Modeling Facial Expressions in 3D Avatars from 2D Images

Emma Sax

Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA

12 November, 2016
Morris, MN

The Big Idea

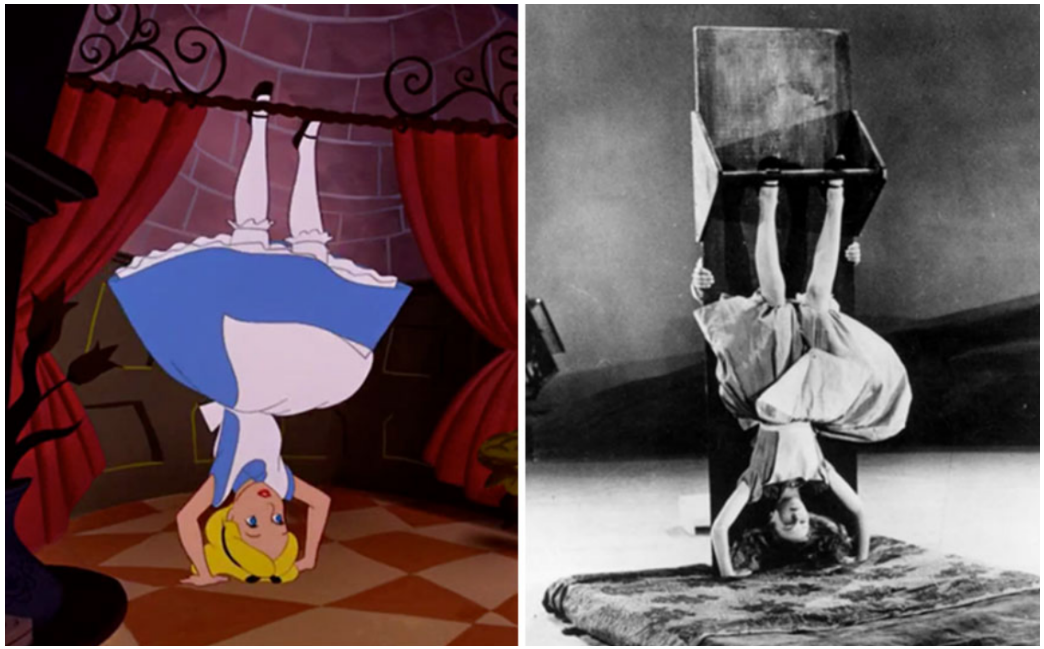
- Take a 2D image of a user
- Use specific facial features on the user as guidelines
- Render a 3D avatar with the same facial shape and expressions as the user

[2]

Outline

Overview

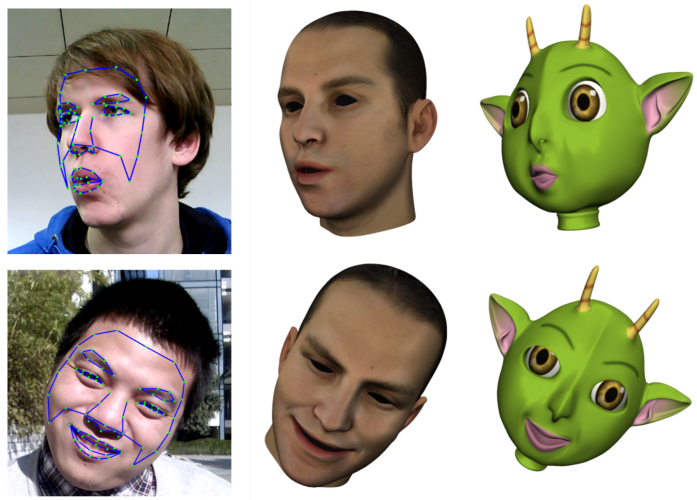
Traditionally, animators have found it is easier to model humans and their facial expressions when they use physical humans as models.



[1]

The Basic Process

- Take a single 2D video frame of a user
- Track specific landmarks on the user's face
- Render a 3D virtual image or avatar with the same facial shape and expression
- Entire process should occur in real time



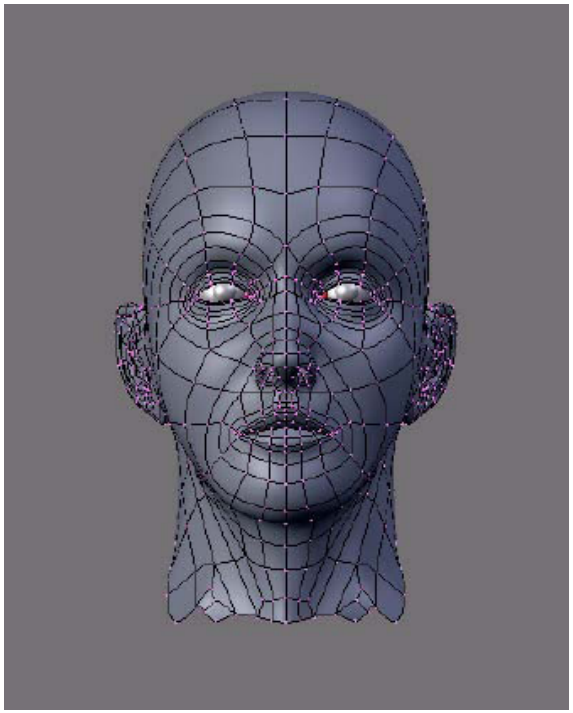
[3]

Ekman's Facial Action Coding System (FACS)

- Any facial expression can be represented by the contractions and relaxations of specific facial muscles (also known as *Action Units* or AUs)
- Categorizes human facial movements by the specific AUs that are used to create the facial expression
- Any anatomically possible facial expression can therefore be represented through AUs

Linear Blendshape Models

- A *linear blendshape model* generates a facial pose as a linear combination of a number of *blendshapes*
- *A blendshape* is a visual approximation of a facial expression in which a single collection of points (*mesh*) have deformed to a series of fixed vertex positions
- Each blendshape contains a *blend modifier* which contains an intensity slider that controls the amount of contraction and relaxation of the AUs



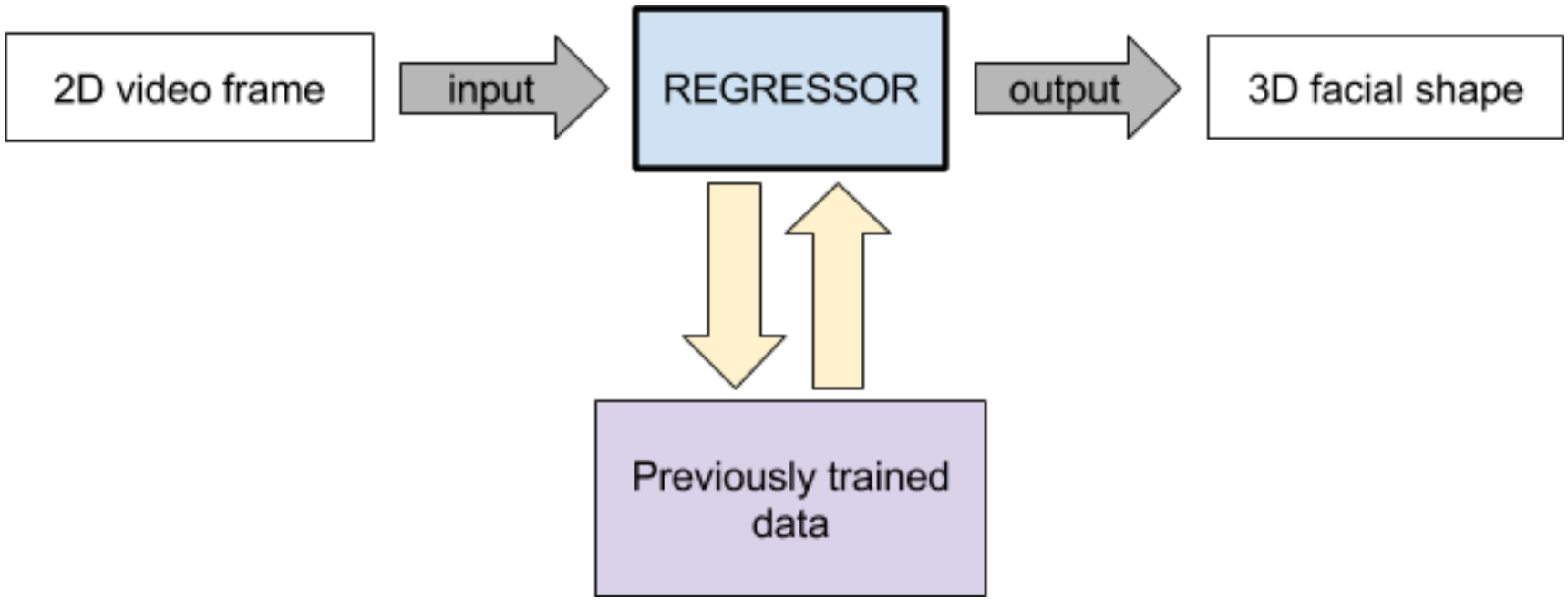
[4]

FaceWarehouse

- Public, online database of 3D facial expression models and blendshapes
- Composed of 150 individuals
- Tracked features on each individual's face using FACS, and composed generic blendshape models for each generic facial expression

What is a Regressor?

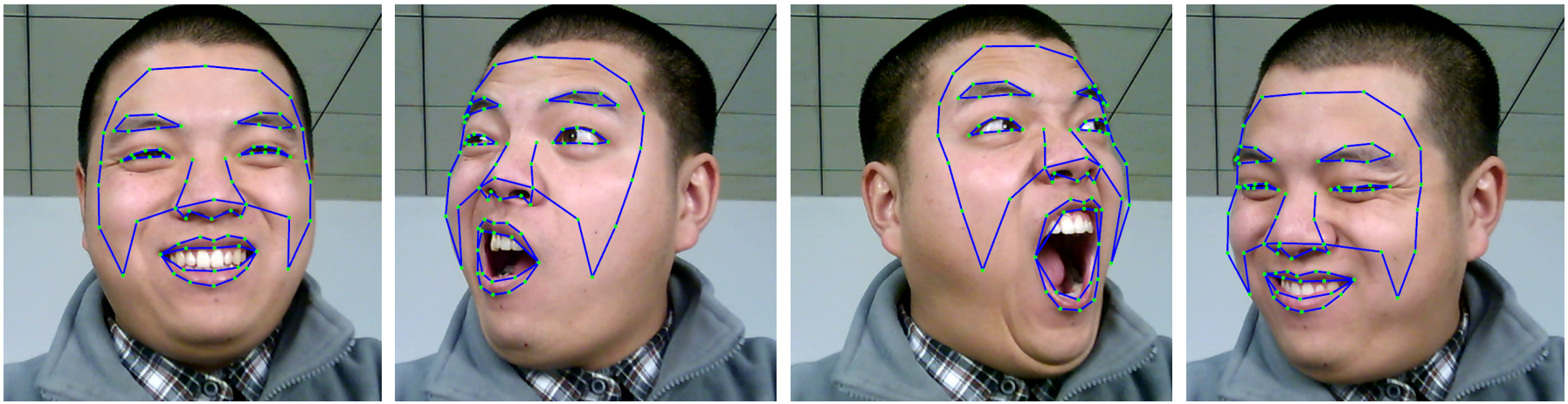
A *regressor* is an algorithm that is specifically trained to look at input states in order to predict the next output state. The regressor does this by analyzing the relationships between the input data and previously trained output data.



Outline

Gathering Data and Locating Facial Landmarks

- 60 images of the user showing pre-defined face positions are captured and categorized into two groups:
  - Various different head poses with neutral facial expression
  - Various different facial expressions
- A set of 75 facial landmarks are automatically located
  - 60 internal landmarks and 15 contour landmarks
  - All landmarks can be overwritten manually

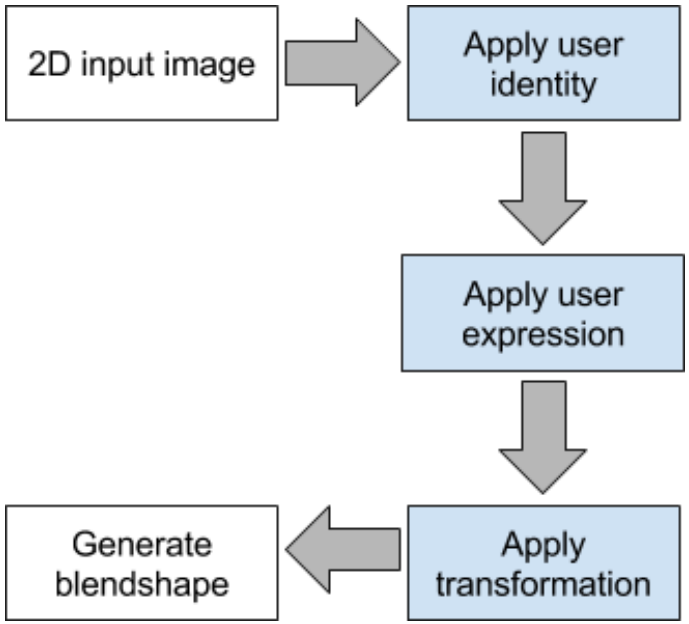


[3]

Generating Blendshapes

Use a FaceWarehouse generic blendshape model to calculate a user-specific blendshape model. For each image, three adjustments must be made to a generic blendshape:

- The user's identity: allow the regressor to transform the blendshapes to have the same facial features and shapes as the user
- The user's expression: in order to provide blendshapes with the same facial expression as the user
- Transformation: allow the regressor to take account for head turns, rotations, and translations



Selecting Training Data and Training the Regressor

Select training data so that the regressor is prepared to output a 3D facial shape from a single 2D image.

The regressor learns a regression function based on the information in the input images. The goal of training the regressor is to teach an effective prediction model through the geometric relationships that a specific user's image data contains.

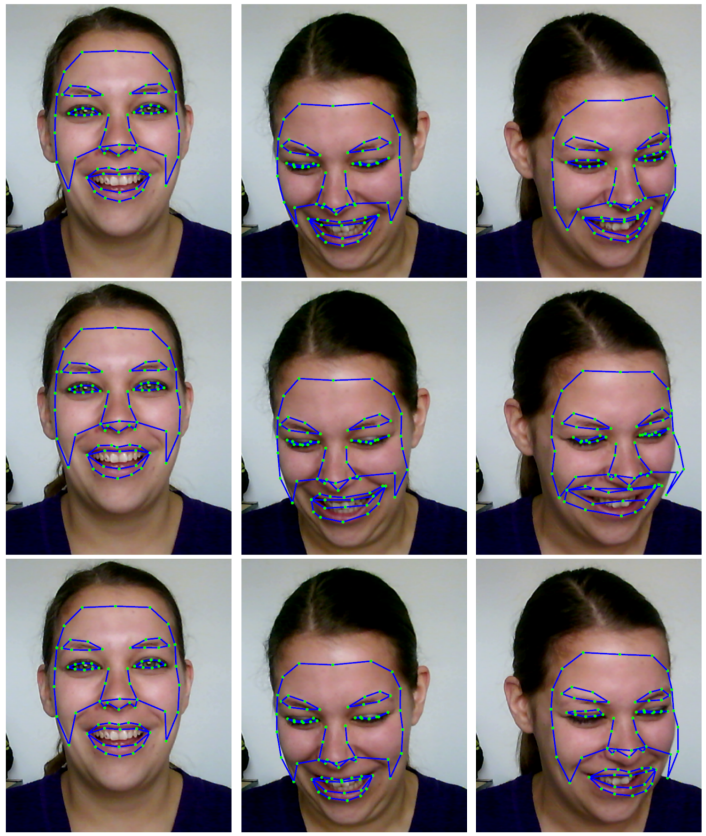
Runtime Regression

At runtime, the 3D shape regressor tracks the 3D positions of facial landmarks from a 2D video stream.

1. For each video frame, find a facial shape that is similar to the previous frame's facial shape
2. Transform the shape to align the previous frame's shape with the 3D shape space
3. Find a set of shapes in the training data that are similar to the transformed shape
4. Pass each shape in the set through the regressor to find the regressor's output
   - Update the current image's shape with the output
5. Average all of the outputs to make the final facial shape

Why this Regression Algorithm Works

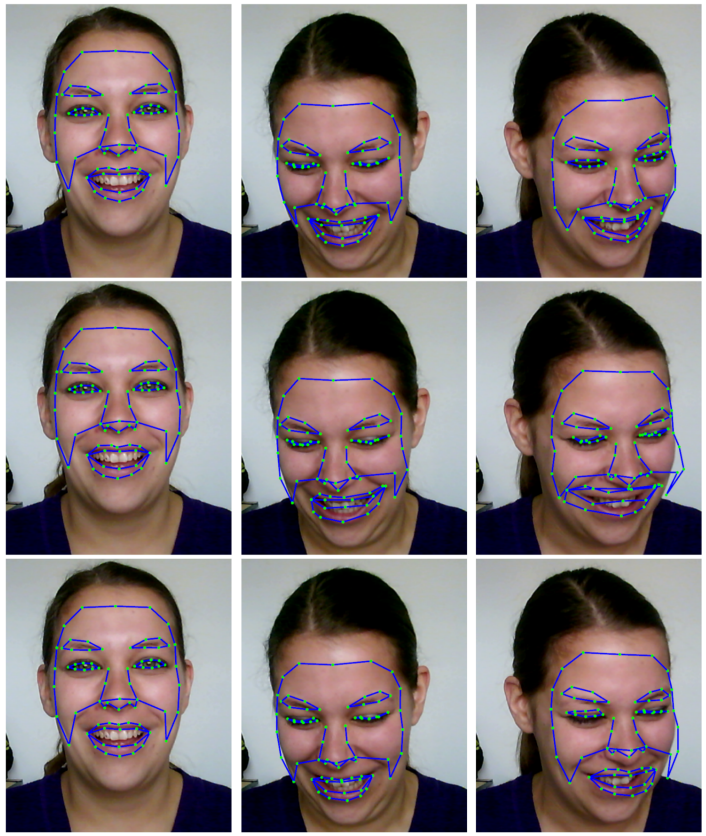
1. Regression uses a set of similar shapes instead of a single shape to generate the current shape
   - This allows the solution to deal with uncertainty and to avoid error accumulation



[3]

Why this Regression Algorithm Works cont.

2. Regression performs a transformation step before selecting a set of similar shapes
   - This allows the algorithm to account for different head positions and rotations



[3]

Results of 3D Shape Regression Tracking

- Implemented on a PC with an Intel Core i7 (3.5GHz) CPU, the overall runtime performance is less than 15 milliseconds
- The setup and preprocessing steps take less than 45 minutes per user
- This solution has limitations when there are large occlusions or when there are dramatic changes in lighting

Outline

DDE Model

- Designed to allow users to be switched automatically, without a user-specific setup step
- Designed to represent:
  - 3D facial shape of the user's facial expressions
  - 2D facial landmarks
- 3D facial shape is represented by a linear combination of expression blendshapes
- To represent a 2D facial landmark, add a 2D displacement to the projection of the landmark's corresponding vertex on the facial mesh
  - 2D displacement of facial landmark is what accounts for changes in user identity

Preparing Training Data

1. Facial images from FaceWarehouse are used as initial data
   - 73 2D landmarks are labeled on initial data to produce a 2D facial shape
2. Training pairs are created for each image to relate differences in parameters to image features
   - Training pairs simulate cases where input parameters may be inaccurate

[2]

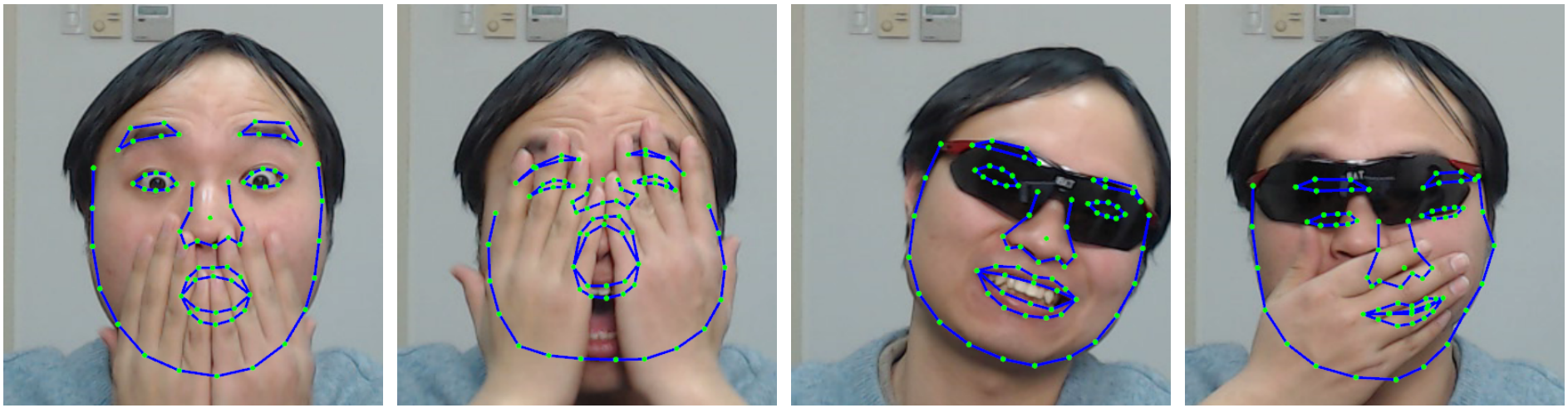
Training the Regressor and Runtime Regression

This approach follows a very similar regression approach to the 3D Shape Regression Training, except this algorithm must account for the use of a DDE model as shape representation.

The overall runtime regression algorithm is the same as the algorithm in the 3D Shape Regression Tracking algorithm.

Results of DDE Regression Tracking

- Implemented on a PC with an Intel Core i5 (3.0GHz) CPU, and the entire approach takes approximately 20 milliseconds on average
- Setup takes about six hours, but this step only occurs once
- This solution does better than the previous solution when dealing with partial occlusions, but it still produces an inaccurate result when there are large occlusions



[2]

Outline

Comparison of the Two Solutions

The DDE Regression Tracking solution is an improvement to the 3D Shape Regression Tracking solution. The improved solution:

- does not require a data gathering step (even if the initial tracking takes much longer)
- is not user-specific; users can be switched and swapped during video without a noticeable lag in output results
- is more robust than the previous solution, especially during dramatic lighting changes

Conclusions

- A preliminary solution for rendering 3D virtual images from 2D video frames in real time: 3D Shape Regression Tracking
- An improved solution that has been shown to be more robust, better at handling lighting changes, handles facial rotations more successfully, and does not requiring user-specific training: DDE Regression Tracking
- Further research includes solutions driven by both visual and audio data, as well as solutions that work completely online without the use of facial markers or training stages

Thanks!

Thank you for your time and attention!

Contact: saxxx027@morris.umn.edu

Questions?

References


alice-in wonderland.net.
About disney's 'alice in wonderland' 1951 cartoon movie.


C. Cao, Q. Hou, and K. Zhou.
Displaced dynamic expression regression for real-time facial tracking and animation.
ACM Trans. Graph., 33(4):43:1–43:10, July 2014.


C. Cao, Y. Weng, S. Lin, and K. Zhou.
3d shape regression for real-time facial animation.
ACM Trans. Graph., 32(4):41:1–41:10, July 2013.


O. Media.
Programming 3d applications with html5 and webgl.

See my *Modeling Facial Expressions* paper for additional references.