ELSEVIER

Computer-Aided Design 36 (2004) 1337–1355

COMPUTER-AIDED DESIGN

www.elsevier.com/locate/cad

Local topological beautification of reverse engineered models

C.H. Gao, F.C. Langbein*, A.D. Marshall, R.R. Martin

School of Computer Science, Cardiff University, PO Box 916, 5 The Parade, Cardiff, Wales CF24 3XF, UK

Received 20 June 2003; received in revised form 5 February 2004; accepted 11 February 2004

Abstract

Boundary representation models reconstructed from 3D range data suffer from various inaccuracies caused by noise in the data and by numerical errors in the model building software. The quality of such models can be improved in a *beautification* step, where geometric regularities need to be detected and imposed on the model, and defects requiring topological change need to be corrected. This paper considers changes to the topology such as the removal of short edges, small faces and sliver faces, filling of holes in the surface of the model (arising due to missing data), adjusting pinched faces, etc. A practical algorithm for detecting and correcting such problems is presented. Analysis of the algorithm and experimental results show that the algorithm is able to quickly provide the desired changes. Most of the time required for topological beautification is spent on adjusting the geometry to agree with the new topology.

Keywords: Beautification; Healing; Topological modification; Reverse engineering; Geometric modelling

1. Introduction

Reverse engineering the shape of a 3D object is the process of reconstructing a geometric model of an object from measured data [18]. The general procedure consists of measuring surface points on an object, usually with a 3D laser scanner, merging multiple views into a single registered data set, segmenting the point set, fitting surfaces to each point subset, and stitching these into a solid model. Our goal is to create a system that, for simple engineering objects, reconstructs a boundary representation (B-rep) model from a physical object, with a minimum of human interaction. It should be usable both by naive users and engineers. In particular, the generated model should have all the intentional *geometric regularities* present in the original, ideal, design of the object, to ensure that the model has maximum utility for manufacturing, redesign, etc. The model should also have the expected *topology*, for example, if we reverse engineer a four-sided pyramid, we expect all four sloping faces to meet at a single vertex.

In reverse engineering, numerical errors occur in the reconstruction algorithms, and noise is present in the measured data. Improving the sensing techniques and the reconstruction methods can reduce errors, but some will always remain. Additional errors may be present due to wear of the object before scanning, and the particular manufacturing method used to make the object (e.g. if the object was cast in a mould, draught angles may have been added). Note that we wish to recover a geometric model of the *ideal* object as conceived by the designer. However, reverse engineering often fits each face individually, and treats it independently of the other faces in the model, losing regularities present in the original design. We propose to improve the reconstructed B-rep model by adjusting it in a separate *beautification* post-processing step. This paper in particular considers the problem of detecting and making any necessary *topological* (and consequent geometric) adjustments to the model; our earlier work considered *geometric* beautification without topological change [6–10,13,14].

For example, if a four-sided pyramid is reverse engineered, and each sloping face is fitted to data points independently, any three of these faces will intersect in a point, but it is extremely unlikely all four as fitted will pass through a single point (Fig. 28). Thus, the initial geometric model created will have either a very short edge or a very small face instead of a vertex at its apex, depending on exactly how the software produces a B-rep model by intersecting and stitching the individual faces. The topology

* Corresponding author. Tel.: +44-29-2087-0110; fax: +44-29-2987-4598.

E-mail address: f.c.langbein@cs.cf.ac.uk (F.C. Langbein).

0010-4485/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cad.2004.02.004

of such a model needs to be adjusted, in conjunction with the geometry of the sloping faces, to produce a new model in which all four sloping faces pass through the same point.

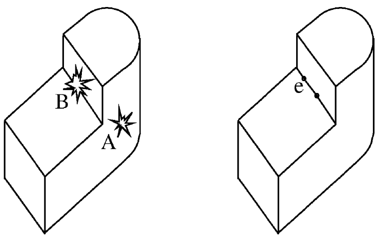
1.1. The topological beautification problem

In this paper, we specifically address topological beautification of 'conventional' reverse engineered models bounded by planar, spherical, cylindrical, conical, and toroidal faces. However, the ideas presented are likely to be applicable to models containing free-form surfaces, too, even if we have not specifically considered such cases. In particular we consider a specific set of problems listed below, where beautification requires adjusting the topology of the model besides the geometry. In the following we refer to these as *topological problems*. Note that this does not mean that the initial topology is invalid, but rather that changes to the topology are needed to resolve such problems.

All such problems depend upon a notion of 'small', e.g. we intend to remove only small, spurious faces. The tolerances employed to decide about this are discussed in Section 3.1.

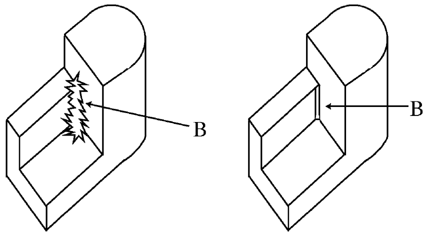
The problems (and their resolutions) listed below have been identified as the ones which can arise from earlier model building processes. They do not represent a list of all possible topological problems, but are problems which are likely to arise during reverse engineering of models. The detailed cases which have to be considered to remove such problems are described in Sections 3.2 and 4.

- *Removing gaps in a single face.* A loop of half-edges may exist in the interior of a face, with nothing on the other side of the loop. Such a case may arise, for example, where the scanner did not collect any data from within a deep concavity in the face. Here the loop of half-edges should be removed, extending the face (see gap A in Fig. 1).
- *Removing gaps crossing an edge.* A loop of half-edges may span two faces, with nothing on the other side of the loop. The edge between the faces is divided into two pieces by the gap. The gap should be removed, the existing faces extended, and the two edge pieces joined (see gap B in Fig. 1).
- *Removing gaps spanning multiple faces.* A loop of half-edges may span multiple faces, with nothing on the other side of the loop. Existing faces and edges must be extended to fill the gap, and new vertices and edges must be added as necessary (see gap B in Fig. 2).
- *Adjusting pinched faces.* If a face narrows to a very thin part it is *pinched*. Other parts of the model should be adjusted to remove the thinning, resulting in a change in connectivity of the face; a face may be split into two faces (Fig. 3).
- *Removing chains of small faces.* Faces should meet in an edge, but instead a chain of small faces may separate them. The chain of small faces should be replaced by an edge (see Fig. 4, where the first step is to reduce a chain of small faces to a chain of short edges).
- *Removing sliver faces.* Two faces should meet in an edge, but instead a long very thin face (a *sliver* face) may separate them. The sliver face should be replaced by an edge (Fig. 5).
- *Removing chains of short edges.* Several consecutive short edges may need to be replaced by a single long edge (Fig. 4). This is in particular a problem which may result from repairing some of the other problems listed.
- *Merging adjacent faces with the same geometry.* Two adjacent faces may share the same geometry across a contiguous edge sequence. Edges and vertices as appropriate should be removed, and the faces merged (Fig. 6).
- *Removing isolated small faces.* Several edges should meet in a single vertex, but instead they meet at several



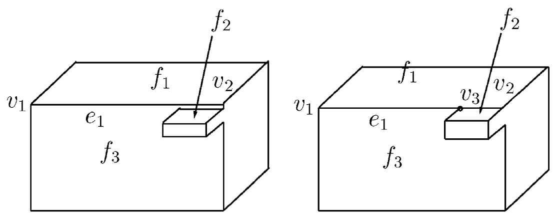
(a) Before modification (b) After modification

Fig. 1. Repairing a face gap and an edge gap.



(a) Before modification (b) After modification

Fig. 2. Repairing a complex multiple face gap.



(a) Before modification (b) After modification

Fig. 3. Repairing a *spiking* pinched face.