sketches_0105: Markerless Facial Motion Capture using Texture Extraction and Nonlinear Optimization

Eugene Vendrovsky Ivan Neulander
Rhythm and Hues Studios

We present a markerless facial motion capture algorithm based on nonlinear optimization of a texture-based error metric that eliminates the need for precise camera calibration and provides flexible controls for tuning the optimization.

Implementation

The use of nonlinear optimization for estimating facial animation parameters is quite popular, as evidenced by [Paterson and Fitzgibbon 2003] and [Williams 2005]. However, these approaches include *precise* camera calibration and model tracking either as prerequisites or as part of the problem space while ours requires only approximations.

We use a polygonal facial model driven by a control rig consisting of 10-50 scalar parameters that deform the model by blending between rest poses or driving a muscle simulation. Any scalar whose variation continuously affects the appearance of any part of the face is a suitable rig parameter.

Our solver adjusts the rig parameters in an attempt to minimize a discrepancy metric between successive frames. We calculate this metric by constructing textures that capture the camera projection of the reference footage onto the deformed model at each frame, taking visibility into account. The textures are efficiently computed using a specialized tool called *Primitex*, which is integrated into our solver.

Primitex

Primitex, an acronym for "PRoject IMage Into TEXture", is a tool for efficiently projecting a "plate" or perspective camera's image of a model—in this case, live footage—into a texture space defined within the model. This technique requires a polygonal model with suitable (bijective) texture coordinates and an approximately calibrated camera in addition to the plate. We rasterize selected parts of the model into their native texture space, keeping track of interpolated positions and normals. To color each output pixel, we sample the plate based on the interpolated position and given camera data. Backface culling and ray tracing detect hidden surfaces, whose corresponding texture pixels are marked invalid.

Working in texture space makes our method highly tolerant of camera calibration errors: as long as the camera sees most of the same geometry in adjacent frames, the textures that Primitex outputs will have many pixels in common, provided that the geometry was deformed to match the plates at both frames, and the lighting conditions were consistent. Our texture-based error metric enables the user to easily mask out specific materials or texture regions from consideration (e.g. use a marquee to select a part of the model in texture space). This expedites the solution and improves accuracy by eliminating the residual noise resulting from the solver's attempt to adjust rig parameters that the user knows *a priori* to be irrelevant to a given area of interest on the model.

Nonlinear Optimization and Pyramid Solver

Our goal at each frame is to deform the facial geometry by setting the facial rig parameters in such a way as to generate a Primitex texture that differs minimally from the previous frame's texture. We use the Levenberg-Marquardt nonlinear optimization algorithm to solve the parameters, with an error metric computed from a pair of textures, as described below.

To address changes in lighting from frame to frame, we image-process the Primitex textures by uniformly subracting the average color over all valid pixels ("centering"), and normalizing by the standard deviation over both frames. Our error metric is analogous to Normalized Cross Correlation: the sum of squares of pixel-to-pixel differences between the two preprocessed textures, over the domain of all pixels that were marked valid in both textures.

A common challenge with nontrivial rigs lies in the multi-modal nature of the search space. Global optimum search algorithms, like simulated annealing, are robust but prohibitively expensive. Local optimum optimization, while fast, tends to get stuck in minima and stay there. We opt for a hybrid "Pyramid Solver" approach, where Primitex textures are initially constructed at low resolution (starting at 256^2) in order to yield a coarse but stable solution. Then we refine this solution by using progressively higher resolutions, up to 2048^2. This produces smooth local minima at low resolutions, which tends to determine the rig parameters that affect large-scale motions. Parameters responsible for finer motions are naturally resolved by the solver when it considers the higher-resolution textures.

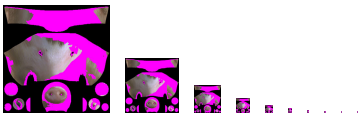

Figure 1: A sample pyramid of Primitex textures for a single frame, with invalid pixels (areas hidden from camera) rendered pink.

Results and Future Work

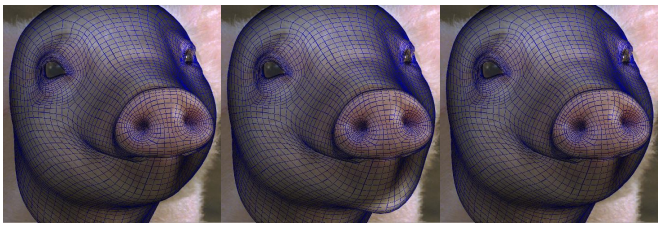

Figure 2: In this example, we used a manually computed rig from frame A (left) and a deliberately incorrect starting rig from frame B (middle) to compute the rig for frame B (right) with our solver.

Our planned future work includes: 1) automatic detection of the texture areas that are relevant to specific rig parameters; 2) adding a regularization term to our error metric in order to penalize unrealistic motion; 3) adjusting the solver to better handle deformations that significantly reduce the number of valid pixels in the Primitex textures, which can yield suboptimal solutions.

References

PATERSON, J., AND FITZGIBBON, A. 2003. 3d head tracking using non-linear optimization. Tech. rep., Oxford University.

WILLIAMS, L. 2005. Case study: The gemini man. In *SIGGRAPH 2005 Course 9: Digital Face Cloning*.