A method for reconstructing sketched polyhedral shapes with rounds and fillets

Pedro Company [1], Peter Ashley Clifford Varley [1]

[1] Department of Mechanical Engineering and Construction, Universitat Jaume I, Spain.[1]
{pcompany, varley}@emc.uji.es

Abstract. In this paper we present a method for detecting rounds and fillets in engineering sketches and drawings, and automatically generating a 3D model of the corresponding object, with rounds and fillets applied. This method is useful both as a component of computer-aided sketching tools and in determining design intent—although rounds and fillets are common in engineering parts, they often conceal design intent, which is more easily determined from the object's underlying polyhedral skeleton.

Keywords: Sketch-based modelling, design intent, features, rounds and fillets.

1 Introduction

Sketches are an important kind of graphics, as they assist product designers during the creative stages of design and help them to develop inventions. Computer-Aided Sketching (CAS) tools should provide their users with a sketching environment which allows them to make full use of their conceptual design and innovation talents, while also providing full integration with the subsequent phases of the design process [1], [2].

We seek to make our CAS tools "smarter". They should recognize *cues* (sometimes *clues* or *regularities*), common properties of sketches which reveal corresponding properties of the object. There are many such cues, and some of them have already been successfully studied [3]. Cues which convey design intent are particularly important. For example, Li et al [4], a recent contribution aimed at finding symmetry, uncovers important design elements embodied as high-level geometric relations.

In this paper we describe a novel method for solving the problem of automatically reconstructing polyhedral shapes which include *rounds* and *fillets*. Unlike previous methods which produce free-form surfaces from sketches (see [5] for a historical survey), our goal here is to detect and detach analytical surfaces which are geometrically simple but constitute common CAD/CAM features. Apart from the obvious advantage of reducing the workload of the designer (avoiding the step of obtaining the mind's eye image of the polyhedral skeleton), the method isolates features which play specific roles in designed parts (such features can be efficiently managed as independent features by current geometrical engines).

Ideally, reconstruction systems should automatically reconstruct shapes such as the one depicted in Figure 1 left. However, overtracing and other similar problems are still difficult to cope with [6]. Recognizing this, we aim for the more modest goal of interpreting tidied hand-drawn line-drawings (Figure 1 middle)—this goal is still a step beyond the current strategy of drawing the skeletal shape (Figure 1 right), constructing the object, and then manually adding blends to the resulting 3D model.
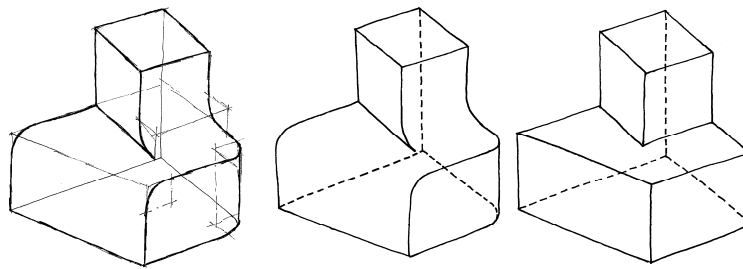
Fig. 1. Sketch of a shape with rounds and fillets (left), its tidied line-drawing (middle), and its polyhedral wireframe skeleton (right).

2 Our method

We shall initially develop our method for *wireframes* of *quasi-normalons* (wireframe sketches and line drawings show all edges of the portrayed object, visible or not, while natural sketches and line drawings show only the part of the portrayed object visible from the chosen viewpoint; quasi-normalons are those polyhedra where non-orthogonal edges may be removed without losing any vertex).

Our method will be divided into four main stages: 1) detect rounded edges and fillets (Section 2.1); 2) obtain the polyhedral skeleton (Section 2.2); 3) reconstruct the skeleton; 4) add rounded edges and fillets.

Stages 1 and 2 are specific to this work and it is these we describe in more detail. Stage 3, geometrical reconstruction, converts tidied line-drawings into 3D models [7]. This has already been partially solved in the context of sketch-based modelling— there is no general solution as yet, but we have developed an algorithm which can construct 3D models of quasi-normalons with as many as 50 faces [8]. Stage 4 is a common task in current CAD modelling environments, which encourage the users to create "skeletons" and then add rounds manually. This simplifies the creation of complex shapes. Current geometrical engines of CAD applications are quite efficient in managing rounds as separate features added on top of model trees. The theoretical problems of adding rounds and fillets while maintaining the design intent is solved, and current academic interest deals with more complex and subtle variations such as infinitely sharp and semi-sharp edges [9] and [10].

2.1 Detect rounded edges and fillets

We shall detect rounded edges and fillets in wireframe sketches by: 1) detecting circular arcs, and 2) forming pairs of circular arcs.

Detecting circular arcs (projected as elliptical arcs) may be easily done after segmenting the sketch strokes into simple lines, which is a solved problem for tidied hand-drawn line-drawings [11].

The task of forming pairs of arcs may be subdivided into the following actions:
- Pair those arcs which are contained in parallel faces and share a tangent contour line (such as the pair depicted in the left part of Figure 2, which share the vertical tangent line). By this means, faces can be detected in the skeleton [7].
- For the remaining arcs, pair those arcs which are contained in parallel faces, are similar in size and orientation, and are connected to mutually parallel lines (such as the pair indicated by the additional thin lines in the middle part of Figure 2, which do not share any tangent line).
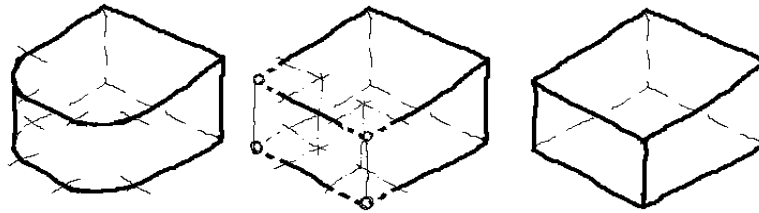
Fig. 2. Part with two rounds (left), arcs removed, edges extended and new vertices and edges added (middle), resulting polyhedral shape (right).

Rounds applied to single edges in quasi-normalon shapes fall in one of the three categories shown in Figure 3: a) both arcs are fully visible, and no one line connects them (since the edge has disappeared because of the rounding operation); b) one arc is fully visible, the other is partially occluded and one contour line is tangent to both, and c) one edge is fully visible, the other is fully occluded and no one line connects both.

Rounds in oblique edges of quasi-normalon shapes can be classified into the same three categories (Figure 3). It does not matter whether or not the rounded edge meets at 90° with the other edges connected to the same junction.
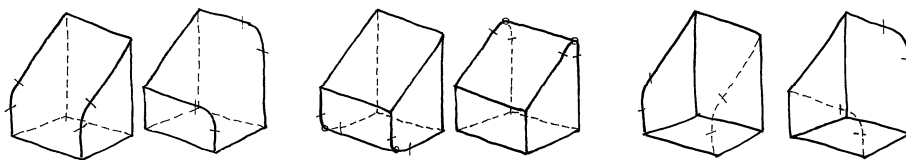
Fig. 3. Rounds of single edges in quasi-normalons: two arcs visible and no line (left), one and a half arc plus tangent line (centre), one arc (right).

Fillets in quasi-normalon shapes can only be classified into the second and third categories, since tangent contour lines may never appear (as fillets are concave shapes and may not belong to the contour of quasi-normalon shapes).

2.2 Obtain the polyhedral skeleton

We shall produce the polyhedral skeleton by repeating the following sequence for every pair of arcs: 1) suppress both arcs; 2) remove any contour line tangent to the suppressed pair of arcs; 3) extend the tangent lines to produce new pairs of vertices; 4) add new lines connecting the new pairs of vertices.

After suppressing the first arc, the lines connected to its ends must be extended until they intersect. The intersection point is the first new vertex, where the new edge must start. The same intersection procedure applied to the second arc produces the second new vertex, where the new edge must finish.

References

1. Company P., Contero M, Varley P.A.C. Aleixos N. and Naya F. Computer-Aided Sketching as a Tool to Promote Innovation in the New Product Development Process. Computers in Industry, 60 (8), pp. 592–603 (2009)
2. Olsen L., Samavati F.F., Sousa M.C., Jorge J. Sketch-based modeling: A survey. Computers & Graphics UK, 31 (1), pp. 85-103, (2009)
3. Yuan S., Tsui L.Y. and Jie S. Regularity selection for effective 3D object reconstruction from a single line drawing. Pattern Recognition Letters 29 (10), pp. 1486-1495 (2008)
4. Li M, Langbein F.C. and Martin R.R. Detecting design intent in approximate CAD models using symmetry. Computer-Aided Design, 42 (3), pp. 183-201 (2010).
5. Kara LB, Shimada K. Sketch-based 3D-shape creation for industrial styling design. IEEE Computer Graphics and Applications, 27 (1), pp. 60-71, (2007)
6. Company P. and Varley P.A.C. Operating modes in actual versus virtual paper-and-pencil design scenarios. 2009 Intelligent User Interfaces (IUI) Workshop on Sketch Recognition, Sanibel Island, Florida, 8th February (2009)
7. Company P., Piquer A., Contero M. and Naya F. A Survey on Geometrical Reconstruction as a Core Technology to Sketch-Based Modeling. Computers & Graphics, 29 (6), pp. 892-904 (2005)
8. Varley P.A.C., Company P. A new algorithm for finding faces in wireframes. Computer-Aided Design 42(4), 279-309, April 2010.
9. Hoppe H, DeRose T., Duchamp T., Halstead M, Jin H., McDonald J., Schweitzer J. and Stuetzle W. Piecewise smooth surface reconstruction. ACM Siggraph, pp. 295-302. (1994)
10 DeRose T., Kass M. and Truong T. Subdivision surfaces in character animation. ACM Siggraph, pp. 85-94. (1998)
11 Pu J. and Gur D. Automated Freehand Sketch Segmentation Using Radial Basis Functions. Computer-Aided Design. 41 (12), pp. 857-864, (2009)