2009 Digital Image Computing: Techniques and Applications

A Quadratic Deformation Model for Facial Expression Recognition

M. Obaid*[†], R. Mukundan[†], R. Goecke[‡§], M. Billinghurst* and H. Seichter*
*HITLab New Zealand, University of Canterbury, Christchurch, New Zealand
[†]Computer Science and Software Engineering Dept., University of Canterbury, New Zealand
[‡]HCC Lab / NCBS, Faculty of Information Sciences and Engineering, University of Canberra, Australia
[§]School of Computer Science, CECS, Australian National University, Canberra, Australia
Email: {mohammad.obaid, mark.billinghurst, hartmut.seichter}@hitlabnz.org,
mukund@canterbury.ac.nz, roland.goecke@ieee.org

***Abstract*—In this paper, we propose a novel approach for recognizing facial expressions based on using an Active Appearance Model facial feature tracking system with the quadratic deformation model representations of facial expressions. Thirty seven Facial Feature points are tracked based on the MPEG-4 Facial Animation Parameters layout. The proposed approach relies on the Euclidean distance measures between the tracked feature points and the reference deformed facial feature points of the six main expressions (smile, sad, fear, disgust, surprise, and anger). An evaluation of 30 model subjects, selected randomly from the Cohn-Kanade Database, was carried out. Results show that the main six facial expressions can successfully be recognized with an overall recognition accuracy of 89%. The proposed approach yields to promising recognition rates and can be used in real time applications.**

***Keywords*-Facial Expression Recognition, Quadratic Deformation Models, Active Appearance Models, Facial Feature Tracking**

I. Introduction

Recognition of facial expressions with computer vision based methods has been used in many application areas, such as ambient interactivity in multimodal and affective applications, or e-learning and security software where salient expression changes can be used to guide interaction strategies.

In this paper, we introduce a new approach for recognizing facial expressions. First we will introduce the related body of work in the area of facial expression recognition, and then outline in detail our approach for recognizing facial expressions. Additionally, we evaluate our approach with a standard database and discuss the results.

II. Background

For the past two decades, facial expression recognition has attracted many computer vision researchers. In general, there are two main approaches to facial expression recognition and tracking: (1) holistic (processes the face as a whole) and (2) feature based (focuses on facial feature patterns) [1], [2]. A number of methods, representing both approaches, have been proposed by the computer vision community for the task of facial feature tracking and recognition including Principal Component Analysis (PCA) [3], Gabor Wavelets [4], hybrid approaches [5], and more recently Active Appearance Models (AAM) [6].

One of the most significant representations used in facial expression recognition is the Facial Action Coding System (FACS), developed by Ekman and Friesen [7]. It defines facial expressions as a combination of facial actions corresponding to movements of particular muscle groups. Ekman and Friesen defined facial muscle actions as Action Units (AU). One of the limitations of FACS is that the facial expressions' motions are described based on local information, making the definition of facial expressions a difficult process for facial expression applications.

Another common representation used in the recognition of facial expression is the MPEG-4 standard [8], which supports the definition, encoding, and transmission of facial animation. The MPEG-4 facial animation standard is defined by 84 feature points (FPs). These points are used to define a set of 68 Facial Animation Parameters (FAPs), consisting of high-level parameters, which specify visemes and expressions, and low-level parameters that describe the movements of feature points defined on head, tongue, eyes, mouth, and ears.

MPEG-4 FAPs and AUs provide a general description of facial muscle movements, but one common limitation is the lack of defining spatial information of facial deformations [9].

Based on FACS and the anatomical facial structure, Obaid et al. [10] attempt to represent each expression as a collection of the most general quadratic transform coefficients that are model independent.

In recent years, research on using real-time tracking methods in facial expression recognition systems has become popular. First proposed by Edwards *et al.* [11], the AAM has attracted much interest in the computer vision community for modeling and segmenting deformable visual objects. The method makes use of linear subspaces, allowing a compact representation of both shape and texture. The AAM's rapid fitting stems from the utility of fixed linear update models. Despite its simplicity, the linear update model has been shown to approximately capture the relation-

978-0-7695-3866-2/09 $26.00 © 2009 IEEE
DOI 10.1109/DICTA.2009.51

ship between the AAM's texture residual and the optimal parameter updates [11]. However, this relationship holds only loosely, as it depends on the current shape and texture parameters. Saragih and Goecke [12], [13] recently proposed a new discriminative-iterative fitting procedure that couples training and testing through simulations of real fitting conditions. It has been shown to exhibit superior fitting and generalization capabilities compared to other AAM fitting methods, while maintaining a high computational efficiency that allows real-time fitting/tracking [13].

This paper introduces a novel facial expression recognition technique based on a real-time AAM facial feature tracking system with the facial expression representations described in [10].

III. Facial Feature Tracking

The AAM's intrinsic variations in shape and texture of deformable visual objects are modeled as a linear combination of basis modes of variation that are composed with a global (similarity) transformation

$$\begin{aligned} S(q_s) : \Re^{n_s} \mapsto \Re^{2n} &= s(I \otimes R)(\mu_s + \phi_s p_s) + \mathbf{1} \otimes t) \\ T(q_s) : \Re^{n_t} \mapsto \Re^{m} &= \alpha(\mu_t + \phi_t p_t) + \beta \mathbf{1} \end{aligned}$$

where S and T denote the generative models for shape and texture, parameterized by $q_s = \{s, \mathbf{R}, \mathbf{t}, \mathbf{p_s}\}$ and $q_t = \{\alpha, \beta, p_t\}$, respectively. Here, $\{\mu_s, \phi_s\}$ and $\{\mu_t, \phi_t\}$ denote the mean and bases of variations of the shape and texture, which are typically obtained by applying PCA on the training data. The intrinsic shape is composed with a similarity transform, parameterized by a global scaling s, a rotation $\mathbf{R}$ and a translation $\mathbf{t}$. The intrinsic texture is scaled by a global gain α and biased by β. Finally, n denotes the number of points in the model's shape and m denotes the number of pixels in the model's texture.

AAM fitting (or tracking for image sequences) is the process of finding the model parameters $p = \{q_s, q_t\}$ which best fit an AAM to an image I. This is usually an iterative procedure that sequentially updates the model parameters $\mathbf{p}$ through an update function

$$\Delta p = U(*, p) \circ F(I; p) \quad . \tag{1}$$

Here, F is a feature extraction function that represents the image I from the perspective of the AAM at its current parameter settings and Δp are the updates to be applied to the current parameters. U is typically chosen to be the linear update model

$$U(F; p) : \Re^{m} \mapsto \Re^{n_s + n_t} = Gf + b \tag{2}$$

where $f = F(I; p)$, although nonlinear mappings have also been used [14]. In any case, a good coupling between U and F is required to ensure good predictions of the updates.

Recently, Saragih and Goecke [12], [13] proposed learning the entire fitting procedure in a discriminative framework rather than extracting the update model. Learning is performed on examples of real fitting scenarios, simulated on the training data. Utilizing a different update model in each iteration, the parameter updates can be written as

$$\Delta p_i = G_i F\Big(I; p + \sum_{j=1}^{i-1} \Delta p_j\Big) + b_i \tag{3}$$

where $\{G_i, b_i\}$ is the fixed update model for the i-th iteration. Given the training set, the optimal update models for all k iterations can be found by minimizing a cost function over the hand-labeled annotations for the j-th training sample

$$p^* = min \; q_s \|s^* - S(q_s)\|^2 \quad . \tag{4}$$

The parameters of this cost function are the update models themselves. A distance function that penalizes the difference between the manually annotated shapes and the predicted model's shape after N_i iterations is used [13]. Compared to texture based error measures, commonly used in generative AAM fitting, this distance function better encompasses all available knowledge about the optimal parameter setting, i.e. the hand-labeled annotations. With this formulation, the training procedure essentially simulates real fitting problems on the set of training images and perturbations.

Having trained the update models, AAM fitting then proceeds by simply applying all pre-computed update models for all trained iterations with no early termination. If the unseen images and their perturbations resemble those in the training set, then the fitting performance of the minimizer can be expected to approach that at training. This discriminative-iterative approach boasts significant improvements over other AAM fitting methods in both convergence accuracy and rate. Furthermore, these improvements are afforded without sacrificing computational efficiency. Also, the method affords excellent generalizability, as evidenced by its high convergence rates [13].

IV. Facial Expression Deformation Models

In this section, we describe a novel approach proposed by Obaid *et al.* [10] to represent facial expressions based on a quadratic deformation model applied to muscle regions. For each expression, the non-linear nature of muscle deformations can be captured using the following steps:

1) Subdivide the face into 16 muscle based facial regions, as shown in Figure 1.
2) Using the most general rubber-sheet transformation of second degree, derive the deformation parameters for each region by applying the least-square minimization technique, as shown in Figure 2.
3) Construct Facial Deformation Tables (FDT) to mathematically represent each expression.

The three steps are described in more details in the following sections.

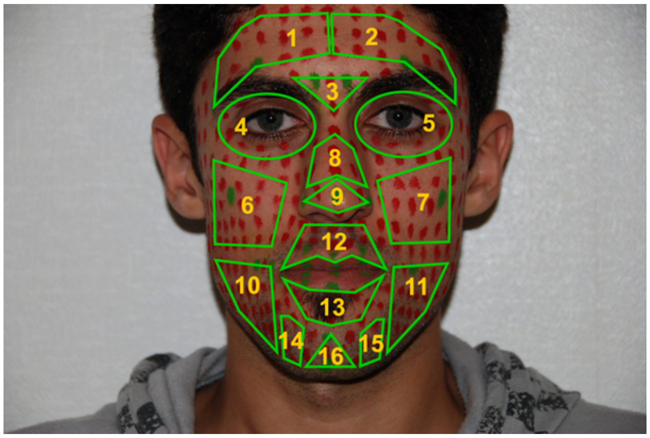

Figure 1. Facial regions.

A. Defining Facial Regions

To define facial regions that can be used to represent facial expressions mathematically, we looked at the FACS description of facial expressions [7]. 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 [15].

Using the FACS definition and the anatomy of the facial muscle system from [16], we defined sixteen regions that represent the deformation of the facial expressions. The sixteen regions form the minimum number for defining independent facial muscle groups. Figure 1 illustrates the defined facial regions.

B. Rubber-sheet transformations to derive the facial expression deformation parameters

Rubber-sheet transformations are higher-order (nonlinear) polynomial transformations [17], [18]. The name comes from the logical analogy of overlaying an elastic piece of rubber to fit over a surface of some shape.

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

$$x'_i = a_1x_i^2 + a_2x_iy_i + a_3y_i^2 + a_4x_i + a_5y_i + a_6 \quad (5)$$

$$y'_i = b_1x_i^2 + b_2x_iy_i + b_3y_i^2 + b_4x_i + b_5y_i + b_6 \quad (6)$$

$$i = 1, ..., n$$

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 the transformation equations above) as shown in Figure 3. 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.

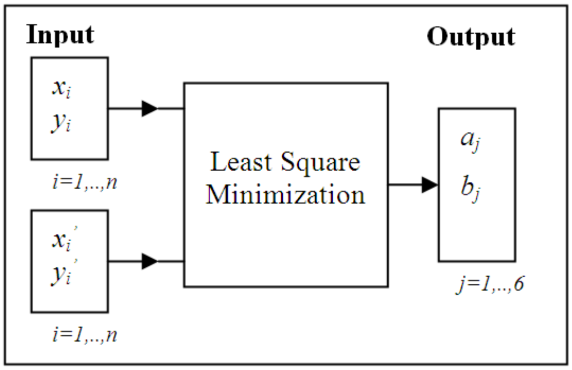

Figure 2. Derivation of deformation parameters.

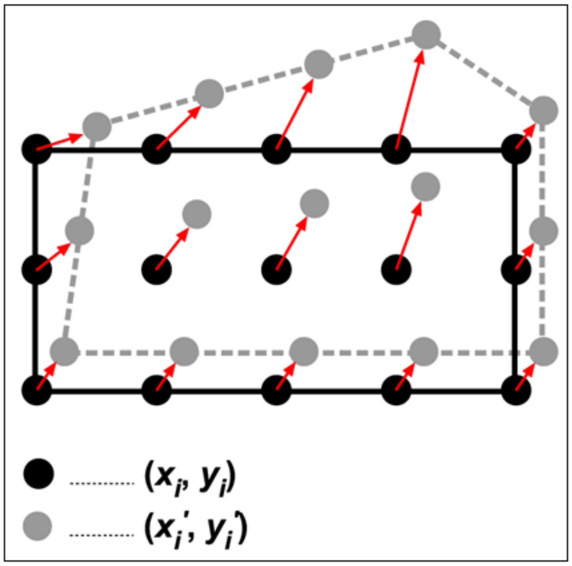

Figure 3. Deformation of a set of points.

To find the 12 parameters representing a transformation, the minimum of the sum of squares (or least square minimization) [17] is carried out on the pair of the quadratic equations above, which represent the x and y rubber-sheet transformations. The process of finding the transformation parameters is illustrated in Figure 2.

For example, the minimum of the sum of squares is achieved when the gradient of the sum-of-error-squares for the quadratic transformation equations is zero. The following equations represent the sum-of-error-squares

$$G_x = \sum \Big[x'_i - (a_1x_i^2 + a_2x_iy_i + a_3y_i^2 + a_4x_i + a_5y_i + a_6)\Big]^2$$

$$G_y = \sum \Big[x'_i - (b_1x_i^2 + b_2x_iy_i + b_3y_i^2 + b_4x_i + b_5y_i + b_6)\Big]^2$$

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.

C. Facial Deformation Tables (FDTs)

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 subject was asked to perform the six main facial expressions from the neutral facial expression state. A series of frontal photographs of the face were taken to capture each of the expressions. The subjects had several markers positioned on their faces for the duration of the photography. The twelve model subjects were 20-50 years old.
- **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 global head movements are eliminated by pose and scale normalizations. The normalization is done based on two global facial locations (eye pupils), this process is described further in step (1) of section V.
- **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.

V. Classification of Facial Expressions

In this section, we demonstrate how to utilize the FDT_E in the recognition of the six main expressions. Figure 4 shows the recognition process which consists of five main steps: (1) tracking feature points, (2) registering the natural expression feature points, (3) using the FDT_E to transform the feature points from step 2, (4) Using the Euclidean distance for similarity measurements, and (5) recognizing the facial expression. The following sections describe the above steps in more details.

A. Step (1): Tracking feature points

The initial locations of the tracked points are based on the MPEG-4 FAPs [8]. The complete set consists of 68 FAPs, but in this study a sub-set of 37 FAPs (Figure 5) was used since many points were considered irrelevant. In particular, we omitted any FAPs that do not get involved in co-articulation nor expressive articulation. This includes the "high-level" FAPs (viseme and expression), and FAPs dealing with ears and global head rotation.

Using the tracking algorithm described in section 3, we tracked the movement of the 37 facial feature points shown in Figure 5.

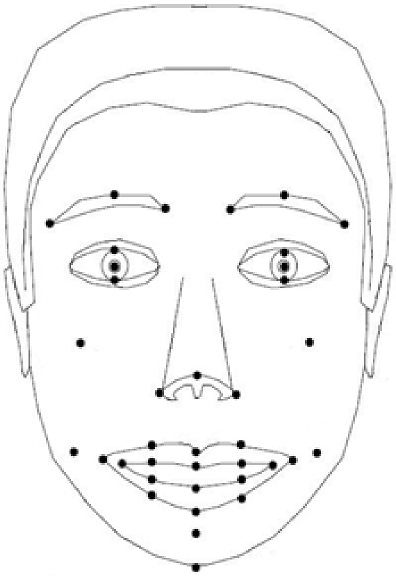

Figure 5. The tracked facial feature points based on the location of the MPEG-4 FAPs.

To analyze the details of the facial feature points, 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 (Figure 6).

The line (m, n) is the perpendicular bisector of the line (p_1, p_2). 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 Figure 6 (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 Figure 6 (ii)).
- Scale all the point to $\left(\frac{x_i}{x_{p_2}}, \frac{y_i}{x_{p_2}}\right)$ as shown in Figure 6 (iii). This operation will scale points p_1 and p_2 to (-1, 0) and (1, 0) respectively, and point n to (0,-1).

B. Step (2): Registering the natural expression and using the FDT_E

The normalized facial feature point locations are registered for the natural expression at the start of the recognition session. The registered points for the natural expression are then used as a reference (R_{natural}) for the duration of the session. Currently, the process of identifying the reference natural expression is done manually; however, as a future extension to this work, and to eliminate this manual

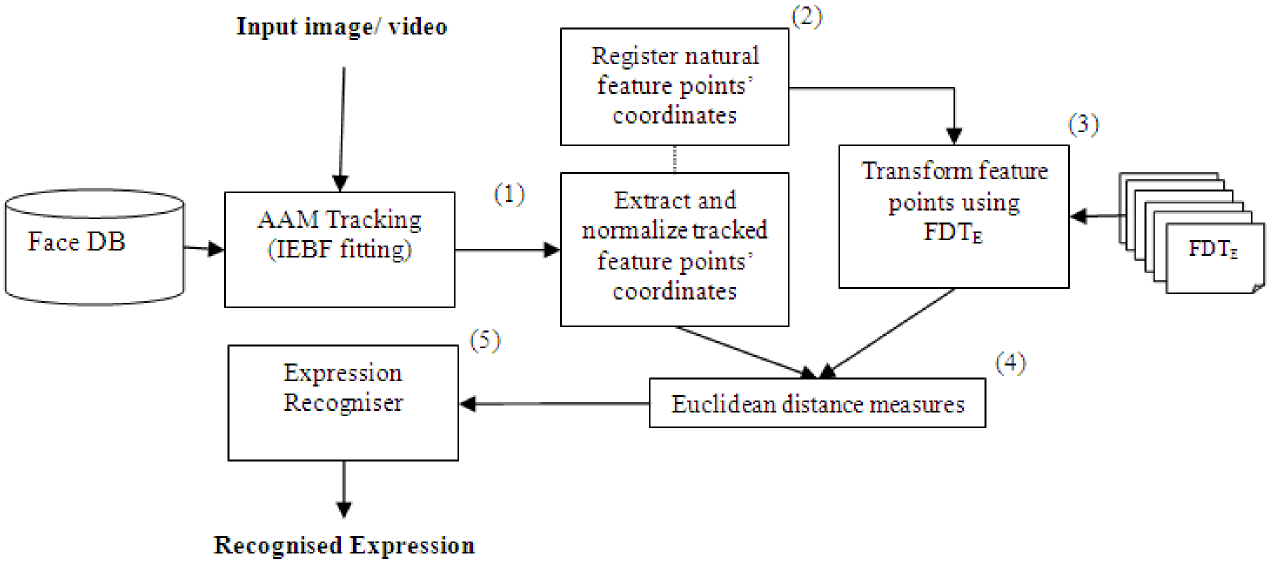
Input image/ video
(2)
Register natural feature points' coordinates
(3)
Transform feature points using FDT$_E$
Face DB
AAM Tracking (IEBF fitting)
(1)
Extract and normalize tracked feature points' coordinates
FDT$_E$
(5)
(4)
Expression Recogniser
Euclidean distance measures
Recognised Expression

Figure 4. Facial expression recognition process.

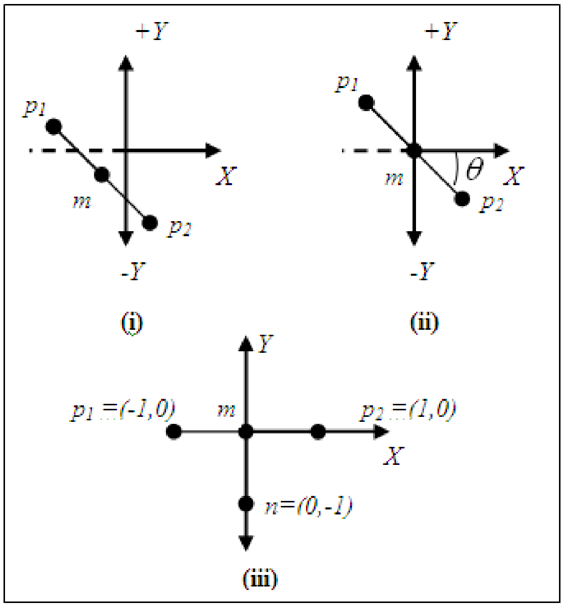

Figure 6. Normalisation operations: (i) translating the points about the origin (ii) eliminating the XY rotations (iii) scaling operation.

process, we will look at using a normalized average of the natural expression's facial feature points obtained from several subjects.

C. Step (3): Using the FDT_E

In this step, we apply the FDTE deformations models (described in section 4) to the reference points R_{natural} from step (3). This will formulate a deformed reference for each of the expressions (R_E, where E = <smile, angry, sad, fear, surprise, disgust>). Using the set of deformed references, we can recognize the facial expression based on similarity measures as explained in step (4).

D. Step (4): Euclidean distance similarity measure

After having extracted the tracked facial feature points from step (1) and using the set of deformed expression references (R_E) from step (3), our next task is to find the similarity measure d such that $d(P_i^T, P_i^E)$ is small if and only if P_i^T and P_i^E are close, where in this case P_i^T represents the tracked points and P_i^E represent the deformed reference points as shown in Figure 7.

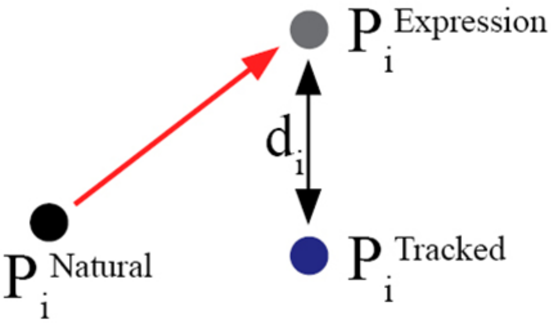

Figure 7. Euclidean distance measures.

In our method, we compute d by summing the Euclidean distances between the tracked points and R_E as follows

$$d_E = \sum_{i=1}^{n} \sqrt{(x_i^T - x_i^E)^2 + (y_i^T - y_i^E)^2} \qquad (7)$$

where n is the total number of facial feature points and d_E is the total sum of distances for expression E. Since E is a set of the six main expressions, we compute a vector of six Euclidean distance measures, which will be used in Step (5) for recognizing the facial expression.

E. Step (5): Facial Expression Recognition

Using the Euclidean distance measures' vector, computed from step (4), facial expressions are classified based on the following rule [19]: ”A small distance is equivalent to a large similarity.”

Therefore, the smallest value computed in the Euclidean distance vector is chosen to be the recognized expression.

VI. Experimental Results and Discussion

To evaluate our facial expression recognition method in terms of accuracy, we have preformed experiments on 30 subjects (15 females and 15 males) chosen randomly from the Cohn-Kanade Database [20]. Using the CKDb to randomly select facial expression images allows us to ensure the diversity of the chosen images in terms of facial characteristics such as shape, size and texture. Each subject expressed the six main facial expressions from the natural facial expression state.

Overall, the selected 30 subjects database consists of 992 natural images and 2432 expressed images, of which, 448 images for Anger, 296 images for Disgust, 346 images for Fear, 532 images for Smile, 423 images for Sad and 387 images for Surprise. For each subject, one natural image and all expressed images were used in the evaluation experiment.

Using the classification method, described in section 5, we performed the evaluation experiments for each subject. The overall recognition accuracy results are summarized in Table I.

The results show 88.9% overall recognition accuracy of all expressions with the highest recognition rates of Surprise (100%) and the lowest recognition rates of Fear (66.7%).

The distinct deformation feature of the surprise expression (opening the mouth and dropping the jaw) allows the system to have a 100% recognition accuracy rate of the surprise expression. While, the deformation features of the Fear expression (pulling the mouth corners and a slight opening of the mouth) can overlap with other expressions (such as Disgust), which explains its lower recognition rate of 66.7%.

Comparison results of the proposed approach with recent results achieved by other methods that use AAM face appearance vectors and use the same CKDb database, such as Support Vector Machines (SVM) classifier [21] or Neural Networks classifier (NN) [22] are shown in Table II.

Table II
Facial expression recognition accuracy rates of different methods.

	SVM [21]	**NN** [22]	**Proposed Approach**
Smile	72.6%	99.0%	93.3%
Sad	82.8%	78.05%	96.7%
Anger	75.9%	87.0%	93.3%
Fear	84.7%	83.0%	66.7%
Disgust	80.4%	91.0%	83.3%
Surprise	83.8%	95.0%	100%
Overall	80.0%	88.8%	88.9%

Generally, our approach outperforms the SVM method, and has nearly the same accuracy performance as the NN method. However, if we exclude the recognition rate of the Fear expression, our approach outperforms the other two methods. More work toward improving the accuracy of the 'Fear' expression will certainly result in achieving a higher overall recognition rate. One way to improve recognition rates of facial expressions that deform in a similar manner is to compute similarity measures based on distinct features of different facial expressions by assigning different weight values. This will be explored in further details as a future extension to this work.

Though we utilized a manual annotation method, our approach is geared towards real-time use. On a standard PC, with Intel®Core™2 Quad CPU Q6600 @ 2.40GHz, our approach without any optimization performs at 16ms per frame with one face tracked (around 62 frames per second).

VII. Conclusion

We proposed a novel facial expression recognition method by utilizing an AAM facial feature tracking system with a quadratic deformation model representations of facial expressions. The method tracks 37 facial feature points corresponding to the layout of the MPEG-4 standard FAPs. The reference natural expression facial feature points are extracted and deformed according to the FDTs described in [10]. Applying the Euclidean distance similarity measures on the tracked points and the reference deformations allows us to recognize facial expression.

Our results show that the overall average facial expression recognition rate is 88.9%. The achieved accuracy clearly shows encouraging results and can lead to future direction for higher rates in facial expression analysis and recognition.

A future direction for this research is to eliminate the manual process of registering the reference natural expression to fully automate the facial recognition process. One could also research the minimum number of tracked feature points required to maintain high recognition rates. Finally, further work is employed on improving the accuracy performance by identifying distinct deformations of facial features for different expression.

Acknowledgment

Research conducted for this project was partly funded through the FRST Grant UOCX0608 and is part of the CALLAS project.

The authors would like to thank Jason Saragih, Carnegie Mellon University, Pittsburgh, USA, for the use of the DeMoLib software library.

Table I
CONFUSION MATRIX FOR FACIAL EXPRESSION RECOGNITION

Expression	Smile	Sad	Anger	Fear	Disgust	Surprise
Smile	**93.3%**	0.0%	0.0%	0.0%	3.3%	0.0%
Sad	0.0%	**96.7%**	6.7%	0.0%	0.0%	0.0%
Anger	0.0%	3.3%	**93.3%**	3.3%	13.3%	0.0%
Fear	0.0%	0.0%	0.0%	**66.7%**	0.0%	0.0%
Disgust	6.7%	0.0%	0.0%	23.3%	**83.3%**	0.0%
Surprise	0.0%	0.0%	0.0%	6.7%	0.0%	**100.0%**

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

REFERENCES

[1] B. Fasel and J. Luettin, “Automatic facial expression analysis: a survey,” *Pattern Recognition*, vol. 36, pp. 259–275, 1999.

[2] K. Zakharov, “Affect recognition and support in intelligent tutoring systems,” Masters thesis, University of Canterbury, New Zealand, 2007.

[3] A. Calder, A. Burton, P. Miller, A. Young, and S. Akamatsu, “A principal component analysis of facial expressions,” *Vision Research*, vol. 41, no. 9, pp. 1179–1208, Apr. 2001.

[4] M. Dailey and G. Cottrell, “PCA = Gabor for expression recognition,” University of California at San Diego, La Jolla, CA, USA, Tech. Rep. 0629, 1999.

[5] F. Bourel, C. Chibelushi, and A. Low, “Robust facial feature tracking,” in *Proc. 11th British Machine Vision conference BMVC2000*, vol. 1, Bristol, England, Sep. 2000, pp. 232–241.

[6] F. Dornaika and J. Ahlberg, “Efficient active appearance model for real-time head and facial feature tracking,” in *AMFG ’03: Proc. IEEE International Workshop on Analysis and Modeling of Faces and Gestures*. Nice, France: IEEE Computer Society, 2003, p. 173.

[7] P. Ekman and W. Friesen, “Manual for the facial action coding system,” *Palo Alto: Consulting Psychologists Press*, 1977.

[8] I. Pandzic and R. Forchheimer, *MPEG-4 facial animation: the standard, implementation and applications*. John Wiley & Sons, 2003.

[9] P. Hong, Z. Wen, and T. Huang, “Speech driven face animation,” in *MPEG-4 facial animation - the standard, implementations, and applications*. John Wiley & Sons, Inc, 2002.

[10] M. Obaid, R. Mukundan, M. Billinghurst, and M. Sagar, “Facial expression representation using a quadratic deformation model,” in *International Conference on Computer Graphics, Imaging and Visualization*. IEEE Computer Society, Aug. 2009.

[11] G. Edwards, C. Taylor, and T. Cootes, “Interpreting face images using active appearance models,” in *FG ’98: Proceedings of the 3rd International Conference on Face & Gesture Recognition*. Nara, Japan: IEEE, Apr. 1998, pp. 300–305.

[12] J. Saragih and R. Goecke, “Iterative error bound minimisation for AAM alignment,” in *International Conference on Pattern Recognition ICPR2006*, vol. 2. Hong Kong: IEEE Computer Society, 2006, pp. 1192–1195, doi: 10.1109/ICPR.2006.730.

[13] J. Saragih and R. Göcke, “Learning AAM fitting through simulation,” *Pattern Recognition*, vol. 42, no. 11, pp. 2628–2636, 2009, doi: 10.1016/j.patcog.2009.04.014.

[14] J. Saragih and R. Goecke, “A nonlinear discriminative approach to AAM fitting,” in *Proc. IEEE International Conference on Computer Vision ICCV2007*. Rio de Janeiro, Brazil: IEEE Computer Society, 2007, doi: 10.1109/ICCV.2007.4409106.

[15] “Facial Action Coding System,” Jun. 2007, http://web.cs.wpi.edu/~matt/courses/cs563/talks/face_anim/ekman.html.

[16] V. Flores, “ARTNATOMY (Anatomical basis of facial expression interactive learning tool),” in *SIGGRAPH ’06: ACM SIGGRAPH 2006 Educators program*. Boston, Massachusetts: ACM, 2006, p. 22, doi: 10.1145/1179295.1179318.

[17] E. Weisstein, “Least squares fitting–polynomial,” *From MathWorld–A Wolfram Web Resource*, Aug. 2008, http://mathworld.wolfram.com/ LeastSquaresFittingPolynomial.html.

[18] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed. Prentice-Hall, 2007.

[19] A. Fielding, “Cluster analysis - distance measures,” Apr. 2009, http://uk.geocities.com/ahf_alternate/dist.htm.

[20] T. Kanade, Y. Yingli, and J. Cohn, “Comprehensive database for facial expression analysis,” in *FG ’00: Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition 2000*. Washington, DC, USA: IEEE Computer Society, 2000, p. 46.

[21] D. Datcu and L. Rothkrantz, “Facial expression recognition in still pictures and videos using active appearance models: a comparison approach,” in *CompSysTech ’07: Proceedings of the 2007 international conference on Computer systems and technologies*. Bulgaria: ACM, 2007, pp. 1–6, doi: 10.1145/1330598.1330717.

[22] H. Kuilenburg, M. Wiering, and M. Uyl, “A model based method for automatic facial expression recognition,” in *Machine Learning: ECML 2005*, ser. Lecture Notes in Computer Science, vol. 3720. Springer, Nov. 2005, pp. 194–205.