Facial Feature Tracking and Expression Recognition for Sign Language

İsmail Arı
Computer Engineering
Boğaziçi University
İstanbul, Turkey
Email: ismailar@boun.edu.tr

Asli Uyar
Computer Engineering
Boğaziçi University
İstanbul, Turkey
Email: asli.uyar@boun.edu.tr

Lale Akarun
Computer Engineering
Boğaziçi University
İstanbul, Turkey
Email: akarun@boun.edu.tr

***Abstract*— Expressions carry vital information in sign language. In this study, we have implemented a multi-resolution Active Shape Model (MR-ASM) tracker, which tracks 116 facial landmarks on videos. Since the expressions involve significant amount of head rotation, we employ multiple ASM models to deal with different poses. The tracked landmark points are used to extract motion features which are used by a Support Vector Machine (SVM) based classifier. We obtained above 90% classification accuracy in a data set containing 7 expressions.**

I. Introduction

There has been a growing interest in extracting and tracking facial features and understanding the performed expression in automatically. Many approaches [1], [2] include three distinct phases: First, before a facial expression can be analyzed, the face detection must be done in a scene. This process is followed by the extraction of the facial expression information from the video and localizing (in static images) or tracking (in image sequences) these features under different poses, illumination, ethnicity, age and expression. The outputs of this process are given as the input for the following step which is the recognition of the expression. This final step is a classification problem where the expression is classified into one of the predefined classes of expressions.

A. Face Detection

In most of the research, face is already cropped and the system starts with tracking and feature extraction. In others, vision-based automated face detectors [3], [4] or pupil tracking with infrared (IR) cameras [5] are used to localize the face. Alternatively, a face detector can be used to detect the faces in a scene automatically [6].

B. Facial Feature Extraction and Tracking

Numerous features were tried in order to make a better recognition of the facial expression. Image-based models rely on the pixel values of the whole image (holistic) [7] or related parts of the image (local) [8]. On the other hand, model-based approaches create a model that best represents the face by using training images [4], [9]. Moreover, difference images are used to find the eye coordinates from the image pairs gathered by IR cameras [10].

Statistical model-based approaches have three main components: "capture", "normalization" and "statistical analysis". In the capture part, one defines a certain number of points (landmarks) on the contour of the object for the shape and uses image warping for the texture. The following shape normalization is done using Procrustes Analysis and texture normalization is done by removing global illumination effects between frames. Finally, Principal Component Analysis (PCA) is performed to analyze the correlations between object shapes or textures and this information is also used for synthesis. Active Shape Models (ASM) and Active Appearance Models (AAM) are two widely used statistical approaches. AAM approach is used in facial feature tracking due to its ability in detecting the desired features [9]. In addition, ASMs - which are the simpler version of the AAMs that only use shape information and the intensity values along the profiles perpendicular to the shape boundary - are also used [11] to extract features. Because of the difficulty in describing different face views using a single model, view-based AAMs are also proposed [12].

C. Facial Expression Recognition

It is a common approach to use variants of displacement based motion feature vectors for expression recognition. Generating a motion displacement vector for each pixel in the image [13], measuring the total amount of motion relative to each of the three axes [14], and converting 3D position data into a representation of motion composed of displacement vectors [15] are some feature extraction methods used in gesture recognition. The extracted features are input to the expression classifier. Some popular machine learning algorithms used in classification are Hidden Markov Models (HMM), Support Vector Machines (SVM), Bayesian networks, decision-tree based classifiers and neural networks [1], [2], [16], [17].

D. Sign Language Scope

Sign language expressions are composed of manual (hand gestures) and non-manual components (facial expressions, head motion, pose and body movements). Some expressions are performed only using hand gestures whereas some change the meaning where a facial expression accompanies hand gestures. Therefore, a robust high-performance facial feature tracker and facial-expression classifier is a must in sign language recognition. Although most of the sign language recognition systems rely only on hand gestures and lack non-manual components of a sign [14], [18], there are also some unified systems [19].

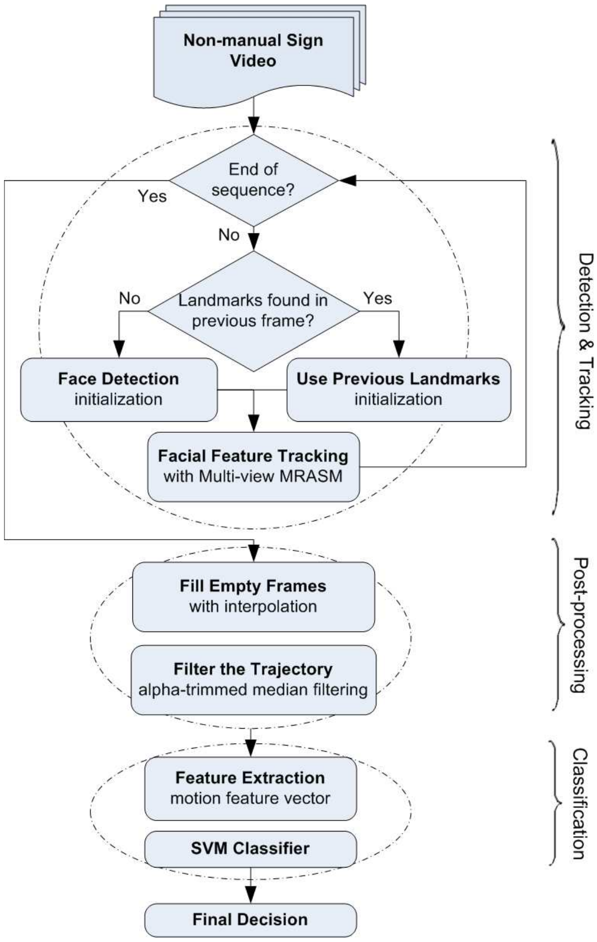

Fig. 1. Flow chart of the proposed system

The subject of this research is feature point based expression recognition in sign videos. The system flow of our approach is illustrated in Figure 1. We employ state-of-the-art techniques such as ASMs for tracking and SVMs for classification to deal with this problem. Since standard ASMs have difficulty dealing with extreme poses [12], we train multiple view-based ASMs to track these difficult poses. Section II describes the details of our tracking approach. Section III gives our expression classification methodology. Section IV gives the properties of the database we used and the experiments we performed on this database with the results are given. We conclude in Section V with relevant discussion.

II. Facial Point Tracking

A. Statistical Analysis of Samples

The statistical analysis of shapes can be divided into three distinct steps; capture, normalization and PCA.

We start with annotating L feature points manually for each of the N randomly captured frames from videos and create the sample shape space $\mathbf{\Phi_s}$ containing shapes $\mathbf{s}_i$ where

$$\mathbf{s}_i = (x_1, y_1, x_2, y_2, \ldots, x_L, y_L)\,,\ i = 1, \ldots, N$$

is a shape containing the coordinates of the landmarks shown in Figure 2.

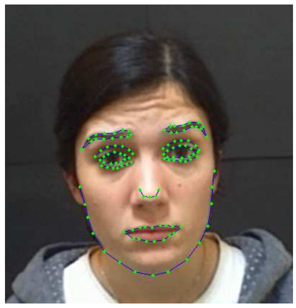

Fig. 2. Selected Feature Points

Secondly, shape normalization is done using Procrustes Analysis where we choose the first shape as the reference shape and align all others to it. Then we recalculate the estimate of the mean from all and repeat the process until convergence.

Finally, PCA is applied to the normalized shape space to find the orthogonal basis vectors satisfying

$$\mathbf{C}_s \mathbf{e}_k = \lambda_k \mathbf{e}_k \tag{1}$$

where

- $\mathbf{C}_s$ is the covariance matrix constructed using normalized shapes
- $\mathbf{e}_k$ is the k^{th} eigenvector
- λ_k is the k^{th} eigenvalue

Any shape $\mathbf{s}$ can also be described using all eigenvectors in a lossless way and the coefficients of eigenvectors form the parameter vector $\mathbf{b}$.

$$\mathbf{b} = \mathbf{E}_{2L}^T (\mathbf{s} - \bar{\mathbf{s}}) \tag{2}$$

$$\mathbf{s} = \bar{\mathbf{s}} + \mathbf{E}_{2L} \mathbf{b} \tag{3}$$

where

$$\bar{\mathbf{s}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{s}_i, \quad \mathbf{E}_{2L} = \left[\ \mathbf{e}_1 \quad \ldots \quad \mathbf{e}_{2L}\ \right]$$

The motivation behind applying PCA is to reduce the dimension and use $K < 2L$ eigenvectors, yet preserving the most of the variation. K is chosen where it satisfies

$$\sum_{k=1}^{K} \lambda_k \geq 0.95 \times \sum_{k=1}^{2L} \lambda_k$$

Let $\lambda_1, \ldots, \lambda_k$ be the first K eigenvectors. Then with $\hat{\mathbf{b}} = (b_1, \ldots, b_K)$, we can synthesize $\hat{\mathbf{s}}$ which is an estimate of $\mathbf{s}$ that is similar to the shapes in $\mathbf{\Phi_s}$.

$$\hat{\mathbf{b}} = \mathbf{E}_K^T (\mathbf{s} - \bar{\mathbf{s}}) \tag{4}$$

$$\hat{\mathbf{s}} = \bar{\mathbf{s}} + \mathbf{E}_K \hat{\mathbf{b}} \tag{5}$$

B. Creating a Model from Landmark Profiles

Let $p_{j,i}$ be the j^{th} landmark in the i^{th} shape, such that $p_{j,i} = (x_{j,i}, y_{j,i})$. $\mathbf{g}_{j,i}$ is the gradient of pixel intensities along the profile of $p_{j,i}$ as in Figure 3.

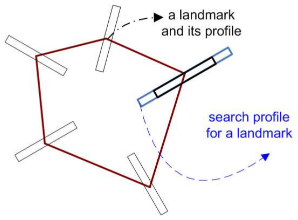

Fig. 3. Profiles of landmarks

Then we calculate $\bar{\mathbf{g}}_j$ as the mean gradient vector and $\mathbf{C}_j$ as the covariance of gradients. Thus a single model is composed of particular $\bar{\mathbf{s}}$, $\mathbf{E}_K$, $\bar{\mathbf{g}}_j$ and $\mathbf{C}_j$ $(j = 1, \ldots, L)$.

C. Fitting a Shape to a Test Image

The initialization is done by detecting the face using OpenCV's face detector [6] and $\bar{\mathbf{s}}$ is placed on the found face. Then the shape is iteratively perturbed along the profile until convergence. Each iteration involves two steps as follows

1) Finding the Best Fit: Let us say n is the profile width and m is the search width as in Figure 3 where $m > n$.

For each landmark, we find the best fit along the profile where the best profile gradient $\hat{\mathbf{g}}_j$ gives the minimum Mahalanobis distance with the model, i.e. the term

$$(\hat{\mathbf{g}}_j - \bar{\mathbf{g}}_j)^T \mathbf{C}_j^{-1} (\hat{\mathbf{g}}_j - \bar{\mathbf{g}}_j) \tag{6}$$

is minimized.

2) Constraining the Best Fit: The best fit is constrained by finding the approximate shape parameters $\hat{\mathbf{b}}$ using Eq. (4) and constraining each coefficient b_k satisfying $-3\sqrt{\lambda_k} \geq b_k \leq 3\sqrt{\lambda_k}$ for $k = 1, \ldots, K$. That is, if the value is out of the allowed limits, then it is changed to the nearest allowed value. Finally, the constrained parameters are projected back to $\hat{\mathbf{s}}$ using Eq. (5).This way, the fitted shape avoids deformation and will be similar to the ones in $\mathbf{\Phi_s}$.

D. Multi-resolution Approach

Instead of using a single level ASM search, a model is created for each level of the pyramid where the original size images are in the lowest level and higher models involve sub-sampled images. The search is first done at the highest level and the found shape is passed to the lower level as the initial shape for that level. So a rough estimate is found in the highest level and fine-tuned at each level it goes through. This procedure is called Multi-resolution ASM (MRASM).

E. Data Refitting

Since $\mathbf{\Phi_s}$ is gathered by clicking feature points manually, the training set is error-prone to human mistakes. To reduce this bias, data refitting is performed. So, a model is trained using $\mathbf{\Phi_s}$. Then, each shape is initialized by its own shape and refitted using this model. Finally, a new model is trained by using the fitted shapes as described in [20].

F. Tracking in an Image Sequence

Since there is head motion in addition to facial expressions in sign videos, a single view model is not sufficient for handling all views of the face. So, three different models are trained which are for frontal, left and right views. Sample images are shown in Figure 4. We use two metrics to calculate the spatial and temporal errors to choose the best fitting model during tracking.

Fig. 4. Three different views

$rms_{g,frontal}$, $rms_{g,left}$ and $rms_{g,right}$ are the root mean square errors between the fitted shape's profile gradient values and the used model's (frontal, left or right) mean gradient values. So, they give a similarity measure for us to decide on a model view.

$rms_{f,f-1}$ is the root mean square error of shapes found in f^{th} and $(f-1)^{th}$ frame, where $f = 2, \ldots, F$. It informs us about the change in shape in the given interval. We assume that the first frame is frontal, so we start with a single view MRASM search for $f = 1$. Then the algorithm is

for $f = 2, \ldots, F$ **do**
  for *each model (frontal, left, right)* **do**
    apply MRASM search;
  end
  eliminate the models giving $rms_{g,model} > threshold$;
  if *no model remains* **then**
    mark this frame as empty (not found);
    re-initialize the tracker;
  else
    choose the model giving the minimum $rms_{f,f-1}$;
    set the found shape as initial shape for next iteration;
  end
end

Finally, the empty frames are filled by interpolation and α-trimmed mean filter is applied to eliminate the spikes encountered during tracking.

III. EXPRESSION CLASSIFICATION

Classification is the final stage of the facial expression recognition system and consists of two sub-stages which are motion feature extraction and classification using SVM. Since the data have the nature of a time series, one needs to either normalize it or extract features that would represent the whole sequence. Motion history images used by Michel and Kaliouby [16] is one such feature. Here, we follow a similar approach and use maximum displacements. The tracker extracts the coordinates of facial landmarks in consecutive frames of the video sequences. Then these coordinates are used to evaluate the maximum displacement values for each feature point in four directions x_+, x_-, y_+ and y_- across the entire image sequence.

A. Motion Feature Extraction

We use displacement based and time independent motion feature vector as the input to the SVM classifier. The motion feature vector includes information about both the magnitude and the direction of motion for each landmark.

A similar displacement based approach has been applied in [16] to extract the motion information between the initial and the "peak frames" in image sequences. In our study, we find the maximum displacement of points where "peak location" for each point may be in different frames.

We define $\mathbf{V}^i$ as the i^{th} video

$$\mathbf{V}^i = \left[\mathbf{s}_1^i, \ \ \mathbf{s}_2^i, \ \ \cdots \ \ \mathbf{s}_F^i\right] \tag{7}$$

where

$$\mathbf{s}_f^i = \left(x_f^{i,1} \ \ y_f^{i,1} \ \ x_f^{i,2} \ \ y_f^{i,2} \ \ \cdots \ \ x_f^{i,L} \ \ y_f^{i,L}\right) \tag{8}$$

is the set of landmarks in the f^{th} frame of i^{th} video.

For each video, the initial frame ($\mathbf{s}_1$) is chosen as the reference frame and the displacements of the points between each frame and the reference frame have been measured.

Then, the maximum displacement values of each point in four directions have been chosen as the motion features.

$$dx_{max}^{i,l} = \max_f \left\{x_f^{i,l} - x_1^{i,l}\right\}$$

$$dx_{min}^{i,l} = \min_f \left\{x_f^{i,l} - x_1^{i,l}\right\}$$

$$dy_{max}^{i,l} = \max_f \left\{y_f^{i,l} - y_1^{i,l}\right\}$$

$$dy_{min}^{i,l} = \min_f \left\{y_f^{i,l} - y_1^{i,l}\right\}$$

The output of this process is a single motion vector $\mathbf{z}^i$ for each video.

$$\begin{aligned}\mathbf{z}^i = \Big(&dx_{max}^{i,1} \cdots \ dx_{max}^{i,L} \ dx_{min}^{i,1} \ \cdots \ dx_{min}^{i,L} \\ &dy_{max}^{i,1} \ \cdots \ dy_{max}^{i,L} \ dy_{min}^{i,1} \ \cdots \ dy_{min}^{i,L}\Big)\end{aligned} \tag{9}$$

B. Classification Using SVM

Because of the superior classification performance and its ability to deal with high dimensional input data, SVM is the choice of the classifier in this study for facial expression recognition. A motion feature vector is extracted in the previous stage and classified into one of the predefined expression classes in this stage. A brief definition of SVM is given below.

Given a set of training data pairs (x_i, y_i), $y_i \in \{+1, -1\}$, the aim of the SVM classifier is to estimate a decision function by constructing the optimal separating hyperplane in the feature space [21]. The key idea of SVM is to map the original input space into a higher dimensional feature space in order to achieve a linear solution. This mapping is done using kernel functions. Final decision function is in the form:

$$f(x) = \left(\sum_i \alpha_i y_i K(x_i \cdot x) + b\right) \tag{10}$$

where $K(x_i \cdot x)$ is the Kernel transformation. The training samples whose Lagrange coefficients α_i are non-zero are called "support vectors" (SV) and the decision function is defined by only these vectors.

IV. Experiments & Results

A. Used Database

We used the database of Aran et al. [17] which involves 8 different classes of non-manual sign videos. The classes (as shown in Figure 5) are briefly

1) neutral expression
2) head shaking to left and right (used for negation)
3) head up, eyebrows up (used for negation)
4) head forward (used when asking a question)
5) lips turned down, eyebrows down (sadness)
6) head up and down (used for agreement)
7) lips turned up (happiness)
8) classes 6 and 7 performed together (happy agreement)

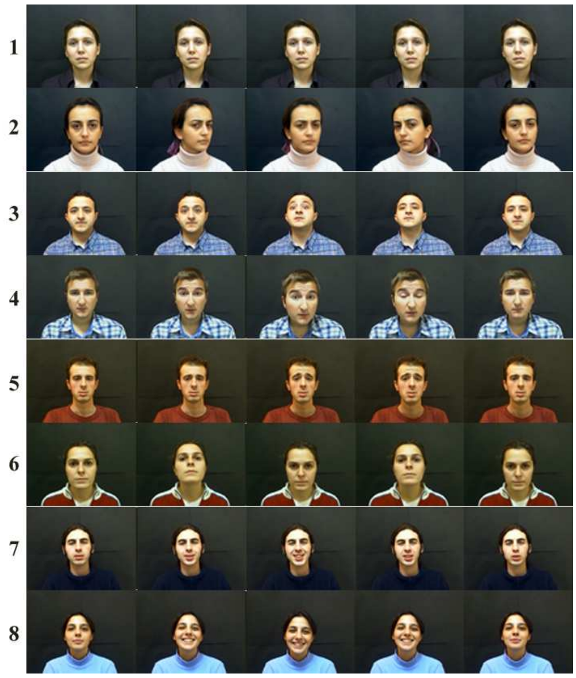

Fig. 5. The classes in the database

There are 5 repetitions for each class from 11 (6 female, 5 male) different subjects and each video takes about 1-2 seconds.

B. Facial Point Tracking

We selected 2 subjects and trained frontal, left and right models for each. So, we trained a person-specific multi-view ASM from 35 frontal, 15 left and 15 right images per subject. These images were randomly selected from videos and manually annotated. A few (8-10) eigenvectors seemed to be enough to describe 95% of the total variation in each model (Figure 6).

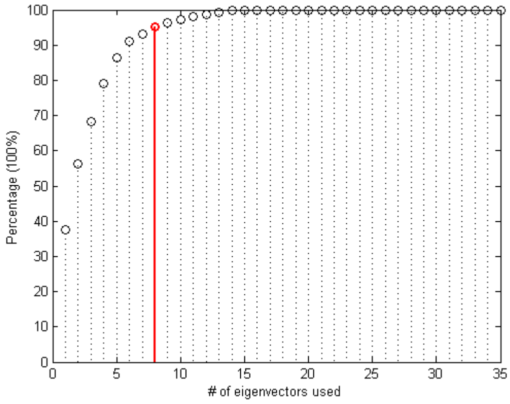

Fig. 6. Eigenvector contributions to the total variance

For MRASM search, we used 3 levels and $n = \{13, 9, 9\}$ and $m = \{15, 13, 13\}$ are the parameters we chose for $L = \{0, 1, 2\}$ respectively. For each level, at most 4 iterations are allowed and the convergence ratio r is selected as 90% of the number of landmarks.

Sample tracking results can be seen in Figures 7 and 8 where the index on the top right stands for the frame number in that video.

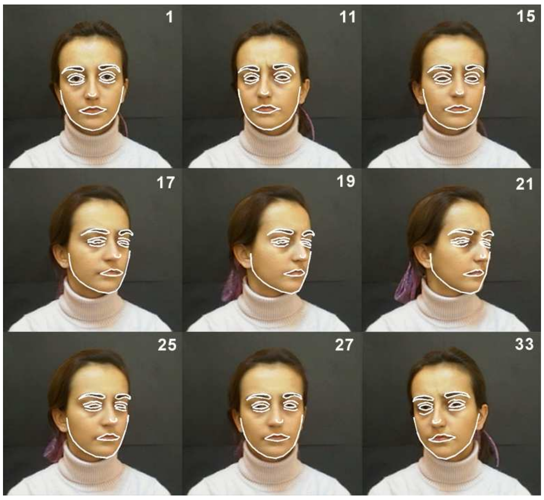

Fig. 7. Found landmarks in a video that belongs to 2^{nd} class

C. Expression Classification

We performed the classification experiments by LIBSVM [22]. Gaussian kernel has been the kernel choice and the kernel parameters *cost* and *gamma* have been searched in the ranges $[2^{-5}, 2^{15}]$ and $[2^{-15}, 2^{3}]$ respectively.

We prepared the following sets for our experiments:

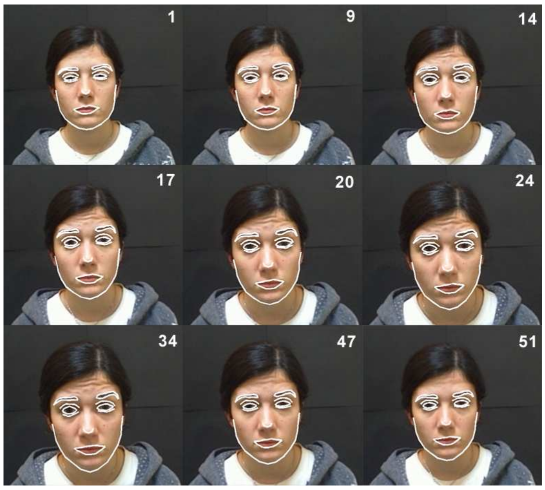

Fig. 8. Found landmarks in a video that belongs to 3^{rd} class

- $\mathbf{\Phi_1}$ and $\mathbf{\Phi_2}$: Each involves 7 classes and 5 repetitions for a single subject found with the tracker. The class numbers are 2 to 8. (35 samples each)
- $\mathbf{\Phi_{1,2}}$: Involves 7 classes and 5 repetitions for two subjects found with the tracker. The class numbers are 2 to 8. (70 samples)
- $\mathbf{\Phi_{gt}}$: Involves 4 classes and 3 repetitions for each of 9 subjects (excluding the subjects tracked) in the ground truth data. The class numbers are 2, 3, 4 and 7. (108 samples)

Then we designed three tests on these sets:

I- Use $\mathbf{\Phi_1}$ for training and $\mathbf{\Phi_2}$ for testing and vice versa.
II- Use $\mathbf{\Phi_{1,2}}$ for training and testing by 5-fold cross-validation.
III- Use $\mathbf{\Phi_{gt}}$ for training and $\mathbf{\Phi_{1,2}}$ for testing.

Tests I and II are performed using both 4 and 7 classes to compare with Test III.

The accuracy results found with the best parameters are shown in Table I.

It is observed that the best accuracy is obtained when training data includes instances of the test subject (Test II). When we train the system with a different person, performance drops (Test I). However, when the number of subjects in the training set is increased, high performance person-independent expression classification is possible (Test III). Since we have tracking results of only two subjects, we have used hand-labeled landmarks in the training set of Test III. Performance is expected to increase further if tracked landmarks are used instead.

The confusion matrix showing the results (in percentage) for the worst case are given in Table II. Overall accuracy is 67.1%. It is observed that the most confused expressions are classes 5-7 (happiness-sadness) and 6-8 (agreement-happy agreement).

V. Conclusion

In this study, we presented an automatic facial expression and head motion recognition system. First, we have tracked

TABLE I

EXPRESSION CLASSIFICATION RESULTS

Dataset	Training Options	# expression classes	# training samples	# test samples	Optimized Gaussian Kernel Parameters		Accuracy (%)
					Cost	Gamma	
Automatically Tracked Features	Test I	4	20	20	32	0.00001	**95.0**
		7	35	35	32	0.001	67.1
	Test II	4	20	20	4	0.001	**96.6**
		7	35	35	16	0.01	92.5
Ground Truth Features	Test III	4	108	24	16	0.0001	**91.6**

TABLE II

CONFUSION MATRIX FOR 7 EXPRESSION CLASSES (TEST I)

Class	2	3	4	5	6	7	8
2	**1.0**	0	0	0	0	0	0
3	0	**1.0**	0	0	0	0	0
4	0	0	**0.7**	0	0	0.1	0.2
5	0	0	0	**0.5**	0	0.5	0
6	0	0	0	0.1	**0.5**	0	0.4
7	0	0	0	0.1	0	**0.9**	0
8	0	0	0.1	0.3	0.5	0	**0.1**

facial feature points in prerecorded non-manual sign videos by using multi-view MRASM. Then maximum displacement of each landmark have been used as a component of motion feature vector and these vectors are given to SVM classifier as input.

In the tracking part, we used a single subject 3-view MRASM (frontal, left and right). The results were promising and we showed a way to track facial feature points robustly. The proposed tracking system is easily extendible by training the model for more than three views and more subjects. Since it is based on ASM, the time complexity is less than AAM approach because it uses only the intensity values along the landmark profiles.

We have tested tracker and SVM based classifier on 7 expressions from Turkish Sign Language. We have obtained above 90% person independent recognition accuracy. Our future work will focus on making the system work in real time and integrate the expression classification in our sign language recognizer.

ACKNOWLEDGMENT

This study has been supported by TUBİTAK under project 107E021.

REFERENCES

[1] M. Pantie and L. Rothkrantz, "Automatic analysis of facial expressions: the state of the art," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 12, pp. 1424–1445, 2000.

[2] B. Fasel and J. Luettin, "Automatic facial expression analysis: A survey," *Pattern Recognition*, vol. 36, no. 1, pp. 259–275, 2003.

[3] L. Jordao, M. Perrone, J. Costeira, and J. Santos-Victor, "Active face and feature tracking," *Proceedings of the 10th International Conference on Image Analysis and Processing*, pp. 572–577, 1999.

[4] J. Ahlberg, "An active model for facial feature tracking," *EURASIP Journal on Applied Signal Processing*, vol. 2002, no. 6, pp. 566–571, 2002.

[5] Y. Zhang and Q. Ji, "Active and dynamic information fusion for facial expression understanding from image sequences," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 5, pp. 699–714, 2005.

[6] I. Inc, "The opencv open source computer vision library."

[7] X. Wei, Z. Zhu, L. Yin, and Q. Ji, "A real time face tracking and animation system," *Computer Vision and Pattern Recognition Workshop, 2004 Conference on*, pp. 71–71, 2004.

[8] J. Lien, T. Kanade, and J. Cohn, "Automated facial expression recognition based on facs action units," *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*, pp. 390–395, 1998.

[9] T. Cootes, G. Edwards, and C. Taylor, "Active appearance models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 681–685, 2001.

[10] A. Kapoor and R. Picard, "A real-time head nod and shake detector," *Proceedings of the 2001 workshop on Percetive user interfaces*, pp. 1–5, 2001.

[11] G. Votsis, A. Drosopoulos, and S. Kollias, "A modular approach to facial feature segmentation on real sequences," *Signal Processing: Image Communication*, vol. 18, no. 1, pp. 67–89, 2003.

[12] T. Cootes, G. Wheeler, K. Walker, and C. Taylor, "View-based active appearance models," *Image and Vision Computing*, vol. 20, no. 9-10, pp. 657–664, 2002.

[13] T. Gee and R. Mersereau, "Model-based face tracking for dense motion field estimation," *Proceedings of the Applied Imagery and Pattern Recognition Workshop*, 2001.

[14] P. Vamplew and A. Adams, "Recognition of sign language gestures using neural networks," *Australian Journal of Intelligent Information Processing Systems*, vol. 5, no. 2, pp. 94–102, 1998.

[15] H. Hienz, K. Grobel, and G. Offner, "Real-time hand-arm motion analysis using a single video camera," *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, pp. 323–7, 1996.

[16] P. Michel and R. El Kaliouby, "Real time facial expression recognition in video using support vector machines," *Proceedings of the 5th international conference on Multimodal interfaces*, pp. 258–264, 2003.

[17] O. Aran, I. Ari, A. Guvensan, H. Haberdar, Z. Kurt, I. Turkmen, A. Uyar, and L. Akarun, "A database of non-manual signs in turkish sign language," *Signal Processing and Communications Applications, 2007. SIU 2007. IEEE 15th*, pp. 1–4, 2007.

[18] S. Ong and S. Ranganath, "Automatic sign llanguage analysis: A survey and the future beyond lexical meaning," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 6, pp. 873–891, 2005.

[19] O. Aran, I. Ari, A. Benoit, A. Carrillo, F. Fanard, P. Campr, L. Akarun, A. Caplier, M. Rombaut, and B. Sankur, "Sign language tutoring tool," *Proceedings of eNTERFACE 2007, The Summer Workshop on Multimodal Interfaces*, vol. 21, 2006.

[20] R. Gross, I. Matthews, and S. Baker, "Generic vs. person specific active appearance models," *Image and Vision Computing*, vol. 23, no. 12, pp. 1080–1093, 2005.

[21] C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[22] C. Chang and C. Lin, "Libsvm: a library for support vector machines," 2001.