Defeaturing using Machine Learning

Steven J. Owen, Timothy Shead and Shawn Martin

Abstract New machine-learning based methods for driving defeaturing of CAD models for tetrahedral meshing are proposed. The ability to predict mesh quality at geometric features of a CAD model prior to meshing is used to identify potential problem areas. A prioritized list of geometric operations can be presented to a user to improve meshing outcomes. New methods are introduced for generating training data based on both geometric and topological features of the CAD model with labels defined by local quality metrics. Implementation of the proposed machine learning-driven defeaturing environment is demonstrated in Sandia's Cubit Geometry and Meshing Toolkit.

1 Introduction

An engineering analyst may receive a CAD model or assembly from a designer which may have been developed based on manufacturing specifications which are not directly useful for analysis. Following inspection of the model, the analyst will devise a strategy for model preparation which may include many complex and lengthy geometric modifications including defeaturing. While machine learning is widely used in text, image, audio, and video analysis, there has been little research on the application of machine learning to model preparation for simulation. One notable work in this area from Danglade et. al. [1]. They propose a limited environment for defeaturing CAD models where machine learning is driven by heuristic rule-based outcomes. In contrast, this work proposes the predicted quality of the FEA mesh as the training objective.

Machine learning methods have become widespread and available through robust open source tools such as scikit-learn [2]. These methods require input training

---

Steven J. Owen, Timothy Shead and Shawn Martin
Sandia National Laboratories, Albuquerque, New Mexico, e-mail: sjowen@sandia.gov. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

data in the form of comma-separated value (.csv) files that include feature and label information. For our application we identify features as geometric and topologic information of local curves and surfaces of the CAD model, while label data is based on resulting local mesh quality at these features. Since we would like to drive improvement of the CAD model, we identify features based on selected geometric operations designed to simplify or improve local topology of the CAD model. After collecting sufficient training data, the machine learning models are able to predict local mesh quality and provide a prioritized list of geometric operations for improving the CAD model.

2 Features

The features identified for training data are based upon a series of geometric operations that have proven useful for manually modifying a CAD model. While there are many possible operations we could have considered, our initial study focussed on the following operations available in the Cubit toolkit [3].

(1) remove surface	(6) tweak remove topology curve
(2) tweak replace surface	(7) tweak remove topology surface
(3) composite surfaces	(8) regularize curve
(4) collapse curve	(9) blunt tangency add material
(5) virtual collapse curve	(10) blunt tangency remove material

Each of these 10 operations represent a separate machine learning model that has a unique set of associated features based on nearby geometry and topology. In addition, for comparison, models for vertices, curves and surfaces are also introduced where no operation is performed, making a total of 13 models. This allows identification of those regions in the model that may be most problematic and to predict potential mesh quality improvement compared with a given geometric operation.

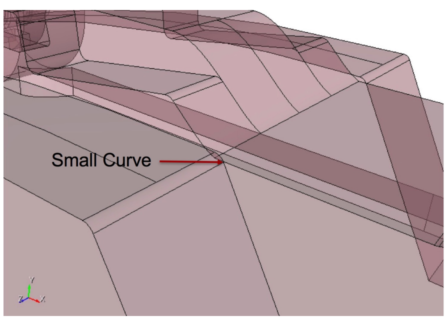

Fig. 1 Small curve in model that requires defeaturing before meshing

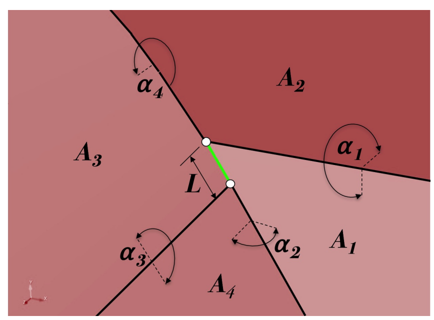

Fig. 2 Example topology-based features used for training data at a small curve

2.1 Topology-based Features

Figure 2 illustrates topology-based features for a given small curve shown in the model in figure 1. In this case, information such as the curve length, adjacent surface areas, angles between neighboring surfaces as well as local vertex valence, loop information and other information is used. The number of features used for each model is based upon the geometric entities involved in the operation. For instance, a composite operation would include information about the surfaces involved in the operation as well as those surrounding. In contrast, a collapse curve operation would include only information about the curve and its adjacent surfaces.

2.2 Geometry-based features

Figures 3, 4 illustrate geometry-based features which are derived from the concept of surflet pairs introduced by Wahl [4]. Surflet, $S = (\alpha, \beta, \gamma, \delta)$ is a function of distance and angles between two normals on the surface as illustrated in figure 5. To maintain a constant size vector of features, a histogram is computed based upon categorization of S values computed between points on the surface. Points to include in the surflet calculations are defined from a local triangulation of the surfaces at the entities involved in the local geometry operation.

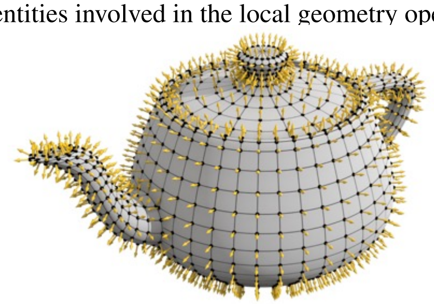

Fig. 3 Example model showing points and normals used for computing surflets

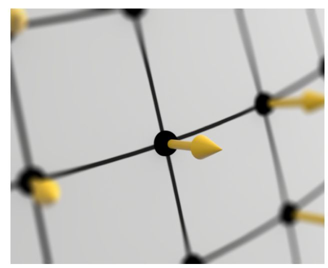

Fig. 4 Close-up of a point and normal used for surflet calculation

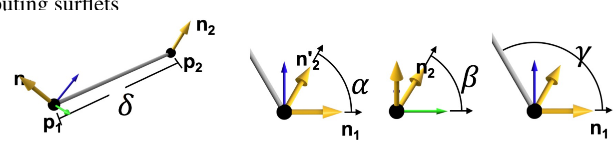

Fig. 5 Surflet, S is a function of the distance, δ and angles α, β, γ between two points and normals on the surface

3 Labels

In order to validate the effectiveness of a geometry operation, the model is meshed [5] following execution of the local operation. A bounding box with dimensions relative to the target mesh size is defined around the entities involved in the geometric operation. Tet elements falling within the bounding box are used to compute a minimum scaled Jacobian m_{SJ} and minimum in-radius m_{IR} value which are used as characteristic labels for the given geometric operation.

4 Machine Learning Models

To generate training data, topology data and feature data is first extracted, and then features computed based on one of the geometric operations described above. The geometry is then meshed, metric labels assigned, and the data written as a single row to one of the 13 operation training data model files. This process is repeated for each *small* feature in the model, where *small* is a function of the target mesh size. A variety of test models were used for the study including several proprietary models as well as those obtained from open internet resources such as grabcad [6]. Once sufficient training data was accumulated for each of the 13 models, scikit-learn was used to compute training models using the effective decision tree (EDT) method. Pickled data files, one for each model, are then written to be used in real-time within the defeaturing tool. The process used for generating training data and utilizing the data in a run-time environment is outlined in figure 7.

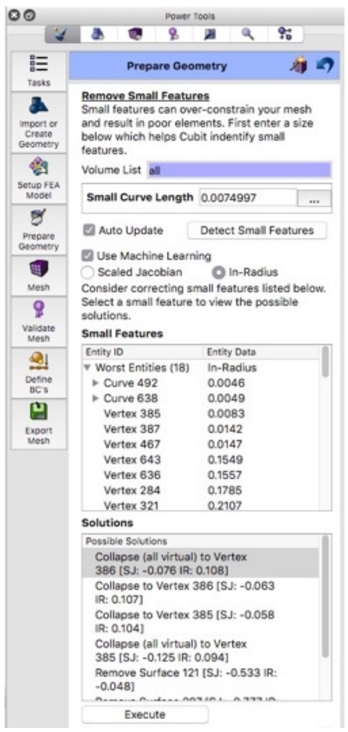

Fig. 6 The Cubit ITEM interface showing enhancements to incorporate machine learning models to drive defeaturing

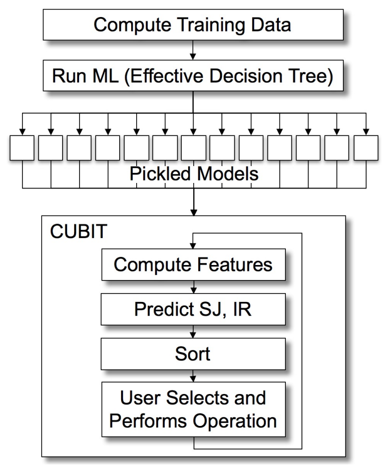

Fig. 7 Proposed process for generating and utilizing machine learning data in a CAD-based defeaturing environment.

5 Application

The Cubit ITEM [7] interface was used as the defeaturing environment for the machine learning models. Figure 6 shows the user interface with a list of prioritized features ranked by the worst quality metric with the predicted local mesh quality. Selecting one of the entities reveals a prioritized list of operations that can be performed along with the predicted mesh quality improvement for each.

6 Results

Results were validated by running a set of computed features not included in the initial models and using the pickled data to predict their m_{SJ} and m_{IR}. Figure 8 illustrates the mean error in m_{SJ} and m_{IR} for each of the 13 operation models. It also compares the results of topology-based vs geometry-based features. In our study we note that topology-based features appear to perform better than geometry-based.

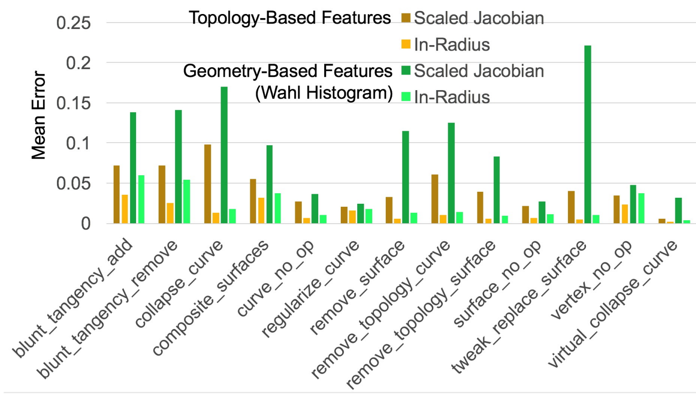

Fig. 8 Results from 13 machine learning models based on Cubit geometry operations. Mean average error from prediction of scaled Jacobian and In-radius from resulting tet meshes.

7 Conclusion

A new application of modern machine learning technologies to model preparation for simulation has been introduced. This initial work is at its beginning stages, but has already proven successful and demonstrated for user-interactive defeaturing of CAD models. New work is underway to expand and apply these powerful machine learning models to everyday practice.

References

1. Danglade, F., J-P. Pernot and V. Philippe (2013) On the use of Machine Learning to Defeature CAD Models for Simulation, Computer Aided Design and Application, Vol 11/3
2. Scikit-Learn, Machine Learning in Python (2018) http://scikit-learn.org
3. Cubit Geometry and Meshing Toolkit, Sandia National Laboratories (2018) http://cubit.sandia.gov
4. Wahl, Eric, U. Hillenbrand, G. Hirzinger, (2003) Surflet-Pair-Relation Histograms: A Statistical 3D-Shape Representation for Rapid Classification, Proceedings 4th International Conference on 3-D Digital Imaging and Modeling, IEEE Computer Society Press, pages 474-481
5. Meshgems-Tetra, Distene, S.A.S. (2018) http://www.meshgems.com/volume-meshing-meshgems-tetra.html
6. GrabCad (2018) https://grabcad.com
7. Owen, S.J., B.W. Clark, D.J. Melander, M. Brewer, J.F. Shepherd, K. Merkley, C.D. Ernst, R. Morris (2008), An Immersive Topology Environment for Meshing, Proceedings, 16th International Meshing Roundtable pp 553-577