2012 IEEE International Symposium on Multimedia

FaceFetch: A User Emotion Driven Multimedia Content Recommendation System Based on Facial Expression Recognition

Mahesh Babu Mariappan, Myunghoon Suk, Balakrishnan Prabhakaran

Department of Computer Science
University of Texas at Dallas
Richardson, Texas
{maheshbabu.mariappan, mhsuk, bprabhakaran}@utdallas.edu

Abstract—**Recognition of facial expressions of users allows researchers to build context-aware applications that adapt according to the users' emotional states. Facial expression recognition is an active area of research in the computer vision community. In this paper, we present FaceFetch, a novel context-based multimedia content recommendation system that understands a user's current emotional state (happiness, sadness, fear, disgust, surprise and anger) through facial expression recognition and recommends multimedia content to the user. Our system can understand a user's emotional state through a desktop as well as a mobile user interface and pull multimedia content such as music, movies and other videos of interest to the user from the cloud with near real time performance.**

Keywords-Emotion Recognition; Facial Expression Recognition; Context-Based Content Recommendation; Computer Vision.

I. INTRODUCTION

Human emotions communicated through facial expressions are very valuable as a source of information about how humans feel at a particular moment. This fact is exploited in several areas of research such as robotics, video games and animation, behavioral science and psychiatry, driver safety, education, etc. Furthermore, an understanding of the user's emotional state allows researchers to build very powerful and intuitive context-sensitive multimedia applications with a natural user interface. These applications are called affect-sensitive applications.

Facial expressions are produced by the often involuntary manipulation of muscles in the face in a manner and fashion that could change the position and shape—both absolute and relative—of facial features such as eyes, eye lids, eye lashes, nose, lips, cheek muscles, etc. This oftentimes creates wrinkles and bulges in the face typically conveying a user's emotional state. The intensity of a facial expression depends on the relative change in the facial features from the baseline values. These baseline values are measured when the subject's face is in the neutral state.

In this paper, we present FaceFetch, a novel context-based multimedia content recommendation system that understands users' emotional states through facial expression recognition and recommends multimedia content to them.

Our main contributions in this paper include (1) FaceFetch and our (2) ProASM feature extraction algorithm based on Active Shape Models [5]. ProASM improved the accuracy and convergence time of the feature extraction process besides making FaceFetch robust to significant illumination changes. We used our ProASM facial feature extractor for FaceFetch's facial expression recognition engine. We present the details of ProASM in section IV.B. In section II, we present some recent related work. In section III, we detail the dataset that we used for FaceFetch's facial expression recognition engine. In section IV, we present the algorithm for facial expression recognition and the system architecture for FaceFetch. We provide system performance details in Section V and finally conclude this paper in section VI.

II. RELATED WORK

We now present some recent work related to systems that use facial expressions in innovative ways and content recommendation systems that use emotions. Arapakis et al in [6] use facial expressions and other physiological signals to model user affective responses and to predict the topical relevance of information items in the absence of explicit user judgments. Valenti et al in [7] present a visual creativity tool that uses facial expressions to produce sounds. Hornof et al present in [8] EyeMusic, which uses users' eye movements to create musical compositions that are played with the eyes. Rho et al [1] propose a context-based music recommendation system which employs SVR (Support Vector Regression) to map the musical feature vectors to moods. Cabredo et al [2] propose a content-based music recommendation system based on their analysis of music emotion by discovering motifs in physiological data using data mining techniques. In [3], Zhao et al present a music therapy system that recommends music based on the quality of users' sleep. Their approach is an EEG (Electroencephalography) signal analysis based approach that measures sleep quality. In [4] Zhao et al present a video recommendation system that recommends videos to watch to users based on affective analysis. They build a facial expression classifier by embedding the process of building compositional Haar-like features into Hidden Conditional Random Fields (HCRFs).

III. DATASET

For training FaceFetch's facial expression recognition engine, we used the Extended Cohn Kanade (CK+) dataset [9]. The CK+ dataset consists of 593 sequences from 123

978-0-7695-4875-3/12 $26.00 © 2012 IEEE
DOI 10.1109/ISM.2012.24

posers aged between 18 and 50. The resolution of each sequence is either 640 x 480 or 640 x 490 with 8-bit grey-scale or 24-bit color values. Each sequence has between 10 and 60 frames. The sequences begin with a neutral frame and end in a peak frame. All the sequences are FACS (Facial Action Coding System) coded by human experts who assign labels to the sequences after studying the FACS information in the peak frame of the sequence.

IV. ALGORITHM AND SYSTEM ARCHITECTURE

We describe the algorithm we used for FaceFetch's facial expression recognition engine in sections A through E and the system architecture for FaceFetch in section F.

A. Facial Expression Recognition Engine for FaceFetch

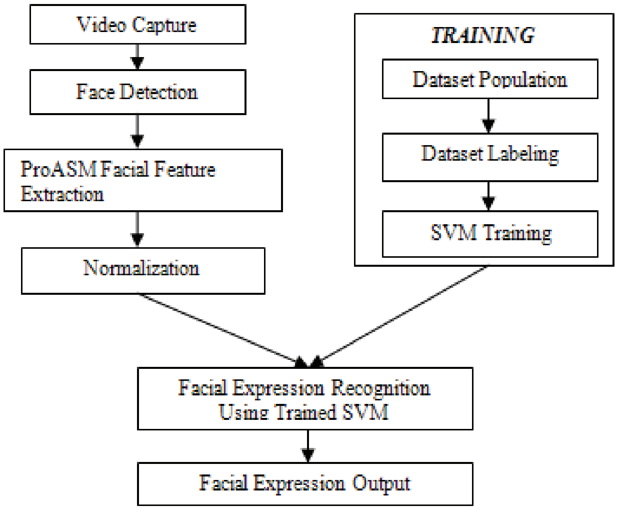


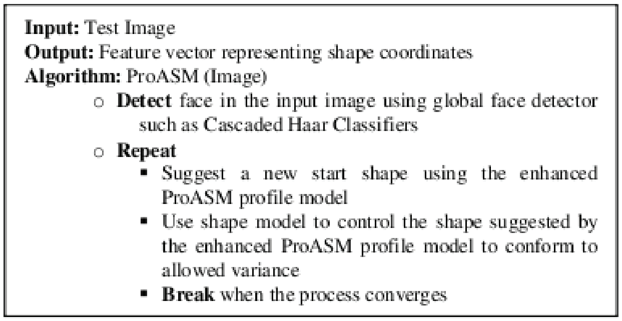
Figure 1. Facial Expression Recognition Algorithm

In the following sections, we present the details of the algorithm used by FaceFetch's facial expression recognition engine. Fig. 1 provides an overview of this engine. We first extracted individual frames from the captured video sequence, and preprocessed the extracted frames. Then a face detection algorithm was applied to the frames to locate the faces. Once the face in the frame was located, we applied our ProASM feature extractor, which extracted features from the face. We then normalized the feature points thus obtained. A feature selection algorithm was then applied to obtain a higher level feature set. We used this higher level feature set to train the SVM (Support Vector Machine) classifier, which was used during testing for recognizing facial expressions of the users in real time.

B. ProASM Feature Extractor

Since run time performance and feature extraction accuracy are key to FaceFetch as a multimedia content recommendation engine, we extracted facial features from the face using our own variant of the Active Shape Models [5] which we call ProASM. ProASM [10] builds an enhanced profile model by incorporating texture differences of neighboring pixels around each landmark along with their intensity values in a way that reduces the point error rate during fitting and results in a decrease in the process convergence time. The profile model is built using a four-step process. (1) Convolution. (2) Normalization. (3) Equalization. (4) Weight Masking. ProASM's enhanced profile model ensures that the initial start shape is more accurate than when using ASM's traditional profile model. Since the shape model now gets more accurate information and a better start shape to work with, it converges faster taking lesser number of iterations than usual. Table 1 gives the ProASM search algorithm.

TABLE 1. PROASM SEARCH ALGORITHM



```
Input: Test Image
Output: Feature vector representing shape coordinates
Algorithm: ProASM (Image)
    ○ Detect face in the input image using global face detector
      such as Cascaded Haar Classifiers
    ○ Repeat
        ▪ Suggest a new start shape using the enhanced
          ProASM profile model
        ▪ Use shape model to control the shape suggested by
          the enhanced ProASM profile model to conform to
          allowed variance
        ▪ Break when the process converges
```

For shape modeling, the object of interest, face in this case, is represented by a set of points. This set of points also known as markups are placed in samples in a consistent fashion manually. Let {(x1,y1), (x2,y2), ... , (xk,yk)} be the landmarks placed on a training image. Then the 2k element vector that represents each image X in the training set is as given in equation (1).

$$Xi=(x1\,,x2\,,\ldots xk\,,y1\,,y2\,,\ldots yk) \qquad (1)$$

Let's say we have m images in the training set. Then we will have m such vectors, each denoted by Xi. This is followed by an alignment phase shown in equation (2), to minimize the sum of distances of each shape to the mean shape.

$$Minimize\,D=\Sigma|Xi-\bar{X}| \qquad (2)$$

During the training process, a base shape and deviations from the base shape were learnt from training examples. This essentially captured the natural variability present within a class. The dimensionality of the data was then reduced from 2k by applying Principal Component Analysis (PCA).

$$Xi\approx\bar{X}+Pb \quad (3)$$

$$b=P^{T}(Xi-\bar{X}) \quad (4)$$

In equation (3), P represents the set of orthogonal basis, and b represents the shape parameters vector of the deformable model. The shape model of the face, represented by a set of points that are constrained according to a Point Distribution Model, deforms in iterations to fit the face during testing. First, a Euclidean transformation as shown in equation (5) was applied. It rotates the point (x, y) by W, scales it by s and translates it by l (x- dimension) and k (y-dimension).

$$T^{(l,k,s,W)}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} l \\ k \end{pmatrix} + \begin{pmatrix} s\cos W & -s\sin W \\ s\sin W & s\cos W \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} \qquad (5)$$

$$Minimize \mid Z - T^{(l,k,s,W)}\,(\bar{X} + Pb) \mid^{2} \qquad (6)$$

Then, the sum of squares between model and the image points are minimized as shown in equation 6. Once the model was fitted with the test image, feature points were located in the test image to obtain the base level feature set. We provide ProASM's performance details in section 5.

C. Normalization

The feature points obtained from the feature extraction phase were normalized for scale, translation and rotation variations. Translation normalization ensured that the feature points across different images were aligned to a common coordinate system. Scaling normalization was important because not all subjects in the video sequences in the dataset were recorded at the same distance from the camera. Besides, different subjects had differently sized faces. We also normalized for rotational variations to account for the variability induced by subjects who tilted their heads towards one side or another when their facial expressions were recorded.

D. Higher Level Feature Extraction

We also extracted some higher level features by applying our higher level feature extraction algorithm. We computed changes in the height and the width of the mouth. We also obtained users' brow activation information and computed changes in the angle of the lips. In addition, the vertical displacement of the feature points representing the corner-most points of the mouth and the vertical displacement of those feature points representing the corner of the nose were computed.

E. Classification

Finally, we used the higher level feature information thus extracted for classification using an SVM classifier. SVM is a maximal margin classifier that tries to find for a set of given data points the hyperplane that best separates them, in the sense that the gap between classes is as wide as it can be. We used SVM polykernel of degree 3 for FaceFetch. We first trained the SVM using the CK+ dataset sequences. During the testing phase, new datapoints were mapped to this space, and given a class label based on which side of the gap they fell on.

F. FaceFetch System Architecture

Fig. 2 shows the system architecture for FaceFetch. A mobile phone or a webcam connected client computer captures live video of the user's face in action. Our facial expression recognition engine then recognizes the user's emotion (denoted by arrows 1 and 2 in Fig. 2). Our web server is then queried by the clients for context-relevant multimedia content based on the recognized emotion parameter. We call it the Level 1 Client Queries (denoted by arrow 3 in Fig. 2). The web server then queries our online cloud servers for content. We call it the Level 2 Server Queries (denoted by arrow 4 in Fig. 2). Alternatively, we could also use online music servers, user video servers and movie database servers for retrieving context-relevant content. The retrieval process is denoted by arrow 5 in Fig.2.

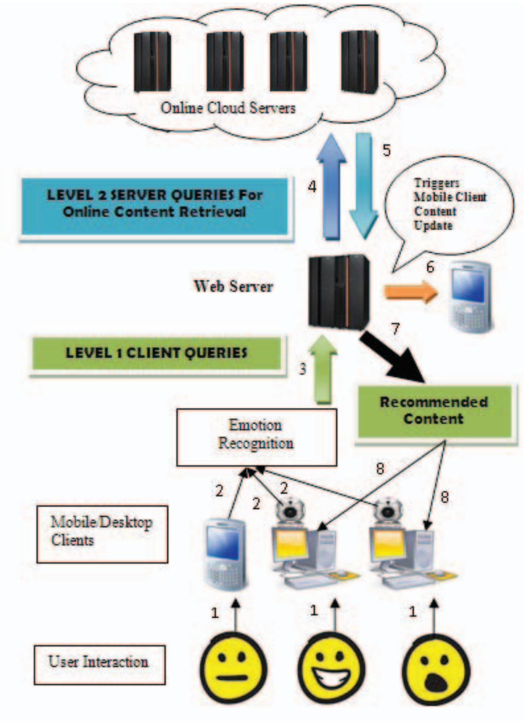


Figure 2. FaceFetch System Architecture

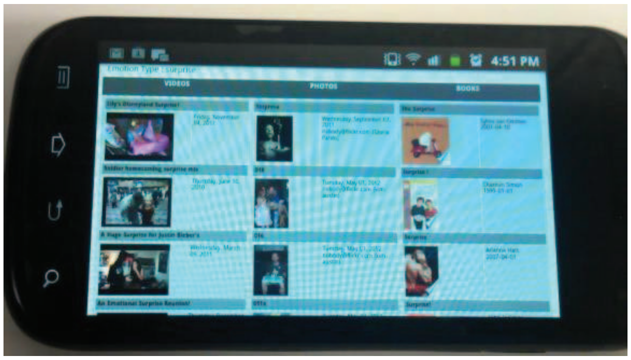


Figure 3. FaceFetch Multimedia Content Recommendation Page

Content is recommended based on user preferences—whether a user prefers to listen to happy songs or melody songs when he is sad. The web server then triggers an update on the mobile phone content, ensuring that the user can carry the recommended content with her on her mobile phone wherever she goes in the spirit of anytime, anywhere ubiquitous computing (denoted by arrow 6 in Fig. 2). The web server also ensures that the desktop clients are updated with the recommended content (denoted by arrows 7 and 8 in Fig. 2). Fig. 3 shows the FaceFetch multimedia content recommendation page. The user is recommended videos, photos, music, news articles and other multimedia content.

V. SYSTEM PERFORMANCE

FaceFetch has moderate hardware requirements. The demonstrator runs on a dual-core 64-bit 2 GHz P6100 Intel Pentium machine with 2 GB of RAM. The machine is connected to a webcam with a video capture resolution of 640 x 480. The demonstrator also has a mobile interface that runs on Android smartphones capable of running Gingerbread. FaceFetch was implemented using C/C++, OpenCV (version 2.1) and the LIBSVM libraries. Table 2 provides point error rate reduction information using our ProASM feature extractor compared to ASM. Our proASM algorithm is more accurate than ASM.

Also, ProASM is more robust to significant illumination variations compared to ASM. Table 3 gives the confusion matrix for FaceFetch's facial expression recognition engine. The rows represent the actual class. The columns represent the predicted class. The accuracy of our SVM-based facial expression recognition system for Leave-One-Out cross validation is 93.3 percent. FaceFetch is very good at recognizing happy, surprise, anger and disgust expressions. Table 4 gives running time benchmarks for FaceFetch modules.

TABLE 2. PROASM VERSUS ASM

Number of Landmarks	**Reduction in Point Error Rate**
28	48.3
68	57.83
82	61.4

TABLE 3. CONFUSION MATRIX

Anger	**Disgust**	**Fear**	**Happy**	**Sad**	**Surprise**	**□Classified as**
0.95	0	0	0	0.05	0	**Anger**
0	0.93	0.07	0	0	0	**Disgust**
0	0.05	0.72	0.18	0	0.05	**Fear**
0	0	0	1	0	0	**Happiness**
0.24	0	0.16	0	0.6	0	**Sadness**
0	0	0.01	0	0	0.99	**Surprise**

TABLE 4. BENCHMARKS

FaceFetch Process/Module	**Time in ms**
Video Acquisition	51
Face Detection	59
ProASM Feature Extraction	81
SVM Classification	70
Content Recommendation	160

VI. CONCLUSION AND FUTURE WORK

In this paper, we presented FaceFetch, a novel facial expression recognition driven context-based multimedia content recommendation system for mobile phones and desktops. FaceFetch received very good response from all the users who tested our system. FaceFetch's ProASM feature extractor is more accurate, faster and more robust to illumination changes than ASM.

REFERENCES

[1] Seungmin Rho, Byeong-jun Han, Eenjun Hwang, "SVR-based Music Mood Classification and Context-based Music Recommendation," In Proc. 17th ACM Int'l Conf. Multimedia (MM 09), 2009.

[2] Rafael Cabredo, Roberto S. Legaspi, Masayuki Numao, "Identifying Emotion Segments in Music by Discovering Motifs in Physiological Data," Proc. 12th Int'l Conf. Music Information Retrieval (ISMIR 11), Int'l Soc. Music Information Retrieval, 2011.

[3] Wei Zhao, Xinxi Wang and Ye Wang, Automated Sleep Quality Measurement using EEG Signal-First Step Towards a Domain Specific Music Recommendation System, In Proc. 18h ACM Int'l Conf. Multimedia (MM 10), 2010.

[4] Sicheng Zhao, Hongxun Yao, Xiaoshuai Sun, Pengfei Xu, Xianming Liu, Rongrong Ji, "Video Indexing and Recommendation Based on Affective Analysis of Viewers," In Proc. 19h ACM Int'l Conf. Multimedia (MM 11), 2011.

[5] Fang-Fei Kuo, Meng-Fen Chiang, Man-Kwan Shan and Suh-Yin Lee, "Emotion-based Music Recommendation By Association Discovery from Film Music", In Proc. 13h ACM Int'l Conf. Multimedia (MM 05), 2005.

[6] Ioannis Arapakis, Ioannis Konstas, Joemon M. Jose, "Using Facial Expressions and Peripheral Physiological Signals as Implicit Indicators of Topical Relevance," In Proc. 17th ACM Int'l Conf. Multimedia (MM 09), 2009.

[7] Roberto Valenti, Alejandro Jaimes, Nicu Sebe, "Sonify Your Face: Facial Expressions for Sound Generation," In Proc. 18h ACM Int'l Conf. Multimedia (MM 10), 2010.

[8] Hornof, A. J., Rogers, T., & Halverson, T. (2007). EyeMusic: Performing live music and multimedia compositions with eye movements. In Proc. Conf. on New Interfaces for Musical Expression (NIME 2007), 2007.

[9] P.Lucey, J. Cohn, T Kanade, J. Saragih, Z. Ambadar, and I. Matthews. The extended Cohn-Kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In IEEE Computer Vision and Pattern Recognition Workshops (CVPRW), 2010.

[10] Srinivasan, R. 2011. Enhanced ASM Using Texture-Based Profile Model. Master's Thesis. University of Texas at Dallas.