Carnival: a modular framework for automated facial animation

Citation for published version:

Digital Object Identifier (DOI):
10.1145/1836845.1836851

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
ACM SIGGRAPH 2010 Posters

General rights
Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.
Facial animation is difficult to do convincingly, particularly when synchronizing with speech. There are various ways to automate facial animation:

- Performance-driven animation
- Audio-driven animation
- Audio-visual text-to-speech synthesis (AVTTS)

The problem with these solutions is they bring together software and data formats from different fields—particularly speech technology and graphics technology—that are not well integrated.

Conversion of facial dynamics into animation is cumbersome, slow and offline.

Lack of live connection between speech and rendering pipelines. Difficult to backtrace animation problems, or see outcome of edits in the speech processing level.

No standard control interface for different facial models, so adaptation process must be repeated in each case.

Software framework called "Carnival" which places speech and graphics components within a single object-oriented system.

- Fast and automatic end-to-end processing
- Real-time animation and linked display of time-varying representations for instantaneous feedback/feed-forward information
- Standardized object interfaces for easy integration of new components

The core of our solution is a platform independent C++ API.

The Component class hierarchy of the Carnival API. Components are objects that can be loaded, edited and saved. They include both data objects and the processing modules out of which systems are built.

The Component class hierarchy of the Carnival API. Components are objects that can be loaded, edited and saved. They include both data objects and the processing modules out of which systems are built.

Applications

- API may be used for fast prototyping of automated animation systems
- Suitable for performance-driven, audio-driven, or AVTTS applications
- Our implemented tool built on the API is suitable for in-house industrial or academic use