Playable Universal Capture

George Borshukov, Jefferson Montgomery, Witek Werner, Barry Ruff, James Lau, Paul Thuriot, Patrick Mooney,
Stefan Van Niekerk, Dave Raposo, Jean-Luc Duprat, John Hable, Håkan Kihlström, Daniel Roizman, Kevin Noone, Jeff O'Connell
Electronic Arts, Inc.

Introduction

There are a vast number of factors involved in reproducing the subtleties of realistic facial animation. So many, in fact, that it is near impossible for an animator or a computer simulation to achieve genuinely realistic results. One reason for this is simply a lack of sufficient models for the important characteristics and subtle dynamics of facial expressions expected by human observers, who are all ultimate experts at watching faces! The *Uncanny Valley* has recently become the standard term for the unintentional creepy appearance of near-photoreal, computer generated, animated faces. This phenomenon, first predicted in the context of robotics, is being increasingly seen in human characters in video games. The work described here can be viewed as a direct attempt to build a bridge across the Uncanny Valley and start climbing up the other side for the most challenging applications of all for digital humans—those requiring real-time interactivity. Ultimately, we hope that our collection of techniques will help human characters in video games and other real-time applications to approach the believability and emotional impact offered by their film counterparts.

Motion Capture with Texture Acquisition

We have developed a more robust production level version of the Universal Capture (UCap) system introduced in [1]. To guarantee robustness we deploy a state-of-the-art facial motion capture system consisting of: 8 IR cameras plus 3 synchronized, high-definition, color cameras framed on the actor's face in an ambient lighting setup. Around 70 small retroreflective markers are placed on the actor's face. The reconstructed 3d motion path of the face markers is then used as the direct source of deformation for the static scanned and surfaced head geometry. We achieve the deformation with a facial *bone* rig in Maya. The number of bones is equal to the number of markers on the face. This deformation approach achieves excellent results in accurately reconstructing the facial shape and motion in all areas of the face; except for the interior contour of the lips for obvious reasons of not being able to place markers there. We solve the lip contour problem with a small set of additional bones which control the lip interior and are quickly keyed by an animator using the background images as a reference. Once we have achieved successful reconstruction of the facial deformations we can reproject and blend in UV space the input image sequences from each camera for every frame to produce animated facial texture maps. The presence of facial markers is not a problem as our motion capture system needs retroreflective markers only 1.5 mm in diameter which can easily be keyed out and removed in the final textures.

Data Compression

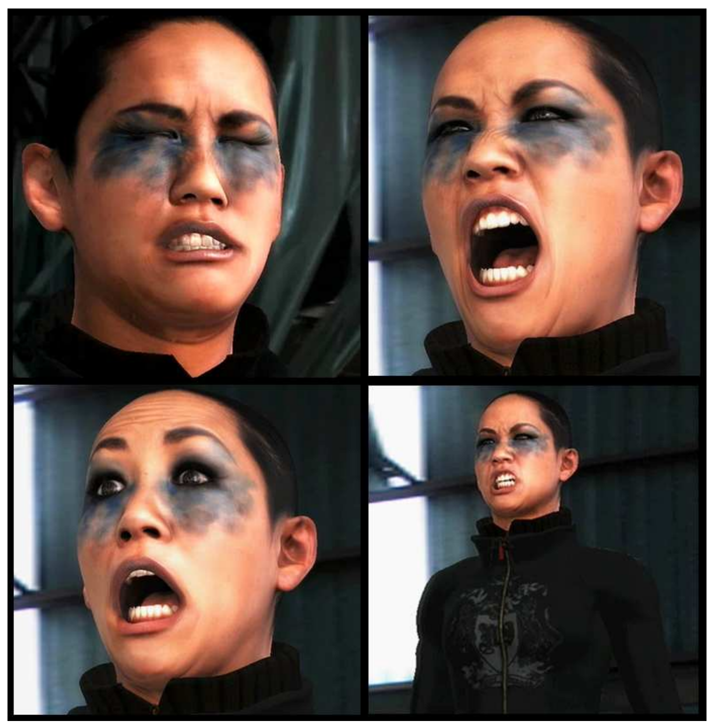
Our capture technique produces an enormous amount of data which prohibits recreation at interactive rates. We use a variant of PCA to compress both geometry and textures for each character. We analyze all data captured for an actor and store it using a common basis and a set of per frame weights for each captured clip. This compression technique offers significantly lower decompression and blending complexity than codecs such as MPEG that would be needed for the textures and bone SQT (transform) streams that would be required to reproduce the geometry using matrix palette skinning. PCA decompression is a simple dot product while blending during clip transitions can be accomplished by simply linearly interpolating the weights and then decompressing using the new weights.

Sequencing Performances

Our compression allows us to encode large libraries of expression clips and then sequence short constituent expressions with blending across the transitions. The fact that this technique preserves the captured subtlety and variety allows us to apply *motion graphs* to facial animation. Motions graphs (a.k.a *move trees*) are common practice in video game development and have been applied successfully to produce rich and realistic body movement from both captured and hand-animated clips.. To apply the motion graph approach to facial animation a library of facial expression clips, fundamental to a certain context, are captured; a motion graph is designed to connect the expressions, and the graph can be traversed, rendering the clips and interpolating both textures and geometry across the transitions. Performances can be triggered either by an AI sub-system or directly by the user through a game controller.

Real-time Interactive Results

We show use of this approach in a real-time interactive application that delivers a compelling visual experience and allows interactive switching between different captured clips at any point by triggering and blending appropriate facial motions from a state graph, complete freedom of camera movement, control over the character's shading, lighting, and environment, stylization either by capturing made-up actors or by adjusting the shaders for additional artistic effect, plus integration with body animation. We consider our approach primarily a "next gen" game technique and therefore have implemented our decompression on the PC and Xbox 360 using the GPU, as well as, the Playstation 3 using one or more of the available Cell SPEs. The following images are all direct captures from a custom prototyping engine based on OpenGL ES and Cg implementing the features described above.



[1] BORSHUKOV, G., PIPONI, D., LARSEN, O., LEWIS, J. P., AND TEMPELAAR-LIETZ, C. 2003. Universal Capture: Image-based facial animation for "The Matrix Reloaded". In *Proceedings of ACM SIGGRAPH 2003 Sketches and Applications*