

# Performance-driven Facial Animation

Garoe Dorta Perez, Ieva Kazlauskaitė, Richard Shaw

University of Bath  
Centre For Digital Entertainment

27 May 2015

## Data Capture and 3D Reconstruction

Blendshape Model

Skin Rendering

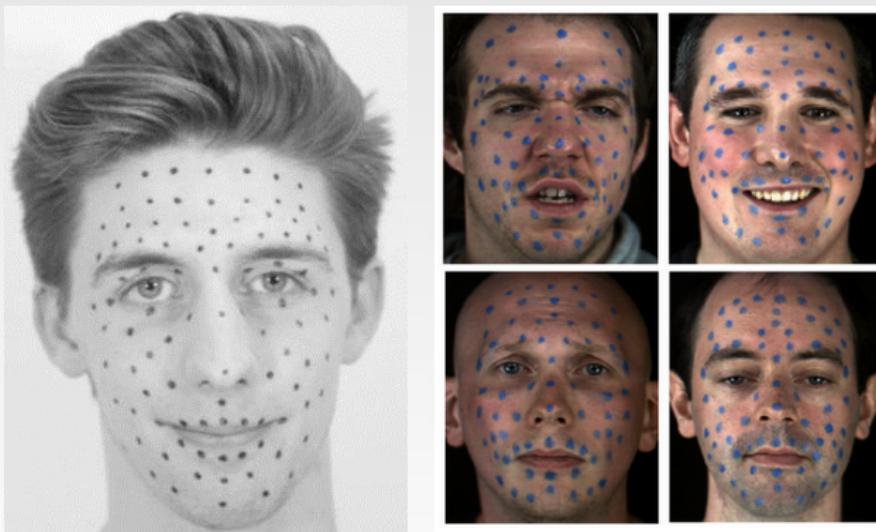
Results

- Performance capture using two DSLR cameras in stereo.
- Video recorded at 60fps with resolution  $640 \times 480$  pixels.
- Video streams synchronised using audio signals.



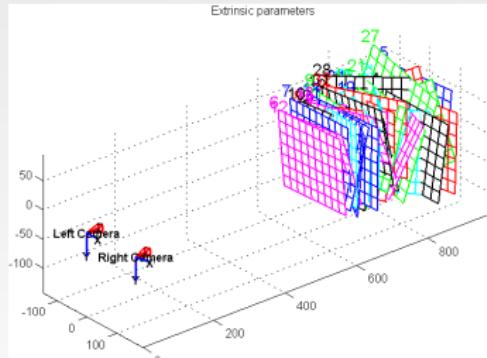
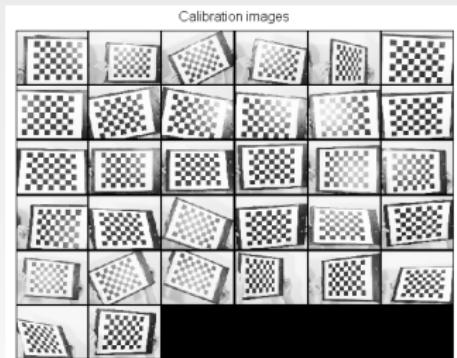
The data capture session using two DSLR cameras in stereo.

- Markers are drawn onto the actor's face to track the facial performance.



Marker positions were roughly based on the Surrey Audio-Visual Expressed Emotion (SAVEE) Database.

- A checkerboard pattern used to calibrate stereo camera setup.
- Obtain the cameras' intrinsic and external parameters.
- Compute the projection matrices and fundamental matrix.



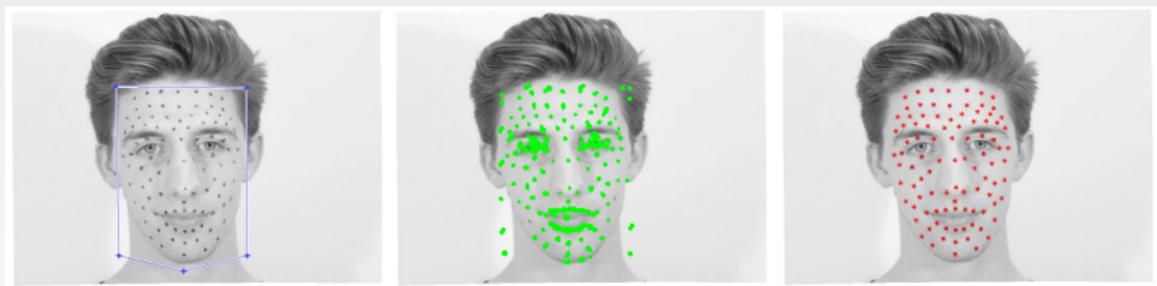
The stereo cameras were calibrated using a checkerboard pattern.

- Using the camera parameters we rectify each stereo pair for a captured image sequence.
- Corresponding epipolar lines lie on the same pixel rows.
- Reduces correspondence problem to a 1D search.



Each frame of an image sequence is stereo rectified.

- Face markers are detected by computing SIFT features in the left image of the first frame.
- Wrongly detected pixels can be removed and additional points included interactively.



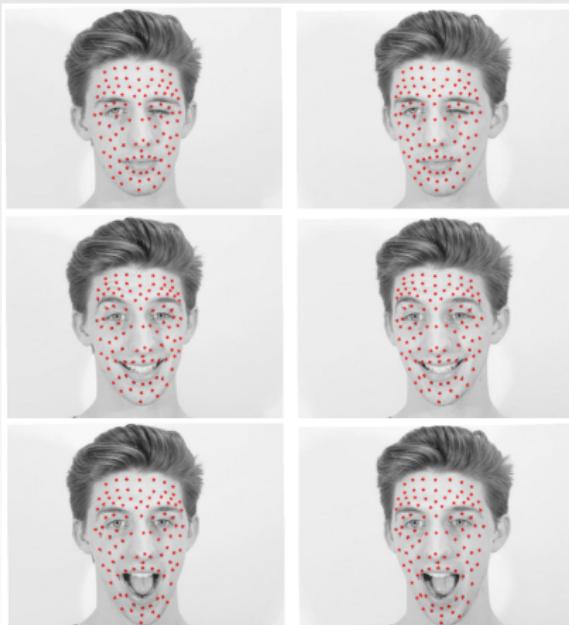
Markers on the face are detected using SIFT features.

- Corresponding markers in the right image are found by searching along the corresponding epipolar lines.
- Matching features are found by computing the normalised cross-correlation for image patches around each feature point.

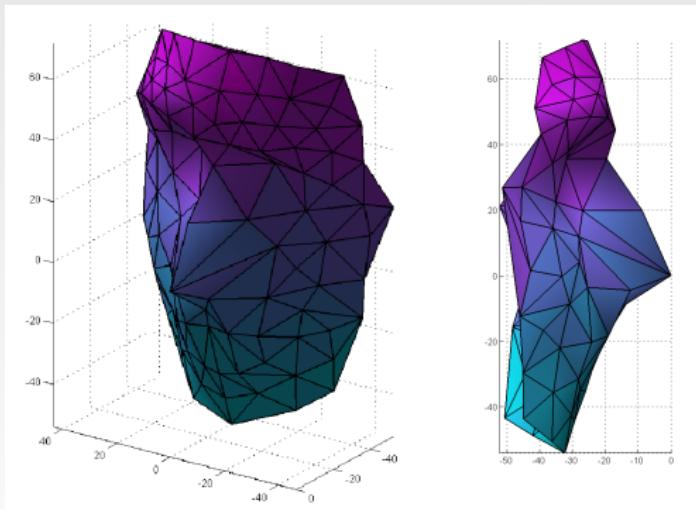


Corresponding markers are found in the right image.

- We track the face markers throughout the entire image sequence using the KLT tracking algorithm.

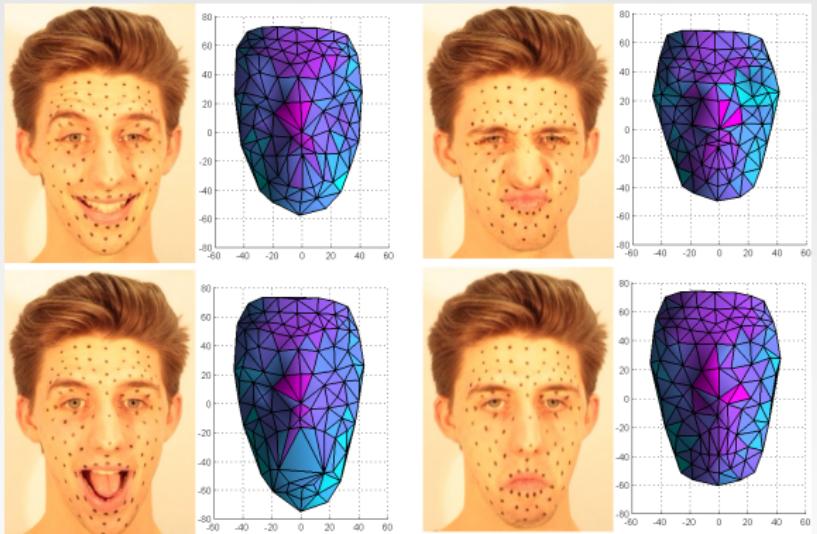


- Using the known camera projection matrices, we compute a sparse 3D reconstruction by triangulating.



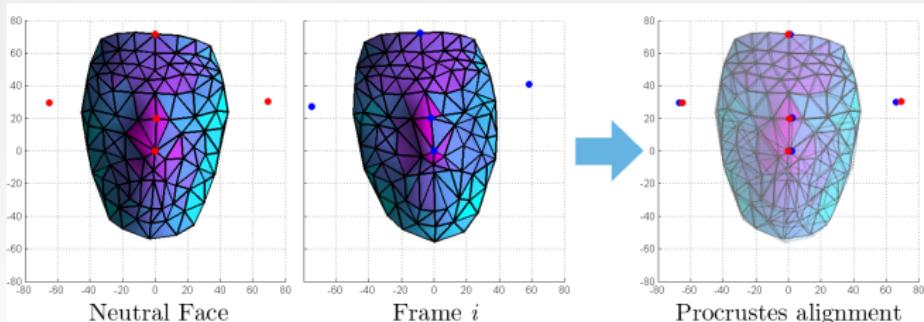
The 3D reconstruction of the neutral expression.

- Using the known camera projection matrices, we compute a sparse 3D reconstruction for every frame.



A sample of reconstructed frames from a captured image sequence.

- We attempt to remove the rigid head motion by performing Procrustes alignment.
- The head pose in each frame of an image sequence is aligned to the neutral pose.
- The 3D point trajectories are also smoothed to remove jittery head motion.



Rigid head motion is removed using Procrustes analysis.

Data Capture and 3D Reconstruction

Blendshape Model

Skin Rendering

Results

Given a set of blendshapes  $\mathbf{B} = [\mathbf{b}_1, \dots, \mathbf{b}_n]$  and a neutral face  $\mathbf{b}_0$ , a new facial expression  $\mathbf{F}(\mathbf{w})$  is a linear combination of the offsets of the basic shapes:

$$\underbrace{\mathbf{F}(\mathbf{w})}_{\text{new expression}} = \underbrace{\mathbf{b}_0}_{\text{neutral face}} + \sum_{i=1}^N \underbrace{w_i}_{\text{weights}} \underbrace{|\mathbf{b}_i - \mathbf{b}_0|}_{\text{offsets from neutral}} .$$

The **weights** describe how much each of the basic shapes affect the new expression.

Option 1 Create blendshapes **by hand** ⇒ correspond to a basic facial expression, for example a raised eyebrow.

- × Hard to produce.

Option 2 Use **Principal Component Analysis** ⇒ automatic construction.

- × Hard to adjust manually.

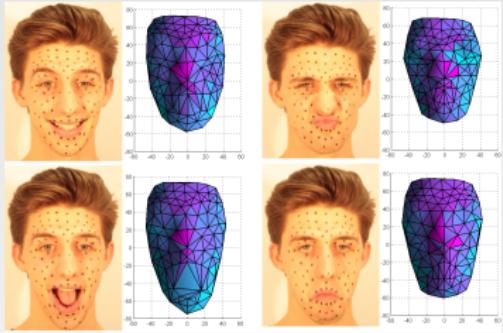
The weights  $w$  are estimated by solving the following:

### Minimisation problem

$$\min_w \| \underbrace{\hat{B}}_{\text{matrix with offsets}} \underbrace{w}_{\text{vector of weights}} - \underbrace{(F(w) - b_0)}_{\text{offsets of new expression}} \|,$$

where the weights add up to 1.

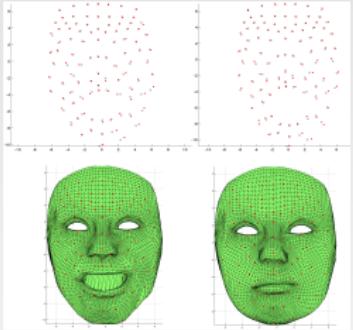
- Reduced set of blendshapes.
- Corrective shapes.



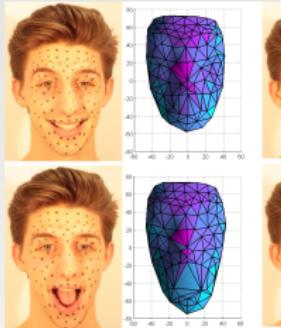
Recorded sequence



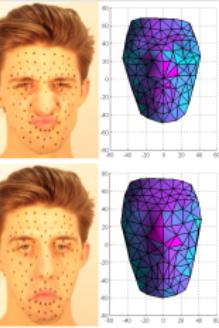
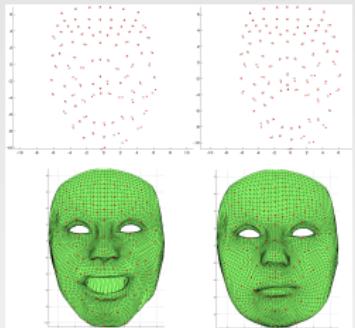
Thin plate  
splines



Emily sequence



Recorded sequence

Thin plate  
splines

Emily sequence



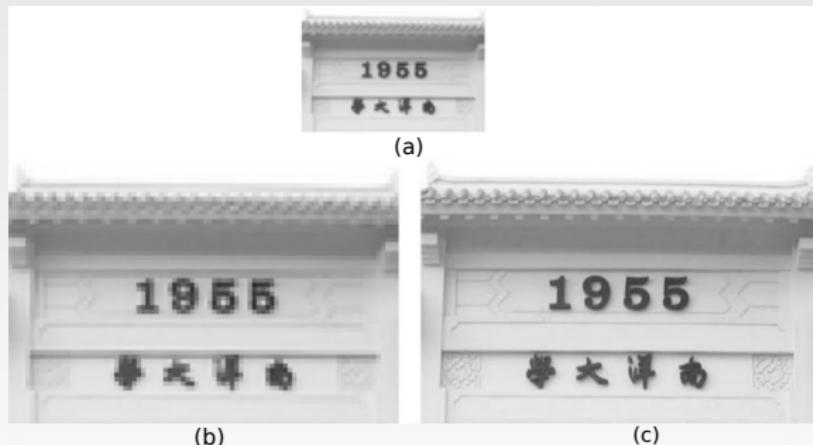
Data Capture and 3D Reconstruction

Blendshape Model

Skin Rendering

Results

- Improve rendering quality by increasing skin texture quality
- Increase the resolution of a texture image



Super-resolution example, (a) is original image, (b) is enlarged with pixel duplication and (c) is enlarge with super-resolution technique.

- General filter based on image examples
  - Pixel match for  $A'$
  - Coherence match for patch in  $B'$

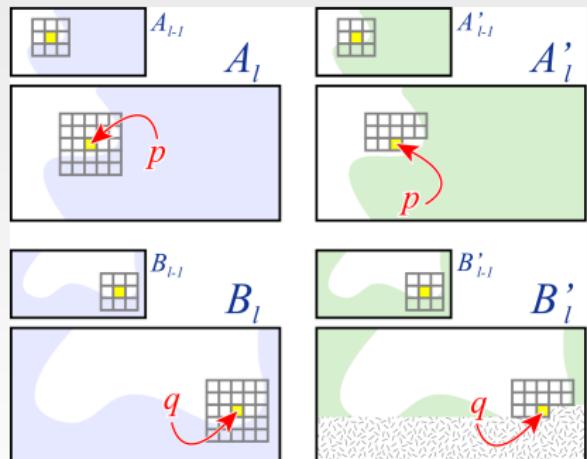
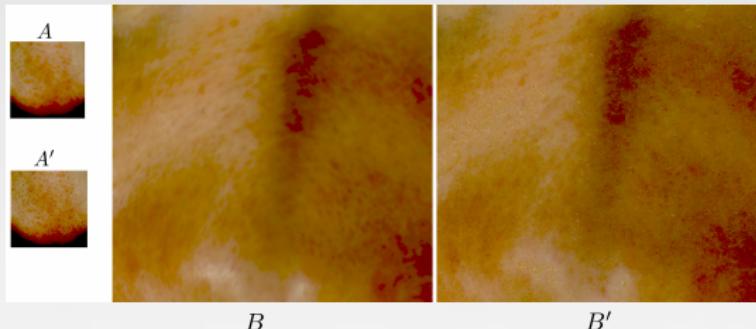
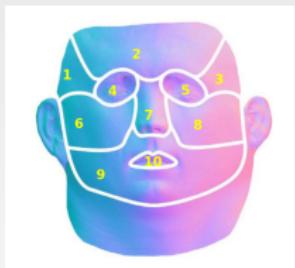


Image analogies diagram,  $A$  and  $A'$  are example images,  $B$  is input image,  $B'$  is output image,  $l$  is the synthesis level,  $q$  is the current pixel in  $B, B'$  and  $p$  the current pixel in  $A, A'$ .



With inputs  $A$ ,  $A'$  and  $B$ , Image Analogies filter outputs  $B'$ , images is falsoed-coloured to highlight differences.



Areas of face segmentation.



Original texture.



Filtered texture.

- Built dictionary of high resolution samples
- Synthesize new image taking patches from the dictionary

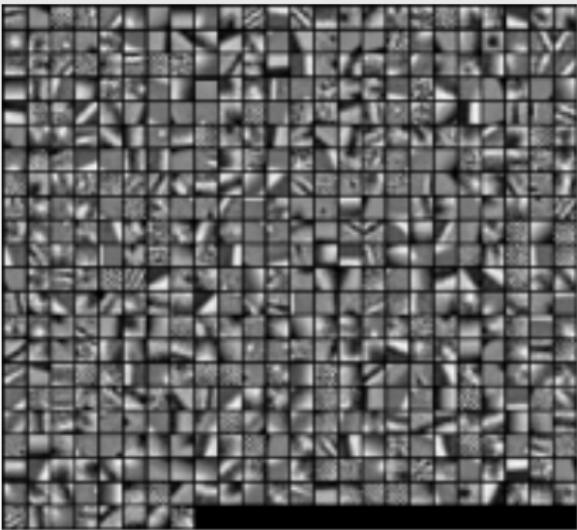


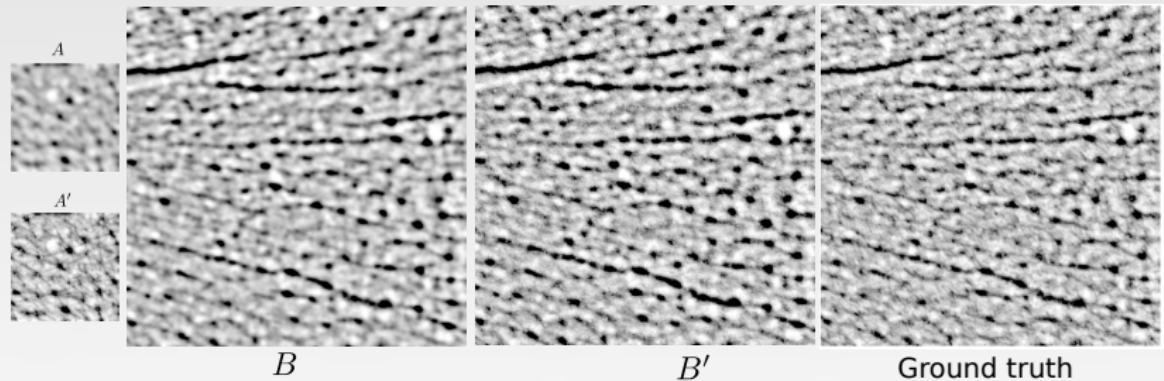
Image patches in the dictionary.



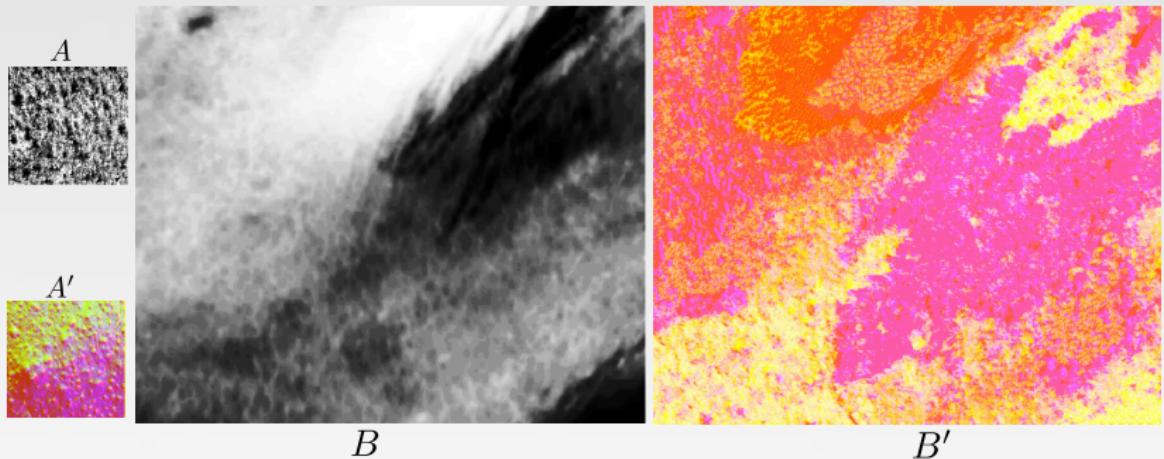
Texture super-resolution for Emily, (left) original texture, (right) synthesized texture with higher resolution.



Close up of texture super-resolution for Emily, (left) original texture, (right) synthesized texture with higher resolution.



Adding high frequency detail to a bump map.



Generating normal maps from gray scale textures, images is  
false-coloured to highlight differences.

Data Capture and 3D Reconstruction

Blendshape Model

Skin Rendering

Results



