Facial Animation by Synthesis of Captured and Artificial Data

Zsófia Ruttkay, Paul ten Hagen, Han Noot, and Mark Savenije

CWI, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands
{zsofi, paulh, han, markjoe}@cwi.nl

1 Introduction

Performer-driven animation has been used with success [1], first of all to reproduce human body motion. While there are different capturing hardware-software systems to map the motion of a performer on the motion of a model of the body or face, little has been done both on the technical and on the theoretical level to support the inventive re-use of captured data.

The topic of this paper is an animation editing environment, which can be used to visualize and analyze captured animation data as well as to process the captured data and re-use it several times in different ways.

The Animation Editor has been designed in the framework of the FASE project, aiming at performer-driven facial animation. The performer data is used to drive both a physically-based 3D realistic model, as well as different, cartoon-like 2D faces. We are also experimenting with driving non-human faces with (processed) performer data. The editor is coupled to the models in such a way, that snapshots of the animation or selected parts of it can be seen interwoven with editing.

We will discuss animation editing in the context of facial animation. However, most of the ideas could be applied to other animation domains.

When animating the human body, techniques used to animate or describe joints the dynamics of motion of physical bodies can be applied. The situation is different in the case of facial animation. The 'laws' of facial movements are based on too complex and partly unknown physical structures (friction between muscles, precise location and parameters of muscles and tissue, etc.). In addition, there is still very little known about generic and individual characteristics of the dynamics of the face [2]. Hence, both the physical as well as the behavioral laws of facial motion could be, and should be learnt from a large body of empirical captured data. Once generic and individual characteristics of certain facial motions, e.g. expressions, are known, these can be used to process captured data or make facial animations from scratch.

2 Capturing Facial Data

The motion of the performer's face is coded in terms of feature point based MPEG action units (further on: AUs) [3]. In the present stage of the project

we can recognize the feature points identified in the MPEG standard by applying marker points to the performer's face. Later, more AUs will be captured, and based only on video recordings of the face without markers. The recognition based capturing subsystem is being developed by our partners in the FASE project [4]. We assume for this paper that captured data are provided through the above interface. The paper will not discuss the recognition task further but concentrate on the synthesis aspects, in particular by means of interactive animation editing.

3 The Animation Editor

Animation Editor is an interactive tool for the graphical specification, presentation and modification of the values of animation parameters for computer (facial) models. The parameters can represent muscle contraction values, coordinates of feature points or any parameters used to control a facial model.

The animation editor is as much a research tool as it is an animators tool. In body animation performer input is still superior to model-input for driving animations. In facial animation the situation is more complicated because accurate performer data are hard to get. Moreover the correct interpretation of observed data needs better underlying models than currently available. The animation editor will in our approach play an important role in finding better models. It can be used to do a detailed analysis of performer input parameter values including time dependent behavior. The analysis can be about individual parameters or ensembles of them. Hypotheses can immediately be 'edited in' and tested (visually) on a variety of models. Particular attention can be given to the possibilities of analyzing the dynamics of facial behavior.

Underlying utilities allow for controlled superposition of animation (e.g. speed and expressions). The strategy is to let the analysis tools smoothly evolve towards animators tools. The current version has been evaluated by animation artists through hands on exercising. The results are encouraging. During this experimental phase it is vital to maintain access to the lowest level editing operations, as they are the building blocks for higher level actions targeted for.

The animation editor operates on a window which looks like a musical score. There is a 'staff' for every animation parameter; the lines on each staff reflects the values the parameter can take. The behavior in time of an animation parameter is specified by placing points on its staff. They are entered, moved and deleted by mouse-operations.

One can perform editing operations — cut and paste operations, time- and value scaling — on portions of curves and on sets of them. Performer data can be read in and presented in the scores. Parts of earlier made animations — also performer data — can be copied and pasted to a different set of parameters, allowing e.g. to animate different — realistic 3D as well as 2D cartoon-like — faces on the bases of performer data.

The editor can produce as output animation scripts (movies) in ASCII format by sampling the curves at a rate which is set by the user. Such a file can be used as

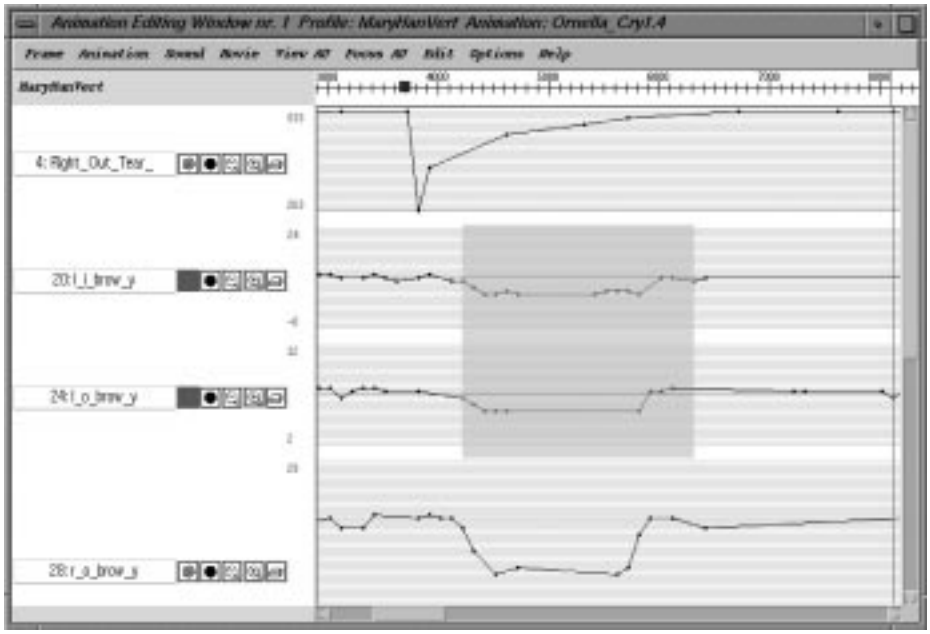

Fig. 1. Snapshot of an Animation Editor window, showing 4 animation parameter staves, the top with artificial data, the 3 below with captured data. Part of the left eyebrow movements are selected.

input for the corresponding model, or the corresponding model can communicate directly with the editor, allowing for feed back — snapshots, replay of selected parts — while editing the animation curves.

4 Processing Captured Data

The Animation Editor supports the (re-)usage of captured data in different ways:
Analysis The curves of the captured data give an overview of the 'content' of the facial movement. One can quickly identify certain events (jaw opening, eyebrow movements). Also, from the extreme values and shape of the curves one can evaluate the animation by large, at a glance.
Scaling Parts of the captured data can be scaled in magnitude and/or time. The selection can be restricted to certain parameters in a given time interval (e.g. to make an eyebrow movement more emphasized or shorter), or to one parameter (e.g. to achieve asymmetrical motion).
Editing and Extension of Captured Data The editing environment can be used to add non-captured motion (e.g. eye-gaze movement) and change the captured data locally (e.g. to eliminate recognition mistakes, to smoothen the animation, to refine synchronization).
Re-using Library Entries Pieces of captured (and possibly processed) anima-

tions can be saved and re-used later. When re-using an animation, different sets of motion parameter channels can be used than the ones of the saved animation. Automatic bi-linear re-scaling takes place when pasting a curve of one channel to another one.

Constraint-Based Editing Constraints can be used to express both the physical and behavioral 'laws' of the face to be animated (e.g. co-activation of action units, limits on ranges and derivate of parameters, symmetric motion of the face) to define compound animations as 'building blocks' to be used (e.g. a smile is an animation fulfilling certain constraints), to prescribe synchronization and other requirements according to the 'story' of the animation (eyes should be fixed somewhere, at the time of a noise effect the face should blink).

5 Implementation and future plans

The first version of the Animation Editor supporting the most essential constraints only has been implemented in Java, and is being tested by potential users. This version has been used with success in our environment. (Demos are available.)

For the next version, the constraint-related features will be added [5], allowing the user to define, switch on/off constraints, helping him to fulfill the current set of constraints during editing, adjusting an animation to a new set of constraints. The latter feature will allow to define transformations of captured data on a high conceptual level (e.g vigorous facial data can be transformed to an animation of the face of a tired/sleepy person).

Based on a large body of empirical captured data, we will support the inclusion and blending of animations as high-level building blocks (expressions). In the next version, for the mapping of channels a choice of possibilities will be available, allowing the definition of interesting, not one-to-one usage of performer data channels and animation parameters.

The work has been carried out in the framework of the ongoing FASE project, supported by STW grant CWI66.4088.

References

1. L. Williams, "Performance-driven facial animation," in *Computer Graphics (SIGGRAPH '90 Proceedings)* (F. Baskett, ed.), vol. 24, pp. 235–242, aug 1990.
2. I. Essa, S. Basu, T. Darrel, and A. Pentlands, "Modeling, tracking and interactive animation of faces and heads using input from video," in *Proceedings of Computer Animation '96*, pp. 68–79, 1996.
3. ISO/IEC 14496-2, "Coding of audio-visual objects: Visual," tech. rep., International Organisation for Standardisation, Tokyo, March 1998.
4. FASE home page: `http://www-it.et.tudelft.nl/FASE`.
5. Z. Ruttkay, P. ten Hagen, and H. Noot, "Constrained-based keyframing," in *Proceedings of the ECAI '98 workshop on Constraint technologies for artistic applications*, ECAI, August 1998.