Computer-Aided Design 37 (2005) 263–270

COMPUTER-AIDED DESIGN

www.elsevier.com/locate/cad

A piecewise hole filling algorithm in reverse engineering

Yongtae Jun*

Department of Mechanical Engineering, Sejong University, Gunja-dong 98, Gwangjin-gu, Seoul 143-747, South Korea

Received 30 December 2003; received in revised form 16 June 2004; accepted 17 June 2004

Abstract

While scanning a complex part in reverse engineering, it is not possible to acquire all part of the scanned surface. Data are inevitably missing due to the complexity of the scanned part or imperfect scanning process. Missing scanned data cause holes in the created triangular mesh, so that a hole-free mesh model is prerequisite for fitting watertight surfaces. Although a number of hole filling algorithms have been investigated, they enable to fill holes only on the smooth regions of a model. They are not always robust in the regions of high curvature. This paper proposes a novel methodology that can automatically fill complex polygonal holes with a piecewise manner. It incrementally splits a complex hole into several simple holes with respect to the 3D shape of the hole boundary, and then it consecutively fills each divided simple hole with planar triangulation method until the entire complex hole is firmly closed. Finally smoothing and subdivision techniques are applied for enhancing the hole triangles. The newly created vertices and triangles are added to their respective lists and the topology information is updated. The method has proven to be robust and effective from the result of test with a variety of complex holes. Examples are given and discussed to validate the methodology.

Keywords: Piecewise hole filling; Triangular mesh; Reverse engineering

1. Introduction

Reverse engineering (RE) is a process to create CAD models of a physical part. It begins with scanning the surfaces of a target physical part to take data points by appropriate 3D scanners. Then, the scanned point cloud is refined and edited with the point- or polygon-based technologies such as sampling, triangulating, registering/merging, smoothing, hole filling, etc. In the final stage, CAD models or physical replicas are created from the refined polygonal data. RE process has helped to reduce product introduction costs and lead-times since designers often need to physically interact with their design for testing the design performances and obtaining an optimal product modification. It encompasses many engineering approaches in which an existing product is investigated either prior to or during the reconstruction process. That is, the existence of geometric models provides enormous profits in improving the quality and efficiency of design, manufacture, and analysis. RE can also be applied into some recent emerging applications such as custom-made manufacture, design simulation, medical application, animation, sculpture, digital duplication, etc [1,2].

We have been developing a fully automated RE system to duplicate a 3D part [3,4]. The actual duplicating time depends on the size and the complexity of the part. The system employs four non-contact 3D scanners based upon the coded structured light, a suite of polygon-based data processing modules, and a multi-axis NC machine. The scanners capture the shape of a scanned part sequentially rather than simultaneously, so that the first scanner captures the front image of the part and others capture the left, the right, and the rear shape, one by one. In order to achieve a full 3D model, some more scans are usually taken and aligned. A main problem in the proposed automated RE system is restoration of missing data during data capture due to the almost infinite range of shapes to be scanned. Data that close to cavities and other small sized concave region is fairly easy to be missed. If the concave region is smaller than the angle created by the triangulation process of employed structured light scanners, the projector

* Tel.: +82-2-3408-3784; fax: +82-2-3408-3333.
E-mail address: ytjun@sejong.ac.kr.

0010-4485//$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cad.2004.06.012

and sensors can neither project nor see the light. Unseen portions of the scanned part will appear as holes that must be filled and connected to form a continuous shape in the reconstruction since hole-free (watertight) polygonal models are useful for: fitting surfaces to polygons, manufacturing models, and aesthetic renderings. Thus it must be necessary to close all the holes in order to successfully automate RE process.

3D hole filling methods have recently drawn attention from researchers in surface modelling and polygon-based modelling area. Some researchers investigated algorithms to fill polygonal holes on parametric surfaces. Levin [5] presented an algorithm for computing N-sided surface patches that satisfy C^1 continuity. The algorithm generated a subdivision surface that connects smoothly to given surfaces surrounding a polygonal hole. Chui et al. [6] used the method of energy minimization to fill any number of sided of polygonal holes. They used C^1 piecewise cubic triangular spline functions to construct the filling surfaces. The potential energy of the filling surfaces was minimized to reduce the bumpness of the surface. Several algorithms have also been proposed for filling polygonal holes. Curless and Levoy [7] proposed a hole filling approach to interpolate non-sampled surfaces in concave regions of volumetric parts. They produced watertight models using a volumetric method of filling holes in voxel space for reproduction using rapid prototyping techniques. Carr et al. [8] used radial basis functions (RBF) to compute an implicit representation to a set of holes in scattered data. The existence of an implicit representation supports the smooth reconstruction of hole regions. Liepa [9] described a method for filling holes in unstructured triangular meshes. The resulting patching meshes interpolate the shape and density of the surrounding mesh. However, most hole filling research works made an assumption that the holes are relatively small with respect to the entire model and oriented boundary curve (simple hole) in the 3D space. Their algorithms were confined to holes in oriented connected meshes in smooth regions, so that they are not always robust on regions of high curvature.

The contribution of this paper is to build an automated algorithm that fills complex holes. It identifies polygonal holes in a 3D triangular mesh and consecutively fills them regardless of the shape complexity of the holes. The proposed algorithm is based upon a piecewise scheme. It incrementally divides a complex hole into several simple sub-holes with respect to the complexity of the hole. Then all the sub-holes are sequentially filled with the planar triangulation until the whole complex hole is firmly closed. Smoothing and subdivision modules are also applied to the new triangles on the hole to refine the shape quality with existing neighbouring triangles. Finally the newly created vertices and triangles are added to their respective lists and the topology information is updated.

The remaining of the paper is organized as follows: Section 2 describes preliminaries and basic algorithm to fill simple holes. The details of the proposed algorithm to fill complex holes are precisely described in Section 3. The implementation of the proposed method and several examples are given and discussed in Section 4. Finally some concluding remarks are given in Section 5.

2. Basic of filling holes

2.1. Preliminaries

A polygonal *mesh* is defined as a set of vertices and a set of triangles. Two triangles normally share a common edge. A *boundary edge* is an edge that only connects to one triangle and a *boundary vertex* is a vertex that is adjacent to a boundary edge. A *hole* is a closed loop of boundary edges. Polygonal holes fall into two categories: *simple hole* and *complex hole*. Some of polygonal holes can be simply filled with planar triangulation if all the boundary edges of the hole can be projected on a plane without self-intersection (Fig. 1a). The holes are denoted as *simple holes*. The main idea of planar triangulation is to fill holes with flat triangles on the plane where the boundary edges are projected. By inserting several new triangles, the hole can be easily filled. Although all simple holes have been filled with a planar triangulation, other holes may leave unfilled in the triangulation due to the amount of shape complexity. These holes are denoted as *complex hole* since its boundary edges cannot be projected on any plane without self-intersection (Fig. 1b).

Most data structures of polygon-based applications store a list of vertices and a list of triangles that contain indices into the vertex list. To efficiently handle the triangles there is the need for data structures supporting efficient operations on the object's position and capable of handling queries on attributes. The main data structure in the paper is based upon a *triangle-based data structure*. It associates with six data members: three pointers to adjacent triangles and three pointers to vertices.

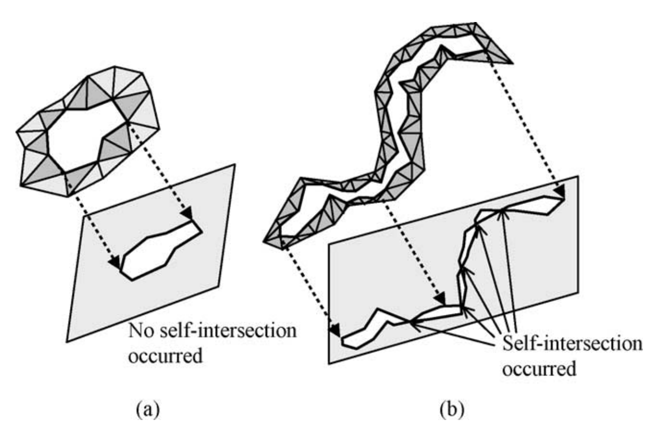

Fig. 1. Projection of polygonal holes on a plane. (a) Simple hole; (b) complex hole.