Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2006)
M.-P. Cani, J. O'Brien (Editors)

Sketch-Based Facial Animation

G. Nataneli and P. Faloutsos

University of California Los Angeles

Abstract

We present a sketch-based interface for animating virtual face models. Our approach first converts an input sketch into a set of strokes (curves). Each stroke is automatically associated with a facial element, such as eyes and mouth, and then compared against a set of template curves. The degree of similarity between input strokes and a template element defines the weights for a blendshape-based facial model. Shape and curve matching is performed using a combination of distance functions and similarity metrics, such as the Fréchet distance, modified Hausdorff distance and a simple Euclidean scheme.

Our approach can be used for the fast-prototyping of facial expressions in production as well as educational applications.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation; I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction Techniques.

1. Introduction

Sketching is a powerful tool used by animators and graphic artists to concretely capture their visions. Computer tools, on the other hand, are often at odds with such artistic vision and require artists to manually tweak numerous attributes in order to achieve desired results. Facial animation exposes this limitation more than other applications. Typically a large number of parameters is needed to provide users with enough flexibility to produce convincing results. We propose a method that simplifies facial animation by allowing users to control facial expressions directly by means of sketches.

2. Previous Work

Sketch-based interfaces are studied extensively in the literature and find applications in a variety of contexts within the realm of computer graphics. The pioneering work by SutherLand [Sut88] established the field by building an electromechanical device for CAD applications. Sutherland introduced a scheme for constraint satisfaction that inspired our development of classifiers. Igarashi [IMT99] developed a sketch-based interface for modeling rotund objects. Among other things, he suggests interesting solutions for analyzing and interpreting freeform drawings. More recently, Van De Panne [TBvdP04] introduced an interface for specifying character animation with cursive strokes. Li also proposed an alternative application for stylizing existing motions. However, the paper that more closely relates to our work [YSvdP05] is an interfaces for modeling parameterized objects.

3. Approach

The user first needs to draw the actual sketch. The only requirement is that the sketch includes the stroke that represents the head. This is needed for the system to analyze the proportions of the face in a consistent manner. When the drawing is committed, few image processing operations are used to segment the sketch and isolate individual strokes. A stroke for us is formally defined as being any element of the drawing that can be fully described with a single closed polygon. For any given stroke we compute several descriptors, such as its position with respect to the head, its size, and its polygonal contour. This process is meant to provide a consistent representation of any given sketch regardless of how the sketch was created or imported in the program. Our problem space is hence fully defined by a finite set of strokes with their associated properties.

The analysis of the sketch is performed in two steps.

Figure 1: *Sketches and the corresponding 3D Models (shown in pairs).*

Firstly, the overall number of strokes to be considered is reduced and only relevant strokes are retained. We call this the classification stage. Secondly, the shape of the strokes that are classified successfully is compared against a given set of templates. A user-defined mapping exists between templates and animation blend shapes; therefore if a good matching is found between a stroke and a template, the corresponding blend shape is selected for the final animation. Simple composition of the blend shapes produces a facial expression that resembles the input sketch. The face can then be animated by varying the corresponding weights on the blend shapes. One important consideration is that the animator does not need to understand or tweak these weights explicitly, which is generally a rather tedious task. Instead he or she can focus on drawing the intermediate facial expressions that define the animation.

The classification stage improves our approach in many ways. The user can draw as many or as few strokes as desired and the program will still be able to produce pleasing results within reasonable bounds. This is due to the fact that classifiers can silently discard strokes that are not considered useful for the analysis. Classification also makes the analysis more robust and improves performance, since fewer shapes need to be matched later in the pipeline. We devised a novel approach for the classification. We defined a set of simple functions called evaluators than quantify some specific property of one or more strokes, such as position size, aspect ratio, or symmetry. Every evaluator returns a score from zero to one that quantifies how well a stroke satisfies a given criterion. For every evaluator a threshold is defined; if a stroke produces a score below the threshold, it is simply rejected for that criterion. A classifier is an object defined by a set of evaluators and corresponding thresholds. Classifiers are defined for all the strokes that we need to consider for further processing (e.g. eyes, eye brows, mouth, etc.). The classification is then performed by evaluating every subset of strokes in the sketch with the evaluators. Every classifier will choose the stroke that produces the best scores for its corresponding evaluators. Of course, ambiguities may arise in the classification of strokes; in which cases further criteria can be used to refine the analysis and resolve the ambiguity. As an example, if we are able to uniquely identify the stroke corresponding to the nose, we can resolve the ambiguity between eyes and eyebrows by specifying that the eyes lie directly above the nose. The strength of this method is that it mirrors how humans evaluate the characteristics of a drawing – proportions, shape, and balance. As a result, the classification tends to produce answers that are visually sound. Notice also that the use of thresholds also enables the classifiers to rejects strokes that do not satisfy the desired criteria. We employ a combination of widely accepted and robust metrics to match strokes with template curves, such as the Hausdorff and the Frechet distance. Simple heuristics are used to improve the matching and select which metric to use among the ones that are available.

References

[IMT99] IGARASHI T., MATSUOKA S., TANAKA H.: Teddy: a sketching interface for 3d freeform design. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1999), ACM Press/Addison-Wesley Publishing Co., pp. 409–416.

[Sut88] SUTHERLAND I. E.: Sketchpad a man-machine graphical communication system. In *25 years of DAC: Papers on Twenty-five years of electronic design automation* (New York, NY, USA, 1988), ACM Press, pp. 507–524.

[TBvdP04] THORNE M., BURKE D., VAN DE PANNE M.: Motion doodles: an interface for sketching character motion. *ACM Trans. Graph. 23*, 3 (2004), 424–431.

[YSvdP05] YANG C., SHARON D., VAN DE PANNE M.: Sketch-based modeling of parameterized objects. In *Eurographics Workshop on Sketch-Based Interfaces and Modeling* (2005), pp. 63–72.