SilF: A Sketching Tool for Cartoon-Like Pseudo-3D Illustrations Based on 3D Outlines

Kota Yonezawa*
Graduate School of Information Science and Engineering, Tokyo Institute of Technology
kota.yonezawa@is.titech.ac.jp

Shin Takahashi
Graduate School of Systems and Information Engineering, University of Tsukuba
shin.takahashi@acm.org

Etsuya Shibayama
Graduate School of Information Science and Engineering, Tokyo Institute of Technology
etsuya@is.titech.ac.jp

1 Introduction

There have been several papers on pseudo-3D representations that can represent 3D scenes without using full 3D geometry, such as polygons ([Cohen *et al*. 2000], [Bourguignon *et al*. 2001]). This paper presents a new pseudo-3D representation and its rendering algorithm for producing cartoon-like illustrations, which tolerates changes in viewpoint/viewing angle. Cartoon-like illustrations emphasize outlines, which are drawn in thick, dark lines. In our representation, the target 3D objects are represented as outlines only, not as full 3D geometries. These outlines of the illustration are located in 3D space, not on a 2D plane. We call this 3D curve the *3D outline*. Our 3D outline representation is intermediate between typical 3D representations and 2D illustrations. Although this model contains limited 3D information, we can view and rotate the illustration as if there were full 3D geometry. (Figure 1)

We have implemented a prototype system, SilF, for viewing and editing 3D outlines with a 2D-like sketching interface using standard 2D input devices.

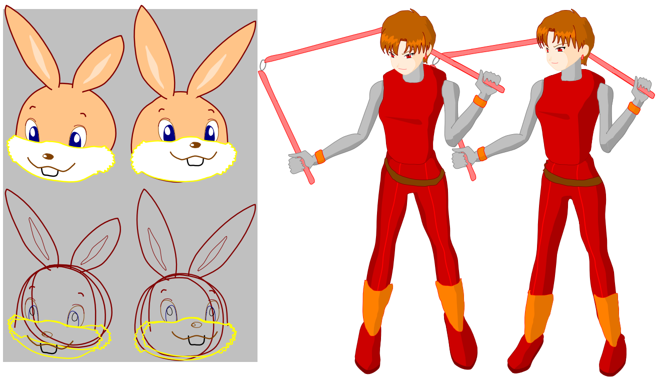

Figure 1: Sample models

2 3D Outline

Although our model does not contain surface information, a small number of 3D outlines can represent a 3D object. For example, Figure 2 shows a simple sample of a face. The outline of the face is modeled with a single 3D outline. The chin of the 3D outline is bent toward the front of the face. (Figure 2: left) This shape looks natural, even from various viewing angles, because the projected 2D shape of the 3D outline approximates the silhouette of the real 3D geometry. In addition, the parts on the face (eyes, mouth, nose...), which also consist of 3D outlines, are important for representing the 3D shape. They give the impression of a surface with the parts on it.

The rendering process consists of two steps. First, the 3D outline, which is implemented as a piecewise cubic 3D Bézier curve, is transformed into a 2D curve on the viewplane using a parallel projective transformation. This step is a simple process because the system needs to compute only the projected 2D coordinates of the anchor and the control points of the 3D outline. Then, the system draws the 2D curve and fills the area inside it using standard 2D Bézier curve drawing functions.

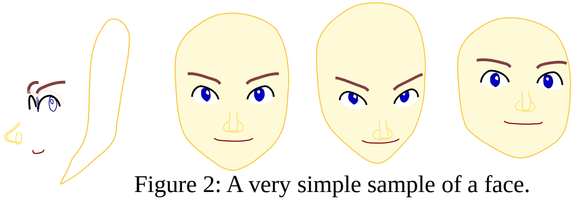

Figure 2: A very simple sample of a face.

In some cases, a single 3D curve is insufficient for representing the target 3D object. For example, if we look at a face squarely, the silhouette of the face is the line of the chin, but if we look at the face diagonally from the front, the cheek line appears.

To achieve such transformation of the silhouette in our system, we provide *composite 3D outlines*. A composite 3D outline is a group of basic 3D outlines that are rendered in a special way. The areas surrounded by 2D curves projected and constructed from 3D outlines are merged, and only the outermost part of the outlines is rendered. (Figure 3)

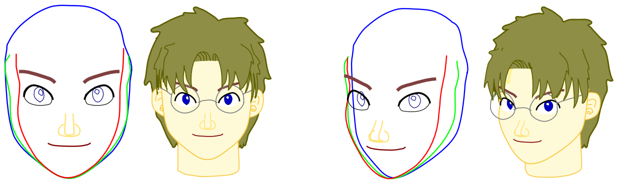

Figure 3: A sample of a composite 3D outline. If viewed from in front of the face, the blue outline is rendered. (Left) On changing the viewpoint, the red and green outlines are partially rendered. (Right)

Our 3D outline model can be viewed correctly from a limited range of viewing angles. In our samples, this is about 20 degrees from the frontal viewpoint, but this depends on the model. We will consider this problem in future work.

3 Sketching Interface

The user edits 3D outlines in creation and modification steps. In the creation step, the user first creates a flat 3D outline by drawing a stroke. If the user starts drawing from a point on the existing 3D outline, or stops drawing at a point on the existing 3D outline, the system creates a new 3D outline that contains the existing point in 3D space. If the start and end points of the stroke are both on existing 3D outlines, the new 3D outline will contain both points in 3D space.

To modify an existing 3D outline, the user first selects it, and then adds a new stroke. If the user changes the viewing angle, the system deforms the 3D outline, but only in the "depth" direction, so the shape viewed from the "front" doesn't change. In order to determine the "front" of each 3D outline, the system stores viewing angle information on 3D outlines that are drawn first.

References

DAVID BOURGUIGNON *et al*. Drawing for illustration and annotation in 3D. *EUROGRAPHICS 2001 Conference Proceedings*, pages 114-122, 2001.

J.M. COHEN *et al*. Harold: A world made of drawings. *NPAR 2000 Conference Proceedings*, pages 83-90, 2000.

* Current position: Sony Corporation