Facial Expression Representation Using A Quadratic Deformation Model

[1]M. Obaid, [2]R. Mukundan, [1]M. Billinghurst
[1]HITLab NZ, [2]Computer Science and Software Engineering Department
University of Canterbury
Christchurch, New Zealand
{mohammad.obaid, mark.billingurst}@hitlabnz.org, mukund@cosc.canterbury.ac.nz

M. Sagar
Weta Digital
Wellington, New Zealand
marks@wetafx.co.nz

Abstract - In this paper we propose a novel approach for representing facial expressions based on a quadratic deformation model applied to muscle regions. The non-linear nature of muscle deformations can be captured for each expression, by subdividing the face into 16 facial regions and using the most general rubber-sheet transformation of second degree.

The deformation parameters are derived using a least-square minimization technique, and used to construct a Facial Deformation Table (FDT) to mathematically represent each expression. The generalized nature of the transformations allows us to easily map expressions from one model to another, and employ the FDTs in facial expression applications such as facial recognition and animations. The paper presents experimental results using the smile expression.

Keywords --- Rubber Sheet Transformations, Facial Expression Representations, Quadratic Deformation Model.

I. Introduction

Recently, with the exponential growth in the power of computer graphics hardware, facial animation has become increasingly important in various application areas such as games, animation, teleconferencing and multimedia education. The main application areas for facial animation are in the film industry, gaming, and advertising, where the focus is on the modeling and animation of Virtual Humans and 3D-digital characters. In this context, facial animation has proven to be one of the most challenging parts of building and animating a virtual character model. The complexity of the human face and the high sensitivity humans have in identifying facial expressions make the representation of facial expressions complicated and challenging to produce.

This paper presents a novel approach to representing facial expressions in terms of mathematical transformation functions. The main advantage of our approach is the generic representation of facial expressions that can be employed in facial expression applications such as facial animations and recognition.

This paper is organized as follows. Section II outlines the previous research and background in the area of facial expression representations. Section III presents the rubber-sheet transformations. Section IV explains how the proposed approach uses the rubber-sheet transformation for representing facial expressions. Section V presents the analysis and results for constructing the Facial Deformation Tables for facial expression representations. Finally, section VI concludes the paper and outlines some future research directions.

II. Background

Several earlier approaches have attempted to analyze and represent facial expressions. These include, image-based approaches (Gabor wavelets, principle component analysis and neural networks), model-based approaches (active appearance models, 3D geometric face models), and motion-based approaches (dense optical flow, 3D motion models). Full survey details are given in [6] and [9] .

One of the most significant descriptions of facial expressions is the Facial Action Coding System (FACS), developed by Paul Ekman and Wallice Friesden [2] . This defines facial expressions as a combination of facial actions corresponding to movements of particular muscle groups. Ekman and Friesden defined facial muscle actions as *Action Units* (AU). Originally, FACS used forty four AUs to define all facial muscular actions. Recently, a new version of FACS was published by Ekman, Friesen, and Hager [3] where facial expressions are categorized into seventy two AUs. One of the limitations of FACS is that the facial expressions' motions are described based on local information, making the definition of facial expression a difficult process for facial expression applications.

Essa and Pentland [4] described an extension to FACS called FACS++, which addresses the lack of emotional spatiotemporal information by using vision-based observations techniques with the dynamics of facial expressions. They used an optical flow method to estimate muscle actuation and to generate flow on the face model.

Matsumura, Nakamura, and Matsui [8] described a method to extract a mathematical function which can be used to reconstruct facial expressions from face portraits. Their method extracts the facial function from a 2D face image by using the metamorphosis by thin-plate splines.

Another common description of facial expression is the MPEG-4 standard [7] which supports the definition, encoding, transmission, of facial animation. The facial features specified by the MPEG-4 standard, are a representation of the human facial structure in a way that the facial expressions allow the recognition of the speaker's mood, and also the support of speech reproduction. The MPEG-4 facial animation standard is defined by 84 feature points (FPs). These points are used to define the 68 Facial Animation Parameters (FAPs). The MPEG-4 standard also defines Face Animation Parameter Units (FAPU), which define the distance ratios between key facial feature points. A detailed description about MPEG-4 Facial animation is in [7] .

One limitation with most facial representation standards, such as FACS and MPEG-4, is their very general description of muscle movements as shown in Fig 1. Our approach attempts to represent each expression as a collection of transform coefficients that are model independent.

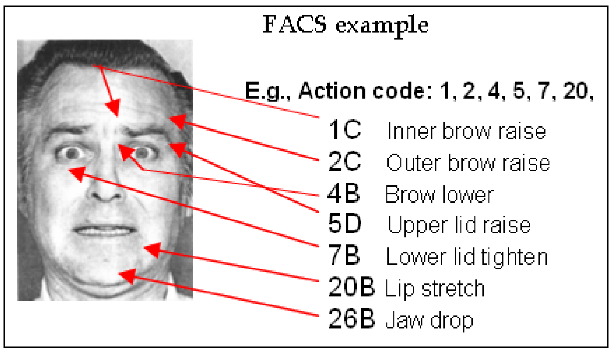

Figure 1. An example of the FACS description for the fear expression [17] .

III. RUBBER-SHEET TRANSFORMATION

Rubber-sheet transformations are higher-order (non-linear) polynomial transformations [13] [14] . The name comes from the logical analogy of overlaying an elastic piece of rubber to fit over a surface of some shape. Geographical Information Systems (GIS) extensively use rubber-sheet transformations for operations such as converting map coordinates, map alignments and geometrical correction between maps [10] .

In the two-dimensional space, rubber-sheet transformations are defined by a pair of equations:

$$x_i^{'} = a_1 x_i^2 + a_2 x_i\, y_i + a_3 y_i^2 + a_4 x_i + a_5 y_i + a_6 \quad (1)$$

$$y_i^{'} = b_1 x_i^2 + b_2 x_i\, y_i + b_3 y_i^2 + b_4 x_i + b_5 y_i + b_6 \quad (2)$$

$$i = 1,\ldots,n$$

Where n is the number of transformed points and a_i, b_i are the transformation parameters.

Generally, the above 12 transformation parameters are not known, but the coordinate points before and after the transformation are known (i.e. ($x_i^{'} y_i^{'}$) and ($x_i\, y_i$) from equations (1) and (2)) as shown in Fig. 2. With enough coordinate points, the *solution* for the 12 parameters that best fit the transformation can be worked out by solving from the known coordinate points. The process of finding the solution is described in the following sections.

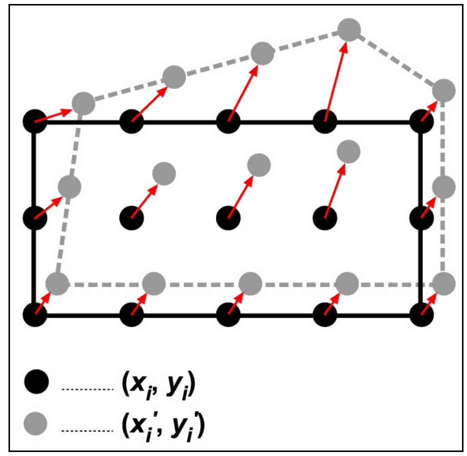

Figure 2. Deformation of a set of points.

A. *Transformation parameters*

To find the 12 parameters representing a transformation, the minimum of the sum of squares (or least square minimization) [13] is carried out on the pair of equations (Equation (1) and (2)) representing the x and y rubber-sheet transformations. The process of finding the transformation parameters is illustrated in Fig. 3.

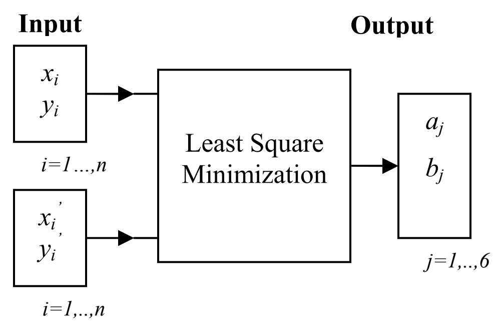

Figure 3. Derivation of deformation parameters.

For example, the minimum of the sum of squares is achieved when the gradient of the sum-of-error-squares for equations (1) and (2) is zero. The following equations represent the sum-of-error-squares for equations (1) and (2) respectively.

$$G_x = \sum \left[x_i^{'} - (a_1 x_i^2 + a_2 x_i\, y_i + a_3 y_i^2 + a_4 x_i + a_5 y_i + a_6) \right]^2$$

$$G_y = \sum \left[y_i^{'} - (b_1 x_i^2 + b_2 x_i\, y_i + b_3 y_i^2 + b_4 x_i + b_5 y_i + b_6) \right]^2$$

The minimum of the sum of squares is found by setting the gradient of G_x and G_y to zero with respect to all unknown variables as follows

$$\frac{\partial G_x}{\partial a_j} = 0\,; \frac{\partial G_y}{\partial b_j} = 0 \qquad j = 1,\ldots 6.$$

Differentiating G_x and G_y with respect to all unknown variables and equating to zero gives the conditions for the minimum error in the transform coefficients in the least-square sense. For example, differentiating G with respect to a_1 $(\frac{\partial G_x}{\partial a_1} = 0)$ is carried out as follows

$$2\sum_{i=1}^{n}\Big[x_i^{'} - (a_1 x_i^2 + a_2 x_i\, y_i + a_3 y_i^2 + a_4 x_i + a_5 y_i + a_6\Big]x_i^2 = 0$$

$$\sum_{i=1}^{n}\Big[x_i^{'}x_i^2 - a_1 x_i^4 - a_2 x_i^3 y_i - a_3 x_i^2 y_i^2 + a_4 x_i^3 - a_5 x_i^2 y_i - a_6 x_i^2\Big] = 0$$

$$\sum_{i=1}^{n} x_i^{'}x_i^2 - \sum_{i=1}^{n} a_1 x_i^4 - \sum_{i=1}^{n} a_2 x_i^3 y_i - \sum_{i=1}^{n} a_3 x_i^2 y_i^2 - \sum_{i=1}^{n} a_4 x_i^3 - \sum_{i=1}^{n} a_5 x_i^2 y_i - \sum_{i=1}^{n} a_6 x_i^2 = 0$$

This can also be written as

$$x_i^{'} n_1 - a_1 m_{11} - a_2 m_{12} - a_3 m_{13} + a_4 m_{14} - a_5 m_{15} - a_6 m_{16} = 0$$

Where

$$n_1 = \sum_{i=1}^{n} x_i^{'} x_i^2 \; ; \quad m_{11} = \sum_{i=1}^{n} a_1 x_i^4 \; ; \ldots ; \; m_{15} = \sum_{i=1}^{n} a_6 x_i^2$$

For the derivative of G with respect to the other unknown variable, the same process is applied to obtain the values for n_i and m_{ij}, where $i = 1,\ldots 6;\ j = 1,\ldots 6$.

Having found those values we can now solve for the parameter values $P_a(a_1,\ldots,a_6)$ and $P_b(b_1,\ldots,b_6)$ by solving the following Matrix setup

$$P\,M = N\,,\ P = M^{-1}N$$

To illustrate, the matrix elements to solve for the parameter values $P_a(a_1,\ldots,a_6)$ are as follows

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} & m_{15} & m_{16} \\ m_{21} & m_{22} & m_{23} & m_{42} & m_{25} & m_{26} \\ m_{31} & m_{32} & m_{33} & m_{43} & m_{35} & m_{36} \\ m_{41} & m_{42} & m_{43} & m_{44} & m_{45} & m_{46} \\ m_{51} & m_{52} & m_{53} & m_{54} & m_{55} & m_{56} \\ m_{61} & m_{62} & m_{63} & m_{64} & m_{65} & m_{66} \end{bmatrix}^{-1} \begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \\ n_5 \\ n_6 \end{bmatrix}$$

The solution to the set of parameters can now be used to obtain the transformed coordinate $(x_i^{'}, y_i^{'})$ of any point by substituting the $(x_i\,, y_i)$ coordinates into equations (1) and (2).

In the following two sections we show how the rubber-sheet transformation parameters can be used to represent facial expressions.

IV. Method

The method for representing facial expressions mathematically consists of two steps: (1) defining the facial regions, and (2) facial expression representation. At the first step, the aim is to divide the face into regions based on the muscular anatomy of the face. We define facial regions based on individual muscles or a small group of muscles that elastically deform the facial skin shape. At the facial expression representation step, we use the rubber-sheet transformation method (described in section III) to the represent each of the defined facial regions in terms of deformation parameters. To find the those parameters, face markers were captured and tracked for the six main expressions (happy, sad, fear, surprise, anger and disgust) to obtain a set of coordinate data for each of the facial regions. The data is normalized before applying the rubber-sheet transformation method to find the deformation parameters. The following two sections describe in more detail the process for the steps used to represent facial expressions mathematically.

A. Defining Faical Regions

To define facial regions that can be used to represent facial expressions mathematically, we looked at the FACS description of facial expressions [2] . FACS defines facial expressions based on an anatomical analysis of facial behavior and on all visually distinguishable facial movements, in which every facial movement is a result of a muscular movement [16] .

Using the FACS definition and the anatomy of the facial muscle system from [1] , we defined sixteen facial regions that represent the deformation of the facial expressions. The sixteen regions form the minimum number for defining independent facial muscle groups. Fig. 4 illustrates the defined facial regions.

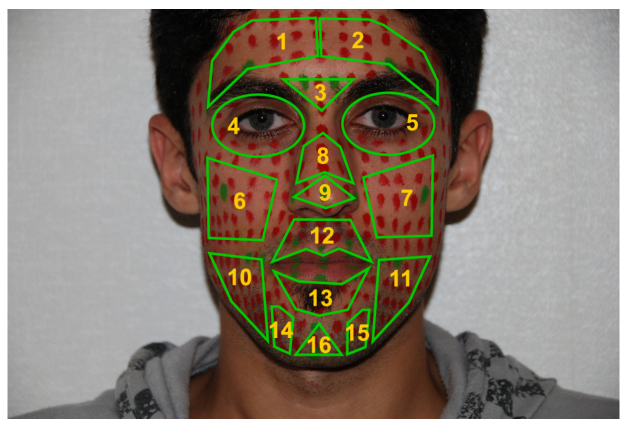

Figure 4. Facial Regions.

B. Facial Expression Representations

In this section, we describe how we represent facial expressions using the rubber-sheet transformations and the defined facial regions from sections III and IV(A) respectively. The main question we are answering is: "How does a region deform for a particular expression?" (Fig. 5). To answer this question, the defined facial regions are represented as non-linear shapes that can deform into any other shape depending on the facial expression. To find how each of regions deforms, several face markers were placed on an actors face and used to track how each region was deformed from the neutral facial expression to any performed expression (happy, sad, fear, surprise, anger and disgust).

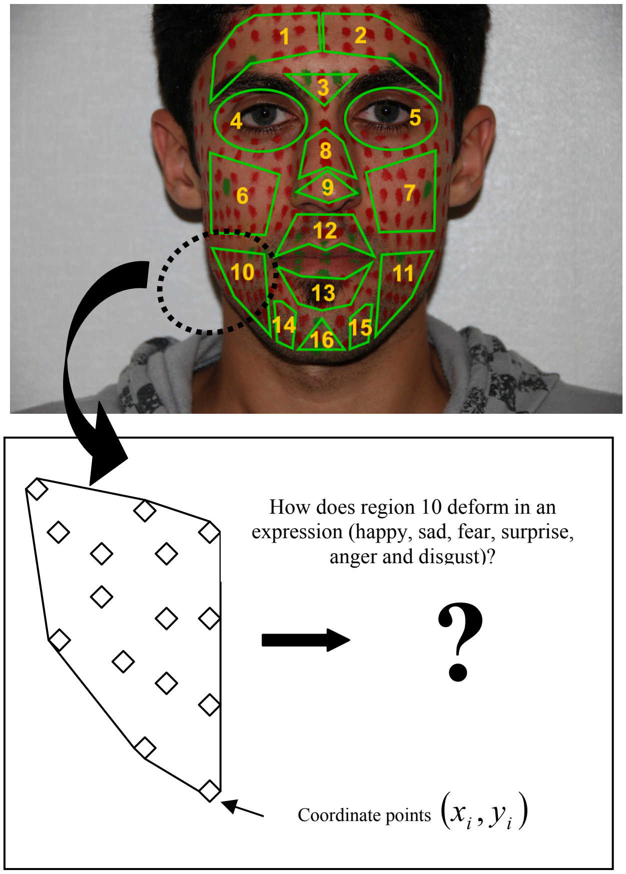

Figure 5. Example to demonstrate region deformations.

To find the region deformation parameters using the rubber-transformations (described in section III), we collected, from 2D images, the (x, y) coordinates for the neutral expression and then collected the corresponding coordinates (x', y') for the preformed expression (i.e. $x_i \Rightarrow x_i'; y_i \Rightarrow y_i' \quad i = 1,...,n$, where n represent the number of points in a region). The data was normalized to eliminate all global head movements by pose- and scale-normalizations, as described in section IV (C). Using the normalized data, we computed the parameter values for each facial region to obtain the following set of parameters for an expression

$$(a_1, b_1, \ldots, f_1, a_2, b_2, \ldots, f_2)_k \qquad (k = 1,...,16)$$

where k is the number of facial regions.

Using the computed deformation parameters for region k, we can obtain the transformed coordinates $(x_i', y_i')_k$ of any point within a region by substituting the parameter values $(a_1, b_1, \ldots, f_1, a_2, b_2, \ldots, f_2)_k$ and the original coordinate points $(x_i, y_i)_k$ into the right hand-side of equations (1) and (2) from section III.

Therefore, applying this process to all facial regions will result in deforming the facial regions from the neutral facial expression to the desired facial expression.

C. Normalization Process

To analyze the details of the facial regions, global head movements are eliminated by pose- and scale-normalizations. The normalization is done based on two global facial locations (eye pupils) p_1 and p_2 as shown in Fig. 6, where m is the middle point between p_1 and p_2.

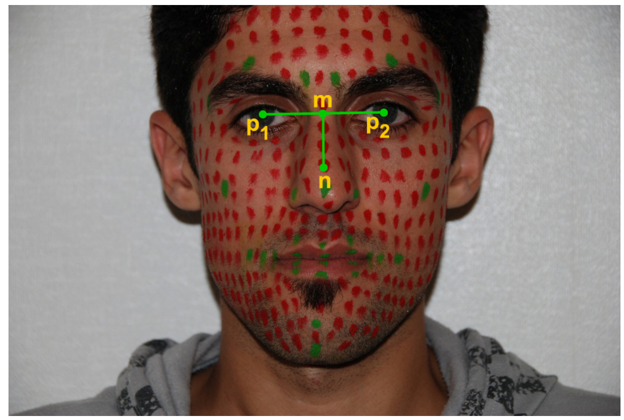

Figure 6. Global facial locations for normalising.

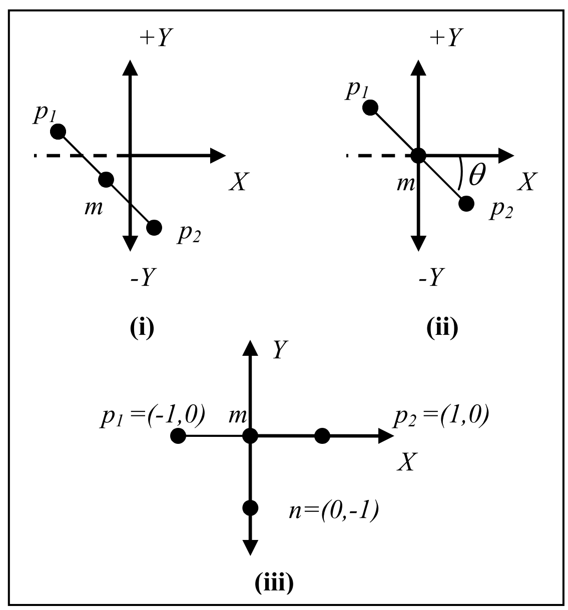

Figure 7. Normalisation operations: (i) translating the points about the origin (ii) eliminating the XY rotations (iii) scaling operation.

The line (m, n) is perpendicular to the line (p_1, p_2) and has the same distance as the line (m, p_1). All of the points were normalized through the following three transforms

- Translate point m to the origin, and then translate all other points about the origin by $(-m_x, -m_y, -m_z)$ as illustrated in Fig. 7 (i).

- Rotate all points about the origin by θ around the *Z*-axis, where θ is the angle between vector (p_1 , p_2) and the *X*-axis (see Fig. 7 (ii)).
- Scale all the point to $\left(\frac{x_i}{x_{p_2}}, \frac{y_i}{x_{p_2}}\right)$ as shown in Fig. 7 (iii).

  This operation will scale points p_1 and p_2 to (1, 0) and (1, 0) respectively, and point *n* to (0,-1).

V. DERIVATION OF FACIAL DEFORMATION TABLES (FDTS)

The aim of our work was to develop a new method for representing facial expressions in terms of deformation parameters, where those parameters can be employed in facial expression applications such as facial animations, recognition and interpretations.

To achieve the facial expression representations, we defined a Facial Deformation Table (FDT_E), for an expression E, that represents the deformation parameters for each facial region of expression E. The following describe the procedure to compute a FDT_E for each of the six main expressions (happy, sad, fear, surprise, anger and disgust).

- **Facial expression data acquisition:** We analyzed the facial expressions of twelve model subjects. Each model subject was asked to perform the six main facial expressions (happy, sad, anger, fear, disgust and surprise) from the neutral facial expression state. A series of frontal photographs of the face were taken to capture the each of the expressions. The model subject had several markers positioned on their face for the duration of the expression photography. The twelve models participated in the data acquisition stage are aged from 20-50 years.
- **Collecting Coordinate points:** The image data was analyzed for each facial expression by collecting the coordinate points for the neutral expression and the performed expression (before and after the expression). The collected coordinate points were grouped into their facial regions as described in section IV(A).
- **Normalization:** All of the facial expression coordinate data were normalized as described in section IV (C).
- **Computing the deformation parameters:** For each of the facial expression coordinate data sets, the deformation parameter values were computed as described in section IV (B). This computed result forms the FDT_E for expression E.

A. Analysis and Results

The results from the above procedure gave a set of FDT_E that represent the deformation parameters for each of the participants' facial expressions.

To compute a generic set of FDT_E that can be used to represent expressions, the mean (μ) of the FDT_E parameter

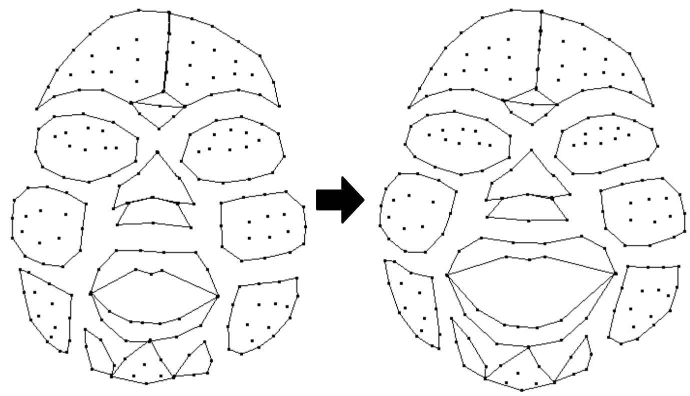

Figure 8. Example of using Table 1 to transform (a) the neutral expression to (b) the smile expression.

values for expression E are computed. The computed results represent each of the six main expressions in a FDT_E^{μ} . For example, TABLE I shows the FDT_{smile}^{μ} containing the deformation parameters for the smile expression. Fig. 8 shows the results from using the deformation parameters to transform facial regions.

Using the standard deviation (σ) analysis of the mean, we can compute and represent the range value for each of the deformation parameters as $\mu \pm \sigma$.

VI. CONCLUSION AND FUTURE WORK

This paper proposed a novel approach in representing facial expressions based on a quadratic deformation model applied to muscle regions. The rubber-sheet transformation functions are used to compute the deformation parameters for facial regions, where we defined sixteen facial regions based on the anatomy of the facial muscle system. The results form a set of FDTs, which we generalized by taking the average of the FDTs for each expression. The FDTs can be used and employed in facial expression applications such as facial recognition and facial animations.

Future research will focus on the following:

- Compressing the size of the FDT representations using suitable data structures.
- Improve the method so a broader range of facial expressions can be generated using FDTs.

TABLE I: DEFORMATION PARAMETERS FOR THE SMILE FACIAL EXPRESSION

Facial Region Number	Deformation Parameters											
	a_1	a_2	a_3	a_4	a_5	a_6	b_1	b_2	b_3	b_4	b_5	b_6
1	-0.003	0.009	-0.002	0.991	0.007	0.000	-0.007	0.015	0.011	-0.018	0.987	0.000
2	-0.002	0.000	-0.007	1.005	0.015	0.000	0.005	0.005	0.004	-0.021	0.998	0.000
3	-0.016	-0.119	0.093	1.070	-0.045	-0.001	0.009	-0.037	0.073	-0.009	0.951	0.001
4	-0.009	-0.019	-0.104	0.977	-0.024	0.000	-0.009	0.086	0.101	-0.015	0.994	0.001
5	0.001	-0.005	0.104	1.001	0.026	-0.001	-0.012	-0.126	0.089	0.021	1.037	0.001
6	0.040	-0.041	-0.026	1.054	-0.012	0.000	-0.078	0.022	0.003	-0.194	1.012	0.000
7	-0.050	-0.088	-0.031	1.035	-0.048	0.000	-0.082	-0.030	-0.015	0.197	0.992	0.000
8	0.064	-0.035	-0.022	0.978	-0.023	-0.001	0.132	-0.021	0.057	-0.022	1.072	0.004
9	-0.024	-0.207	0.010	0.803	0.008	0.000	0.247	-0.059	0.010	-0.088	1.048	0.000
10	0.157	-0.007	0.022	1.387	-0.012	0.000	-0.113	-0.028	-0.020	-0.286	0.931	0.000
11	-0.107	-0.008	-0.048	1.222	-0.112	0.000	-0.132	0.022	-0.028	0.314	0.911	0.000
12	0.013	0.050	-0.006	1.308	-0.012	0.000	0.107	0.018	0.007	0.047	0.961	0.000
13	0.021	0.065	0.005	1.326	0.015	0.000	0.306	0.057	-0.023	0.133	0.972	0.000
14	0.318	0.061	0.003	1.609	-0.014	0.000	-0.173	-0.322	0.073	-1.521	1.315	0.000
15	-0.321	0.066	0.003	1.673	0.042	0.000	-0.218	0.146	0.038	1.052	1.222	0.000
16	0.044	0.153	0.005	1.632	0.015	0.000	0.003	0.036	0.023	0.042	1.110	0.000

- Looking at how the FDT representations can be applied to the MPEG-4 standard applications, such as facial animations.
- Evaluate the proposed facial representation approach by integrating the FDTs into a facial animation system.

ACKNOWLEDGMENT

The authors would like to thank Karin Fitz for her help and time with collecting the facial expression data.

REFERENCES

[1] V. C. Flores. ARTNATOMY (anatomical basis of facial expression interactive learning tool). In ACM SIGGRAPH 2006 Educators Program (Boston, Massachusetts, July 30 - August 03, 2006).

[2] Ekman P and Friesen W, "Manual for the Facial Action Coding System", Consulting Psychologist 1977, Press Palo Alto California.

[3] DataFace: Psychology, Appearance, and Behavior of the Human Face. Retrieved June 6th, 2007, from: http://www.face-and-emotion.com.

[4] I. Essa and A. Pentland. Coding, Analysis, Interpretation and Recognition of Facial Expression. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(7):757-763, 1997.

[5] M.J. Black and Y. Yacoob, “Tracking and Recognizing Rigid and Nonrigid Facial Motions Using Local Parametric Models of Image Motion,” Proc. Int"l Conf. Computer Vision, pp. 374-381, 1997.

[6] B. Fasel and J. Luettin, "Automatic facial expression analysis: A survey," Pattern Recognition, vol. 36, no. 2, 2003, pp. 259-275.

[7] I. S. Pandzic and R. Forchheimer, Eds. 2003 Mpeg-4 Facial Animation: the Standard, Implementation and Applications. John Wiley & Sons, Inc.

[8] K. Matsumura, Y. Nakamura, and K. Matsui, “Mathematical Representation and Image Generation of Human Faces by Metamorphosis”, Electronics and Comm. in Japan-3, vol. 80, no. 1, pp. 36-46, 1997.

[9] M. Pantic and L.J.M. Rothkrantz, “Automatic Analysis of Facial Expressions: The State of the Art,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 22, no. 12, pp. 1424-1445, Dec. 1996.

[10] Shimizu, E. and Fuse, T. 2003. Rubber Sheeting of historical maps in GIS and its application to landscape visualization of old-time cities: focusing on Tokyo of the past. University of Tokyo. Proceedings of the 8th International Conference on Computers in Urban Planning and Urban Management, Reviewed Papers, CD-ROM, 2003. http://planner.t.u-tokyo.ac.jp/member/fuse/rubber_sheeting.pdf.

[11] Wootton, R., Springall, D. R., Polak, J. M. (eds.) (1995). Image Analysis inHistology—Conventional and Confocal Microscopy. Cambridge UniversityPress, London.

[12] Goerge Wolberg. Digital Image Warping. IEEE Computer Society Press, Los Alamitos, California, 1990.

[13] Weisstein, Eric W. "Least Squares Fitting--Polynomial." From MathWorld--A Wolfram Web Resource. http://mathworld.wolfram.com/LeastSquaresFittingPolynomial.html.

[14] Rafael C. Gonzalez, Richard E. Woods. Digital Image Processing. Third Edition, Prentice-Hall, 2007.

[15] British Ordnance Survey. Features of transformations options. Retrieved on February 1st,, 2009, from http://www.ordnancesurvey.co.uk/oswebsite/pai/pdfs/transformation_descriptions.pdf

[16] Facial Action Coding System. Retrieved June 2007 from ttp://web.cs.wpi.edu/~matt/courses/cs563/talks/face_anim/ekman.html.

[17] Ekman, P. & Friesen, W. V. (1975). Unmasking the face. A guide to recognizing emotions from facial clues. Englewood Cliffs, New Jersey: Prentice-Hall.