Carnival: A Modular Framework for Automated Facial Animation

Michael Berger* Gregor Hofer† Hiroshi Shimodaira‡
University of Edinburgh

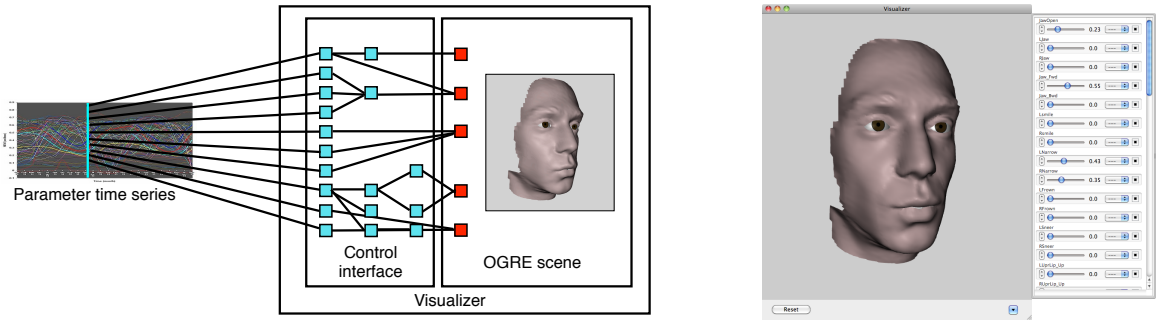

Figure 1: *(a) Schematic of a Visualizer, a real-time, modular animation component that is a key class in the Carnival system. The Visualizer consists of a control interface and an OGRE scene. The control interface comprises a set of deformation parameters (DPs) (blue squares), which may be bound to the current time point in a parameter time series, or to other DPs by linking functions. Ultimately, DPs link to mesh deformers (red squares) in the OGRE scene. (b) GUI window for a Visualizer, with manual slider controls for the deformation parameters.*

1 Introduction

Facial animation is difficult to do convincingly. The movements of the face are complex and subtle, and we are innately attuned to faces. It is particularly difficult and labor-intensive to accurately synchronize faces with speech. A technology-based solution to this problem is automated facial animation. There are various ways to automate facial animation, each of which drives a face from some input sequence. In *performance-driven animation*, the input sequence may be either facial motion capture or video of a face. In *automatic lip-syncing*, the input is audio (and possibly a text transcript), resulting in facial animation synchronized with that audio. In *audio-visual text-to-speech synthesis* (AVTTS), only text is input, and synchronous auditory and visual speech are synthesized.

The problem with such solutions is that they bring together software and data formats from different fields—in particular speech technology and graphics technology—that are not well integrated. The typical relationship between these areas is exemplified for automatic lip sync in Figure 2. First, speech motion parameters are generated from audio input; second, these must be adapted for use in animating a facial model using animation software. There is no single framework relating these processes. Thus:

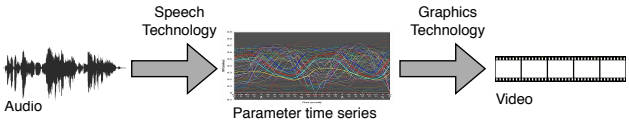

Figure 2: *Process flow for automatic lip sync.*

- The interface is laborious and slow. Parameter tracks must be converted into appropriate animation curves and imported into animation software, where they are usually rendered offline.
- The lack of live connection between the speech and rendering pipelines means it is difficult to backtrace animation problems to earlier processing stages, and difficult to see the effect of corrections to those stages in animation output.
- There is no standard control interface for different facial models, so the adaptation process must be repeated in each case.

2 Our solution

We introduce a software framework for automated facial animation, called "Carnival," which places speech and graphics components within a single object-oriented system. It provides fast and automatic end-to-end processing; real-time animation and linked display of speech representations and rendering for instantaneous feedback/feed-forward information; and standardized object interfaces for easy integration of new components.

The core of our solution is a platform-independent C++ API. One of the main features of the API is a class called Visualizer, which is a modular, real-time rendering engine. A Visualizer provides a uniform interface to speech parameter generation, in the form of deformation parameters (DPs), which have the range $[0, 1]$ for unidirectional or $[-1, 1]$ for bidirectional deformations, with rest state 0. DPs may be hierarchically related by link functions. Abstractly, a Visualizer is essentially an image decoder, converting a vector of deformation parameters into an image. Our first implementation of the Visualizer class is based on OGRE (Object-Oriented Graphics Rendering Engine) [Streeting 2010] (Figure 1), and can accommodate any facial model created in standard animation packages.

The Carnival framework also provides functionality for memory management; processing pipelines; a stack of "undoable" operations; and synchronous output of audio, video and animation.

References

STREETING, S., 2010. Ogre: Object-oriented graphics rendering engine. http://www.ogre3d.org.

*e-mail: m.a.berger@sms.ed.ac.uk
†e-mail: ghofer@inf.ed.ac.uk
‡e-mail: h.shimodaira@ed.ac.uk

Copyright is held by the author / owner(s).
SIGGRAPH 2010, Los Angeles, California, July 25 – 29, 2010.
ISBN 978-1-4503-0210-4/10/0007