.

All the works mentioned above, tend to handle imperfect geometric models, turn them into “repaired” geometric models using various, different methods, and finally perform mesh generation on the models. They only address the handling of a specific area in CAD repair, such as repairing gaps or slivers removal. Repairing geometry is not an easy task, especially when one is

dealing with complex surface and volume representations, and automating the full repair process in that manner is almost impossible, not to mention the subsequent and more tedious finite element modeling. In the present work, we are looking for a fully automatic, combined solution for CAD repair and finite element model generation. Some works under Leon *et al*. [20-22] and Noel *et al*. [23-25] focused on idealization processes of FEA model such as surface mesh generation techniques dealing with inconsistent geometry and geometric adaptation performed on polyhedral representations of the models, and attempt to generate a valid finite element triangulation directly from `dirty' geometry. These works are similar to our proposed work in favour, but does not satisfy the consistency and fully automation constraints of the proposed work. Table 2.1 shows a summary of some promising current-state-of the-art approaches on gaps and overlaps repair. Most of the methods repair erroneous geometric models directly. However, the repaired geometries may differ from the actual desired.

Table 2.1 Summary of Some Current-State-of-Art on Gaps and Overlaps repair

State-of-art	Methodology	What it can do	Limitations/ Remarks
Overlay Grid Based Geometry Cleanup J. T. Hu [14]	Intersection computation with structured grid. Recreate geometry from the intersections of the hex mesh.	Ability to fill holes, gaps and remove overlapping areas.	Dependent on mesh size. Cannot tackle complex features. Cleaned geometries may differ from the actual desired.
Procedural CAD Model Edge Tolerence Negotiation for Surface Meshing JP Steinbrenner [4]	Mismatched surface meshes are rendered watertight through recursive splitting and merging of edge curves.	Ability to fill gaps and remove overlapping areas	No repair to missing parts/entities. Triangular-elements are used for surface meshes. Only perform repairing for errors due to inconsistency in tolerances.

Repair by Shifting vertices of Polygons (RSVP) G.Barequet [6]	Converts an unordered collection of polygons to a shared-vertex representation. Allow user to visualize the errors and override corrections through implementing a feedback system	Ability to close gaps and remove overlapping areas.	Algorithm is not robust enough. Only targets at removing bulk errors (extraneous geometry) and small positional errors (erroneous geometry). May not detect large intersecting polygon. (A small box lying on top of a large box always results in two separate solids). This algorithm does not handle trimmed patches with intersection curve boundaries and general B-rep models involving non-regular arrangement of surface patches
Consistent Solid and Boundary Representations from Arbitrary Polygonal Data T. Murali [8]	Determine regions of space that lie inside a solid using spatial partitioning, and use the partition as the description of solids.	This is a simple and promising technique which generates topologically-correct (valid) solids.	No control on the topology of the result, which can be significantly different from the input. Works badly in the presence of degeneracy or narrow angles between adjacent polygons. May mishandle missing polygons and add cells that do not belong to the model.
Virtual tools for model correction S. M. Morvan [10]	A virtual environment that provides tools for model correction, controlled primarily by the user.	Ability to fill holes and gaps, remove overlapping areas and do most of repairs.	User-controlled environment proves to be too cumbersome and inefficient for large models; Errors are easily missed by the user and new errors may even introduce. Not robust enough in term of automation.

Zippered Polygon Meshes G. Turk [12]	Topological construction method to remove overlaps of polygon by clipping them against each other in order to generate polygonal models from range data.	Ability to remove overlapping areas.	Does not consider geometric intersections. Inconsistent topology may be presence.
Geometric Hashing Approach	Fill gaps in the boundary of a polyhedron by adding new faces to close the gaps. It is done so by connecting points along the same or different borders.	Ability to fill holes and gaps.	When a large number of cracks is involved, simple-minded hole filling may result in an explosion of the number of polygons needed to describe the model.

2.2 Current-State-of-the-Art on Hole-Filling Techniques

Hole-filling can be performed as a pre- or post-processing operation, applied after surface reconstruction, or it can be integrated into a surface reconstruction algorithm. Holes which have generally convex boundaries and which lie over a nearly planar region can be mapped to a disc topology. In these cases, simple triangulation algorithms can be employed [26, 27] to repair the holes. However, when the holes have convoluted boundaries or when they lie over a highly non-planar region, such as sharp curves, joints or crevices, these methods will not work well [28, 29]. For such holes, having multiple boundary components, many topologies are possible; hence the problem becomes even more difficult. This problem often arises when the model is acquired via hardware scanning or when the model is reconstructed from segmentation images from medical scans.

In general, hole-filling algorithms of triangulated many triangulations, and model can be categorized into the voxel-based approach or the model-based approach. In the voxel-based approach, the polygonal model is converted into a volumetric representation consisting of discretized volumes called voxels. Signs are then assigned to the voxels representing the interior or exterior of the model. Different techniques are then used to patch up the holes by closing up the gaps in the volumetric space.

Curless and Levoy [30] employed a space carving and iso-surface extraction technique to fill holes in the volumetric grid. In the work by Davis *et al.* [31], gaps in the volumetric space were filled using volumetric diffusion. The dual contouring technique recently proposed by Ju [32] had the advantage of modeling sharp features in the original model.

In the model-based approach, holes are patched by working directly on the triangles. Holes with regular boundaries over a relatively planar region can be easily patched as demonstrated by [26, 27]. However, over irregular geometry, the underlying shape has to be estimated. To address this issue, Carr *et al.* [33] used a surface interpolation technique which fitted the depth-maps of a surface with radial basis functions. The advantage of using radial basis functions was that holes with irregular geometry could be handled with fewer restrictions. This method worked well with convex surfaces but difficulty arised when the underlying surface was too complex to be described by a single-valued function. Also, this method currently applied only to regular rectangular grids and it was not clear how the method could be extended to

unstructured triangular meshes. Liepa [34] employed an umbrella-operator to fair the triangulation over the hole to estimate the underlying shape.

Chui et al. [35] filled N-sided polygonal holes using an energy minimization technique. They employed a C1 piecewise cubic triangular spline functions to construct the filling surfaces. However, most hole filling researches worked well with relatively small holes with respect to the entire models, in smooth regions where curvatures' values were low with minimum convolutions. Jun[36] used a robust piecewise technique to perform hole partitioning. This method worked well with large holes, but when there were a lot of convolutions in the hole, his method has difficulties in generating consistent filling surfaces that match and blend well with the regions at a hole.

2.3 Current-State-of-the-Art on Meshing Algorithms using Genetic Algorithms (GA)

As a result of their global optimization property [48, 49], Genetic Algorithms (GAs) have been widely used in various fields such as state space search, nonlinear optimization, machine learning, traveling salesman problems, etc. [50]. Both theoretical [51, 52] and experimental studies [53, 54] show that the genetic technique is an efficient and robust heuristic for search in complex spaces solving complex optimization and combinatory optimization problems. In recent years, preliminary study on GAs in triangulation has been reported. Absaloms and Tomikawa proposed a GA to triangulate two adjacent contour data from a digitized geographical map [55]. They claimed that GA based triangulation was a relatively simple technique and could be

implemented by parallel processing. A GA based method for simple 2D triangulation was also reported recently [56]. In the report, the GA based triangulation was compared to the result from greedy triangulation. It was concluded that GA based method would lead to better optimization results.

Genetic algorithms were first developed by John Holland at the University of Michigan in the mid 70s and were the subjects of much research today. G.A.s are robust and, although they may not find a perfect solution, they come close enough for most engineering work for a wide verity of problems. Multimodal and highly discontinuous problems are taken in stride by G.A.s. There are numerous variations on G.A.s including different types of crossover algorithms, hybrids combining G.A.s with fuzzy logic, simulated annealing and neural networks. Hamann proposed a data reduction scheme for the triangulated surface [57]. He removed a triangle based on the curvatures at the three vertices. A user could specify a percentage of triangles to be removed. His research had smooth surface fitting in mind. Hoppe et al. [58] used an energy function to represent the trade-off between geometric fit and compact representation. A user desired parameter was used to control the trade-off between geometric fit and compact representation. A large value indicated that a sparse representation was strongly preferred, but at the expense of degrading the fit. A merging algorithm based on edge collapse was proposed for automatically computing the approximations of a given triangulated object at different levels of detail [59]. Edges were queued according to their cost functions, which indicated the error caused by edge collapse. Approximation levels were controlled by prescribing geometric tolerances. Gieng et al. [60]

classified mesh simplification algorithms into three types: removing vertices, removing edges and removing faces.

2.4 To-date Research Drawbacks on Mesh Repair and Optimization

The limitations of current research work on mesh repair and optimisation are summarized as follows:

1. So far, there is no noteworthy attempt to automate the processes of various model repairs as a single process that handles all the common errors that are present in defective models.
2. Some researches repair model in their geometrical forms. This approach is very challenging but demanding. It is difficult to manipulate and modify geometric entities like curves and surfaces in a model.
3. Defective models contain many missing information. Many researches in the area of model repair put emphasis on obtaining valid and usable models suitable for analysis. They do not focus on the quality of the repaired models in terms of geometric fidelity to the original or user-intended shape of the model, for example, how to shape a large hole when being filled or, how the intersection edges will be like when closing gaps in between disconnected surfaces.
4. Many researches have been carried out in the area of optimizations using biologically–inspired algorithms such as the Genetic Algorithms (G.A.). However, most mesh-related processes favour

heuristic techniques coupled with brute force computing methods, due to the advancement in computing technologies. Little exploration was done in the meshing processes such as mesh coarsening, refinement, smoothing and optimization in mesh quality, using such algorithms. Since the repaired models will eventually be served as visualization or analysis models, it would be interesting to investigate if biologically-inspired algorithms, such as G.A., can bring novelty to mesh processing.

Chapter 3:

Research Objectives

3.1 Research Objectives and Approaches

This research explores methodologies to reduce design cycle time at the pre-processing stages of the simulation process. It aims to *reduce human intervention* via replacement with intelligent and automatic algorithms for polygonal model repair, modification and mesh generation. This new approach holds the promise of higher fidelity levels, higher automation levels, speed and robustness when compared to more traditional interactive cleanup methodologies.

As such, the research objectives of this thesis are listed as follows:

1. To attempt to automate the processes of various model repairs as a single process that handles all the common errors that are present in defective models, so as to reduce human effort and computational costs.

2. To put emphasis on mesh repair and to deny model repair in their geometrical forms. This approach requires all kinds of geometrical models to be converted into mesh forms to under go the automatic mesh-repair process.
3. Although defective models contain limited information. This research believes that the quality and fidelity of a computational model is as importance as its validity and usability. Attempts will be made to perform mesh repair with shape predictions to create good representations for the geometric models or for satisfying the users' desires, which lead to high level of accuracy in analysis and visualization.
4. Meshes will be optimized in terms of the shapes of the models, the quality of the meshes (for analysis models) and the resolution of the meshes (for visualization models). Novel attempts using Genetic Algorithms to optimize meshes will be made.

Often, inexperienced users may not even able to decide how the defective models are to be repaired. They may also not able to visualize how the repaired models will appear to be like. As shown in Figure 3.1, one key contribution of this research is not only to ensure validity of the repaired models, but also to ensure quality in the repaired models by repairing them as close as the actual desired, through making intelligent decisions based on heuristics with suitable shape approximations, geometrical interpolations and optimizations.

Figure 3.2 shows the proposed automatic operation sequence of the repair process for typical models such as solid and surface models, faceted models and finite element mesh models. The data will be interpreted to identify problems, followed by a series of repair algorithms to tackle each individual error. The final product will be a 2D or 3D finite element model with perfect topology and quality mesh.

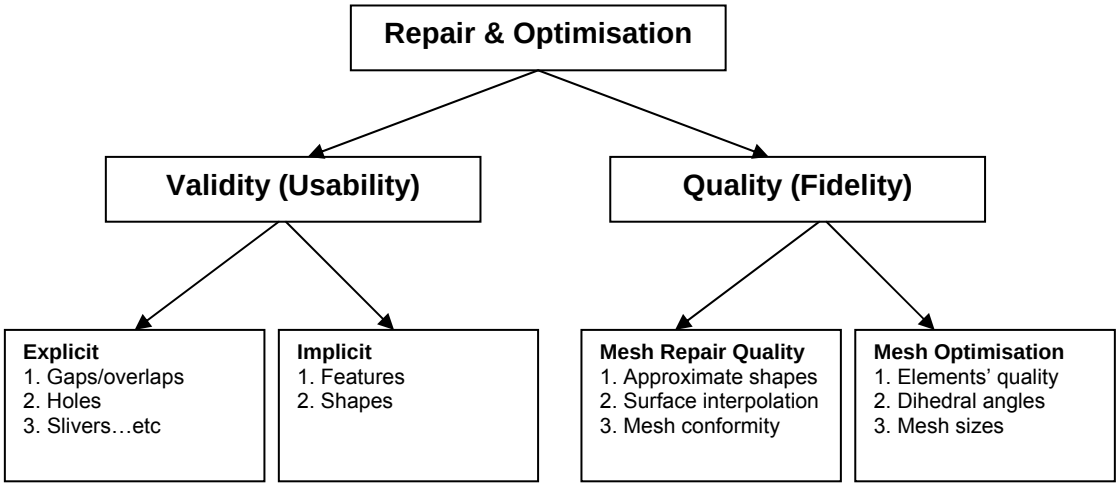


Figure 3.1 The components of a mesh repair and optimization system.

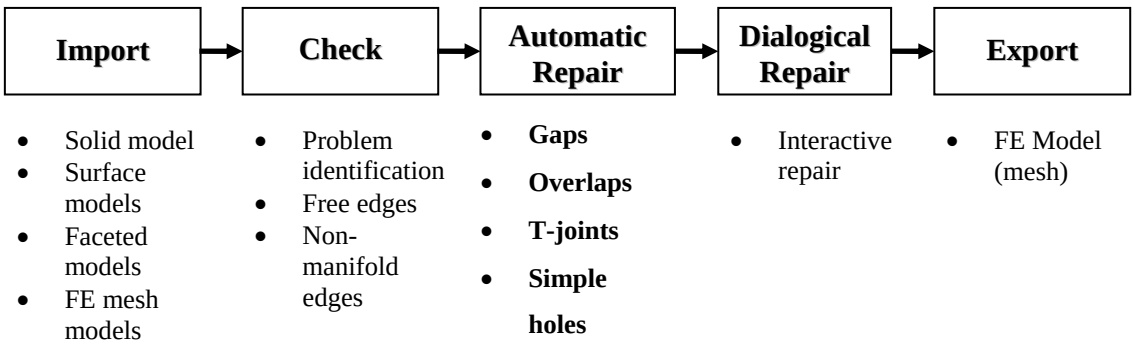


Figure 3.2 Proposed automatic repair operating sequence

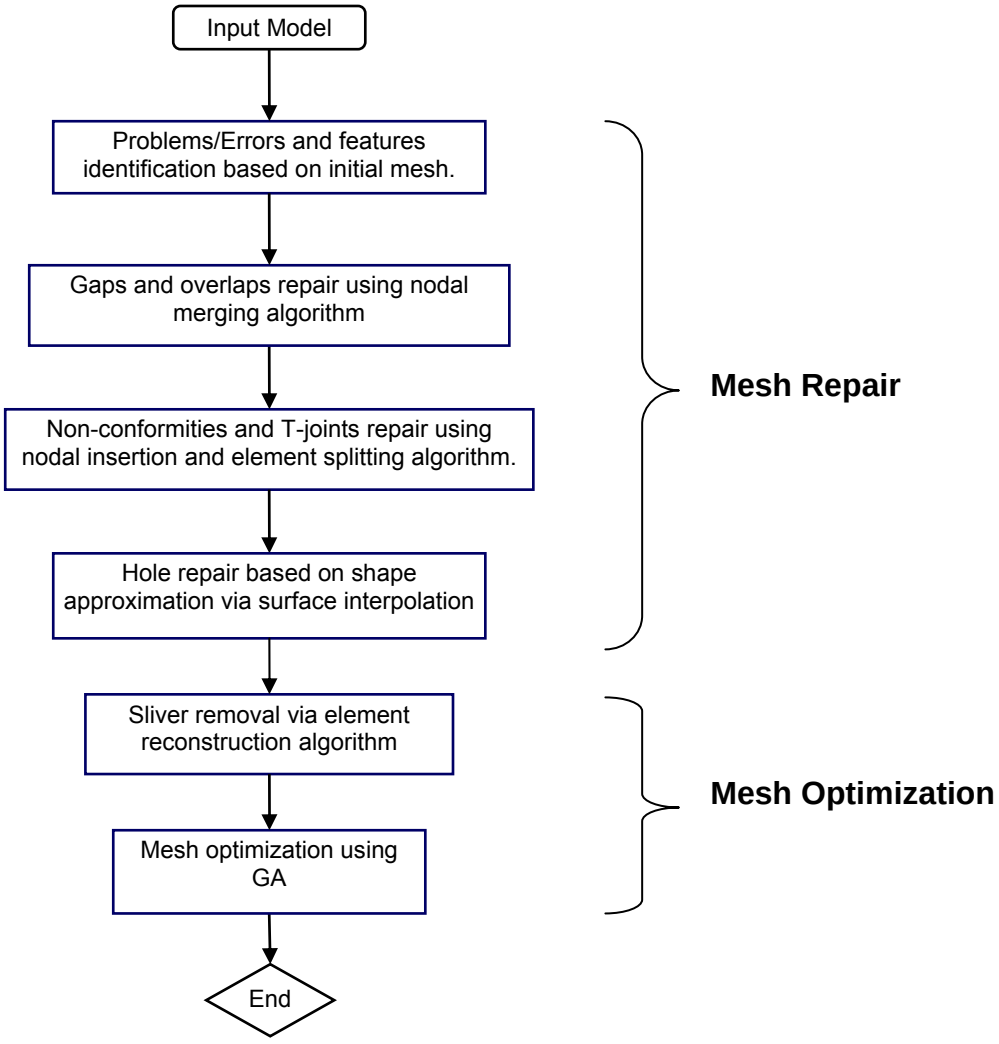


Figure 3.3 The proposed model repair routine that repair and optimize a mesh model.

The flow of the mesh repair and optimization algorithm is illustrated in Figure 3.3. The complete solution will consist of two components, namely the mesh repair component and the mesh optimization component. The first module of mesh repair component is an error identification module which is important, as it helps the users to check and identify geometrical and topological errors that are often not easily noticeable. The rest of the modules of the mesh repair component are different mesh repair techniques which are developed to handle different kinds of errors in the models. The sequence of repairing different errors is important and it is proposed to handles the gaps and

overlaps, before repairing the holes, which are usually much bigger in size by nature. The mesh optimization component will help to further improve the quality of the mesh after it has been repaired. It will take care of degenerated elements, mesh smoothness and other meshing requirements such as mesh resolution and mesh adaptivity.

3.2 Benefit of this research

In this research, a full solution is presented to create surface meshes/polygons from imperfect models that contain errors such as holes and missing surfaces patches, and also undesired surfaces' or elements' topology. A new element based hole-filling algorithm is introduced to fill complex holes and to ensure water-tightness due to missing surface patches in both 3D surface models and faceted models. We specifically address situations in which the holes are too geometrically and topologically complex to fill using normal triangulation algorithms.

The benefits of this research include

- Rapid problem identification of geometric models before models are released thus saving the time and expense associated with model rework. Significant reduction of design cycle time through the removal of the conventional bottleneck pre-processing procedures.
- Significant reduction in manpower requirement in the design cycle through the employment of intelligent and automatic algorithms in the pre-processing stage.

- Significant reduction in human error caused as compared to traditional method of manual repair and mesh generation of highly complex geometries.
- Significant reduction of design cycle time through the full dimensional reduction or partial reduction (mixed model) of the solid model to be analyzed.
- Techniques developed in geometry and mesh repair have high potential of cross disciplinary applications, such as visualization and bio-medical modelling, etc. Some of these techniques may even evolve into useful solutions for CAD modelling applications, such as model reductions, feature preservation and suppressions.

Chapter 4:

Automatic Mesh Repair for Triangular Meshes with Cubic Curve Approximation

---

Current state-of-art algorithms handle the problem of gaps and overlaps by repairing the geometries directly. The novelty of the proposed method is that the mesh-repair process is to include model repair and mesh generation into a black-box. The mesh-repair algorithm essentially simplifies the problems of the imperfect models and allows one to deal with simple polygons rather than complex surface representations. This proposed work will address typical but hard-to-handle errors such as gaps, overlaps, T-Joints and slivers.

4.1 Proposed Methodology

A heuristic method is proposed based on the observation that most of the gaps and overlaps occur due to the file translational and numerical errors in the computation of vertex coordinates. The proposed mesh-repair algorithm enables repair to be done to the CAD model at the meshing stage. The method is well suited to repair models for mesh generation, finite element

analysis and visualization applications needs. One of the important contributions made in this research is the development of a problem identification module. This module essentially help users to automatically identify problems and errors in their models, instead of discovering these problems and errors themselves at a later stage when they have already spent days and weeks in building the computational models. The key features that identify the errors and enables full automation are described in Table 4.1.

The proposed approach aims to correct the errors of a geometric model after the process of mesh generation. The algorithm will handle the following errors:

(i) *Gaps between the surfaces*. Gaps are typically formed between adjacent surfaces where their supposedly common edges do not match. This can be due to modeling error or truncation error during the exporting and importing process.

(ii) *Overlaps of geometry*. Small overlap can be detected by checking the proximity between a pair of candidate surfaces and their edges. Similarly, overlaps arise due to modeling error or truncation error during the exporting and importing process.

(iii) *T*-joints. T-joints arise due topological errors as a result of non-congruency between surfaces that are connected to one another. The connection is such that an edge is one-manifold. This condition is unacceptable since a valid 3D model requires that it is watertight.

(iv) *Sliver surfaces*. A typical sliver surface is long and slender. In the geometrical sense, the ratio of its surface area over its perimeter is

significantly small in value. The mesh of a sliver surface tends to yield elements of very poor aspect ratio.

(v) *Simple hole*. A hole is present when a surface is missing from the CAD model. To enable the model to be water-tight, holes will need to be patched. In this chapter, typical simple holes are defined as holes which are near planar and a simple advancing front meshing technique is introduced to fill up the holes. Subsequently in the following chapter, more sophisticated algorithms are implemented when dealing with holes which are highly complex in shapes.

Table 4.1 Problem Identification Module: Detection of geometric/topological errors

	Errors or Uncertainties	**Type of CAD Errors**	**Identifying features**
1	Manifold / non-manifold surface (geometric /mesh) representations.	Geometrical	Free edges of FE elements.
2.	Small gaps and overlaps.	Geometrical	Free edges of FE elements. Proximity checks based on merging tolerance for the free nodes merging method.
3.	Large gaps and overlaps.	Geometrical	Free edges of FE elements. Gaps filling and overlaps removal methods (similar to hole-filling) .
4.	Duplications or undesired elements.	Geometrical	Any duplicated/undesired entity.(non-manifold, e.g. edge adjacent to 3 or more surfaces)
5.	Surface Orientation.	Geometrical	Must be able to identify 1 correct element orientation and the rest of the elements follow suit.
6.	Sliver surfaces or non-regular mesh.	Geometrical	Minimum angles between edges of a FE element and the tolerable minimum width value of the element.
7.	Holes.	Topological	Close loops made up of free edges for non-manifold surface representation.

8.	Missing parts / surfaces.	Topological	Irregularly large holes, etc. If the model is an originally solid model, then these surfaces will be patched using the hole-filling algorithm.

The self-repair technique developed using mesh-repair algorithm works on any meshed model or geometric models with surface definitions. Traditional model repair, except for repair of finite element models, focuses on geometric repair first before getting into mesh generation. The main contribution of this work is the replacement of traditional complex geometry repair processes with a novel but simplistic mesh repair technique. The physical geometric model will not be repaired, but instead the errors will be removed in the finite element model. Small gaps, overlaps, T-joints and holes will be tackled in this work.

Figure 4.1 shows the automatic repair operations developed in the current research. To enable fully automation, all input values used at varies steps are to be input from the start and the following steps are carried out:

1. Introduce the model. If the model is a non-meshed model, mesh the model using triangular elements via a paving algorithm. The user will input the element's edge length (or the element's size).
2. The program will search through the model for free elements' edges (free edges).
3. If any free edges are detected, it will first undergo the gap-repair algorithm which will be described in section 4.2. This will heal gaps and overlaps between the elements whose nodes are within the merging tolerance (another input value). The default merging

tolerance value in this program is a small faction of the initial input of the element's edge length.

4. If the gap's size is within the merging tolerance, and the merging distance between two nodes is larger than the merging tolerance value, a T-Joint repair algorithm is applied. It will perform splitting on the elements to ensure conformity along the common, merging edges. This algorithm is discussed in section 4.3

5. If there still exist free edges but no free edge-pair are within the merging tolerance, they will be treated as large gaps which will be filled up with elements instead of just stitching the gaps. One type of large gaps is holes. Closed loops of connecting free edges are defined as holes. Hole-filling is carried out using a triangular element-filling algorithm based on the advancing front method which will be discussed in section 4.4.

6. During the gaps and holes repair, highly skewed triangular elements are introduced and these skewed elements will be eliminated using an element reconstruction algorithm presented in section 4.5. This algorithm can also be used to eliminate the mesh over sliver surfaces and sliver holes. Finally, a mesh model free of gaps, overlaps, holes and slivers are obtained from the imperfect model input.

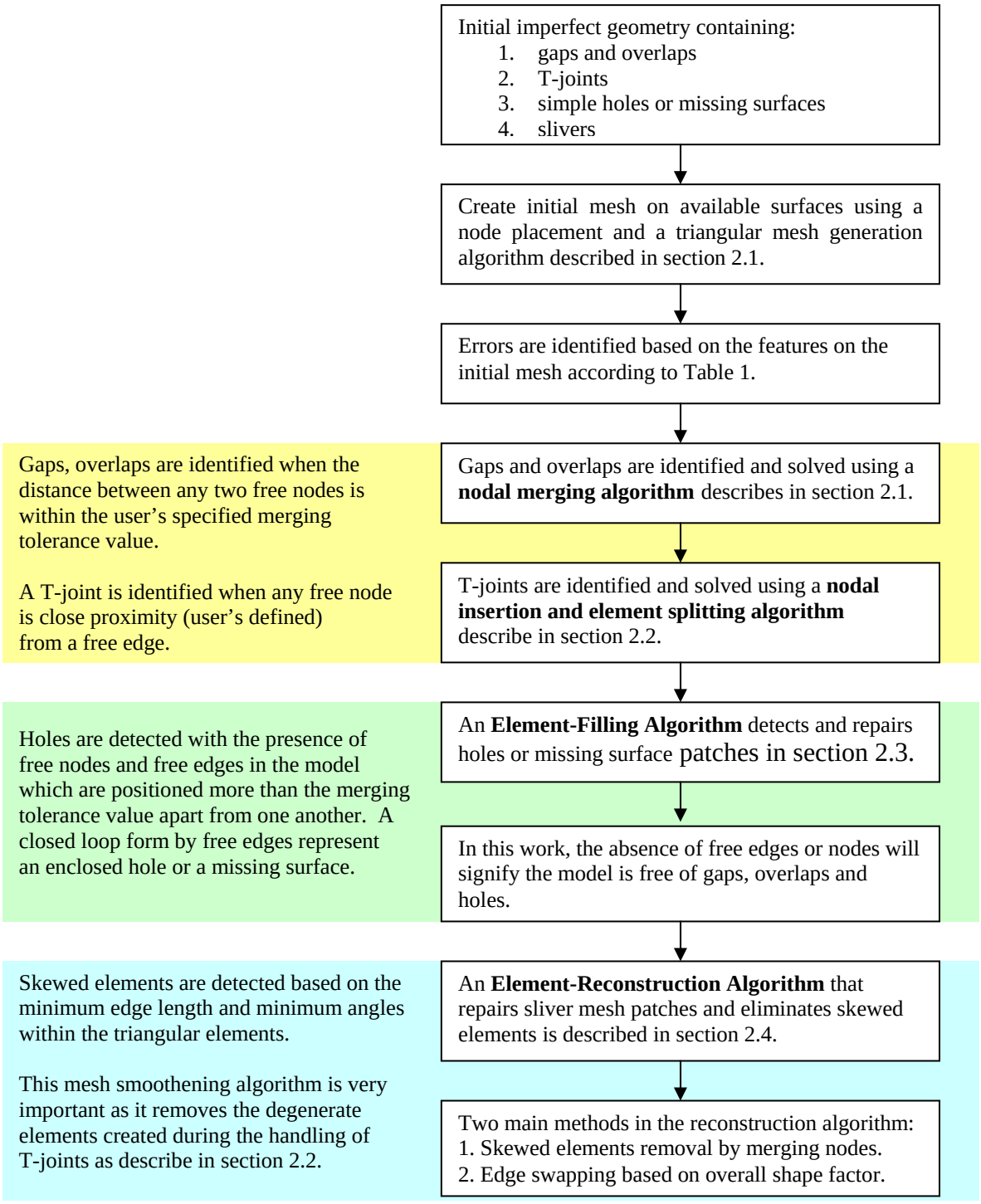


Figure 4.1 Summary of the Automatic Mesh Repair Algorithm

4.2 Automatic detection and closing of gaps and overlaps

Small gaps and overlaps are typically formed between adjacent surfaces where their supposedly common edges do not match. This can be due to modelling error or truncation error during the exporting and importing process. They can be detected by checking the proximity between a pair of candidate surfaces and their edges. Traditionally, small gaps and overlaps are detected by checking the closeness of two edges. This problem can be difficult when the two edges are totally different in length. The proposed nodal merging algorithm will sow mesh seeds along the edges of the original model and thereafter create the surface mesh with triangular elements conforming to the mesh seeds. Nodes on the free element edges will be merged if they are within a user-specified tolerance. The merging of nodes will be done by shifting the nodal positions based on a tangential interpolation.

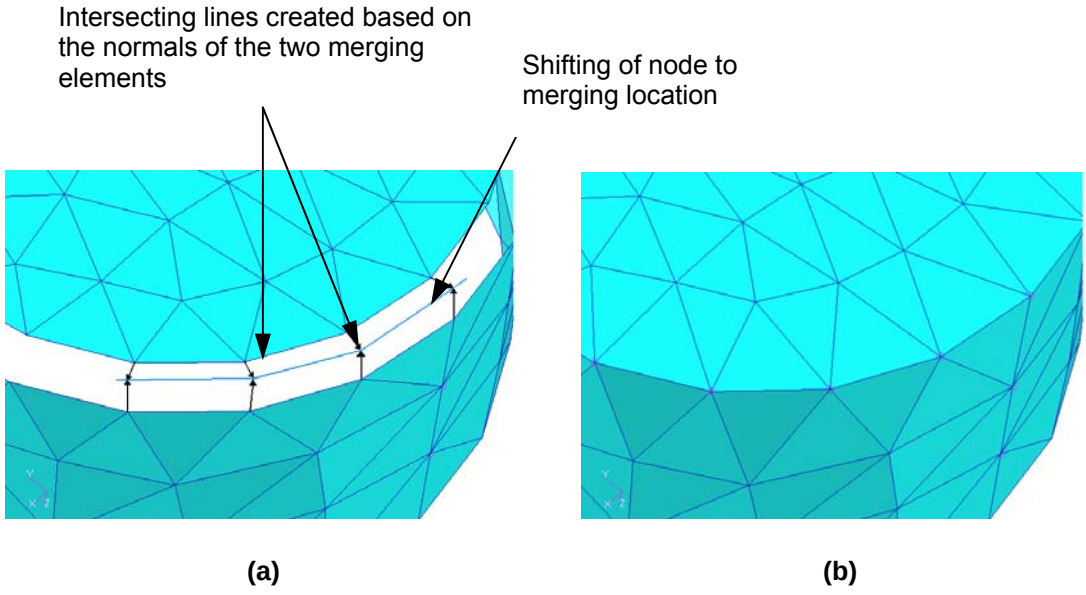


Figure 4.2 (a) Gap between two meshed surfaces; (b) Gap closed by merging nodes.

The boundary mesh seeds will be replaced by nodes after meshing is performed. Starting from the boundary mesh seeds created, we use a paving method to mesh all the surfaces such that all the boundary mesh seeds will be replaced by boundary nodes. The Original model is discretized with triangular elements. From here, free nodes will be paired along the gaps and overlaps based on merging tolerance specified by user. Merging of nodes alongside a boundary edge curve is carried out to remove gaps and overlaps. Figure 4.2 demonstrates how a gap is being closed up. When a node-pair is being detected, a line is formed from intersecting two tangential planes at the two merging nodes. These two nodes are then projected onto the intersecting line as shown in Figure 4.2(a) and the midpoint of the two projected locations will be the merging point of the merging nodes. If the intersecting line is too far away from the free nodes, or no intersecting line is created when the two planes are parallel to each other, then the merging point will be the midpoint of the original positions of the two merging nodes.

Figure 4.3 shows the flow of the initial meshing of the imperfect model and the repair of gaps and overlaps using the nodal merging algorithm. This process flow ends when there is absence of node-pair within the specified merging tolerance.

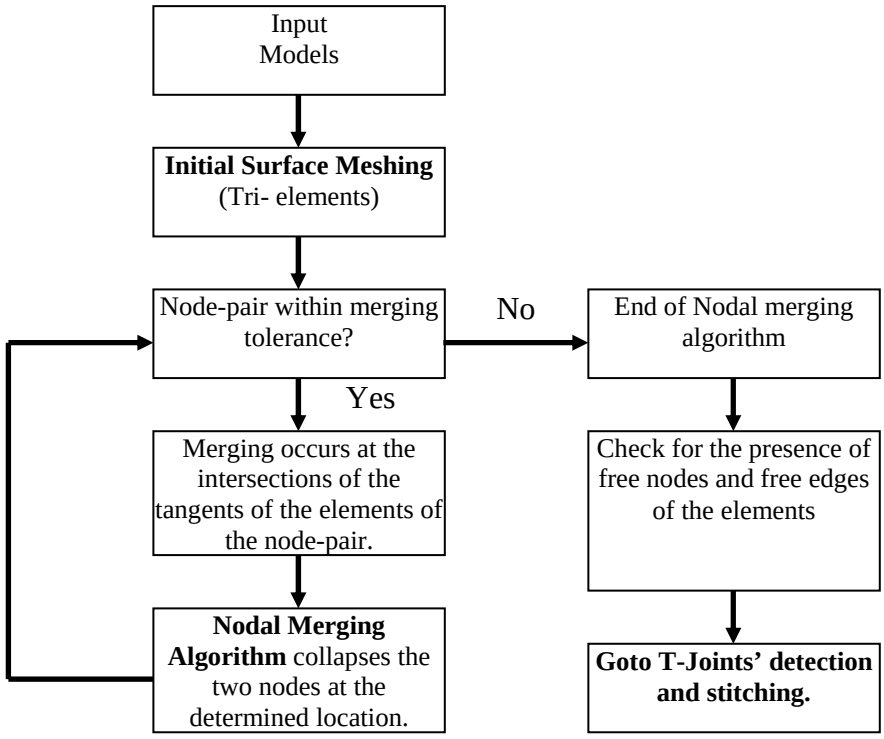


Figure 4.3 Stitching of gaps and overlaps using the nodal merging algorithm

4.3 Automatic detection and stitching of T-joint

After the gap-closing process in section 4.2, free node-pairs (nodes that belong to free element's edges) that are within merging tolerance value will not exist. However, T-joints arise due to topological errors as a result of in-congruency between surfaces that are connected to one another, and the nodes alignment along the common edge curve is very bad. This condition is unacceptable since a valid 3D model requires that it is watertight. The gap at a T-joint can be non-zero, but must be within the merging tolerance value to repair the T-joint. The stitching of T-joints will be achieved by a nodal insertion and element splitting algorithm to realize a conforming mesh along the T-joint. This nodal insertion and element splitting algorithm, as shown in Figure 4.4 and illustrated in Figure 4.5, is as follows:

1. For every available free edge, the algorithm will perform a local search for nearby free nodes that are within the specified merging tolerance distance perpendicular to that particular free edge. A node-to-edge pair is thus obtained.
2. The algorithm will obtain the positions of the perpendicular projections of these nodes project onto the free edges. New nodes are created on these positions along the free edge, and the triangular element connected to this free edge is eventually deleted and replaced by new triangular elements, containing the new nodes that are created.
3. The free nodes and their projected new nodes will be merged or equivalence.

Figure 4.6 shows some T-joints, each with one surface element connecting to two or more surface elements, the nodes along the connecting surfaces will not be able to merge due to their relative distance apart as well as non-conformity. Additional nodes are created to ensure edge conformity along connecting edge curves. Elements are split with the introduction of additional nodes to their free edges. Since the initial surface mesh consists of only triangular-elements, it will be relatively easy to shift the position of nodes, add or subtract elements. Every potential boundary node will not be left out un-paired; water tight model can thus be achieved. However special cases do exist and extra checking will be needed when performing the pairing and merging of nodes.

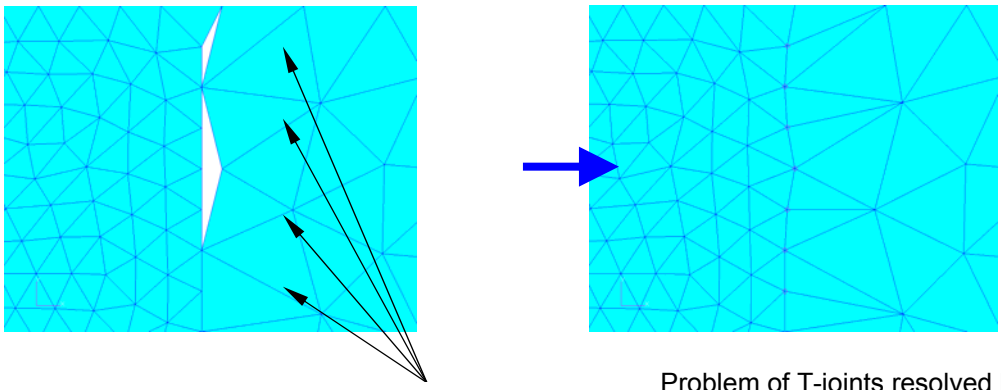


Figure 4.4 Handling of T-joints using nodal insertion and element splitting algorithm

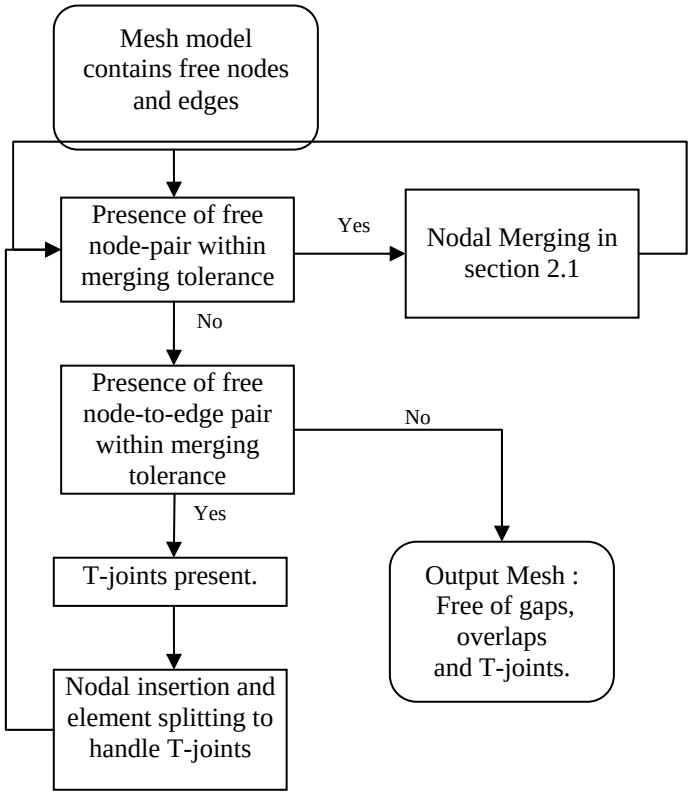
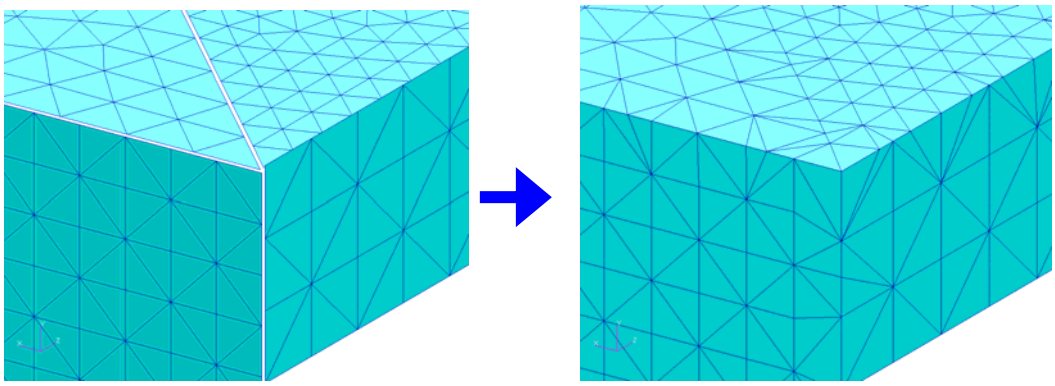


Figure 4.5 Repair of T-joints



(a) (b)

Figure 4.6 (a) Original T-Joint with non-conforming elements along the gaps, and (b)Elements split to obtain conformity along common edges

4.3.1 Approximating boundary curves

The control vertices of the cubic Bezier boundary curve ${}^3C_i(t)$ can be obtained using a point normal interpolation[18]. To ensure that the feature edges are modeled correctly, modifications are made to Walton's formulation to reflect the usage of unique nodal tangent vectors at the vertices of each boundary curve.

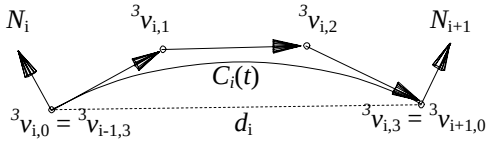


Figure 4.7 Cubic Curve Approximation

The unit normal vector, N_i, as in Figure 4.7, associated with a node on the boundary is calculated according to

$$N_i = \frac{\sum_{i=0}^{m} \alpha_i \hat{n}_i}{\sum_{i=0}^{m} \alpha_i} \tag{4.1}$$

where $\hat{n}_i$ is the unit normal vector of the triangle attached to the node, α_i is the angle subtended by the node and *m* is the number of triangles attached to the node.

Given that $d_i = \|Q_{i+1} - Q_i\|$, $a_i = N_i \cdot N_{i+1}$, $a_{i,0} = N_i \cdot T_{i,0}$ and $a_{i,1} = N_{i+1} \cdot T_{i,1}$, the control vertices ${}^3v_{i,j}$ (see Figure 4.7) are calculated based on the following equations:

$${}^3v_{i,0} = Q_i \tag{4.2}$$

$${}^3v_{i,3} = Q_{i+1} \tag{4.3}$$

$${}^3v_{i,1} = Q_i + \frac{d_i(6T_{i,0} - 2\rho_i N_i + \sigma_i N_{i+1})}{18} \tag{4.4}$$

$${}^3v_{i,2} = Q_{i+1} - \frac{d_i(6T_{i,1} + \rho_i N_i - 2\sigma_i N_{i+1})}{18} \tag{4.5}$$

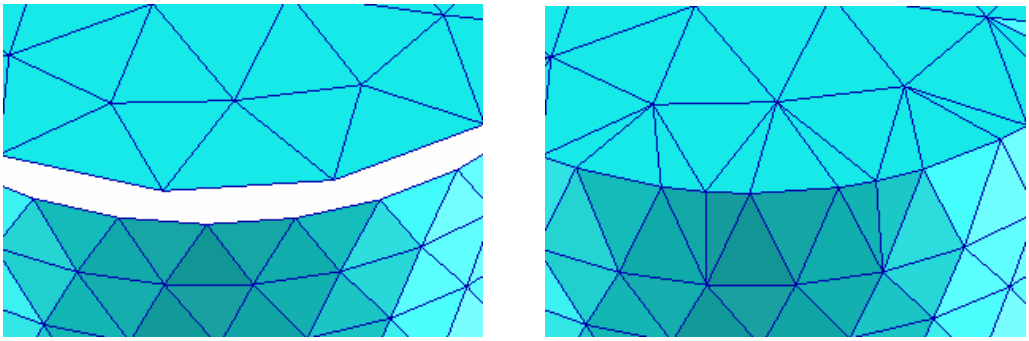
where

$$\rho_i = \frac{6(2a_{i,0} + a_i a_{i,1})}{4 - a_i^2} \tag{4.6}$$

and

$$\sigma_i = \frac{6(2a_{i,1} + a_i a_{i,0})}{4 - a_i^2} \tag{4.7}$$

The Cubic Curve Approximation algorithm helps to provide a smooth transition at the areas of merging and stitching. The key idea behind this algorithm is to incrementally modify the edges of the finite elements so that they conform to the curvature criteria specified by the upper and lower bounds defined by the user. These limiting values are defined in terms of the circular segmentation method, an interpolation technique, to determine the local curvature of the mesh. In this way, the curvature and any new node to be present at any location along the curve edge, ${}^{3}C_i(t)$, can be accurately calculated as in Figure 4.7. Figure 4.8 demonstrates how a gap is being closed up when the boundary nodes along the two sides of a gap are not conformed in positions and numbers as compared to Figure 4.2.



(a) **(b)**

Figure 4.8 (a) Gap between two meshed surfaces; (b) Gap closed by merging nodes using Cubic Curve Approximation.

4.4 Automatic hole filling using a heuristic elements-filling algorithm

After closing up the gaps and remove the overlaps, errors such as holes or missing surfaces may still present. In hole-filling, there are two main types of holes and they are categorized as simple and ring holes. A simple hole here is

defined as a hole of any shape with only one boundary loop. A ring hole is defined as a hole consisting of at least two peripheral loops (see Figure 4.9).

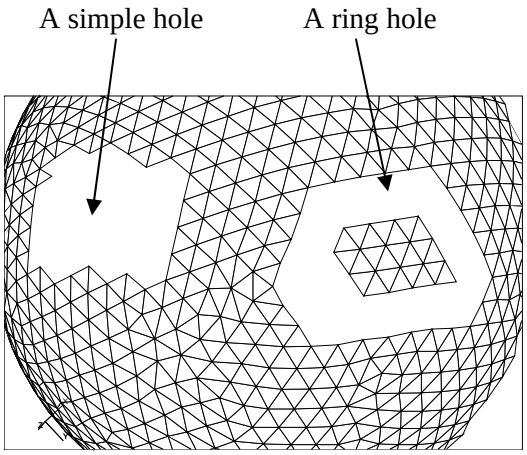


Figure 4.9 Typical examples of a simple hole and a ring hole on a surface mesh/polygonal representation

The steps of the element-based hole-filling algorithm are as follows:

1. Detect closed loops made up of free elements' edges. Handle each closed loops one at a time.
2. Calculate minimum angles, α, between two adjacent free elements' edges of the closed loop.
3. Fill element within the holes:
    a. If the angle α is smaller than 75^o, form a triangular element with these two free elements' edges, edges A and B, as shown in Figure 4.10(a).
    b. Else if the minimum angle α is between 75^o and 135^o, two new equilateral triangular elements will be created using the two free elements' edges. The normals of the two new elements, n_1 and n_2, as in Figure 4.11, are calculated from the vectors n, n_A and n_B at nodes 1, 2 and 3 respectively. The values of n, n_A and n_B are computed using equations (4.8) – (4.12), which calculate the

averages of the normal vectors of the adjacent elements of nodes 1, 2 and 3 respectively. The two new nodes of these two elements are averaged and merged (see Figures 4.10(b) and 4.11). Whenever, a new node is formed, there will be a proximity check with the other free nodes. If any two nodes are within a close proximity, as specified by the user, says half the length of a element's edge, then they will be collapsed into a single node.

$$n = \frac{\sum_{i=1}^{m} n_i}{m} \tag{4.8}$$

$$n_A = \frac{\sum_{i=1}^{m_A} n_{Ai}}{m_A} \tag{4.9}$$

$$n_B = \frac{\sum_{i=1}^{m_B} n_{Bi}}{m_B} \tag{4.10}$$

$$n_1 = \frac{(n + n_A)}{2} \tag{4.11}$$

$$n_2 = \frac{(n + n_B)}{2} \tag{4.12}$$

c. Else if the minimum angle α is larger than 135 ^{o}, perform step 5 except that the third nodes are not required to be merged as shown in Figure 4.10(c).

4. Repeat steps 2 and 3 until the all the holes are being patched with triangular elements as shown in Figure 4.10(d).

The numbers "75 and 135" are user-input and they are chosen because the best triangular element is in the shape of an equilateral triangle with angles at

60 degrees. Thus, a good tolerable angle size would be 75 degree. If the available angle size is between 75 and 135 degrees, it would be advisable to squeeze in two triangles. This would assure that the triangle will at least have a corner with an angle size in between 37.5 and 65.5 degrees. If the angle is smaller than a tolerable value, the particular triangle will undergo element swapping or removal in section 4.5.

For cases of ring holes, in Step 1, we have to identify ring holes interactively. It is in fact not possible to co-relate the hole's boundary and its islands automatically, especially when both are not in close proximity. Two or more boundary loops will be grouped together if the ring holes are identified. If they are not being identified, then the outer peripheral loop will be treated as a simple loop and the inner loop will be treated as a dangling geometry/meshed patch, which will be discarded eventually. The resulting model is still topologically consistent; however, its level of fidelity is lowered. In our solution, a function is developed where two nodes, belonging to two different boundary loops are identified by user. The algorithm will search for a shortest bridge to be created between these two successive peripheral loops (see Figure 4.12(a)). Elements will be created along this bridge, thus converting the ring holes into simple holes (see Figure 4.12(b)). Using the element-based hole filling method, the hole is to be patched with triangular elements (see Figure 4.12(c)).

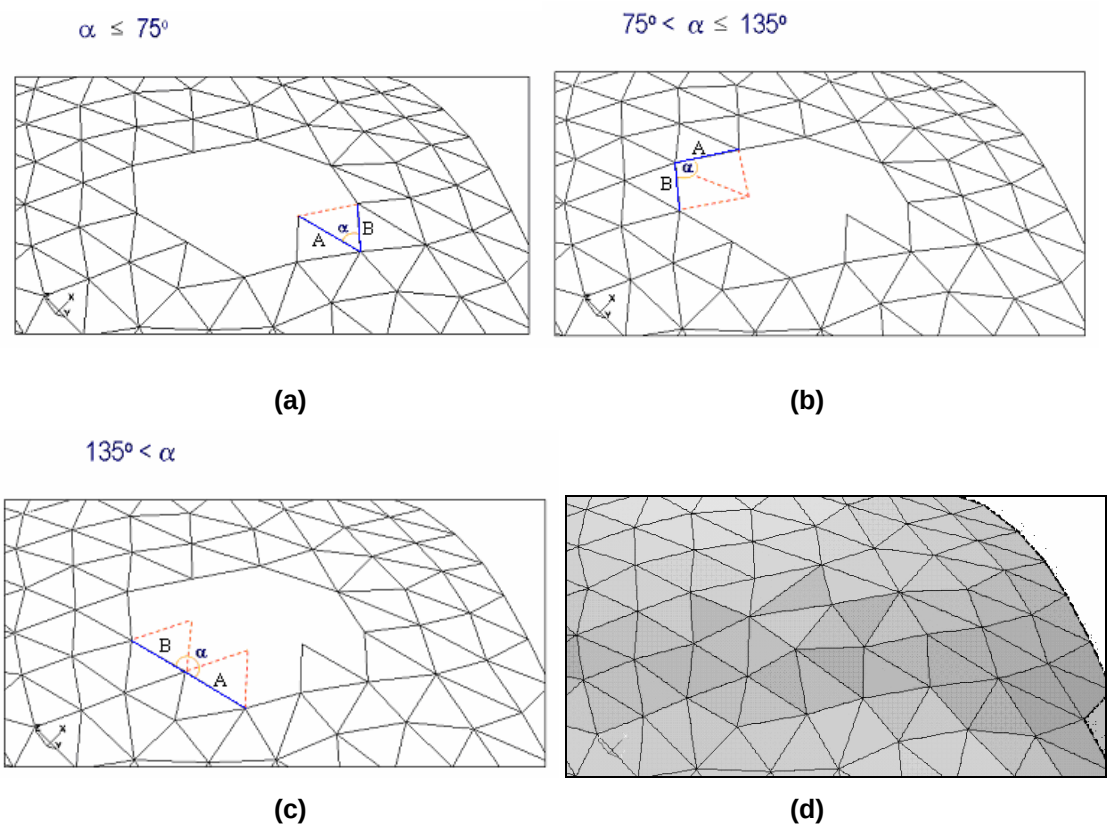


Figure 4.10 (a) Elements-filling when α is less than 75°; (b) Elements-filling when α is between 75° and 135°; (c) Elements-filling when α is larger than 135°; and (d) Simple hole filled using elements-filling algorithm

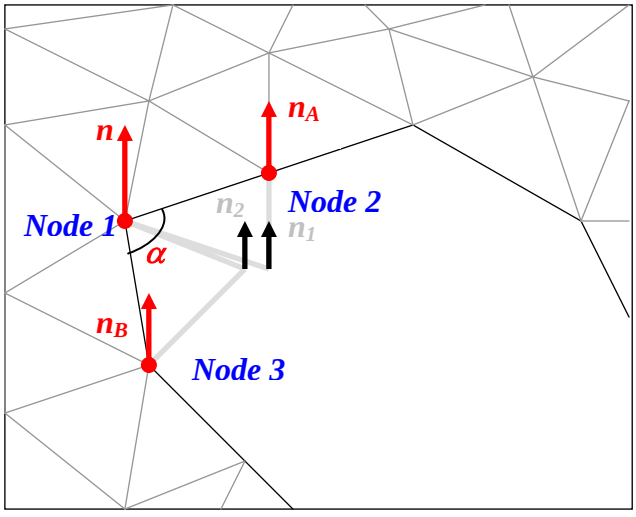


Figure 4.11 Elements-filling when α is large than 75°

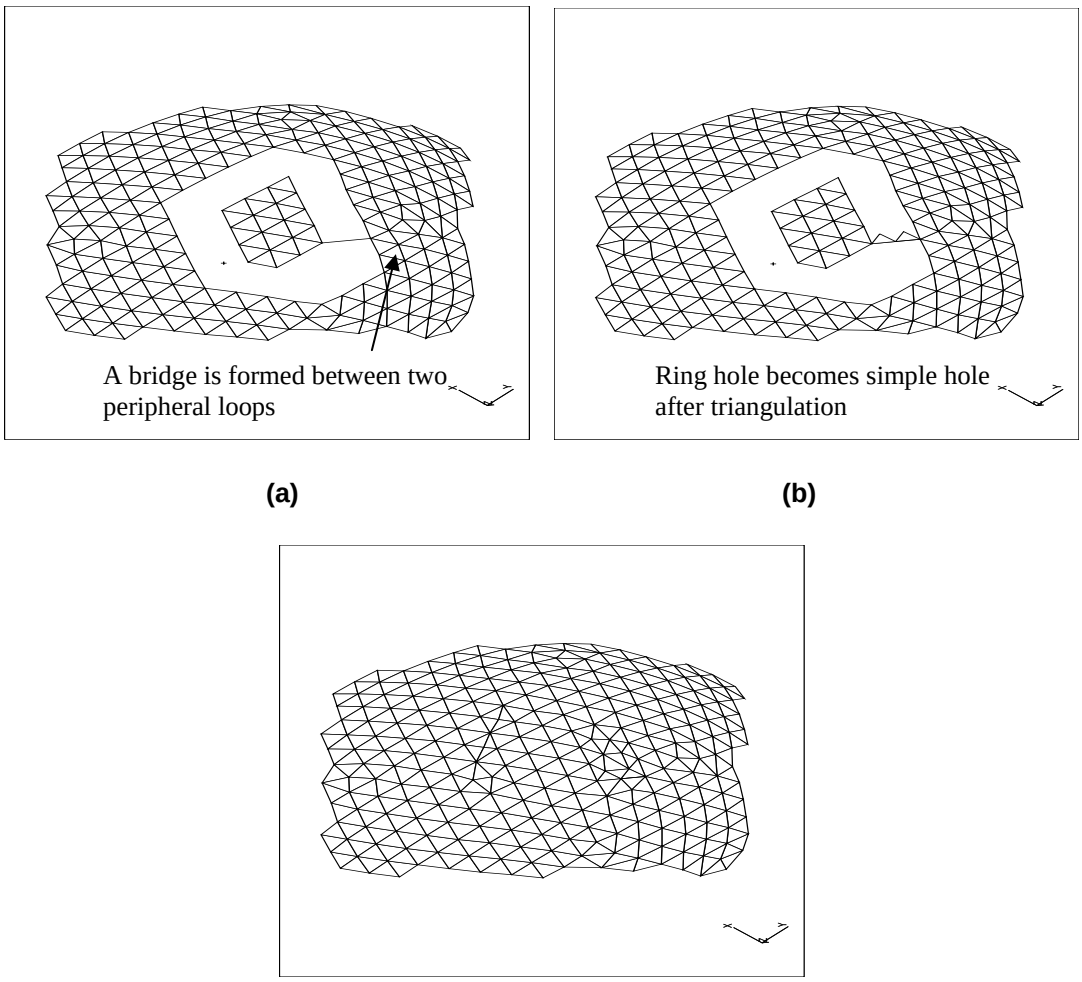


Figure 4.12 (a) Tackling of a ring hole by forming a bridge between two peripheral loops; (b) Elements are created along the bridge, and (c) Elements created to fill up the hole.

4.5 Automatic detection and removal of skewed elements and sliver surfaces.

Automatic detection and removal of skewed triangles is extremely useful and important, during the handling of T-joints and holes, when elements of high degeneracy are usually generated in these repair processes. A typical sliver surface/triangle is long and slender. In the geometrical sense, the ratio of its surface area over its perimeter is significantly small in value. The mesh of a

sliver surface tends to yield elements of very poor aspect ratio. This is undesirable as far as numerical solution is concerned.

To rectify the problem, the method will improve the quality of the meshes of sliver surfaces by an element reconstruction algorithm. This reconstruction algorithm performs nodal-merging (as shown in Figure 4.13) and edge-swapping to improve overall quality of the mesh. Edge swapping is done by changing the connectivity of the common edge between two connecting triangles as shown in Figure 4.14. The user input to the reconstruction algorithm is the tolerance edge length. Any elements with edges shorter than the tolerance edge length will face the fate of possible elimination or reconstruction. The shape factor, *f*, and the smoothness angle, α , are the two main factors that affect how an element will to be reconstructed or removed. These two factors will help to preserve the features in the finite element model. In equation (4.13), *E* represents the equilateral factor of a triangle, and θ_j is the internal angles of the triangle. A triangle having a value of *E* = 0, denotes an equilateral triangle. The maximum as well as most undesirable value of *E* is $\frac{2}{3}\pi^2$, thus the value of its shape factor, *f,* as shown in equation (4.14), is low if the shape of the triangle is highly skewed. The shape factor is a user-input value. Setting of the shape factor to 0.7 would remove all the skewed elements that likely to inhibit an analysis. The elements with edge lengths shorter than the input tolerance edge length value and a low shape factor will probably undergo an element removal by merging nodes, while the elements with only low shape factor will undergo edge swapping to achieve optimal solutions.

$$E = \sum_{j=1}^{3}\left(\theta_j - \frac{\pi}{3}\right)^2 \tag{4.13}$$

$$f = \frac{2}{3}\pi^2 - \sum_{j=1}^{3}\left(\theta_j - \frac{\pi}{3}\right)^2 \tag{4.14}$$

Figure 4.13(a) shows an example of a degenerate element with small face area and one of the element edges being very small in length as compared to the other two element edges. Figures 4.13(b) and 4.13(c) show that certain edges of the degenerate element will be retained. It is determined by the principle angles between the degenerate element's face and the adjacent elements' faces. Presently, we are using a default angular value of 135 degree. If the angle at a certain edge of the degenerate element is less than 135 degree, then this element edge will be treated as a feature edge and will be retained. If the angle is larger than 135 degree, the element will be treated as if it is lying on a smooth surface and it can be collapsed with an averaging method (see Figure 4.13(d)). Figure 4.14 shows another example of a degenerate triangular element that has small face area. The reconstruction of the degenerate mesh will be performed edge swapping as shown in Figure 4.14(b), if edge B or edge C are feature edges. If edge A is to be retained, the reconstruction of the degenerate mesh will be done as shown in Figure 4.14(c). Figure 4.15 shows an example of the reconstruction of the mesh of a sliver surface.

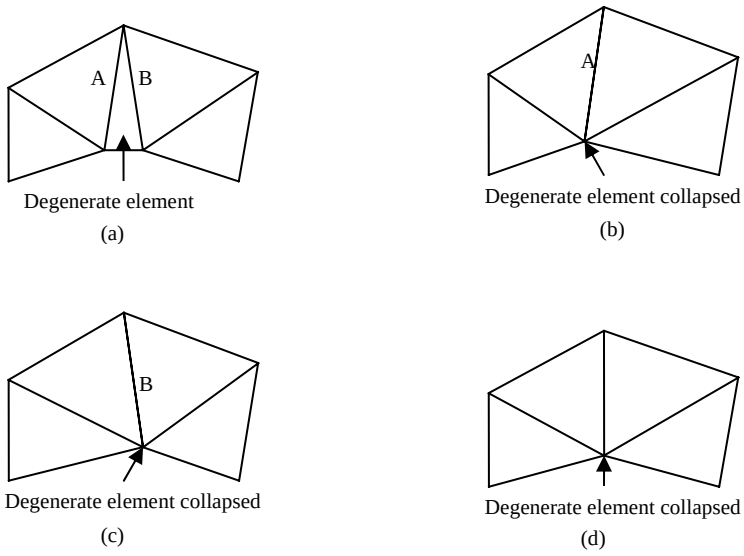


Figure 4.13 (a) Original degenerate mesh, (b) Edge A is a feature edge, (c) Edge B is a feature edge, and (d) Both edges are non-feature edges.

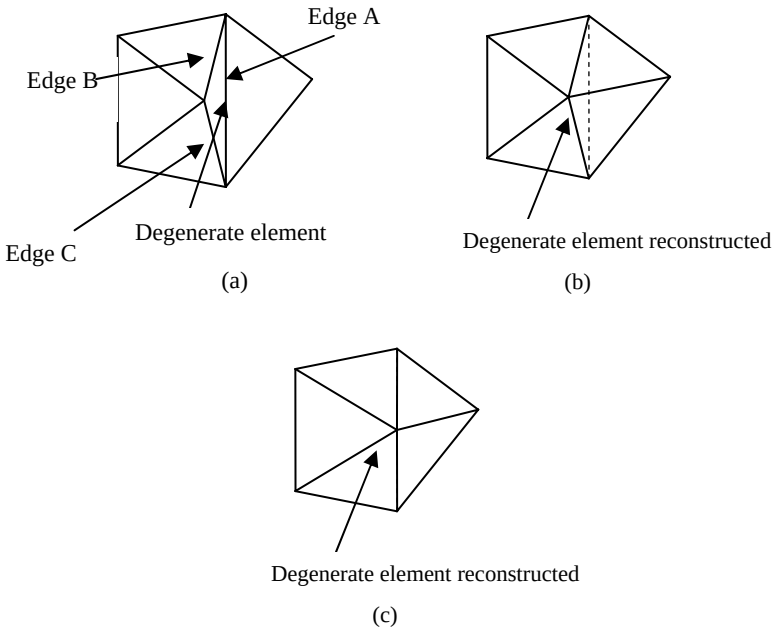
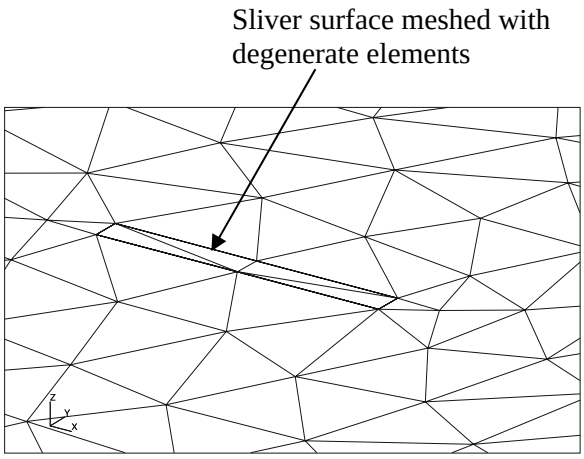
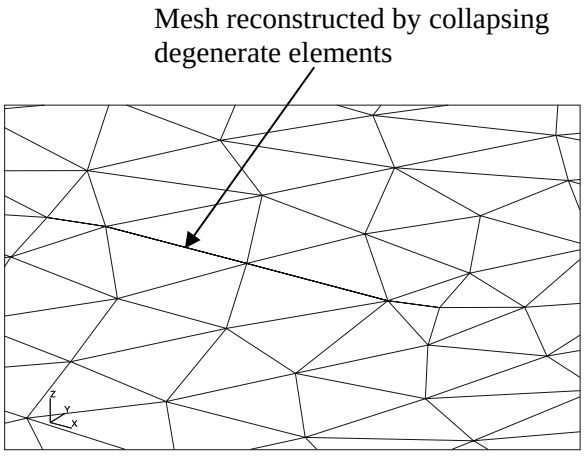


Figure 4.14 (a) Original degenerate mesh, (b) Collapsing algorithm if line A does not need to be retained, and (c) Collapsing algorithm if line A need to be retained



(a)

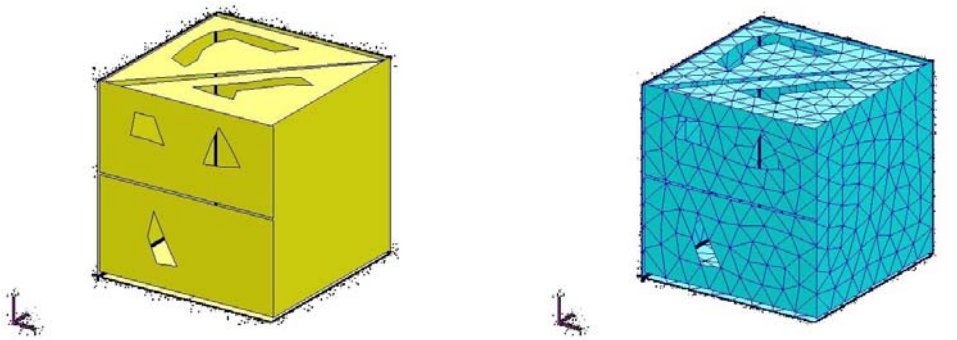


(b)

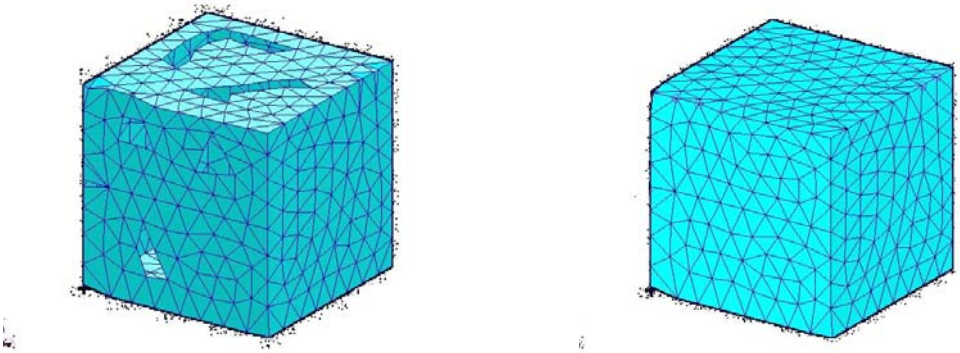
Figure 4.15 (a) Sliver surface meshed with degenerate elements; (b) Mesh reconstructed

4.6 Results and discussions

Figure 4.16(a) shows an example of a surface model of a cube that is made up of disconnected surfaces with gaps and holes present. The mesh healing algorithm will first mesh the disconnected surfaces as in Figure 4.16(b) and then the gaps and overlaps in between the elements are closed and removed respectively to create close topology for the manifold model as shown in Figure 4.16(c). The skew elements, caused by the element-splitting method that handles T-joints, can be easily removed by a mesh smoothing algorithm. The holes present on the surfaces are filled using the element-filling method and the resulting healed model is shown in Figure 4.16(d). Figure 4.17(a) shows a polygonal model of a sphere contains gaps, overlaps, joints and holes. After mesh-repair, the sphere achieves closed topology as shown in Figure 4.17(b). Figure 4.18(a) shows a casing part taken from a model of a car, described by six surfaces in IGES format. When this part is being imported into a mesh generation system, even using non-adaptive meshing technique will cause non-conformity among elements. This can be due to the present of gaps and overlaps as well as T-joints in between surfaces. Figure 4.18(b) shows that the mesh of the whole casing part is not conforming. Elements are being split to ensure conformity along the gaps, overlaps and T-joints in between surfaces as shown in Figure 4.18(c).

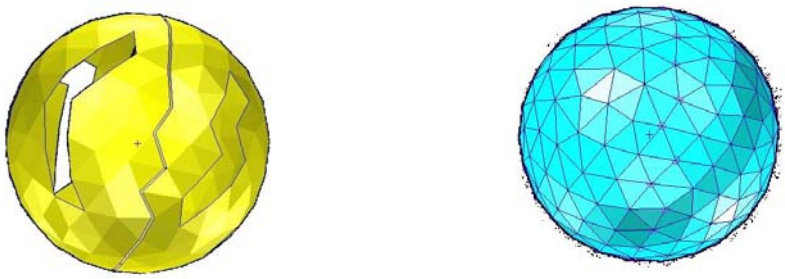


(a) **(b)**



(c) **(d)**

Figure 4.16 (a) Surface model of a cube made up of disconnected surfaces with gaps, overlaps and holes; (b) The initial mesh of the surface model; (c) Mesh healing of gaps, overlaps and holes; and, (d) The resulting surface mesh after element-reconstruction and element-filling.



(a) **(b)**

Figure 4.17 (a) A polygonal representation of a sphere with gaps and holes; and, (b) The "mesh-healed" sphere

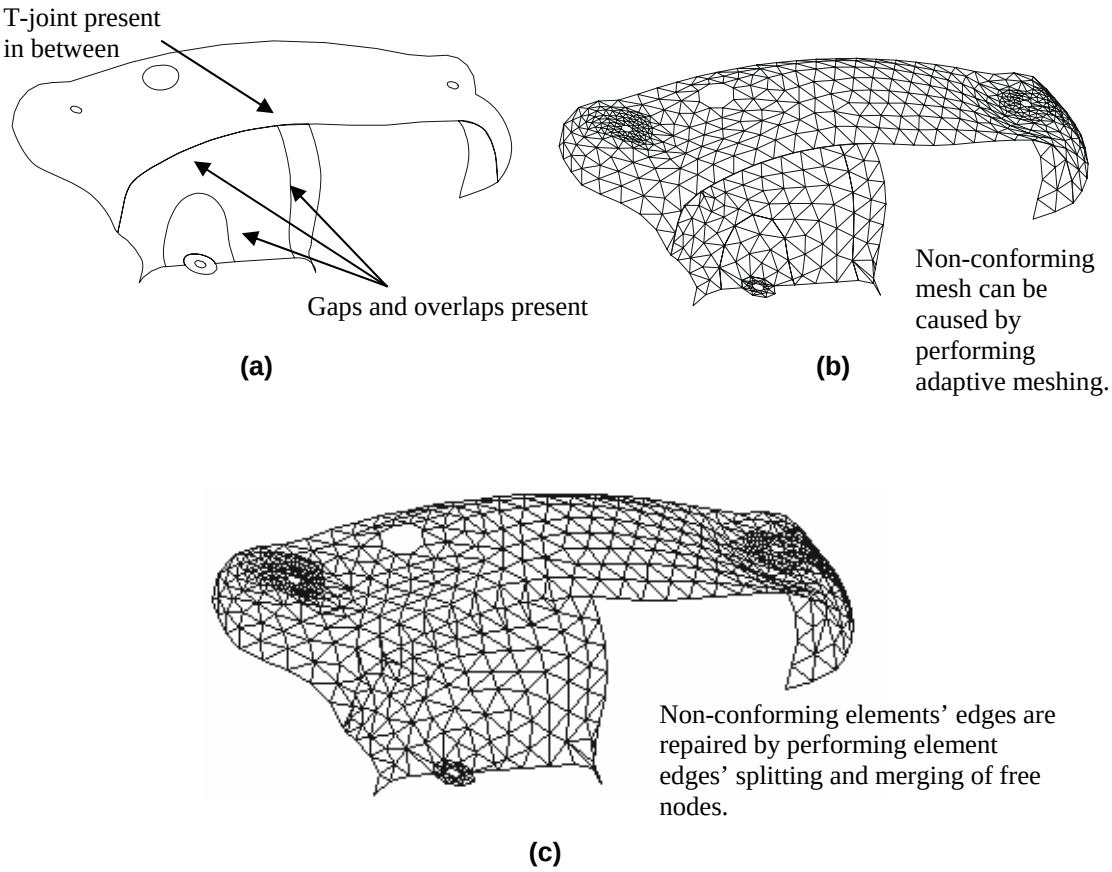


Figure 4.18 (a) Surface representation of a casing; (b) Triangular mesh created on the surfaces of the casing; (c) Surface mesh after mesh repair process.

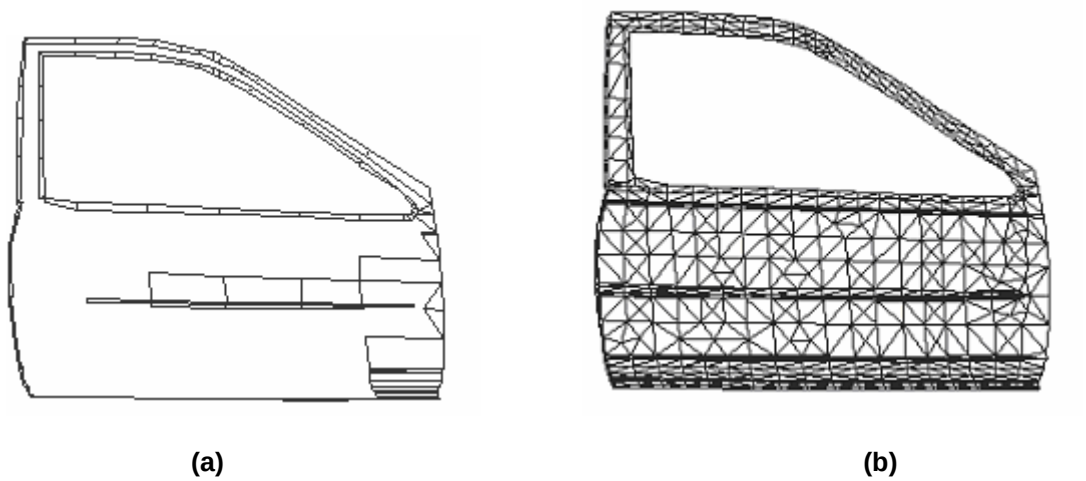
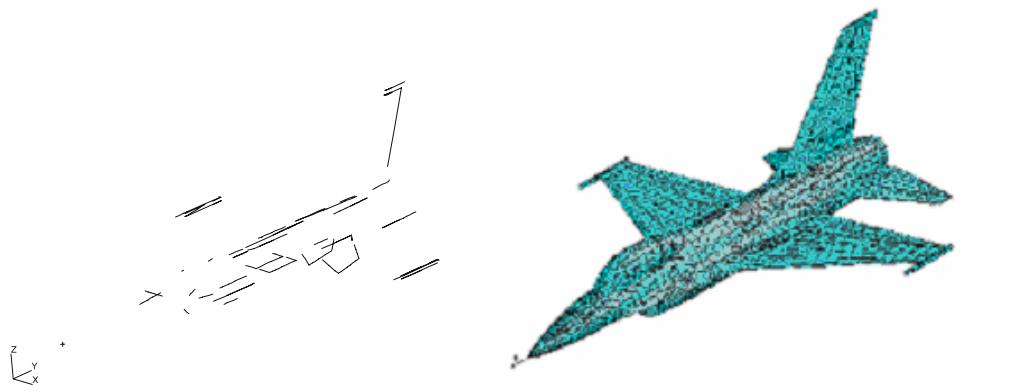


Figure 4.19 (a) Incongruent surfaces of the car door, and (b) Healed surface mesh of the car door.



(a) (b)

Figure 4.20 (a) Boundary gaps or non-conforming edges of the initial mesh of the aircraft before mesh repair process; and (b) Water-tight surface mesh of an aircraft after mesh repair process.

Figure 4.19(a) shows a surface model of a typical car door which contains congruent surfaces with gaps and overlaps. The mesh-repair algorithm creates triangular mesh over each surface, stitches the gaps and overlaps by merging the nodes. Figure 4.19(b) shows the final repaired model with conforming mesh. Figure 4.20(a) shows the final surface mesh of an aircraft. Figure 4.20(b) shows the boundary contour formed by the free element edges, due to the gaps and overlaps within the initial surface mesh. Consequently, the holes remained after the node merging step will undergo triangulation to obtain a fully repaired model. Figure 4.21 illustrates how a model translation can cause errors in the model and how the automatic mesh-repair algorithm can repair the imperfect model to obtain water-tightness. Figure 4.21(a) shows a solid model of a connector assembly created using a CAD system. This model is exported into an “IGES” format CAD file. Errors such as gaps and holes due to truncation errors and missing surfaces are found, when this IGES file is later imported into another (mesher) system as shown in Figure 4.21(b). Here the automatic mesh repair technique is applied and initial mesh is

generated on the imperfect model as shown in Figure 4.21(c). The gaps and holes are repaired by the automatic mesh-repair process and the final healed model is shown in Figure 4.21(d).

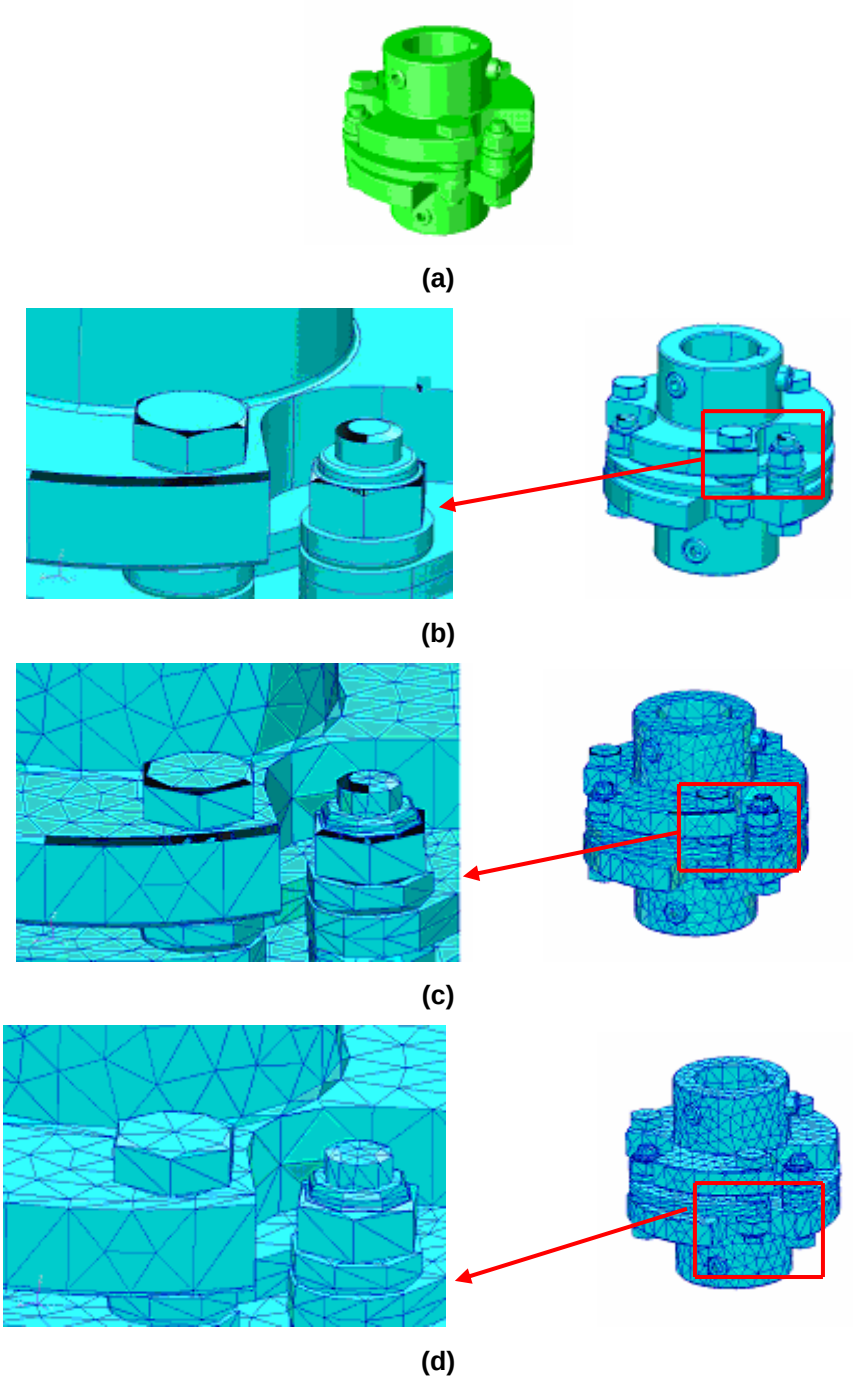


Figure 4.21 (a) Solid model of a connector in, (b) Model is exported into IGES format and read into a meshing system which shown errors, (c) Imperfect model is meshed and undergoes automatic mesh-repair, and (d) Final output of the mesh with enclosed water-tightness.

Chapter 5:

High Fidelity Hole-Repair in Meshes with Shape Prediction

The research here will focus on developing an automatic hole-filling algorithm to handle defective CAD models in unstructured triangular surface mesh representation, such as those commonly found in the stereo-lithography format. The key feature of the proposed algorithm is the capability to approximate the missing shape or geometry over the hole. The algorithm handles holes of any arbitrary boundaries in an oriented manifold mesh and it ensures water-tightness of the mesh after patching the missing surfaces. To cater to complex geometric configurations, a Genetic Algorithm is proposed for the purpose of optimal triangulation based on a global minimization of dihedral angles. A quartic Bézier surface interpolation is then proposed to be performed over the optimal initial triangulation to approximate the shape over the hole. Next, an unstructured triangular mesh is to be generated over the hole using a customized Advancing Front meshing algorithm which based its geometric references on the surface interpolation. This allows the mesh to model the missing shape using geometric information in the vicinity of the hole. The customized Advancing Front meshing algorithm also ensures that elements of good quality are achieved and that the resolution of the mesh at the patched region matches the mesh density at the locality of the hole.

In this work, the objective is to develop a robust and automatic technique for filling holes in triangulated models such that the underlying shape is approximated with good fidelity. Our method uses Genetic Algorithm to obtain a valid and optimal initial triangulation even when the hole is geometrically and topologically too complex, such as, a hole over a highly non-planar region which consists of positive and negative curvatures. The shape approximation capability is achieved by exploiting the geometric information provided by the mesh that surrounds the hole. This allows us to model the underlying shape by making use of as much localized information as possible, hence allowing highly, and varying curvatures to be modeled. A customized Advancing Front meshing is then performed over the approximated shape to generate an unstructured triangular mesh over the region. This method is not only well suited for the automatic repair of mesh models used in simulation-driven applications, but it can also be used to restore incomplete or impaired biomedical models obtained from data-acquisition devices, such as in cranioplasty applications.

5.1 Methodology

In this section, we describe the main stages of our proposed hole-filling algorithm. They are: hole identification and smoothing, initial triangulation using Genetic Algorithm, shape approximation based on quartic surface interpolation, and unstructured mesh generation using a customized Advancing Front technique. Figure 5.1 shows the flowchart of the complete hole-filling process.

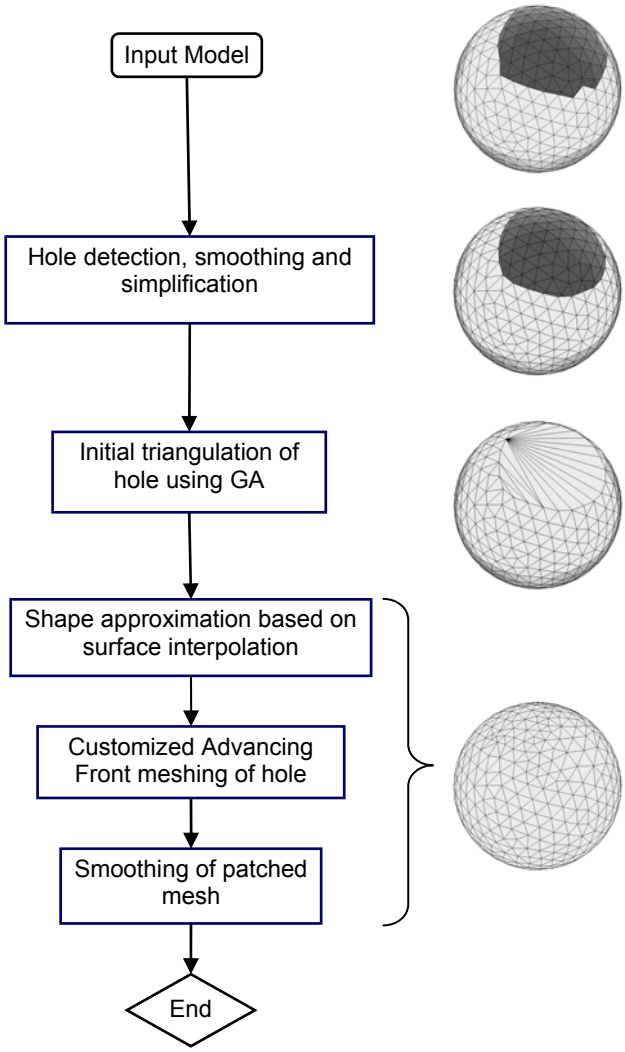


Figure 5.1 A flowchart of the hole-filling algorithm

In the hole identification process, the boundary representing the hole is extracted from the mesh model. The boundary then undergoes smoothing to remove crenellations and jagged artifacts so as to improve the result of the subsequent shape approximation. Next, an initial triangulation is performed to cover the hole using the Genetic Algorithm (GA) so that the triangles transit smoothly across their edges. The existing nodes on the boundary of the holes are used so that the model is water-tight. Next, a shape approximation process converts the triangles created during the initial triangulation to a set of

continuously connected triangular Bézier patches, based on a quartic surface interpolation [37]. These patches form the underlying geometry and they are used as basis for a customized Advancing Front meshing algorithm to create an unstructured triangular mesh over the hole. Finally, the mesh over the hole is smoothed using a mean curvature flow smoothing algorithm [38] to further improve the quality of the triangles. The subsequent sections will elaborate on each of these stages in detail.

5.2 Hole Identification

A hole in a triangulated model is defined as a closed connected loop of one-manifold edges, or free edges, bounding any arbitrary shape. Here, the assumption is made that the elements of the triangulated model are correctly oriented and connected, and that a given hole will not have islands. Each hole is thus bounded and completely described by a set of connected straight line segments.

5.3 Hole Simplification

After the hole is identified, it is triangulated with a Genetic Algorithm having an aim to minimize the dihedral angles between the newly created triangles. A hole that is relatively planar is simpler to triangulate as compared to a hole that is highly non-planar. The challenge is even greater when the boundary contains crenellations, or jagged edges. Such crenellations will lead to sharp folds at the edges after the initial triangulation, which is undesirable for the subsequent surface approximation process.

5.3.1 Hole Smoothing

To ensure that the triangles do not transit with sharp folds, the free edges are smoothed to reduce the crenellations. The smoothing is achieved by re-positioning the boundary nodes based on

$$B' = \frac{1}{2}\left(\frac{A+C}{2} + B\right) \qquad (5.1)$$

where B is the position of the node to be smoothed, B' is the new position of B, and A and C are the positions of the adjacent boundary nodes to B, as shown in Figure 5.2. All the new positions of the boundary nodes are calculated and the positional update is done synchronously. Equation (5.1) is simple to compute and it provides a good approximation in terms of positions and points' normals for the new boundary points. There is no node being added or removed. The boundary points' normal are computed from its adjacent triangular elements' normal, which are described and used in the up-coming stages. An example of boundary edge smoothing is shown in Figure 5.3. The free edges forming the hole can also be extracted separately and performed edge reduction so as to reduce the computation effort to perform initial triangulation using the Genetic Algorithm at the following stage.

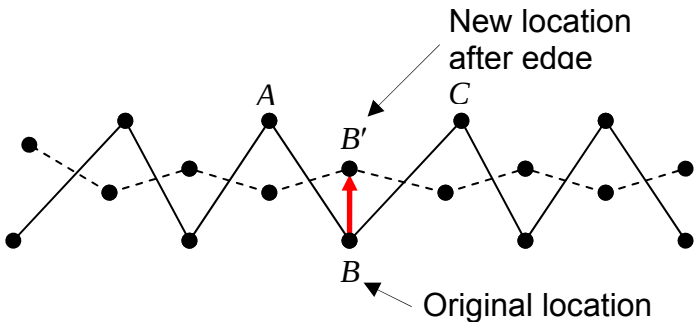


Figure 5.2 Boundary edge smoothing technique

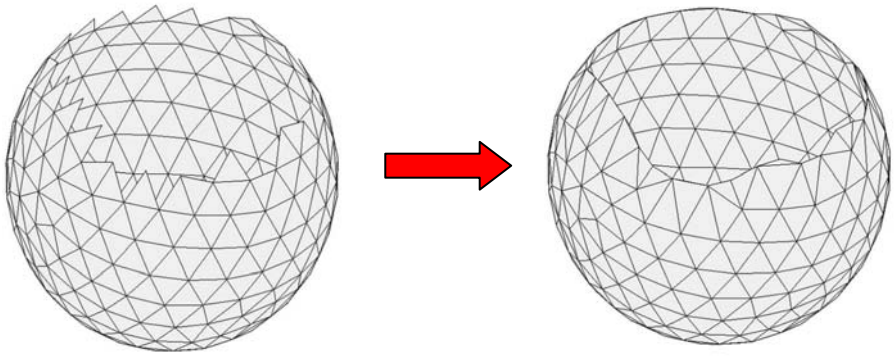


Figure 5.3 Boundary edge smoothing to reduce crenellations

5.3.2 Hole Simplification using Rough Set Theory

The application of Rough Set Theory in hole simplification is basically to speed up the process of the initial triangulation of the hole using Genetic Algorithm (G.A.). The computational time required for the execution of a single GA process increases exponentially with respect to its population size. G.A. processes are highly computational intensive and a closed, complex hole in a mesh can be easily made up of hundreds of boundary edges (population size). After the crenellations are removed in chapter 5.2.1, the number of boundary edges of the hole ought to be reduced so as to reduce the computational effort by reducing the population size in the G.A. processes that are used for the initial triangulation of the hole in chapter 5.3.

The theory was originated by Zdzislaw Pawlak [39] in 1980's. Its methodology is concerned with the classificatory analysis of imprecise, uncertain or incomplete information or knowledge expressed in terms of data acquired from experience [40]. The primary notions of the theory of rough sets are the approximation space and lower and upper approximations of a set. The approximation space is a

classification of the domain of interest into disjoint categories. The classification formally represents our knowledge about the domain, i.e. the knowledge is understood here as an ability to characterize all classes of the classification, for example, in terms of features of objects belonging to the domain. Objects belonging to the same category are not distinguishable, which means that their membership status with respect to an arbitrary subset of the domain may not always be clearly definable. This fact leads to the definition of a set in terms of lower and upper approximations. The lower approximation is a description of the domain objects which are known with certainty to belong to the subset of interest, whereas the upper approximation is a description of the objects which possibly belong to the subset. Any subset defined through its lower and upper approximations is called a rough set. This rough set here is a boundary region that is defined by the difference between the lower and the upper approximations, consisting of all objects, cases or possibilities which cannot be classified with certainty to the concept or its complement employing available knowledge. The greater the boundary region, the vaguer is the concept. If the boundary region is an empty set, the concept will be precise.

The data set, in Table 5.1, is obtained from some basic geometric rules and visual inspections of several holes present in typical bio-models. Most surfaces of these bio-models are smooth and continuous, with very few features, such as sharp corners and edges. The preparation of this data is done interactively, where each boundary node is manually highlighted whether it is to be preserved or removed in the simplification of a hole. This information is then processed together with the attribute values of α and β computed as shown in Figure 5.4. The principle

angle, α, is defined here as the smallest angle between two connected edges adjacent to the affected nodes. The normal deviation angle, β, is the angle between the two normal vectors of the two boundary triangular elements of the affected node. The nodal sample size of the data set, *N*, is 226. Each attribute of α and β is being subdivided into 5 sub-ranges, i.e. $\alpha 1$ τo $\alpha 5$ and $\beta 1$ to $\beta 5$, as shown in Table 5.1.

Table 5.1 Formulation of data sample set

Node	Principle angle, α, in degree		Normal deviation angle, β, in degree		Nodes to be preserved or be removed
n1	very large, $\alpha 1$	$\alpha > 165$	very small, $\beta 5$	$0 <= \beta < 15$	Removed
n2	Large, $\alpha 2$	$120 < \alpha <= 165$	very small, $\beta 5$	$0 <= \beta < 15$	Removed
n3	large, $\alpha 2$	$120 < \alpha <= 165$	Small, $\beta 4$	$15 <= \beta < 30$	Preserved
n4	medium, $\alpha 3$	$90 < \alpha <= 120$	very small, $\beta 5$	$0 <= \beta < 15$	Preserved
n5	medium, $\alpha 3$	$90 < \alpha <= 120$	Medium, $\beta 3$	$30 <= \beta < 60$	Preserved
n6	small, $\alpha 4$	$45 < \alpha <= 90$	very small, $\beta 5$	$0 <= \beta < 15$	Preserved
n7	small, $\alpha 4$	$45 < \alpha <= 90$	large, $\beta 2$	$60 <= \beta < 90$	Preserved
n8	very small, $\alpha 5$	$\alpha <= 45$	very small, $\beta 5$	$0 <= \beta < 15$	Removed
n9	very small, $\alpha 5$	$\alpha <= 45$	very large, $\beta 1$	$\beta >= 90$	Removed
. . . .	. . . .	. . . .	. . . .	. . . .	. . . .

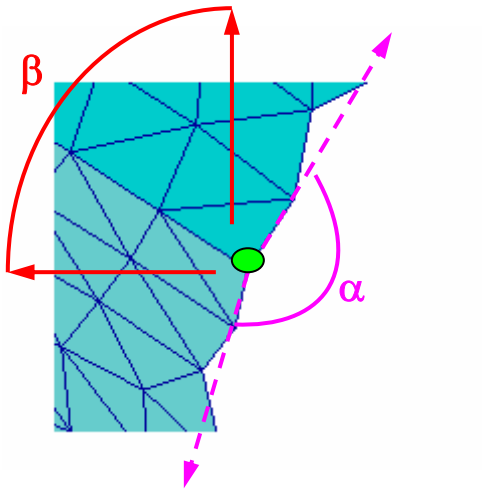


Figure 5.4 Definitions of α, the angle between the two boundary edges, and β, the angle between the normals of two boundary elements adjacent to the affected node in green

The set of removed nodes is given by S_{Remove} = {*n1, n2, n3, n8, n9,…*}

The set of removed nodes is given by $S_{Preserve}$ = {*n4, n5, n6, n7,…*}

The Lower Approximation for a node to be removed, L_{Remove}, is the number the definite and unique cases (e.g. $\alpha 2$ and $\beta 5$) of a node being removed, and there cannot exist another case (also $\alpha 2$ and $\beta 5$) where this other node is being preserved based on the same attributes in α and β. The Upper Approximation for a node to be removed, U_{Remove}, is the total number of cases of a node being removed. The Lower Approximation for a node to be removed, $L_{Preserve}$, is the number the definite and unique cases of a node being preserved, and there cannot exist another case where another node is being removed based on the same attributes of α and β. The Upper Approximation for a node to be preserved, $U_{Preserve}$, is the total number of cases of a node being preserved. The roughness or accuracy coefficient, AC_{Remove} , for the concept of "node to be removed" based on our sample data has a value of 0.74 where is given by:

$$AC_{Remove} = L_{Remove} / U_{Remove} \qquad (5.2)$$

Similarly, $AC_{Preserve}$ has a value of 0.46 based on our sample data and it is given by:

$$AC_{Preserve} = L_{Preserve} / U_{Preserve} \quad (5.3)$$

A low value of $AC_{Preserve}$ is obtained because more nodal removal can be made based on our sample data and that there are many "border-line" cases whereby nodes are being preserved as well as removed based on the same attributes of α and β. Due to the existence of these "border-line" cases, there is bound to have uncertainty on whether a node, with a set of unique attributes of α and β, is to be removed or preserved. Therefore in order to discuss the problem of uncertainty from the rough set perspective, it is necessary to define the rough membership function, $R(X)$, where $X \subseteq N$ and N is the finite set of the nodal sample data. Given that $x \in N$ *and* $A(x) \subseteq X$, the rough membership function is given by

$$R(x) = | X \cap A(i) | / | A(i) | \quad (5.4)$$

Thus the numerical function of the rough membership function for node *n1* for removal, is given by

$$R_{Remove}(n1) = | S_{Remove} \cap \{ n1, n_a, n_b, n_c \ldots\} | / |\{ n1, n_a, n_b, n_c \ldots\}| \quad (5.5)$$

where $\{ n1, n_a, n_b, n_c \ldots\}$ is the set of nodes that is made up of node *n1* and other nodes, $n_a, n_b, n_c, \ldots$, which have the same attributes of α and β, as *n1*.

It is obvious that $0 \leq R(x) \leq 1$. These numerical values of $R(x)$ determine whether a node should be removed or preserved when it has a certain attributes of α and β. Two sets of rules, as shown in equations (5.6) and (5.7), are determined to decide whether to remove or preserve a node during the hole simplification process. A particular node with attributes, says, $\alpha 1$ and $\beta 5$, under the set of attributes, D_{Remove}, will be removed during the simplification process. Similarly, a particular node with attributes, says, $\alpha 1$ and $\beta 1$, under the set of attributes, $D_{Preserve}$, will be preserved during the simplification process.

$$D_{Remove} = \{(\alpha 1, \beta 5), (\alpha 1, \beta 4), (\alpha 1, \beta 3), (\alpha 1, \beta 2), (\alpha 2, \beta 5), (\alpha 3, \beta 2), (\alpha 3, \beta 1), (\alpha 4, \beta 2), (\alpha 4, \beta 1), (\alpha 5, \beta 5), (\alpha 5, \beta 4), (\alpha 5, \beta 2), (\alpha 5, \beta 1)\} \quad (5.6)$$

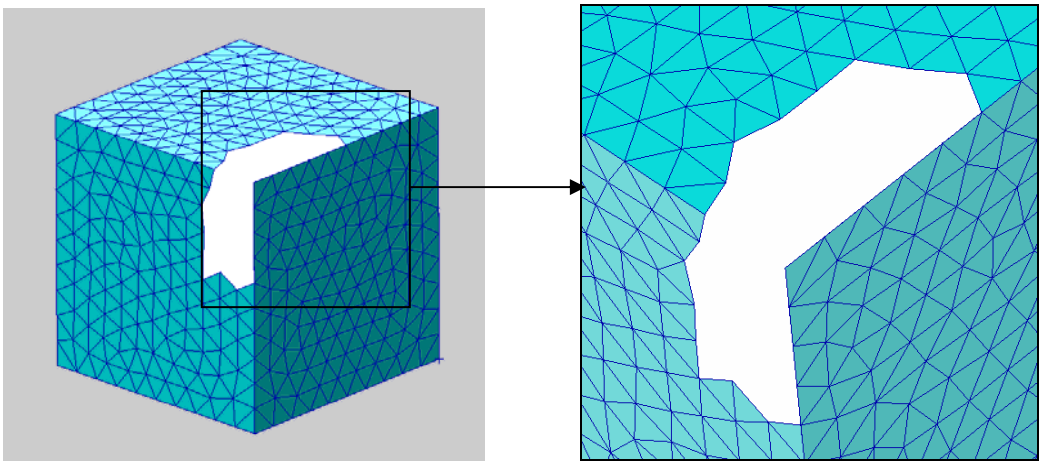
$$D_{Preserve} = \{(\alpha 1, \beta 1), (\alpha 2, \beta 4), (\alpha 2, \beta 3), (\alpha 2, \beta 2), (\alpha 2, \beta 1), (\alpha 3, \beta 3), (\alpha 3, \beta 4), (\alpha 3, \beta 5), (\alpha 4, \beta 5), (\alpha 4, \beta 4), (\alpha 4, \beta 3), (\alpha 5, \beta 3)\} \quad (5.7)$$

For example in Figure 5.5, there consists a primary set of boundary nodes that describe the hole to-be-filled. The task here is to simplify the hole by reducing this set of boundary nodes. In Figure 5.5(c), each of the boundary nodes in red is connected to two adjacent boundary edges that are near or fully collinear with each other. The adjacent boundary triangular elements of these two edges have similar or almost similar face's normal directions. These boundary nodes in red can be classified as the redundant nodes and can be removed from the primary set of boundary nodes for the benefit of hole simplification. Similarly, in Figure 5.5(d), a boundary node in green connects to two adjacent boundary edges that form an acute angle with each other, and that the face's normal directions of their adjacent elements may have high degree of varying directions. This node is

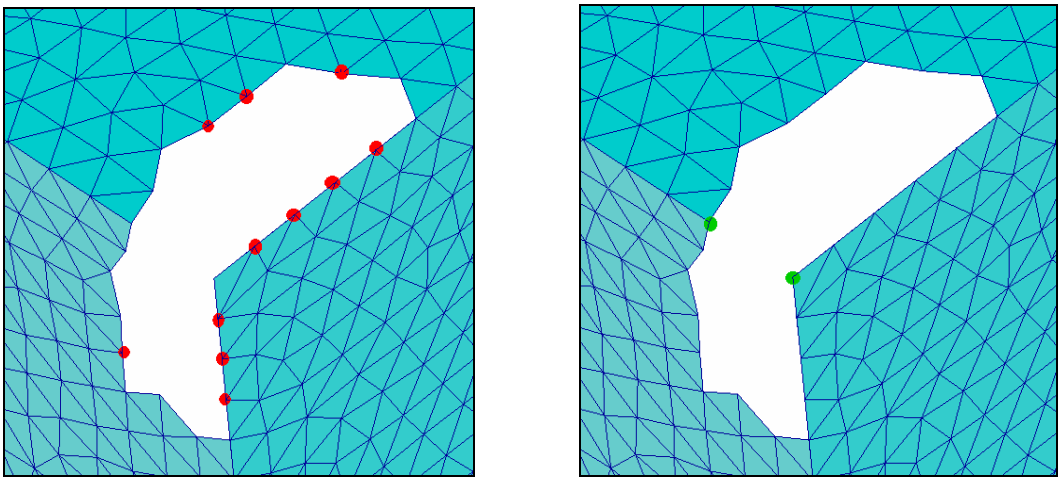
classified as an essential node and will be retained in the set. However, there are many combinations which contain different degree of vagueness in terms of the angles between adjacent boundary edges and the angles between the face's normals of the boundary elements that contain these edges. These vague cases are known as the rough set, which we are going to perform the classification.

By apply this Rough Set theory to the classification of the boundary nodes of a typical hole, a set of essential boundary nodes are obtained to simplify the hole as shown in Figure 5.5(f). This essential set of boundary nodes are triangulated using the Genetic Algorithm (G.A.) described in chapter 5.4 and the result is shown in Figure 5.5(g). Critical edges are important edges that break up the hole into many sub-holes. In Figure 5.5(g), there exists a critical edge which breaks up the hole into two sub-holes. The two sub-holes can then be easily triangulated as shown in Figure 5.5(h) as the shapes of these sub-holes are assumed near-planar. The final mesh can be obtained based on the work in chapter 5.5 on overlaying approximated patches over the hole. The final outcome of the example after the entire hole-filling process is shown in Figure 5.5(i).

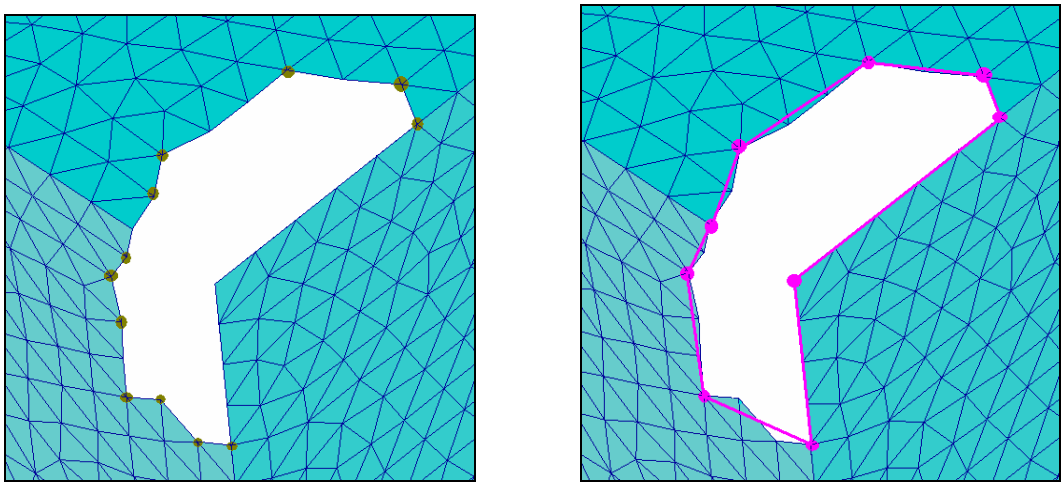
This application of Rough Set Theory essential reduces more than 80% of the time required to create the initial triangulation of the hole using G.A., by reducing the population size of G.A. to a much smaller number, depending on the complexity of the hole.



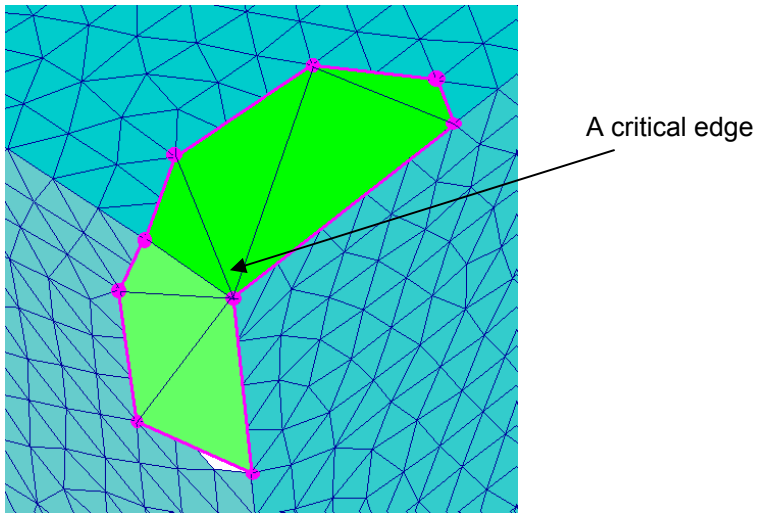
(a) (b)



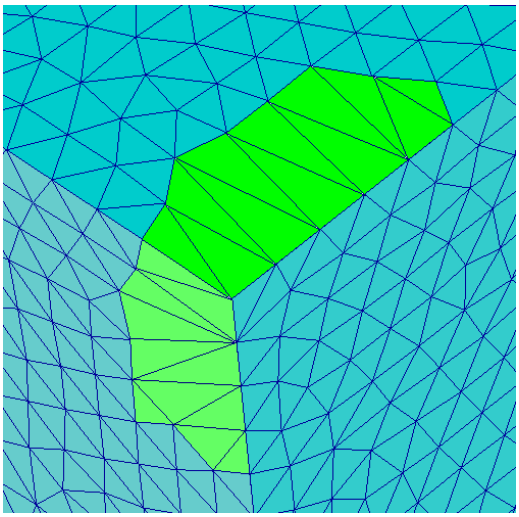
(c) (d)

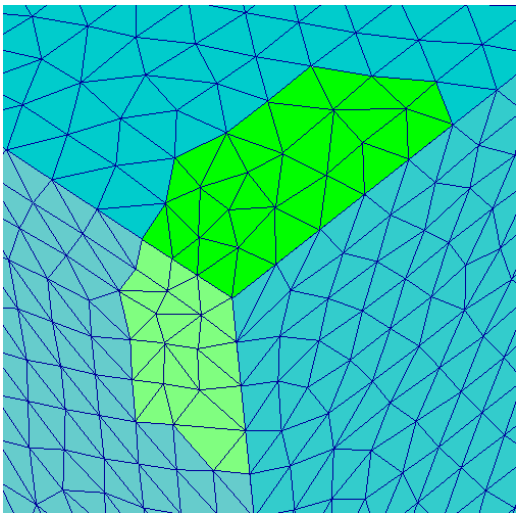


(e) (f)



(g)





(h) (i)

Figure 5.5 An example to illustrate the simplification of a hole using Rough Set Theory which leads to faster triangulation in the subsequent processes

5.4 Initial Triangulation Using Genetic Algorithm

The main aim of the initial triangulation is to cover the hole with a preliminary patch which forms the basis for the approximation of the underlying missing geometry. The triangulation is constrained by the bounding free edges and it

uses all the existing nodes on the boundary of the hole. The initial triangulation links all the nodes on the boundary of the hole to form triangular elements with the constraints of no crossing between the edges of the triangles, and that all edges of newly formed triangles must be inside the hole. The emphasis is to ensure that the triangles transit as smoothly as possible across their edges. Such a requirement is a non-linear multi-objective optimisation problem and it can be suitably solved using Genetic Algorithms (GA). In this section, in order not to confuse the readers, the effort of reducing the boundary nodes of the hole is not implemented. The focus is limited to the generation of an initial triangulation based on all the boundary nodes of a hole using Genetic Algorithm.

5.4.1 Generation of Initial Population

The chromosome in the Genetic Algorithm is represented by a line segment set similar to that in [41], and the length of the chromosome depends on the number of valid segments associated with a particular hole. Table 5.2 shows the total number of possible segments and their combination with respect to the number of free edges in the hole. The total number of possible segments *C* is given by

$$C = (n-3) + \sum_{i=1}^{n-3} i \qquad (5.8)$$

where *n* is the number of free edges present in the hole and *i* is an integer such that {*i* = 1,2,…*n*-3}.

Table 5.2 Construction of line segments between nodes of the hole boundary

		No. of free edges in the hole (n)							
		3	4	5	6	7	8	…	n
No. of segments starting from:	1st node	0	1	2	3	4	5	…	n-3
	2nd node	0	1	2	3	4	5	…	n-3
	3rd node	0	0	1	2	3	4	…	n-4
	4th node	N.A.	0	0	1	2	3	…	n-5
	5th node	N.A.	N.A.	0	0	1	2	…	n-6
	6th node	N.A.	N.A.	N.A.	0	0	1	…	…
	7th node	N.A.	N.A.	N.A.	N.A.	0	0	…	…
	8th node	N.A.	N.A.	N.A.	N.A.	N.A.	0	…	…
	…	…	…	…	…	…	…	…	…
	(n - 3)th node	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	1	2
	(n - 2)th node	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	1
	(n - 1)th node	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	0
	nth node	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	0
Total no. of possible segments (C)		0	2	5	9	14	20	…	$(n-3)+\sum_{i}^{n-3} i$

Depending on the configuration of the boundary of the hole, this initial combination of line segments (Table 5.2) may contain many invalid line segments. To illustrate, Figure 33(a) shows a hole configuration with 6 free edges. According to Equation (5.8), there are 9 possible segments. To extract the valid line segments, the non-admissible segments must be identified and removed. These include segments which are outside the boundary of the hole,

such as segment [2-6] in Figure 5.6(a). The remaining valid segments form the initial population for the GA.

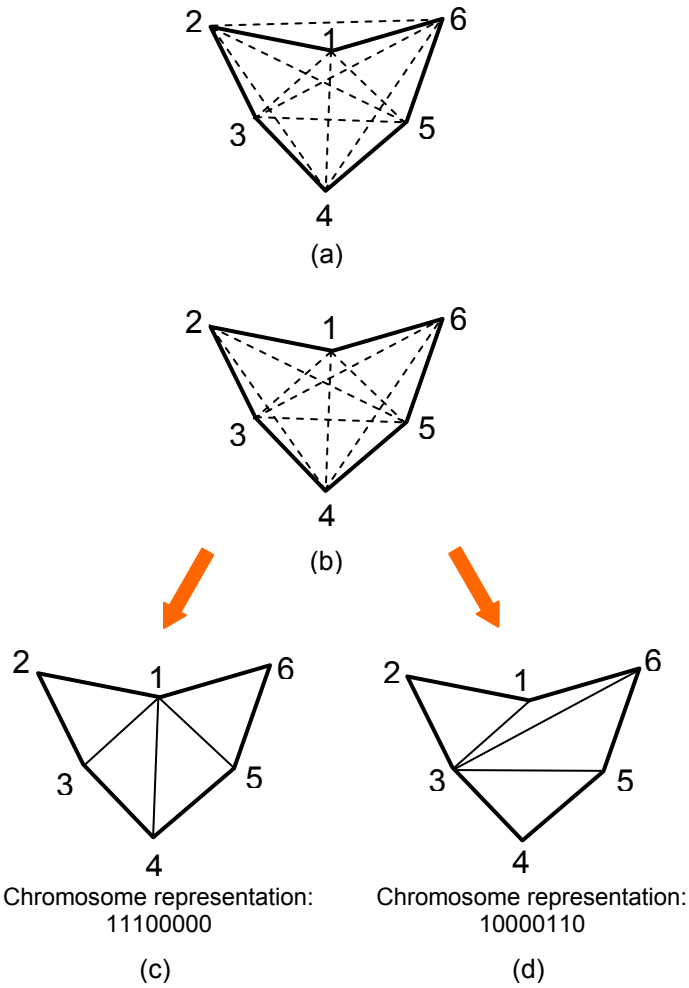


Figure 5.6 (a) Possible line segments for a 6-edged hole configuration, (b) initial population set {[1-3], [1-4], [1-5], [2-4], [2-5], [3-5], [3-6], [4-6]}, (c) possible line segment solution set {[1-3], [1-4], [1-5]} and (d) another possible line segment solution set {[1-3], [3, 5], [3-6]}

In summary, the following steps are taken to extract the initial population:

1. Set any boundary point as the first node and enumerate the rest of the points sequentially in an anti-clockwise (or clockwise) manner, as shown in Figure 5.6(b).
2. Assign the number of possible segments to each boundary nodes

starting from the first node, according to Table 5.2.

3. Construct line segments from each node N_i based on the following criteria:
    a. Connect N_i to a node N_j such that N_j is not an immediate neighbour of N_i, that is, $j \neq i + 1$ and $j \neq i - 1$.
    b. The connected node N_j must be of a higher enumeration than N_i, that is, $j > i$. This will ensure that there will be no repeated definition of line segments.

    In Figure 33(a), the possible line segment set contains the 9 segments {[1-3], [1-4], [1-5], [2-4], [2-5], [2-6], [3-5], [3-6], [4-6]}.
4. Remove all invalid segments which are outside the boundary of the hole, for example, segment [2-6] in Figure 5.6(a). The normal computed along the existing vertices of the polygonal hole will be used to assist in the decision of whether a line segment is outside the boundary of the hole in 3D space
5. Form the initial population using the remaining valid line segment set. In the case of Figure 5.6(b), the initial population set contains the 8 segments {[1-3], [1-4], [1-5], [2-4], [2-5], [3-5], [3-6], [4-6]}. This set will be strictly arranged and that its chromosome length is 8 bits. If all the segments are presented in a solution, the chromosome is “11111111”, and if, say [1-5], is not part of the solution, the chromosome is “11011111”.
6. The initial population is used to construct the chromosome, which is a binary vector of *n* bits of ‘0’s and ‘1’s. After the initial population is defined, all *n* bits of the chromosome are initialized randomly such that

there are (*n* - 3) '1' bits, where '1' represents a possible solution. Figure 5.6(c) and 5.6(d) illustrate 2 possible initial solutions, with chromosomes "1110000" and "10000110" respectively.

5.4.2 Evaluation of Fitness

To obtain optimal solutions, crossover and mutation operations are performed on the chromosome and the new population is evaluated using fitness measures. The crossover can occur in a variety of choices, such as single, uniform and multi-point crossovers [49]. In our work, we use the multi-point crossover to create more perturbation in the solution to prevent the GA from being trapped in a local minimum. The geometric equivalent of the mutation process is an edge swapping operation in the triangulation.

Fitness measures are used to quantify how good a solution is. The "fitter" solutions reproduce and the solutions which are less "fit" die off. The fitness values are calculated by taking into consideration the smoothness of each created triangle with respect to its neighbours. The *Internal Smoothness Fitness Factor* $f_{\text{int}\,ernal}$ is calculated at the links using

$$f_{\text{int}\,ernal} = \frac{1}{(n-3)\pi^2} \sum_{i=1}^{n-3} (\pi - \alpha_i)^2 \qquad (5.9)$$

where α_i is the angle between the normal vectors of two adjacent newly created triangles, as illustrated in Figure 5. The smaller the angle, the larger the value of $f_{\text{int}\,ernal}$.

Similarly, we calculate the *Boundary Smoothness Fitness Factor* $f_{boundary}$ based on the angle between the normal of a newly created triangle and that of the existing adjacent triangle at the boundary using

$$f_{boundary} = \frac{1}{n\pi^2}\sum_{i=1}^{n}(\pi - \beta_i)^2 \qquad (5.10)$$

where β_i is the angle between the normal vectors of the two adjacent triangles along the boundary of the hole, as illustrated in Figure 5.7.

Deviation Fitness Factors are computed to describe the variations of the smoothness for the initial triangulation patches. Equations (5.11) and (5.12) provide the *Deviation Fitness Factors* of α_i and β_i, respectively, such that

$$d_\alpha = 1 - \frac{Max(\alpha_i) - Min(\alpha_i)}{\pi} \qquad (5.11)$$

$$d_\beta = 1 - \frac{Max(\beta_i) - Min(\beta_i)}{\pi} \qquad (5.12)$$

The *Total Fitness Factor* f_{fit} is the summation of all the sub-factors, each multiplied by a weighting factor, as

$$f_{fit} = w_a f_{internal} + w_b f_{boundary} + w_c\left(\frac{d_\alpha + d_\beta}{2}\right) \qquad (5.13)$$

where $w_a + w_b + w_c = 1$. In our implementation, $w_a = w_b = w_c = \frac{1}{3}$.

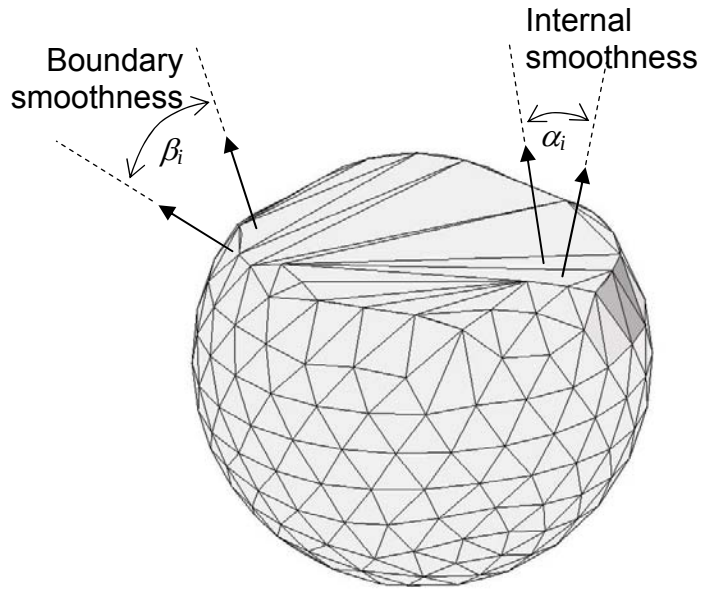


Figure 5.7 Evaluation of fitness factors based on triangle face normal vectors

There are instances where the chromosome is considered totally "unfit" or invalid and the solution is discarded. This occurs when the segments are crossed, that is, two interior line segments intersect each other, such as segments [1-4] and [3-5] in Figure 5.6.

The GA iterates until a terminating condition is met. In our implementation, we set the maximum number of iteration to be 500 and the best solution set is taken as the initial triangulation. Moreover, we allow the user to specify a desired total fitness value (f_{fit} = (0,1)) so that the algorithm terminates when the value is reached. A good empirical value for f_{fit} is 0.7. Figure 5.8 shows the flow chart of the GA triangulation process.

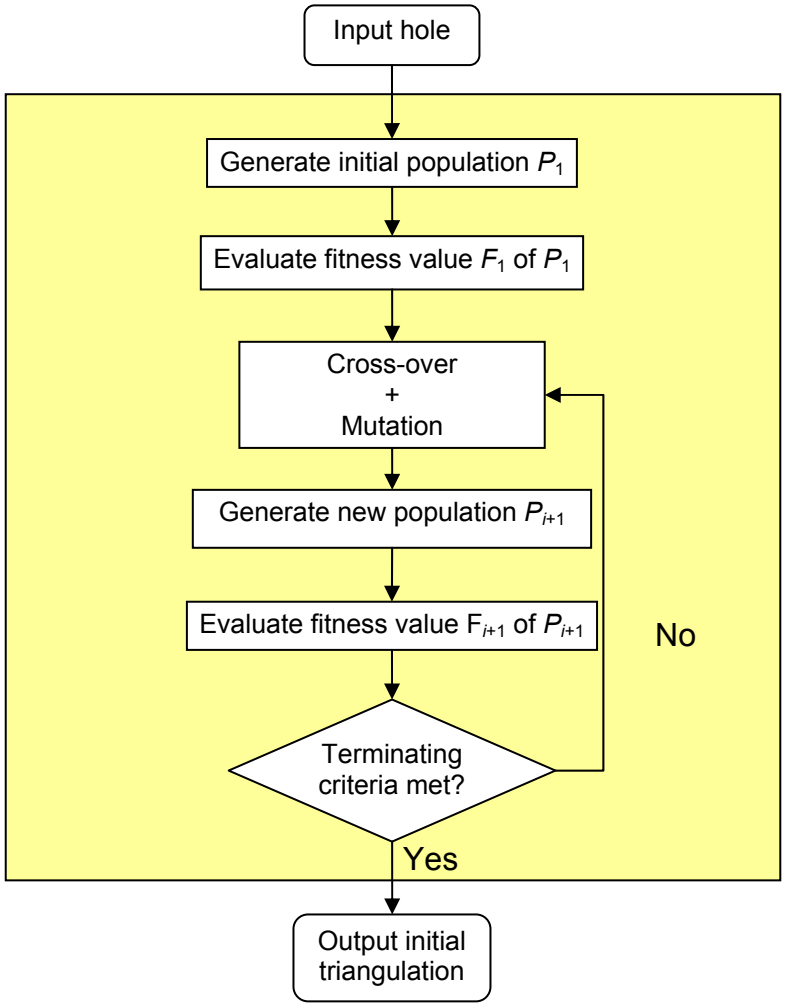


Figure 5.8 Work flow of the initial triangulation process using Genetic Algorithm

5.5 Shape approximations based on quartic Bézier interpolation

After obtaining the initial triangulation using Genetic Algorithm, the shape of the underlying geometry of the hole is approximated using a quartic Bézier interpolation. Essentially, this process uses the information obtained from the initial triangulation, like the coordinates of the triangle vertices and the normal vectors associated with the vertices, to construct a triangular quartic Bézier patch over each triangle. These quartic Bézier patches models the geometric

variations in the vicinity of the hole to provide a localized approximation of the shape. They also serve as geometric basis for the computation of new nodal points in the customized Advancing Front meshing over the hole, which is elaborated in the next section.

For a mathematical understanding of the construction of triangular quartic Bezier patch, we recommend the work by Walton and Meek [37]. For an implementation-friendly version, the reader can refer to Su and Senthil Kumar [42]. Essentially, the interpolation process involves, firstly, approximating the edges of the triangular face using quartic Bézier curves, and, secondly, determining the interior control points of the quartic Bézier patch using a blending function.

A degree *n* triangular Bézier patch *S* is described by Equations (5.14) and (5.15):

$$S(u,v,w) = \sum_{i+j+k=n} B^n_{i,j,k}(u,v,w) P_{i,j,k} \tag{5.14}$$

where

$$B^n_{i,j,k} = \frac{n!}{i!\,j!\,k!} u^i v^j w^k \tag{5.15}$$

and $P_{i,j,k}$ is the control points of *S* and (*u*, *v*, *w*) are the Barycentric coordinates.

The minimum order of a triangular Bézier patch necessary to model a G^1 surface is $n = 4$ [43]. As such, Equation (5.14) can be written as:

$$S(u,v,w) = \sum_{i+j+k=4} P_{i,j,k} \frac{4!}{i!\,j!k!} u^i v^j w^k \tag{5.16}$$

where $u,v,w \geq 0$, $u+v+w=1$ and $i, j, k \geq 0$. Expanding Equation (5.16), we obtain

$$\begin{aligned} S(u,v,w) &= P_{0,4,0}v^4 + 4P_{0,3,1}v^3w + 4P_{1,3,0}u^3v + 6P_{0,2,2}v^2w^2 + 12P_{1,2,1}uv^2w + 6P_{2,2,0}u^2v^2 \\ &+ 4P_{0,1,3}vw^3 + 12P_{1,1,2}uvw^2 + 12P_{2,1,1}u^2vw + 4P_{3,1,0}u^3v + P_{0,0,4}w^4 + 4P_{1,0,3}uw^3 + 6P_{2,0,2}u^2w^2 \\ &+ 4P_{3,0,1}u^3w + P_{4,0,0}u^4 \end{aligned} \tag{5.17}$$

Since the interest is to approximate the triangle with a quartic triangular Bézier patch, the degree of the cubic boundary curve ${}^3C_i(t)$ (as obtained in page 41 in chapter 4.3.1) must be elevated to form a quartic Bézier curve ${}^4C_i(t)$ with control vertices given by

$${}^4v_{i,j} = \frac{1}{4}\left(j({}^3v_{i,j-1}) + (4-j)({}^3v_{i,j})\right) \tag{5.18}$$

where j = {0, 1, 2, 3, 4}. Then the control points of the boundary curves of the quartic triangular Bézier patch are given by

$$P_{o,j,4-j} = {}^4v_{0,j}, \quad P_{j,4-j,0} = {}^4v_{1,j}, \quad P_{4-j,0,j} = {}^4v_{2,j} \tag{5.19}$$

Figure 5.9(a) shows the polar values of a triangular quartic Bézier patch. The surface interpolation problem involves using the point-normal interpolation [44] method to calculate the control points associated with the boundary quartic Bezier curves of the patch, i.e. $P_{o,j,4-j}, P_{j,4-j,0}, P_{4-j,0,j}$, where j = {0, 1, 2, 3, 4}.

To calculate the 3 interior control points $P_{1,1,2}$, $P_{1,2,1}$ and $P_{2,1,1}$, intermediate control points associated with a boundary curve are derived by imposing tangential continuity constraints across that boundary.

5.5.1 Determining interior control points

Next, to define the interior control points $P_{1,1,2}$, $P_{1,2,1}$ and $P_{2,1,1}$ of the quartic triangular Bezier patch, the control points adjacent to a boundary curve are derived by imposing tangential continuity constraints across that boundary. Since each interior control point is associated with two boundary curves, it is determined twice, yielding two different locations *Gi*,1 and *Gi*,2. These are then blended to give the interior control point by ensuring tangential plane continuity across each associated boundary. The locations of *Gi*,1 and *Gi*,2 are obtained by

$$G_{i,1} = \tfrac{1}{8}\left({}^3v_{i,0} + 5({}^3v_{i,1}) + 2({}^3v_{i,2})\right) + \tfrac{2}{3}\lambda_{i,0}w_{i,1} + \tfrac{1}{3}\lambda_{i,1}w_{i,0} + \tfrac{2}{3}\mu_{i,0}A_{i,1} + \tfrac{1}{3}\mu_{i,1}A_{i,0} \quad (5.20)$$

$$G_{i,2} = \tfrac{1}{8}\left(2({}^3v_{i,1}) + 5({}^3v_{i,2}) + v_{i,3}\right) + \tfrac{1}{3}\lambda_{i,0}w_{i,2} + \tfrac{2}{3}\lambda_{i,1}w_{i,1} + \tfrac{1}{3}\mu_{i,0}A_{i,2} + \tfrac{2}{3}\mu_{i,1}A_{i,1} \quad (5.21)$$

where

$$w_{i,k} = {}^3v_{i,k+1} - {}^3v_{i,k}, \quad A_{i,0} = \frac{N_i \times w_{i,0}}{\left\|w_{i,0}\right\|}, \quad A_{i,2} = \frac{N_{i+1} \times w_{i,2}}{\left\|w_{i,2}\right\|}, \quad A_{i,1} = \frac{A_{i,0} + A_{i,2}}{\left\|A_{i,0} + A_{i,2}\right\|} \quad (5.22)$$

The parameters λ and μ are calculated from

$$\lambda_{i,0} = \frac{D_{i,0} \cdot w_{i,0}}{w_{i,0} \cdot w_{i,0}}, \quad \lambda_{i,1} = \frac{D_{i,3} \cdot w_{i,2}}{w_{i,2} \cdot w_{i,2}}, \quad \mu_{i,0} = D_{i,0} \cdot A_{i,0}, \quad \mu_{i,1} = D_{i,3} \cdot A_{i,2} \quad (5.23)$$

where

$$D_{0,j} = P_{1,j,3-j} - \tfrac{1}{2}\left(P_{0,j+1,3-j} + P_{0,j,4-j}\right) \quad (5.24)$$

$$D_{1,j} = P_{j,3-j,1} - \tfrac{1}{2}\left(P_{j+1,3-j,0} + P_{j,4-j,0}\right) \quad (5.25)$$

$$D_{2,j} = P_{3-j,1,j} - \tfrac{1}{2}\left(P_{3-j,0,j+1} + P_{4-j,0,j}\right) \quad (5.26)$$

for j = {0, 3}.

In Figure 5.9(b), $G_{0,1}$ and $G_{0,2}$ are the intermediate control points associated with the first edge; $G_{1,1}$ and $G_{1,2}$ are the intermediate control points associated with the second edge; and $G_{2,1}$ and $G_{2,2}$ are the intermediate control points associated with the third edge. Since each intermediate control point is associated with two boundary curves, a blending function is used to calculate the interior control point such that

$$P_{1,1,2} = \frac{1}{u+v}(uG_{2,2} + vG_{0,1}) \quad (5.27)$$

$$P_{1,2,1} = \frac{1}{w+u}(wG_{0,2} + uG_{1,1}) \quad (5.28)$$

$$P_{2,1,1} = \frac{1}{v+w}(vG_{1,2} + wG_{2,1}) \quad (5.29)$$

where (u, v, w) are the Barycentric coordinates. Now that all the control points are determined, the quartic triangular Bezier patch can be obtained from Equation (5.17).

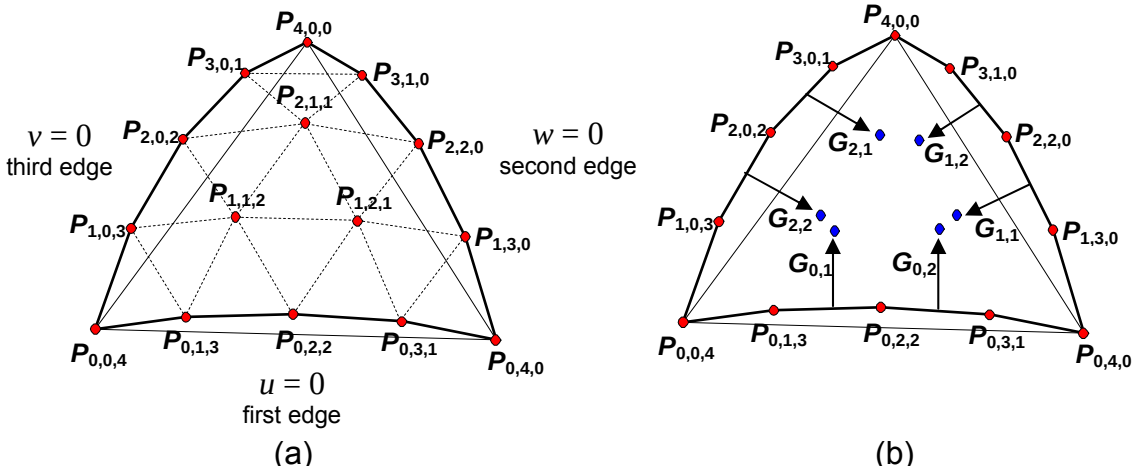


Figure 5.9 (a) Polar values of a triangular quartic Bézier patch and (b) the intermediate interior control points $G_{i,j}$

The required input parameters for the interpolation of each triangle are the coordinates of its 3 nodes and their associated unit normal vector. For the hole-filling problem, the unit normal vector $\hat{n}$ associated with a node on the boundary is calculated according to

$$\hat{n} = \frac{\sum_{i=0}^{m} \alpha_i \hat{n}_i}{\sum_{i=0}^{m} \alpha_i} \tag{5.30}$$

where $\hat{n}_i$ is the unit normal vector of the triangle attached to the node, α_i is the angle subtended by the node and *m* is the number of triangles attached to the node, as shown in Figure 5.10.

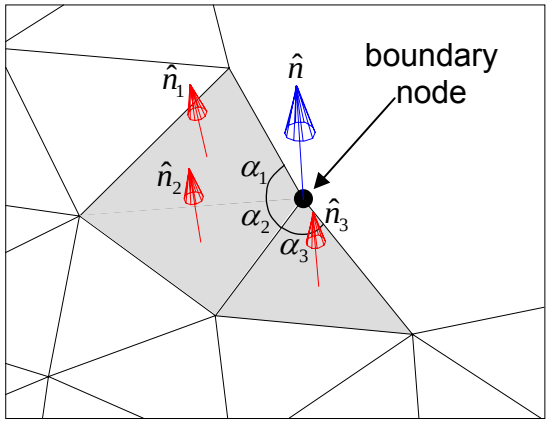


Figure 5.10 Calculation of the unit normal vector associated to a boundary node

5.6 Customised Advancing Front hole-filling technique with projection to Bézier patches

After approximating the underlying geometry using the Bézier patches, a customized Advancing Front meshing technique is used to mesh the hole. The word "customized" is used because the mesh is generated over a non-existing hole, and extra effort is needed to calculate imaginary surfaces where the new nodes within the hole will be projected and sited. A typical Advancing Front technique is described in [45]. New nodal points, which are only temporary nodes, are created during the Advancing Front meshing of the hole as described in the steps below. For each of these temporary nodal points, say, point *X* in Figure 5.11, we find the triangle *T* which contains *X′* by performing a normal projection of *X* onto the nearby triangles. The position of each new node in the mesh generation is calculated based on the geometric definition of the patches, similar to the procedure by Owen *et al* [46]. The Barycentric coordinates of *X′*on *T* are calculated as

$$u = \frac{area(X'BC)}{area(ABC)}, v = \frac{area(AX'C)}{area(ABC)}, w = \frac{area(AX'B)}{area(ABC)} \qquad (5.31)$$

where $T = \Delta ABC$, as illustrated in Figure 9. The interpolated coordinates in the Cartesian space are then calculated based on the Bézier patch associated with *T* using Equation (5.17).

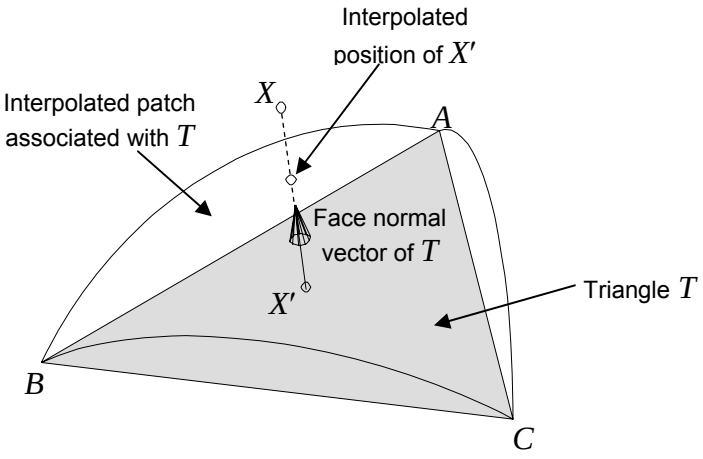


Figure 5.11 Point interpolation mechanism

The customized Advancing Front hole-filling algorithm uses the nodal creation mechanism described above. The steps are as follows:

1. Use the free boundary edges of the hole as the initial front.
2. Calculate the angle θ_i formed by the two adjacent free edges at every node on the front.
3. Starting from the node n_i with the smallest angle, construct new triangles based on the following rules:

a. If θ_i is smaller than 75° (angle α in Figure 5.12(a)), a triangular element is formed by connecting the 2 adjacent nodes (n_{i-1} and n_{i+1}) of n_i, thus introducing a new free edge to the front.

b. If θ_i is between 75° and 135° (angle β in Figure 5.12(b)), two triangular elements are created using the two free edges (e_{i-1} and e_{i+1}) connected to n_i. A new node n_{new} is created by dissecting θ_i such that

$$\left\| n_{new} - n_i \right\| = \frac{\left\| e_{i-1} \right\| + \left\| e_{i+1} \right\|}{2} \tag{5.32}$$

and where n_{new} lies on the plane containing e_{i-1} and e_{i+1}. This introduces two new free edges to the front. Next, n_{new} is mapped to the interpolated location by using the underlying Bézier patches.

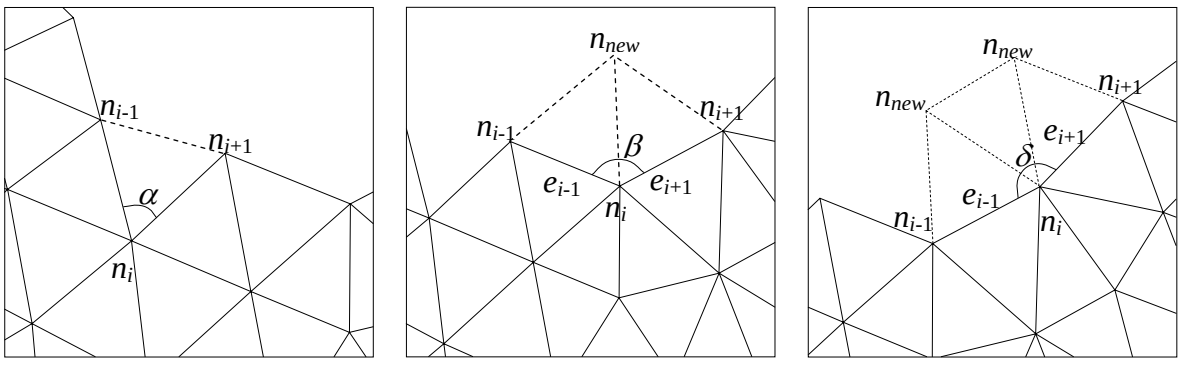
c. If θ_i is larger than 135° (angle δ in Figure 5.12(c)), three triangular elements are created using the two free edges (e_{i-1} and e_{i+1}) connected to n_i. Two new nodes n_{new1} and n_{new2} are created by trisecting θ_i such that

$$\left\| n_{new1} - n_i \right\| = \left\| n_{new2} - n_i \right\| = \frac{\left\| e_{i-1} \right\| + \left\| e_{i+1} \right\|}{2} \tag{5.33}$$

and where n_{new1} and n_{new2} lie on the plane containing e_{i-1} and e_{i+1}. This introduces three new free edges to the front. Next, n_{new1} and n_{new2} are mapped to the interpolated locations by using the underlying Bézier patches.

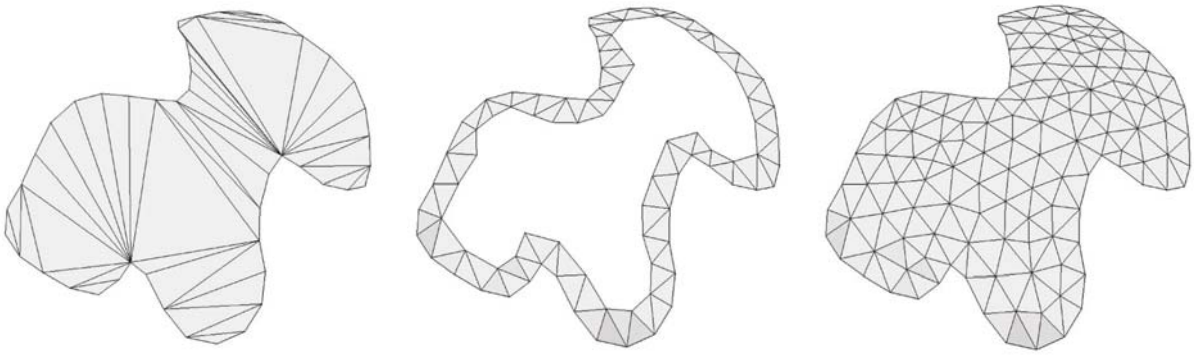
4. Check each newly created node against other existing boundary nodes to see if they are within a tolerance distance of $\frac{1}{4}\|n_{new} - n_i\|$. If so, they will be merged.
5. Map the newly created nodes to the interpolated locations by using the underlying Bézier patches.
6. Update the front.
7. Repeat Steps 2-6 until the whole region is patched with triangular elements.
8. Smooth the created mesh using a mean curvature flow smoothing algorithm [11]. Once again, the interpolated patches serve as guiding surfaces so that the shape is preserved in the mesh smoothing process.

The customized Advancing Front meshing is illustrated in Figure 5.13.



(a) (b) (c)

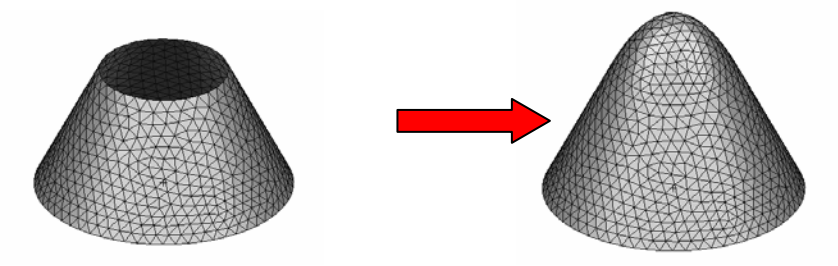
Figure 5.12 Boundary nodes on the front where (a) $\alpha \leq 75^o$, (b) $75^o < \beta < 135^o$ and (c) $\delta \geq 135^o$.



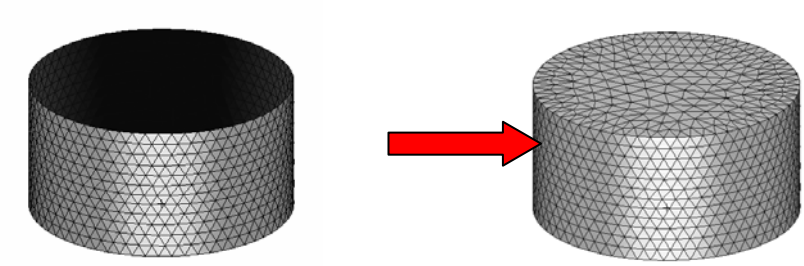
(a (b (c)

Figure 5.13 (a) Region to be meshed using customized Advancing Front method, (b) mesh at intermediate stage and (c) the final mesh.

The current hole-filling algorithm has yet to fully consider topological features, such as edges, as the initial assumption is that the holes to be filled are significantly huge and that topological features are probably difficult to be predictable computationally. However, based on the some restrictions and criterion, as shown in Figs. 5.14(a) and (b), the filling surfaces will very much be dependent on the normals at the vertices of the triangle that are used to calculate the quartic Bézier patch over each triangle.



(a)



(b)

Figure 5.14 (a) Filling of a hole on a conical model and (b) filling of a hole on a cylindrical surface model.

5.7 Results and discussions

The hole-filling algorithm is applied to an impaired sphere with irregular boundary, as shown in Figure 5.15. Here, the initial triangulation using GA plays a significant role to ensure that the triangles transit smoothly across the edges. This ensures that the mesh created by the customized Advancing Front algorithm preserves the underlying shape with high geometric fidelity. Figure 5.16 demonstrates the capability of the hole-filling technique as it handles holes with more complex underlying shape. The triangles forming the boundary of the hole have normals which are directed at widely varying directions. In fact, the underlying shape of the hole consists of regions with both positive and negative curvatures. The holes on this impaired torus

surface model are first triangulated using GA and the underlying geometry is approximated using quartic Bézier patches. The holes are then meshed based on the Advancing Front meshing algorithm with nodal projection onto the approximated Bézier patches.

As the hole-filling in Figures 5.15 and 5.16 are based on standard shapes with analytical geometric definition, error analyses are performed by calculating the absolute geometric deviation of the newly created nodes from the actual surface. The average errors in the E_1 (average error), E_2 (root mean square error) and E_∞ (normalized error) are shown in Table 5.3. Note that the normalized average error values are with reference to the bounding box diagonals of the models. In both cases, the maximum error, as reflected in the E_∞ norm, is less than 1%. This shows that the underlying shape is predicted with good geometric fidelity.

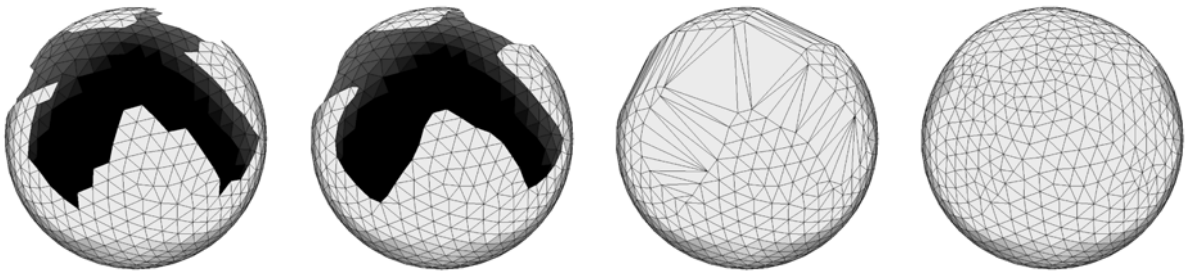


Figure 5.15 (a) Hole on a sphere with complex boundary, (b) boundary after smoothing, (c) initial triangulation of hole using Genetic Algorithm and (d) repaired model of sphere after customized Advancing Front meshing

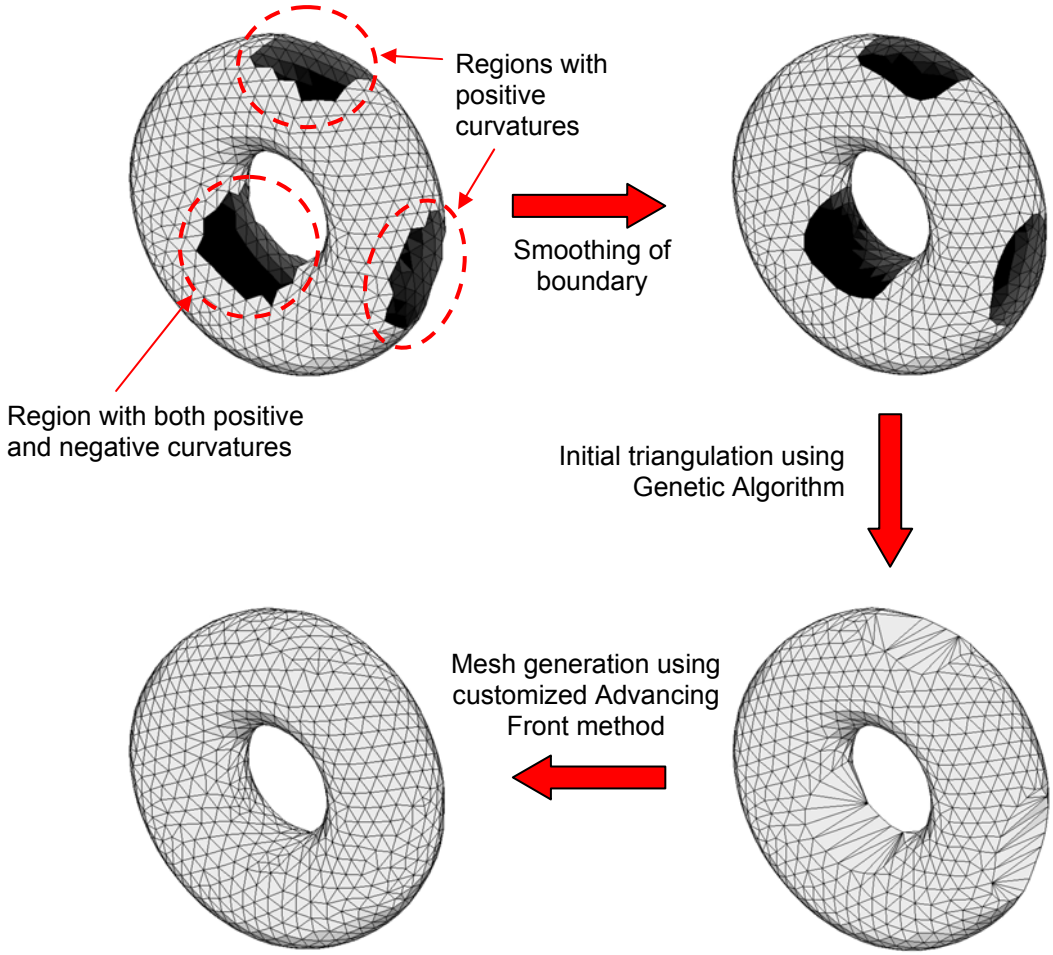


Figure 5.16 Filling holes on a torus

Table 5.3 Error analysis in the E_1, E_2 and E_∞ norm

	Average Error			Normalized Average Error		
	E_1	E_2	E_∞	E_1	E_2	E_∞
Sphere	0.00501	0.00031	0.01173	0.00145	0.000091	**0.00339**
Torus	0.00982	0.00028	0.03866	0.00225	0.000065	**0.00887**

The proposed hole-filling algorithm here is not only meant to be applied to erroneous scanned models. The highlight of this algorithm is that it can fill large complex holes, and at the same time, tries to approximate the shape of

the filled holes as close to the users' intention. For large and complex holes, it will not be adequate if we just perform simple triangulation without any surface approximation.

Chapter 6:

Techniques and Potential Applications using Mesh Repair Algorithms

---

One difficult task in performing research studies is to bridge research with applications. Coincidently, it is discovered that, it is possible to make use of the techniques and algorithms developed in mesh repair to design applications that can be used in area other than the finite element model repair. The Main Contribution to the work in this chapter is to apply the complete hole-filling algorithm that is described in chapter 5, on potential applications in CAD modelling. Two possible avenues to apply the developed techniques are as follows:

6.1 Feature Suppression based on Hole-Repair Algorithm

In many computer-aided engineering applications, it is not necessary to model every detailed geometric feature. One method for idealizing finite element models is to suppress features which are of little meaning and contribution to

the FE analysis. Some features, which are small size and come with complex shapes, often impede the users to perform desirable analysis. To suppress these features, the geometric model is simplified so that the simulation model is more compact and this process is usually tedious and time consuming. Typically, a user would remove these features in the geometric CAD data and then manually repair the model, which is essential due to the removal of these features. This process is usually tedious and time-consuming, and the quality of FE model depends very much on the experience of the user. Feature suppression is also performed in visualization models to reduce the resolution (the number of triangles) of the models.

For model feature suppression, we developed a feature suppression technique via the hole-repair process. Figure 6.1(a) shows a CAD model with features which are to be suppressed. A novel way to achieve this is to simply remove the unwanted surfaces, as shown in Figure 6.1(b), and mesh the remaining model to obtain the simulation model in Figure 6.1(c). The hole in the mesh can then be automatically patched using the automatic hole-filling algorithm described in this thesis to obtain the final mesh in Figure 6.1(d). In Figure 6.2, the hole-filling algorithm is applied to a screw model to suppress the features on the screw face. Figure 6.2(a) shows a screw model with features on the screw face, which are to be suppressed. A novel way to achieve this is to simply remove the unwanted surfaces, as shown in Figure 6.2(b), and mesh the remaining model to obtain the simulation model in Figure 6.2(c).

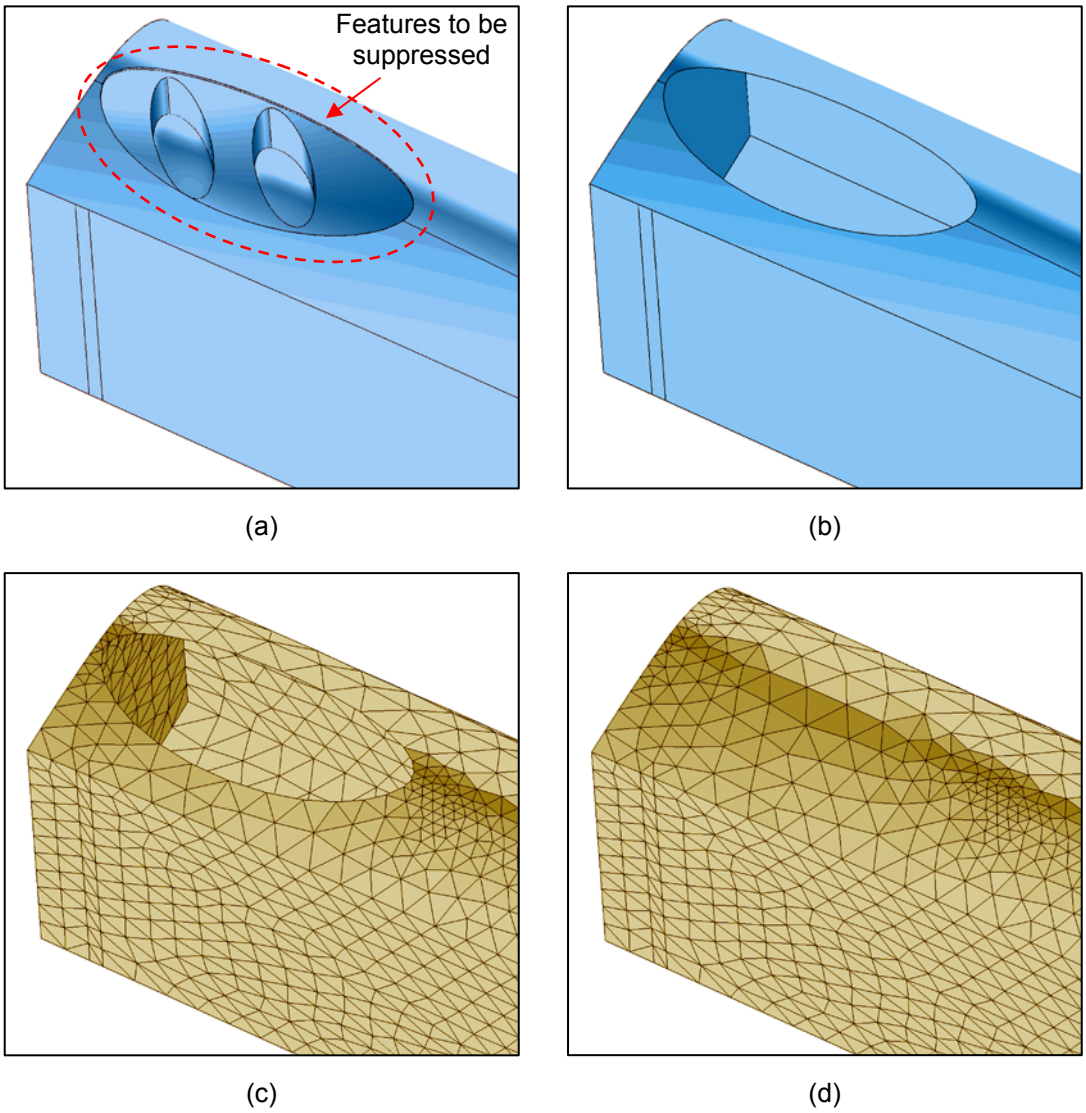


Figure 6.1 (a) CAD model with features to be suppressed, (b) CAD model after feature removal, (c) mesh of incomplete CAD model and (d) final mesh after hole-filling.

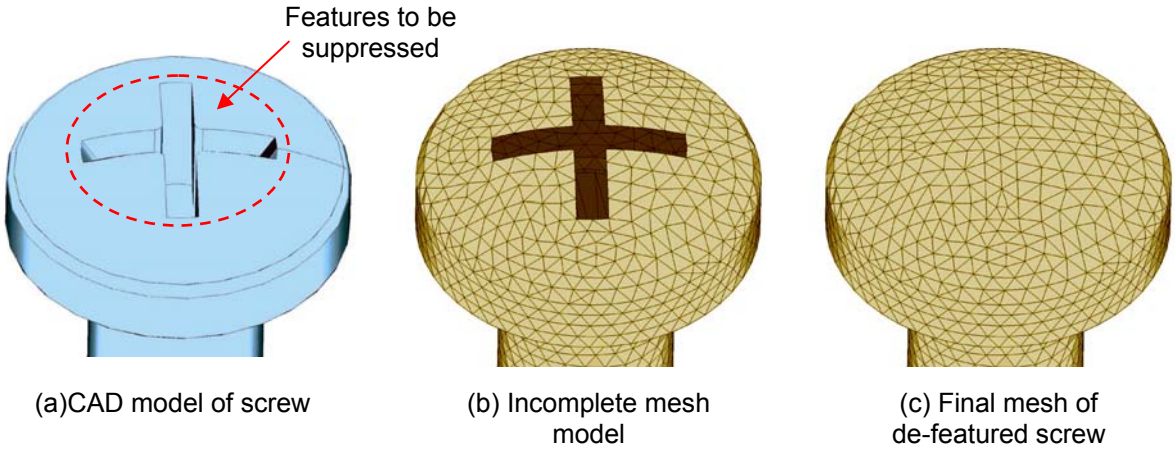


Figure 6.2 Hole-filling algorithms in feature suppression of screw model

6.2 Restoration and reverse engineering of bio-models, artifacts, and the designing of implants.

This research is initially motivated to explore the possibility of automating the process of creating the skull models and cutting and shaping the titanium plates. Surgeons have reported that using Bio-Modeling software to custom shape implants preoperatively can reduce operating time and risk of infection, and improve implant cosmoses and fit. In many cases, facetted/meshed models may be preferred or may be the only representation available to model a machined part. For example, where natural processes are simulated such as in bio-medical or geo-technical applications, NURBS (Non-uniform Rational B-Splines) representations can be difficult or impossible to fit to the prescribed data. For example, when the bone from the hole is missing, damaged or infected, the defect needs to be covered with an artificial plate to protect the brain. Here, the hole-repairing algorithm discussed in the previous section can be re-engineered and employed to aid shaping artificial plates for cranioplasty where a punctured skull model is being repaired. The resulting patching meshes interpolate the shape and density of the surrounding mesh.

6.2.1 Hole filling in Cranioplasty

In many cases, facetted/meshed models may be preferred or may be the only representation available to model a machined part. Surgeons have reported that using Bio-Modeling software to custom shape implants preoperatively can reduce operating time and risk of infection, and improve implant cosmetics and fit. These software import three-dimensional (3D) CT scans and export STL

files which are similar to that of meshed models. Present design of an implant or prosthesis is not always a straightforward application of Computer-Aided Design (CAD) and manufacturing and current Bio-Modelling softwares do not allow for the design of implant when there is no underlying surface. Figure 6.3 shows a BioModel generated from CT scans showing a defect/hole.

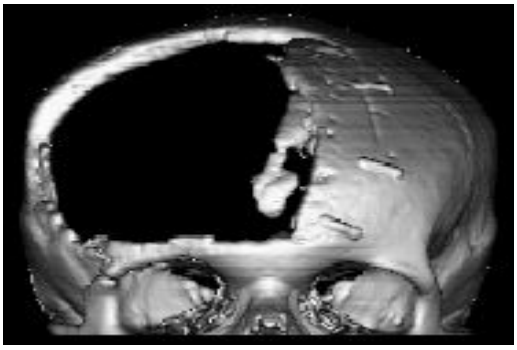


Figure 6.3 3D CT scans demonstrate the extent of the defect of a skull

Cranioplasty is the process of restoring defects, usually holes, in the skull with the aid of cranial implants. Cranioplasty can be due to trauma or they can be made surgically to access some part of the brain. Small defects or holes (5cm^2 to 15cm^2) are generally corrected for aesthetic reasons while large defects (> 25cm^2) require repair for protection of the brain from external injury and for the prevention of cortical thinning, which may cause unnecessary neurological symptoms. Cosmetic and psychological factors are also of importance, and defects in the frontal or temporal bones are restored to hide mutilation. The complexity of cranioplasty depends on the material used to form an implant and upon the size and location of the defect. In general, implants for larger defects (> 50cm^2) are not formed intra-operatively. When the bone from the hole is missing, damaged or infected, the defect needs to be covered with an

artificial plate to protect the brain. The process of making these plates is shown in Figure 6.4. The context for this work is the repair of defects in the skull with cranial implants. In this application, medical graphics are used to reveal bone surfaces in X-ray CT data. By incorporating some geometric functions which smoothly interpolate across holes in the surface, a hard plastic mold can be milled by a computer numerically controlled (CNC) mill. A titanium prosthesis can then be formed by pressing a flat titanium plate into the mold under high pressure in a hydraulic press. The desired characteristics of the fitting mesh patch are therefore as follows: the patch should be a smooth and high fidelity surface; it can interpolate across irregularly shaped holes and that it tends towards a flat plate far from the interpolation centers. Prefabrication of a cranial implant requires an accurate model of the defect area to ensure that a good fit is achieved at the time of surgery and the need for alteration is minimized. There is no standard process for the design and manufacture of prefabricated cranial implants. Each treatment facility may vary in the details of its method, and even then, each case is usually approached individually. However, most methods involve forming a model to which the implant, or a template for the implant, is manually fitted.

In this thesis, we focus on hole-filling with shape prediction for cranioplasty, where missing polygons/surfaces are a common source of holes that awaits implants in bio-models. Here, we customized and applied the prescribed hole-filling technique (described in chapter 5) for filling holes in erroneous or scanned models in the design of customized cranial implants without underlying surfaces by processing a surface mesh representation, where we

extend the incomplete surface description until it forms a watertight or hole-free model. The hole –filling method developed is thus not only limited to repairing polygonal or faceted models containing holes, models produced by data-scanning methods but even creating approximated patches covering the holes that can be used to design custom implants. In some cases the result may not totally match the topology of the original object, but it is always topologically consistent (i.e. manifold), it is not self-intersect, and maintains fidelity to the original data wherever it exists.

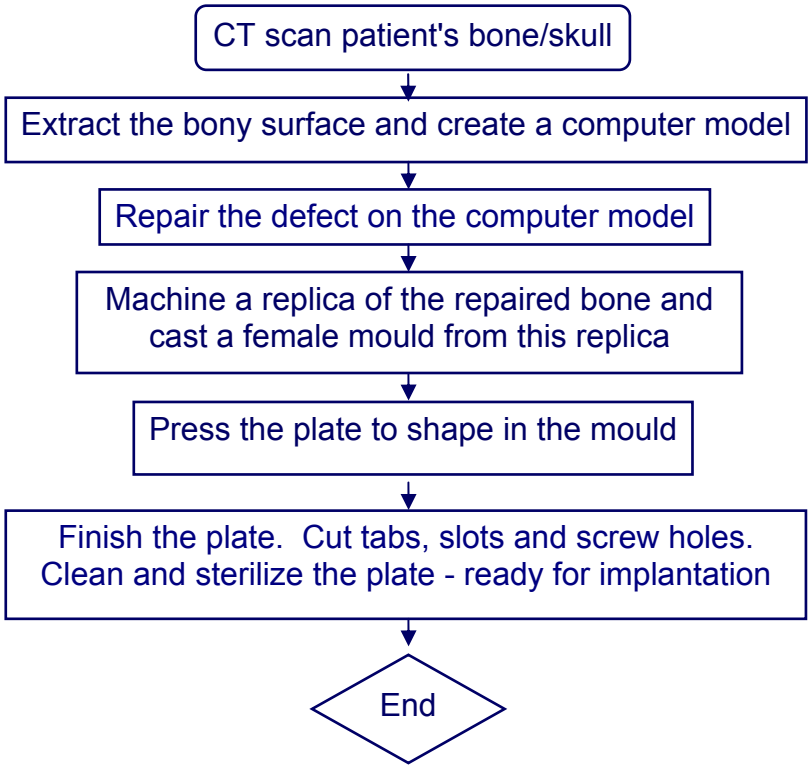


Figure 6.4 Process of making bone/skull's implants

6.3 Results and Discussion

Figure 6.5(a) shows the mutilated mesh of the skull with a significantly large hole on one side of the skull; Figure 6.5(b) shows the optimal triangulation of the hole using GA; Figure 6.5(c) shows the mesh of the skull after hole-filling. It is worthy to investigate that as the hole becomes larger and more complex, more information from the original model is lost, and hence poorer accuracy may be obtained in terms of shape approximation. Figures 6.6(a) and 6.6(b) show the original mesh of a skull and the mutilated mesh of the skull with a large hole respectively. The initial triangulation of the hole based on GA is shown in Figure 6.6(c). The mesh of the skull after hole-filling in Figure 6.6(d) can be used to approximate the design of custom implants to cover the holes, assuming the patient underwent a brain surgery. Note that the original mesh is provided for comparison purpose and they are not present in the computation of the hole-filling process.

The present method can also be used to reconstruct the mesh of the skull with an even larger hole. Figures 6.7(a) shows the mutilated mesh of the skull with the whole of the top portion as a hole. Figure 6.7(b) shows the optimal triangulation of the hole using GA. Figure 6.7(c) shows the mesh of the skull after hole-filling. One result to note from Figures 6.6 and 6.7 is that, as the hole becomes larger and more complex, more information of the original model is lost, and hence one will obtain lower shape accuracy when comparing the repaired model in Figure 6.7(c) with the original model in Figure 6.6(a). In the simulation of firearms injury to the human cranium studied by Mota *et. al.*[47],

the skull model with a small hole defect is investigated here as shown in Figure 6.8. The hole-repair of small and simpler peripheral with small curvatures poses little challenge to the hole filing algorithm.

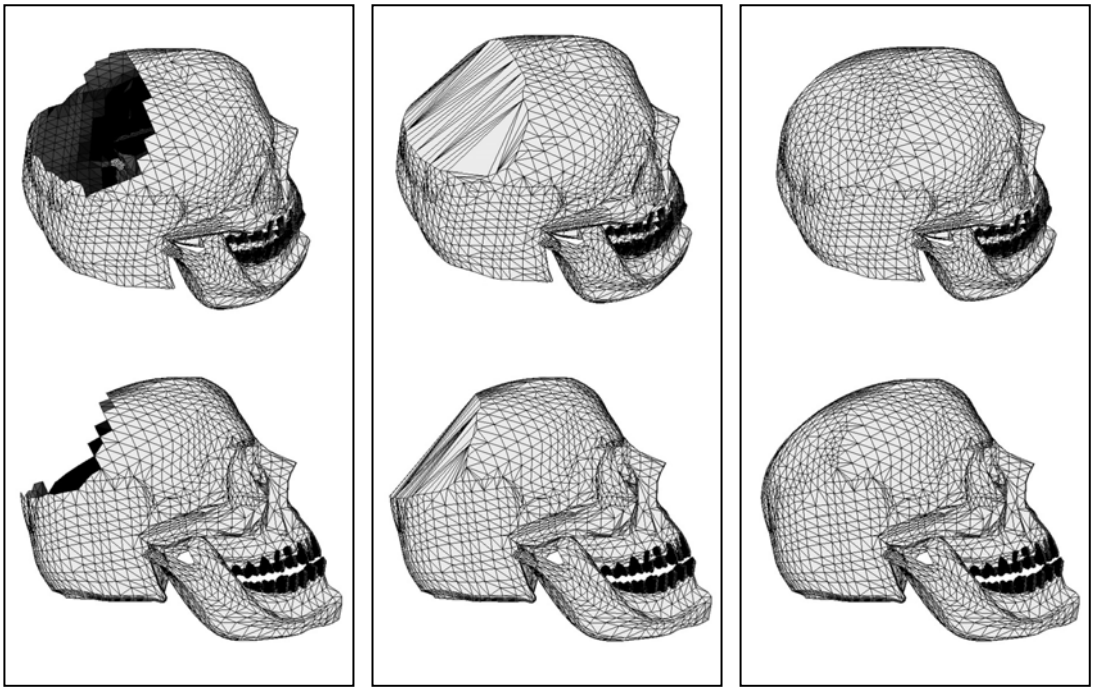
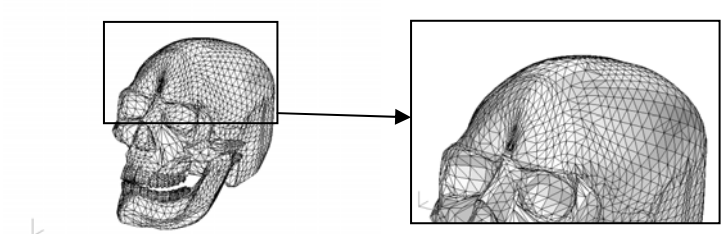
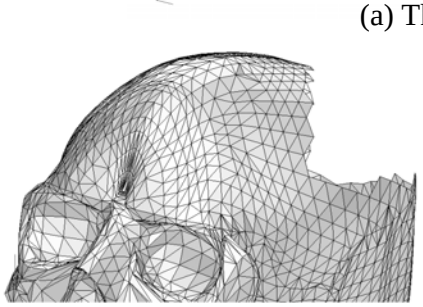


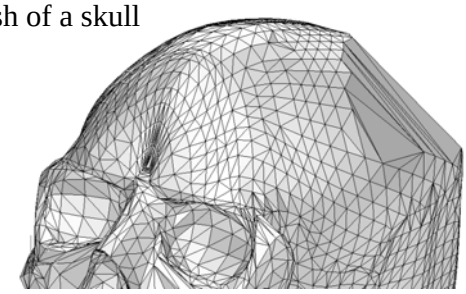
Figure 6.5 (a) Mutilated mesh of the skull, (b) initial triangulation of skull using Genetic Algorithm and (c) repaired model of skull after customized Advancing Front mesh generation.



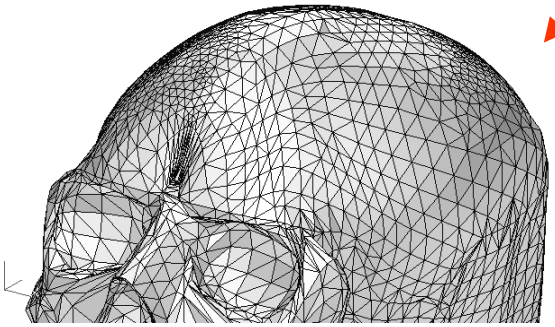
(a) The original mesh of a skull



(b) Mutilated mesh of the skull.



(c) Mesh of the skull after triangulation using GA.



(d) Mesh of the skull after Surface Approximation.

Figure 6.6 Hole-filling for defect in the skull's surface

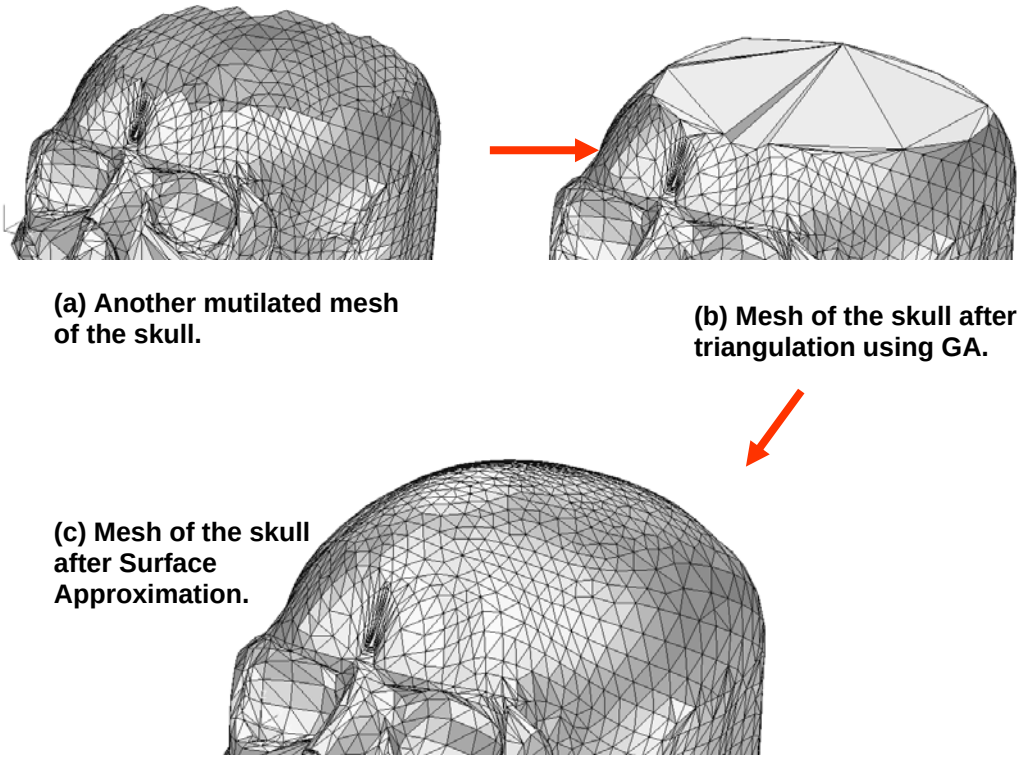
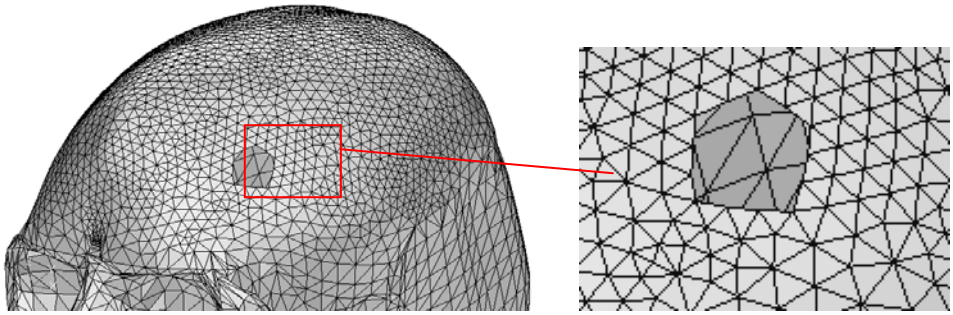
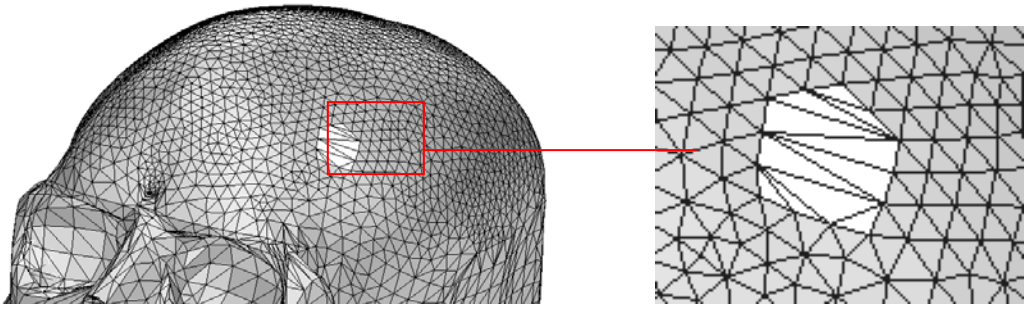


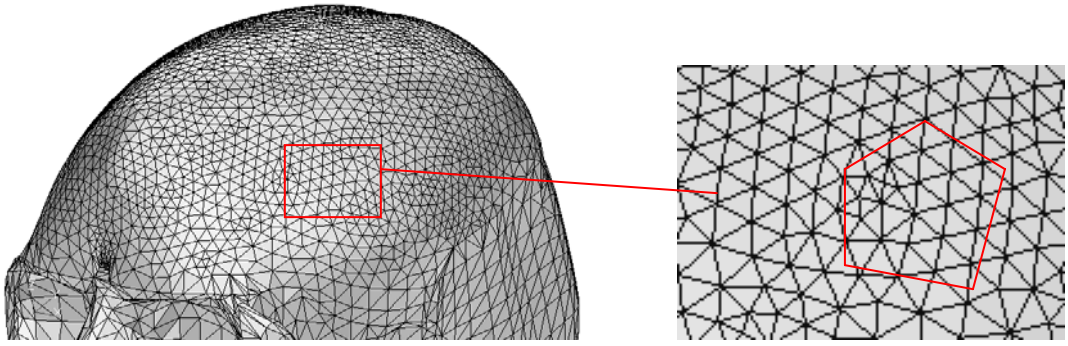
Figure 6.7 Hole-filling for large defect in the skull's surface



(a) Mutilated mesh of the skull simulating firearm injury (hole) to the human skull.



(b) Mesh of the skull after triangulation using GA.



(c) Mesh of the skull after Surface Approximation.

Figure 6.8 Hole-filling for small defect in the skull's surface

Chapter 7:

Mesh Optimization using Biologically-Inspired Algorithms

Genetic algorithms transpose the notions of evolution in Nature to computers and imitate natural evolution. Basically, they find solution(s) to a problem by maintaining a population of possible solutions according to the 'survival of the fittest' principle. In this proposal, we study the main features of genetic algorithms and the ways in which they can solve difficult design problems, such as mesh optimization: mesh reduction and mesh quality enhancement.

7.1 Proposed Methodology

This work proposes and investigates the use of some Biologically-Inspired Algorithms, such as the Genetic Algorithms (G.A.) to perform the large-scale triangular mesh optimization process. This optimization process consists of a combination of mesh reduction and mesh smoothing that will not only improve the speed for the computation of a 3D graphical or finite element model, it will also improve the quality of its mesh. The genetic algorithms (G.A.) are developed and implemented to replace the original mesh with a re-

triangulation process. The G.A. features optimized initial population, constrained crossover operator, constrained mutation operator and multi-objective fitness evaluation function. While retaining features is important to both visualization models and finite element models, this algorithm also optimizes the shape of the triangular elements, improve the smoothness of the mesh and perform mesh reduction based on the needs of the user.

In this chapter, attempts are made to create a mesh optimization system using Genetic Algorithms that would allow the user to create low detail models interactively without significant loss of mesh fidelity. Figure 7.1 shows the workflow of the entire optimization process. 3-D triangulated surface models that are made up of triangles and triangles are the basic elements in finite element modeling. The first step of the optimization process is to remove patches of triangular elements that form empty regions which will be "re-meshed" and optimized subsequently based on a genetic algorithm. There are two methods of removing triangular mesh patches being developed in this research work. These two methods can be executed independently or together depending on whether the user just wants to improve the mesh quality or just wants to reduce the mesh-size of a model or to perform both, depending on his requirements. Often in 3D visualization models, the quality of the triangles is not an issue as compare to the smoothness and the features of the meshed model and the number of triangles present in it. However for finite element models, the shape of the triangle is vital for finite element simulation. Mesh reduction coupled with mesh smoothing is

necessary if the user want to coarsen a very finely meshed model for faster computation.

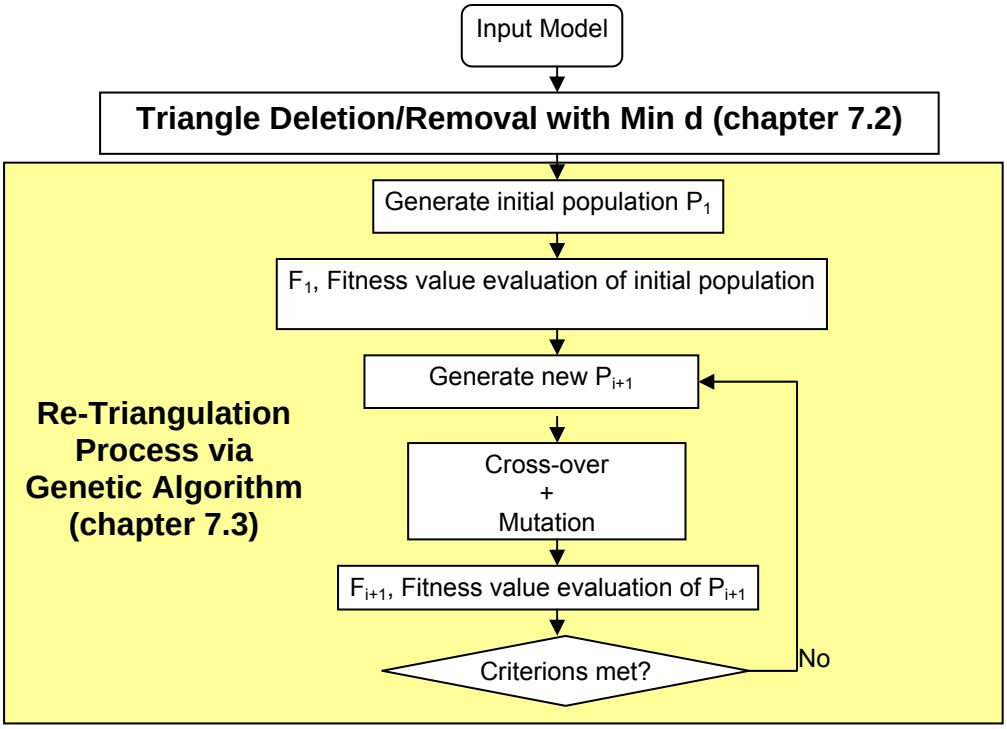


Figure 7.1 Work flow of the mesh optimization process.

7.2 Removal of triangles

7.2.1 Feature Retention

The idea of feature retention is equivalent to the idea of node retention. For example, one would retain the node on the vertex or corner of a box so that the shape of the box would not be altered in the mesh coarsening. Also, during mesh optimisation, it would be desirable to coarsen or smooth the mesh along the edges of the box so that these features are retained in the coarsened or smoothed mesh. The nodes in those regions apart from these features are selected for retention such that they are not too close together. The aim is to

remove the remaining nodes via edge collapsing to achieve larger elements with reasonable aspect ratio.

With this in view, prior to the mesh reduction and smoothing processes, the edges of the mesh in consideration are classified as follows:

- An edge is considered a *feature* if it has only one adjacent face.
- An edge is considered a *feature* if its adjacent faces forms an interior angle smaller than a given angular tolerance.
- An edge is considered *non-feature* otherwise.

After the classification of the edges, the corresponding nodes are classified based on the following:

- A node is *interior* if it does not lie on a feature edge and is removable.
- A node is *fixed interior* if it does not lie on a feature edge and is marked for retention.
- A node is a *removable edge feature* if it lies on a feature edge and is removable.
- A node is a *fixed edge feature* if
    - it lies on a feature edge and is marked for retention, or
    - it is connected to two feature edges with an angle less than a given angular tolerance.
- A node is considered a *vertex* node if it is connected to
    - only one feature edge, or
    - more than two feature edges.
- A node is considered unclassified otherwise.

7.2.2 Maximal Independent Set (MIS)

After extracting the nodes which lie on the important features, it remains to classify all the unclassified nodes into a *maximal independent set* (MIS). The MIS is a set of selected nodes such that those included nodes are not too close in proximity to each other, and those excluded nodes are too close to at least one included node. Figure 7.2 illustrates the MIS of the triangular mesh of a plate. The MIS of the node connectivity graph is simple to implement and is known to give excellent results for one or two levels of coarsening. However, after multiple coarsening, mesh quality degrades significantly, even with mesh optimization between each coarsening. This might pose a problem for meshes intended for analysis purposes.

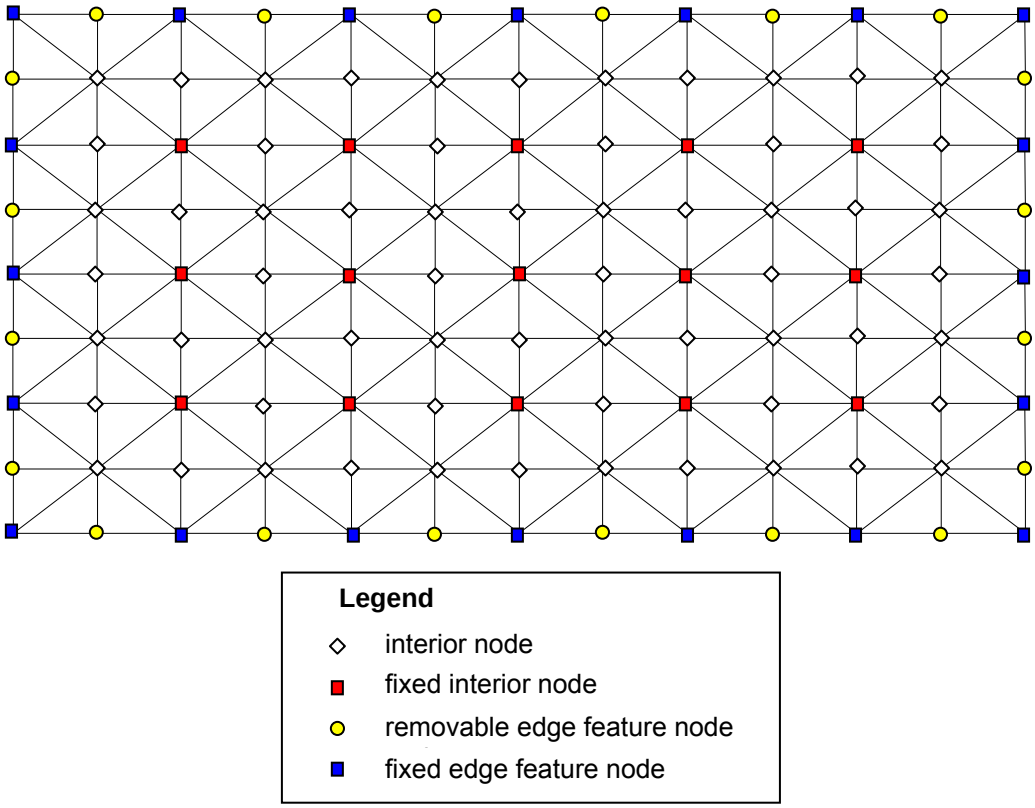


Figure 7.2 Maximal Independent Set (MIS) of a triangular mesh

The reason for coarsening or reducing a mesh based on the MIS rather than a simplistic edge length criterion is so that the inherent solution-adaptivity in the original mesh can be retained. Usually, in those regions where the variables of interest have large gradients, the mesh would be of a higher resolution, as is commonly the case in adaptive meshing techniques. An indiscriminate coarsening through these regions of fine mesh will remove the required resolution. Using the MIS approach, regions which are of a finer mesh will retain the nature of being graded, with finer mesh size relative to the coarsen mesh.

The pseudo code to build a MIS of the whole model is as follows:

```
/* Classify edges */
for every edges
   Calculate the interior angle alpha between the two adjacent
faces.
   if alpha < tolerance angle
      Set the edge type to feature.
   else
      Set the edge type to non-feature.
   end if
end for

/* Classify nodes */
for every node
   if the number of attached feature edges is 1 or more than 2
      Set the node type to vertex.
   else if the number of attached feature edges is 2
      if the angles made by these 2 edges < tol
         Set the node type to fixed edge feature.
      else
         Set the node type to removable edge feature.
      end if
   end if
end for

/* Classify nodes on the feature edges */

nlist_feature = List of nodes that are lying on the feature
      edges

for every nodes in nlist_feature
   Get the list of neighbouring feature nodes = n1.
   if at least 1 node in n1 is a fixed edge feature or a vertex
      Set the node type to removable edge feature.
   else
```

```
      Set the node type to fixed edge feature.
   end if
end for

/* Classify nodes which are not on the feature edges */

nlist_nonfeature = List of nodes that are not on the feature
edges

for every nodes in nlist_nonfeature
   Get the list of neighbouring nodes = n1.
   if at least 1 node in n1 is a fixed node
      Set the node type to interior.
   else
      Set the node type to fixed interior.
   end if
end for
```

7.2.3 Removal of triangles

In this research, two methods of removing triangles are presented as follows:

a. Removal of triangles associated to the nodes of an element

This algorithm for the removal of elements for subsequent mesh re-creation is design for reducing the mesh-size of visualization or a finite element model to speed up processing time. Here we are trying to remove a group of connected elements that are adjacent to the nodes of a specified element. Figure 7.3 shows the surrounding elements that are associated with the nodes (in orange) of the chosen element. A facing possible removal when this element with a unit normal of A is chosen. A weighing factor will decides the sequence and the elements for removal. The calculation of the weighing factor for element removal is as follows:

1. First the deviations of the unit normals of the surfaces of the surrounding elements from the chosen element are calculated using equation (7.1) and (7.2), where A is the unit normal of the chosen

element, B_i are the unit normals of n number of affected elements, and d_i is the deviation of the unit normal of B_i from A with i = 1, 2, 3, …, n.

2. Next the average deviation is calculated using equation (7.3) and the final weighing factor will consist of the total value of the average deviation and the difference in the maximum and minimum value of d_i as shown in equation (7.4). The smaller the value of d, the higher the probability the chosen elements and its affected surrounding elements will be removed. The weighing factor will rank in a hierarchical manner and the user can specify a tolerable value for weighing factor or a percentage of elements with the lowest weighing factor, and usually start off with the element having the lowest weighing factor.

$$AB_i = B_i - A = x_i\, i + y_i j + z_i\, k \qquad (7.1)$$

$$d_i = \sqrt{x_i^2 + y_i^2 + z_i^2} \qquad (7.2)$$

$$d_{avg} = \frac{\Sigma d_i}{n} \qquad (7.3)$$

$$d = d_{avg} + d_{(max\text{-}min)} \qquad (7.4)$$

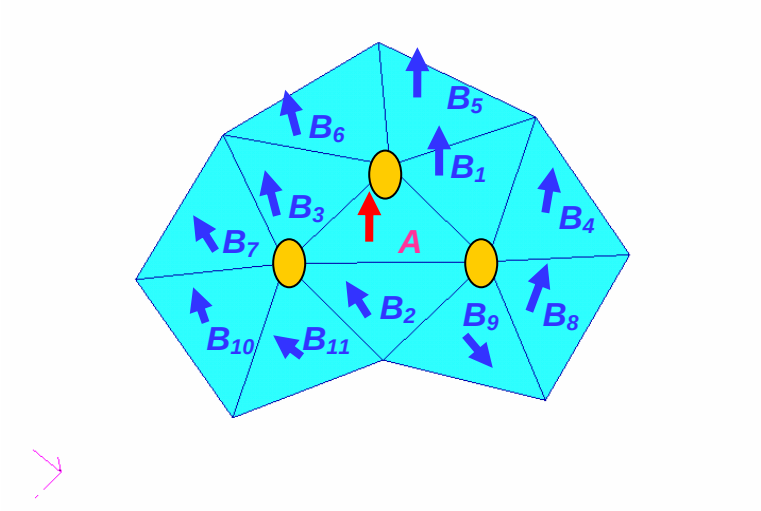


Figure 7.3 Removal of triangles associated to the nodes of an element

b. Removal of triangles associated to the edges of an element

This method for the removal of elements for subsequent mesh re-creation is design for mesh smoothing, in order to achieve quality mesh that is critical for accurate finite element analysis. The number of elements of the mesh will not be reduced. A group of connected elements which are adjacent to the elements of a specified element will be selected to be removed. Figure 56 shows the surrounding elements, including the element with unit normal, A, facing possible removal when this element with unit normal A is chosen. The weighing factor that decides the sequence and the elements for removal is the same as the above using equations (7.1) to (7.4). Note that if any node in Figure 7.3 lies along a feature edge or any edge in Figure 7.4 is a feature edge, the related triangular elements will not be involved in the removal.

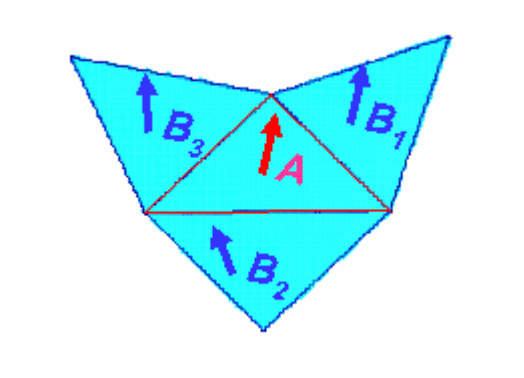


Figure 7.4 Removal of triangles associated to the edges of an element

7.3 Re-Triangulation using Genetic Algorithm

The re-triangulation takes place when a region of elements is being removed. Re-triangulation links all points on the boundary of the empty region to form triangular facets (elements). Figure 7.5 shows an example of re-triangulating an empty region caused by the removal of elements associated to the three edges of a selected element with a low deviation weighing factor. From Table 7.1, we can calculate the total number of possible links and their combination. This table can be computed based on the number of boundary edges or points (nodes) present in the empty region using equation (7.5).

Total number of possible links: $C = (n - 3) + \sum_{i}^{n-3} i$ (7.5)

where n is the number of boundary edges present in the empty region.

The possible links are derived as follow:

7. Set any boundary point as the 1st node and label the rest of the points in

an anti-clockwise (or clockwise) manner as in Figure 7.5(b).

8. Assign number of possible links to each boundary points starting from the 1st node. Node 1 and node 2 will always have (n-3) links and the subsequent nodes will have (n-4) links, (n-5) links,….., and the last two nodes will always have 0 link. In this way, no duplicated links will be present.
9. Next we will describe each link starting from node 1. In Figure 7.5, there are 6 boundary edges present in the empty region. The three links of Node 1 will be [1-3], [1-4], [1-5] as the [1-2] link will be ignored as it is the boundary edge. Similarly for node 2, the [2-3] link will be ignored and the three possible links are [2-4], [2-5], [2-6]. The possible links for node 3 will be [3-5], [3-6] and node 4 will be [4-6]. The last two nodes will not have any link.
10. Check for any invalid links among the possible links. For example in Figure 7.5(b) and (c), [2-6] is an invalid link and will be discarded.
11. So the valid links as in Figure 3 are {[1-3], [1-4], [1-5], [2-4], [2-5], [3-5], [3-6], [4-6]}.

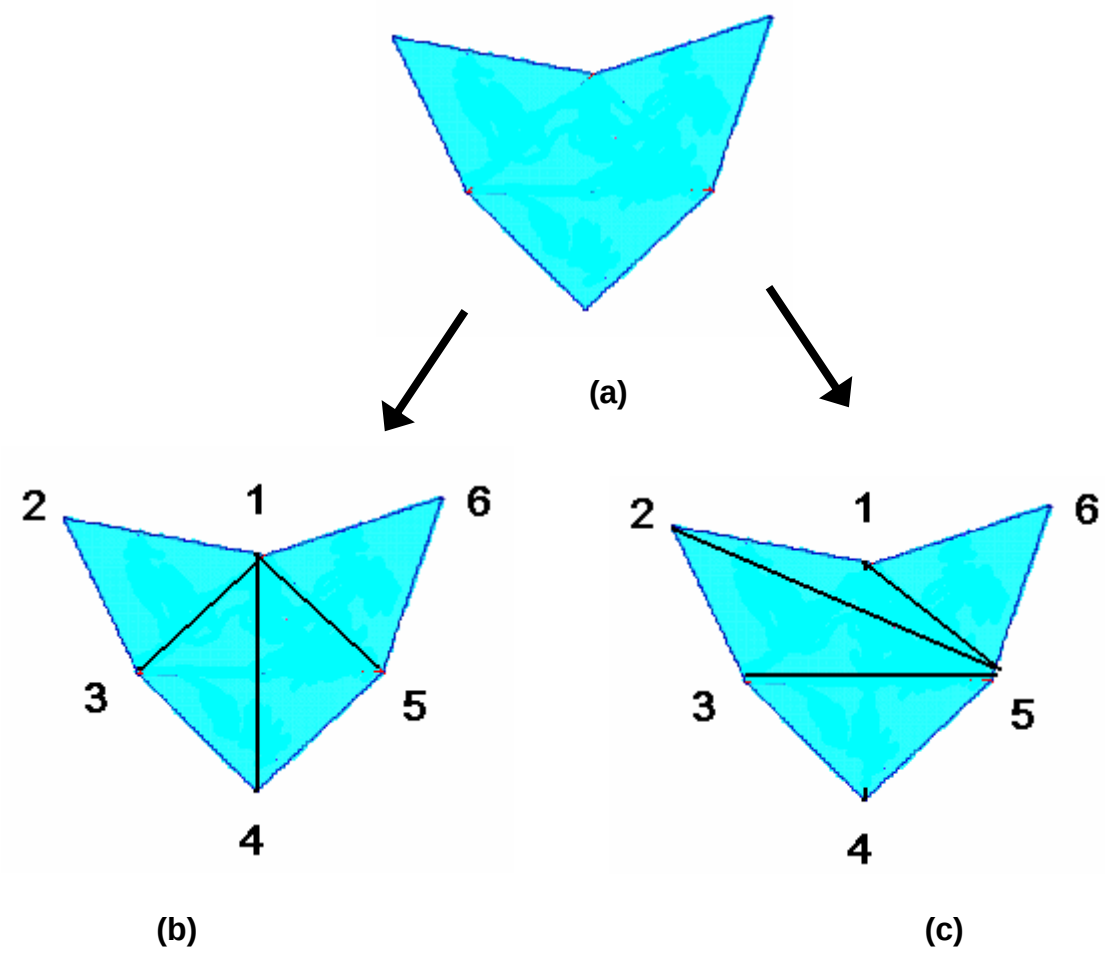


Figure 7.5 (a) Re-triangulation of an empty region, (b) {[1-3], [1-4], [1-5]} links and (c) {[1-5], [2, 5], [3-5]] links

Table 7.1 Formulation of chromosomes

No of edges	3	4	5	6	7	8	…
1st edge	0	1	2	3	4	5	…
2nd edge	0	1	2	3	4	5	…
3rd edge	0	0	1	2	3	4	…
4th edge		0	0	1	2	3	…
5th edge			0	0	1	2	…
6th edge				0	0	1	…
7th edge					0	0	…
8th edge						0	…
…							…
Total no. of possible links	0	2	5	9	14	20	…

Thus in the chromosome representation, 8 binary characters or bits represent the 8 links. The chromosomes will be like 11100000 as in Figure 7.5(b). There will be checks to ensure that the new triangles will not over-lap one another, i.e. no crossings of the links. At any one time during the crossover process, there will only be three 1s. It can be obviously seen from Figure 7.5 that no matter how the crossover takes place, the resulting triangulation always consist of 4 triangles which is the same number as the removed triangle. Thus, the method of removing elements that are associated with the three edges of an element will not result in any change in the number of element but will improve the quality of the mesh during the crossover process based on a fitness test. However for the removal of elements that is associated to the three nodes of a selected element may result in the formation of an empty region with anything equal to or more than 3 boundary edges. This often results in a reduction of elements when we use the genetic algorithm for re-triangulation of the empty regions.

The GA and the solution method taken in this approach to the smoothing process works as follows:

1. The initial population is filled with chromosomes that are generally created at random.

   Each chromosome in the current population is evaluated using the fitness measure. The more fit solutions reproduce and the less fit solutions die off. The fitness value is computed taking in consideration of the shape of the triangles to be created, the smoothness of the triangles to be created and

the smoothness of the triangulated patch with its surrounding triangles. The shape factor will tell us how "equilateral" the triangle is. Each of the triangles created is represented by i from 1 to n-2. Every re-triangulation of an empty region due to the removal of element associated to the selected element's nodes will reduce the mesh-size of the model by 2. θ_j denotes the three angles of each of the triangles created. The larger the value of f_a the better the overall quality of the new triangles created. The calculation of the shape factor is shown in equation (7.6). The overall smoothness of the triangles created is calculated at the links where the angles between the normals of two adjacent triangles are calculated. The smaller this angle is the large the value of f_b in equation (7.7) which denotes smoothness. Similarly for f_c in equation (7.8), we calculate the angles between the normals of the newly created triangles with the surrounding original triangles. The total factor in equation (7.9) is the summation of all the sub-factors, each multiplies by a weighing factor. In the case of reducing the mesh of a 3D graphical model where the triangles' shapes are not critical, the weighing factor of the shape fitness value can set to a lower value. Equations (7.6) – (7.8) will be normalized for better control in equation (7.9).

Shape fitness factor,

$$f_a = \sum_{i=1}^{n-2}\left[\frac{2}{3}\pi^2 - \sum_{j=1}^{3}\left(\theta_j - \frac{\pi}{3}\right)^2\right] \qquad (7.6)$$

Smoothness factor of the created triangles,

$$f_b = \sum_{i=1}^{n-3}(\pi - \alpha_i)^2 \qquad (7.7)$$

where α_i is the angle between the normals of 2 adjacent elements to be created

Smoothness factor of the triangulated patch with its surrounding triangles

$$f_c = \sum_{i=1}^{n} (\pi - \beta_i)^2 \qquad (7.8)$$

where β_i is the angle between the normals of the elements to be created and the unaffected surrounding elements

Total fitness value, $f_{fit} = w_a f_a + w_b f_b + w_c f_c$ (7.9)

2. If the termination criterion is met, the best solution is returned.

   Once an empty region is re-triangulated and improved as much as possible, the GA moves on to the next worst region. This process of moving to the worst area of the model continues until a global model minimum has been reached. Here a global minimum considers the normal deviations for all elements in the model. More work needs to be done in fine tuning the GA's parameters and the convergence criteria since all of the input parameters influence the convergence of the GA.

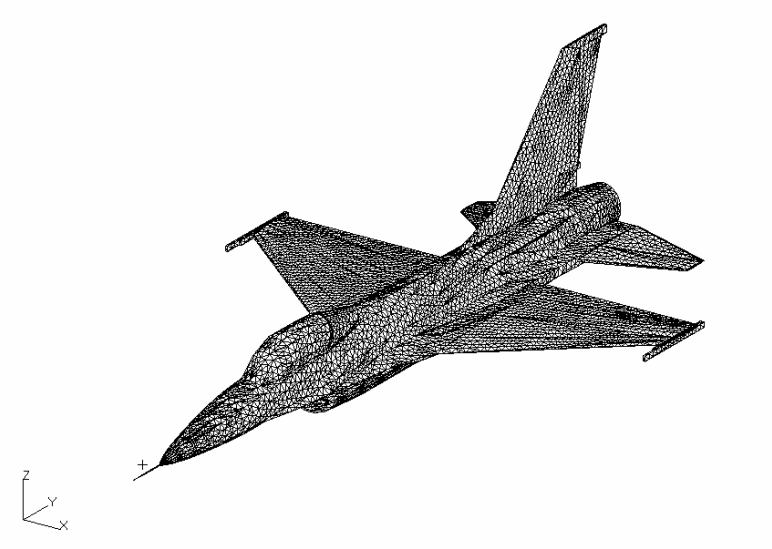
3. Actions starting from step 2 are repeated until the termination criterion is satisfied.

   The crossover can come in a variety of choices. Single, uniform and multi-point crossovers are a few of the types. In this thesis, multi-point crossover is used. Here crossover is used on two mates, producing two children. Mutation adds diversity to the GA. It is a random walk in the search space; basically perform edge swapping in the empty region. The convergence

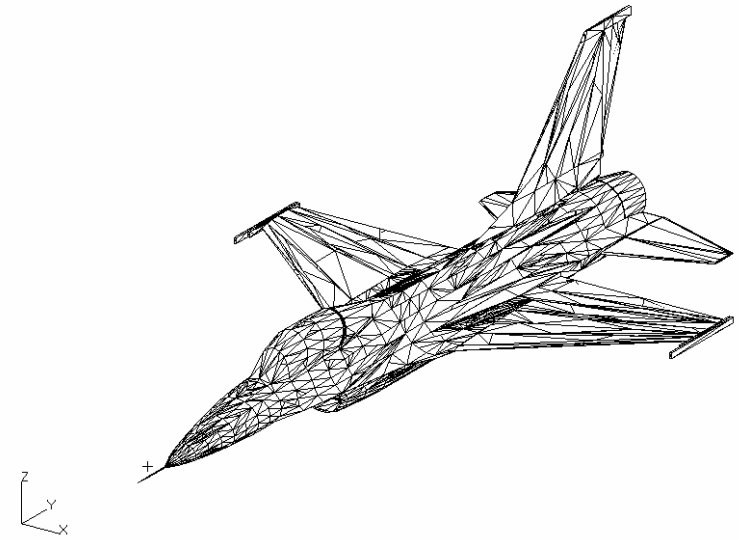
process may be a good application for fuzzy logic. We are not so lucky as to have a simple "go" or "no go" situation. We have a "maybe" or "maybe not" which is a natural for fuzzy logic. The way the GA works, a gradual relaxation of the convergence is required.

7.4 Results and Discussion

A complex aircraft model in Figure 7.6, which contains quite a fair bit of planar regions, demonstrates the capability of the developed GA mesh optimizer. It is to undergo mesh reduction for faster reducing and manipulation in a 3D visual environment. This time, less emphasis is placed on the shape of the triangles but the features of the model that will be retained. Figure 7.6(b) shows the outcome of the algorithm where this model, initially containing 32186 elements, is being reduced to only 3709 elements, an almost 90% reduction in the number of elements. Figure 7.7(a) shows the original polygonal (mesh) model of the heart with 7120 elements. Figure 7.7(b) shows the model being reduced almost 60 % in the number of elements. The fitness's weights associating with the triangles' quality in the mesh is set higher to ensure good mesh quality during mesh reduction. This model of the heart contains very little "almost planar" regions and thus the reduction rate is lower if we choose to preserve the features of the model.

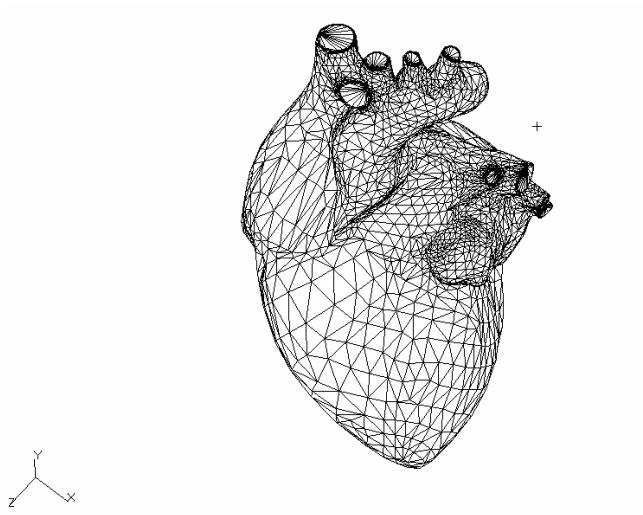


(a)

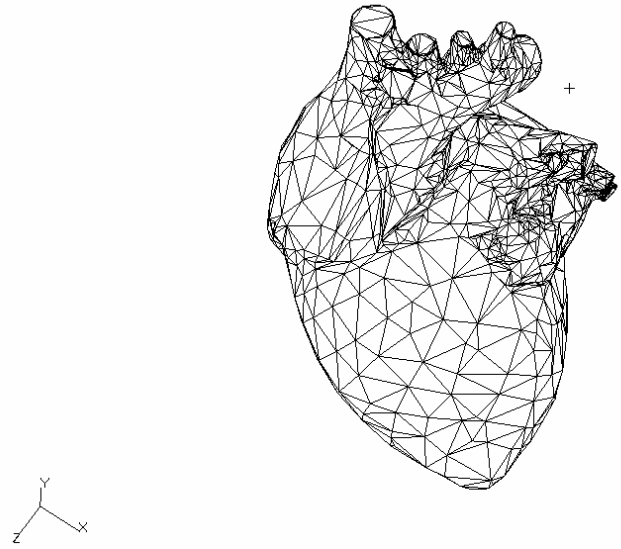


(b)

Figure 7.6 (a) A triangular mesh of an aircraft with 32186 elements, and, (b) the optimized mesh of the aircraft with 3709 elements



(a)



(b)

Figure 7.7 (a) A triangular mesh of heart with 7120 elements, and, (b) the optimized mesh of heart with 3128 elements

Chapter 8:

Case-Studies

Two case-studies will be discussed in order to demonstrate the mesh repair algorithms that are developed in this research work. The complete meshing solution consists of two components, namely the mesh repair component and the mesh optimization component.

The test-cases will undergo the process flow of mesh repair and optimization that is illustrated in Figure 3.3 in chapter 3. First each test-case will undergo an error identification process to check mesh validity and identify geometrical and topological errors that are present in the model. The sequence of repairing different errors is important, for example, the problems of gaps and overlaps will be handled before repairing the holes, which are usually much bigger in size by nature. The mesh that is resulted from the repair processes is likely to contain poor quality elements, even after sliver repair. The mesh optimization component will help to further improve the quality of the mesh after it has been repaired. It will take care of degenerated elements, mesh smoothness and other meshing requirements such as mesh resolution and mesh adaptivity. The mesh optimisation component can also help to reduce

the mesh-size drastically to speed up the visualization process of the model, which does not take into consideration of the quality of the triangular elements in the model.

8.1 Case-study 1

In Figure 8.1, a defective model of a work-piece is being presented. This geometrical model contains several errors such as, gaps and holes. The gaps are probably the results of tolerance or truncation issues. The holes are usually present due to missing surfaces. Figure 8.1(a) – (d) show some errors inherent to this defective work-piece model. Note that since this model is a geometrical model, a mesh is being generated over this defective model as shown in Figure 8.2(a). Figure 8.2(b) – (d) show some zoomed-in regions containing gaps and holes. In Figure 8.2(e), with the implementation of the developed gaps repair algorithm along with T-joint repair algorithm, the affected gap is being closed forming water-tight regions using algorithms described in chapter 4. In Figure 8.3, the mesh is being repaired, and the affected holes are being filled using the developed hole-filling algorithm and sliver repair algorithm as described in chapter 4 and 5.

In Figure 8.4, an attempt is being made to look at the feature suppression capability as discussed in chapter 6. A bolt feature is being removed from the model in Figure 8.4(b). Making use of the hole-filling algorithm described in chapter 5, this feature hole is being patched up in Figure 8.5.

At the tail-end in mesh optimisation, this repaired mesh undergoes two operations separately. Using the mesh optimisation algorithm developed in chapter 7, the first operation, as shown in Figure 8.6, focuses on the quality of the triangular coupled with minimal triangulation reduction. The result of this first operation in this example shows a reduction of 22% in terms of mesh size with constancy in the mesh quality. Figure 8.7 shows the second mesh optimisation operation based on a tweak-able set of evaluation variables. This operation focuses heavily on mesh reduction with no or little consideration for mesh quality. This is especially useful in optimizing mesh for visualization purposes. The example shows an 83% reduction in the mesh-size of the work-piece.

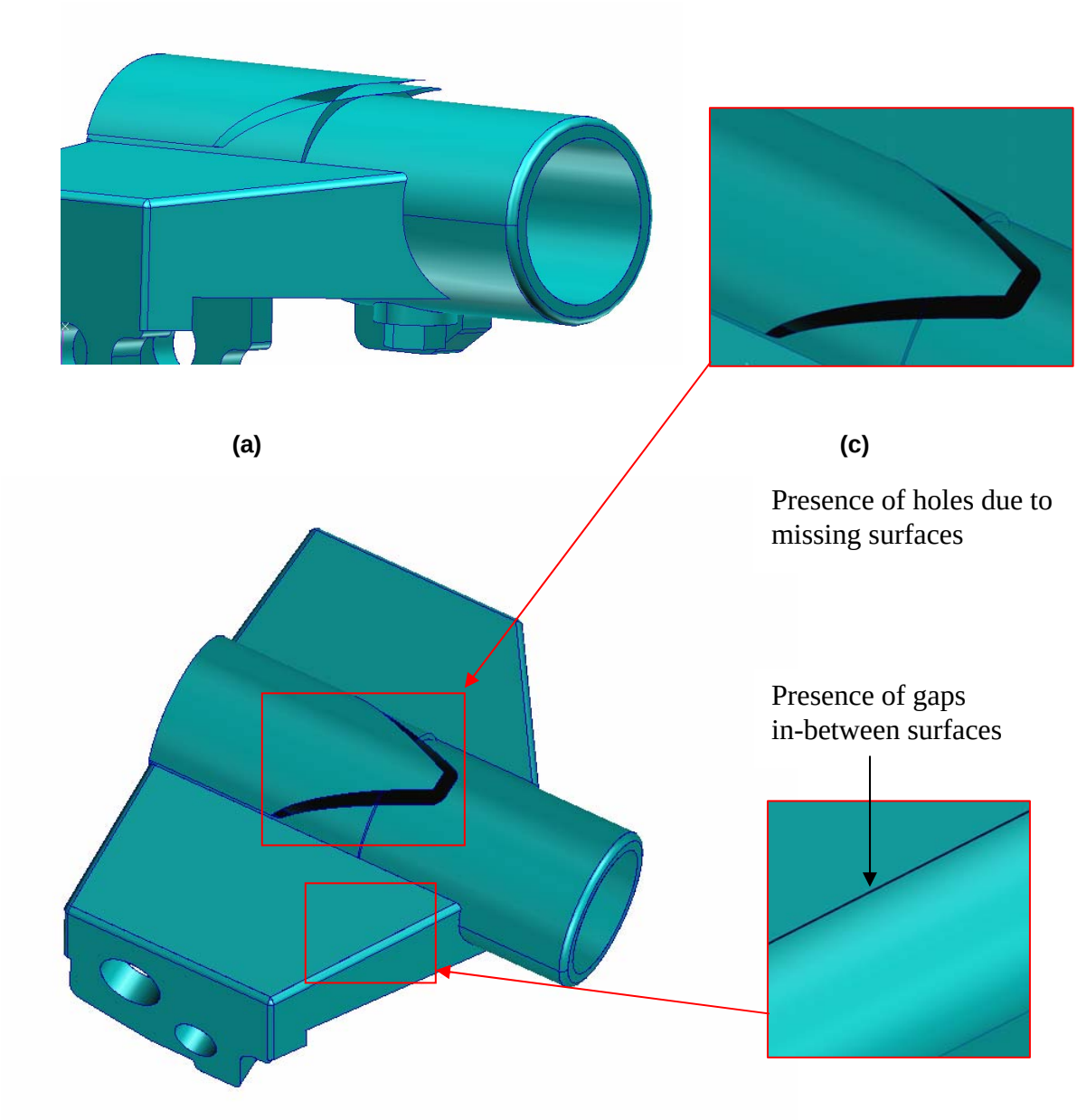


Figure 8.1 (a) View 1 of a defective geometrical mode of a work-piecel, (b) View 2 of the defective geometrical model, (c) Presence of holes in the model due to missing surfaces, and, (d) Presence of gaps in-between surfaces due to tolerance and truncation errors.

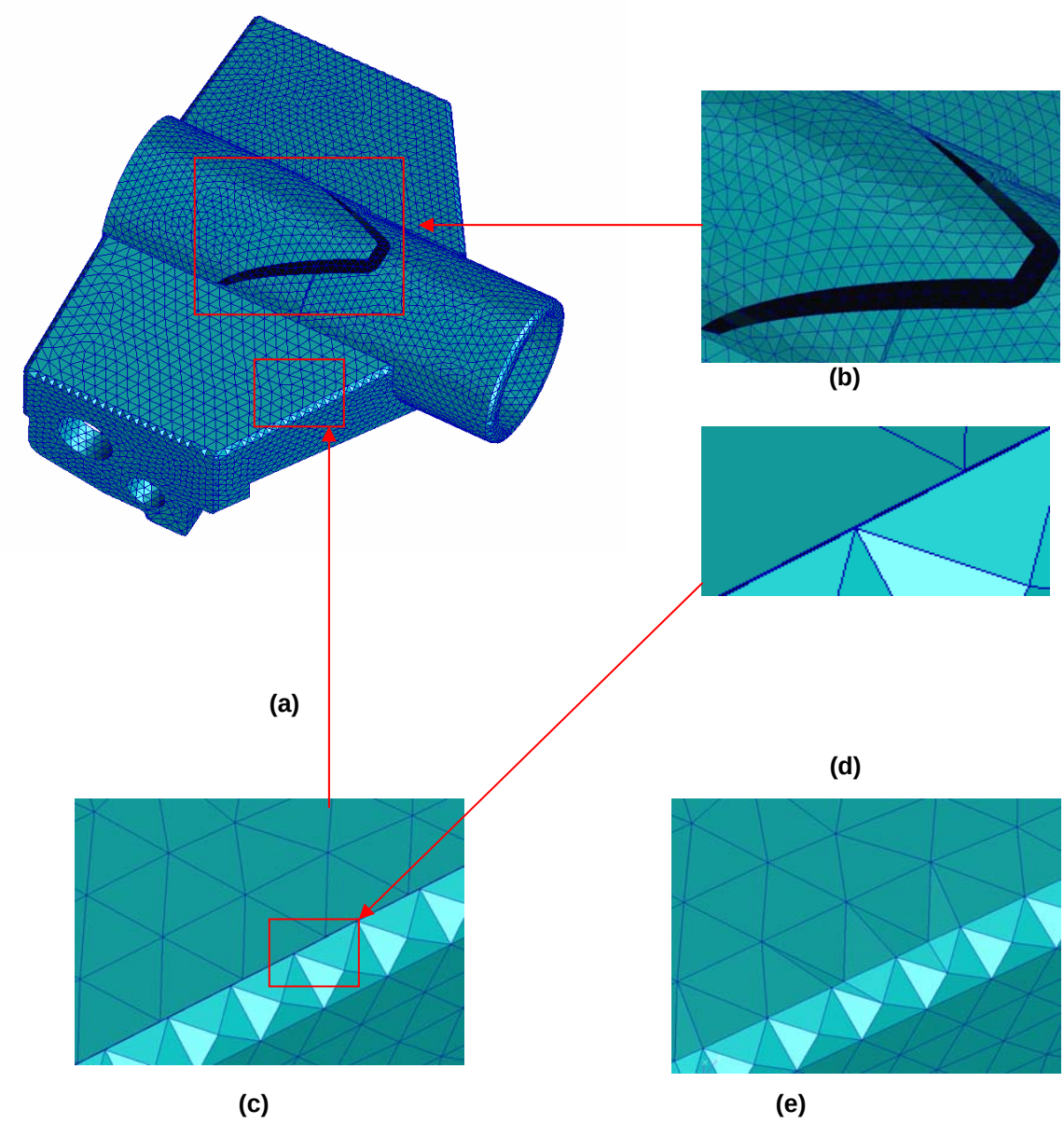


Figure 8.2 (a) Mesh generated for the defective geometrical model, (b) The mesh of the hole region, (c) The mesh region along a gap, (d) The zoomed-in mesh region along the gap and, (e) The mesh along the gap after mesh repair.

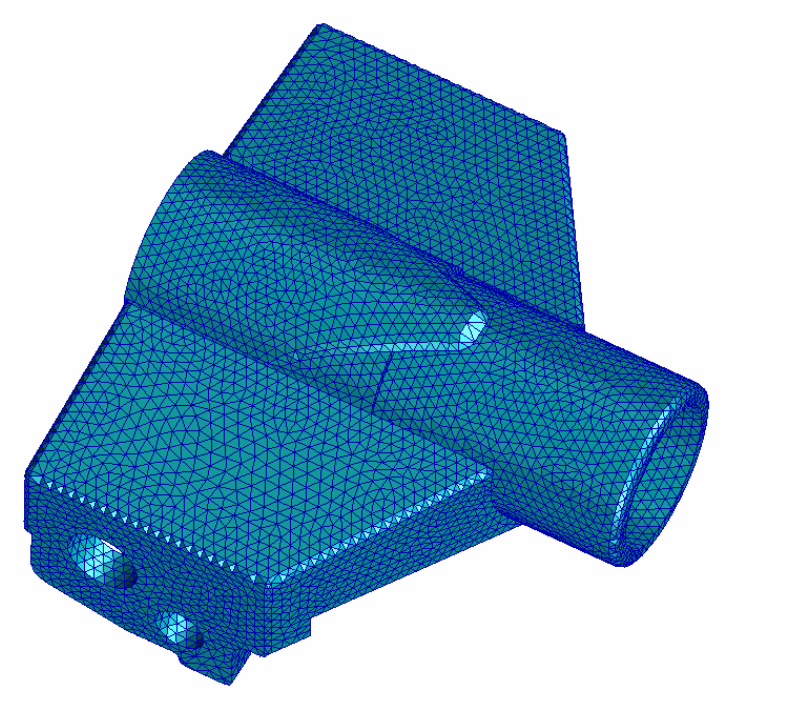
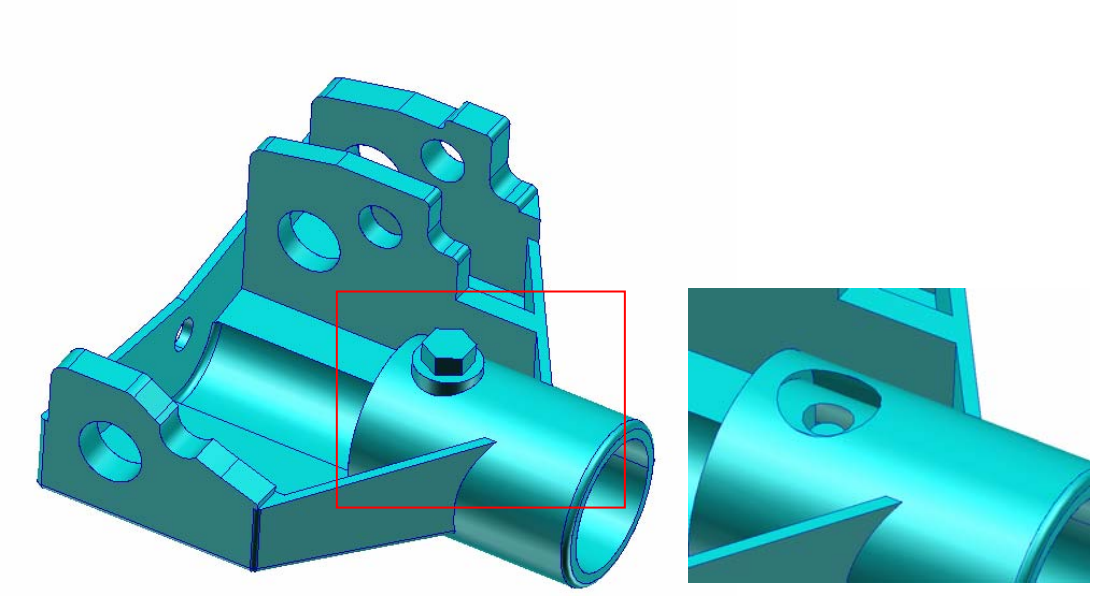
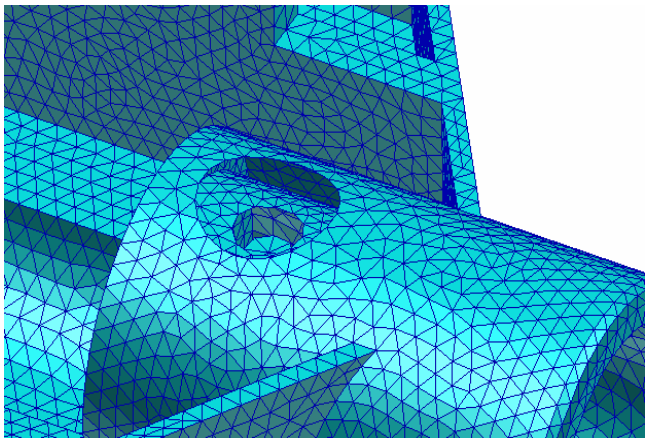


Figure 8.3 The final repaired mesh after hole filling with 24375 triangular elements.

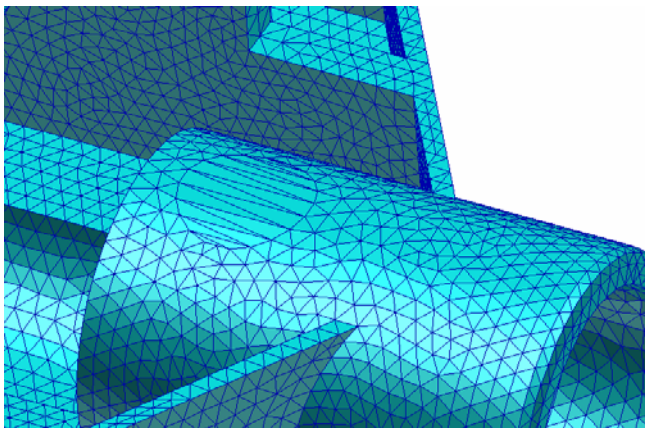


(a) **(b)**

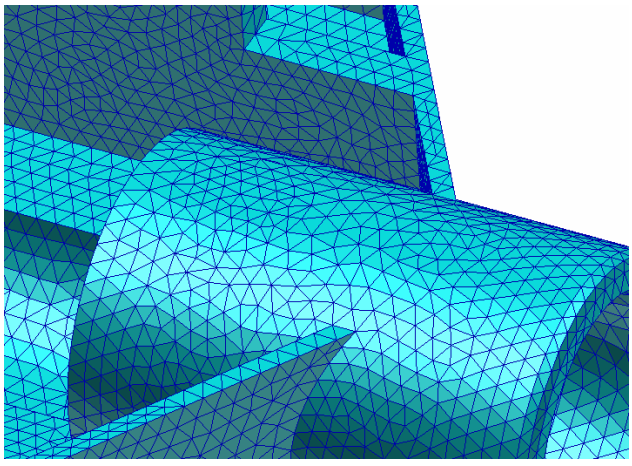
Figure 8.4 (a) View 3 of the defective geometrical model, and (b) A bolt feature is removed from the geometrical model forming a hole.



(a)



(b)



(c)

Figure 8.5 (a) The mesh at the hole region, (b) the initial triangulation of hole based on Rough Set and G.A., and (c) the final mesh of the hole after being filled based on surface approximation.

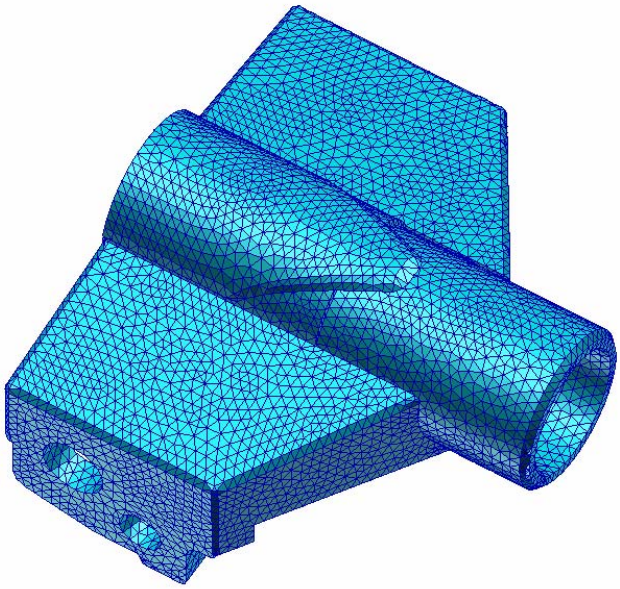
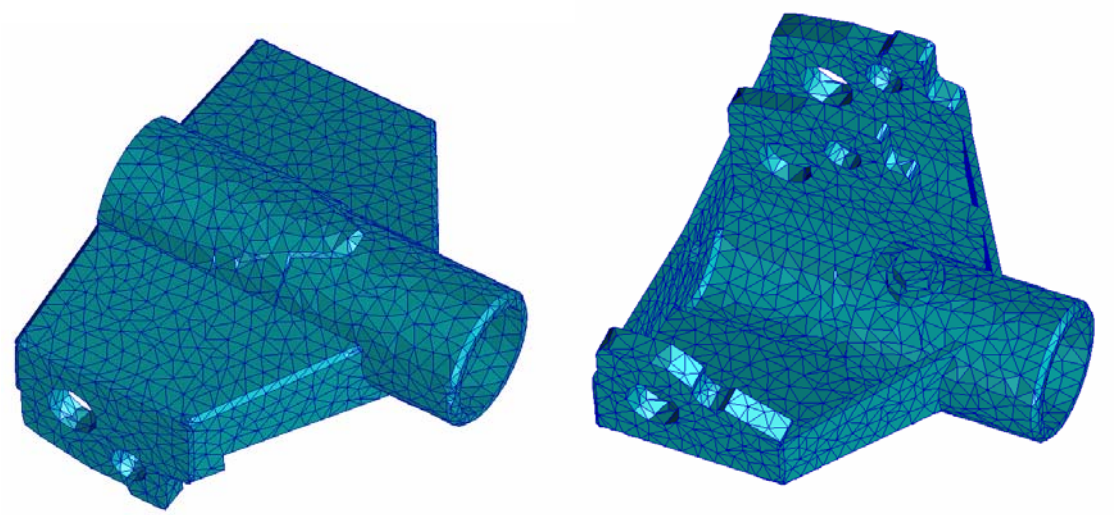


Figure 8.6 The mesh of the work-piece after mesh optimisation for triangular quality with 19235 triangular elements.

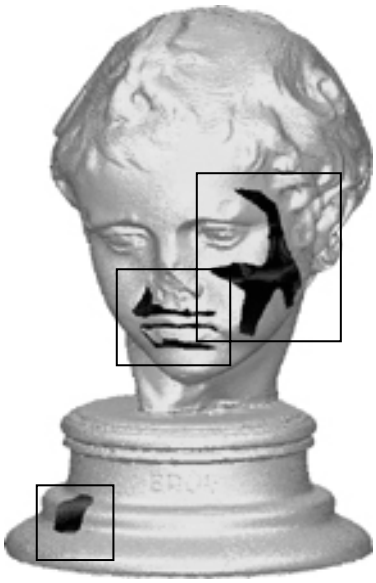


(a) **(b)**

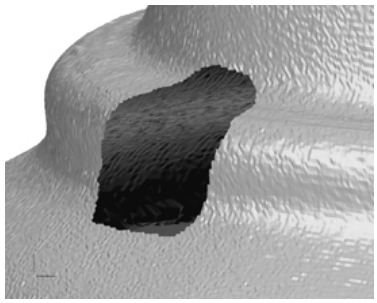
Figure 8.7 (a) View 1 of the mesh of work-piece optimized to improve the speed of model visualization, and (b) View 21 of the same mesh of work-piece optimized to improve the speed of model visualization with 4254 triangular elements.

8.2 Case-study 2

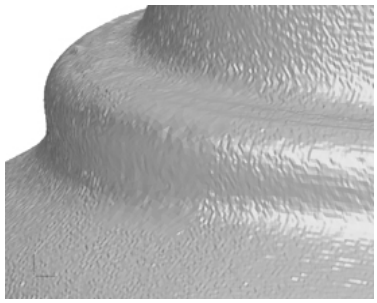
Figure 8.8 is designed to specially demonstrate the capability of the developed hole-filling algorithm. Figures 8.8(b) to (g) illustrate that the proposed algorithm has the ability to handle regions consist of positive and negative curvatures, and also highlights the importance of the implementation of Genetic Algorithms to construct a proper triangulation that covers the hole. One advantage of our proposed algorithm over Jun's method [36] is that we are consistent with the blending of the filling surfaces with the region of the hole, with varying curvatures, based on our initial triangulations using GA, which provided an initial underlying shape to a given hole. This way, we can craft the filling surfaces as close to the user's intended shape of the repaired model. In Figure 8.9, the develop mesh optimisation algorithm is applied to the repaired model of Figure 8.8, which contains 953125 triangular elements. With such large-scale mesh, it is undesirable to use it straightaway for visualization purposes. By preserving a high level of visual fidelity, the model is being reduced to 121765 triangular elements using the mesh optimisation tool developed in chapter 7, as shown in Figure 8.9. Figure 8.9(a) shows the mesh concentrations at different regions on the surface of the model, and Figure 8.9(b) demonstrates the visual smoothness of the model when the edges of the elements are hidden. With an 87% of reduction in mesh–size, the overall features and shape of the model is still very well preserved.



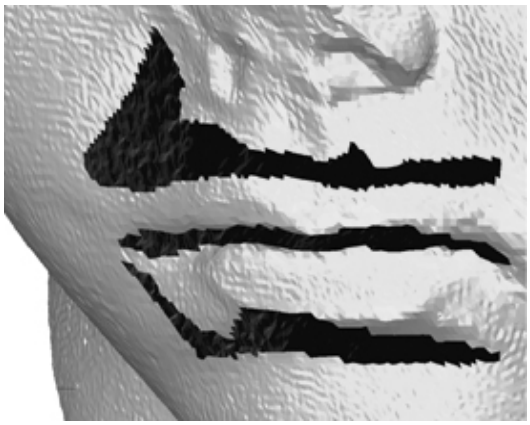
(a)



(b)



(c)



(d)

(e)

(f)

(g)

Figure 8.8 (a) A human head sculpture model with defects; (b),(d)and(f) Polygons with holes; and (c), (e) and (f) repaired model with proposed hole-filling algorithm.

(a) (b)

Figure 8.9 (a) The Sculpture model with mesh concentration, and (b) Visual model shown when the edges of the elements are hidden.

8.3 Results and Discussion

In this chapter, two case studies are presented to illustrate the advantages of the developed algorithms that are described in various chapters of this thesis. The main highlights are the fully automation of the repair and optimisation algorithms, the emphasis of mesh repair over defective geometrical models, the high quality of mesh-repair, in terms of gaps and hole repair and last but not least, the mesh optimisation using genetic algorithms that can optimize a model for model fidelity or model feasibility in terms of mesh-resolutions.

Chapter 9:

Conclusions and Future Work

9.1 Contributions

With reference to the research objectives set in chapter 3, contributions are highlighted accordingly as follows:

9.1.1 Contribution 1: Automate the mesh repair process

The first contribution is the development of a mesh repair solution that automatically rectifies common geometrical and topological errors that are inherent in the processes of CAD modelling and simulation. A problem detection and identification module is developed which helps users to automatically identify problems and errors in their models, instead of discovering these problems and errors themselves at a later stage. The mesh repair algorithms also replace traditional complex geometry repair processes with a novel but simplistic mesh repair technique to create water-tight models that suit the meshing needs for finite element analysis. These algorithms are generic and can be applied to many types and formats of CAD/CAE models.

The methodology of the proposed algorithm is summarized as follows:

(i) *Closing of gaps and overlaps*. The proposed algorithm will constrain the edges of the original model with mesh seeds and thereafter discretized the surfaces using triangular elements. Nodes on the free element edges will be merged if they are within a user-specified tolerance. The merging of nodes will be done by shifting the nodal positions based on a tangential interpolation. To compensate for the accuracy and the smoothness along the merging edges, a cubic Bézier approximation technique is proposed.

(ii) *Sewing of T-joint*. The sewing of T-joints is achieved by a nodal insertion and element splitting algorithm to realize a conforming mesh along the T-joint. The nodes on the free element edges will then be equivalence using a similar algorithm of closing of gaps and overlaps.

(iii) *Enhancing mesh of sliver surfaces*. The mesh of a sliver surface usually consists of elements of very poor aspect ratio. This is undesirable as far as numerical solution is concerned. To rectify this problem, the enhancement improves the quality of the meshes of sliver surfaces by an element reconstruction algorithm.

9.1.2 Contribution 2: Emphasis on performing repair in meshes

The novelty of this proposed method is that the mesh-repair process is to include model repair and mesh generation into a black-box. This automatic mesh-repair algorithm essentially simplifies the problems of the imperfect

models and allows one to deal with simple polygons rather than complex surface representations.

9.1.3 Contribution 3: Shape prediction in hole filling

In this work here, the main contribution is to give a novel and complete account of an effective geometrical method for the automatic patching of complex holes and ensures water-tightness, due to missing surface patches in both 3D surface models and faceted models. The main stages of this method are: hole identification, hole simplification using rough set, hole triangulation using Genetic algorithm, surface fitting based on a quartic Bézier patch and element-based hole meshing with nodal projection based on customized advancing–front technique. This algorithm makes use of the Genetic Algorithm to obtain an optimal initial triangulation over the hole which is subsequently used for surface interpolation to approximate the underlying shape of the hole. A quartic Bézier surface interpolation is then performed over the optimal initial triangulation to approximate the shape over the hole. Next, an unstructured triangular mesh is generated over the hole using a customized Advancing Front meshing algorithm which based its geometric references on the surface interpolation. This allows the mesh to model the missing shape using geometric information in the vicinity of the hole. The customized Advancing Front meshing algorithm also ensures that elements of good quality are achieved and that the resolution of the mesh at the patched region matches the mesh density at the locality of the hole. Results show that this technique is

able to handle holes with complex boundary and is able to produce a mesh over the hole which corresponds closely to the original geometry.

9.1.4 Contribution 4: Mesh optimisation using Genetic Algorithms

The Fourth contribution is the investigation of the use of a genetic algorithm (GA) to perform the large-scale triangular mesh optimization process. This optimization process consists of a combination of mesh reduction and mesh smoothing processes that will not only improve the speed for the computation of a 3D graphical or finite element model; it will also improve the quality of its mesh. The genetic algorithm (GA) is developed and implemented to replace the original mesh with a re-triangulation process. While retaining features is important to both visualization models and finite element models, this algorithm also optimizes the shape of the triangular elements, improve the smoothness of the mesh and perform mesh reduction based on the needs of the user.

Although this work using genetic algorithm is still yet to be conclusive, it does provide some promising results. Though Genetic Algorithm may not be the most efficient algorithm, due to the time taken to run the optimization, its flexibility makes it a potentially useful system for handling model containing huge data or large number of elements, for example, in the visualization of 3D models in graphical systems with data-size constraint, such as those immersive virtual environment systems. The domain-independent genetic algorithms make them perfect solutions in situations where the factors affecting the mesh fidelity are not fully understood or not easily enumerated. With a little work and some more

experimentation, this work may become a practical alternative to existing level of detail system.

9.1.5 Contribution 5: Discovery and implementation of potential applications arising from the mesh-repair algorithms

One difficult task in performing research studies is to bridge research with applications. Coincidently, it is discovered that, it is possible to make use of the techniques and algorithms developed in mesh repair to design applications that can be used in areas beside the finite element model repair. This contribution involves the discovery of two possible avenues to apply the developed techniques, and they are as follows:

1. Model Feature Suppression based on Hole Repair Algorithm
2. Restoration and reverse engineering of bio-models, artifacts, and the designing of implants in Cranioplasty

Although the intended hole-filling technique is surface repair and fitting, the patches may be used in a CAD environment. Since the patches are defined analytically, the representation is compact and easy to work with and the resulting composite surface can be edited locally in an interactive graphics environment to aid in the design of custom implants.

9.2 Conclusions

Typical CAD model problems involve structure (is the model definition correct?), realism (can it be manufactured?) and accuracy (is it accurate enough?). The problem of bad CAD data should not be ignored. Defective part files inhibit

analysis and manufacturing processes. They impede reuse of data for product improvements, sap productivity. All of these problems prevent CAD system owners from getting the return on investment they expect from CAD use. Only by understanding the nature of CAD model defects can CAD system owners ask intelligently for needed improvements. This will help engineers and software writers to understand in terms requirements, validity and quality in the area of mesh manipulation.

The novelty of the work presented in this thesis is the development of error identification and mesh-repair algorithms that automatically identify and rectify common geometrical and topological errors that are inherent in the processes of CAD modelling and simulation. A new technique, based on genetic algorithm (GA) and surface interpolation, is also introduced to specifically fill complex holes in unstructured triangular meshes. New areas of applications, extending the usage of the developed mesh-repair algorithms, are addressed. They are model feature suppression and reverse-engineering of bio-models, artifacts, and the designing of implants in cranioplasty. Finally, GA is employed to perform the large-scale triangular mesh optimization, where the shapes of the triangular elements are optimized, improves the smoothness of the mesh and performs mesh reduction based on the needs of the user.

9.3 Recommended future work

The approach presented in gaps repairs has some drawbacks, which should be investigated in the future. The setting of tolerance values for merging nodes is currently quite heuristic. A more systematic setting would be

advantageous, e.g. by tagging the tolerance values to the mesh size and the lengths of edge curves. Re-meshing will be carried out ultimately to create the optimal mesh (triangular or quad or mixed surface element) for the surface model. Subsequent self-repair techniques will focus on the repair of invalid models that are defined by invalid surfaces, such as open loops, missing surfaces, complex holes etc., using the mesh-repair method.

The limitation of using quartic Bézier patch is that the shape of the patched hole tends to “shrink” more inwards if the size of the hole gets larger which can be spotted in case-studies. One of the future works in this area is to look into controlling shapes of the filled mesh patches without affecting tangent plane continuity. Possible approach will also involve building a skull shape library to get an idea of the normal variation and to help predict the effectiveness of the reconstruction of the skull.

The main drawback of using GA is the speed of the mesh optimization process. The main weakness of this algorithm is the slow computational speed even with high performance workstations. Research on parallelization will be studied in the hope to shorten the loading as well as the computational time.

References

Current-State-of-the-Art on Gaps and Overlaps Repair

1. “Initial Graphics Exchange Specification (IGES)”, version 5.1. Nat’l Computer Graphic Assoc., 1991.

2."Keith Koster", “PDES and STEP”, http://lrc.csun.edu/~icostea/CIM/keith/pdes.htm.

3. “Stereolithography Interface Specification”, p/n 50065-S01-00, 3D Systems Inc., Valencia, Calif., 1989.

4. J. P. Steinbrenner, N. J. Wyman and J. R. Chawner, “Procedural CAD Model Edge Tolerance Negotiation for Surface Meshing”, *Engineering with Computers*, Vol. 17 p. 315-325, 2001.

5. G. Barequet and S.Kumar, “Repairing CAD Models”, *Proc. IEEE Visualization*, Phoenix, AZ, p.363-370, 1997.

6. G. Barequet and M.Sharir, “Filling gaps in the boundary of a polyhedron”, *Computer Aided Geom. Design*, 12, p.207-229, 1995.

7. G. Barequet, “Using geometric hashing to repair CAD objects”, *IEEE Comput. Science & Engineering*, p. 22-28, 1997.

8. T. M. Murali and T. A. Funkhouser, “Consistent Solid and Boundary Representations from Arbitrary Polygonal Data”, *Proc. 1997 Symposium on Interactive 3D Graphics*, Providence, Rhode Island (A. Van Dam, Ed.), p. 155-162, Assoc. Comput. Mach. Press, New York, 1997.

9. N. Anders Peterson and Kyle K. Chand, “Detecting Translation Errors in CAD Surfaces and preparing geometries for Mesh Generation”, Center for Applied Scientific Computing, Lawrence Livermore National Labs, Livermore, CA 94551, Aug 2001, UCRL-JC-144019.

10. S. M. Morvan and G. M. Fadel, “IVES: An Interactive Virtual Environment for the Correction of .STL files”, *Proc. Conf. Virtual Design,* Univ. of California, Irvine, Aug. 1996.

11. S. M. Morvan and G. M. Fadel, “An Interactive Correction of .STL files in a Virtual Environment”, *Proc. Solid Freeform Fabrication Symposium*, U. Texas, Austin, Texas USA, Aug. 1996.

12. G. Turk and M. Levoy. “Zippered Polygon Meshes from Range Images”, *Proc. of ACM SIGGRAPH*, p. 311-318, 1994.

13. H. T. Yau, C.C. Kuo and C.H. Yeh, “Extension of Surface Reconstruction Algorithm to the Global Stitching and Repairing of STL Models”, *Computer Aided Design*, 35, p. 477-486, 2003.

14. J. T. Hu, Y. K. Lee, T. Blacker and J. Zhu, “Overlay Grid Based Geometry Cleanup”, *Proc. 11th International Meshing Roundtable*, http://www.imr.sandia.gov/11imr/main.html, Ithaca, New York, USA, Sep. 2002**.**

15. I. Makela and A. Dolenc, “Some Efficient Procedures for Correcting Triangulated Models”, *Proc. Solid Free Form Fabrication Symp*. H.L. Marcus et al., eds., pp. 126-134, Univ. of Texas, Austin, Aug. 1993.

16. X. Sheng and I.R. Meier, “Generating Topological Structures for Surface Models”, *IEEE Computer Graphics and Applications*, vol.15, no. 6, pp. 35-41, Nov. 1995.

17. L.P. Chew, “Guaranteed-quality Delaunay meshing in 3D”, *Short paper in Proc. 13th Ann. Sympos. Comput. Geom*, pp. 391-393, 1997.

18. S.W. Cheng, T.K. Dey, H. Edelsbrunner, M.A.Facello and S.H. Teng, “Sliver Exudation”, *J. Assoc. Comput. Mach*., New York, vol. 47, no.5, pp.883-905, Sep. 2000.

19. H. Edelsbrunner, X.Y. Li, G. Miller, A. Sathopoulos, D. Talmor, S.H. Teng, A. Ungor and N. Walkington, “Smoothing and cleaning up slivers”, Proc. 32nd Annu. ACM Sympos. Theory Comput., pp. 273-277, 2000.

20. L. Fine, L. Remondini and J-C. Leon, “Automated generation of FEA models through idealization operators”, *International Journal for Numerical Methods in Engineering*, John Wiley, Vol 49, Num 1, pp.83-108, Sep. 10-20, 2000.

21. P. Vron, J-C. Leon, “Static polyhedron simplification using error measurements”, *Computer-Aided Design*, Vol.. 29, No 4, pp. 287-298, 1997.

22. S. Maza, F. Noel and J.C. Leon, "Generation of quadrilateral meshes on free-form surfaces", *Computers and Structures*, Pergammon, Vol 71, pp.505-524, 1999.

23. F. Noel, "Adaptation of CAD surface meshes to a map of sizes through the IGATOMM concept", *International Journal for Numerical Methods in Engineering*, John Wiley, Vol 49, Num 1, pp.313-327, September 10-20 2000

24. F. Noel, J.C. Leon and P. Trompette, "A New Approach to Free-form Surface Mesh Control in a CAD Environment", *International Journal for Numerical Methods in Engineering*, Wiley, Vol 38, Num 18, pp.3121-3142, September 1995

25. Noel, J.C. Leon and P. Trompette, "A Data Structure Dedicated to an Integrated Free-form Surface Environment", *Computers and Structures*, Pergammon, Vol 57, Num 2, pp.345-355, 1995

Current-State-of-the-Art on Hole Filling Techniques

26. G. Barequet, M.Dickerson, D. Eppstein. “On triangulating three-dimensional polygons”, *In: Proc. Twelfth Annual Symposium on Computational Geometry*, 38-47, May 24-26, 1996, Philadelphia, Pennsylvania, United States.

27. B. Chazelle, Triangulating a simple polygon in linear time, *Discrete Computational Geometry,* 1991; 6:485-524.

28. Bern M, Eppstein D. Mesh generation and optimal triangulation. In: Hwang FK, Du DZ, editors. *Computing in Euclidean Geometry, Lecture Notes Series on Computing, Volume 1, World Scientific*; 1992, p. 23-90.

29. H. Edelsbrunner, L.Guibas, “Topologically sweeping an arrangement’, *J Comput Syst Sci,*1989; 38(1):165-194.

30. B.Curless, M.Levoy, “A Volumetric Method for Building Complex Models from Range Images”, *In: Proc. Computer Graphics SIGGRAPH '96*, ACM; 1996, p. 221-227.

31. Davis J, Marschner SR, Garr M, Levoy M. Filling holes in complex surfaces using volumetric diffusion. *In: Proc. 3D Data Processing Visualization and Transmission*, First International Symposium; 19-21 June 2002, p. 428 - 861.

32. Ju T. Robust repair of polygonal models. *ACM Trans Graph 2004*; 23(3):888-895.

33. Carr JC, Fright WR, Beatson RK. Surface Interpolation with Radial Basis Functions for Medical Imaging. *IEEE Trans Med Imaging 1997*; 16(1):96-107.

34. Liepa P. Filling Holes in Meshes. *In: Proc. Eurographics/ACM SIGGRAPH Symposium on Geometry Processing, Eurographics Association*; 2003, p. 200-205.

35. Chui C, Lai MJ. Filling polygonal holes using C1 cubic triangulaer spline patches. Comput Aided Geom Des 2000, 17: 297-307.

36. Yongtae Jun. A piecewise hole filling algorithm in reverse engineering. Comput-Aided Des 2005; 37:263-270.

37. Walton DJ, Meek DS. A triangular G^1 patch from boundary curves. Comput-Aided Des 1996; 28(2):113-123.

38. Desbrun M, Meyer M, Schröder P, Barr AH. Implicit fairing of irregular meshes using diffusion and curvature flow. In: Proc Computer Graphics SIGGRAPH '99; 1999: p. 317-24.

39. J. Komorowski, Z. Pawlak, L. Polkowski, and A. Skowron. "Rough sets: a tutorial". In S.K. Pal and A. Skowron, editors, Rough-Fuzzy Hybridization: A New Method for Decision Making, Springer-Verlag, Singapore, 1998. http://citeseer.ist.psu.edu/komorowski98rough.html

40. Jin-Mao Wei, Rough Set Based Approach to Selection of Node, International Journal of Computational Cognition Volume 1, Number 2, June 2003

41. Michalewicz Z. Genetic Algorithms + Data Structures = Evolution Programs. 3rd ed. New York: Springer-Verlag; 1998.

42. Su Y, Senthil Kumar A. Templatized refinement of triangle meshes using surface interpolation. Int J Numer Methods Eng, article in press.

43. Piper B. Visually smooth interpolation with triangular Bezier patches. In: Farin G, editor. Geometric Modelling: Algorithms and New Trends, SIAM, Philadelphia; 1987, p. 221-233.

44. Walton DJ, Yeung M. Geometric modelling from CT scans for stereolithography apparatus. In: Tang Z, editor. New Advances in CAD & Computer Graphics (Proc. CAD/Graphics '93), International Academic Publishers, Beijing, China; 1993, p. 417-422.

45. George, Louis P and Seveno E, The Advancing-Front Mesh Generation Method Revisited. Int J Numer Methods Eng, 1994, 37, p.3605-3619.

46. Owen SJ, White DR, Tautges TJ. Facet-based surfaces for 3D mesh generation. In: Proc. 11th International Meshing Roundtable, Sandia National Laboratories; 2002, p. 297-312.

47. A. Mota, W. S. Klug, M. Ortiz, A. Pandolfi, Finite-element simulation of firearm injury to the human cranium, Computational Mechanics 31:115–121, Springer-Verlag , 2003, DOI 10.1007/s00466-002-0398-8.

Current-State-of-the-Art on Optimization Algorithms in Meshing using Genetic Algorithms

48. Caponetto R, Fortuna L, Graziani S, Xibilia MG. Genetic Algorithms and applications in system engineering: a survey. *Transactions of the Institute of Measurement and Control* 1993;15(3):143-156.

49. Rudolph G. Convergency analysis of canonical genetic algorithms. *IEEE Transactions on Neural Networks* 1994;5(1):96-101.

50. Michalewicz Z. Genetic algorithms + data structures = evolution programs. 3. *Berlin: Springer*, 1996.

51. Goldberg DE. Genetic algorithms in search, optimization, and machine learning. *Reading, MA: Addison-Wesley*, 1989.

52. Holland JH. Adaptation in natural and artificial system. *Ann Arbor, MI: The University of Michigan Press*, 1975

53. Dorsey RE, Mayer WJ. Genetic algorithms for estimation problems with multiple optima, non-differentiability, and other irregular features. *Journal of Business and Economic Statistics* 1995;13(1):53-66.

54. Liu X, Begg DW, Fishwick RJ. Genetic approach to optimal topology/controller design of adaptive structures. *International Journal for Numerical Methods in Engineering* 1998;41(5):815-830.

55. Absaloms H, Tomikawa, T. Surface reconstruction by triangulation using GA. *Proceedings of the 20th International Conference on Computers and Industrial Engineering*, 6-9 October, Kyongju, Korea, 1996, pp. 441-444.

56. Qin KH, Wang WP, Gong ML. A genetic algorithm for the minimum weight triangulation. *Proceedings of the IEEE Conference on Evolutionary Computation*, 13-16 April, Indianapolis, IN, USA, 1997, pp. 541-546

57. Hamann B. A data reduction scheme for triangulated surfaces. *Computer Aided Geometric Design,*1994;11(2):197-214.

58. Hoppe H, DeRose T, Duchamp T, McDonald J, Stuetzle W. Mesh optimization. *ACM SIGGRAPHICS Proceedings* 1993; 1:19-26.

59. Ronfard R, Rossignace J. Full-range approximations of triangulated polyhedra. *Proceedings of EUROGRAPHIVS'96, Computer Graphics Forum* 1996;15(3):67-76.

60. Gieng TS, Joy KI, Schussman GL, Trotts IJ. Constructing hierarchies for triangle meshes. *IEEE transactions on Visualization and Computer Graphics* 1998;4(2):145-161.

Publications arising from this thesis:

Journal Publications

A1. C.S. Chong, A. Senthil Kumar, H. P. Lee, "Automatic Mesh-Repair Technique for Model Repair and Finite Element Model Generation", *Journal of Finite Elements in Analysis and Design* – accepted for publication.

A2. C.S. Chong, Y. Su, H. P. Lee and A. Senthil Kumar, "Automatic Hole-Filling of Triangulated Models with Shape Approximation", *Journal of Computer-Aided Design* – article under review.

A3. C. S. Chong, H. P. Lee, A. Senthil Kumar, "Automatic Hole Repairing for Cranioplasty using Bézier Surface Approximation", *International Journal of Craniofacial Surgery*. 16(6):1076-1084, November 2005.

A4. C. S. Chong, H. P. Lee, A. Senthil Kumar, "Genetic Algorithms in Mesh Optimization for Visualization and Finite Element Models", Journal of Neural Computing and Applications (2006) 15: 366-372, DOI 10.1007/s00521-006-0041-2.

Conference Publications

A5. C.S. Chong, A. Senthil Kumar, H. P. Lee, "High Geometric Fidelity Hole Repair for Meshed Models", *CAD Conference 2005*, 20-24 June 2005, Bangkok, Thailand.

A6. C.S. Chong, A. Senthil Kumar, H. P. Lee, "Region Filling and Hole Repair for Bio-Medical Models", *Biomedical Engineering Conference 2005*, Austria, 16-18 Feb 2005, pp 26-31.