Computer-Aided Design 42 (2010) 213–220

Contents lists available at ScienceDirect

Computer-Aided Design

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

Patch layout from feature graphs

Matthias Nieser, Christian Schulz *, Konrad Polthier

Freie Universität Berlin, Mathematical Geometry Processing Group, Arnimallee 6, 14195 Berlin, Germany

ARTICLE INFO

Article history:
Received 8 April 2009
Accepted 5 November 2009

Keywords:
Reverse engineering
Surface decomposition
Curvature based segmentation

ABSTRACT

The structuring of surface meshes is a labor intensive task in reverse engineering. For example, in CAD, scanned triangle meshes must be divided into characteristic/uniform patches to enable conversion into high-level spline surfaces. Typical industrial techniques, like rolling ball blends, are very labor intensive.

We provide a novel, robust and quick algorithm for the automatic generation of a patch layout based on a topology consistent feature graph. The graph separates the surface along feature lines into functional and geometric building blocks. Our algorithm then thickens the edges of the feature graph and forms new regions with low varying curvature. Further, these new regions – so-called fillets and node patches – will have highly smooth boundary curves, making the algorithm an ideal preprocessor for a subsequent spline fitting algorithm.

1. Introduction

Reverse engineering deals with the reconstruction of CAD surfaces, typically from scanned three-dimensional (3D) geometries. Since current CAD system are based mainly on spline geometries, a scanned triangle mesh must be converted into a highly structured and segmented data structure. Our algorithm aims to automate the reconstruction process. It is a two-step process. In the first step we generate the topology of the final patch layout. This topology is encoded in a feature graph;, i.e., there exists a one to one relation between the feature graph elements (such as nodes, edges and regions) and the patches of the final layout. Furthermore, the feature graph is an intersection free graph embedded on the surface, whereas its smooth edges are oriented along geometric surface features. In the second step, a geometrically reasonable patch layout is generated out of the feature graph. The resulting patches have a uniform curvature distribution and are encircled by smooth boundaries. Such an automatic algorithm avoids many labor intensive manual segmentation approaches.

1.1. Previous work

Our patch layout algorithm is related to many previous techniques in surface segmentation and graph smoothing algorithms.

A general overview about surface decomposition methods is given in [1]. It starts with its roots in image processing, where surfaces are treated as height fields; i.e., there exists a canonical parameterization of the surface over a planar domain, as used in [2]. The main part of [1] contains a detailed overview of segmentation algorithms working on general triangulated surfaces, showing their variety of applications and implementations. Due to different aims, the possible objectives range from remeshing, animation [3], shape matching [4], mesh editing to geometry compression and other areas. Here, we focus on segmentation of CAD parts for reverse engineering.

Some related work focuses on surface segmentation by approximation with several kinds of predefined types of primitives. In [5], planes are being fitted, whereas [6] uses a collection of CAD primitives, such as spheres or rolling ball blends.

The use of parameterized shapes is suggested in [7]. This idea is further explored in [8], where the notion of morphological properties of shape templates is introduced. In this work the author proposes a two-step generic algorithm to identify a surface part with an instance of a shape template. It starts to assign a shape instance to a surface region by varying its morphological properties, followed by an embedding of the matched template into the surface.

Another approach for partitioning is demonstrated in [9,10]. There, vertices of a triangulated surface are clustered into groups belonging to a specific shape type using multiresolution.

Julius et al. and Shatz et al. [11,12] show a tiling of a given model into nearly developable charts. This kind of chart tiling makes it possible to recreate the given surface as a paper craft model.

Levy et al. [13] use a region growing algorithm for creating patches whose boundaries run along sharp features. In a first step, some surface features are detected. Then a set of regions which meet at these features is constructed.

There are many approaches to computing a feature layout using Morse theory. In [14], an eigenvector of the Laplacian is computed and used as the Morse function. The Morse complex which is then built from this function segments the surface into quads. In [15,16],

* Corresponding author. Tel.: +49 3083875874.
E-mail address: cschulz@mi.fu-berlin.de (C. Schulz).

0010-4485/$ – see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cad.2009.11.002

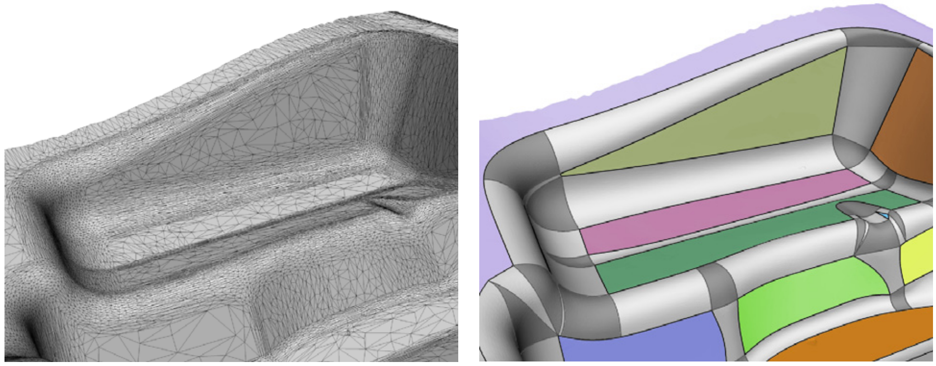

Fig. 1. Left: Typical CAD part as a triangle mesh. Right: Patch layout of the CAD part.

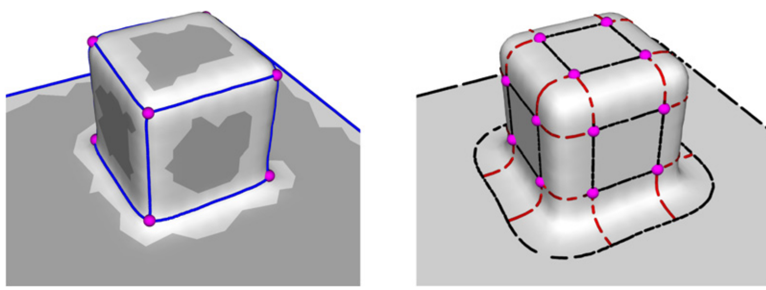

Fig. 2. Left: Feature graph on a CAD part consisting of faces, feature edges and node points. The dark grey parts within each face denote the plane-like or weakly curved parts whereas in the light grey part the surface starts to get curved. Right: Patch layout with face patches, fillets and node patches, as well as offset and node curves and offset nodes.

the construction of a Morse–Smale complex is described. By prescribing an adequate Morse function which represents the important parts of the surface, one can control the alignment of the feature layout. In [17], a curvature based Morse function is used to construct a Morse–Smale complex which aligns to the surface features. This approach is applied to CAD models in [18].

We also need to smooth patch boundary curves. In [19], the use of snakes for the generation of smooth curves on triangulated surfaces is proposed. This approach requires the repeated projection of the actual curve onto a two-dimensional domain. The curve smoothness is controlled via an energy term. Recasting the problem of smooth curves on triangulated manifolds to a high-dimensional optimization problem is described in [20]. Furthermore, the alignment of curves along features can be driven by the use of the feature sensitive metric introduced in [21]. Thickening of smooth curves is mentioned in [18], but without going into the actual details of the thickening procedure.

1.2. Contributions

The underlying structure of a given CAD surface is determined by its main building blocks, i.e., a set of characteristic CAD surface types. The methods mentioned above aim for such a decomposition into meaningful patches. The boundaries of these patches form an embedded graph on the surface. Assuming CAD models with round geometric feature edges, i.e., providing no clear defined boundary between primitives, the faces encircled by the graph's edges are not uniformly curved. Thus, they are not suited for low-order spline fitting. Regarding CAD surfaces having round geometric features, our contribution can be summarized as follows. We provide

- an algorithm to generate a net of curves, running along geometric surface features, such as valleys or ridges, called the feature graph;
- an energy formulation to align and smooth a curve within a feature region;
- a method to decompose a surface into its functional parts based on a given feature graph using an edge thickening – offsetting – procedure: the single parts are encircled by smooth boundaries aligned to nearby surface features.

We will show how to generate a consistent feature graph. Based on this graph, a patch layout is computed in a reliable and fast way. For the creation of both structures, no primitive fitting, i.e., template matching, is required. Starting with a triangle mesh, as shown in Fig. 1, left, we will end up with a decomposition like the one in Fig. 1, right.

Within our setting the feature graph resembles the embedded graph resulting from other methods mentioned above. Thus the feature graph needed for our patch layout generation could be replaced by any other graph structure describing a surface partitioning.

1.3. Organization of the paper

In Section 2, we explain our basic concepts and underlying notions of a feature graph and a patch layout. Section 3 deals with the generation of a feature graph. The creation of the patch layout from a given feature graph is explained in Section 4. All the steps described in Sections 3 and 4 are illustrated on the geometry shown in Fig. 3, left. Finally, the results of our tests are given in Section 5.