Tracking Facial Features under Occlusions and Recognizing Facial Expressions in Sign Language

Tan Dat Nguyen and Surendra Ranganath
Department of Electrical and Computer Engineering
National University of Singapore
4 Engineering Drive 3, Singapore 117576
{ntdat, elesr}@nus.edu.sg

Abstract

This paper presents work towards recognizing facial expressions that are used in sign language recognition. Facial features are tracked to effectively capture temporal visual cues on the signer’s face during signing. A Bayesian framework is proposed as a feedback mechanism to the Kanade-Lucas-Tomasi (KLT) tracker for reliably tracking facial features in the presence of head motions and temporary occlusions by hand. This mechanism relies on a set of face shape subspaces learned by Probabilistic Principal Component Analysis with an update scheme to adapt to persons with different face shapes. The results show that the proposed tracker can track facial features with large head motions, substantial facial deformations, and temporary facial occlusions by hand. The tracked results were input to a recognition system comprising HMMs and a NN to recognize four common American Sign Language facial expressions.

1. Introduction

The deaf communicate through sign language (SL). This mode of communication has two channels: manual and non-manual. In the manual channel, hands are used to express lexical meanings, while in the non-manual channel, people use facial expressions, head and upper body movements to express syntactic and semantic information. Non-manual expressions usually co-occur with manual signs.

Our goal is to construct a system which can translate signed American Sign Language (ASL) sentences to English. Here, we consider grammatical marker expressions which provide the grammatical structure to a sentence. There are seven types of grammatical markers [1]: Wh-question, yes/no question, rhetorical question, topic, conditional clause, relative clause, and negation. Each type is commonly composed of a combination of facial feature movements and head motions.

To recognize facial expressions and head motions in SL, facial features are tracked in video sequences of the signer’s face when he/she is signing. Here, we have to deal with head pose variations, and occlusions caused by the hand during signing. Recent works on tracking facial features used sets of Active Shape Models to constrain face shapes and also considered head motions [15, 8]. However, these active models rely on matching by global optimization which are sensitive to occlusions as the number of model points is large. Local Active Appearance Models used in [5] can overcome this weakness, but they did not make use of the correlation of face shape deformations during tracking. Tian et al. [13] used KLT to track facial feature points, but their 2D local models for shape constraints which were based on frontal face might not cope well under varying head pose. There are only a few works on tracking facial features for recognizing facial expressions in SL such as [16, 3]. However, the 3D model used in [16] is computationally expensive and sensitive to noise, and the anthropometric 2D model used in [3] might not be suitable when 3D head rotations occur. Also, few of these works addressed the issue of coping with face shapes of different people.

The research works on automatic SL analysis were reviewed in [10], from which it appears that there is little research which focuses on the non-manual channel, particularly facial expressions. Some of these works only considered feature extraction [2, 16]. The work reported in [3] only tried to recognize two brow states and head shake.

For tracking during SL communication, it is advantageous to use a sparse set of local feature points on the face which can be independently tracked; if there is temporary, local occlusion by hand, any mistracking will be limited to the occluded features, and will not have a global effect. The KLT tracker is suitable from this point of view, but SL facial communication involves not only occlusions, but also facial expressions and natural head motions. These issues can cause gradients in the vicinity of tracked feature points to
change abruptly and present a challenging tracking scenario for the KLT tracker. In this paper, we propose a tracking scheme which uses the KLT tracker and augments it with a corrective feedback mechanism, with the idea of enabling the tracker to cope better with rich facial expressions, head motions, and temporary occlusions. For this, we impose global shape and deformation constraints on the tracked facial features, and integrate these constraints into a Bayesian framework, utilizing eigenspaces obtained by Probabilistic Principal Component Analysis (PPCA) [9] and probabilistic transitions between these subspaces. An update scheme called Incremental PPCA suggested in [8] was adopted to improve the robustness of this framework to face shapes of different people.

We also propose a multichannel temporal framework to recognize common grammatical facial expressions used in SL. For this, appropriate distance measures are derived from tracked facial features to be invariant to natural head motions, and are input to a set of Hidden Markov Models (HMM) to evaluate likelihood of characteristic facial feature motions. These likelihoods were input to a Neural Network (NN) to identify four common grammatical expressions: Yes/no question (YN), Wh question (WH), Topic (TP), and Negation (NEG).

2. Tracking framework

For capturing facial expressions in SL, 21 facial feature points (red dots shown in Fig. 1a) are tracked using the KLT method and a corrective mechanism which uses a Bayesian framework inspired by [6, 12]. We implemented this framework using a prelearned model for deformation of face shapes and deformation history during tracking to provide suitable correction to the KLT.

2.1. Bayesian framework

Let facial shapes be represented by a vector of N feature points \( Z = \{(x_i, y_i), i = 1, ..., N\} \) where \( x_i, y_i \) are the coordinates of the \( i \)th feature point, and let \( Z_t \) be the observation vector of the shape at frame \( t \) of a video sequence. Let \( S \) be the face shape space, and assume that \( S \) is composed of \( M \) subspaces denoted by \( S^1, S^2, ..., S^M \).

\( Z_t \) can be considered to be a noisy version of a corresponding face shape \( X_t \in S \), in other words, \( \exists i \in \{1, 2, ..., M\}, X^i_t \in S^i, X^i_t = F_P(Z_t, S^i) \), where \( F_P \) is a function projecting \( Z_t \) onto subspace \( S^i \).

The probability of the current interpretation \( X^i_t \) by the model, based on the tracking results \( Z_{0:t} \), is computed using Bayes theorem:

\[
P(X^i_t|Z_{0:t}) = \frac{P(Z_t|X^i_t, Z_{0:t-1})P(X^i_t|Z_{0:t-1})}{P(Z_t|Z_{0:t-1})} \tag{1}
\]

Using the conditional independence assumption and assuming

\[
P(X^i_t|X^j_{t-1}) = P(S^i|S^j), P(Z_t|X^i_t) = P(Z_t|S^i) \tag{2}
\]
gives

\[
P(X^i_t|Z_{0:t}) = \frac{P(Z_t|S^i)}{P(Z_t|Z_{0:t-1})} \sum_{j=1}^{M} P(X^j_{t-1}|Z_{0:t-1})P(S^i|S^j) \tag{3}
\]

In the above equation, the normalizing factor in the denominator can be computed as:

\[
P(Z_t|Z_{0:t-1}) = \sum_{i=1}^{M} P(Z_t|S^i)P(X^i_t|Z_{0:t-1}) \tag{4}
\]

We use PPCA to estimate \( P(Z_t|S^i) \), and estimate subspace transition probabilities \( P(S^i|S^j) \) from training data.

2.2. Construction of face shape subspaces

These subspaces are learned by estimating a PPCA mixture model [14] from training data which consists of face shapes obtained at every frame from training video sequences. PPCA groups similar face shapes and head poses into a cluster, and allows good generalization of face shape deformation across different expressions. Embedding all face shapes into one manifold as in [4] may face difficulties from the variations of head poses.

We estimate the number of subspaces using a hierarchical clustering method using Ward’s linkage measure to merge two clusters together. The PPCA mixture model is computed using the EM algorithm suggested in [14]. The information learned by PPCA is the probabilistic intrinsic deformation of face shapes, which allows computing the projection vector \( X^i_t \), and the data distribution \( P(Z_t|S^i) \) in the Bayesian Eq. 3.

2.3. Transition matrix

The transition matrix between any two subspaces is learned following the training steps below:

1. Training data are face shapes \( Z = \{z_{k,t}\} \) extracted from video sequences, where \( k \) is the index of the sequence, and \( t \) is the index of the frame belonging to that sequence.

2. Assign samples \( z_{k,t} \) to subspaces and denote \( z^i_{k,t} \) to be \( z_{k,t} \) assigned to the subspace \( S^i \). This is computed as

\[
z^i_{k,t} = \arg \max_j P(z_{k,t}|S^j) \tag{5}
\]

3. The transition probability from \( S^i \) to \( S^j \) is computed as:

\[
P(S^j|S^i) = \frac{\text{Count}(\{z^i_{k,t}, z^j_{k,t+1}\})}{\text{Count}(\{z^i_{k,t}\}, \forall k, t)} \tag{6}
\]
which is the fraction of the number of \( i \to j \) transitions relative to the total number of samples in \( S^i \).

### 2.4. Updating of face shape subspaces

To deal with problems arising from person-dependence, we updated the learned subspaces, particularly their means and covariances, by new observations during tracking by utilizing the incremental PPCA method proposed in [9].

### 2.5. Tracking algorithm

The overall tracking algorithm is as follows:

1. The observation \( z_t \) is obtained using the KLT algorithm.
2. Find \( x_t \) (using Eq.3 to compute \( P(X_t^j|Z_{0:t}) \)):
   
   \[
   i = \underset{j}{\arg \max} P(X_t^j|Z_{0:t}), \quad x_t^i = M_t^{-1}W_t^T(z_t - \mu_t^i)
   \]

3. Reconstruct \( \tilde{z}_t \) from \( x_t^i \):
   
   \[
   \tilde{z}_t = W_t^i(W_t^{iT}W_t^i)^{-1}M_t^i x_t^i + \mu_t^i
   \]
   where \( M_t^i, W_t^i, \) and \( \mu_t^i \) are parameters of the \( i^{th} \) subspace obtained by PPCA.

4. Repeat steps 1-3 with \( \tilde{z}_t \) as the initial input for the KLT algorithm until either of the following stop conditions is met:
   - The Euclidean distance \( ||z_t - \tilde{z}_t|| < \epsilon \), where \( \epsilon \) is a predefined threshold.
   - A maximum number of iterations is exceeded.

5. Update the chosen subspace \( S_t^i \) with the last observation \( z_t \) using the method in Section 2.4.

### 3. Recognition Framework

#### 3.1. Features

Coordinates of the tracked facial features are used to compute distances used for classification by HMM. Following is the list of these distances (as shown in Fig. 1a)

- **Seven eyebrow parameters:** Left inner brow height (\( B_{IL} \)), Right inner brow height (\( B_{IR} \)), Left middle brow height (\( B_{ML} \)), Right middle brow height (\( B_{MR} \)), Left outer brow height (\( B_{OL} \)), Right outer brow height (\( B_{OR} \)), Distance between brow (\( B_B \)).
- **Four eye parameters:** Left top eye height (\( E_{TL} \)), Right top eye height (\( E_{TR} \)), Left bottom eye height (\( E_{BL} \)), Right bottom eye height (\( E_{BR} \)).

The reference line is defined as the line passing through the two inner eye corners. The height parameters are the perpendicular distances of the feature points from the reference line. The distances between the eyebrows and the width of lips are also computed. All distance parameters are normalized with respect to their corresponding values in the first frame to remove scaling effects across video sequences.

To recognize head motions, tracks of non-deformable facial feature locations are used, and their mutual distances and motions are utilized for classification. Features used to classify head motion include:

- The distance between eye inner corners (\( E_{E} \)).
- Two motion vectors\(^1\) (\( v_{e1} \) and \( v_{e2} \)) of two eye inner corners (\( E_{L3} \) and \( E_{R3} \)).
- Two motion vectors (\( \tilde{v}_{n1} \) and \( v_{n3} \)) of two nose corners (\( N_3 \) and \( N_3 \)).

Motion vectors of tracked features are normalized by the distance between two eye inner corners \( \tilde{v}_i^j = \frac{\tilde{v}_i^j}{\sum_{i=0}^n v_i^j} \), where \( v_i^j \) is the \( j^{th} \) motion vector in \( i^{th} \) frame; \( E_{E}^0 \) is the distance between eye inner corners in the first frame.

These features are used to recognize three classes of head motions: Head Downward, Head Shake, and Head Forward.

#### 3.2. Multichannel temporal framework for recognition

Many types of expressions are used in ASL including the six universal expressions [1]. We are interested in developing a framework that can be extended to recognize not only grammatical expressions but other expressions also. With this motivation in mind, we use a multichannel temporal recognition framework to identify facial expressions in two stages (Fig. 1b). Stage 1 evaluates the likelihoods of facial feature movement categories and head motions using HMMs, and stage 2 identifies the facial expression using a NN whose input vector is formed from likelihood outputs of the HMMs. The four expressions (shown in Fig. 2) which we consider here are described in terms of certain facial feature movement categories and head motions as listed in Table 1. Seven HMMs are trained for seven corresponding facial feature movements and head motions: Brow Knit, Widen, Raise, Forward, Squint, Downward, and Shake.

### Table 1. Description of YN, Wh question, Topic, and Negation in SL.

<table>
<thead>
<tr>
<th>SL expressions</th>
<th>Brow</th>
<th>Eye</th>
<th>Head</th>
</tr>
</thead>
<tbody>
<tr>
<td>YN</td>
<td>Raise</td>
<td>Widen</td>
<td>Forward</td>
</tr>
<tr>
<td>WH</td>
<td>Knit</td>
<td>Squint</td>
<td>Forward</td>
</tr>
<tr>
<td>TP</td>
<td>Raise</td>
<td>Widen</td>
<td>Downward</td>
</tr>
<tr>
<td>NEG</td>
<td>Knit</td>
<td>Squint</td>
<td>Shake</td>
</tr>
</tbody>
</table>

\(^1\)Motion vector \( v_i^{j+1} = (x_i^{j+1}, y_i^{j+1}) - (x_i^j, y_i^j) \)
Brow Raised, Eye Widen, Eye Squint, Head Forward, Head Downward, and Head Shake. Inputs to these HMMs are distance parameters listed in Section 3.1. The HMMs have different topologies because of the difference in nature of the facial feature movements and head motions. A mixture of Gaussians was chosen for the probability density function for observation at each state of all the HMMs. Training of HMMs followed the Baum-Welch re-estimation algorithm [11].

We used a three-layer feed forward NN with seven nodes in the input layer corresponding to the output of seven HMM classifiers for four facial feature movements (Eyes Widen, Eye Squint, Brow Raise, and Brow Knit) and three head motions (Head Forward, Head Downward, and Head Shake). The output layer had four nodes for YN, WH, TP, and NEG. This NN was trained using Levenberg-Marquardt optimization.

4. Experiments

We conducted two sets of experiments to evaluate the performance of the proposed tracker. In the first, the tracking results from our algorithm were compared to those from the KLT tracker. In the second, we assessed the performance of our tracker when used as input to the facial expression recognition system described in Section 3.2, and compared it to other trackers used for this purpose.

4.1. Experimental Data

Several video sequences were obtained for training and testing. Type 1 video sequences contained facial feature movements such as raising of eyebrows, squinting of eyes, etc. along with natural head motions. These sequences were obtained from some subjects in our lab and some were from the CMU facial expression dataset [7]. Type 2 video sequences contained seven facial expressions with natural and significant amount of head motion, which are used as grammatical markers in ASL. These expressions were performed by deaf subjects from the Deaf and Hard-of-Hearing Federation of Singapore (DHHFS) while signing corresponding short sentences in ASL. For the current experiments on recognition, however, we considered only four of the facial expressions, viz., WH, YN, TP, and NEG, though all seven expressions were used to train the tracker. About one third of these sequences contained temporary occlusion of facial features by hand during signing, and also blurring due to fast head motion. We split the available data into four disjoint subsets for training and testing as follows:

- Set A: 87 Type 2 video sequences with a total of 2175 frames for learning the facial expression subspaces, and other statistics required for the tracker.
• Set B: 186 Type 1 video sequences with a total of 2869 frames for training and validating the structure of the HMMs which were used to provide the likelihoods of seven types of facial feature movement and head motion to the NN.

• Set C: 46 Type 2 video sequences with a total of 895 frames. In all there were 11, 15, 8, and 12 video sequences of the YN, WH, TP, and NEG expression, respectively. There was no blurring or occlusion by hand in these sequences.

• Set D: Type 2 video sequences with 705 frames. The difference from videos in Set C was that this set contained blurring and occlusion. Number of video sequences for each expression was: 17, 9, 2, and 10 for YN, WH, TP, and NEG, respectively.

The ground truth for the tracked facial feature points was obtained by manually marking the feature points in the first frame of each video, and then tracking these points through the video by the KLT algorithm. The errors in the tracked points were manually corrected.

Our tracker was trained on set A and tested on set C and set D. Set B was used for determining the optimum HMM structures, and for training them using the leave-one-out method. The output likelihoods of these trained HMMs corresponding to videos from set C were used to train and test the NN.

Because HMMs were constructed by using Type 1 data, the performance of the NN evaluated by using Type 2 data also represents the performance of the whole system. Set D was used as an unseen and challenging set to test the performance of the whole system including tracker, HMMs, and NN.

4.2. Tracking facial features

Fig. 3 and Fig. 4 shows the results for tracking facial features while signing a conditional sentence: "If it rains, I’ll go". In each figure, the blue rectangles show results from our tracker, and the red triangles show the KLT results.

In Fig. 3, KLT reveals its weakness: The tracked feature point on the lower middle lip started drifting away from the true location at frame 21 because of fast head motion and lip movement which caused blurring. In frame 49, after suffering from high face shape deformations and large head motions, many tracked points have moved away from the corresponding facial features and formed inconsistent shapes around the eyes, brows, and mouth. Results obtained from our tracker were stable even though tracked points were not always exactly at corresponding facial features. The results show that our tracking framework is flexible enough to deal with deformations caused by facial expressions and head motions. In Fig. 4, points tracked by KLT easily drifted away from occluded facial features but our system was robust to occlusion. We obtained similar results on new video sequences (obtained from ASL project in Boston University), whose subject was never seen by our system, as shown in Fig. 5. Besides, Fig. 6 shows the poor tracking result obtained for frame 32 for the same video sequence in Fig. 5 without using the subspace update scheme. This demonstrates the importance of adapting to face shapes of people not in the training set.

The tracking results were evaluated against manually labeled ground truth for each image frame. The normalized displacement error $\tilde{e}_{i,k}$ of the tracking result in the $i$th frame of the $k$th video sequence was computed as follows:

$$\tilde{e}_{i,k} = \frac{\sum_{j=1}^{n_f} e_{i,k,j}}{E_{M_0}^{0,k,n_f}},$$

where $n_f$ is the number of tracked feature points, $E_{M_0}^{0,k,n_f}$ is the distance between two inner eye corners in the first frame of $k$th video sequence, and serves as a normalizing factor, and $e_{i,k,j}$ is the distance between the $j$th feature point and the corresponding ground truth feature point. Fig. 7 shows a quantitative comparison between our approach and KLT: 97% of the feature points tracked by our system had normalized displacement error below 0.18, while only 70% of those tracked by KLT did.

4.3. Recognizing facial expressions

The input to the recognition system consisting of HMMs and the NN are the tracked features in each video frame. To assess the influence of the tracker on the recognition performance we trained and tested the system with various combinations of tracking inputs obtained from the Gold, Bayes,
Figure 3. This expression includes many facial feature movements due to facial expressions and head motions (blue rectangular marks: our method, red triangular marks: KLT tracker).

Figure 4. The face of this signer is occluded temporarily during signing.

Table 2. Recognition results of facial expressions with different trackers on set C and set D. Set C2 and D2 are respectively set C and set D without Topic expression data. Bayes is the proposed tracking framework. KLT tracker is the tracker implemented using KLT algorithm.

<table>
<thead>
<tr>
<th>Tracked inputs for training</th>
<th>Tracked inputs for testing</th>
<th>Set C</th>
<th>Set C2</th>
<th>Set D</th>
<th>Set D2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gold</td>
<td>Gold</td>
<td>74%</td>
<td>84%</td>
<td>69%</td>
<td>75%</td>
</tr>
<tr>
<td>Gold</td>
<td>Bayes</td>
<td>52%</td>
<td>66%</td>
<td>53%</td>
<td>58%</td>
</tr>
<tr>
<td>Gold</td>
<td>KLT</td>
<td>50%</td>
<td>63%</td>
<td>40%</td>
<td>45%</td>
</tr>
<tr>
<td>Bayes</td>
<td>Bayes</td>
<td>63%</td>
<td>82%</td>
<td>62%</td>
<td>66%</td>
</tr>
<tr>
<td>KLT</td>
<td>KLT</td>
<td>59%</td>
<td>76%</td>
<td>54%</td>
<td>50%</td>
</tr>
</tbody>
</table>

Table 2 lists the recognition results. The results for set C are average recognition rates obtained using the leave-one-out method on the 46 video sequences in that set. For example, when the recognition system was trained using the Gold tracks and tested using KLT tracks, the recognition rate was 50%. The best recognition rate of 74% was obtained by training and testing with the Gold tracks, while training and testing with the Bayes tracks and KLT tracks yielded 63% and 59%, respectively. Not surprisingly, the results are worse when the system is trained and tested using different trackers. The best recognition accuracy of 74% for the four expressions is somewhat low. This may be due to the fact that the video sequences represent natural signing where the different signers’ expressions did not follow textbook linguistic definition. More detailed analysis of the results showed that there was a confusion between the Head Forward motion for YN and Head Downward motion for TP. As shown in Fig. 8a and 8b, subjects tended to combine both motions for the YN and TP. To investigate how these confusing head motions affected the recognition rate, we removed the Topic expression from set C to form set C2 and repeated the experiments. The results with set C2 in Table 2 show a marked improvement in recognition, with training and testing by the Bayes tracker providing 82% accuracy. Hence, it appears that a more discriminative method must be devised for recognizing these natural head motions. We then used the system trained with set C (or set C2) and tested it on the unseen set D consisting of challenging video that includes motion blur and facial occlusion by hand. The corresponding results for set D and set D2 (set D without Topic) are also shown in Table 2. For set D, there is a small drop in Gold case from 74% to 69%, and in the Bayes case from 63% to 62%. For set D2 without the Topic expression, the Gold and Bayes results are 75% and 66%, respectively.

5. Conclusion

Here, we have proposed a suitable tracking scheme for facial feature points in sign language video, and assessed its performance with a temporal multichannel framework for recognizing four common facial expressions in ASL. We have used the KLT tracker as the basic tracker. How-
ever, this tracker is unable to cope with large head motions and occlusions that occur during natural signing. Hence, we have proposed a Bayesian feedback mechanism which incorporates PPCA and an online update mechanism to improve results and adapt to face shapes of different people. The experimental results show that our system can stably track facial features under natural head motion, temporary facial occlusions, and significant expression changes. The proposed recognition framework utilized temporal visual cues captured by the tracker using seven HMMs, and a NN which combined the HMM likelihoods of facial feature movements and head motions for identifying the facial expression. The experimental results show that the proposed Bayesian tracker worked well and stably with the recognition system for both training and testing on both normal and challenging data. An improvement of the recognition scheme will be considered for obtaining higher precision with our proposed tracking framework.

References


