COMPUTER ANIMATION AND VIRTUAL WORLDS
Comp. Anim. Virtual Worlds 2004; **15:** 15–26 (DOI: 10.1002/cav.4)

Analysis of co-articulation regions for performance-driven facial animation

*By Douglas Fidaleo and Ulrich Neumann**

A facial gesture analysis procedure is presented for the control of animated faces. Facial images are partitioned into a set of local, independently actuated regions of appearance change termed co-articulation regions (CRs). Each CR is parameterized by the activation level of a set of face gestures that affect the region. The activation of a CR is analyzed using independent component analysis (ICA) on a set of training images acquired from an actor. Gesture intensity classification is performed in ICA space by correlation to training samples. Correlation in ICA space proves to be an efficient and stable method for gesture intensity classification with limited training data. A discrete sample-based synthesis method is also presented. An artist creates an actor-independent reconstruction sample database that is indexed with CR state information analyzed in real time from video. Copyright © 2004 John Wiley & Sons, Ltd.

Received: 6 March 2003; Revised: 30 June 2003

KEY WORDS: facial analysis; independent component analysis; performance-driven facial animation; action units; animation control

Introduction

Biologically inspired parameterizations of facial state have been used extensively for control of animated faces. The set of facial muscles and their actuations provides an intuitive basis for human face state-space, and consequently the majority of this work has focused on mapping facial motion data to virtual muscles fit to a 3D model. Dense optical flow data derived from facial motion provides a rich set of features commonly used for face parameter extraction[1,2] but is computationally expensive. Sparse point motion tracking increases the computational tractability of the problem, but to ensure stability, physical markers are often needed on the face.[3]

While muscles are an intuitive parameterization, we only have access to the visual effects of their contraction on the surface of the skin. Due to the large number of muscle degrees of freedom, and the corresponding non-linear deformations, analysis of true muscle contractions is very difficult. This is exemplified in recent work by Choe *et al*., who used hand-generated muscle actuation basis models both to estimate muscle actuation parameters and to produce 3D facial animation.[3] An initial set of muscle models was hand sculpted by an artist but an optimization phase was required to accurately reflect the muscle appearances. The system requires an actor to apply a set of 27 facial markers and uses motion information from the markers to determine muscle actuation. Though the approach achieved excellent results, robust estimation of muscle parameters requires a fitted model of the actor that can be costly and time consuming to produce.

Systems driven by sparse feature motion do not exploit all of the information present in the face. Regions such as the cheek have no trackable features, but do produce dynamic wrinkling effects that are key indicators of facial state. Appearance-based analysis can utilize this information to build a more robust control system.

Appearance-based facial analysis using the Facial Action Coding System (FACS) is widely used in the psychology community.[4] FACS defines 46 independent units of facial appearance change called action units (AU) and establishes a methodology for the manual estimation of AU activity from images of facial expression. Though multiple muscles are involved in AU actuation, each AU is parameterized by a single intensity parameter. Automated approaches to AU estimation have been explored.[5–7]

*Correspondence to: Ulrich Neumann, University of Southern California, PHE 404, 3737 Watt Way, Los Angeles, CA 90089, USA.
E-mail: uneumann@graphics.usc.edu

For AU data to be useful in animation, a fine level of actuation intensity must be determined; however, AU analysis research to date has focused only on binary classification (whether an AU is on or off).[5,7] Lien *et al*. analyzes intensity for full-face expressions by correlation to existing training images, but AU level intensity is not estimated.[6] The large number of AUs defined in FACS makes it a cumbersome interface for creating and editing facial animation. In addition, AUs are difficult to actuate independently for untrained subjects, often making training data acquisition impossible.

For many performance-driven facial animation systems (PDFA), a complete parameterization of facial state (as provided by muscles and FACS) is not necessary. We can constrain the parameters to a set of facial gestures that span *expression space*. For PDFA this expression space should contain emotional expressions (i.e., happiness, sadness, fear, disgust) as well as communicative gestures (i.e., smile, grimace, frown).

Our approach to facial analysis for PDFA control is a blend between muscle and action unit analysis. We introduce the notion of a co-articulation region (CR): a concise, intuitive, and physically meaningful local description of facial state. CR analysis with independent component analysis provides a fine level of actuation intensity. The parameters are actor independent and analysis can be performed in real time on modest consumer hardware. No encumbering facial markers are required. The collection of CRs defines a discrete facial state space that allows an artist to easily create new characters for flip-book style animation. Character creation consists of populating the state space with local face appearance data. Character appearance data can also be extracted from the original training video images for sample-based video synthesis. As the parameter space is identical for each actor, the created characters are actor independent.

Previous Work

Animation Control

Direct approaches to animation control have been widely explored. Facial motion capture data is used to directly interpolate 3D models using smoothing techniques in Guenter *et al*.[8] Noh[9] transfers motion vectors from existing animation to models with different geometry. In Brand,[10] statistical analysis is performed on synchronized facial motion capture data and digitized speech. Facial animation (2D and 3D) is produced entirely from speech input. Full-face analysis of expressions is used to interpolate base 3D models in Lo and Chung.[11]

Physically parameterized animation is performed in Choe and Ko[3] where muscle parameters are extracted from sparse point flow and used to compose muscle basis geometry for 3D character animation. Terzopoulos and Waters[12] track contour features on eyebrows and lips to animate the physically based muscle structure of a synthetic character.

An approach for model-based coding by Eisert and Girod[13] combines MPEG4 facial animation parameter (FAP) estimation with traditional block-based coding of head and shoulder video sequences. Dynamic facial features such as wrinkles and eye blinking are block coded at each frame. A feedback loop is used to estimate FAPs to achieve higher final signal-tonoise ratio. The process is offline (10 seconds per frame), and not suited for real-time application.

Image samples are extracted from video footage of a speaker and rearranged to generate animation driven by speech.[14,15] Such approaches where audible speech is analyzed and phoneme co-articulation effects are considered are causal in nature and therefore cannot be run in real time without noticeable delay. Our work, however, as it derives all information from instantaneous image data, is non-causal and can be use to drive a PDFA system in real time.

Facial Analysis

Facial gesture-unit analysis considers the parameterization of the face by a discrete set of independently actuated units. Low-level units of facial expression defined by the FACS AUs are analyzed using hidden Markov models trained on dense optical flow data and gradient information in Cohn *et al*.[5] and ICA in Donato *et al*.[7] Essa[1] uses optical flow and a finite element model of the face surface to estimate a set of actuation parameters termed FACS+ that encapsulates gesture velocity information.

An image of a face can be interpreted as a point in a high-dimensional space where the dimensionality is defined by the number of pixels in the image. Linear models for representing images of facial gestures, visemes, and head pose based on principal component analysis (PCA) have been explored extensively in the vision literature. A general framework for analysis and synthesis using example-based linear models is explored in Vetter and Poggio[16] with applications to facial pose and gesture modeling. Example-based learning is

also used in Beymer *et al*.[17] for representation of prior knowledge of faces under pose variations, and subsequently for facial recognition. Okada and von der Malsburg[18] build a piecewise linear subspace model of pose variations from images using PCA. PCA is also used in Reveret and Essa[19] for modeling the space of visual speech parameters and applied to 2D and 3D mouth animation. Local PCA kernels are used in the eigenfeatures[20] approach, which is most similar to our analysis methods.

In these methods, analysis and synthesis are performed using the same PCA basis. PCA serves this purpose well as it has optimal reconstruction properties;[21] however, facial state parameters are only encoded implicitly. For parameterized facial animation the facial state parameters must be made explicit and therefore classification rather than reconstruction is of primary importance. Donato *et al*.[7] have empirically shown that ICA performs better at classification of facial gestures using the FACS representation. We therefore use ICA to model the space of changes in each CR.

As most facial appearance changes due to muscle contraction are local, a more concise representation can be obtained by constructing local subspaces that encapsulate these changes. This approach is used in the eigenfeatures technique, where local appearance bases are constructed using PCA.[20] In our work, locality is defined by the co-articulation regions with each subspace modeled using ICA and parameterized by a set of facial gestures. CRs are defined to explicitly restrict the number of muscle degrees of freedom in a set of gesture samples.

Statistical learning algorithms generally require a large amount of data for effective classification. We show that correlation-based classification in ICA space proves to be both efficient and robust, enabling real-time control of synthetic faces with limited training data.

Facial Gesture Set and Partitioning Motivation

Of the approximately 200 muscles in the human head, there are 26 facial muscles primarily responsible for communication. Many of these are co-articulated (contracted in unison for a given gesture), and hence may be merged into fewer muscle groups. Faigin[22] identifies 15 muscle groups that are key indicators of emotion in artistic renderings of faces. We isolate and analyze nine facial gestures tied to these muscles. Though these muscles are not a complete basis for facial state, they span a large portion of the state space and are suitable for performance-driven facial animation. As we are focusing on emotional gestures, we have left out muscles such as the masseter, temporalis, and medial pterygoid, responsible for motion of the mandible.

Though we can represent facial state by a linear muscle basis, we only have visual access to the effect of the contraction on the skin. The resulting images of non-linear skin deformation have a prohibitively high number of degrees of freedom. We therefore partition the face into a set of regions such that the number of muscle degrees of freedom of each region is constrained.

Co-articulation Regions

Each facial muscle group listed in Figure 1 is capable of contracting independently and causing secondary

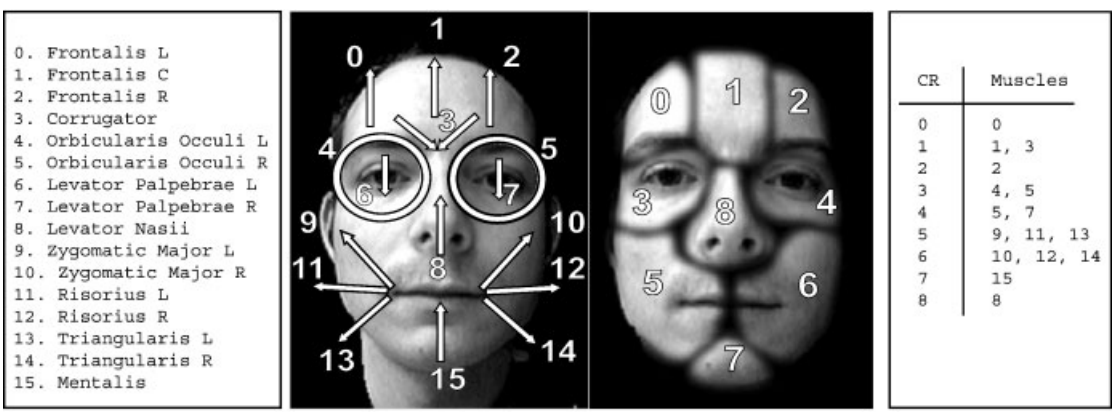

Figure 1. (Left) List of muscle groups and their locations and directions of contraction. (Right) Labeled co-articulation regions and the list of contributing muscles.

motion of the skin in a continuous and local area on the face. This area is defined as the muscle's region of influence (ROI). The state of each muscle is parameterized by a contraction value between 0 (relaxed) and 1 (fully contracted). When a muscle is actuated independently, the changes that are propagated to the skin surface are local and fully determined by the muscle's level of contraction. When two or more muscles have an overlapping region of influence, the resulting skin change is a combination of the effects of the involved muscles. We model the changes occurring on the surface of the face as a set of contiguous local regions of skin deformation called co-articulation regions (CRs).

A CR is analogous to a FACS AU but is more specifically defined as a non-empty intersection of n muscle regions of influence. The activation level of a CR is defined by the n-tuple of activation levels of each muscle contributing to the region's deformation (in contrast to a FACS AU, which defines a 1D activation level for each AU.[4] Each CR defines the state space for a local region on the face whose dimensionality is given by the number of muscles acting on the region. This CR subspace parameterization defines a mapping from a point in CR appearance space (the space defined by all possible image samples in a given CR), to a point in muscle space.

For each muscle there is a related facial gesture that exercises that muscle (Figure 2). We compute the ROI of each gesture by subtracting the maximally actuated gesture image from a neutral expression image. The ROI boundary is identified manually around the area(s) exhibiting large variations in the difference image. ROIs with significant overlap are merged and the resulting region boundaries define the nine CRs. A specific example of CR state space parameterization is illustrated in Figure 3.

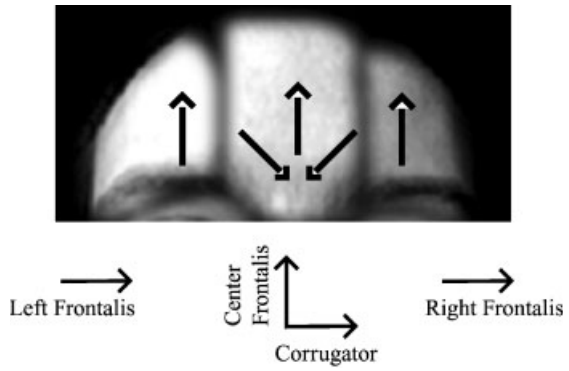

Figure 3. State space defined for three co-articulation regions in the forehead area. The two inward pointing muscles are the corrugator and are contracted in unison. CR0 and CR2 are independent whereas CR1 has a 2D state space.

Gesture Sample Acquisition

Labeled data is required to model the appearance changes in the co-articulation regions. Ideally we would like to have a uniform sampling of the entire CR state space; however, it is difficult to acquire such data from a human subject. This would require the subject to contract their muscles precisely and be conscious of each amount of contraction, but most people do not have such absolute (conscious) control. However, most subjects can produce the gestures in Figure 2 with practice (the same gestures used to define the CRs). These gestures correspond to the basis actuations of a given CR and, by controlling the gesture from neutral to full contraction, we can label the frames with respect to their relative actuation levels for a single actuation.

The subject is instructed to perform each gesture starting from a neutral expression, holding at full actuation, and releasing back to neutral, repeated six times. These gestures are recorded to a 320×240-pixel grayscale image sequence. The recorded gestures are broken into three segments: positive, apex, and negative (Figure 4). The positive segment spans from the first non-neutral frame indicating gesture motion to the frame preceding maximum actuation. The apex segment consists of all frames of maximal actuation. The negative segment spans the first frame after the apex to the frame preceding the final neutral state. The video frame rate

Figure 2. Facial gestures selected to actuate the defined muscle set. These gestures provide appearance information for the given muscle contractions.

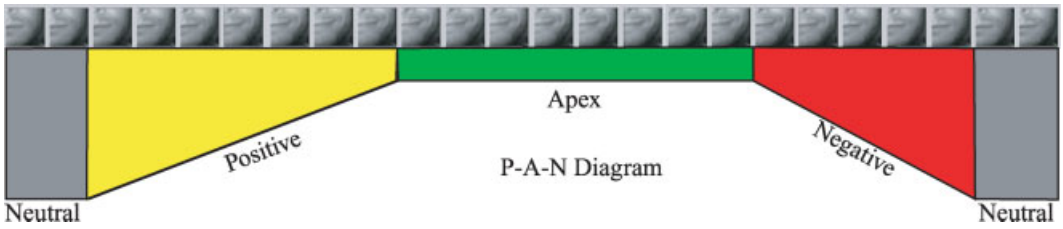

Figure 4. Each gesture sequence is broken into three segments. Only the positive segment is used for training.

Gestures	Muscles
1. Forehead Raise	0, 1, 2
2. Eye Squint	4, 5
3. Eye Blink	6, 7
4. Smile	9, 10
5. Frown	13, 14
6. Cheek Pinch	11, 12
7. Brow Furrow	3
8. Chin Crinkle	15
9. Snarl	8

Figure 5. Gesture labels and the muscles for which they provide appearance data.

and actuation velocity determine the number of frames in a given segment, and hence the quantization resolution of the state space. Figure 5 shows the gestures and the muscles for which they provide appearance information. Labeling is performed separately for each CR to account for potentially asymmetric actuation of gestures.

For each muscle affecting a given CR, we extract a set of samples from the positive segment of the corresponding gesture. All samples are masked by the CR template, and only the pixels in the bounding box around the CR are retained. Each sample x_j is transformed by horizontal scan to a column vector and concatenated to form the muscle data matrix $X_i = [x_0 x_1 \ldots x_k]$, where $1 \leq i \leq 15, 0 \leq j \leq k$ and k depends on gesture velocity and can vary for different muscles. The data matrix C_n is formed for CR n by concatenating muscle matrices according to the region-to-muscle mapping. Each column in C_n is labeled with respect to the gesture and intensity it represents. A pictorial representation of the data matrix for CR_6 is shown in Figure 6, with the samples from each muscle gesture ordered by their temporal progression. The data matrices defined above are used for gesture training.

System Overview

Figure 7 illustrates the online analysis and synthesis system. All input images must be normalized with respect to the actor's head pose and environmental lighting conditions prior to analysis. The actor's head frame is defined by a set of four infrared LEDs fixed to a pair of empty glasses frames. The LEDs are tracked and the image subsequently warped to a canonical pose. The warped image is convolved with a Laplacian kernel.

In an initial training phase the actor's CRs are extracted and the appearance properties are learned. Character creation is performed offline, where an artist creates a set of representative gesture states for each region and desired intensity levels.

In each video frame, all nine CRs are analyzed and a CR state vector is extracted. With the assumption of a neutral state outside of the CR set, an actor's face is represented by the collection of individual CR states. The state vector indexes the appearance samples in a

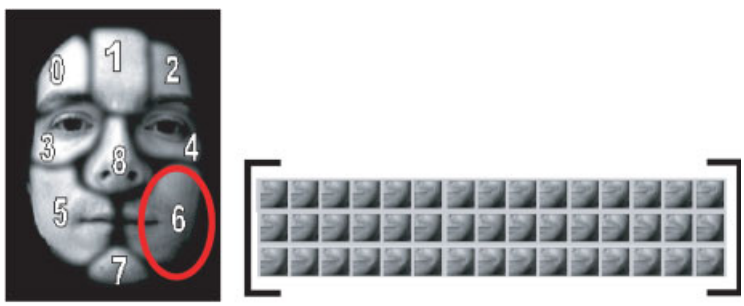

Figure 6. Gesture samples for CR6 are vectorized and assembled by column into a single data matrix. The gesture label and intensity are retained for classification.

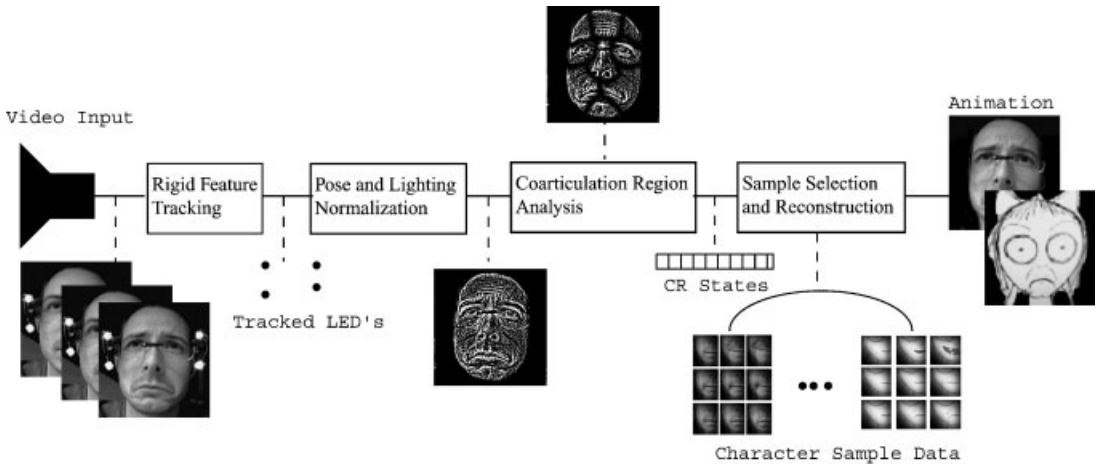

Figure 7. Online system diagram.

character database which are composited to generate a frame of the animation sequence.

Pose and Lighting Normalization

LED Tracker

By identifying at least three rigid points on the face and tracking them across adjacent frames, we can compute the affine transformation to a canonical pose and correct the image. The accuracy of this transformation is dependent on the accuracy of the feature tracking. Several generic feature-tracking methods exist;[23,24] however, our experiments using these methods to track features on the face showed large feature point drifts due to sensitivity to lighting conditions, rapid head motion, face deformation, and the low resolution and poor quality of acquired images. This problem is exacerbated by the fact that there really are no truly rigid points on the face; even the eye corners deform during squinting. (The nose tip is arguably rigid, but this is very hard to locate precisely.) To obtain a balance of accuracy, simplicity, computational efficiency, and subject freedom, we developed a pair of easily detectable and trackable eyeglass frames outfitted with a quadrature of infrared LEDs.

The four LEDs attached to the glasses appear as high-intensity spots in a grayscale image that are invariant to environmental lighting conditions. The image is thresholded and binarized, and a set of potential LED clusters is isolated in the image by collecting connected regions. The clusters created by the projection of the LEDs onto the image plane have certain properties that allow them to be reliably distinguished from the other cluster in the image.

- Each LED is covered with a spherical light diffuser and hence its cluster will be approximately circular.
- All four LED clusters will be approximately equal in area.
- Due to a limited working volume, the area of a cluster will be constrained.

The circularity of a cluster is described by its compactness descriptor.[25] If p is the cluster's perimeter and a is its area, the compactness c is

$$C = \frac{p^2}{a} \tag{1}$$

which is minimual for circular clusters. The area range is determined empirically. In practice for a working area of 1–3 feet from the camera, the LED projection lies within 0.13% and 0.8% of the image size. Using these simple heuristics the LEDs are automatically detected in most cases. In problematic cases (for example, if the actor is wearing a white spotted shirt) the LEDs can be manually identified in the first frame.

LED tracking can be performed by reapplication of the detection algorithm, but to reduce the computation time involved in recomputing all connected regions we opt for a simple prediction-based method. A new LED position is predicted using the current velocity estimate of its centroid. The four predicted locations are tested for the presence of clusters meeting the heuristic criteria above. If they are not met, the detection algorithm is reapplied.

Pose Normalization

Given the 2D pixel coordinates of each LED and a set of reference coordinates we model the transformation between two coordinate frames as a bilinear distortion. The transformed coordinates x' and y are given by

$$x' = c_1x + c_2y + c_3xy + c_4$$
$$y' = c_5x + c_6y + c_7xy + c_8 \quad (2)$$

Given the four pairs of corresponding LED positions, we have a set of eight linear equations that allows us to solve for the coefficients $(c_1 \ldots c_8)$. The warp equations are then applied to each pixel in the source image and the final normalized image is produced by bilinear interpolation. The bilinear model accurately corrects for head translation and rotation in the image plane but does not account for large out-of-plane rotations.

Lighting Normalization

In Donato *et al.*[7] lighting normalization was performed by subtracting each image from an image of the actor with a neutral expression. The difference images provided adequate normalization for data acquired in a single session. However, if the lighting conditions are changed with respect to the neutral image, errors will arise. Improved results can be achieved by convolving each image with a Laplacian kernel. Laplacian images are less sensitive to changes in lighting conditions and help compensate for small changes in physical appearance (limited facial hair growth, modest amounts of make-up, etc.) in different acting sessions that can otherwise cause problems during classification.

Co-articulation Region Analysis using ICA

Independent Component Analysis

Independent and principal component analysis are common techniques for modeling appearance variations in digital imagery. PCA derives an orthogonal basis from second-order statistics of the data (the covariance matrix) but ICA uses higher-order statistics to compute a potentially non-orthogonal basis that can better reflect the structure of the data. While ICA is not intrinsically better for classification, Donato *et al.*[7] have demonstrated empirically that features extracted from face images using ICA are better for classification of FACS action units than PCA. The no-noise case of ICA is expressed as

$$C = AS \quad (3)$$

where C is a matrix with the observed data vectors along the columns (often called the signals), S is a matrix of statistically independent data vectors (sources), and A is the mixing matrix. The observed signals are thus expressed as a linear combination of some unknown independent sources. Neither the source nor mixing matrices are known, yet with the assumption of statistically independent sources they can be estimated with various ICA algorithms.

Several ICA algorithms exist;[26, 27] however, most have a choice of non-linear transfer function as a free parameter. This parameter must be adjusted according to the distribution of the sources. This distribution is not always known (and in our case, is not) so we must choose the non-linearity parameters empirically. We chose the FastICA algorithm as it does not require such tuning; the default tanh non-linearity is sufficient.[26]

CR Analysis

The training data C_i for CR_i consists of the normalized sample matrix defined above: the ordered set of muscle appearance samples. We regard the independent sources as a set of basis vectors for CR appearance space. This follows the formulation used in Donato *et al.*[7] for full face analysis of FACS action units. These basis vectors resemble the eigenvectors computed in PCA, but exhibit more local structure due to their statistical independence. The matrix A supplies the coefficients of representation in the appearance basis for each training sample in C_i. A new sample x can be represented in CR signature space by solving for the coefficients c:

$$x = Sc \quad (4)$$

As this is, in practice, an over-determined set of linear equations, c is estimated by least squares where $\hat{c}$ minimizes the residual $r = S\hat{c} - x$. The *muscle signature* of x is the normalized coefficient vector

$$K^{ICA}(x) = \frac{\hat{c}}{||\hat{c}||} \quad (5)$$

Similarly, the muscle signatures ϕ^{ICA} of the training data samples are computed by normalizing the columns of A.

Each training sample is treated as its own class center and a new muscle signature is assigned to the class of the training sample with minimal Euclidean distance in ICA space.

$$g(x) = \min_j ||K^{ICA} - \phi_j^{ICA}|| \tag{6}$$

Because S is very large, iterative least squares methods are prohibitively expensive for realtime, online signature extraction. Instead, during training the SVD back substitution matrix M is computed which guarantees the least squares solution to equation (4). The SVD of non-square matrix S is given by

$$S = UDV^T \tag{7}$$

where D is a diagonal matrix consisting of positive or zero singular values of B, and U and V are orthogonal matrices. A pseudo-inverse of B that guarantees the least squares solution to equation (4) is given by $M = VD^{-1}U^T$. For each new sample vector x we compute $\hat{c} = Mx$. M is constant following the ICA basis computation and may therefore be assembled once, and stored for efficient online signature extraction.

Character Creation and Animation

As there is a direct mapping from a CR state to its visual appearance, a new character is defined by creating a neutral face frame and assigning an explicit reconstruction sample for each CR state. Samples can take the form of hand-drawn frames for the case of cartoon animation, or may be populated from the original training samples. If training samples are used, the region under each CR is used as a mask to extract reconstruction pixel data from each discrete CR state. Figure 8 shows a set of appearance samples assigned to CR 5 in the construction of a 2D character as well as the original samples used for video reconstruction.

For each input image frame, all nine CRs are analyzed, resulting in a 16-element face state vector (of which a maximum of nine will be non-zero). Each state vector applied to the character database defines a set of reconstruction elements and alpha masks for the current frame. Alpha masks are the same size as the reconstruction elements and have a value of 1.0 everywhere with a 10-pixel gradient to 0 approaching the boundary. Reconstruction elements are composited into the final image using alpha blending with the neutral image as a base. Animation is thereby performed by reconstructing frames using the stream of CR state vectors.

Results

Figure 9 shows results of the classification system applied to the video reconstruction set acquired from the same actor. Input images consisted of 360×240-pixel grayscale images of an actor exercising various expressions. Lighting conditions were kept constant throughout the performance. On a 1.7 GHz Pentium 4 CPU with an NVidia Quadro2, the classification and reconstruction is performed in real time (35 fps). All image samples are stored in the graphics card memory and composited using OpenGL texture mapping hardware. Note that

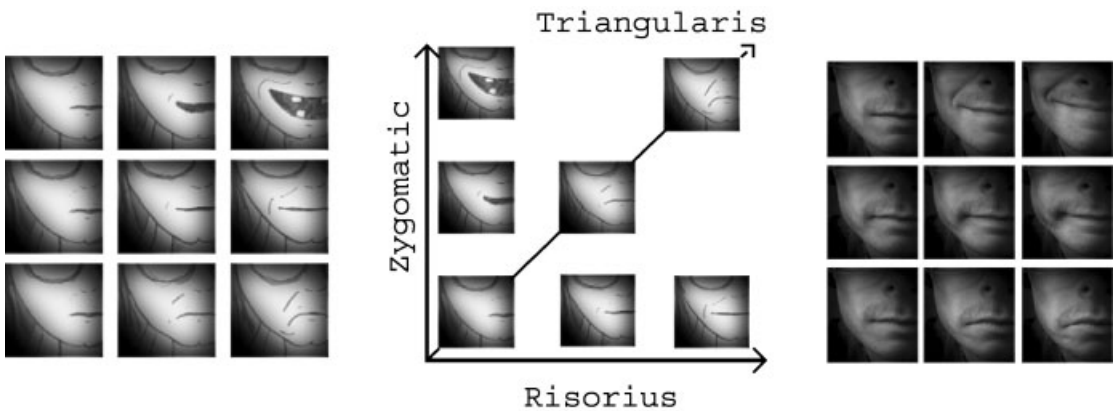

Figure 8. (Left) Reconstruction samples for CR5 of an animated character. (Right) Samples extracted from original input frames for video reconstruction. (Center) Locations of samples in CR parameter space.

Figure 9. (Top) Reconstructed video frames. (Bottom) Original video frames.

multiple CRs are concurrently analyzed and reconstructed. The system is able to accurately match the source CR states. In Figure 10 the same training set is used to animate a hand drawn character. The same input data are used to drive both sequences, demonstrating the ability to mix and match reconstruction databases.

Discussion

Local actuation regions proposed in FACS and others provide an intuitive partitioning of the space of facial motion. Unfortunately such approaches favor many small areas of change that become difficult to manage and interpret independently. Co-articulation region parameterization merges the benefits of these two representations. CR analysis by ICA proves to be efficient and robust.

Independence between the actor and controlled object is maintained by mapping through a single set of control parameters. Region-based flip-book style animation using the CoArt[28] system has been demonstrated. CoArt can alleviate the tedium of manual frame arrangement for 2D face animation by allowing an actor to control the gestures using his/her own face. Video footage of the actor is analyzed in real time and used to control expressive gestures. Analysis and synthesis are performed at the region level, and therefore allows for assembly of gestures that were not explicitly constructed by the animator. This can be especially beneficial for quick prototypes of scenes and can assist in creating more expressive storyboards. The ability to arrange such frames in real time enables other applications such as fully expressive remote 2D avatar control and, using real video samples to populate the frame database, non-causal sample-based video synthesis.

CR analysis was only demonstrated on 2D characters, but the CR/muscle data is equally applicable to the control of 3D faces with an animation system that can be parameterized by a set of facial gestures. For example, CR data can be used directly as contraction levels of muscles in a virtual muscle system or as interpolation parameters to control the blending of local morph targets.

In this work, gesture parameters have been analyzed and applied directly to control of animated characters. However, as gestures are selected to span expressive space, it is possible to perform a second-order analysis of the assembled gestures and inferfacial expression. However, the CoArt work quantizes the input space into a set of predefined gestures without accounting for the subtle variations that can exist within a given gesture class. This renders the CoArt system more suitable for cartoon type control where the desired semantics of the final animation can be similarly quantized.

Limitations

The analysis system presented has a few limitations. Ideally we would like to analyze a new incoming sample from a CR and determine its exact position in CR state space, thereby determining the contributions from each muscle basis vector. This turns out to be extremely difficult, as the resulting visual mixture is a non-linear combination of skin appearances. We simplify the problem by assuming a single active muscle in a CR at any point in time, but this does not account for the co-articulation ability of muscles in the CR. A possible solution is to treat the co-articulation gestures as pseudo-muscles by acquiring these samples in training and assign the bounded gesture frames to a new basis vector of the CR. However, this places extra

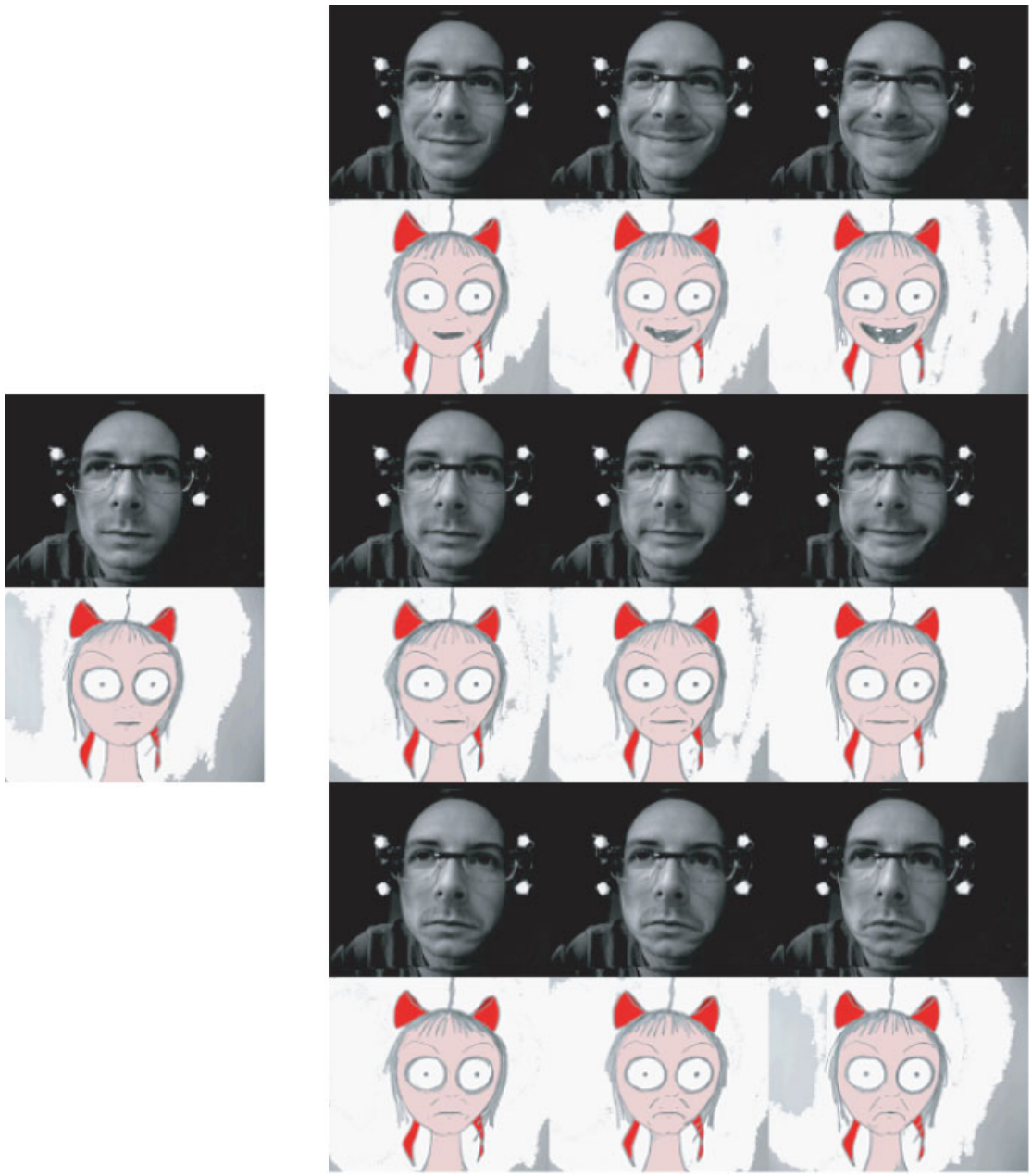

Figure 10. Animation results using gesture analysis to control a hand-drawn character.

burden on the actor during the training phase and limits classification of co-articulation effects to those seen during training.

The current gesture intensity model also assumes a linear relationship between the temporal position of a sample in a training gesture sequence and its intensity. Though the subjects are coached to generate a smooth continuous gesture from neutral to apex, as the facial muscles compress due to contraction the resistance of the muscle fibers to further compression increases. This breaks the linearity assumption as demonstrated in Essa.[1] For the animation system presented this is not a problem if the animator matches the hand-drawn samples to training samples, rather than to the gesture intensity value. However, this can present a problem if the intensity parameters are used directly.

The current decomposition of the face was specifically designed for analysis of expressive gestures. Speech is

arguably more complex as a large number of muscles are active in a relatively small area of the face. Many phonemes require the use of the orbicularis oris, masseter, temporalis, or medial pterygoid muscles, which are not as common in expressive gestures and as such were not included in the co-articulation region decomposition. Future work should include a finer decomposition of the mouth state space tuned specifically for speech or superimposing a separate speech analysis region that operates independent of the expression analyzer.

The CoArt flip-book system gives full flexibility to the animator as he or she hand-generates a visual representation of each gesture state. From an artistic point of view, this has some interesting implications as the synthetic gesture states need not correspond to those analyzed. One can envision a filter mechanism whereby the analyzed input gestures are modulated or mapped to a set of semantically different visual representations. This flexibility, however, comes with the burden of significant preparation time for each character.

ACKNOWLEDGEMENTS

Our thanks to J. P. Lewis and Kazunori Okada for their insight and many discussions. Funding and research facilities were provided by the NSF through its ERC funding of the Integrated Media Systems Center. Additional support was also provided by NCR.

References

1. Essa A. Analysis, interpretation, and synthesis of facial expressions. PhD thesis, Massachussetts Institute of Technology, 1995.
2. DeCarlo D, Metaxas D. Deformable model-based shape and motion analysis from images using motion residual error. In *ICCV Proceedings*, 1998; pp 113–119.
3. Choe BW, Ko HS. Analysis and synthesis of facial expressions with hand-generated muscle actuation basis. In *Proceedings of Computer Animation*, 2001; pp 12–19.
4. Ekman P, Friesen W. *Facial Action Coding System: A Technique for the Measurement of Facial Movement*. Consulting Psychologists Press: Palo Alto, CA, 1978.
5. Cohn JF, Zlochower AJ, Lien J, Wu Y, Kanade T. Automated face coding: a computer-vision based method of facial expression analysis. *Psychophysiology* 1999; **35**(1): 35–43.
6. Lien JJ, Kanade T, Cohn J, Li C. Subtly different facial expression recognition and expression intensity estimation. In *IEEE Conference on Computer Vison and Pattern Recognition*, 1998; pp 853–859.
7. Donato G, Bartlett M, Hager J, Ekman P, Sejnowski T. Classifying facial actions. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1999; **21**(10): 974–989.
8. Guenter B, Grimm C, Wood D, Malvar H, Pighin F. Making faces. In *SIGGRAPH 98 Proceedings*, 1998; pp 55–66.
9. Noh JY, Neumann U. Expression cloning. In *SIGGRAPH 2001 Proceedings*, 2001; pp 277–288.
10. Brand M. Voice puppetry. In *SIGGRAPH 99 Proceedings*, 1999; pp 21–28.
11. Lo HC, Chung R. Facial expression recognition approach for performance animation. In *Proceedings of the Second International Workshop on Digital and Computational Video*, 2001; pp 132–139.
12. Terzopoulos D, Waters K. Analysis of facial images using physical and anatomical models. In *International Conference on Computer Vision*, 1990; pp 727–732.
13. Eisert P, Girod B. Analyzing facial expressions for virtual conferencing. *IEEE CG&A* 1998; **18**(5): 70–78.
14. Bregler C, Covell M, Slaney M. Video rewrite: driving visual speech with audio. In *SIGGRAPH 97 Proceedings*, 1997; pp 353–360.
15. Cosatto E, Graf HP. Photo-realistic talking-heads from image samples. *IEEE Transactions on Multimedia* 2000; **2**(3): 152–163.
16. Vetter T, Poggio T. Linear object classes and image synthesis from a single example image. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1997; **19**(7): 733–742.
17. Beymer D, Shashua A, Poggio T. Example based image analysis and synthesis. MIT AI Memo 1431, November 1993.
18. Okada K, von der Malsburg C. Analysis and synthesis of human faces with pose variations by a parameterized piecewise linear subspace method. In *Proceedings of CVPR 2001*, December 2001; pp 761–768.
19. Reveret L, Essa I. Visual coding and tracking of speech related facial motion. Technical Report GIT–GVU–TR–01–16, GVU Center, 2001.
20. Welsh WJ, Shaw D. Facial feature image coding using principle components. *Electronics Letters* 1992; **28**: 2066–2067.
21. Theodoridis S, Koutroumbas K. *Pattern Recognition*. Academic Press: London, 1999.
22. Faigin G. *The Artists Complete Guide to Facial Expressions*. Watson-Guptill: New York, 1990.
23. Neumann U, You S. Integration of region tracking and optical flow for image motion estimation. In *Proceedings of the IEEE International Conference on Image Processing*, 1998.
24. Tomasi C, Kanade T. Detection and tracking of point features. Technical Report CMU–CS–91–132, Carnegie Mellon University, 1991.
25. Gonzalez R, Woods R. *Digital Image Processing*. Addison-Wesley: Reading, MA, 1993.
26. Hyvärinen A. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks* 1999; **10**(3): 626–634.
27. Bell A, Sejnowski T. An information maximisation approach to blind separation and blind deconvolution. *Neural Computation* 1995; **7**(6): 1129–1159.
28. Fidaleo D, Neumann U. Coart: coarticulation region analysis for control of 2d faces. In *Computer Animation 2000 Proceedings*, 2000; pp 17–22.

Authors' biographies:

Douglas Fidaleo is a Postgraduate Research Scientist at the University of California, San Diego. He earned a BA in Computer Science from Pomona College in 1996 with an award of distinction for his senior thesis Digital Empathy: An experiment in emotion driven computer music composition, an MSCS from the University of Southern California (USC) in 1998, and a PhD in Computer Science from USC in 2003. His current research focuses on facial gesture and affect analysis and synthesis with applications to performance driven facial animation. Other related interests include interactive art applications of graphics and vision technology. In 2000 he collaborated to develop the Comfort Control art installation using facial expression recognition and a computer controlled punishment/reward system to explore social responses to imagery.

Ulrich Neumann is the Charles Lee Powell Professor of Engineering and an Associate Professor of Computer Science at the University of Southern California. Dr. Neumann is also the Director of the Integrated Media Systems Center (IMSC), an NSF Engineering Research Center (ERC). He earned an MSEE from SUNY at Buffalo in 1980 and he completed his computer science Ph.D. at the University of North Carolina at Chapel Hill in 1993 where his focus was on parallel algorithms for interactive volume-visualization. His current research relates to immersive environments and virtual humans. He won an NSF CAREER award in 1995 and the Jr. Faculty Research award at USC in 1999. In his commercial career, he designed multiprocessor graphics and DSP systems, cofounded a video game corporation, and independently developed and licensed electronic products.