Auto-Optimized Multimodal Expression Recognition Framework Using 3D Kinect Data for ASD Therapeutic Aid

Amira E. Youssef, Sherin F. Aly, Ahmed S. Ibrahim, and A. Lynn Abbott

***Abstract*—This paper concerns the automatic recognition of human facial expressions using a fast 3D sensor, such as the Kinect. Facial expressions represent a rich source of information regarding emotion and interpersonal communication. The ability to recognize expressions automatically will have a large impact in many areas, particularly human-computer interaction. This paper describes 2 frameworks for recognizing 6 basic expressions using 3-dimensional data sequences that are captured in real time. Results are presented that demonstrate accuracy levels for the different techniques, and for different methods of preprocessing, registration and classification. We also describe the potential to use such a system for treatment of children with autism spectrum disorders (ASD).**

***Index Terms*—Kinect, SVM, Emotion, ASD.**

I. Introduction

Facial expressions represent an important aspect of human communication, particularly in conveying emotional states. The ability to recognize facial expressions automatically will be very useful for Human-Computer Interaction (HCI). Potential applications include video games, educational software, automobile safety, mental health monitoring, "affective computing," and many others.

This paper describes a system that attempts to recognize facial expressions using a fast three-dimensional (3D) sensor, such as the Kinect. Previous researchers have addressed the subject using different techniques for detection, such as facial action units (AU), wavelets, and scale-invariant feature transform (SIFT) descriptors. For classification, techniques such as support vector machines (SVM), neural networks, and *k*-Nearest-Neighbor (*k*-NN) have all been considered. These will be discussed further in the next section.

Of particular interest is the application of this technology to assist children with autism spectrum disorders (ASD). ASD represents a significant barrier to participation and inclusion in all aspects of society, including education, employment, and social activities. In Baron-Cohen's "mind-blindness" hypothesis [1], those with ASD are thought to have problems intuitively understanding the emotions and minds of others. The system described here could eventually be used to identify and display emotions as a part of ASD therapy.

The next section of this paper presents a brief survey of expression recognition. Section III describes the suggested framework, and Section IV discusses experimental results. Finally, concluding remarks are given in Section V.

Manuscript received January 12, 2013; revised March 19, 2013.

The authors are with the Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, USA, on leave from Scientific Research and Technology Applications City, Alexandria, Egypt (e-mail: amira@vt.edu, sherin@vt.edu, nady@vt.edu, abbott@vt.edu).

II. Expression Recognition Survey

A. Early Studies of Human Emotions

An early method of characterizing the physical expression of emotions is known as the Facial Action Coding System (FACS), which was developed in the 1970s by Ekman and Friesen [2] and is still widely used today. This system is a muscle-based approach in which changes in facial muscles, either individually or combined, cause changes in the expression. FACS allows for the decomposition of any facial expression into Action Units (AUs), which refer to the contraction or relaxation of particular muscle groups in the face or neck.

B. Emotions in Computer Animation

The earliest attempts to apply computer animation to facial expressions were also in the 1970s. In the 1990s, fixed standards became available. In the past several years, facial expression recognition has become a very active field. The most common techniques use AUs as a basis of distinguishing different expressions. Although the portrayal of expressions is universal, the amount of activation of the AUs differs from one person to the next, and from one ethnic group to another.

Lemaire et al. describe a method for facial recognition that is based on localized (region-based) image analysis [3]. In this work as well as many others, the aim is to distinguish 6 primary emotions that were listed by [2]: happiness, sadness, anger, disgust, fear and surprise. Systems that perform facial expression analysis using 3D data can be characterized as static or dynamic. Dynamic systems are sometimes called four-dimensional (4D), where time is the fourth dimension [4]. Detection techniques vary greatly, and include the use of Gabor wavelets [5], SIFT descriptors [6] and quadTree decomposition [7].

C. Data registration

Before applying classification techniques, a preprocessing step is needed in which 3D point sets are aligned, or registered. Many techniques have been proposed to solve the registration problem, including the iterative closest point algorithm (ICP) [8]. However, ICP usually converges to a local minimum, and therefore good initialization is needed. For registration in the system described here, the Procrustes superimposition algorithm was used [9].

DOI: 10.7763/IJMO.2013.V3.247

D. Classification Techniques

For static analysis, naïve Bayes (NB) methods have been used, as well as the Tree Augmented Naïve Bayes (TAN) classifier. For dynamic analysis, hidden Markov models (HMM) have traditionally been used. Anderson and McOwen suggested the use of motion averaging over specified regions of the face to condense the data into an efficient form [10]. They reported that the classifiers worked better with condensed data, rather than with entire data sets.

III. Frameworks for Automated Expression Recognition

We propose two different frameworks to recognize emotions in real time using the Kinect sensor and a modest PC. Any automatic facial expression detection system can be divided into 3 steps: 1) tracking and detection, 2) feature extraction, and 3) expression classification. For the first step, skeletal and facial tracking is performed using the Kinect for Windows SDK, version 1.5, which is based on the Active Appearance Model [11]. For our system, the face tracking engine operates in the range of 4-8 ms per frame. For step 2, feature extraction, we considered complete sets of 131 3D points representing the face (as provided by the SDK), and we also considered reduced data sets as obtained using principal components analysis (PCA) with the 3D point sets.

The final stage of any face expression recognition system is the classification module. We choose to use an auto-optimized SVM classification and compare it to a *k*-NN (*k*-Nearest-Neighbor) classifier. These two classifiers were chosen because of their simplicity and their good performance for a wide variety of applications.

In this work, two main frameworks will be presented. Framework I represents using the SVM classifier on the whole set of facial keypoints, while the second one represents the multimodal SVM applied to upper and lower keypoints separately.

Framework I:

A. Training

The training set contains 4D data for 14 different persons performing the 6 basic facial expressions. The data has been labeled in 7 classes (the 6 main facial expressions plus a class dedicated to the neutral expression).

1) Point Extraction

The Kinect sensor sends out frames consisting of 121 3D points each, representing the main interest points of the user face. Although the Kinect, as any other sensor, drops some frames, the Kinect sends the points in a rate close to 30 frames per second. Each frame is stored as a vector, and is labeled with one of the 7 classes.

2) Data Registration

In the context of data registration, data transformations could be rigid or non-rigid. A transformation is said to be rigid if it preserves relative distances. That is to say, if the sets of points P and Q are transformed to p^* and Q^*, then the distance from P to Q is the same as that from p^* to Q^* [12]. Rigid transformations include rotation, translation and reflection. If a transformation changes the shape and/or size of a shape, then this transformation is non-rigid (i.e. the distance from P to Q is not the same as that from P^* to Q^*). Non-rigid transformation examples include vertical and/or horizontal stretching/shrinking. We employ Procrustes superimposition shape registration technique, that apply a rigid transformation (rotation, translation, and scaling) to one of the point clouds and bring it as close to the other as possible.

3) Pre-Processing

Data reduction is used to improve processing speed and accuracy. In one form of data reduction, one frame was retained from each set of M successive frames. A typical value was $M = 30$. The frame that was chosen exhibited the highest variance as computed for all (x, y, z) values in each frame. This technique is called "reduction" in the next sections.

Another form of data reduction was to remove individual attributes (particular x or y or z components) across all frames. The attributes that were discarded were those having the lowest variance across all of the frames. In our implementation, Approximately 40 attributes were retained for each individual. This is called the "variance" technique in the next sections.

PCA-based dimensionality reduction was also employed in some tests. PCA was used to map the 363 scalar values from each frame to a smaller number of values. In our tests, 18 and 19 dimensions were considered in separate tests.

4) SVM auto-Optimization

Support vector machines were trained using the LibSVM library [13], which supports multi-class recognition systems. Several kernel types were considered, and empirically the best results were obtained using the RBF (radial basis function) kernel. The main parameters to be selected during training are known as cost and gamma. In order to pick these values, the loose grid search method suggested by [14] was used. The SVM auto-optimization module calls the SVM training module iteratively with different cost and gamma values. The auto-optimization module measures the trained SVM model performance and then choses the best values for cost and gamma accordingly. The chosen SVM model is stored to be recalled later in real time.

B. Real-Time Operation

The main operational scenario for the implemented system is to display sample images showing facial expressions to the user. The user should mimic the expression shown on the display. Then the system, using the Kinect sensor and the trained SVM model, should detect the user expression and decide if the user mimicked the expression correctly or not. The main modules used for real time operation of the system are shown in Fig. 1.

Framework II

In this framework the 3D feature points are partitioned into two sets according to their locations on the face. The first set represents the upper part of the face (containing the eyes and eyebrows), while the other set represents the lower part of the face (containing the mouth). Each set is fed to a separate SVM module, and then a third SVM makes the final decision based on the output values from the first two SVMs. The high-level structure is shown in Fig. 2. The motivation was to allow for separate training and processing of the two main

parts of the face.

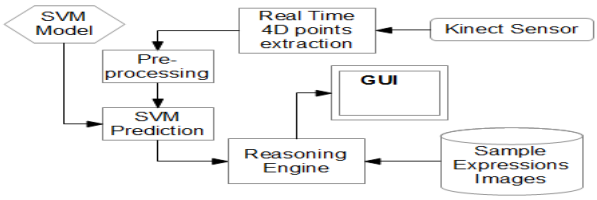

Fig. 1. Real-time operation

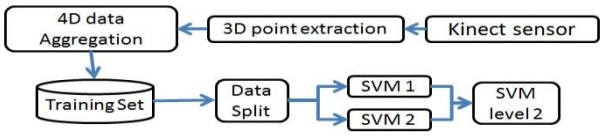

Fig. 2. Face-split framework

IV. Experimental Results

A. Data set Building

We collected a database of 4D data for 14 different subjects performing the 6 basic facial expressions. Considerations were taken so that the data contains subjects from different age groups, different ethnicities, and different types of facial features (particularly facial hair, glasses, and head coverings). Multiple sessions were recorded for each participant. The data has been labeled in 7 classes (the 6 main facial expressions plus a class dedicated to the neutral expression).

The database is divided into training and testing sets. In all of our experiments, no frames from a test set were used for training. In the results reported below, "subset 1" refers to the case that the same set of human subjects provided frames for both the training and testing sets. For "subset 2," almost all subjects in the testing set did not appear in the training set.

B. Training

1) Data registration

Non-rigid registration is applied on the training-set frames to minimize the MSE between pairs of frames. In our method, all frames in the database were aligned with the first frame in the database using Procrustes analysis [9]. A brief description of Procrustes analysis is outlined here for two 3D point clouds:

- Arbitrarily choose a reference shape X (we take the first frame as the reference shape)
- Superimpose all consequent, Y_i, instances to the reference shape, X.
- Compute the mean shape of the current set of superimposed shapes.
- If the Procrustes distance between mean and reference shape is above a threshold τ, set reference to mean shape and continue from step 2. Otherwise terminate.

2) Pre-processing

Pre-processing was performed the same as for Framework I.

3) Using k-NN

For the purpose of comparison, the k-NN classifier was used with the value $k = 1$. All training samples were used with this classifier. This approach was used instead of k-means clustering to avoid potential problems with outliers.

4) Using SVM

As recommended by the SVM documentation, the whole data set is scaled to the rage (-1, 1) to obtain the best performance out of the SVM classifier. After scaling, the labeled training set is presented to the SVM. The SVM kernel is set to RBF, and loops are made to iterate the cost and gamma values. In each iteration the SVM is trained with the set cost and gamma. Then, the trained SVM is tested against the test data set. The highest score determines the best cost and gamma values to build the final SVM model.

C. Testing

The implemented system is not supposed to work in offline mode. But, during the auto-optimization part of the training process the system is tested to determine the best parameters for the SVM. Also the k-NN was tested offline to compare its performance to the SVM. The k-NN could achieve a 81% accuracy for subset 1 with scaling. As shown in Table 1, while The SVM with the optimized parameters achieved better results for subset 2 and also for subset 1 without scaling which are more difficult cases to recognize. The SVM is expected to perform better than k-NN in all cases if the data set available was larger than the one used.

Now for the full seven expressions we had 2 subsets for testing, subset1with the test data for the same subjects as training data and subset2 with different subjects for training and testing.

The experiments showed that the neutral class degrades the system performance. So another experiment has been performed to measure the system performance without the neutral class.

As shown in Table 2, the unreduced dataset generated 69.69% accuracy while it generated 19% and 22.5% after applying PCA to reduce the dimensions from 363 to 18, and 19 respectively. Considering the previous results we experiment with the removal of the neutral class. The results were 38.76% using a single SVM on subset2 and 74.17% using a single SVM on subset1. Table 3 shows the experimental results for the face split framework. As these results are promising, further research is being conducted to enhance it.

TABLE I: K-NN Results (without Face Splitting).

registration	dataset	Accuracy
With scaling	Subset 1	**81.8%**
With scaling	Subset 2	34.0%
without scaling	Subset 1	40.5%
without scaling	Subset 2	17.7%

TABLE II: SVM Results (without Face Splitting).

Pre-processing Technique	registration		dataset	Accuracy
Variance	No registration	Single frame	Subset 1	54.0%
PCA (18)	With scaling	Frame/class	Subset 1	19.0%
PCA (19)	With scaling	Single frame	Subset 1	22.5%
No preprocessing	With scaling	Single frame	Subset 1	69.7%
No preprocessing	Without scaling	Single frame	Subset 1	54.4%
No preprocessing	Without scaling	Frame/class	Subset 1	24.8%
No preprocessing	With scaling	Single frame	Subset 2	35.0%
Reduction (1/30)	With scaling	Single frame	Subset 2	34.6%
Neutral removal	With scaling	Single frame	Subset 2	38.8%
Neutral removal	With scaling	Single frame	Subset 1	74.2%

TABLE III: SVM RESULTS (WITH FACE SPLITTING).

Level	Segment	dataset	Accuracy
1	Top only	Subset 1	31.0%
1	Bottom only	Subset 1	29.0%
1	Top only	Subset 2	25.8%
1	Bottom only	Subset 2	19.8%
2	Combined	Subset 1	**78.6%**
2	Combined	Subset 2	38.8%

V. POSSIBLE USE FOR ASD THERAPY

Those with autism spectrum disorders (ASDs) are known to have problems processing emotional information. They are thought to have problems with intuitively understanding the minds of others. But this does not mean that individuals with autism are incapable of understanding the mental states of others, but possibly some explicit effort is needed. The proposed system will allow them to practice recognizing the expressions of others by repeatedly imitating the expression displayed by the system.

VI. CONCLUSION

This paper has proposed a system for automatic recognition of human facial expressions in real time using the Kinect camera, a modest PC and two different classifiers. We constructed a training set containing 4D data for 14 different persons performing the 6 basic facial expressions. We compared results from the SVM and *k*-NN classifiers. For individuals who did not participate in training of the classifiers, the best accuracy levels were 38.8% (SVM) and 34.0% (*k*-NN). When only considering individuals who did participate in training, however, the best accuracy levels that we observed rose to 78.6% (SVM) and 81.8% (*k*-NN). An important potential application of this work is in helping children with ASD recognize emotions. It is possible that expression recognition combined with computer animation can lead to a powerful interactive tool that will engage children and provide valuable visual feedback.

REFERENCES

[1] U. Frith, "Mind blindness and the brain in autism," *Neuron*, vol. 32, no. 6, pp. 969–979, 2001.

[2] P. Ekman and W.V. Friesen, *Manual for the Facial Action Coding System*, Consulting Psychologists Press, 1977.

[3] P. Lemaire, B. Ben Amor, M. Ardabilian, L. Chen, and M. Daoudi, "Fully automatic 3D facial expression recognition using a region-based approach," in *Proc. 2011 Joint ACM Workshop on Human Gesture and Behavior Understanding*, pp. 53–58, New York, N.Y., 2011.

[4] G. Sandbach, S. Zafeiriou, M. Pantic, and L. Yin. "Static and dynamic 3D facial expression recognition: A comprehensive survey," *Image and Vision Computing*, Oct. 2012.

[5] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, "Coding facial expressions with Gabor wavelets," in *Proc. Third IEEE Conf. Face and Gesture Recognition*, pp. 200-205, Nara, Japan, Apr. 1998

[6] S. Berretti, B. Ben Amor, M. Daoudi, and A. del Bimbo, "3D facial expression recognition using SIFT descriptors of automatically detected keypoints", *The Visual Computer*, vol. 27, no. 11, 2011.

[7] G. Sandbach, S. Zafeiriou, M. Pantic, and D. Rueckert, "A dynamic approach to the recognition of 3D facial expressions and their temporal models," in *Proc. 9th IEEE International Conference on Automatic Face and Gesture Recognition*, March 2011.

[8] P. J. Besl and N. D. McKay, "A method for registration of 3D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 239–254, 1992.

[9] B. Luo and E.R Hancock "Matching point-sets using Procrustes alignment and the EM algorithm," in *Proc. 10th British Machine Vision Conference*, pp. 43–52, 1999.

[10] K. Anderson and P.W. McOwan, "A real-time automated system for the recognition of human facial expressions" *IEEE Trans. Systems, Man and Cybernetics Part B*, vol. 36, no. 1, pp. 96-105, 2006.

[11] M. Zhou, L. Liang, J. Sun, and Y. Wang, "AAM based face tracking with temporal matching and face segmentation," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2010.

[12] O. Bottema and B. Roth, *Theoretical Kinematics*, Dover Publishing, 1990.

[13] LIBSVM. (30 Jan. 2013). *A Library for Support Vector Machines*. [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm/

[14] C. W. Hsu, C. C. Chang, and C. J. Lin, "A practical guide to support vector classification," *Technical Report*, Department of Computer Science, National Taiwan University, 2003.

Amira E.Youssef is a researcher in Scientific Research and Technology Applications City since 2010. During 2002-2010 she worked as a teaching Assistant in the Alexandria Institute for Engineering and Technology. She received M.Sc. and PhD in Electrical Engineering, Faculty of Engineering, University of Alexandria, in 2005 and 2011, respectively. She received B.Sc. from Faculty of Engineering, University of Alexandria in 2002. Her research interests include Image processing, Computer vision and Virtual-reality.

Sherin F. Aly is a Ph.D. student at the Bradley Department of Electrical and Computer Engineering of Virginia Tech since Spring 2012. She received her M.Sc. in Information Technology from Alexandria University in 2010, and her B.Sc. in Computer Science and Automatic Control from Alexandria University in 2005. She was a visiting scholar at the Department of Electrical and Computer Engineering of the University of Miami during the Spring of 2011. She also holds a Teaching Assistant position at the Institute of Graduate Studies and Research (IGSR) of Alexandria University, Egypt, since 2006. Her research interests include Computer Vision, Image processing, Biometrics, and Artificial intelligence.

Ahmed S. Ibrahim He is a Ph.D. student in Bradley department of electrical and computer engineering, Virginia Tech. He also is a Teaching assistance in the faculty of engineering Benha University since 2003. He Received M.Sc. in computer engineering from Faculty of Engineering Benha University in 2007. He received B.Sc. from faculty of engineering Benha University in 2001. His research interests include Image processing, computer Vision and Virtual-reality.

A. Lynn Abbott is an assistant professor in the Bradley Department of Electrical and Computer engineering, Virginia Tech. He received his Ph.D. from University of Illinois in 1990, his Msc from Stanford University in 1981, and his B.S. from Rutgers University in 1980. His research interests include Computer vision, image processing, biometrics, sensing for autonomous vehicles.