Real-time Avatar Animation from a Single Image

Jason M. Saragih[1], Simon Lucey[1] and Jeffrey F. Cohn[2]
[1]ICT Center, CSIRO, 1 Technology Court, Brisbane, QLD 4069, Australia
[2]Department of Psychology, University of Pittsburgh, 4323 Sennott Square, Pittsburgh, PA 15213, USA

Abstract— **A real time facial puppetry system is presented. Compared with existing systems, the proposed method requires no special hardware, runs in real time (23 frames-per-second), and requires only a single image of the avatar and user. The user's facial expression is captured through a real-time 3D non-rigid tracking system. Expression transfer is achieved by combining a generic expression model with synthetically generated examples that better capture person specific characteristics. Performance of the system is evaluated on avatars of real people as well as masks and cartoon characters.**

I. INTRODUCTION

Non-verbal cues play a crucial role in communicating emotion, regulating turn-taking, and achieving and sustaining rapport in conversation. As such, face-to-face conversation often is preferable to text-based exchanges. Until recently, real-time conversation over distance was limited to text or voice transmission. With increased access to fast, reliable broadband, it has become possible to achieve audio-visual face-to-face communication through video-conferencing.

Video-conferencing has become an efficient means to achieve effective collaboration over long distances. However, several factors have limited the adoption of this technology. A critical one is lack of anonymity. Unlike text- or voice systems, video immediately reveals person-identity. Yet, in many applications it is desirable to preserve anonymity. To encourage a wider adoption of the technology to realise its advantages, video-conferencing needs to incorporate a range of privacy settings that enable anonymity when desired. A common solution is to blur the face, but this option compromises the very advantages of video-conference technology. Blurring eliminates facial expression that communicates emotion and helps coordinate interpersonal behaviour.

An attractive alternative is to use avatars or virtual characters to relay non-verbal cues between conversation partners over a video link. In this way, emotive content and social signals in a conversation may be retained without compromising identity. As reviewed below, person-specific active appearance models (AAM) have been proposed to achieve this effect. A system developed by Theobald and colleagues [16] enabled real-time transfer of facial expression to an avatar in a video conference. Avatars were accepted as actual video by naïve observers. Person-specific systems, however, require extensive labor and computational costs to train person-specific AAMs. For a system to be widely adopted, it must entail minimal overhead relative to standard video conference. We propose a real-time system that requires minimal effort to initialise, achieves convincing near photo-realistic avatars, and runs in real time over a video link.

Specifically, the system requires only a single image, in frontal pose with neutral expression. In the case of animating a non-human character, the user must additionally provide a set of predefined landmark points on that image. Despite the minimal use of avatar- and user-specific information, the system achieves convincing avatar animations.

II. RELATED WORK

Avatar animation is often referred to as facial puppetry, where the avatar/puppet acts is controlled by the user/puppeteer's facial expressions. A facial puppetry system consists of two main components: face tracking and expression transfer. Face tracking captures the user's facial deformations. Expression transfer then animates an avatar so that its expression best matches that captured from the user.

Non-rigid face tracking is one of the most widely researched topic in computer vision. The reason for this is the difficulty in handling inter-personal variabilities stemming from both shape and appearance as well as extrinsic sources including such things as lighting and the camera noise. The difficulty is compounded by the typical expectation of real time performance. Most non-rigid face tracking systems use a linear model to characterise variability of the human face. Examples include active shape models [3], active appearance models [11], 3D morphable models [2] and constrained local models [4]. Alignment is effected via generative or discriminative approaches. In generative approaches [2][3][11], the parameters of a linear model that minimise the distance between the model and image appearance is searched for using some kind of deterministic optimisation strategy. Discriminative approaches [12][13] , predict the optimal model parameters from the appearance of the face in the image.

Facial expression transfer has also received significant interest in the research community in recent years. It consists of learning a mapping between facial model parameters describing the user and the avatar. The most common approach to tackle this problem is by projecting the deformation field (i.e. the difference between features of a neutral and expressive face) of the user onto the subspace describing the expression variability of the avatar [16][17]. However, this approach requires a pre-learned basis of variation for both the user and avatar, which in turn requires a set of images or a video sequence that represents the span of facial expressions for that person. Such data may not be readily available or may be difficult to collect. As such, a number of works propose methods for generating images of different facial expressions from a single image [1][6], from which the

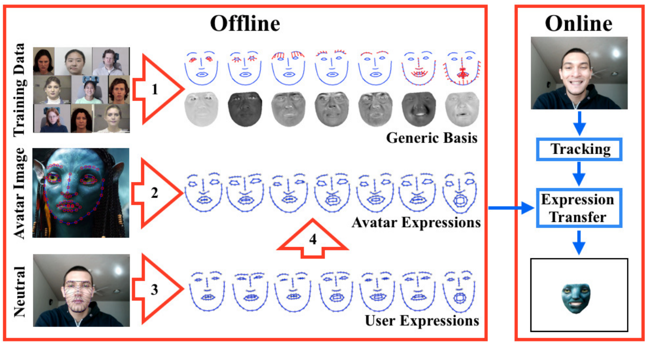


Fig. 1: Overview of the proposed system. Offline processing consists of four steps. **1:** Learn generic shape deformation and appearance bases that account for changes due to expression. **2/3:** Given an annotated image of an avatar/user, generate a set of prototypical expressions. **4:** Learn a mapping between corresponding expressions of the user and avatar. The online process involves non-rigidly tracking the user's facial features, performing expression transfer to the avatar and rendering the result.

person specific basis can be learned. Alternatively, one can use an automatic expression recognition system to detect the user's broad expression category and render the avatar with that expression [9]. Although such an approach requires a set of images for the avatar, no user-specific model needs to be learned. However, since transfer is performed at the coarse level of broad expressions only, this approach is not suited to applications where *realistic* avatar animation is desired.

III. SYSTEM OVERVIEW

An overview of the system proposed in this work is presented in Figure 1. It consists of two phases: offline and online. In the offline phase, models of the user and avatar are learned as well as the relationship between them. First a generic basis of variation that captures changes in shape and texture due to expression is learned from a training set of annotated images (§V-A). This same database is also used to learn a mapping between neutral facial shapes and a set of discrete expressions (§V-B). This map is used to generate synthetic expressive facial shapes for both the avatar and the user. A mapping is then learned between the user's shapes and corresponding ones of the avatar, where the generic basis learned previously regularises the solution (§V-C).

In the online phase, the user's face in the video stream is tracked using a nonrigid face alignment algorithm (§IV). The tracking algorithm provides the user's shape and texture that are then mapped onto the avatar using the mapping function learned in the offline phase. Finally, the avatar's face is rendered onto an image of any desired background using the mapped shape and texture.

It should be noted that whilst the generation of the user's discrete facial expression (step three in Figure 1) are placed in the offline phase, the user's neutral facial shape is captured using the same tracking algorithm used in the online phase. Given this shape, learning the mapping between the chosen avatar and the user takes less than a second, and new user registration can be performed seamlessly online.

IV. NON-RIGID FACE TRACKING

The real time non-rigid face tracking algorithm used in this work is based heavily on that in [15]. The approach is an instance of the constrained local model (CLM) [4] with the subspace constrained mean-shifts (SCMS) algorithm as an optimisation strategy. In the following we describe our additions to that work, which allows robust and real time nonrigid tracking suitable for avatar animation.

A. 3D CLM

Changes in head pose, such as nodding and shaking, are salient non-verbal cues in communication. In order to capture such variations we extend the work in [15] by using a deformable 3D linear shape model. Since the SCMS algorithm used for optimisation is invariant to the particular parameterisation of the shape model, the original fitting algorithm needs only be modified with respect to the computation of the shape Jacobian. The generative shape model we use takes the following form:

$$\mathcal{S}_i(\boldsymbol{\theta}) = s\mathbf{R}(\bar{\mathbf{s}}_i + \boldsymbol{\Gamma}_i\gamma) + \mathbf{t} \quad ; \quad \boldsymbol{\theta} = \{s, \mathbf{R}, \gamma, \mathbf{t}\}, \qquad (1)$$

where $\bar{\mathbf{s}}_i$ is the 3D coordinate of the mean i^{th} point, $\boldsymbol{\Gamma}$ is a 3D linear shape basis, and $\{s, \mathbf{R}, \mathbf{t}\}$ are the weak perspective projection parameters: scale, rotation and translation.

B. Fast Re-initialization

As with most face alignment algorithms, SCMS is initialisation dependent. Empirically we observed that when head movement between frames is large, the CLM is prone to loosing track. However, due to its locally exhaustive search procedure, we also observed that rapid changes in rotation can be handled effectively by SCMS since its landmarks typically move only within the range of the effective search regions. As such, we found it sufficient to re-initialise the model in each frame to account for head translation only. For this, we simply performed normalised cross correlation over the entire image for the location most similar in appearance to that of the face in the previous frame. Optimisation then proceeds from that location. The algorithm does not suffer from drift since the region describing the face in each image is inferred through the CLM optimisation procedure.

C. Failure Detection

To facilitate uninterrupted interactions, the system should be able to recover from cases where it fails to track the face. However, in order to recover from failure, the system must know when it has failed. Although this aspect is rarely discussed in the literature, it is a crucial component of a real-world system since there are no efficient algorithms that guarantee global convergence in each frame.

In this work we propose a very simple yet effective failure detection mechanism. Specifically, we use a linear support vector machine (SVM) trained to distinguish aligned from misaligned configurations. For SVM features we use normalised raw pixels since linear dimensionality reductions typically fail to preserve variations in appearance due to misalignment, and we found nonlinear approaches too computationally expensive for real time evaluation.

In order to specialise the failure detector to the particular fitting algorithm used in tracking, the SVM was trained with negative data that corresponds to local minima of the CLM objective in each training image. For this, we randomly initialised the CLM around the optimal configuration in each training image and the SCMS algorithm was run until convergence. If its distance at convergence from ground truth is above a user defined threshold, then the appearance at that configuration is used as negative data.

D. Acquiring 3D Shapes

In §V, we will describe a method for facial expression transfer that assumes 3D shapes for both the puppet and puppeteer in their neutral expression are available. The process of acquiring these shapes is described below.

For human avatars, the 3D face alignment algorithm described above can be used. However, when the avatar is non-human, the face alignment algorithm cannot be used since the appearance may not correspond to that of a typical face and the shape may not be spanned by the basis of shape variation. In this case, we require the user to annotate the avatar image with a set of 2D landmark locations corresponding to the 3D landmarks in the face model. The problem then reduces to *lifting* these 2D landmarks to 3D.

In order to perform lifting, we require that the non-human avatar is *human-like*, in the sense that the 3D geometry of its facial landmarks are *similar* to that of humans. This is not a very strong requirement in practice since the vast majority of virtual characters share many characteristics of the human face. We proceed then by assuming that the depths of each landmark point can be approximated by fitting the face shape model to the annotated landmarks and assigning the depths of those landmarks to that of the fitted face. Since the aim of this work is to provide a convincing avatar rather than an accurate 3D reconstruction, we find that this simplification works well in practice. Specifically, we solve:

$$\min_{\{z_i\}_{i=1}^n, \boldsymbol{\theta}} \sum_{i=1}^{n} \rho\left(\|[x_i; y_i; z_i] - \mathcal{S}_i(\boldsymbol{\theta})\|^2; \sigma\right), \tag{2}$$

where ρ is a robust penaliser, $\{x_i, y_i\}$ are the 2D-coordinates of the i^{th} user supplied landmark, z_i is its corresponding depth, and $\mathcal{S}$ is the 3D linear shape model in Equation (1). Following [14], we use the Geman-McClure function for the robust penaliser and derive σ from the median error. Equation (2) is minimised by the iteratively re-weighted least squares procedure. The pose normalised 3D shape of the avatar is finally obtained by inverting the image formation process, assuming a weak perspective projection:

$$\bar{\mathbf{x}}_i = s^{-1}\mathbf{R}^T([x_i; y_i; z_i] - [t_x; t_y; 0]), \tag{3}$$

where $\bar{\mathbf{x}}_i$ is the avatar's i^{th} pose normalised 3D landmark, and $\{s, \mathbf{R}, t_x, t_y\}$ are the rigid transformation parameters extracted from $\boldsymbol{\theta}$. Some example reconstructions using this approach are shown in Figure 2.

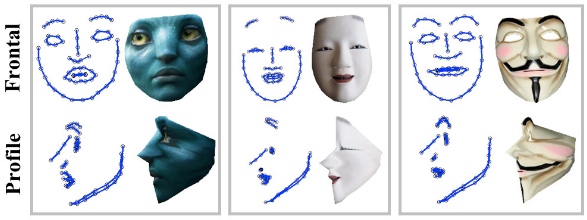


Fig. 2: 3D reconstruction from 2D landmarks.

V. FACIAL SHAPE TRANSFER

Given a pair of images, one of the puppet and the other of the puppeteer, along with the 3D shape of the face in each, which we denote $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$, the problem reduces to finding a mapping between them for various facial expressions, knowing only the 3D structure of their neutral expressions.

A. Generic Basis

In the absence of sufficient training data to build a fully descriptive model of shape variability, one can use a generic basis of variations. For example, in [1] the authors used the MPEG-4 facial animation parameters [10], which represent a complete set of basic facial actions, enabling the animation of most facial expressions. Using such a basis, the generative model of an avatar's shape takes the form:

$$\mathbf{x} = \bar{\mathbf{x}} + \boldsymbol{\Phi}\mathbf{p}, \tag{4}$$

where $\boldsymbol{\Phi}$ is the generic expression basis and $\mathbf{p}$ are the deformation parameters.

Although $\boldsymbol{\Phi}$ exhibits sufficient capacity to generate most facial expressions, it does not preserve identity. During puppetry, this may lead to shapes that depart from the avatar's space of variability. In works such as [16][17], this problem is alleviated by projecting the shape deformations onto the avatar's person specific subspace $\boldsymbol{\Psi}$[1]:

$$\mathbf{x} = \bar{\mathbf{x}} + \boldsymbol{\Psi}\boldsymbol{\Psi}^T\boldsymbol{\Phi}\mathbf{p}. \tag{5}$$

Although such a projection ensures that a generated shape perserves the avatar's identity, there are two shortcomings of this approach. First, as identified earlier, learning $\boldsymbol{\Psi}$ requires a large set of annotated data. For example, around 200 images and whole video sequences were used in [17] and [16], respectively, to learn their person-specific subspaces. In practice, collecting and annotating such large amounts of data online can be cumbersome, difficult or impossible. Secondly, such a formulation assumes that corresponding expressions between individuals can be described entirely by the reconstruction of these deformations. This can lead to under-articulation when the underlying models of deformation between the two faces differ significantly as a result of inherent differences in facial structure.

B. Semantic Expression Transfer

Given a large number of examples of both the puppet and puppeteer, along with semantic correspondences be-

[1] It should be noted that in [16], the subspace $\boldsymbol{\Phi}$ relates to the person specific basis for the puppeteer rather than a generic basis.

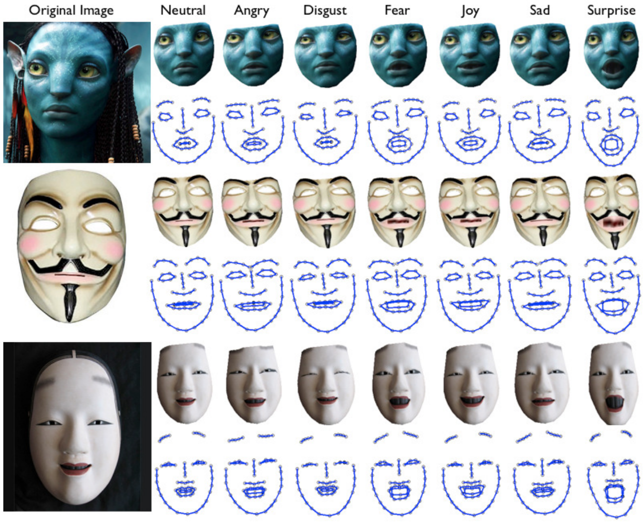


Fig. 3: Semantic expression transfer examples.

tween them, expression transfer can be treated as a supervised learning problem. Specifically, given pairs $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{N}$, where $\mathbf{x}_i$ is the i^{th} example of the puppet and $\mathbf{y}_i$ an example of the puppeteer with the same expression, the problem can be formulated as finding a mapping that minimises the prediction error over deformations:

$$\min_{\mathbf{M}} \sum_{i=1}^{N} \|(\mathbf{x}_i - \bar{\mathbf{x}}) - \mathbf{M}(\mathbf{y}_i - \bar{\mathbf{y}})\|^2, \tag{6}$$

where $\mathbf{M}$ denotes a linear mapping between the deformations of the puppet and puppeteer. Expression transfer then takes the form:

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{M}(\mathbf{y} - \bar{\mathbf{y}}). \tag{7}$$

However, as noted previously, in this work we assume that only a single example of both the puppet and puppeteer are available. Therefore, we follow recent work on expression synthesis [6] to generate *synthetic* examples which are used to learn this mapping. Specifically, given a database of multiple people, each captured displaying the same set of expressions, one can learn a set of mapping functions between the neutral face and each expression in the database:

$$\min_{\mathcal{M}_e} \sum_{i=1}^{N} \|\mathbf{x}_i^e - \mathcal{M}_e(\bar{\mathbf{x}}_i)\|^2, \tag{8}$$

where $\bar{\mathbf{x}}_i$ is the neutral expression for the i^{th} subject in the database, $\mathbf{x}_i^e$ is the same subject with expression e, and $\mathcal{M}_e$ is the mapping function for expression e. Once the mapping functions have been learned, examples for both the puppet and puppeteer can be synthesised and used in Equation (6) to find the mapping between their deformation fields. Some examples of this mapping are shown in Figure 3, where kernel ridge regression with a Gaussian kernel was used to parameterise $\mathcal{M}_e$.

The main problem with this approach to learning the relationship between the puppet and puppeteer's deformation fields is its data requirements. Existing expression databases, such as Multi-PIE [7] and KDEF [5], exhibit only a small set of facial expressions (typically corresponding to the seven basic emotions: neutral, joy, angry, sad, surprised, fear and disgust). Such small sets are insufficient for learning the mapping between the deformation fields since Equation (6) will be underdetermined. Although this situation may improve in the future, in the following section we present an approach that can leverage existing databases to learn more meaningful mappings.

C. Combined Generic-Specific Models

Although the synthesised samples described in the preceding section may not be sufficient to learn a complete mapping between the puppet and puppeteer, it is possible to leverage such data to learn a more meaningful mapping than that afforded by a generic deformation basis alone. Consider the following cost function for learning the mapping:

$$\min_{\mathbf{R}} \alpha \underbrace{\|\mathbf{R} - \mathbf{I}\|^2}_{\text{generic term}} + (1-\alpha) \underbrace{\sum_{e \in \mathcal{E}} \|\mathbf{R}\mathbf{p}_e - \mathbf{q}_e\|^2}_{\text{specific term}}, \tag{9}$$

where $\mathbf{R}$ is the desired mapping function, $\mathbf{I}$ is the identity matrix, $\mathcal{E}$ is the set of expressions in the database, $\alpha \in [0, 1]$, and:

$$\mathbf{q}_e = \mathbf{\Phi}^T(\mathcal{M}_e(\bar{\mathbf{x}}) - \bar{\mathbf{x}}) \tag{10}$$

$$\mathbf{p}_e = \mathbf{\Phi}^T(\mathcal{M}_e(\bar{\mathbf{y}}) - \bar{\mathbf{y}}) \tag{11}$$

are the projections onto the generic deformation basis of the synthesised shapes for puppet and puppeteer respectively. With the solution of Equation (9), expression transfer then takes the form:

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{\Phi}\mathbf{R}\mathbf{\Phi}^T(\mathbf{y} - \bar{\mathbf{y}}). \tag{12}$$

The first term in Equation (9) assumes deformations between the puppet and puppeteer have the same semantic meaning. Specifically, as $\alpha \to 0$, the mapping approaches the identity mapping, which simply applies the deformation of the puppeteer directly onto the avatar in a similar fashion as [1].

The second term in Equation (9) encodes semantic correspondence between the puppet and puppeteer as defined by the database $\mathcal{E}$. As $\alpha \to 1$, the problem approaches that in Equation (6), but with the addition of a generic subspace projection, which can be rewritten in matrix form[2]:

$$\min_{\mathbf{R}} \|\mathbf{R}\mathbf{P} - \mathbf{Q}\|_F^2, \tag{13}$$

where the columns of $\mathbf{P}$ and $\mathbf{Q}$ are $\mathbf{p}_e$ and $\mathbf{q}_e$ respectively for $e \in \mathcal{E}$. Given the small set of expressions in existing databases (typically seven expressions) and the high dimensionality of the generic basis $\mathbf{\Phi}$ (i.e. the MPEG-4 has 68 facial animation parameters), Equation (13) is typically underdetermined. However, this system of equations can be solved using truncated SVD as regularisation [8], which gives the solution:

$$\mathbf{R} = \mathbf{Q}\mathbf{P}^T\tilde{\mathbf{V}}\tilde{\mathbf{S}}^{-1}\tilde{\mathbf{U}}^T, \tag{14}$$

[2] $\|\mathbf{A}\|_F^2$ denotes the Frobenius-norm of matrix $\mathbf{A}$.

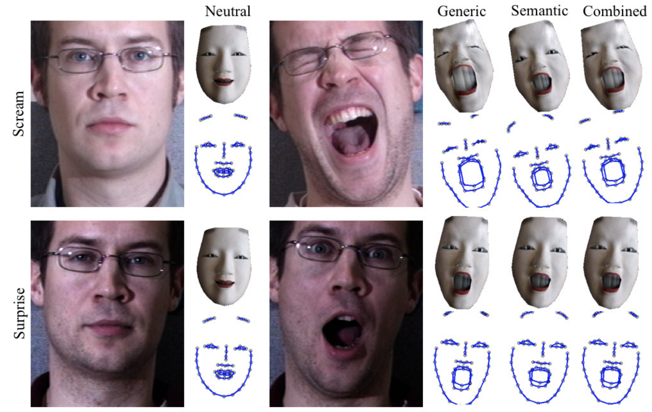


Fig. 4: Comparison between shape transfer methods, where semantic correspondence was not learned for the scream expression. Although the generic method better captures deformations that are not accounted for in the training set, person specific characteristics of the avatar are lost during animation (i.e. the expression appears exaggerated due to significant differences in facial shape at neutral). The semantic method fails to capture eye closing in scream, and appears to display a surprised expression instead. The combined method both preserves the avatars specificity and enables the transfer of expression components not modelled by semantic correspondence.

where, for $\mathbf{P}\mathbf{P}^T = \mathbf{U}\mathbf{S}\mathbf{V}^T$, we have:

$$\mathbf{U} = \begin{bmatrix}\tilde{\mathbf{U}} & \mathbf{0}\end{bmatrix}, \mathbf{V} = \begin{bmatrix}\tilde{\mathbf{V}} & \mathbf{0}\end{bmatrix} \quad \text{and} \quad \mathbf{S} = \begin{bmatrix}\tilde{\mathbf{S}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0}\end{bmatrix}. \qquad (15)$$

Since $\text{rank}(\mathbf{R}) \leq |\mathcal{E}|$ in this case, from Equation (12) it is clear that the effective span of the puppet's deformation is at most $|\mathcal{E}|$. With such a mapping, the puppet will be able to mimic only those expressions spanned by the training database $\mathcal{E}$.

By setting α to be a value between zero and one, one effectively learns a mapping that is both respectful of semantic correspondences as defined through the training set as well as exhibiting the capacity to mimic out-of-set expressions, albeit assuming direct mappings for these directions. The optimal choice for α will depend on the number of expressions in the training set as well as their variability. As a general rule, one should decrease α as the number of training expressions increases, placing more emphasis on semantic correspondences as data becomes available. Figure 4 illustrates the advantages of using a combined model as opposed to generic or semantic model's alone.

VI. Facial Texture Transfer

Unlike more sophisticated parameterisations that model the face shape using a dense point set [2], in our approach changes in facial texture cannot be modelled by a generative lighting model. This is because the sparse set of tracked points can not capture detailed changes in shape that give rise to changes in texture (i.e. the labial furrow in disgust etc.). As such, we must complement changes in the avatar's shape with that of texture.

The problem of facial texture transfer has many similarities to that of facial shape transfer discussed in the preceding section. However, the problem is complicated by the curse of dimensionality, where inference must now be performed over the space of pixels (typically > 10000) rather than over a sparse set of fiduciary facial landmarks (typically ≈ 100). In the following we describe an efficient approach that is capable of efficiently generating plausible changes in facial texture stemming from expressions.

A. Generic Texture Basis

Following the work in [16], we model facial texture in a shape normalised reference frame, where instances of the puppet are generated by inverse-warping the texture onto the image using a piecewise-affine-warp [11]:

$$\mathbf{I}(\mathcal{W}^{-1}(x, y; \mathbf{x})) = \mathbf{T}(x, y), \qquad (16)$$

where $\mathbf{I}$ denotes the synthesised image, $\mathbf{T}$ denotes texture in the shape normalised reference frame, (x, y) denotes image coordinates in the reference frame and $\mathcal{W}$ denotes the piecewise affine warp which is parameterised by the avatar's shape $\mathbf{x}$ as defined in Equation (12).

In a similar fashion to the generic shape basis discussed in §V-A, we use a generic basis of texture variation to model changes in appearance due to expression. In particular, we assume that changes in texture are linearly correlated with that of shape, and synthesise texture as follows:

$$\mathbf{T}(x, y) = \bar{\mathbf{T}}(x, y) + \sum_{i=1}^{K} p_i \mathbf{A}_i(x, y) \; ; \; \mathbf{p} = [p_1; \ldots; p_K], \qquad (17)$$

where $\bar{\mathbf{T}}$ is the neutral texture, $\mathbf{A}_i$ are the bases of texture variation and $\mathbf{p}$ are the shape deformation parameters (see Equation (4)). The texture basis is learned from a training set by solving the following least squares cost[3]:

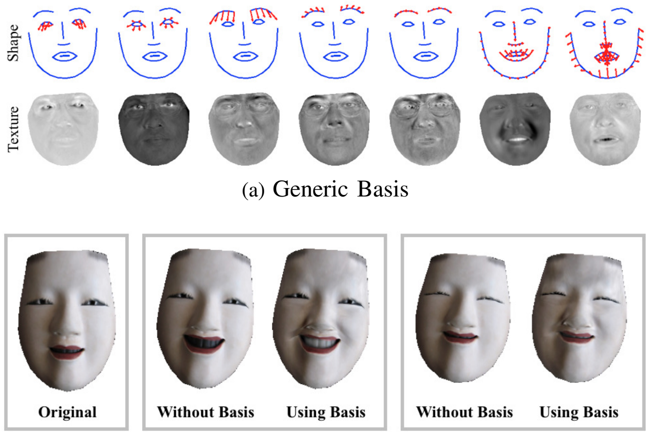
$$\min_{\mathbf{A}} \sum_{i=1}^{N} \|\bar{\mathbf{t}}_i + \mathbf{A}\mathbf{p}_i - \mathbf{t}_i\|^2 \; ; \; \mathbf{A} = [\text{vec}(\mathbf{A}_1) \ldots \text{vec}(\mathbf{A}_K)], \qquad (18)$$

where $\mathbf{p}_i$ denotes the shape parameters describing the expression in the i^{th} image, $\mathbf{t}_i$ is the vectorised texture for that image and $\bar{\mathbf{t}}_i$ is the vectorised texture for the same subject but in a neutral expression. In essence, Equation (18) learns a (non-orthogonal) basis that best models changes in texture as described through changes in shape. Since no further estimation is required apart from evaluating Equation (17) using the current shape parameters, this model yields rapid texture synthesis suitable for real-time applications. Figure 5 illustrates the utility of using this basis for rendering a more convincing avatar than that without texture augmentation.

B. Gaze Transfer

So far, the avatar is capable of mimicking the user's facial expression, but not her eye movements. Since changes in gaze direction can embody emotional states, such as depression and nervousness, an avatar equipped with gaze mimicking can appear much more realistic than one with a fixed gaze.

[3] The $\text{vec}(\mathbf{X})$ operator vectorises the matrix $\mathbf{X}$ by stacking its columns.



(a) Generic Basis

(b) Rendering Examples

Fig. 5: **a:** Generic shape deformation basis and their corresponding texture basis. **b:** Effects of rendering with and without the use of a texture basis. Notice that changes in texture due to expression, such as the appearance of the labial furrow in smile and disgust, add substantially to the perceived expression.

Learning a statistical appearance model of the eyes with only a few point correspondences is challenging. This is because the appearance model must account for translational effects of the pupil relative to the eyelids. It is well known that linear appearance models work poorly in such cases, with synthesis often resulting in significant ghosting artefacts that are not visually appealing.

Instead, in this work we explicitly synthesise the pupil within a region enclosed by the eyelids. The pupil is approximated by a circle whose appearance is obtained from the single training image. In addition to annotating the avatar's facial landmarks, this requires the user to also annotate the centre of the avatar's pupils and its radius. Parts of the pupil that are obscured by the eyelids in that image are replaced by assuming circular symmetry of the pupil's appearance. An example of the extracted pupil appearance is shown in Figure 6.

Gaze transfer is achieved by placing the avatar's pupils at the same relative location as that of the user's. First the location of each of the user's pupils, $\mathbf{x}_p$, are estimated as the centre of mass within the eye region, $\mathbf{\Omega}$, as determined by the tracking algorithm described in §IV:

$$\mathbf{x}_p = \frac{\sum_{\mathbf{x}\in\mathbf{\Omega}} w(\mathbf{x}_p)\mathbf{x}_p}{\sum_{\mathbf{x}\in\mathbf{\Omega}} w(\mathbf{x}_p)} \quad ; \quad w(\mathbf{x}_p) = \mathcal{N}(\mathcal{I}(\mathbf{x}_p); \mu, \sigma^2), \quad (19)$$

where $\mathcal{I}$ denotes the grayscale image and $\mathcal{N}$ denotes the Gaussian distribution with mean μ and variance σ^2. A similarity transform relating the inner and outer eye corners of the user and avatar is then applied to the pupil location, placing it in the avatar's image. Finally, the avatar's iris and sclera colours are scaled according to the eyelid opening to mimick the effects of shading due to eyelashes. An illustration of this process is show in Figure 6.

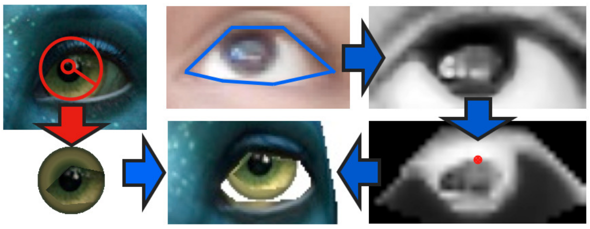


Fig. 6: An illustration of pupil extraction and gaze transfer.

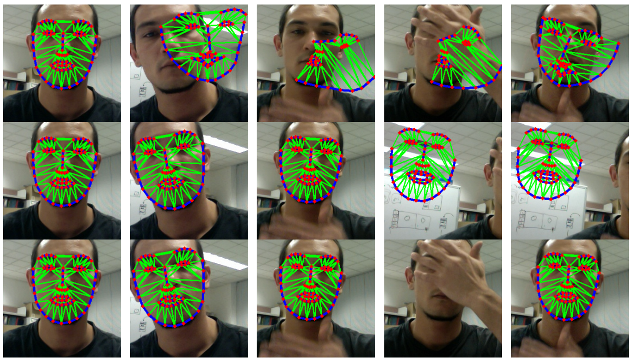


Fig. 7: Tracking example. **Top row:** Tracking using [15], **Middle row:** Tracking using fast re-initialisation (§IV-B). **Bottom row:** Tracking using fast re-initialisation and failure detection (§IV-C).

It should be noted that the ad-hoc approach described above will not, in general, obtain the precise gaze direction. Although more principled approaches to this problem exist, we stress that the aim of gaze synthesis in our application is not to infer gaze direction precisely, but rather to capture coarse eye movements that convey non-verbal cues. In §VII, we shown that this approach adequately captures such cues with little processing overhead.

C. Oral Cavity Transfer

Modelling texture variation in a shape normalised frame allows one to build a model by considering a fixed number of pixels. However, since the shape of the mouth can change dramatically between expressions, a single reference shape can not adequately capture changes in texture in the oral cavity, resulting in poor synthesis. Furthermore, variations in teeth, gum and tongue make learning generative models for the oral cavity extremely challenging.

As such, rather than modelling the appearance of the oral cavity, in this work we simply copy the user's oral cavity onto the avatar, using the piecewise affine warp defined within the mouth region. This way, the whole gamut of appearances can be accounted for with little computational cost. Furthermore, such a mapping acts to obscure small misalignments of the tracker around the mouth region. Results in §VII show that this simple strategy is effective in practice.

VII. Results

The evaluation of facial puppetry systems is inherently qualitative and is best seen through video. Nonetheless, in this section we present various snapshots of animation that act to highlight the various contributions of this paper[4].

[4] A collation of the videos used in all experiments in this section can be viewed at: http://www.youtube.com/watch?v=u6zTMQglQsQ

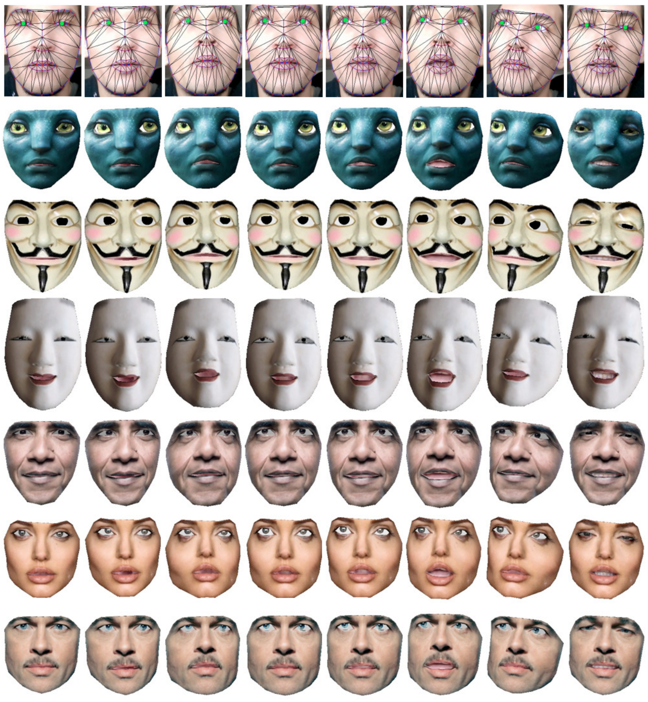


Fig. 8: Examples of gaze transfer for human and non-human avatars. The user's inferred pupil location is marked with a green circle.

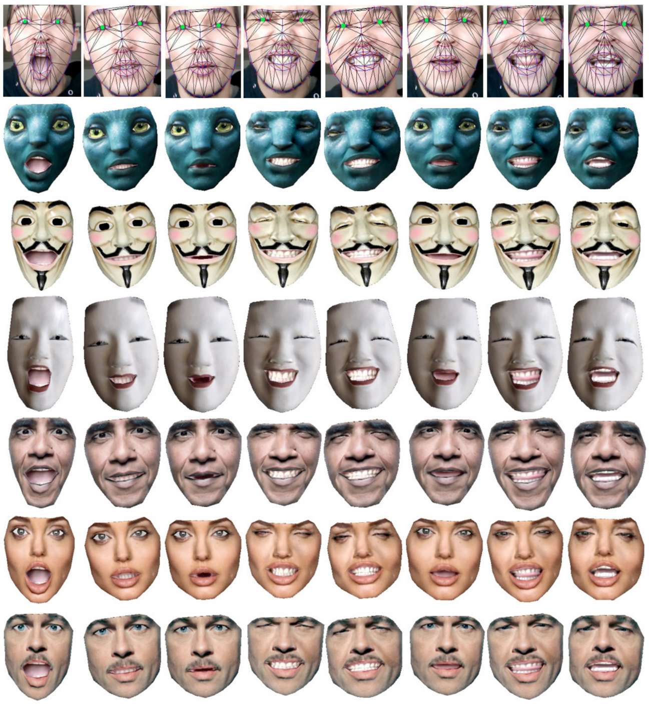


Fig. 9: Examples of oral-cavity transfer for human and non-human avatars.

A. Implementation Details

The facial puppetry system was implemented in C++ on a 2.66GHz MacbookPro with 4GB of memory. The average computation time for the various components of the system were: 58ms for new user registration, 23ms for tracking, and 19ms for expression transfer. The effective frame-rate of the complete system is 23fps, which is suitable for videoconferencing applications.

1) Tracking: The 3D CLM described in §IV was trained using 3000 images from the Multi-PIE database [7]. The 3D shape model was obtained by applying nonrigid structure from motion [18] on a 66-point manual annotation of these images, which retained 30 basis of nonrigid variation. All other components of the CLM were set according to descriptions in [15].

2) Shape Transfer: The generic basis used in facial shape transfer described in §V-A was trained using the Multi-PIE database. Using the 3D reconstructions obtained from structure from motion, the face was divided into three separate components: eyes, eyebrows and mouth/nose/jaw. For each of these components, a generic expression basis was learned by applying SVD to difference vectors between expressive and neutral shapes. The full basis was then constructed by appending the basis for each of the components:

$$\mathbf{\Phi} = \begin{bmatrix} \mathbf{\Phi}_{\text{eyes}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Phi}_{\text{eyebrows}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{\Phi}_{\text{mouth}} \end{bmatrix}. \tag{20}$$

By learning a basis for each component independently of all others, the resulting expression basis can generate more expressions than those present in the training set [2]. The resulting generic expression basis consisted of 30 modes of expression variation.

The synthetically generated expressive shapes described in §V-B were obtained using kernel ridge regression with a Gaussian kernel. The regressors were trained on images from the Multi-PIE [7] and KDEF [5] databases, where the total number of examples were 27, 229, 27, 478, 27, and 204 for anger, disgust, fear, joy, sadness and surprise expressions, respectively. The kernel width and regularisation constant were found through cross validation.

Finally, the weighting coefficient α in Equation (9) was set to 0.001, which was found to give good qualitative results through visual inspection.

3) Texture Transfer: The generic texture basis described in §VI-A was defined in a reference frame described by the convex hull of the mean face shape with a total of approximately 20,000 pixels. Since changes in texture due to expression mainly effect the luminance of the face, and because differences in camera colour models can cause undesirable changes in facial colour, this basis was learned from grayscale images only. When applying changes to the neutral face texture as is Equation (17), the changes were applied to each RGB channel equally.

B. Tracking Results

Figure 7 illustrates the utility of the additions to the method in [15] we outlined in §IV. In the second column of Figure 7, the subject executes rapid head movement, resulting in the failure of the algorithm in [15]. With the fast re-initialisation strategy described in §IV-B, the algorithm

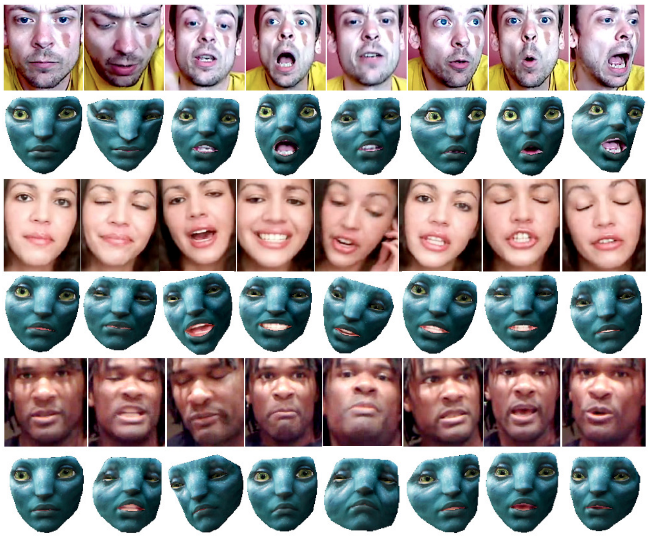


Fig. 10: Examples of animating youtube clips.

continues to track effectively. The fourth column in Figure 7 illustrates the case where there is gross occlusion of the face, resulting in tracking failure. However, the failure-detector described in IV-C successfully detects such cases and continues to track effectively once the occlusion is removed by re-initialising using a face detector.

C. Gaze and Oral Cavity Transfer Results

The puppeteering examples in Figure 8 and 9 were extracted from video captured using the inbuilt webcam on a MacBookPro. Figure 8 illustrates the efficacy of the method for gaze transfer proposed in §VI-B. Despite significant differences in eye size and shape, the proposed method successfully transfers the user's eye movements to the avatar. The method also allows the original pupil colour of the avatar to be used or changed in accordance with the user's preference. For example, the *V for Vendetta* mask (third row in Figure 8) uses black pupils since the avatar image contains no pupils (see Figure 3).

The efficacy of the oral-cavity transfer method proposed in §VI-C is illustrated in Figure 9. Despite significant differences in mouth size and shape, the proposed method generates convincing renderings of complex oral cavity appearances, including the presence of teeth and tongue. The method also has the effect of obscuring tracking inaccuracies, as exemplified in the last column of Figure 9.

D. Animating Youtube Clips

To illustrate the generalisation properties of the proposed system, we processed a number of youtube clips exhibiting people of varying ethnicity talking in front of a camera. Some example renderings are shown in Figure 10. These videos exhibit compression artefacts, sudden camera movements, discontinuous clips and unconstrained head motion. Despite these sources of variability, the proposed approach generates convincing animations without the need for user intervention.

VIII. Conclusion

In this paper, a facial puppetry system was proposed that runs in real time (23 fps), works on typical hardware and requires only that the user initialise the system by presenting a neutral face to the camera. The system achieves robust and real time non-rigid face tracking by appending a fast re-initialisation strategy and a failure detection scheme to a 3D-multiview variant of an existing tracking algorithm. Convincing facial expression transfer is achieved by leveraging generic deformation basis and synthetically generated expressive faces that are generated online. Gaze mimicking is achieved through pupil detection and oral cavity transfer is performed directly. Further improvements of the proposed system can be expected by improving the precision of the tracking algorithm and the use of training data with more variations in expression.

References

[1] A. Asthana, A. Khwaja, and R. Goecke. Automatic Frontal Face Annotation and AAM Building for Arbitrary Expressions from a Single Frontal Image Only. In *International Conference on Image Processing (ICIP'09)*, Cairo, Egypt, November 2009.

[2] V. Blanz and T. Vetter. A Morphable Model for the Synthesis of 3D-faces. In *International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'99)*, pages 187–194, 1999.

[3] T. Cootes and C. Taylor. Active Shape Models - 'Smart Snakes'. In *British Machine Vision Conference (BMVC'92)*, pages 266–275, 1992.

[4] D. Cristinacce and T. Cootes. Feature Detection and Tracking with Constrained Local Models. In *British Machine Vision Conference (BMVC'06)*, pages 929–938, 2006.

[5] A. F. D. Lundqvist and A. Öhman. The karolinska directed emotional faces - kdef. Technical Report ISBN 91-630-7164-9, Department of Clinical Neuroscience, Psychology section, Karolinska Institute, 1998.

[6] H. Dong and F. De la Torre. Bilinear Kernel Reduced Rank Regression for Facial Expression Synthesis. In *European Conference on Computer Vision (ECCV'10)*, 2010.

[7] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-pie. In *IEEE International Conference on Automatic Face and Gesture Recognition (FG'08)*, pages 1–8, 2008.

[8] P. Hansen. The Truncated SVD as a Method for Regularization. *BIT*, 27:534–553, 1987.

[9] I. Kemelmacher-Shlizerman, A. Sankar, E. Shechtman, and S. Seitz. Being. John Malkovich. In *European Conference on Computer Vision (ECCV'10)*, September 2009.

[10] F. Lavagetto and R. Pockaj. The Facial Animation Engine: towards a high-level interface for the design of MPEG-4 compliant animated faces. *IEEE Transactions on Circuits System Video Technology*, 9(2):277–289, 1999.

[11] I. Matthews and S. Baker. Active Appearance Models Revisited. *International Journal of Computer Vision*, 60:135–164, 2004.

[12] M. Nguyen and F. De la Torre Frade. Local Minima Free Parameterized Appearance Models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08)*, pages 1–8, 2008.

[13] J. Saragih and R. Goecke. A Nonlinear Discriminative Approach to AAM Fitting. In *IEEE International Conference on Computer Vision (ICCV'07)*, pages 1–8, 2007.

[14] J. Saragih and R. Goecke. Monocular and Stereo Methods for AAM Learning from Video. In *Computer Vision and Pattern Recognition (CVPR'07)*, June 2007.

[15] J. Saragih, S. Lucey, and J. Cohn. Face Alignment through Subspace Constrained Mean-Shifts. In *International Conference of Computer Vision (ICCV'09)*, September 2009.

[16] B. Theobald, I. Matthews, M. Mangini, J. Spies, T. Brick, J. Cohn, and S. Boker. Mapping and Manipulating Facial Expression. *Language and Speech*, 52(2-3):369–386, 2009.

[17] T. Weise, H. Li, L. Van Gool, and M. Pauly. Face/Off: Live Facial Puppetry. In *Eighth ACM SIGGRAPH / Eurographics Symposium on Computer Animation (SCA'09)*, 2009.

[18] J. Xiao, J. Chai, and T. Kanade. A Closed-Form Solution to Non-Rigid Shape and Motion Recovery. *International Journal of Computer Vision*, 2(67):233–246, 2006.