A Facial Affect Mapping Engine (FAME)

Leonardo Impett, Tadas Baltrušaitis, Peter Robinson
Computer Laboratory
University of Cambridge, Cambridge, UK
li222@cam.ac.uk, tb346@cam.ac.uk, pr10@cam.ac.uk

Figure 1: (a) The puppeteer controlling the animation showing the automatic facial landmark and head pose tracking. (b) The avatar image being animated. (c) The final result of avatar being overlaid on the puppeteers image.

ABSTRACT
Facial expressions play a crucial role in human interaction. Interactive digital games can help teaching people to both express and recognise them. Such interactive games can benefit from the ability to alter user expressions dynamically and in real-time. This is especially important for people that have an impaired ability to understand and display facial expression, such as people with Autism Spectrum Disorders.

In this paper, we present the Facial Affect Mapping Engine (FAME), a framework for mapping and manipulating facial expression in real-time across images and video streams. Our system is fully automatic, and takes advantage of several computationally inexpensive methods to work in real-time, on generic hardware. FAME presents new possibilities for the designers of intelligent interactive digital games. We compare our approach with previous similar systems demonstrating its advantages. We also suggest some use-case scenarios enabled by our framework scenarios.

Author Keywords
Augmented Reality; Facial Affect; Face Tracking; Facial Puppetry; Face Swapping; Intelligent Games.

ACM Classification Keywords
H.5.1. Multimedia Information Systems

1. INTRODUCTION
Real-time augmented and virtual reality (AR and VR) systems are important tools in the creation of intelligent digital games. Often these are used to model static solid objects interactively, for instance to facilitate play in children with autism spectrum conditions (ASC) [11]. However, the ability to manipulate dynamic and non-rigid objects such as the face and head is more difficult, and an ability to do so would be particularly useful for AR and VR.

Figure 2: The role of animator and avatar in the system. The user can be either or both of these.

The transposition of facial emotion and identity is particularly relevant to video-interactive intelligent games. Facial expression is important in the communication of emotions, conversational turn-taking and empathy [8, 13]. In this paper, we present the Facial Affect Mapping Engine (FAME), a framework for animating unlabelled static images with video streams, or more generally, modifying mapping facial expression in real time across images and video streams (see Figure
1. Our framework allows for facial expression and head pose to be modified dynamically as it is mapped from one face (the animator) to another (the avatar) (see Figure 2).

This leads to three useful modes of operation; well-described as an interaction between a user, captured by web-cam, and an unlabelled reference source image or video, which could be from a dataset or synthesised. These two video streams swap roles as animators or avatars:

(a) A user live-animates a face from a source image. The reference face (avatar) is then blended onto the user’s video-stream. The user’s expression and head pose are transferred to the character in the source image.

(b) The source video animates the user’s face. For non-extreme head movements, that animated face is again overlaid on the user’s video-stream, so that the source video’s expression and head pose are transferred to the user.

(c) The user’s facial expression and head pose is exaggerated, attenuated or altered dynamically, then transferred back to the user’s video stream. Only one video-stream is used; the identity and expression are both from the same person.

2. RELATED WORK

A fundamental part of this system is avatar animation, known as facial puppetry, where a digital avatar is animated by a puppeteer through live video. Traditional facial puppetry systems are two-stage, based on face tracking and subsequent expression transfer. Saragih et al. [15] created a facial puppetry system that runs in real-time and requires no labour-expensive person-specific models, requiring only a single unlabelled avatar image in neutral expression.

A similar facial puppetry system has been used for a psychological experiment that explored the effect on attenuation of head-pose and facial expression in photo-realistic video-conferencing scenarios [4]. In this system, the cropped faces were shown to participants with no background. Our system enables researches to conduct similar experiments, with natural and realistic blending of the modified face onto the original video background.

Face swapping is also of relevance to this system, and research on this has tended to focus on image-based manipulation [3]. Successful systems for replacing faces in single-camera video have been developed [7, 17], though mainly based around complex and computationally expensive 3D models, rendering near-photo-realistic videos offline, or online using 3D video from a stereo scanner [18]. These offline or specialist-hardware implementations are unsuitable for use in interactive intelligent games; our system runs in real-time on conventional home computing hardware.

Video face swapping has been demonstrated by McDonald and Castro [5] in real time, focussing on identity swapping rather than expression transfer. Their technique involves warping a face-image directly over a live feed, and interpolating the average pixel values between the warped face and the original feed; their system shows some significant lighting and texture imperfections. A sample is shown from their system in Figure 4.

Our approach aims to place more focus on realistic re-synthesis of the video through computationally inexpensive techniques. In order to be used in intelligent games, the system must satisfy the following requirements:

1. Fully automatic (no training and set-up at consumer end)
2. Real-time
3. Run on commodity hardware
4. Realistic for user interaction

Each of the above systems meets some, but not all, of these requirements. Our approach aims to place more focus on realistic re-synthesis of the video through computationally inexpensive techniques. We incorporate the ideas from previous work to specifically meet these requirements for intelligent games.

3. ARCHITECTURE

The Facial Affect Mapping Engine uses techniques from both facial puppetry and face swapping to transfer identity and expression through video streams. Our system, unlike existing systems, uses three stages (see Figure 3):

1. Face tracking
2. Expression manipulation
3. Video re-synthesis

The user live-video and reference video can switch roles as ‘puppet/avatar’ or ‘puppeteer/animator’. Face tracking operates on both puppet and puppeteer in parallel, whilst expression manipulation combines the two, along with any expression modification, to make an animated puppet. Finally, the output video is synthesised by re-situating the animated puppet in either the user or reference video-streams. This is effectively a face-swapping operation.

3.1 Face Tracking

In order to track the geometry of the face, we need to track landmark points on the face together with the head pose. For tracking faces we use a Constrained Local Neural Field (CLNF) model [1], an instance of the Constrained Local Model (CLM) framework [14, 6]. CLNF model we use tracks 66 feature points in the face, including: face outline, outline of the eyes, eyebrows, nose outline, and inner and outer outlines of the lips.

Figure 4: A comparison of McDonald and Castro’s system (left) with ours (right)
Figure 3: The system architecture; from unlabelled video input to affect-mapped output

**CLNF Tracker**

The CLNF model is parametrised using $p = [s, R, q, t]$ terms, that can be varied to acquire various instances of the model: the scale factor $s$, object rotation $R$ (first two rows of a 3D rotation matrix); 2D translation $t$; and a vector describing non-rigid variation of the identity shape $q$. Our point distribution model (PDM) is:

$$x_{i} = s \cdot R(x_{i} + \Phi_{i}q) + t. \quad (1)$$

Here $x_{i} = (x, y)$ denotes the 2D location of the $i$th feature point in an image, $x_{i} = (X, Y, Z)$ is the mean value of the $i$th element of the PDM in the 3D reference frame, and the vector $\Phi_{i}$ is the $i$th eigenvector obtained from the training set that describes the linear variations of non-rigid shape of this feature point. The PDM was constructed from the Multi-PIE dataset using non-rigid structure from motion [16].

In CLNF we estimate the maximum a posteriori probability (MAP) of the face model parameters $p$:

$$p(p|\{l_{i} = 1\}_{i=1}^{n}, I) \propto p(p) \prod_{i=1}^{n} p(l_{i} = 1|x_{i}, I), \quad (2)$$

where $l_{i} \in \{1, -1\}$ is a discrete random variable indicating if the $i$th feature point is aligned or misaligned, $p(p)$ is the prior probability of the model parameters $p$, and $\prod_{i=1}^{n} p(l_{i} = 1|x_{i}, I)$ is the joint probability of the feature points $x_{i}$ being aligned at a particular point $x_{i}$, given an intensity image $I$.

Patch experts are used to calculate $p(l_{i} = 1|x_{i}, I)$, which is the probability of a feature being aligned at point $x_{i}$ (Equation 1). The tracker used in this work can either use an SVR based patch expert, or a more accurate Local Neural Field [1], that models non-linear relationships. This allows for a trade-off: faster but less accurate or slower but more accurate tracking.

We employ a common two step CLM fitting strategy [6, 14]; performing an exhaustive local search around the current estimate of feature points leading to a response map around every feature point, and then iteratively updating the model parameters to maximise Equation 2 until a convergence metric is reached. For fitting we use Non-Uniform Regularised Landmark Mean-Shift [1]. We employ a multi-scale approach for increased robustness and accuracy.

As a prior $p(p)$ for parameters $p$, we assume that the non-rigid shape parameters $q$ vary according to a Gaussian distribution with the variance of the $i$th parameter corresponding to the eigenvalue of the $i$th mode of non-rigid deformation; the rigid parameters $s, R, t$ follow a non-informative uniform distribution.

### 3.2 Expression Manipulation

In this stage, we extract the faces from both images, modify the expression in the puppeteer’s face and map that expression onto the puppet’s face. Having tracked the important feature points in the puppet’s and puppeteer’s video streams, we extract the tracked face image from the whole image frame by cropping. We then triangulate the face images with the points tracked, and piecewise affine-warp these faces to a neutral shape. This neutral-shape face image is then converted from the Red, Green, Blue (RGB) colour space to a floating-point representation in the Luma-Chrominance (YUV) colour space, represented by an achromatic brightness and two chromatic components. The luminance is normalised to facilitate subsequent comparison between expressions, and the YUV histograms are equalised to give better robustness against the variation of ambient lighting conditions.

At this point, the expression in a video source has been separated into a texture (image) and shape (triangular mesh). Changes in expression influence both texture and shape, but these can be controlled separately at this point.

The expression in the face shape can be modified through manipulation of the expression vector ($p$). We can attenuate or exaggerate facial expression by scaling the elements of the non-rigid shape vector ($q$). We can also attenuate or exaggerate rotation and translation of the head by manipulating the rigid shape parameters $t, R, s$. More sophisticated expression modification can be used to control specific fa-
Facial regions or expressions associated with specific emotions. Figure 5 shows an example of the expression modification possible with this technique.

Figure 5: Attenuation of expression using the expression vector. Left to right: expression amplitude = 0.0, 0.5, 1.0

Facial expression and head pose both produce changes in the face texture, due to changes in illumination, wrinkles and so on. For expression to be mapped realistically, these textural changes must also be present in the animated face. The dynamic features (illumination, wrinkles) from the puppeteer texture must be combined with the static features (identity) from the puppet texture.

Previous systems have used dense 3D mesh relighting techniques [18] or synthesised textures based on active appearance models [15]. Both methods are too computationally expensive for our purposes. We therefore apply a video expression ratio image (ERI), based on the approach outlined by Liu et al. [12]. The face images are manipulated in four stages, as illustrated in Figure 6:

1. We capture a single neutral-expression face image of the puppeteer. We then map that face image to a standard shape, so that we can compare changes in texture between this and subsequent face images.

2. We blur both the neutral face and the current puppeteer face to remove very fine noise, especially to do with misalignment of features despite warping to a common standard shape. For computational efficiency, we downsample the image at this point.

3. We create a new matrix, the luminance-component Expression Ratio Image (ERI), of the same size of the faces: each pixel within is the floating-point division of the corresponding pixels in the current puppeteer face by the pixel in the neutral face.

4. This ERI matrix is then used as a pixel-by-pixel multiplication factor for the neutrally-warped target avatar face (effectively making wrinkle and shadow areas darker).

If we take $A$ to be the neutral facial image of person A (the puppeteer), $A'$ to be their face image with current expression, and $B$ to be the neutral avatar face, then the animated avatar $B'$ can be expressed as:

$$B'(u, v) = \frac{B(u, v) \cdot A'(u, v)}{A(u, v)} \quad (3)$$

If face tracking is successful, ERI leads to very believable results; however, slight misalignments in facial tracking can easily cause spurious results. For instance, if a mole or dark spectacles deviate in position slightly from in the 'neutral' texture, the ERI will lead to high-luminance 'ghosts' around their original positions. Similarly, changes in luminance around the lips and hairline can be very sharp, and can lead to similar artefacts. Although slight misalignment is made tolerable by the blurring phases, the trade-off between loss of detail and sensitivity of re-alignment is unfavourable.

In order to counter these effects, we limit the domain of the ERI to only pixels which get darker (shadows, wrinkles and frown-lines). In fact, we found that almost no visual expression information was lost by discarding the 'lightening' data. The amplitude of this ERI is then attenuated by a dynamically-controllable weight factor (initially 0.33, which gave good results by visual inspection).

Figure 6: Our four-stage ERI system (left to right): the original puppeteer image; the puppeteer image warped to a standard shape; the resulting ERI map; ERI texture transferred onto the puppet image

3.3 Face re-synthesis

To resynthesise the face, the expression avatar texture is warped to the modified shape of the puppeteer. Hardware-assisted graphics is used to blend the new face and the background image. Previous offline photo and video systems [17] have used Poisson blending techniques to fit the face into the image without inconsistencies in lighting or colour, or used 3D models for re-lighting. These techniques are too computationally expensive for our real-time interactive applications on normal hardware. However, to approach photorealism, the output video must have credible lighting and colour.

Figure 7: An extreme case of face re-colouring: left, the animated blended directly onto background. Right, the face recoloured using our approach, then blended on

The output from the expression ratio image has been normalised in luminance, and inherits its colour distribution directly from the puppet’s initial image. It therefore needs significant re-colouring in order to be able to blend well with the puppeteer’s video stream. Various different recolouring techniques are implemented, such as histogram-matching in the chroma (YUV) or red-green-blue (RGB), linear shifting and multiplication.
The most effective and least computationally expensive method is to work in the RGB space, assume the colour spectra roughly follow Gaussian distributions, and linearly scale and shift the spectra to equalise the mean and standard deviation for each colour component. This technique is robust to large differences in skin colour and lighting conditions between the puppet and puppeteer (see Figure 7).

The colour compensation above acts in the same way on each pixel, and cannot account for asymmetric differences in lighting. Although local dynamic shadows and wrinkles are captured by the ERI mapping, face-wide static asymmetrical lighting conditions (such as strong side-lighting) change little from the neutral reference image, and so are not well-synthesised by ERI alone. Blending the face directly with transparency, while preserving global lighting, creates undesirable artefacts that reduce realism and expression transfer. Therefore, in order to preserve some of this lighting, a blurred undercoat is used (see Figure 8).

The facial region from the puppeteer image is blurred heavily, to eliminate facial features (which have a higher spatial frequency) but preserve face-scale lighting information (which has a low spatial frequency). The animated puppet face is then blended with this undercoat, which re-lights the puppet’s face appropriately for the scene with very little computational overhead, without any artefacts caused by the puppeteer’s facial features.

Figure 8: The facial undercoat shown above (not normally visible) helps to preserve local lighting conditions

4. RESULTS

The ERI system transmits dynamic shadows well. Although not as strong or as well-defined in the re-synthesised video as in the source video, local shadow shapes (such as those on the nose or brow under moving light sources) are replicated. Furthermore, the speed of reaction to changing shadow shapes in the re-synthesised video stream adds substantial believability and engagement in the case where the user is the puppeteer.

One common failure case for the system is occlusion in the puppeteer video stream. Although the face tracker is quite robust to occlusion, it is difficult to correct for it during face re-synthesis. Spectacles and hands in front of the face both disappear in the puppet face, and are blended out (see Figure 9).

The software produces near-photorealistic results in a wide range of cases (see Figure 10), displayed in real time (at 15-30 fps on a 2.4 GHz PC with 8GB of RAM). The engine itself is implemented in C++, using the OpenCV library and hardware-accelerated OpenGL graphics.

Figure 9: Left: successful affect mapping. Right: failure due to occlusion of the face; note the glasses and hand failing to display realistically

A stand-alone GUI prototype for FAME has been developed, allowing for dynamic control and user/reference role switching. Figure 11 shows this prototype in action with the live user as a puppeteer and an unlabelled image as a puppet, i.e. mode (a) operation; with windows showing puppeteer (i), puppet (ii) and re-synthesised video (iii).

Figure 10: Mapping

5. APPLICATIONS AND FURTHER WORK

Mode (a) operation, where the user animates a reference image, could be used in interactive games where the user is represented in the virtual space by another character. Interactive expression mapping may help the user relate to their virtual character, which is an important consideration for serious games. Furthermore, real-time emotive inference techniques such as vision-based computational mind-reading [9] can be used to exploit the information on facial expression present in the face-tracking system system. This information could be useful in the design of intelligent games; for controlling the emotive response of virtual characters in reaction to the user, and so on.

Our GUI prototype operating in mode (b), where a reference video animates the user’s face, has been used on videos from Baron-Cohen’s mindreading DVD [2] (see Figure 2). This was a collection of videos made by professional actors, replicating a set of emotions to assist children with Autism Spectrum Disorder (ASD) to recognise these emotions from facial expression and body-language. Children play through the videos on the interactive DVD, and choose a representative expression for each video clip. Although with training, the children successfully recognised the emotions expressed
in the videoclip dataset, they were less successful at recognizing these emotions in unseen videoclips from different actors [10]. It is proposed that changing the identity of the faces in these videoclips will help the children to generalize these expressions across people. In this instance, the ease of adding avatars to the system (with no manual assistance) becomes useful; being able to identify with the animated face (be it the user’s friends, or the user themselves) could improve recognition, learning/recall and generalisation of these emotions.

Mode (c) (users are both puppet and puppeteer) operation could usefully be exploited to assist with communication or empathy in multi-user gaming. With the appropriate hardware, such as a bidirectional augmented-reality communication screen or wearable augmented-reality hardware such as Google Glass, users could see their (or another) face on other real or virtual participants. The identity and facial expression of other participants from the user’s viewpoint could be controlled dynamically, which could be useful for interfering with interpersonal communication in a multi-participant game.

It would be of particular interest to combine this engine with a generative system. For instance, with reference videos providing the appropriate ERI texture mapping, the control in the shape-modification stage of expression manipulation could be exploited to introduce realistic variability. It would also be possible to extend the system with other, more sophisticated synthetic forms of facial manipulation at identity level: changing or varying age, race, gender and so on.

6. CONCLUSIONS
We have presented a Facial Affect Mapping Engine, a tool for use in interactive intelligent games where the modification, transposition or generalization of facial expression of emotion is important. It advances the state-of-the-art in real-time convincing, separate expression and identity mapping on non-specialised hardware, which is of real value for interactive intelligent games. Both the puppet and puppeteer can be controlled either from a reference video or from a standard web-cam, giving rise to three distinct modes of operation. The expressions from the puppeteer can be altered and reliably mapped to a synthesised face which maintains the puppet’s identity. Finally, the synthesised face is graphically blended into either the puppet or puppeteer’s source video, with a high degree of realism. This technique for facial affect mapping has important applications in interactive intelligent games, particularly in cases where facial expression and identity need to be separated realistically. It facilitates the generalisation of expressed emotions, and allows a game to ‘trick’ users by dynamically changing the expressions and identities of other participants.

7. ACKNOWLEDGEMENTS
The research leading to these results has received funding from the European Community’s Seventh Framework Programme (FP7/2007-2013) under grant agreement n 289021 (ASC-Inclusion).

REFERENCES


