Computers & Graphics 36 (2012) 101–110

Contents lists available at SciVerse ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Technical Section

Automatic hole-filling of CAD models with feature-preserving ☆

Xiaochao Wang [a], Xiuping Liu [a,*], Linfa Lu [c,d], Baojun Li [e], Junjie Cao [a,b], Baocai Yin [f,*], Xiquan Shi [g,1,*]

[a] School of Mathematical Sciences, Dalian University of Technology, Dalian, China
[b] Department of Engineering Mechanics, State Key Laboratory of Structural Analysis for Industrial Equipment, Dalian University of Technology, Dalian, China
[c] School of Information Science & Technology, Sun Yat-sen University, Guangzhou 510006, China
[d] National Engineering Research Center of Digital Life, Guangzhou 510006, China
[e] School of Automotive Engineering, Faculty of Vehicle Engineering and Mechanics, State Key Laboratory of Structural Analysis for Industrial Equipment, Dalian University of Technology, Dalian, China
[f] Beijing Municipal Key Lab of Multimedia and Intelligent Software Technology, College of Computer Science and Technology, Beijing University of Technology, Beijing 100124, China
[g] Department of Mathematical Sciences, Delaware State University, Dover, DE 19901, USA

ARTICLE INFO

Article history:
Received 21 June 2011
Received in revised form
28 November 2011
Accepted 12 December 2011
Available online 23 December 2011

Keywords:
Hole-filling
Feature-preserving
Feature set matching
Corner recovery

ABSTRACT

In this paper, we propose an automatic hole-filling method, particularly for recovering missing feature curves and corners. We first extract the feature vertices around a hole of a CAD model and classify them into different feature sets. These feature sets are then automatically paired, using ordered double normals, Gaussian mapping and convex/concave analysis, to produce missing feature curves. Additionally, by minimizing a newly defined energy, the missing corners can be efficiently recovered as well. The hole is consequently divided into simple sub-holes according to the produced feature curves and recovered corners. Finally, each sub-hole is filled by a modified advancing front method individually. The experiments show that our approach is simple, efficient, and suitable for CAD systems.

1. Introduction

Triangular meshes are widely used to represent 3D models. However, for many reasons, such as occlusion, limitation of scanners and the damaging of original models, triangular meshes may contain holes. These holes make it difficult for many subsequent operations, such as model rebuilding, rapid prototyping and finite element analysis. So, it is necessary to fill holes in a reasonable manner. In addition, hole-filling also plays important roles in other applications, such as feature suppression [1], mesh merging [2] and mesh parametrization [3].

A large number of hole-filling methods have been proposed in the current literatures. Most of them work well for small holes located on smooth regions; however, it is still a challenge to fill larger and complex holes with missing sharp features. Due to the diversity and complexity of the holes, none of the existing methods works for all holes. In this paper, we focus on the hole-filling problem of incomplete piecewise smooth meshes that may have holes locating on feature regions. In addition, holes may also contain missing corners. Inspired by the idea of divide-and-conquer, we propose an automatic feature-preserving hole-filling framework for restoring the missing corners and sharp edges. Besides recovering the feature structures, our results are of high quality in approximating the original meshes. On top of that, we also present a new energy function to recover the missing corners.

To recover the missing sharp features, three main steps are involved. First, feature vertices around the hole are extracted and classified into different feature sets. Then, we automatically match the feature sets to construct the missing feature curves to divide the original hole into some simple sub-holes. A simple sub-hole is a hole of containing not any feature boundary vertex except those of sharing as common boundary vertices with other sub-holes. Finally, the sub-holes are filled by the modified advancing front method (MAFM) [4]. In this process, the reconstructed feature curves are preserved unchanging once they are constructed. If there are missing corners in the hole, they can also be efficiently recovered by minimizing a newly defined energy.

The main contributions of this paper can be summarized as follows:

- The extracted feature vertices are automatically paired to recover the missing sharp features, which can avoid tedious user interactions and thus improve the algorithm's efficiency.
- The missing feature curves are explicitly reconstructed by cubic splines, which interpolate the corresponding feature

☆This article was recommended for publication by A. Tal.
* Corresponding authors.
E-mail addresses: xpliu@dlut.edu.cn (X. Liu), ybc@bjut.edu.cn (B. Yin), xshi@desu.edu (X. Shi).
[1] Part of Xiquan Shi's work is finished when he visited Dalian University of Technology.

0097-8493/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cag.2011.12.007

vertices and guarantee the accuracy of the recovered feature structures.
- Significantly different from previous works, the missing corners can be recovered by minimizing a newly presented energy before the hole is filled.

The rest of the paper is organized as follows. The related work is briefly reviewed in Section 2. Section 3 gives the outline of our method and the details are described in Section 4. Experimental results are presented in Section 5, and Section 6 concludes the paper.

2. Related work

Numerous hole-filling methods have been proposed. Generally speaking, they can be classified into two categories: volume-based methods and mesh-based methods [5].

Volume-based methods. The input model is first converted into an intermediate volumetric grid. After hole-filling, the volume presentation is again converted to a triangular mesh by different iso-surface extraction methods [6–12]. Davis et al. [7] defined a signed distance function in the vicinity of observed surfaces. Then a diffusion process was applied to extend this function through the volume until its zero set bridged all holes. A voxel-based method [8] was proposed to simplify and repair polygonal models by adopting open and close morphological operators. Bischoff et al. removed typical mesh artifacts by using volumetric geometry representation in [10,11]. Hétroy et al. [13] extracted a valid two-manifold surface from a voxel set in geometrical and topological defect regions in a user-friendly manner.

Volume-based methods excel in robustness in filling complex holes. However, as stated in [5], most volume-based methods use the Marching Cubes algorithm for reconstruction, which generates blobby surfaces to result in losing the sharp corners and edges of the input model. Although feature-preserving contouring algorithm [14] has been proposed and used in [9,10], it still cannot efficiently reproduce all geometric features, especially the missing sharp features.

Mesh-based methods. Unlike volume-based method, mesh-based methods fill holes locally with the rest of the surface unchanged. Barequet and Sharir [15] combined matched border stitching and minimum area triangulation to fill holes. Instead of a simple step hole-filling, Liepa [16] used refinement and fairing to refine the obtained patch mesh, so that the triangle density agrees to that of the surrounding mesh. This method is further used in [17]. Zhao et al. [18] first got an initial patch mesh by advancing front method (AFM), then optimized vertex's position by solving the Poisson equation. Pernot et al. [19] filled holes by minimizing the curvature variant between the surrounding and inserted meshes. However, for some complex cases, stiffener lines have to be created manually. This method was further improved in [20].

Jun [21] presented a piecewise hole-filling algorithm for complex holes. The algorithm incrementally splits a complex hole into simple sub-holes with respect to the 3D shape of the hole boundary, and then consecutively fills each sub-hole.

Different from filling holes directly on 3D meshes [22,23], triangulated holes on 2D parametric domain and embedded the triangulation into 3D meshes by minimizing some energy functions. The hole-filling patches can also be obtained by adapting scattered data fitting techniques, such as radial basis function (RBF) [24–26] and moving least squares [27].

Although the methods mentioned above work well for holes located on smooth regions, most of them cannot propagate feature structures therein. The features involved in hole-filling can be divided into two categories. One is geometric detail or texture structure, such as the geometric details of a golf or the hair region on a head. The other is sharp features, such as sharp edges or corners in CAD models, on which is what this work focusing.

Handling geometric details. For reconstructing geometric details of the missing holes, a variety of methodologies have been proposed. In the example-based methods [28–30] or context-based methods [31,32], holes are iteratively filled by copying similar matching patches from the input model itself or from some existing models. Kraevoy et al. [33] first computed a mapping between the incomplete mesh and a template model, then employed this mapping to glue together the components of the input mesh and to fill the holes simultaneously. Nguyen et al. [34] reconstructed 3D geometric information of holes from synthesizing local gradient images on 2D parametric domain. Xiao et al. [35] presented a texture synthesis and context based method for appearance and geometric completion of point set surfaces. Recently, Li et al. [36] developed a skull completion method by combining symmetric detection and template-based method.

Handling sharp features. Comparing to handling geometric details, a few methods exist for processing sharp features. Chen et al. [25,26] reconstructed sharp features via sharpness dependent filter [37]. Two processes are involved in this approach: producing an initial repaired model by using RBF based interpolation method and then recovering the sharp features by a sharpness dependent filter.

Although [25,26] can recover sharp features, there are two potential problems. First, RBF cannot provide well-shaped initial patch meshes for sharpness dependent filtering when the hole is big or complex. Second, the sharpness dependent filter has an important parameter that should be carefully selected to prevent the positions of the features unexpectable. These two drawbacks may very easily prevent the method from recovering the missing sharp features of the input models.

Following the piecewise scheme of [21], a feature-preserving hole-filling method was presented in [38] by using polynomial blending technique. Polynomial blending curves are constructed based on the detected feature vertices around the hole to complete the missing parts of the feature curves. Generally speaking, this method has two limitations. One is that the feature curve segments around the hole are matched interactively. The other is that the method cannot recover missing corners. To overcome these limitations, an automatic hole-filling method is proposed in this paper, which can not only reconstruct the missing feature curves, but also recover the missing corners.

3. Method overview

Following most mesh-based methods [16,18–20], we assume that all meshes are triangular, oriented, manifold and connected, with boundary is allowed. That is, two separate holes will have no vertices in common and any hole contains no islands (otherwise the mesh cannot be connected). A triangular mesh is represented by $\mathcal{M}=(V,E,F)$, where $V=\{\mathbf{v}_1,\mathbf{v}_2,\ldots,\mathbf{v}_n\}$ denotes the set of vertices, E denotes the set of edges and $F=\{f_1,f_2,\ldots,f_m\}$ denotes the set of faces.

The flowchart of our framework is shown in Fig. 1. At the beginning, the method described in [18] is adopted to detect all holes of the input mesh. To this end, we first identify all boundary vertices based on the property that the numbers of a boundary vertex's 1-ring triangles and 1-ring edges are not equal. Then, starting from any boundary vertex, a set of connected boundary edges are traced until getting a closed loop. This loop should be