

Blend Shapes for Facial Animation

Vanna Bushong

University of California, Los Angeles

vbushong@ucla.edu

Abstract

This project presents a method for facial animation using blend shapes. The program accepts a starting and ending mesh and uses either linear or Hermite interpolation to create an animation blending the two. The goal in creating this blend shape tool was to build a piece of a larger project which generates models of the human head based on two reference images - a front and side view. The final program will be able to build models using images of people making various facial expressions, and use them as blend shapes for animation. In order for this to be an easy-to-use tool for artists, the program is being developed as a plugin for Autodesk Maya software, the industry standard for modeling and animation.

1 Introduction

Facial animation is a challenging task, especially for human characters. Since our eyes have been trained by watching other humans' faces all our lives, it is easy to spot abnormalities in a human character's facial expressions. One technique for facial animation is the use of blend shapes. The process involves creating a mesh for two different facial expressions and interpolating between the vertices to blend them.

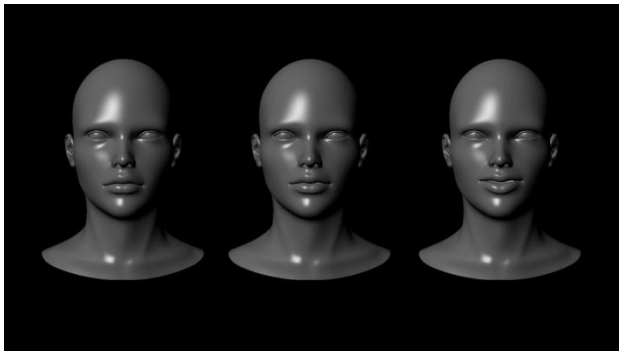


Figure 1: Still frames from a sequence of animation showing a base mesh, target mesh, and one interpolated frame in between.

The blend shape tool developed for this project utilizes two different interpolation methods - linear and Hermite. The starting and ending poses are specified in the form of two OBJ files, and the number of frames to use in the animation is set by the user in Maya. When the program runs, every vertex in the starting pose is interpolated between the corresponding vertex in the ending pose at each frame, and the resulting vertex positions are saved in Maya for playback.

This project was created to be a piece of a larger project which is still in development - a modeling program which uses reference

images taken from both the front and side of a person's head to automatically generate a model of the head. To accomplish this, facial feature landmarks are detected in two given images and the user is allowed to make adjustments if desired. The coordinates of the feature points are then used to adjust the vertices of a generic head. Since the head is created in Maya, the modeler can easily make any touch-up adjustments and begin using the model immediately.

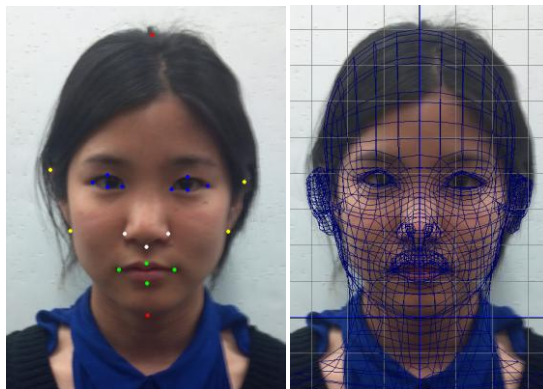


Figure 2: The image on the left shows facial feature landmark detection on a front view. On the right, the landmark coordinates have been used to generate a mesh.

Eventually the blend shape tool will be an additional feature of this project, allowing animators to input images of people making various facial expressions, generate models from the images, and create animations by blending them. Since the head modeling program has not been perfected, a generic head was used in the testing of the blend shape program.

The rest of this paper focuses on the approach used in creating blend shape animations and the results of the program. For a more detailed explanation of the Maya API and the functions used from it, refer to the Autodesk white paper [Autodesk 2007].

2 Importing Models

The first step in creating a blend shape animation was to load the meshes used for the starting and ending poses. For the purposes of testing this program, a generic head with a neutral expression was used for the starting pose. For the ending pose, six models were created to represent the six universal facial expressions determined by the findings of psychologist Paul Ekman: anger, disgust, fear, happiness, sadness, and surprise [Ekman 1971].

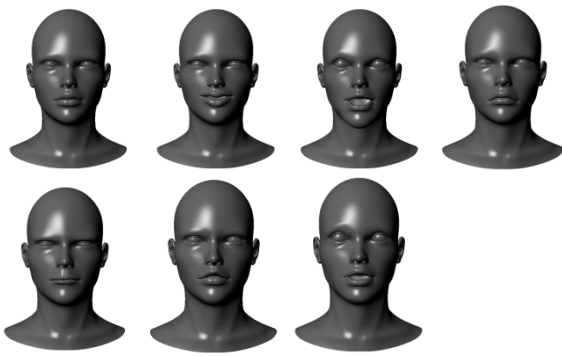


Figure 3: The models used to represent a neutral pose and the six universal facial expressions. From top left: neutral, happiness, surprise, sadness, anger, disgust, fear.

The models were saved as OBJ files and loaded into Maya with a call to Maya's import function. In the current implementation, only one target mesh may be set at a time, and the interpolation is calculated between the current state of the neutral pose and the target.

Using the Maya API, a mesh object was created that stores data for both poses including vertex coordinates, UVs, and color and normal values. Using the mesh object, a vertex iterator was created to easily iterate over the vertices of both poses and adjust the position of the current pose during interpolation.

3 Interpolation

In this project, the blend shape interpolation occurs over a number of frames specified by the user in Maya. For example, the traditional frame rate for film is 24 frames per second; so to create a one second animation, the user would set the number of frames in the Maya timeline equal to 24.

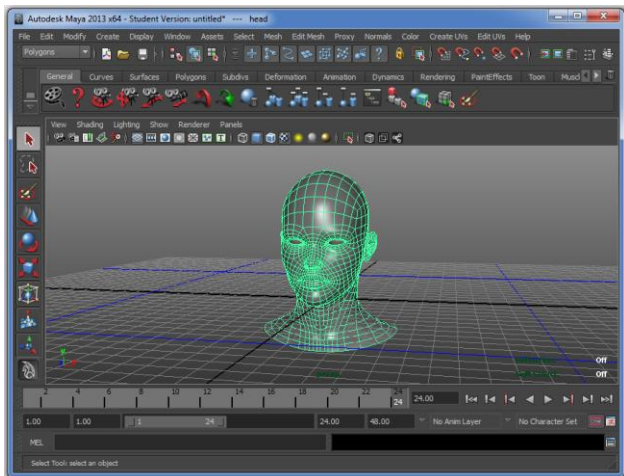


Figure 4: The Maya environment after running the blend shape plugin. The timeline at the bottom was set to a length of 24 frames, which was the number of frames used for interpolation.

Using the Maya API once again, the values of the first and last frame in the timeline were stored and used to calculate the time step for the animation as shown below:

$$timestep = 1/(lastFrame - firstFrame) \quad (1)$$

Next, for each frame in the timeline, the position of every vertex in the current base mesh was accessed and used in the interpolation relative to the target mesh. Two different methods for interpolation - linear and Hermite - were tested. The linear form is given by:

$$currentPoint.x = (1 - t) * currentPoint.x + t * endPoint.x$$

$$currentPoint.y = (1 - t) * currentPoint.y + t * endPoint.y$$

$$currentPoint.z = (1 - t) * currentPoint.z + t * endPoint.z \quad (2)$$

where t is incremented by the time step at every frame.

The Hermite form of interpolation uses a polynomial function. Two blending functions were calculated using:

$$f_1 = 2t^3 - 3t^2 + 1 \quad (3)$$

$$f_2 = -2t^3 + 3t^2 \quad (4)$$

where t is once again incremented by the time step at every frame. The results are then applied to the current and end points to set the x , y , and z values for each new position.

$$currentPoint.x = f_1 * currentPoint.x + f_2 * endPoint.x$$

$$currentPoint.y = f_1 * currentPoint.y + f_2 * endPoint.y$$

$$currentPoint.z = f_1 * currentPoint.z + f_2 * endPoint.z \quad (5)$$

The new position of each vertex was saved in Maya at every frame for playback.

Initially, I wanted the interpolation to take place in real time. This would make it possible for the user to move the target mesh around on the timeline and instantly test the animation at various lengths. But for this to happen, I needed to create a callback in the plugin for playing the animation using the API. I also needed a way to move only the vertices necessary in creating the animation, rather than looping through every single vertex, whether it needed to be moved or not. Due to time constraints on this project, I was not successful in making either of these happen. With more research and a better understanding of the Maya API, however, I am confident real time interpolation will be possible in the future.

4 Results

The resulting program was successful in morphing a base mesh into a target mesh over a specified number of frames. The interpolation was tested over 24, 12, and 6 frames. Between the linear and Hermite interpolation methods, there was no noticeable difference in the final animation. Both solutions produced smooth, believable facial movements. To draw a conclusion on the best interpolation method would require more testing of additional methods and models.

The program was also tested by running it over 12 frame sections of the Maya timeline with 12 frame gaps in between. Each time the program was run, a new target mesh was specified. This resulted in a mesh that seamlessly blended from one expression to the next.

5 Conclusion

Creating a blend shape tool for Maya was a successful step in the development of a larger project. In the future, the tool will be part of a program that can create and animate faces based on reference images. In the current implementation, the program interpolates vertices between two meshes to create a convincing facial animation, using either linear or Hermite interpolation.

While the results show promise for a useful tool, there is still much room for improvement. In the future, the interpolation

needs to occur in real time for greater flexibility in testing the length of the animation. Additionally, different types of interpolation methods could be added, along with an option for setting multiple target meshes to blend to and from. Overall, blend shapes have proved to be a good solution for animating facial expressions and will be an important part of the future tool for automated facial modeling and animation.

References

Autodesk Maya API White Paper. *Autodesk*. 2007. Autodesk, Inc. 18 June 2013 <http://images.autodesk.com/adsk/files/maya_api_whitepaper.pdf>.

EKMAN, P. AND FRIESEN, W. V. 1971. Constants across cultures in the face and emotion. In *Journal of Personality and Social Psychology*, Vol 17(2), 124-129.