Available online at www.sciencedirect.com
SCIENCE DIRECT®
Advances in Engineering Software 37 (2006) 667–683
ADVANCES IN ENGINEERING SOFTWARE
www.elsevier.com/locate/advengsoft

Repairing triangular meshes for reverse engineering applications

Jiing-Yih Lai *, Hou-Chuan Lai

Department of Mechanical Engineering, National Central University, Jhong-Li 320, Taiwan, ROC

Received 19 September 2005; received in revised form 9 January 2006; accepted 19 January 2006
Available online 24 March 2006

Abstract

When an object is digitized and represented in a triangular model, erroneous facets may exist and affect the accuracy of the downstream data analysis algorithms. We here propose an approach to detect and eliminate erroneous facets that might exist in a triangular model. Five types of erroneous facets are identified in this study: degenerate, non-manifold vertices, self-intersection, incomplete connection and inconsistent plane normal. Of these erroneous facets, the first two types must be processed first since they are correlated to the other three types of errors. An individual algorithm is proposed for each type of the errors, and an integrated procedure is then proposed to detect and eliminate all errors automatically. Finally, several examples are presented to demonstrate the feasibility of the proposed method.

Keywords: Erroneous facets; STL data; Reverse engineering

1. Introduction

In reverse engineering, an object is typically digitized by an optical digitizing device. The sample points or range images are then triangulated and saved as an STL file for downstream data processing and/or surface reconstruction. Because of the huge amount of data points that are randomly distributed, some inaccuracies and topological problems may occur on the triangular meshes. Erroneous triangular meshes may result in erroneous topological information in the data structure, and hence make downstream analysis algorithms less effective or even invalid. Therefore, error-free triangular meshes are important for most reverse engineering applications. The objective of this study is to address the problems that are likely to occur on triangular meshes and to propose new algorithms to eliminate these problems.

The file format "STL", related to "stereolithography", was developed by 3D systems to record triangular meshes, using a list of triangular facets corresponding to an object. The general format of an STL file has been previously described [1,2]. Each facet is uniquely identified by a unit normal and three vertices. Since a vertex is recorded again when it shares more than two triangles, redundant information exists in the STL file. In addition, topological information is not captured in an STL file. The erroneous facets could exist and are difficult to detect if no appropriate data structure is established. Some of the errors commonly found in an STL file are: gaps, degenerate facets, overlapping facets, non-manifold points, etc.

Several studies have been conducted to address the problems of triangular or polygonal surfaces for computer graphics, virtual reality, finite elements analysis and rapid prototyping applications. Such problems are primarily due to non-robustness of the algorithms to convert CAD models into triangular and/or polygonal meshes. Gueziec and Taubin [3] identified topological singularities from non-manifold sets of polygons and proposed a cutting and stitching process to create manifold polygonal surfaces. The singularity conditions essentially occurred on vertices and edges, and a cutting procedure was implemented to eliminate them. Two different edge stitching

* Corresponding author.
E-mail address: jylai@ncu.edu.tw (J.-Y. Lai).

0965-9978/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.advengsoft.2006.01.008

operations, pinching and snapping, were then implemented to bind the cracks or holes created in the cutting procedure. Barequet et al. [4,6] pointed out the typical problems on polygonal surfaces, including cracks, degenerates, duplication, holes and overlapping. They then proposed a topologically-based geometric algorithm and implemented in a system called RSVP (repair by shifting vertices of polygons) to yield an adjacency structure of the corrected polygons. Barequet [5] applied a geometric hashing algorithm to repair small gaps in the boundary of the polyhedron. Small gaps were caused by missing surfaces, by incorrect handling of adjacent patches, or most commonly by incorrect handling of trimming curves. A geometric approach was proposed to identify the gaps in the polyhedron and to add new faces by connecting points along the same or different borders. Botsch and Kobbelt [7] focused on the study of degenerate faces on triangular meshes, and proposed a method by combining slicing and decimation to repair the meshes.

Some studies have focused on the STL issues related to rapid prototyping. Petik [8] summarized seven types of errors by software export modules and three types of errors by rapid prototyping machines. Stroud and Xirouchakis [9] proposed three approaches to improve the accuracy of a rapid prototyping process. One of them is the interaction of various STL approximation control parameters such as the chord length, the normal vector tolerance, the triangle side length, etc. to improve the approximation of the CAD model. For the purpose of rapid prototyping application, Leong et al. [10,11] classified the erroneous triangular meshes into four types, namely, missing facets, degenerate facets, overlapping and non-manifold. That study paid more attention on the problem of missing facets since they seriously affect the layer building process. A generic solution has been proposed to deal with the missing facets problems [10]; and additional methods have been proposed to deal with special cases of missing facets [11]. The problems of missing facets are also called hole-filling problems and have been subsequently studied by other investigators, such as Wang and Manuel [12], Davies et al. [13] and Jun [14].

The advantage of triangular meshes over cloud points is that they can build the topological information among the facets, which is almost impossible for unorganized cloud points. Therefore, the generation of a connectivity data structure is the first step to develop effective algorithms for triangular meshes. Melax et al. [2] proposed a data structure, primarily based on the pointer, to build the topological information among the triangular facets. The vertices are related to each other by pointers, and each vertex records both the number and the indices of triangular facets neighboring to it. Each triangular facet records the indices of three vertices neighboring to the facet. The topological information of edges is not recorded in such a data structure. Botsch et al. [15] proposed a half-edge data structure, primarily based on the pointer also, to record the topological information of the vertices, edges and facets. Such a data structure is suitable both for triangular and polygonal surfaces. In addition, they proposed a method to save the data efficiently so that the triangular mesh can be added or deleted dynamically without affecting the other parts of the data.

The optical devices used in reverse engineering include point-type, line-type and range data capture. In principle, the distribution of the point-type data is more irregular, depending largely on the measurement mode. The range data, obtained by CCD camera in associated with image processing, are ideally distributed in a grid mode. However, multiple views of digitization are usually required to capture the entire object, which affects the distribution of the data, since the overlapping data in multiple views results in irregular distribution of the data. Triangulation of the cloud points is highly affected by such a factor. In addition, registration is a process to merge the coordinate systems of multiple sets of triangular meshes. The overlap area must be removed and a stitching procedure must be implemented to detect and bind the boundary of two separate triangular meshes. Triangulation and registration are the two main causes for the presence of erroneous triangular meshes.

Computational efficiency is usually a crucial issue for dealing with triangular meshes. Although the number of vertices in a typical set of triangular meshes may be up to million points, the computational time is measured in seconds, rather than minutes. Thus, the computational algorithm should be as efficient as possible in order to enhance the computational speed. Searching a set of vertices near a given vertex or a set of facets near a given facet is obviously the most important issue affecting the computational speed. If the entire set of vertices or facets is used in the search, the computational speed would be very slow. Therefore, the development of an efficient algorithm in search will be the most important factor to determine if an algorithm is practically feasible. We here propose a subdivision algorithm to divide the bounding box of the vertices into cells and assign each vertex into an appropriate cell. The main idea is to limit the number of vertices in each search, and hence improve the computational speed as much as possible.

The problems of triangular meshes may vary according to the particular application. Leong et al. [10] classified the typical problems encountered in rapid prototyping as gaps (crack, holes and punctures), degenerates, overlapping and non-manifold. Barequet [4] pointed out five types of errors in CAD Models repairing: cracks, degenerates, duplication, holes and overlapping. These above two classifications are essentially similar. These issues have also been studied by other investigators, such as zero-volume facets [16], degenerate facets [7], manifold surfaces [3], holes filling [6,17–19], managing adjacency [20] and topological information in meshes [21]. Petik [8] pointed out other types of errors that typically appear in CAD to STL conversion, such as chordal tolerance, angle control, truncation error, convex boundary error, etc.

In reverse engineering, the triangular model is generated based on the cloud points instead of the CAD model, and