

# Facial Expression Recognition Using Continuous Dynamic Programming

Haihong Zhang and Yan Guo

*Real World Computing Partnership, Multi-Model Functions Lab, Kent Ridge Digital Labs  
{hhzhang, yguo}@krdl.org.sg*

## Abstract

*This paper describes a new approach to facial expression recognition (FER). We represent facial expressions by Facial Motion Graph (FMG), which is based on feature points and muscle movements. FER is achieved by analyzing the similarity between an unknown expression's FMG and FMG models of known expressions by employing Continuous Dynamic Programming (CDP). Furthermore we propose a method to evaluate edge weights in FMG similarity calculation, and use these edge weights to achieve a more accurate and robust system. Experiments show the excellent performance of this system on our video database, which contains video data captured under various conditions with multiple motion patterns.*

*Keywords: Facial Expression Recognition, Facial Motion Graph, Dynamic Programming*

## 1. Introduction

Facial Expression Recognition has received considerable attention from computer vision community for many years, as it plays an important role in advanced human computer interface as well as other potential applications such as semantic facial communication. However the topic remains a challenge to the researchers, actually even human observers often disagree on expression in FER experiments [6].

Psychological research [3] identified that six principal emotions are universally associated with distinct facial expressions. Ekman proposed a facial expression representation system called FACS (Facial Expression Coding System), an objective method for quantifying facial movement in terms of component actions. Some recognition systems have been developed under FACS [8][1]. However, a growing body of psychological researchers realizes that the lack of temporal and detailed spatial (both local and global) information is a severe

limitation of the FACS model. To tackle this problem, Essa and Pentland [4] proposed an approach for extracting an extended FACS model to improve accuracy in both time and space domain. However it is high computation cost of their system.

We use Facial Motion Graph (FMG) to represent facial expressions, which is of much less computation cost. In our previous work [2], facial expression is interpreted by VQ-HMM (Vector Quantization-Hidden Markov Models). That method can overcome the limitation of the continuous HMM method, which is subject to Gaussian-like probability distribution.

In this paper, we present a more flexible and adaptive facial expression recognition system, which employs Continuous Dynamic Programming (CDP). We refer to the gesture recognition work by Nishimura, et al [5] on using the DP scheme, but both our goal and implementation differ from theirs.

Compared to our previous system, the new system is more suitable for spotting an expression section in image sequences by CDP. Meanwhile using CDP makes the new system more flexible for online training, as the model can be easily updated with the optimized sample, furthermore we can build personalized model with a few samples to improve the system performance.

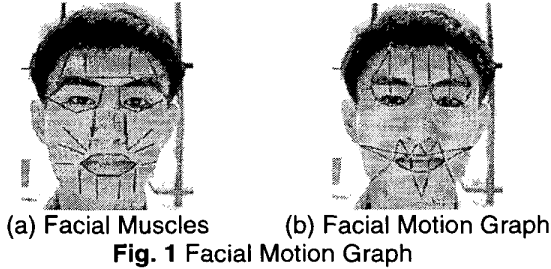
In this paper, Facial expression recognition is achieved by comparing the distances between the expression and the models using CDP on Facial Motion Graph sequences, which will be introduced in Section 2. In section 3, we will illustrate the DP scheme for FER. The edge weight evaluation method will be presented in section 4. Thereafter we will introduce our experiment and give out the conclusion.

## 2. Facial Motion Graph

Facial Motion Graph (see Fig.1) serves as the input of expression classifier, it represents facial action with the key points as well as their relationship [2]:

$$G = \{N, E\} \quad (1)$$

where  $N = \{n_1, n_2, \dots, n_p\}$ ,  $E = \{e_1, e_2, \dots, e_R\}$ ,  $N$  is the set of nodes and  $E$  is the set of edges, the nodes are selected on the facial salient points and the edges are selected according to the distribution of facial muscles (see Fig.1) so as to provide sufficient information for identifying different expressions.



Models are constructed in advance, which include personal Facial Graph templates and universal Motion Graph Models according to different types of expressions. The initial FMG templates are built by an interactive program, where we manually create a personal FMG according to a personal facial image.

On inputting video sequences to the system, face detection and normalization [10] are performed automatically, thereafter face identification [9][11] is accomplished, thus we load the corresponding personal FMG template from database and use it for FMG tracking in image sequence.

In order to obtain the dynamic FMG from the image sequence, the FMG initialization in the first frame as well as FMG tracking is accomplished by employing Gabor wavelet displacement estimation, which was introduced by Wiskott [6].

Gabor wavelet displacement estimation computes Gabor Jets  $J, J'$  at node  $n_i(x, y)$  in successive frames. A Jet is the output of an image convolved with a Gabor filter bank. To estimate the node displacement, one practical method is to minimize the phase-sensitive similarity function [6]:

$$S(J, J') = \frac{\sum_j a_j a'_j \cos(\phi_j - \phi'_j - \bar{d}k_j)}{\sqrt{\sum_j a_j^2 \sum_j a'^2}} \quad (2)$$

where  $a$  is the magnitude and  $\phi$  is the phase of a Jet,  $j$  denote the index of one Gabor filter and  $\bar{k}_j$  is the central

frequency of the filter,  $\bar{d}_j$  is the displacement to be evaluated.

We adopt Gaussian pyramid scheme on feature tracking to improve the performance of estimating large displacement while keeping the accuracy [2].

### 3. Facial Expression Recognition with Continuous DP

When FMG is computed in each frame, we obtain a FMG sequence (see Fig.2)

$$G(t) = \{N(t), E(t)\}, 1 \leq t \leq T \quad (3)$$

where  $T$  is the duration of expression video clip.

$G(t)$  can be considered as the feature vector of an expression. Expression instances may differ considerably to each other in time domain due to local or global scaling, thus it's unpractical to identify the expression by directly computing the distances between the feature vector and the model vectors. Moreover, there are always extra sections before and after the interesting expression section, so it's indispensable to spot the expression in the image sequence.

Dynamic Programming is a nice tool to tackle problems of time warping and spotting, thus we employ Continuous DP [4] as the expression classifier in our system. With the help of CDP, we can obtain expression distance according to the optimal correspondence between FMG sequences.

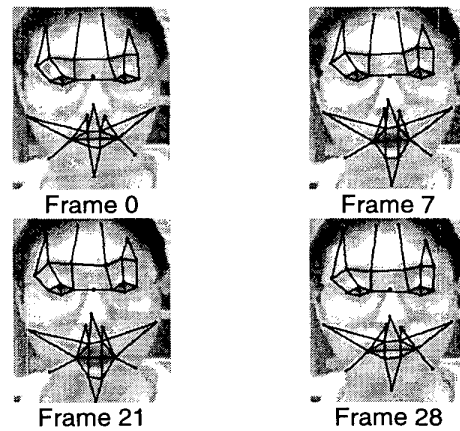


Fig. 2 A FMG Sequence Sample of surprise

#### 3.1 Accumulated Distance between Expressions

To formulate continuous DP, accumulated distance between expression A and B at  $(t, \tau)$  is calculated with the following iterative equations:

$$S(t, \tau) = \min(S_0, S_1, S_2) \quad (4)$$

where  $S_i$  denotes the accumulated distances on 3 paths (see Fig. 3) respectively

$$\begin{cases} S_0 = S(t-2, \tau-1) + c_1 \cdot d(t-1, \tau) + c_2 \cdot d(t, \tau) \\ S_1 = S(t-1, \tau-1) + c_3 \cdot d(t, \tau) \\ S_2 = S(t-1, \tau-2) + c_4 \cdot d(t, \tau-1) + c_5 \cdot d(t, \tau) \end{cases} \quad (5)$$

where  $t$  and  $\tau$  denote the time scalar of two expressions,  $c_1 \dots c_5$  are the parameters controlling the path preference, and  $d(t, \tau)$  denote the distance between two graphs

$$d(t, \tau) = \|G_A(t) - G_B(\tau)\| = \sum_{j=1}^N W_j (e_{A_i} - e_{B_i})^2 \quad (6)$$

where  $W_j$  denote the weight of edge  $j$ , which represent the importance of edge  $j$  in expression action.

At the stage of model training for the first time, there is insufficient information available for weight evaluation, therefore the weights are initialized with 1.0. Once the model is built, weights could be calculated with the method represented in section 4.

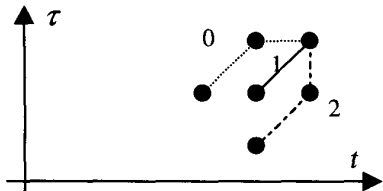


Fig. 3 Local path of Continuous DP

From formula (5), it can be seen that  $S(t, \tau)$ , the minimum accumulated distance between two expressions, allows the expressions shrink or stretch from  $\frac{1}{2}$  to 2 times in time axis. This property enables our system to deal with expression warp in time domain.

Given two FMG sequence, we found that the distance depends on the position of the last frame pair  $(t_0, \tau_0)$  in Continuous Dynamic Programming. In other words, the accurate expression distance depends on the accurate location of expression section in image sequences. In our system, the spotting work is achieved by correspondence determination, which will be introduced in following section.

### 3.2 Correspondence and Model Training

In the previous section we have mentioned the problem of expression spotting in image sequences, actually the problem also exists in model training.

Given a new expression sample  $(E_a(j, t), 1 \leq t \leq T_a)$  belongs to class  $c$ , it's difficult to train the model  $c$  by using  $E_a(j, t)$  directly, because the new expression usually have different duration and different time scale from the model. Moreover a sample clip usually has extra sections before and after the expression, hence it's necessary to determine the correspondence between model and sample.

The correspondence is achieved by minimizing the distance in DP algorithm (see Fig.8). The frame  $t$  in the sample expression with minimal accumulated distance to the model at the last frame ( $T_c$ ) is regarded as the end of sample expression, meanwhile the optimal path is obtained. The correspondence between model and sample could be determined according to the optimal path. At the same time, the training sample with the same duration to the model could be constructed, then expression model can be easily updated by averaging the model and the training sample [4].

### 4. Weighted Local Distance

It's obvious that each edge has different contribution to the facial expression performance, for example, the edges of the connection between midpoints of upper eyelid and lower eyelid may often confuse expression classifier depending on them due to the uncertain blink.

It's reasonable to determine the weight of one edge according to the separability of expressions depending on the edge. Therefore at first we calculate the distribution of the edge waveform, which represent the edge length variance in facial action. Let  $M_c(j, t)$  be the edge waveforms of a model, where  $t \in [1, \dots, t_c]$  is the time scalar,  $c \in [1, \dots, C]$  denote the type and  $j$  is the edge index, on the other hand, we obtain the sample edge waveforms from optimized sample  $E'_{ci}(j, t)$  (see section 3.2). Then each sample waveforms is projected into a new space, named the model space  $\{a_1, a_2, \dots, a_C\}$ . In this space, each sample waveform  $E'(j, t)$  could be approximated by a linear combination of model waveforms as

$$\hat{E}(j, t) = \sum_{i=1}^C a_i M_i(j, t) \quad (7)$$

or could be written as  $\hat{E}_j = A \cdot M_j$ , where  $M_j$  is a  $C \times T$  matrix of waveforms collected from all models at edge  $j$ , and  $\hat{E}_j$  denote the approximation of the sample

waveform,  $A = \{a_1, a_2, \dots, a_C\}$  denote the vector of coefficients.

Using LMS (Least Mean Square) algorithm to minimize the error of  $\|E - \hat{E}\|$  we get

$$A = (M_j^T M_j)^{-1} M_j^T E_j \quad (8)$$

On projecting all samples to the model space, every sample edge could be represented by a  $C$  dimensional vector, which characterizes the similarity between the sample and all the models on one edge. Consequently the separability (i.e. the weight) on one edge can be determined by its distribution in the model space.

Let the between-class scatter for edge  $j$  be

$$S_{jB} = \sum_{i=1}^C N_i (\mu_i - \mu)(\mu_i - \mu)^T \quad (9)$$

and the within-class scatter matrix for edge  $j$  be

$$S_{jW} = \sum_{i=1}^C \sum_{A_k \in X_i} (A_k - \mu_k)(A_k - \mu_k)^T \quad (10)$$

where  $\mu_i$  is the mean vector of class  $X_i$ , and  $N_i$  is the number of samples in class  $X_i$ , then the separability on edge  $j$  could be represented as

$$S_j = \frac{S_{jW}}{S_{jB}} \quad (11)$$

Let  $W_j$  (the weight of edge  $j$ ) be equal to  $S_j$ , therefore new local distance between FMGs could be redefined according to  $W_j$  (refer to formula 3).

#### 4. Experiment

We have built a small video database including 4 types of facial expression: Anger, Dislike, Happy and Surprise. These expression clips were collected from different persons. 15 clips were randomly selected and used for training the models. Fig.4 illustrates the models obtained, where horizontal axis denotes edge index and the variance of each edge's length is represented by grayscale in vertical line.

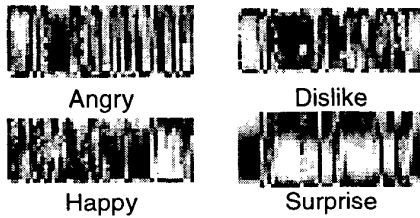


Fig. 4 Expression models

Then 22 clips (including some training samples) were used to do expression recognition test (without edge weighting) and we obtained the results showed in Table 1.

Table 1. FER Results (without edge weighting)

Expression	Anger	Dislike	Happy	Surprise
Anger	4	0	0	0
Dislike	1	5	0	0
Happy	0	0	5	0
Surprise	0	0	0	7
Success	80%	100%	100%	100%

The distance between test samples and models could also be illustrated as Fig.5, where the distance between a sample and a model is represented by the intensity of a pixel in this distance image. Horizontal axis denotes 22 expressions (divided by dash lines as four groups) and vertical axis denotes 4 models.

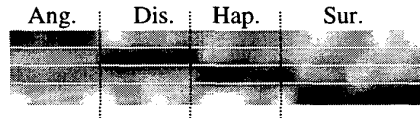


Fig. 5 Distance Image (without edge weighting)

On analyzing the models and some samples, we obtained edge weights as shown in Fig.6.

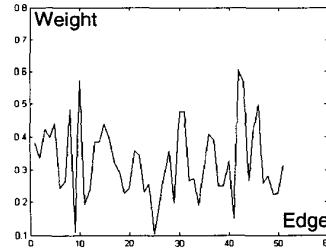


Fig. 6 Edge Weights

In the map we can see the significance of every edge in facial expression, e.g. the edge 42, 43 (the connection lines between upper lip and lower lip) show their prominent status as their significant function in surprise expression.

After edge weighting, the system performance were improved as shown in Table 2:

Table 2. FER Results (with edge weighting)

Expression	Anger	Dislike	Happy	Surprise
Anger	5	0	0	0
Dislike	0	5	0	0
Happy	0	0	5	0
Surprise	0	0	0	7
Success	100%	100%	100%	100%

The distance image also shows an apparent improvement of our system performance in Fig.7, where the peak of classifier output became sharper.

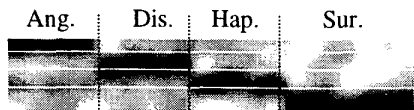


Fig. 7 Distance image (with edge weighting)

Fig.8 illustrates a process of spotting expression and finding optimal path, where the optimal end point is found at frame 34 according to line 2, accordingly the optimal path is determined as line 1.

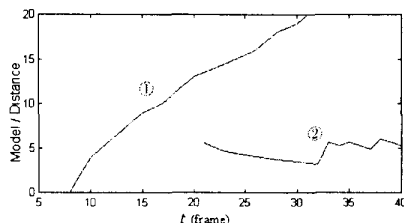


Fig. 8 Expression Spotting and Optimal DP Path

## 5. Conclusion

This paper proposes a new method for facial expression recognition. We create facial motion graph models in the light of facial muscle distribution. Based on Facial Motion Graph, DP scheme is employed to evaluate the similarity of different expressions. Models are trained by averaging samples according to the optimal path found by DP, furthermore we analyzed each edge's contribution to local distance between expressions, and thereby the system performance is improved by using the edge weights.

Our Experiments demonstrate our system performance is very good, indicating our approach to be valid and suitable for spotting an expression and distinguishing it in image sequences.

The CDP based expression classifier is also quite fast. 36 expression clips (there are 30 to 65 frames per clip) can be identified within 2 seconds. However, the FMG tracking is very computation expensive, the speed is only 2 fps or so. This suggests us to improve our algorithm of FMG tracking.

So far our facial expression database is quite small, it urges us to enrich the database and do more experiments.

## References

[1] G.Donato, M.S.Bartlett, J.C.Hager, P.Ekman, etc, Classifying Facial Actions, IEEE Transactions on

Pattern Analysis and Machine Intelligence, Vol.21, No.10, October 1999, pp974-989

[2] D.Q.Zhang, Y.Guo, J.K.Wu, Facial Expression Recognition Using VQ-HMM, Proc. of SCI'2000, Vol.5, pp340-344, 2000.

[3] P.Ekman and W.V.Friesen, The Facial Action Coding System, Consulting Psychologists Press Inc, 1978

[4] I.A.Essa, A.P.Pentland, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.19, July 1997, pp757 -763

[5] T. Nishimura, H.Yabe, R.OKA, A Method of Model Improvement for Spotting Recognition of Gestures Using an Image Sequence, New Generation Computing, Vol.18, pp89-101, 2000

[6] L.Wiskott, J.-M.Fellous, etc, Face Recognition by Elastic Bunch Graph Matching, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.19(7), July 1997, pp775-779,

[7] P.Ekman and W.Friesen, Unmasking the Face, A Guide to Recognising Emotions From Facial Expressions, Prentice Hall, 1975

[8] K.Mase, Recognition of facial expressions for optical flow, IEICE Transactions, Special Issue on Computer Vision and its Applications, E74(10), 1991.

[9] J.K.Wu, Recognition by Recall - A New Paradigm for Object Recognition, IEEE Transactions, SMC'97, pp3090-3095.

[10] W.M.Huang, Q.B.Sun, C.P.Lam, J.K.Wu, A Robust Approach to Face and Eyes Detection from Images with Cluttered Background, The 14th International Conference on Pattern Recognition, Vol.2, pp110-113, 1998.

[11] M.Rajapakse, Y.Guo, Efficient Gabor Feature Based Approach for Face Recognition, Proc. of SCI'2000, Vol.5, pp265-269, 2000.