Emote, a new way of creating animated messages for web enabled devices

V. Helzle, S. Spielmann, N. Zweiling

Institute of Animation, Filmakademie Baden-Wuerttemberg, research@filmakademie.de

Abstract

Emote is a new way of creating superior animated messages for mobile devices. Using text and emotions an individual animated short message can be created and shared with friends worldwide.
The project evaluates the Filmakademie Application Frameworks backend capabilities for Cloud Computing, by utilizing state of the art real-time graphics and a dynamic animation logic.

Keywords: Messaging Service, Cloud Computing, Realtime Rendering, Animation, Virtual Characters.

1 Introduction

Emote[1] is a web based messaging service. Instead of delivering plain text messages, it synthesizes the information using text-to-speech technology. Combined with believable facial animation of a virtual character, this system offers a more immersive internet communication. Our service helps to overcome the exclusion of emotion in modern electronic messaging applications while fulfilling two current web communication needs: anonymity and personalized avatars. Emote enables anyone to create facial animations on the fly, without requiring skills in specialized animation software. The messages are generated faster than real-time using modern graphics hardware on a dedicated server platform.
Our application is suited for any web-enabled device. The workflow is very simple, create your message on our site and send the link to the video via Email or Facebook.
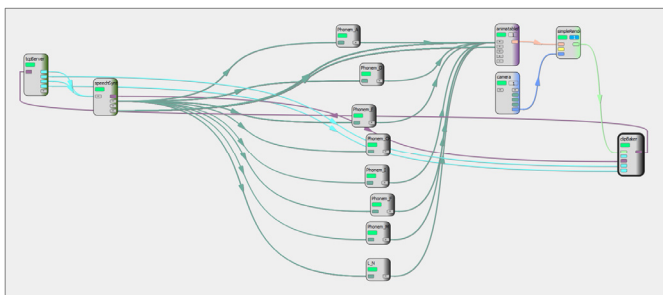

Figure 1: The node graph and processing pipeline in Frapper.

2 Processing Pipeline

Viseme information is extracted from the text and used to generate lip-synch animation for a custom character. Additionally, the user can trigger basic emotions and short animation clips. The user then receives a short clip which contains the combined audio/video mix. The basis for the processing pipeline is our in-house application framework Frapper[2]. Frapper provides a node-based scene model with plug-ins for node types and 3D rendering and animation capabilities. The SVOX text to speech library[3] is employed to synthesize speech and extract necessary viseme information. The web front-end is realized using HTML and PHP.

3 Processing Nodes

The ***TcpServer node*** is the communication port to the web server. When the user issues the generation of a message, the text, the message ID and the character's ID are sent to this node using a TCP/IP transfer. To allow the use of different characters and/or application instances, it is possible to use multiple ports.
The ***EmoSpeechSynth node*** uses the SVOX technology to convert written text into a wave file and generates animation curve from phoneme information.
The ***Phoneme nodes*** are simple controller nodes, used to define limits for the animation triggering.
The ***AnimatableMesh node*** represents the character including its geometry and basic materials. The node also contains all animation clips provided with the character.
The ***SimpleRender node*** represents a render target. Based on the camera and mesh information this node renders the scene into the final image.
The ***ClipBaker node*** generates an image sequence, which is then converted into MPEG-4 (mp4) and Flash Video (flv) movie clips using FFmpeg[4]. There is also an option to use an internal ftp client in order to copy the resulting clips.

4 Website

The webpage represents the front-end and the user interface to create messages on the fly, just by typing text and emoticons into a text box. The system generates synthetic speech from plain text. The Emoticons are divided into two categories, basic emotions and additional animations. Where the basic emotions define the emotional state throughout the message, the additional animation options consist of prepared clips, such as winks or smiles, which are blended into the animation. Thus, we provide the user with an easy-to-use and familiar tool with which to create complex animations in combination with synthetic speech. Furthermore it is not necessary to have any experience in animating or digital content creation.

---

[1] emote.animationsinstitut.de

[2] research.animationsinstitut.de/57.0.html
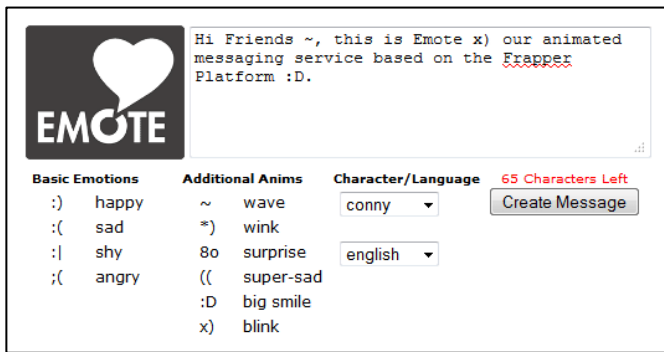
[3] www.svox.com

[4] www.ffmpeg.org

Figure 2. The user front-end on the Emote website.

5 Characters

Currently there are two characters available: a stylized cow and a somewhat realistic woman. Because of the degree of abstraction, the cow is animated using more classic techniques like squash and stretch, anticipation, etc. The shading is also kept to a relatively simple level.

The female character strives for a more realistic animation and shading style using modern real-time rendering techniques.

For creating the animation, we decided to use the Adaptable Facial Setup [3] as it is particularly suitable to animating realistic characters within a non-linear system. Another benefit of the system is the decomposition and structuring of complex animation into small clips. This is necessary to enable instant, dynamic animation in a real-time environment. So, for example, pre animated emotion- or phoneme clips are stored and triggered in real-time later on.
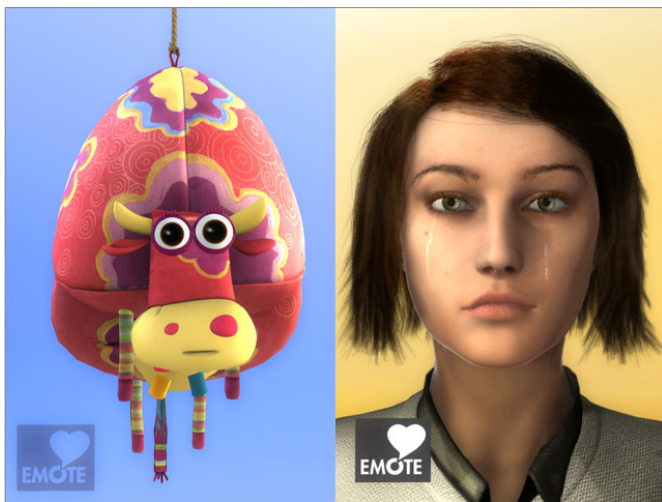

Figure 3: The both real-time characters

Also the shading is on a more complex level. Our priorities here are especially the simulation of the eyes, the hair and the skin. The eyes are realized with a modified ray-casting test [2] using pre-calculated lookup tables to reduce calculation costs. For the Hair we use the Light Scattering for Hair Fibers approach [4]. A BRDF shading model [1] is employed to calculate all other visual surface characteristics. Additionally, in a post step in the rendering pipeline, the skin is blurred in screen space depending on the wavelength (color) of the current pixel. This method simulates light scattering within the lower skin layers. Moreover we have implemented a technique to create facial wrinkles [5], depending on the deformation of the face geometry and a tear-drop simulation

which is controlled by the character's emotional state. We have also taken care to present artist-friendly parameters in our material definitions. More complex parameter sets defining attributes such as skin and hair colour or eye redness are summarized within artist-logical parameters.

6 Future Work

In order to speed up the render-to-clip performance of our system, we created a Cuda[5] based version of the clip baker node, called CudaClipBaker. This node incorporates the Cuda's GPGPU capabilities to convert the image sequence to an h.264 conform video clip.

This implementation works, but the image quality is not as satisfying as the quality delivered by the ClipBaker without the use of Cuda. As Cuda's video encoder library only supports compressed frame formats, we need to convert the RGB files from the 8bit RGB render targets into the NV12[6] format. This conversion is done using a Cuda kernel. The result clips look somewhat aliased. Edges don't look smooth, possibly due to the small sampling kernel.

Furthermore the gained speed-up isn't satisfying. This might increase with a larger render target.

7 Render Times

System: HP Z800, Intel Core i7 Xeon® X5570 @ 2.93 GHz, NVIDIA GeForce GTX 580, 16 GB RAM, 1 Gb LAN

Generic Message containing 130 characters and 2 additional animations with a length of 16.3 seconds by 30 frames per second and a file size of 1.55 MB.

In seconds	Cow	Woman
Redering	6.7	11.8
Compression	1.7	1.7
Upload	1.1	1.1
Sum	9.5	14.6

References

[1] M. Ashikhmin, P. Shirley An anisotropic Phong BRDF model. Journal of Graphics Tools 2000;5(2):25-32.

[2] G. Francois., P. Gautron, G. Breton, K. Bouatouch Anatomically Accurate Modeling and Rendering of the Human Eye, SIGGRAPH 2007 Sketches

[3] V. Helzle, C. Biehn, T. Schloemer, F. Linner Adaptable Setup For Performance Driven Facial Animation, Animationsinstitut Filmakademie Baden-Wuerttemberg 2004

[4] S. Marschner, H. Jensen, M. Cammarano, S. Worley, P. Hanrahan Light scattering from human hair fibers, ACM Transactions on Graphics, vol. 22, no. 3, pp. 780–791, 2003, SIGGRAPH 2003.

[5] C. Oat, AMD Inc. Real-Time Winkles, SIGGRAPH Presentation 2007

---

[5] Parallel computing architecture for GPU's by Nvidia

[6] YUV 4:2:0 image with 8 bit Y samples followed by an interleaved U/V plane containing 8 bit 2x2 subsampled colour difference samples