CLASSIFICATION AND VOLUME MORPHING FOR PERFORMANCE-DRIVEN FACIAL ANIMATION

Douglas Fidaleo, Jun-yong Noh, Taeyong Kim, Reyes Enciso, and Ulrich Neumann

University of Southern California
Los Angeles, CA

{fidaleo | noh | taeyongk | uneumann}@usc.edu, renciso@imsc.usc.edu

Abstract

We present a new facial animation approach that produces a wide variety of realistic facial expressions with advantages over existing geometry deformation and morphing methods for performance-driven animation systems. These systems sense and reconstruct a real person's facial expressions from an image sequence. The novelty of our approach is to sense and animate both textures and geometry using classification, volume morphing, and 3D-texture interpolations. Feature tracking drives our volume morphing animation system directly, without a need for intermediate animation parameters. Classification encodes realistic dynamic skin wrinkles, eye blinking, and eye motions. Classification results control 3D-texture animations that reconstruct the observed facial appearance. Classification leads to extremely low bandwidths for communicating visual information and independence of the rendered resolution and image quality from sensing resolution and pose. We demonstrate our approach with results obtained with a real-time volume morphing engine that animates the geometry of a facial model from sparse motion samples. A wavelet-based classifier and a 3D-texture engine reproduce dynamic textures on the animated geometry, including wrinkling effects, eye blinking, and eye movements.

I . INTRODUCTION

In typical facial animation, a user tunes large numbers of control parameters to achieve a desired expression. The complexity and subtlety of human expressions make it tedious to generate realistic expression sequences, and even more difficult to produce the look and dynamics of a specific individual's expressions. The performance-driven animation technique [1-5] addresses this problem by using live performances of an actor to drive a facial animation. Applications of performance-driven facial animation (PDFA) include character animation for entertainment and communication in shared virtual spaces.

Generally, animations are specified by a set of dynamic parameters. This poses two problems in PDFA. 1) The parameters should be extracted from the sensing data, a task that can be difficult. 2) In the model preparation stage, the adaptation of parameters to each individual person often requires extensive tuning.

In this paper, we present new approaches to the dynamic texture and animation parameter problems, including:

1) Volume morphing for geometry animation from tracking data (Section IV)
2) 3D-texture animation for reconstructing skin and eye appearance from classification data (Section V)
3) Classification for dynamically encoding the appearance of facial skin wrinkles, eye blinking, and eye motions (Section V)

Classification and 3D textures produce the subtle expression components arising from wrinkles or eye blinking that are hard to synthesize with only geometry deformations. Since wrinkles are hard to track[1], we use a wavelet decomposition classification technique. These three techniques, volume morphing, classification, and 3D texture rendering, also lead to very low bandwidth (<12Kb/s) communication for applications such as shared virtual-space telepresence.

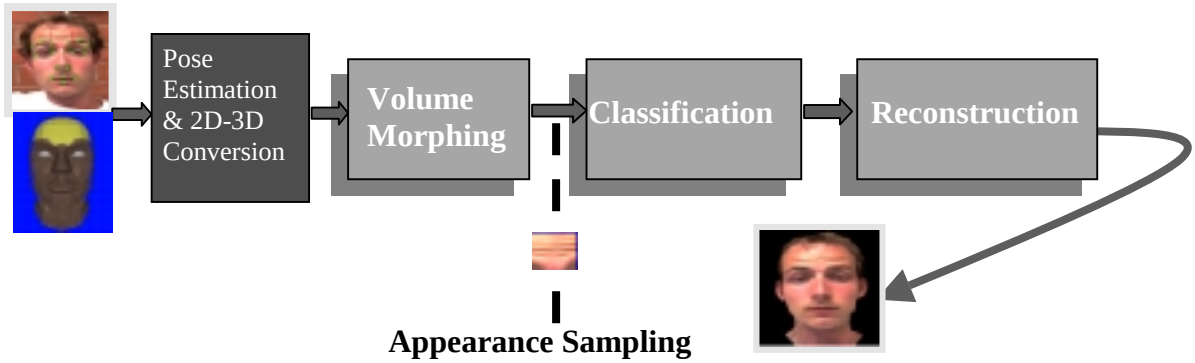

Figure 1. System Diagram. Input to the system is a video sequence of a person's face, and a prepared representative model of the individual. This paper focuses on stages highlighted in orange.

II. BACKGROUND

PDFA requires facial sensing, and computer vision research is extensive in this area [6, 7, 8, 10, 13, 30-32]. One approach uses whole-image optical flow measures [3, 4, 33]. These are currently too slow for use in a live performance. When live performance is important, as in virtual tele-conferencing, tracked feature points can provide sparse but real-time updates. We employ an existing commercial feature tracking system [11] that works without markers. It reports the 2D positions of up to 18 feature points (Fig.1a) including the eye corners, the lip corners, the nose tip, etc.

[1] Wrinkles, for example arise and disappear, so tracking them is futile.

There exists a large number of works in facial animation based on geometry deformation. Vector-based muscle models [14, 18] offer simple and compact representations, however no automatic means of placing muscles within a person-specific mesh have been reported. Other deformation methods include spline models [19, 20], and free-form deformations [21, 22]. See [9] for an excellent survey of these and other methods.

Skin wrinkles and creases can be modeled by physically-based mass-spring system [16, 17]. These methods produce geometry deformations with static skin texture. Tuning these to produce a particular person's skin behavior is difficult. Textures provide an alternative to geometry for representing skin appearance. Guenter et al. [37] represents texture variations as a compressed data stream. Pighin et al. [15] uses region painting to select blends of different morphing targets for different facial regions.

III. GENERIC MODEL PREPARATION AND POSE ESTIMATION

Our animation process begins with a prepared generic head model that has been fitted to a specific individual's mesh. The polygons of the generic mesh are manually tagged for their treatment by the texture and deformation engines.

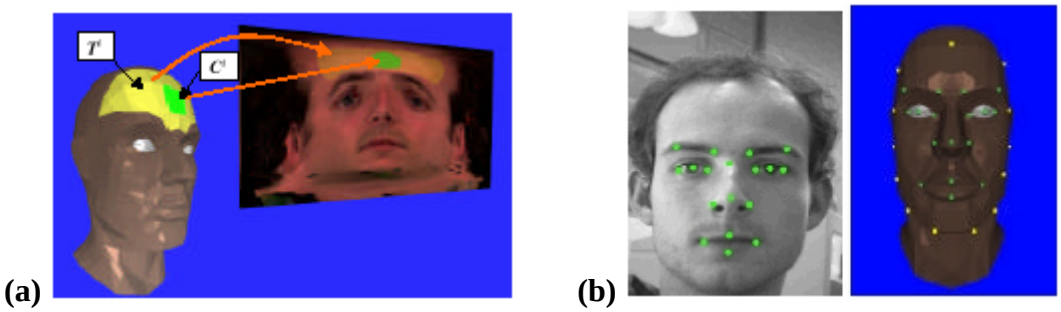

Figure 2. (a) Locations of texture regions and appearance samples on the 3D model, with their respective projections in the final texture image. (b) Correspondence between 2D tracked feature points and 3D features on the model.

Texture regions $\boldsymbol{T^i}$ are positioned on the model to facilitate dynamic 3D-texturing. These regions are where we animate the appearance of skin wrinkling, eyeball movements, and eye blinking. In principle, the entire face could be partitioned into such regions; however, we implement only a subset of regions corresponding to the forehead, eyes, temples, and mouth region. Within texture regions, we locate a small area for the templates $\boldsymbol{C^i}$ that are used in classification.

At the time of model construction, images are acquired while the subject makes faces, that is, exercises their facial expression capabilities. From these expressive images, we extract a set of $\boldsymbol{n}$ expressive texture samples $(\boldsymbol{t^i_0, t^i_1, \ldots, t^i_{n-1}}) \supset \boldsymbol{T^i}$ for region $\boldsymbol{i}$ to represent its various appearance states *i.e.*, wrinkle appearance, or eyelid positions (Fig. 2a). (Note that $\boldsymbol{n}$ can vary from facial region to region.) Texture regions provide the 3D-texture data

used to reconstruct facial appearances (as related in Section V). A texture region set $\boldsymbol{T}^i$ also has an associated texture classification template set $\boldsymbol{C}^i$ that contains small samples of texture regions for use in classification (Fig. 2a).

Our current implementation requires the person to remain in a similar pose for the stereo-image acquisition and expression-capturing step. This ensures good lighting consistency and texture sample similarity. (To correct for pose changes during the expressions, we could derive the head pose from tracking data.) A sample of the resulting prepared head models is shown in Fig. 1b and in the results (Fig. 8). Interested readers are referred to [38] for additional details.

The 2D tracking data must be converted to a 3D-motion field for volume morphing. Our approach follows the formulation in [10] with rigid points weighted higher than non-rigid points. After computing the model pose, the remaining disparities between projected 3D model points and 2D tracking points are taken as the non-rigid 2D motions caused by expressions. We project this residual 2D motion field onto the surface of the 3D model to reconstruct the 3D positions of the tracking data in each new frame.

Noisy or missing feature points are approximated by heuristics. Missing feature points decay slowly toward their neutral expression locations. Facial expressions are assumed symmetric. Feature motion is held from a previous frame whenever it changes abruptly or extremely. The resulting 3D point sets are fed into our volume morphing geometry-deformation engine.

IV. GEOMETRY DEFORMATION: VOLUME MORPHING WITH RBF's

Classical interpolation, approximation, standard regularization, and hyper-surface reconstruction theory [27] support volume morphing with RBFs for reconstructing a facial surface. Advantages include its robustness against noisy data and its efficiency. Real time operation is feasible since the computation requires the inversion of only a relatively small matrix (~ 25× 25).

We compute the coefficients of the RBF that represent the best mapping between source and target feature points in each frame. (In our case, the source points come from the neutral position model, and the dynamic target points are the 3D tracked feature points computed in the previous section.) The rest of the nodes in the facial mesh are transformed using these coefficients. The RBF approximation is of the form

$$F(x) = \sum_{i=1}^{M} c_i h(\| x - x_i \|) \tag{1}$$

When Hardy [28] multi-quadrics $h(r) = \sqrt{r^2 + s^2}$ is used, (1) becomes

$$\vec{x}^{target}_j = F(\vec{x}^{source}_j) = \sum_{i=1}^{M} c_i \sqrt{\| \vec{x}^{source}_j - \vec{x}_i \|^2 + s_i^2} \tag{2}$$

where s is called a stiffness constant that regulates the local and global effects of the feature points, r is the Euclidean distance between a feature point and an input point, M is the number of feature points, and $1 \leq j \leq M$. The dimension of $\vec{x}$ is 3 (*i.e.*, the dimension of feature point positions). The stiffness coefficient s_i is suggested by Eck [29] for softer deformation where data points are widely scattered and stronger deformation for close points.

$$s_i = \min_{j \neq i} \| \vec{x}^{source}{}_j - \vec{x}_i \| \tag{3}$$

Substituting M feature points into equation (2) results in a linear system whose solution is of the form

$$\vec{c} = H^{-1} \vec{x}^{t\,\mathrm{arg}\,et} \tag{4}$$

The solutions of equation (4) assume no spurious data, however, tracking data is noisy. We use the method of Thikhonov and Arsenin [39] to replace the matrix H by $(H + \lambda I)$ in equation (4)

$$\vec{c} = (H + \lambda I)^{-1} \vec{x}^{t\,\mathrm{arg}\,et} \tag{5}$$

where λ is a "small" parameter (we set to 0.01), proportional to the noise. The system is solved by LU decomposition to obtain the coefficient set $\vec{c}$. The decomposition is done only once at initialization. A back-substitution of the motion vectors $\vec{x}^{t\,\mathrm{arg}\,et}$ occurs at each frame, resulting in a real time computation (30 Hz.). Figure 3 shows sample synthesized images.

Figure 3. Top shows deformation generated by muscle model with eight feature points. Bottom row shows synthesized images by volume morphing. Note how closely RBF based volume morphing can synthesize the original shape with a small number of 3D points.

We compared a Gaussian RBF to the Hardy RBF. The Gaussian better approximates the overall face shape with noisy data, but it is less attractive for facial expression synthesis because subtle expressions are over-smoothed. To control the volume morph, we specify the 18 tracked feature points as well as 13 *phantom points* located around the face boundary (figure 2b). The phantom points are used to constrain the volume morphing deformations around the edge of the face to ensure rigid head motion.

V. TEXTURE ANIMATION

Classification, for our purpose, is defined as a mapping from an image appearance sample $\boldsymbol{a^i}$ to a scalar Ω^j that encodes the similarity of the sample to a set of discrete templates $\boldsymbol{C^i}$. Formally, ***classification:*** $\boldsymbol{a^i \rightarrow \Omega^j \mid (0 \geq \Omega^j \geq k\text{-}1)}$, where $\boldsymbol{k}$ is the number of discrete templates $\boldsymbol{\{c^i_0, c^i_1, \ldots, c^i_{k-1}\} \supset C^i}$ for region $\boldsymbol{i}$.

During a PDFA session, appearance samples $\boldsymbol{a^i}$, in facial region $\boldsymbol{i}$ of the input image are classified based on their similarities to the region's classification templates $\boldsymbol{C^i}$. The classification result Ω^j indicates which template(s) are most similar to the sample, so Ω^j also specifies what texture regions $\boldsymbol{T^i}$ will best reconstruct the input region's appearance on the animated model.

This process involves three major steps:

1) Locate the appearance samples $\boldsymbol{a^i}$ in the input image.
2) Classify the sample of a region based on the templates $\boldsymbol{C^i}$ available for the region.
3) Render the animation model with the texture regions $\boldsymbol{T^i}$ associated with the classification results in each region.

V.i. LOCATING APPEARANCE SAMPLES

Appearance samples $\boldsymbol{a^i}$ measure areas of the face that change their appearance dynamically, *e.g.*, wrinkle patches and eyes. Since these areas do not track well with feature tracking or optical flow methods, we locate appearance samples in the input images by projecting static appearance sample positions in the texture map, rather than trying to rack their positions.

During off-line texture map and model preparation we determine each template pixel's 3D position ρ on the model as a weighted-combination of their triangle-vertex v_i positions: $\rho = \mathbf{a}\, v_1 + \mathbf{b}\, v_s + \mathbf{c}\, v_3$. The same (**a, b, c**) coefficients are applied to the vertex positions after geometry animation to locate the template pixels on the input image. These projected positions are easily sampled by bilinear interpolation of the input image to create an appearance sample $\boldsymbol{a^i}$ for classification. This projection is fast to compute and normalizes appearance samples against head pose variations.

V.ii. CLASSIFICATION

Many pattern recognition methods perform classification, *e.g.*, eigenvectors, principle components [13, 31], and Gabor analysis [6, 30]. We adapt a simple Haar basis wavelet decomposition to classify the appearance states. We claim no originality with this wavelet classification method, however the application and adaptation of classification to facial texture animation is unique and an important concept. Our system is similar to that of Jacobs [36], used for image query.

Wavelet representations provide an effective way to represent and extract important structure information from images [40]. Wavelet transformations use a family of functions localized in *both* the spatial and frequency domain to decompose (or transform) a signal. The functions are obtained from a single prototype by translations and dilations:

$$\Psi_{s,x}(t) = \frac{1}{\sqrt{s}} \Psi(\frac{t-x}{s}) \tag{6}$$

where s and x represent dilation and translation, respectively. For an analyzing wavelet, the admissibility condition must be met on $\Psi(t)$:

$$\int_0^\infty \frac{\left|\hat{\Psi}(\omega)\right|^2}{|\omega|} d\omega < +\infty \tag{7}$$

where $\hat{\Psi}(\omega)$ is the Fourier transformation of $\Psi(t)$. The condition ensures that the wavelet transformation is a bounded and invertable operator in the appropriate spaces. The wavelet transformation of a function $f(t)$ is defined by

$$Wf(s, x) = \int_{-\infty}^{+\infty} f(t)\Psi_{s,x}(t)dt \tag{8}$$

For fast and robust matching, we use an orthonormal wavelet decomposition [35] for templates and appearance samples. The largest 50 coefficients of each 16×16 template are stored for similarity matching. Match scores φ are the summation of weighted coefficient differences.

$$\varphi = \sum_{j=1}^{50} \eta_w \left| c_j - a_j \right| \tag{9}$$

where c_j, and a_j, are corresponding template and sample coefficients, and the weighting factors η_w are functions of scale that were selected experimentally. (From the high-scale top of the pyramid to the bottom, the weights are 2, 4, 2, 1, 0.) We use the actual coefficient magnitudes directly rather than quantized values as in [36]. The decomposition and match functions are executed j and j*k times per frame, respectively. In our system j=7 and k=3. For these and similar values the classification system runs in real time.

The best template match produces an output index ($\gamma = 1, 2, \ldots p\text{-}1$) and score φ. The index identifies the matched template, and the score is the scalar match-distance. As

shown in Fig. 6, five diagonally-offset samples are actually created for each classification. The matching scores of the five samples shown in Fig. 6 indicate the best match for template-five with the lower-right sample. This winning sample is kept and the others are discarded.

To compensate for quantization effects caused by discrete templates, we also compute the scores for adjacent ($\gamma+1$ and $\gamma-1$) templates. The ($\gamma.\varphi$) tuples facilitate peak interpolation for the classification result Ω^i = **[*0, p-1*]**.

V.iii. TEXTURE RECONSTRUCTION

Classification results in the "appearance-field" discussed at the start of Section V. Classification determines which templates best represent the input image appearance samples. A correspondence between templates and texture regions is known from the model building phase. In fact, in most cases, templates are extracted from texture regions, although a 1:1 mapping is not necessary. We render the model, applying textures independently in each region, to reconstruct the appearance of the input image.

Real-time animation and smooth interpolation between texture templates is facilitated by the 3D-texture capabilities in the SGI Onyx2 architecture. The texture regions T^i are layered into 3D texture volumes. A ten-pixel linear alpha mask blends each texture region boundary with the neutral expression texture. The edge-blended texture volume is indexed by Ω^i to reproduce continuous appearance variations. Trilinear 3D-texture resampling blends adjacent layers for smooth transitions between layers. We find that three-layer textures are the minimum needed for realistic wrinkle or blink textures. Five or six layers provide perceptually smooth animations.

Eye classification is more complex since eyeballs move and blink independently. We therefore separate the eyelid texture from the eyeball motion texture to maintain independence.

VI. CONCLUSION

Classification leads to very low and fixed bandwidth for communicating the dynamic appearances (<12Kb/s.). Note that half of the tracking points, 9 out of 18, serve only for pose estimation rather than facial motions. The quality of the rendered image is completely independent of the bandwidth. Rendered mage quality is set by the texture size (1024×512 in our system) and the resolution of the images acquired in the modeling process. In principle, the texture and rendering resolutions are limited only by the rendering hardware.

Volume morphing has shown to have advantages for use with feature tracking data. It facilities the simple preparation of models for a given set of tracking features and requires no abstract animation parameters. Classification was shown to encode a wide range of dynamic facial region appearances that can be reconstructed by 3D-texture rendering. With classification, the system communicates less than 12 Kbits/sec between the sensing and rendering components at 30 Hz sensing and rendering rates. In addition, the 3D-

textures can be of arbitrary resolution, making the output image quality completely independent of communication bandwidth and sensing camera resolution.

VII. RESULTS

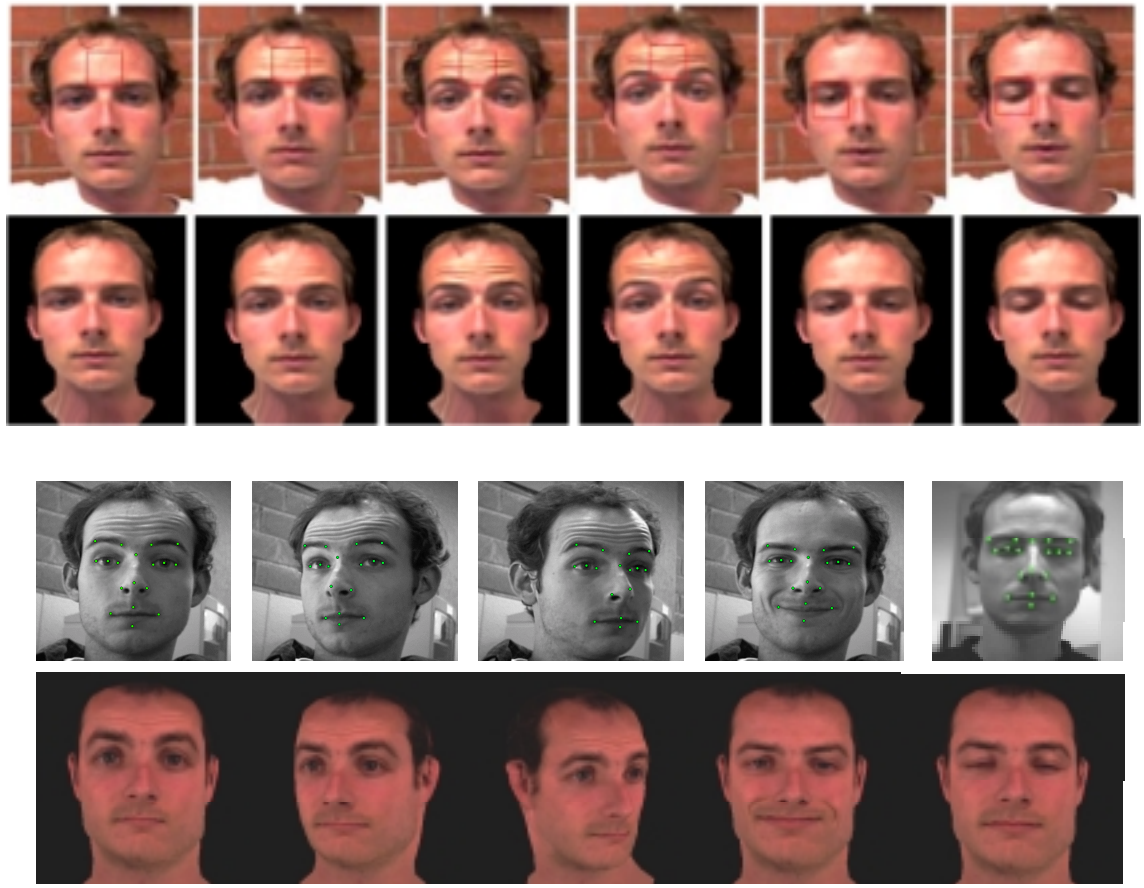

Figure 4) (top) Results of classification and volume morphing system with texture data acquired from video sequence. (bottom) Results applied to model with different facial characteristics.

VIII. ACKNOWLEDGEMENTS

This work is partially funded by the Annenberg School of Communication at USC. We also thank Intel, and the Integrated Media Systems Center for their support.

IX. REFERENCES

[1] L. Williams, Performance driven facial animation, Proc. Siggraph, 1990, pp. 235-242

[2] D. Terzopoulos, K. Waters, Analysis and Synthesis of Facial Image Sequences Using Physical and Anatomical Models, IEEE PAMI, 1993, 15(6), pp. 569-579

[3] H. Li, P. Roivainen, R.Forchheimer, 3-D Motion Estimation in Model-Based Facial Image Coding, IEEE PAMI, 1993, 15(6), pp. 545-555

[4] P. Eisert, B. Girod, Analyzing Facial Expressions for Virtual Conferencing, IEEE CG & A, 1998, 18(5), pp. 70-78

[5] M. Esher, N. M. Thalmann, Automatic 3D Cloning and Real-Time Animation of a Human Face, Proc. IEEE Computer Animation, 1997

[6] Facial Feature Detection By Saccadic Exploration of TheGabor Decomposition, Proc. IEEE ICIP, 1998

[7] I. A. Essa, S. Basu, T. Darrell, A. Pentland, Modeling, Tracking and Interactive Animation of Faces and Heads using Input from Video, IEEE Proc. Computer Animation, 1996

[8] I. A. Essa, T. Darrell, A. Pentland, Tracking Facial Motion, Proceedings of the IEEE Workshop on Non-rigid and Articulate Motion, November, 1994

[9] F. Parke, K. Waters, Computer Facial Animation, A K Peters LTD, 1996

[10] A. Azarbayejani, T. Starner, B. Horowitz, A. Pentland, Visually Controlled Graphics, IEEE PAMI 15(6), June 1993

[11] Eyematic Interfaces, Inc., http://www.eyematic.com

[12] UCSC Perceptual Science Laboratory, http://mambo.ucsc.edu/psl/lipr.html

[13] S. Basu, N. Oliver, A. Pentland, 3D Modeling and Tracking of Human Lip Motions, Proc. ICCV, 1998, pp. 337-343

[14] K. Waters, J. Frisbie, A Coordinated Muscle Model for Speech Animation, Graphics Interface, 1995, pp. 163 – 170

[15] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, D. H. Salesin, Synthesizing Realistic Facial Expressions from Photographs, Siggraph, 1998, pp. 75-84

[16] Y. C. Lee, D. Terzopoulos, K. Waters. Realistic face modeling for animation. Proc. Siggraph, 1995, pp. 55-62

[17] Y. Wu, N. M. Thalmann, D. Thalmann, A Plastic-Visco-Elastic Model for Wrinkles in Facial Animation and Skin Aging, Proc. 2nd Pacific Conference on Computer Graphics and Applications, Pacific Graphics, 1994

[18] K. Waters. A muscle model for animating three-dimensional facial expression. Proc. Siggraph,1987, pp. 17-24

[19] M. Nahas, H. Hutric, M. Rioux, and J. Domey, Facial image synthesis using skin texture recording. The Visual Computer, 1990, vol. 6(6) pp. 337 – 343

[20] M. Nahas, H. Huitric, and M. Saintourens, Animation of a B-spline figure, The Visual Computer, 1988, vol. 3(5), pp. 272-276

[21] M. L. Viad and H. Yahia, Facial animation with wrinkles, Proc. Third Eurographics Workshop on Animation and Simulation, 1992

[22] P. Kalra, A. Mangili, N. M. Thalmann, D. Thalmann, Simulation of Facial Muscle Actions Based on Rational Free From Deformations, Proc. Eurographics, 1992, pp. 59–69

[23] Constructing a Realistic Head Animation Mesh for a Specific Person, Technical Report (Authors and affiliation omitted for anonymity. We will supply copy upon request from paper committee.)

[24] J. Batista, H. Araujo, A. Almeida, Pose View Stability Analysis for Camera Look Angles Computation.

[25] R. G. Brown, Introduction to Random Signal Analysis and Kalman Filtering, John Wiley & Sons, New York, 1983

[26] A. Lerios, C. D. Garfinkle, M. Levoy, Feature-Based Volume Metamorphosis, Proc. Siggraph,1995

[27] M. J. D. Powell, Radial basis functions for multivariate interpolation: a review. Algorithms for Approximation, Clarendon Press, Oxford, 1987

[28] R.L. Hardy, Multiquadric Equations of Topography and Other Irregular Surfaces, J. of Geophysical Research, Vol. 76(8), 1971, pp. 1905 – 1915

[29] M. Eck, Interpolation Methods for Reconstruction of 3D Surfaces from Sequences of Planar Slices, CAD und Computergraphik, Vol. 13(5), 1991, pp. 109 – 120

[30] T. Maurer, C. Malsburg, "Tracking and Learning Graphs and Pose in Image Sequences of Faces," ICAFGR'96 pp.242-247.
[31] B.J. Frey, A. Colmenarez, T. Huang, "Mixtures of Local Linear Subspaces for Face Recognition," CVPR'98, pp.32-37.
[32] U. Neumann, S. You, "Integration of Region Tracking and Optical Flow for Image Motion Estimation, ICIP97.
[33] D. DeCarlo, D. Metaxas, "The Integration of Optical Flow and Deformable Models with Applications to Human Face Shape and Motion Estimation," CVPR'96, pp.231-238.
[34] F. Ulgen, A Step Toward Universal Facial Animation via Volume Morphing, 6th IEEE International Workshop on Robot and Human communication, 1997, pp. 358-363.
[35] S. Mallat, "A Theory for Multiresolution Signal Decomposition: the Wavelet Representation," IEEE PAMI,1983,11,pp. 674-693
[36] Charles E. Jacobs, Adam Finkelstein and David H. Salesin, "Fast Multiresolution Image Querying", Siggraph 1995, pp. 277-286.
[37] B. Guenter, C. Grimm, D. Wood, H. Malvar, F. Pighin, Making Faces, Proc. Siggraph, 1998, pp. 55 – 66
[38] U. Neumann, J. Li, R. Enciso, J.Y. Noh, D. Fidaleo, T.Y. Kim, Constructing a Realistic Head Animation Mesh for a Specific Person. USC-Computer Science TECH REPORT 99-691. 1999.
[39] A.N. Tikhonov, V.Y. Arsenin, Solution of Ill-Posed Problems and the regularization method. Soviet Math. Dokl., 4:1035-1038, 1963.
[40] S. Mallat, W.L. Hwang, "Singularity Detection and Processing with Wavelets," IEEE Transactions on Information Theory," 32(2), 1992.