www.cadanda.com Computer-Aided Design AND APPLICATIONS

Automatic Detection of Geometric Features in CAD models by Characteristics

Peter Chow[1], Tetsuyuki Kubota[2] and Serban Georgescu[3]

[1]Fujitsu Laboratories of Europe Ltd, Peter.Chow@uk.fujitsu.com
[2]Fujitsu Laboratories of Europe Ltd, Tetsuyuki.Kubota@uk.fujitsu.com
[3]Fujitsu Laboratories of Europe Ltd, Serban.Georgescu@uk.fujitsu.com

ABSTRACT

In this paper, we detailed a novel 3D detection method for geometric features in CAD models used in CAD-to-CAE model simplification process. Model preparation is now the most time consuming part in the computer-aided engineering process chain, it is commonly labor intensive. Therefore, the overall objectives are: 1) introduce automation to speed-up the model preparation process; 2) remove redundant features to reduce model mesh size for fast mesh generation and analysis without compromising on solution accuracy. Automatic feature detection that is generic and easily extendable is a key element needed to achieve these objectives. Inside the article, we proposed detection by characteristics to complement existing geometric modeling kernels for CAD systems with defeaturing functions to detect features previously not detectable.

Keywords: feature detection, model simplification, CAD-to-CAE.

1. INTRODUCTION

Since the introduction of computer for product design and development, computer-aided design (CAD) and computer-aided engineering (CAE), the time for design and simulation process is decreasing as computing performance increases. This is good news in time-to-market of product and saves on physical prototypes.

Today, products are design and simulation validated in virtual space before fabrication in physical space. In the case of electronic products, the common simulations are: structural analysis, electromagnetic and thermal-fluid cooling. A major bottleneck to scalability (continue to decrease in time) in this process, as computer performance continues to grow exponentially, is the CAD-to-CAE model preparation (CCMP) stage. The end result from the design stage is a CAD model that needs to be processed to create a model suitable for CAE stage. Using CAD model directly is possible but comes at high computational costs and long solution time. The common practice is to add a simplification process at the CCMP stage; this reduces the CAD model features relevant for the class of simulation without comprising solution accuracy. This results in big efficiency improvements in all downstream applications such as meshing and solving the systems of equations required for simulation.

Figure 1 highlights the simplification process required of a server model for thermal-fluid cooling analysis, from the original CAD model to a simplified CAD model for CAE. The time given for each phase is time required for manual simplification. Automation needs to be introduced in this process to scale with computer performance increase. Key elements are detection and processing of features in the CAD model. The latter can be addressed with advanced geometric modeling kernels for CAD systems such as ACIS [1] and Parasolid [13]. For electronic applications, the detection rate is not sufficient for the purpose and requires a more advanced and flexible method to complement already existing methods and technologies. This is detailed and discussed in the following sections.

2. BACKGROUND

In this section, we first take a summary look at evolution of CAD from first appearance in beginning of 1960 s to fifth generation 3D CAD systems with direct modeling (DM) function of today. Majority of these systems depends on geometric modeling kernels that can be traced back to CAD's early beginning. Then, we look at the previous works related to feature recognition. Finally, the proposed feature detection method

Taylor & Francis
Taylor & Francis Group

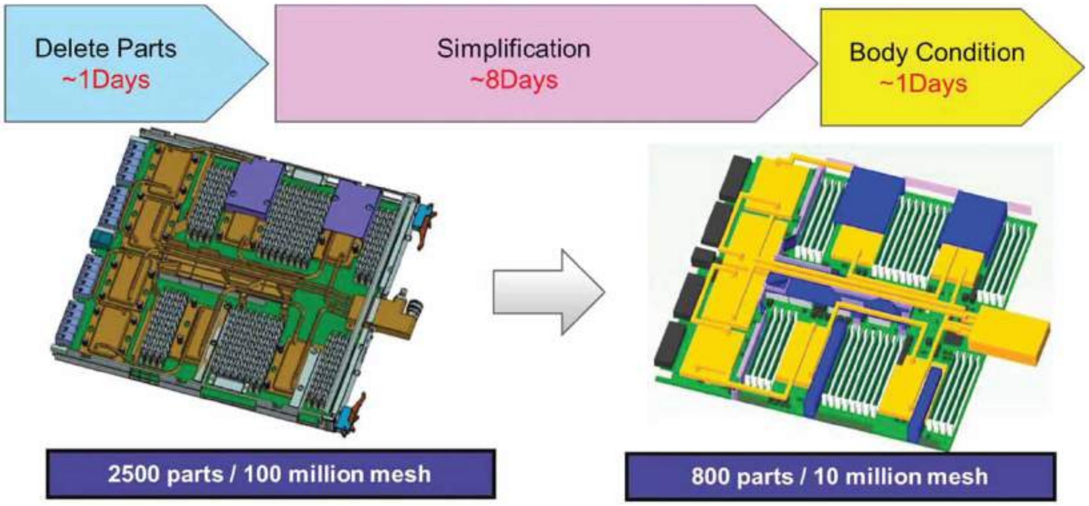

Fig. 1: CAD-to-CAE simplification for a server model.

in the feature recognition domain to CAD-to-CAE defeaturing and simplification.

2.1. CAD Evolution

Ivan Sutherland is frequently considered the father of CAD system [3–5,18]. He developed Sketchpad [16] in 1963 as part of his PhD thesis at MIT. Previously in 1957 Patrick Hanratty, widely known as father of CADD/CAM, developed PRONTO the first commercial numerical-control programming system. In the mid of 1960 s manufacturers developed in-house 2D CAD systems for drafting applications. Commercial use of 2D CAD started in 1970 s for modeling and drafting. In 1977 saw the starts of 3D CAD development. The IGES 3D data exchange format was defined in 1979. The 1980 s saw fast advancements in innovation, the release of boundary representation (B-Rep), parametric and associative solid modeler. And in 1982 geometric modeling kernel Romulus B-Rep solid modeler was released, designed for straightforward integration into CAD software. Its successors ACIS kernel was released in 1989 and shortly follow by Parasolid. Not too long after Microsoft released its first 32-bit operating system in 1994, ACIS and Parasolid were available for Windows NT platform. From there, Windows PC surpasses UNIX workstation platform and made CAD software available to a wider market and audience. Today, CAD systems are in the fifth generation. Tab. 1 summarized the key feature of 1st to 5th generation of CAD systems.

Direct modeling (DM) or history-based free CAD systems are popular with non-traditional CAD users. While parametric history-based approach is powerful but expert knowledge and proficiency are required to use the system. It is not a suitable tool for the uninitiated users such as engineers and designers. The explicit modeling methodology in DM gives designers and engineers flexibility and easy to use, learn

CAD generation	*Key feature*
1st	2D drafting
2nd	3D wireframe
3rd	3D boundary representation (B-Rep)
4th	3D constructive solid geometry (CSG)
5th	3D parametric feature based or history-based system

Tab. 1: Generation of CAD systems.

in a very short time over feature-based CAD systems. It is especially effective in concept design stages and in extended team environments. More information between direct and history-based CAD modeling is available in following references [6,8]. In 2007 SpaceClaim released a history-free direct modeling 3D CAD system. This innovation prompted feature-based CAD developers to integrating DM function in their products, leading to the age of hybrid CAD systems.

The evolution of CAD will continue, with new innovations to opening new markets and users. Today, majority of CAD systems, old and new, are supported by geometric modeling kernels. They are rich with functions and features, encapsulating many years of geometric know-how, advanced and efficient technologies for integration and innovations. This foundation kernel component will continue to support future generations of CAD and associated systems.

2.2. Feature Detection Related Work

CAD feature detection is generally associated with CAD/CAM feature recognition [20,2,11]. There have been over 30 years of research and development efforts from seminal work in 1980 [9]. It spans an industry of machining materials to the desired shape. The automated process of CAD/CAM to machining is a major advancement in manufacturing. Feature recognition plays a key role in such a process, Fig. 2, a prerequisite where a part created through both feature-based modeling and solid modeling. Even in feature model conversion, in situation where is not possible to achieve it comes into play [7].

Defeaturing [11] and simplification [12,17,19,20,10] of CAD models for CAE is a recent development compares to CAD/CAM, driven by the need to reduce the size of mesh model and computing time for analysis. The CAD/CAE model preparation stage is the most time consuming of the entire simulation process, over 80 percent in example given in Fig. 1 when total turnaround time is 12 days (2 days for analysis including meshing). Generally, very small features have no significant influence on solution accuracy but they have a big impact on size of mesh generated. CAE analysts often received the same CAD model for design and manufacturing, they contains much more geometric detail than is necessary. Therefore, removing the unnecessary detail from the 3D CAD model prior to meshing is a prerequisite step typically done manually at significant cost to total turnaround time.

The automated process of defeaturing consists of detection follow by removal. Feature detection in CAD/CAE is similar to CAD/CAM feature recognition process. They shared the basic components but different in requirement and output. Fig. 2 illustrates the feature recognition/detection between the two applications. In CAD/CAM, features are of pockets, holes, slots, bosses, etc. for defining the machining steps to fabrication. For both it is straightforward in the case of parametric feature-based CAD systems, CAD/CAE features such as blends and chamfers. In this occasion, the history-based approach is a key advantage. For other defeaturing elements such as holes and bumps they required extra specification by the creator of the CAD model. When all the parts in an assembly CAD model are created in this manner and integrated with CAE then defeaturing is simple and efficient [10].

In an industrial environment this is commonly not possible. Designers today interact directly with CAD creation uses DM, exchange concepts with extended teams using CAD models and needing to make changes on the fly. Typically, they are not interested in downstream issues such as defeaturing. Part models in an assembly CAD model can come from diverse sources, often conversion to standard CAD format such as STEP where parametric feature-based information and data are lost. Even when models are received in native format, interoperability between CAD models and feature-based data is frequently not possible.

In situation where mesh generation for finite-difference analysis, shape simplification can greatly reduce size of meshes such as modifying a cylinder shape to a rectangular block. This is straightforward with history-based CAD systems but not with history-free systems. Feature detection requirement here is somewhat difference from defeaturing, basic functions are similar but small is no longer a distinctive condition.

2.3. CAD/CAE Defeaturing and Simplification

In this section, we provide a short summary on what is possible and what is not in defeaturing. Defeaturing is the process that combines “feature detection” and “feature processing” for CAD model simplification. Compares to CAD and CAD/CAM, defeaturing is a relatively new technique coming from the need to reduce mesh model size by removing unnecessary features so that to shorten analysis time. Feature processing is native to CAD, thus is more advanced than feature detection. Presently, we are not near finding a

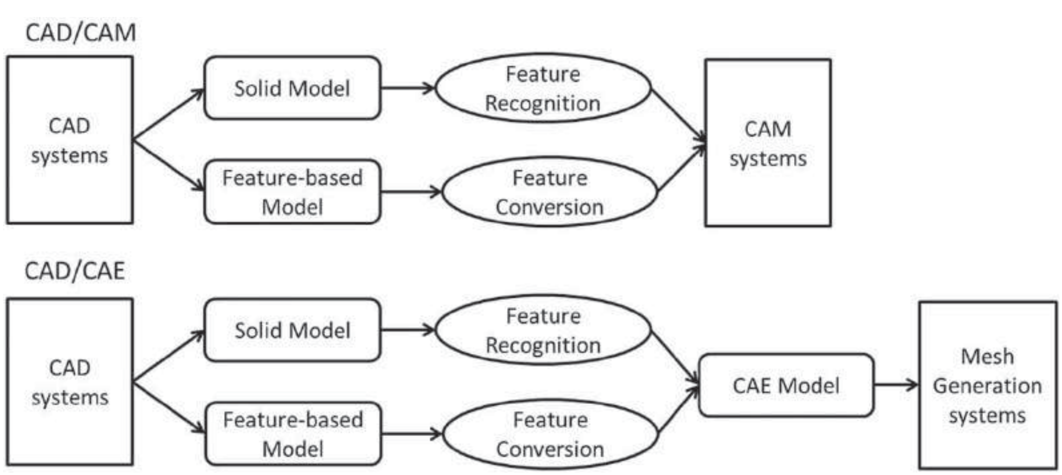

Fig. 2: Feature recognition in CAD/CAM and CAD/CAE applications.

universal solution for detecting features of various classes. Take for example, the class for bumps as depicted in Fig. 3; the bumps in red are frequently not detectable. One thing we know for sure, we can expect more bump shapes that we have not yet encountered in our application field.

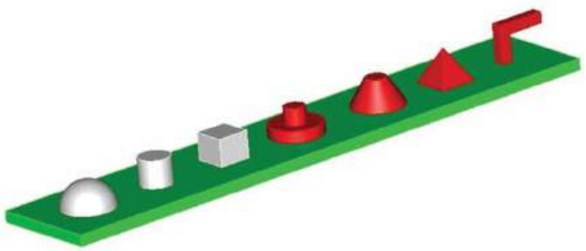

Fig. 3: Class for bumps with varying shapes.

Figure 4 highlights some common cases possible with defeaturing functions, detecting and removing small circle hole and blends. Whereas in Fig. 5 we shows some cases not generally possible. Modifying shape for finite-difference meshing and analysis may perhaps seem irrelevant to finite-element meshing and analysis. Fig. 6 highlights the case for finite-element, where small redundant feature increase mesh density leading to increase time in meshing and subsequent analysis. Here, almost doubling the mesh size and about five folds difference in meshing time.

In feature detection, there are methods proposed and studies conducted on simplification for complex 3D CAD models [12,17,19,20,10]. Basic features of circle holes, blends and chamfers can be detected automatically and are available in geometric modeling kernels such as ACIS. For general features, for example arbitrary holes and bumps, it is thought to be difficult for these methods to detect [14,15,17].

3. METHOD

In this section, we detail the new detection method. It is based on the characteristics of the geometric feature class, thus we call this the "Detection by Characteristics" method. Before we describe the method, let's first take a look at the basic defeaturing process of our application as depicted in Fig. 7, the key processes and technologies involved in the work flows. Feature detection is the first process immediately after the CAD model has been read and prepared for processing with a geometric modeling kernel.

In this work, we start from a CAD model that is a solid model (no parametric feature-based records), for example a CAD model in STEP format read into a geometric modeling kernel for processing. The entity elements in this case are vertex point, edge, surface and body as highlighted in Fig. 8. Edges are derived from vertex points. Surfaces have topology based on edges. In ACIS geometric modeling kernel, edges are classified into lines, arcs, curves, etc. Surfaces are

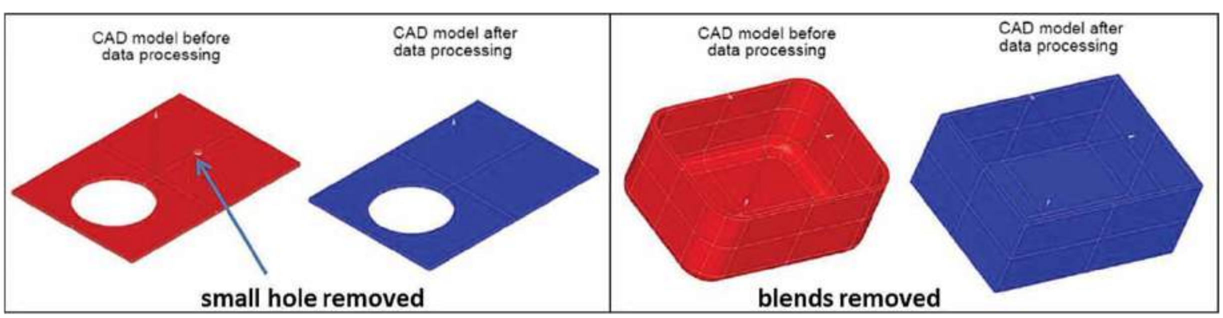

Fig. 4: Possible with existing defeaturing functions.

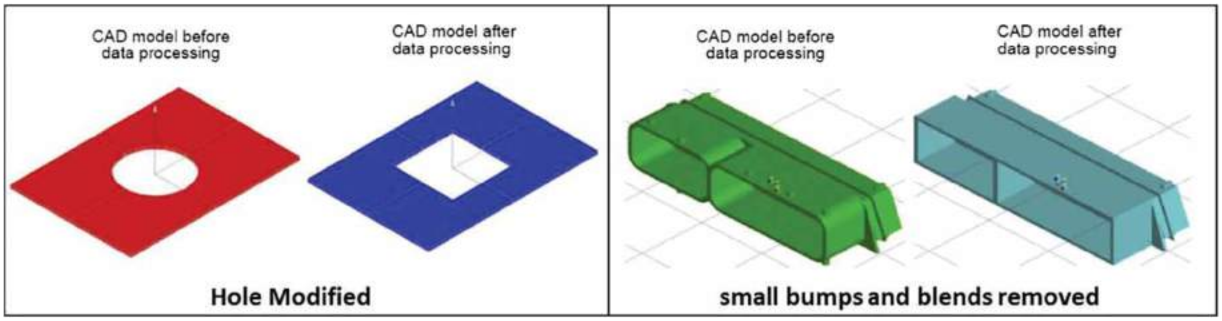

Fig. 5: Generally not possible with defeaturing functions.

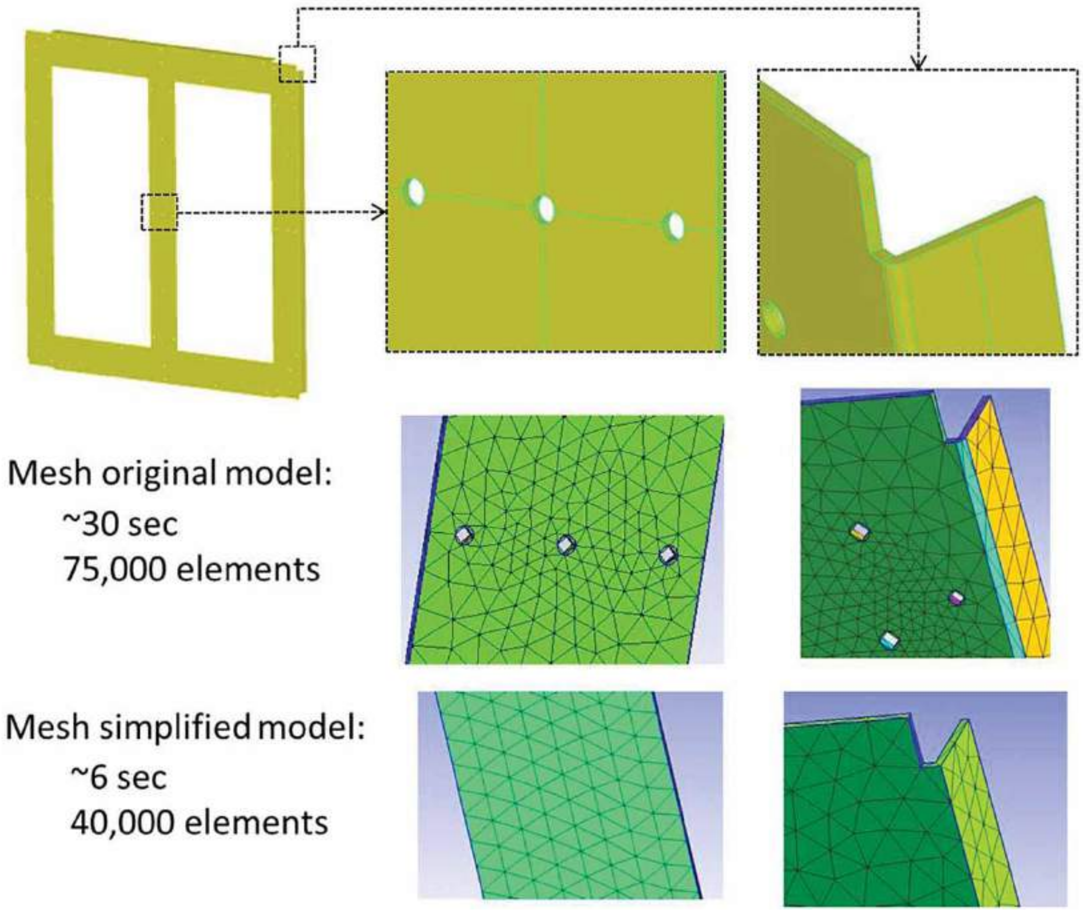

Fig. 6: Benefits of simplified model for finite-element situation.

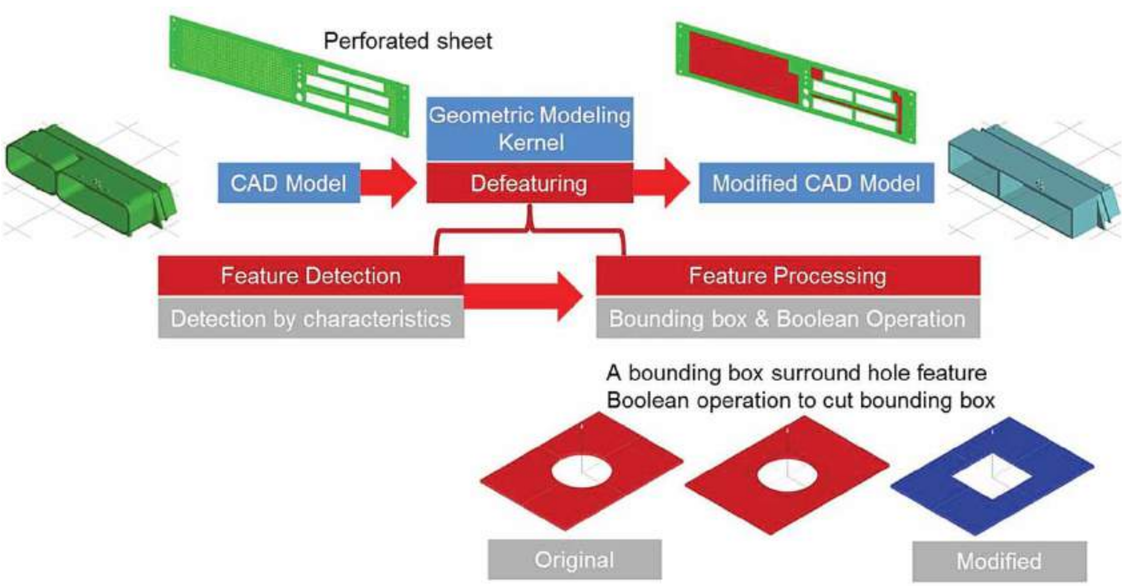

Fig. 7: Overview of defeaturing process and technologies.

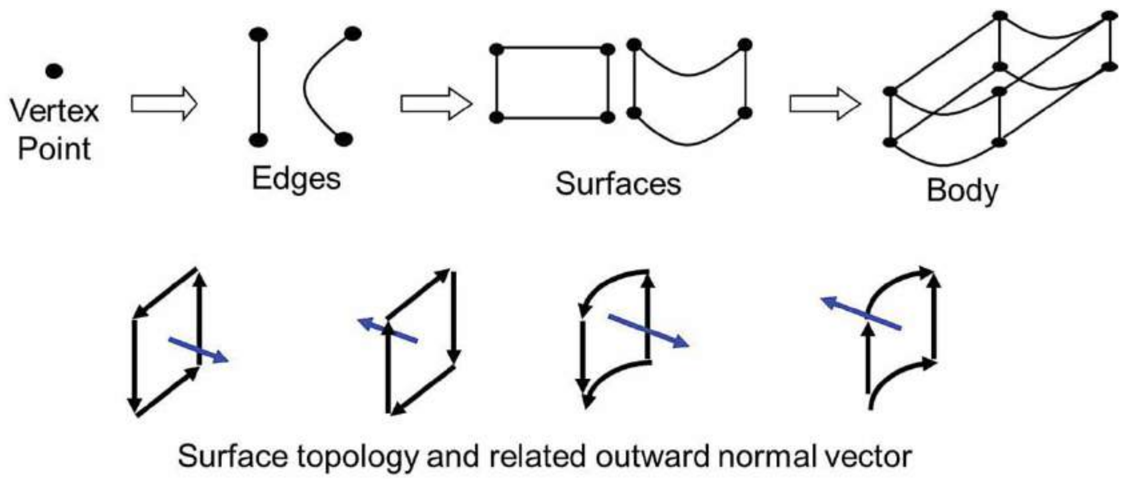

Fig. 8: CAD model entity elements and topology.

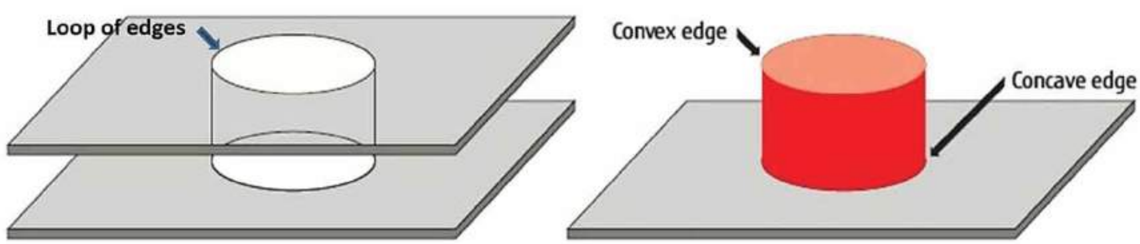

Fig. 9: Convex and concave edge characteristics.

classified into planar, cylindrical, sphere, helix, and so on. B-rep is used in ACIS.

Figure 9 highlights some example of list-data information and edge characteristics that can be derived from these entities and surface topology. These basic and other advanced functions and information are readily available in geometric modeling kernels. The loop of edges is a list-data of edges on the face of a surface where other features are located, in this example the hole and bump. Edges can have either a concave or convex characteristic as highlighted. When the loop of edges all has concave characteristic then it is probably a bump of some kind. Similarly, when the edge characteristic is convex then it is probably an arbitrary hole.

3.1. Arbitrary Hole

Based on these entities and derived functions, we can start describing characteristics in a feature in a generic way. Using the example class for arbitrary holes as depicts in Fig. 10, the shapes in red are frequently not detectable with exiting methods. Using detection by characteristics, it is possible to detect all these class of holes and more with the following three characteristics:

1. A list loop of edges exist inside a surface boundary and all have the convex characteristic;
2. Surfaces connected to the list loop of edges is a closed-loop;
3. Top and bottom surfaces, each with outward normal vector pointing away from each other.

The more characteristics we provide the more precise of identifying the right feature we want to detect. Adding an extra characteristic that all surfaces in the list loop are cylindrical, we can detect circle holes. To detect rectangular holes, four planar surfaces in the

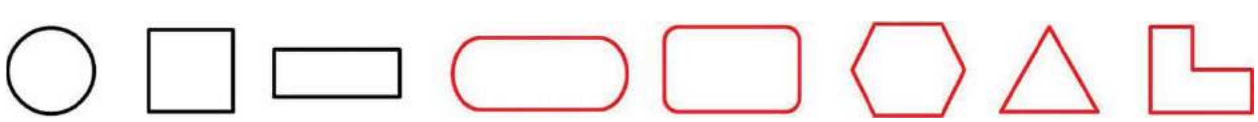

Fig. 10: Some arbitrary hole shapes.

list loop. Hexagon holes, six planar surfaces in the list loop. And so on for different kind of holes. The third characteristic is to make sure it is a through hole and not a cavity hole.

Small holes can be detected by adding the characteristic of size, for example, compare the volume of the arbitrary shape to the small feature volume parameter value. This value is commonly predefined or set by the required application level.

With such a flexible approach, there is a balance between performance and accuracy – more characteristics checking means accuracy at the expense of performance, and vice versa.

3.2. Arbitrary Bump

Similarly, detection by characteristic can describe an arbitrary bump with three characteristics:

1. A list loop of edges exist inside a surface boundary and all have the concave characteristic;
2. Surfaces connected to the list loop of edges is a closed-loop;
3. Bump surfaces forms a solid body.

A list loop of closed surfaces can be derived from edge-surface relationship with respective list loop of edges. The edges here all have the concave characteristic; the two surfaces on either side have outward normal vector pointing toward each other. Bump surfaces forms a solid body means closure, no holes; this can be derived from surface adjacency topology. Like holes, adding extra characteristics can detect various classes of bumps such as cylinder, rectangular, pyramid, etc. Small bumps can be detected by adding the characteristic of size, for example, compare the volume of the arbitrary shape to the small feature volume parameter value.

3.3. Small Gap

In this section, the method is applied to detection of small gap features frequently found in folded sheet metals. Fig. 11 shows some examples. The characteristics to detect such planar surface with gap feature are:

1. Line edge below a specified length;
2. Immediate adjacent edges are parallel.

For sheet metals situation, the feature processing to remove the gap is relatively straightforward. Extrude the respective surface perpendicular to small edge, as depicted in Fig. 12.

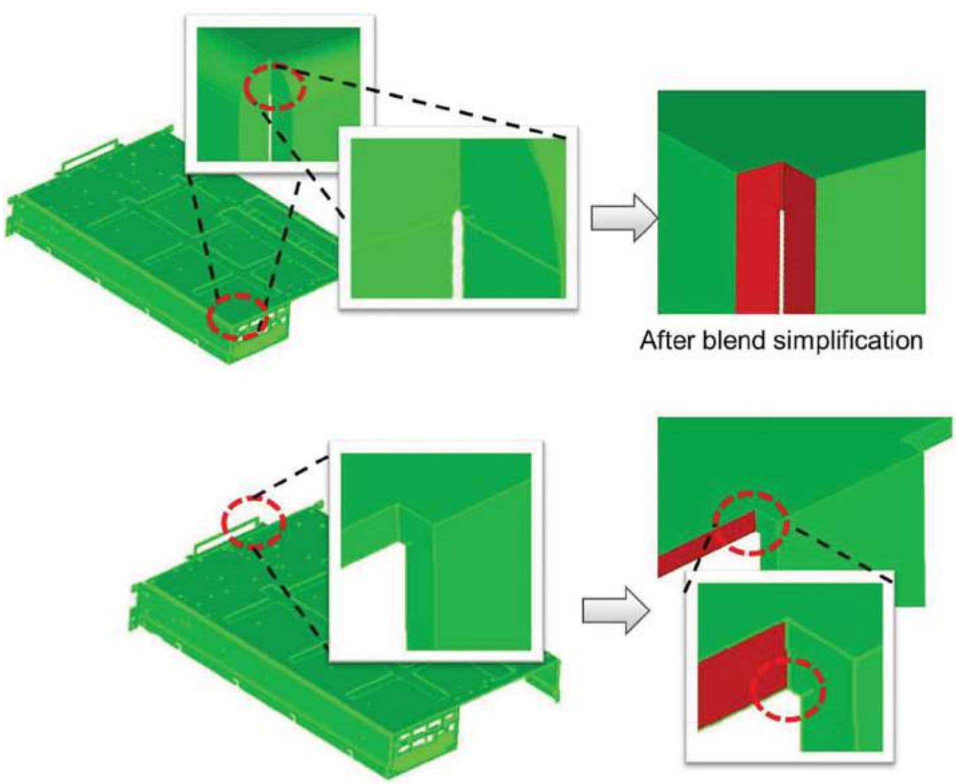

Fig. 11: Some small gap of folded sheet metals.

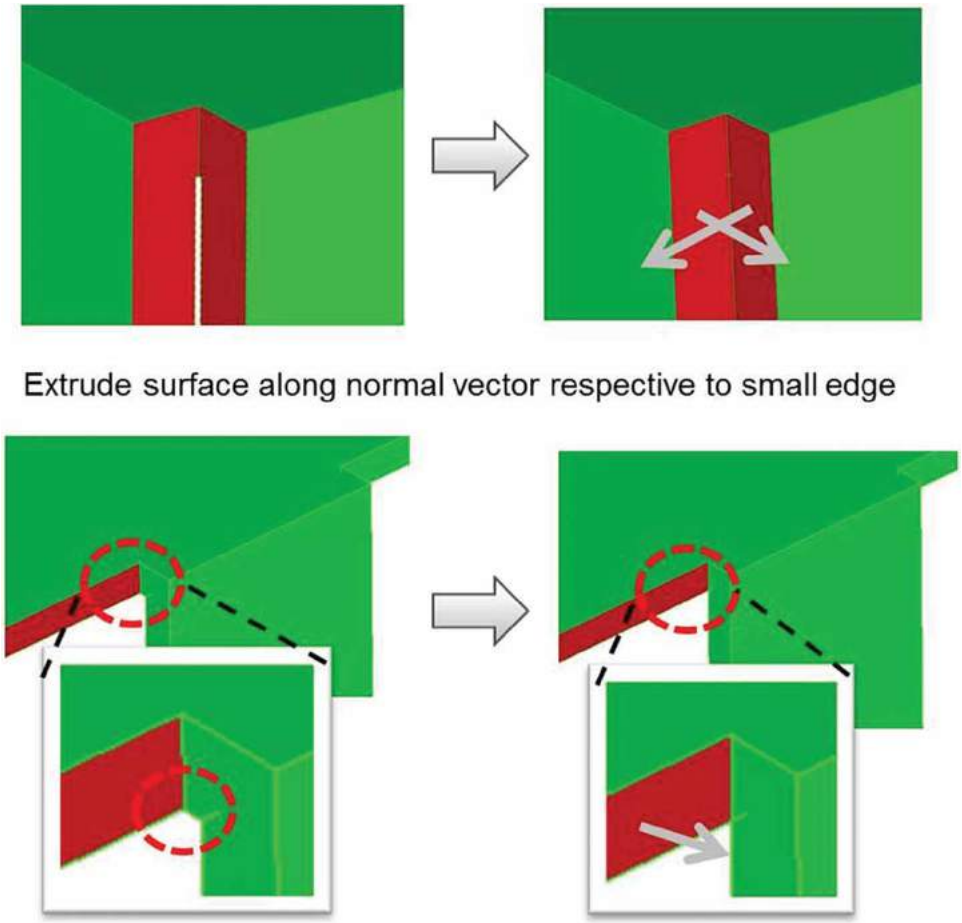

Fig. 12: Removing gap in sheet metals.

4. RESULTS

Figure 13 to 15 shows the three cases used to evaluate the method detection success rate (compared to the manual case) and an idea of the computational performance. The left image is the original CAD model and the right image is the simplified CAD model.

Table 2 shows the result of the three models. The computing performance was achieved on a 64-bit Windows PC with Intel Quad Core at 2.83 GHz and 4GB memory. The result from the CAD model simplification process is fully automatic, once the feature types have been specified. For the cases here, they were: removing blends, chamfers, small holes, small bumps, and gaps, plus modify shape of circle holes to squares. For the assembly model cases, it represents a processing rate of 1.425 and 0.832

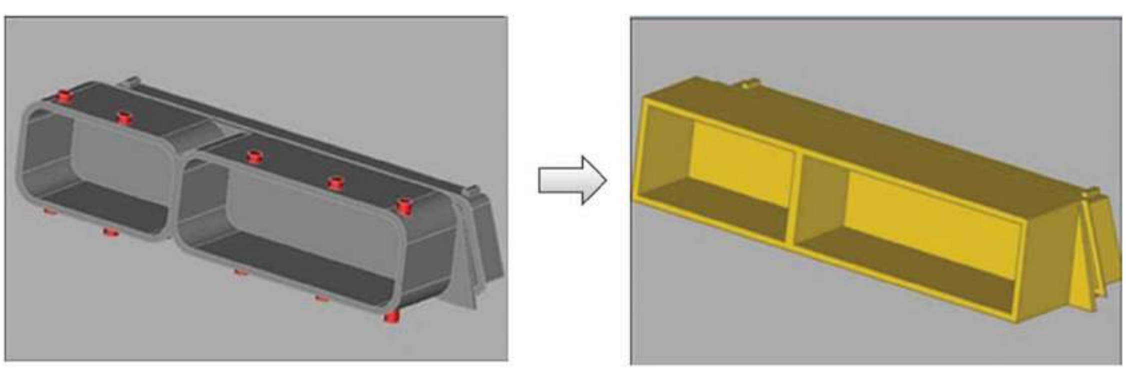

Fig. 13: Connector component model.

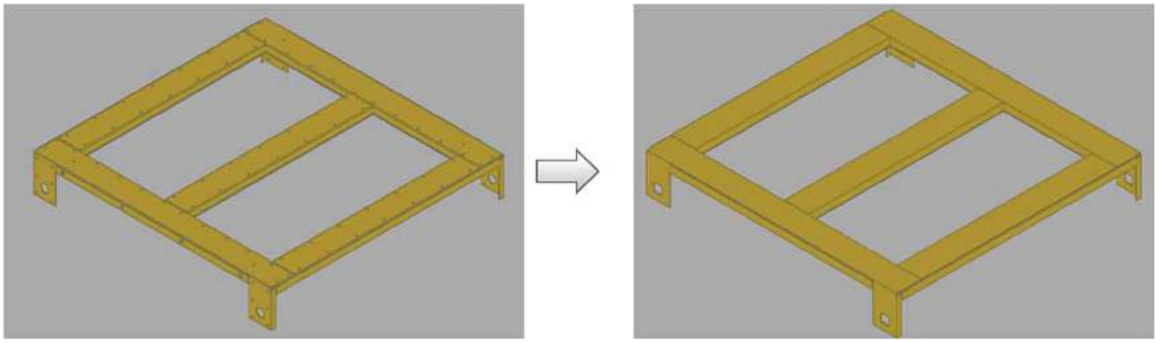

Fig. 14: Assembly-A model, components A1, A2 (2 instances) and A3 (3 instances).

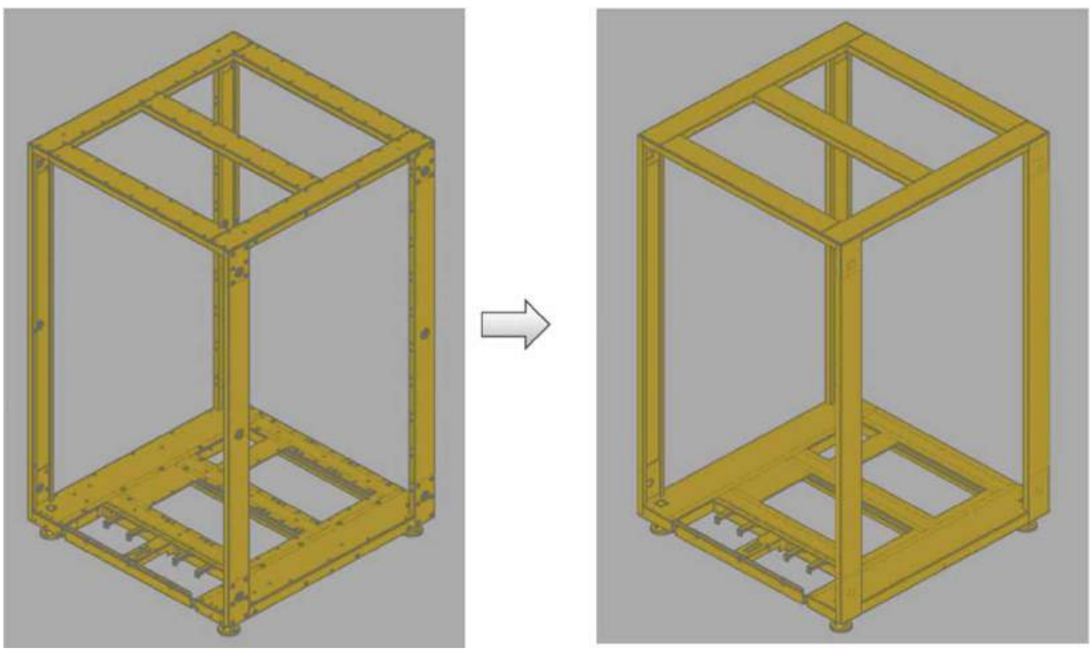

Fig. 15: Assembly-B model with 26 components, 7 sub-assemblies, totaling 267 parts.

Model	*Time (sec)*	*Features*	*Detection Rate*
Connector component	2.42	34	100%
A1 component	3.05	83	100%
A2 component	2.67	52	100%
A3 component	1.89	26	100%
Assembly-A	5.70	-	100%
Assembly-B	21.65	-	100%

Tab. 2: Computing time and detection rate results.

per components, significantly better efficiency than individual component.

5. CONCLUSION

We have described a new method we called "detection by characteristics" for CAD model simplification. The sample cases and results obtained are for server and server rack model class of application. The method is generic and easily extendable, complementary to existing technologies and methods. Presently, the server model in Fig. 1, the 10 days period for CAD-to-CAE model preparation has been reduced to 6 days.

REFERENCES

[1] ACIS, http://www.spatial.com, Spatial Corporation.

[2] Belaziz, M.; Bouras, A.; Brun, J. M.: Morphological analysis for product design, Computer-Aided Design, 32(5–6), 2000, 377–388. http://dx.doi.org/10.1016/S0010-4485(00)00019-1

[3] Bozdoc, M.: The History of CAD, mbdesign.net, 2003, http://mbinfo.mbdesign.net/CAD-History.htm

[4] CADAZZ, CAD software history, 2004, http://www.cadazz.com/cad-software-history.htm

[5] Carlson, W.: A Critical History of Computer Graphics and Animation, Ohio State University, 2003, http://accad.osu.edu/~waynec/history/lesson10.html

[6] Gordon, L.: Direct or history-based modeling?, machine design, 2009, http://machinedesign.com/news/direct-or-history-based-modeling

[7] Han, J. H.; Pratt, M.; Regli, W. C.: Manufacturing feature recognition from solid models: A status report, IEEE transaction on robotics and automation, 16(6), 2000, 782–796. http://dx.doi.org/10.1109/70.897789

[8] Jenkins, B.: 3D Engineering: How 3D direct modeling empowers conceptual engineering and enables simulation-driven product development, white paper, Ora Research LLC, 2009.

[9] Kyprianou, L. K.: Shape classification in computer aided design, PhD thesis, Christ College, University of Cambridge, UK, 1980.

[10] Lee, S. H.: A CAD-CAE integration approach using feature-based multi-resolution and multi-abstraction modelling techniques, Computer-Aided Design, 37(9), 2005, 941–955. http://dx.doi.org/10.1016/j.cad.2004.09.021

[11] Liu, Y.-J.; Lai, K.-L., Dai, G., Yuen, M. M.-F.: A semantic feature model in concurrent engineering, IEEE Transactions on Automation Science and Engineering, 7(3), 2010, 659–665. http://dx.doi.org/10.1109/TASE.2009.2039996

[12] Mounir, H.; Nizar, A.; Abdelmajid, B.: CAD model simplification using a removing details and merging faces techniques for a FEM simulation, Journal of Mechanical Science and Technology, 26(11), 2012, 3539–3548. http://dx.doi.org/10.1007/s12206-012-0869-6

[13] Parasolid, http://www.plm.automation.siemens.com, Siemens PLM Software.

[14] Quadros W. R.; Owen, S. J.: Defeaturing CAD models using a geometry-based size field and facet-based reduction operators, Engineering with Computers, 28(3), 2012, 211–224. http://dx.doi.org/10.1007/s00366-011-0252-8

[15] Rezayat, M.: Midsurface abstraction from 3D solid models: general theory and applications, Computer Aided Design, 28(11), 1996, 905–915. http://dx.doi.org/10.1016/0010-4485(96)00018-8

[16] Sutherland, I. E.: Sketchpad, A man-machine graphical communication system, PhD thesis, Massachusetts Institute of Technology, USA, 1963.

[17] Thakur, A.; Banerjee, A. G.; Gupta, S. K.: A survey of CAD model simplification techniques for physics-based simulation applications, Computer Aided Design, 41(2), 2009, 65–80. http://dx.doi.org/10.1016/j.cad.2008.11.009

[18] Tornincasa, S.; Di Manaco, F.: The Future and the Evolution of CAD, 14th International Research Expert Conference, Trends in the Development of Machinery and Associated Technology, TMT 2010

[19] Woo, Y.: Automatic simplification of solid models for engineering analysis independent of modeling sequences, Journal of Mechanical Science and Technology, 23(7), 2009, 1939–1948. http://dx.doi.org/10.1007/s12206-009-0509-y

[20] Zhu, H.; Menq, C. H.: B-rep model simplification by automatic fillet/round suppressing for efficient automatic feature recognition, Computer-Aided Design, 34(2), 2002, 109–123. http://dx.doi.org/10.1016/S0010-4485(01)00056-2