A Framework for the Recognition of Non-Manual Markers in Segmented Sequences of American Sign Language

Nicholas Michael[1]
http://www.cs.rutgers.edu/~nicholam
Peng Yang[1]
http://www.cs.rutgers.edu/~peyang
Qingshan Liu[1]
http://www.cs.rutgers.edu/~qsliu
Dimitris Metaxas[1]
http://www.cs.rutgers.edu/~dnm
Carol Neidle[2]
http://www.bu.edu/asllrp/carol.html

[1] CBIM Center
Rutgers University,
Piscataway, NJ 08854, USA
http://cbim.rutgers.edu

[2] Linguistics Program
Boston University,
Boston, MA 02215, USA

Abstract

Despite the fact that there is critical grammatical information expressed through facial expressions and head gestures, most research in the field of sign language recognition has primarily focused on the manual component of signing. We propose a novel framework for robust tracking and analysis of non-manual behaviours, with an application to sign language recognition. The novelty of our method is threefold. First, we propose a *dynamic feature representation*. Instead of using only the features available in the current frame (e.g., head pose), we additionally aggregate and encode the feature values in neighbouring frames to better encode the dynamics of expressions and gestures (e.g., head shakes). Second, we use Multiple Instance Learning [10] to handle *feature misalignment* resulting from drifting of the face tracker and partial occlusions. Third, we utilize a *discriminative* Hidden Markov Support Vector Machine (HMSVM) [1] to learn finer temporal dependencies between the features of interest. We apply our signer-independent framework to *segmented recognition* of five classes of grammatical constructions conveyed through facial expressions and head gestures: wh-questions, negation, conditional/when clauses, yes/no questions and topics, and show improvement over previous methods.

1 Motivation

Speech recognition technologies have become standard components of modern operating systems, allowing average users to interact with computers verbally. Such technology has even found its way into the car industry, enabling drivers to make phone calls, play music, find directions to the nearest gas station, etc., without having their hands leave the wheel. It

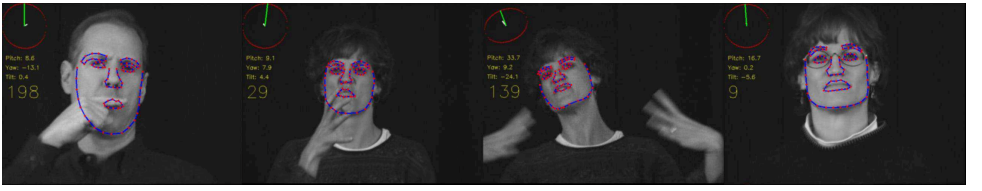

Figure 1: Sample tracked frames under challenging scenarios (partial occlusions, fast movements, and glasses), using the adopted face tracker [10]. Here, red dots represent tracked landmarks. The predicted 3D head pose is shown in the top left corner of each frame

will not be long before every modern device includes an integrated speech recognition module. Unfortunately, technology for the recognition of sign language, which is widely used by the Deaf, is not nearly as well-developed, despite the many potential benefits of such technology. First of all, technology that automatically translates between signed and written or spoken language would facilitate communication between signers and non-signers, thus bridging the language gap. Secondly, such technology could be used to translate sign language into computer commands, favouring the development of additional assistive technologies. Moreover, it could facilitate the efficient archiving and retrieval of video-based sign language communication and could assist with the tedious and time-consuming task of annotating sign language video data for purposes of linguistic and computer science research.

However, sign language recognition poses many challenges. First, many of the linguistic components of a sign that must be recognized occur *simultaneously* rather than sequentially. For example, one or both hands may be involved in the signing, and these may assume various hand shapes, orientations, and types of movement in different locations. At the same time, facial expression may also be involved in distinguishing signs, further complicating the recognition task (see Figure 1). Secondly, there is variation in production of a given sign, even by a single signer. Additional variation is introduced by effects of *co-articulation*, meaning that the articulation of a sign is influenced by preceding and following signs, and by movement transitions between signs (sometimes referred to as "movement epenthesis"). In spite of these challenges, many methods [2, 3, 5, 27, 28, 31] have shown promising results in recognizing manual components of signs.

Furthermore, in sign language, critical grammatical information is expressed through head gestures (e.g., periodic nods and shakes) and facial expressions (e.g., raised or lowered eyebrows, eye aperture, nose wrinkles, tensing of the cheeks, and mouth expressions [11, 16]). These linguistically significant non-manual expressions include grammatical markings that extend over phrases to mark syntactic scope. For example, in *wh-questions* (which involve phrases such as *who, what, when, where, why*, and *how*), the grammatical marking consists of lowered eyebrows and squinted eyes that occur either over the entire wh-question or solely over a wh-phrase that has moved to a sentence-final position. In addition, there may be a slight, rapid side-to-side head shake over at least part of the domain of the wh-question marking. With *negation*, there is a relatively slow side-to-side head shake that co-occurs with a manual sign of negation (such as NOT, NEVER), if there is one, and may extend over the scope of the negation, e.g., over the following verb phrase that is negated. The eyes may squint or close. The non-manual sign for *yes/no* questions extends over the entire sentence and involves raising the eyebrows, widening the eyes and jutting the head forward. *Conditional/when* sentences are two part constructions with the relevant non manual marking

only over the first part (i.e., over the "conditional" or "when" clause). This is characterized by raised eyebrows, wide eyes, head forward (or back) and tilted to the side, followed by a pause, after which the eyebrows and head return to neutral position. Lastly, *topics* are characterized by raised eyebrows, wide eyes, head tilted back, and an optional nod.

Sign language recognition cannot be successful unless these non-manual signals are also correctly detected. For example, depending on the accompanying non-manual markings, the sequence of signs JOHN BUY HOUSE could be interpreted to mean any of the following: (i) "John bought the house." (ii) "John did not buy the house." (iii) "Did John buy the house?" (iv) "Did John not buy the house?" (v) "If John buys the house...".

Motivated by the grammatical importance of head gestures and facial expressions, we present a novel framework for robustly tracking and recognizing such non-manual markings associated with *wh-questions*, *negative* sentences, *conditional/when* clauses, *yes/no* questions and *topics*. Our method extends prior work in [14], where a face tracker first locates facial landmarks and then appearance and head pose features are fed to a Support Vector Machine (SVM) for classification, while making a number of significant contributions. These allow us to recognize a wider class of non-manual markers in *segmented sequences*[1] of American Sign Language (ASL). First, once we track the facial landmarks, we focus on an extended rectangular region of interest (ROI), which includes the eyes, eyebrows and nose, so as to capture a wider range of upper face expressions, e.g., nose wrinkling and cheek tensing. Second, we divide this ROI into a set of smaller patches (henceforth referred to as parts), which correspond roughly to areas of the face relevant for these specific grammatical expressions, e.g., inner and outer eyebrows. We extract from each part a histogram of Local Binary Patterns (LBP) [18]. These are effective for texture classification [22, 32], faster to compute and more robust to illumination variations than SIFT, which is used in [14]. Third, we handle feature misalignment, arising from tracking inaccuracies and partial facial occlusions, by computing a Multiple Instance Feature (MIF) [12] for each part. Fourth, in addition to the head pose and texture features per frame, we explicitly calculate eyebrow height. The final feature descriptor is augmented with a "summary" of the features of future and past frames sampled at regular intervals in the neighbourhood of the current frame, which we call *Oracle Features* (see Figure 2). This representation aims to encode the dynamic nature of facial expressions and head gestures encountered in non-manual grammatical markers. Lastly, by utilizing a discriminative, margin-maximizing, Hidden Markov Support Vector Machine (HMSVM) [1, 8] our method outperforms generative Hidden Markov Models (HMMs) [21], which can over-fit the training data in the absence of sufficient training examples.

2 Previous Work

As already mentioned, most research on computer-based sign language recognition has focused on the manual components of signing [20]. More specifically, [23] uses color tracking and HMMs, while the authors of [28] split the manual signs into independent movement and hand shape channels to handle simultaneous manual events. Bauer and Kraiss [2] break down signs into sub-units using unsupervised clustering. In [29], the authors develop a method that quickly adapts to unknown signers, in an attempt to handle interpersonal variance. Similarly, the authors of [33] use a background model to achieve accurate feature extraction and then perform feature normalization to achieve person independence. Martinez and Ding [7] first

[1] For each video sequence we only attempt to classify segments of frames containing non-manual markers of one of the five classes mentioned.

perform 3D hand reconstruction and then represent hand motions as 3D trajectories. Lately, we have even seen the emergence of some weakly supervised methods that attempt to learn manual signs from TV subtitles [4, 5].

Only recently have researchers begun to address the importance of facial expressions for sign recognition systems [19]. An extensive review of recent developments in visual sign recognition, together with a system that captures both manual and non-manual signs, is provided by [30]. However, it requires the signer to be wearing a glove with coloured markers to enable robust hand tracking and hand posture reconstruction. Most importantly, the tracked facial features are not used to recognize facial expressions that have grammatical meaning. In [25, 26] a 3D deformable model for face tracking is presented, which emphasizes outlier rejection and occlusion handling at the expense of slower run time. The system is used to demonstrate the potential of face tracking for the analysis of facial expressions found in sign language, but is not used for any actual recognition. Lastly, in [14] we used spatial pyramid features and an SVM to recognize segmented *wh-questions* and *negative* sentences only. In the previous approach, we did not exploit temporal dependencies between features. The method proposed here is able to distinguish *wh-questions* and *negative* sentences, as well as *topics*, *conditional/when* clauses and *yes/no* questions, by encoding feature dynamics and modelling temporal dependencies.

3 Feature Extraction and Representation

Essential to our recognition framework is the ability to accurately track facial landmarks of the signers and estimate their 3D head pose (see Figure 1). For this we use the tracking algorithm of Kanaujia *et al.* [10] that is based on Active Shape Models (ASM) [6]. An ASM is a statistical model of facial shape variation, obtained through the application of Principal Component Analysis (PCA) and Procrustes Analysis on an aligned training set of facial shapes. The adopted tracker [10] can track in real time the positions of 79 facial landmarks (e.g., nose, eyes, etc.), utilizing a mixture of experts to map landmark configurations to predictions of head pose (we skip the details, since the actual tracker is not our focus).

3.1 Tracking eyebrow height and head pose

From the tracked landmarks we can compute the 2D position of the signers' left, (x_L, y_L), and right inner eyebrows[2], (x_R, y_R), and their nose tip[3], (x_N, y_N), in each frame. The eyebrow height at time t, denoted as h^t, is derived as the average Euclidean distance between the nose tip and each inner eyebrow:

$$h^t = \frac{1}{2} \times \left(\sqrt{(x_L - x_N)^2 + (y_L - y_N)^2} + \sqrt{(x_R - x_N)^2 + (y_R - y_N)^2} \right) \ . \tag{1}$$

For robustness to tracking noise, we filter the computed (x, y) positions of the key points (eyebrows and nose tip) using a Kalman filter [9], assuming linear state dynamics with Gaussian noise, $\mathbf{w}$. The system state, $\mathbf{x}_t$, includes the position, $\mathbf{p}_t = [x_L, y_L, x_R, y_R, x_N, y_N]^T$, and the velocity, $\mathbf{v}_t = [\dot{x}_L, \dot{y}_L, \dot{x}_R, \dot{y}_R, \dot{x}_N, \dot{y}_N]^T$, of these key points at time t. The dynamic process is governed by:

$$\mathbf{x}_{t+1} = A_k \mathbf{x}_t + \mathbf{w}_t \ , \tag{2}$$

[2] We use the 4 innermost eyebrow landmarks

[3] We use the lower 8 nose landmarks to compute the nose tip position

with

$$A_{t+1} = \begin{pmatrix} 1 & \delta t \\ 0 & 1 \end{pmatrix} \text{ and } \mathbf{w}_t \sim \mathcal{N}(0, Q) \ . \tag{3}$$

The observation process is modelled as:

$$\mathbf{z}_t = H\mathbf{x}_t + \mathbf{u}_t \ , \tag{4}$$

where $H = [1,0]$, $\mathbf{z}_t$ is the observation as obtained by the face tracker and $\mathbf{u}_t \sim \mathcal{N}(0,R)$ is the observation noise at time t. A similar model is also used to filter the predicted head pose. In this case the state vector includes the 3D head pose, $\mathbf{a}_t = [a_P, a_Y, a_T]^T$, and the head pose velocity $\dot{\mathbf{a}}_t = [\dot{a}_P, \dot{a}_Y, \dot{a}_T]^T$, where P, Y and T stand for pitch, yaw and tilt angles respectively.

3.2 Texture Features

Once we track the signer's head, we compute a bounding box of the tracked landmarks around the eyes, eyebrows and nose, forming an extended ROI from which we compute Local Binary Patterns (LBP) [18]. Put simply, LBPs are binary codes that characterize the texture in the neighbourhood of a pixel by thresholding the value of each neighbour by the gray-scale value of the central pixel (set to 1 if larger, set to 0 otherwise) and interpreting the pattern as a binary number, which is converted to a decimal code. Typically, LBP codes are first computed for each pixel in an image patch and then the normalized histogram of LBP codes is generated and used as a texture descriptor of the patch.

3.3 Multiple Instance Feature

Feature misalignment sometimes occurs; i.e., the same features do not always fire up in all positive detection windows, often because of object pose variation. Lin *et al.* [12] introduced Multiple Instance Features (MIF) for boosted learning of part-based human detectors, where an initial boosting seeds the location of an object part from translated candidates, and then multiple instance boosting pursues an aggregated feature for each part. So an MIF is an aggregation function of instances. More specifically, given a classifier, C, it is the aggregated output, y, of a function, f, of classification scores, $\{y_j\}_{j=1}^J$, on multiple instances, $\{x_j\}_{j=1}^J$:

$$y = f(y_1, y_2, ..., y_J) = f(C(x_1), C(x_2), ..., C(x_J)). \tag{5}$$

Each bag, x_i, consists of a set of instances, $\{x_{ij}\}_{j=1}^N$. For each classifier C, the score y_{ij} of an instance x_{ij} can be computed as: $y_{ij} = C(x_{ij})$. The probability of an instance x_{ij} to be positive is given by the logistic function: $p_{ij} = \frac{1}{1+e^{-y_{ij}}}$. In [13], the multiple instance learning problem is formulated as the maximization of diverse density, which measures the intersection of the positive bags minus the union of the negative bags.

The diverse density is probabilistically modelled using a Noisy-OR model to harness the multiple instance learning problem. The probability that a bag x_i is positive is formulated as $p_i = 1 - \prod_{j=1}^{N_i}(1 - p_{ij})$. The Noisy-OR model means the probability of the bag to be positive is high when this bag includes at least one instance with high probability to be positive, otherwise the bag is negative when all the instances inside have low probability of being positive. Following [12], the geometric mean is applied to avoid the numerical issues when N_i is large, so the formula is modified to $p_i = 1 - \prod_{j=1}^{N_i}(1 - p_{ij})^{1/N_i}$. The multiple

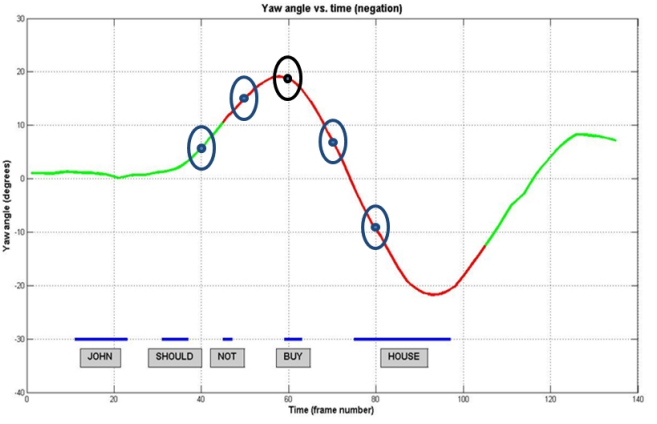

Figure 2: Plot of head yaw angle over time for a sequence containing negation (red segment marks when the non-manual marker occurs), also illustrating computation of yaw oracle features for frame 60 (see Section 3.4)

instance aggregated score y_i is computed from the instance scores y_i as:

$$y_i = \log((\prod_{j=1}^{N_i}(1+e^{y_{ij}})^{1/N_i}) - 1), \tag{6}$$

which comes from the logistic relation between p_i and y_i: $p_i = \frac{1}{1+e^{-y_i}}$. In this paper each y_i is an MIF of texture, obtained by learning weak classifiers (decision tree stumps) on the LBP histogram bins of a part (a part is a patch within the face ROI). See [12] for further details.

3.4 Oracle Features

Facial expressions and gestures are dynamic processes, especially those that have a grammatical meaning in ASL. It is often difficult even for ASL signers to detect non-manual markers using static frames alone. For example, one key component of the non-manual marking of negation is a head shake, whose presence in a sequence cannot be detected solely by looking at the head pose in any single frame. Instead, one needs to have available a "snapshot" of the variation of head yaw angle over time, in order to detect the turning of the head in one way and then in the opposite way.

Therefore, in order to strengthen the representational power of all features (texture MIF, head pose, eyebrow height), we encode information from neighbouring frames. For each frame we sample the feature values at regular offsets (sample points) from the current frame (anchor point). Before sampling, we compute a weighted average (by means of a Gaussian curve) of the feature value in a small window around the anchor and each sample point. This is illustrated in Figure 2 where an example anchor point is shown in black and example sample points are shown in blue. The ellipses indicate the size of the averaging neighbourhoods. Thus, the final feature descriptor of each frame is formed by combining features in that frame with the features obtained from the neighbourhood of the respective sample points. We refer

to these augmented feature vectors as *Oracle features* because for every frame they encode the dynamic evolution of feature values. We show that this richer feature representation allows our method to achieve higher classification accuracy (see Section 5).

4 Hidden Markov Support Vector Machine

In the traditional supervised classification setting, we have a set of labelled training data $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, where $\mathbf{x}_i \in \Re^d$ is the d-dimensional feature vector of training sample i and $y_i \in \Re$ is its corresponding class label. The goal is to learn a mapping function from inputs to outputs $F: \Re^d \rightarrow \Re$ that minimizes some loss function, typically a 0/1 loss. In the sequence tagging problem we have sequences of feature vectors and for each one we have a sequence of corresponding outputs: $D = \{(\mathbf{x}_i^j, y_i^j) | j = 1, \ldots, J\}_{i=1}^{N}$, where J is the length of the i^{th} sequence. Note that sequences need not have the same length. The goal in this setting is to predict the class labels of all instance within each sequence.

A popular model used in sequence tagging problems (most notably for speech recognition) is the Hidden Markov Model (HMM) [21]. Despite its success, the HMM has certain limitations. First of all, it assumes conditional independence between observations when given the current state; an assumption that can be too restrictive for certain problems where there are complex feature interactions. Secondly, HMMs are generative models. During their non-discriminative training, the goal is to learn model parameters that maximize the likelihood of fitting the given training data, instead of optimizing for accurate classification (although recently there has been interest in alternative methods for training [31]).

Altun *et al*. [1] proposed the Hidden Markov Support Vector Machine (HMSVM), which, like the HMM, models the interactions between features and class labels, as well as interaction between neighbouring labels within a sequence. Unlike HMMs, the HMSVM model is trained in a discriminative margin-maximizing learning procedure. This means that it can achieve better generalization performance on test data, hence higher accuracy. Similar to the standard Support Vector Machine (SVM) [4], the HMSVM can also learn non-linear discriminant functions via the kernel trick.

Given a feature sequence $\mathbf{x} = \{\mathbf{x}^j\}_{j=1}^{J}$, where $\mathbf{x}^j$ are instances within the sequence the model predicts the corresponding tag sequence $\mathbf{y} = \{y^j\}_{j=1}^{J}$ using [1]:

$$\mathbf{y} = \arg\max_{\mathbf{y} \in \mathscr{Y}} \Big(\sum_{j=1}^{J} \big(\sum_{k=1}^{K} < \mathbf{x}_j, \mathbf{w}_{y_{j-k}, \ldots, y_K} > + < \phi_{\text{trans}}(y_{j-k}, \ldots, y_K), \mathbf{w}_{\text{trans}} > \big) \Big) \quad , \tag{7}$$

where $\mathbf{w}_{y_{j-k}, \ldots, y_K}$ is an emission weight vector modelling interactions between features and k^{th}order observations, and $\mathbf{w}_{\text{trans}}$ is the transition weight vector modelling transitions between neighbouring tags. Discriminative training aims to minimize the number of misclassified tags, while maximizing the separation margin, hence the training objective is [1]:

$$\min \{ \tfrac{1}{2} \mathbf{w}^T \mathbf{w} + \tfrac{c}{J} \textstyle\sum_{j=1}^{J} \xi_j \} \tag{8}$$

$$\text{s.t.:} \quad z_j(\mathbf{y}) (< \mathbf{w}, \Phi(\mathbf{x}_j, \mathbf{y}) > + \theta_j) \geq 1 - \xi_j \xi_j \geq 0 \quad , \forall j = 1, \ldots, J \quad , \forall \mathbf{y} \in \mathscr{Y} \quad , \tag{9}$$

where c is a parameter that controls the penalty of misclassification trading off training error and margin size. Joachims *et al*. proposed the cutting plane algorithm [8] which offers a significant speed-up in the training time of HMSVMs over the original working set algorithm of [1]. In our framework, from each frame in each segmented sequence, we use the oracle

feature representation of the eyebrow height, the head pose and the multiple instance texture features, with their corresponding class label, and train a one-vs-all HMSVM model[4]. Sequences in our training set contained no overlapping non-manual markers, so we only needed one model to tag each frame in the segmented sequence. Despite this, our method can easily generalize to sequences with overlapping non-manual markers by training n one-vs-all models (one for each class) and running them in parallel on each sequence.

5 Experimental Results

All experiments are based on a linguistically annotated corpus[5] of ASL (as produced by native signers). This publicly available corpus, including 15 short narratives plus hundreds of additional elicited utterances, includes multiple synchronized views of the signing (generally 2 stereoscopic front views plus a side view and a close-up of the face), which have been linguistically annotated using SignStream™ [15, 17] software, which enables identification of the start and end points of the manual and non-manual components of the signing.

From this corpus we selected training and testing sets of 32 and 13 video clips, respectively, of isolated utterances, extracting the segments containing non-manual markers of the classes of interest. Certain sequences contained multiple non-manual markers but there was no overlap between them. The exact composition of these sets per class is shown in Table 1. Both sets contained three different native signers.

Using the methods described in previous sections, we tracked the signers' head, extracting their head pose and computing their eyebrow height. These were post-processed with a Kalman filter for more accurate tracking. From the filtered head pose, we compute the head pose derivative per frame, to avoid learning a dependence on the initial head position of a signer. Eyebrow height is also normalized by the height in the first frame of each sequence and then we compute the height derivative, in order to normalize for subjects of different face proportions and distance from the camera. For each frame we compute oracle features as explained in Section 3.4. We use 5 sample points, offset at 0, +5, +10, +15 and +20 pixels from the current frame respectively, averaged over a 5 frame window, resulting in a 20-dimensional descriptor of head pose (pitch, yaw, tilt) and height variation per frame.

Before extracting texture features from the face ROI, we align all images, rotating frames by the average of the tilt angle and the angle between the centroids of the two eyes, as computed from the ASM landmarks. Faces were normalized by cropping frames to 64x64 pixels [24]. The face ROI is divided into a 4x4 cell grid with each cell being 16x16 pixels. From each cell we compute normalized histograms of uniform LBP features [18] using 8 samples and a radius of 1 pixel. For purposes of computing MIF [12], we consider each cell being one facial part (so we have 16 parts per frame) and translate each cell in a regular grid around its original position, computing additional LBP features. The collection of features for a given part form a bag of instances which we convert to a 5-dimensional MIF score, one for each class of non-manual markers. The idea is that if a positive part, with respect to a class label, is misaligned (as a result of tracking error or partial occlusion), as we translate it around its neighbourhood and compute instances of LBP features, at least one of these instances will capture features from a correct part placement, and the bag will still be positive for that class. As in the case of head pose and eyebrow height, we compute oracle features for the LBP MIF. Here, to avoid increasing feature dimensionality too much, we only use

[4] www.cs.cornell.edu/People/tj/svm_light/svm_hmm.html

[5] Collected at Boston University, searchable via: http://secrets.rutgers.edu/dai/queryPages

Class	Training Set	Testing Set
C/W	4 (292)	2 (158)
Neg	8 (532)	4 (258)
Top	9 (249)	4 (86)
Wh	8 (492)	5 (351)
Y/N	4 (262)	2 (172)

Table 1: Number of segmented sequences per class in datasets (total number of frames in parenthesis)

True Class	Predicted Class				
	C/W	Neg	Top	Wh-Q	Y/N
C/W	100%	0	0	0	0
Neg	0	75%	0	25%	0
Top	0	0	100%	0	0
Wh-Q	0	0	0	80%	20%
Y/N	0	0	0	0	100%

Table 2: Confusion matrix of HMSVM segmented recognition using oracle features of LBP-MIF, head pose and eyebrow height

	% Correct classification
HMM	70.6%
HMSVM	**88.2%**
HMSVM + non-MIF LBP	82.4%
HMSVM + MIF LBP + non-oracle	76.5%

Table 3: Evaluation of models showing the benefit of discriminative HMSVM with the proposed feature representation that handles feature dynamics and feature misalignment

3 sample points, offset at 0, +5 and +10 pixels from the current frame respectively, also averaged over a 5 frame window, resulting in a 240-dimensional texture descriptor.

The three sets of features (pose, height and texture) are concatenated into one feature vector and we train an HMSVM. Because of our small training set, we first optimize the parameter c using 3-fold cross validation on the training set, ensuring that each fold contains at least one sequence from each class, before evaluating on the test set. The recognition accuracy of the HMSVM model is summarized in Table 2. Analysis of the results revealed that for the wh-question sequence that is misclassified as a yes/no question the signer's head is rotated to the side, causing an incorrect estimation of the eyebrow height. Most importantly, this rotation causes a significant change in the appearance of the face ROI since most of the training images are frontal views. We expect to be able to overcome this problem by using training data that includes such cases of non-frontal faces. Additionally, our method mistakes a negative sequence for a wh-question. In this sequence there is a clear head shake that our framework can capture and which is characteristic of negation. However, there is a headshake – albeit somewhat different in character – that frequently occurs with wh-questions, as well as some degree of furrowing of the brows that occurs with both constructions. The model failed on this case, possibly because of insufficient training examples exhibiting this combination of eyebrow appearance and head shaking.

In order to compare the HMSVM with the HMM, we also trained 5 HMMs, one for each class, classifying test sequences as belonging to the class whose HMM yields the highest probability. The number of states of each HMM was decided using 3-fold cross validation. Results are shown in Table 3. Note that with our small dataset, the generative HMM fails to outperform the discriminative HMSVM model. In the same table we also show the result of an experiment where we used oracle pose and height features with non-MIF oracle LBP features and an HMSVM recognizer (HMSVM + non-MIF LBP). Note that this model performs worse, showing that the MIF indeed help improve accuracy. Using non-oracle features (HMSVM + MIF LBP + non-oracle) also hurts performance, as expected, given the dynamic

dependence of features relevant to recognition of facial expressions and non-manual ASL grammatical markers in particular.

6 Discussion

We presented a novel framework for robust *real time* face tracking and facial expression analysis from a single uncalibrated camera. We demonstrated that our framework is successful at isolated recognition of wh-questions, conditional/when clauses, yes/no questions, negation and topics in segmented video data. As demonstrated by our experimental results, the key to the success of our method lies both in the discriminative recognition model (HMSVM) as well as in the rich feature representation that encodes feature dynamics and is able to handle feature misalignment.

As future research, we first aim to improve the face tracking algorithm for more accurate height estimation and better occlusion handling, as well as exploring more complex learning models. Furthermore, we wish to extend this framework to continuous recognition of non-manual markers and to go beyond simply distinguishing among a small set of candidate non-manual markings to recognition of a larger set of expressions that often differ from each other in subtle ways. Such a system will have more practical applications as it will not require the preprocessing step of sequence segmentation to extract the segment containing some non-manual marker that has previously been labelled by human annotators.

References

[1] Y. Altun, I. Tsochantaridis, and T. Hofmann. Hidden markov support vector machines. In *International Conference on Machine Learning*, pages 3–10, 2003.

[2] B. Bauer and K.-F. Kraiss. Video-based sign recognition using self-organizing subunits. In *ICPR*, volume 2, pages 434–437, 2002.

[3] P. Buehler, A. Zisserman, and M. Everingham. Learning sign language by watching TV (using weakly aligned subtitles). In *IEEE CVPR*, pages 2961–2968, 2009.

[4] C. J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

[5] H. Cooper and R. Bowden. Learning signs from subtitles: A weakly supervised approach to sign language recognition. In *IEEE CVPR*, pages 2568–2574, June 2009.

[6] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models – their training and application. In *Comp. Vis. Image Underst.*, pages 38–59, 1995.

[7] L. Ding and A. M. Martinez. Three-dimensional shape and motion reconstruction for the analysis of American Sign Language. In *CVPRW '06: Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop*, page 146, Washington, DC, USA, 2006. IEEE Computer Society.

[8] T. Joachims, T. Finley, and C.-N. Yu. Cutting-plane training of structural svms. *Machine Learning*, 77:27–59, 2009.

[9] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[10] A. Kanaujia, Y. Huang, and D. Metaxas. Tracking facial features using mixture of point distribution models. In *ICVGIP*, 2006.

[11] S. K. Liddell. *American Sign Language Syntax*. Mouton, The Hague, 1980.

[12] Z. Lin, G. Hua, and L.S. Davis. Multiple instance feature for robust part-based object detection. *IEEE Conf. on Computer Vision and Pattern Recognition*, 0:405–412, 2009.

[13] O. Maron and T. Lozano-Pérez. A framework for multiple-instance learning. In *Advances in Neural Information Processing Systems*, volume 10, pages 570–576. MIT Press, 1998.

[14] N. Michael, D. N. Metaxas, and C. Neidle. Spatial and temporal pyramids for grammatical expression recognition of American Sign Language. In *ASSETS*, pages 75–82, October 2009.

[15] C. Neidle. Signstream™: A database tool for research on visual-gestural language. *Journal of Sign Language and Lignuistics*, 4(1/2):203–214, 2002.

[16] C. Neidle, J. Kegl, D. MacLaughlin, B. Bahan, and R. G. Lee. *The Syntax of American Sign Language: Functional Categories and Hierarchical Structure*. MIT Press, Cambridge MA, 2000.

[17] C. Neidle, S. Sclaroff, and V. Athitsos. Signstream™: A tool for linguistic and computer vision research on visual-gestural language data. *Behavior Research Methods, Instruments, and Computers*, 33(3):311–320, 2001.

[18] T. Ojala and M. Pietikäinen. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *PAMI*, 2002.

[19] S. C. W. Ong and S. Ranganath. Automatic sign language analysis: a survey and the future beyond lexical meaning. *IEEE TPAMI*, 27(6):873–891, June 2005.

[20] V. I. Pavlovic, R. Sharma, and T. S. Huang. Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE TPAMI*, 19:677–695, 1997.

[21] L. R. Rabiner. A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[22] Caifeng Shan, Shaogang Gong, and Peter W.McOwan. Robust facial expression recognition using local binary patterns. *IEEE Int. Conf. on Image Processing*, 2005.

[23] T. Starner and A. Pentland. Visual recognition of American Sign Language using Hidden Markov Models. In *International Workshop on Automatic Face and Gesture Recognition*, pages 189–194, 1995.

[24] Y. Tian. Evaluation of face resolution for expression analysis. *Computer Vision and Pattern Recognition Workshop on Face Processing in Video*, 2004.

[25] C. Vogler and S. Goldenstein. Facial movement analysis in ASL. *Univers. Access Inf. Soc.*, 6(4):363–374, 2008.

[26] C. Vogler and S. Goldenstein. Toward computational understanding of sign language. *Technology and Disability*, 20(2):109–119, 2008.

[27] C. Vogler and D. Metaxas. ASL recognition based on a coupling between HMMs and 3D motion analysis. In *ICCV*, pages 363–369, 1998.

[28] C. Vogler and D. Metaxas. *Handshapes and movements: Multiple-channel ASL recognition*, pages 247–258. LNAI. Springer, Berlin, 2004.

[29] U. von Agris, D. Schneider, J. Zieren, and K.-F. Kraiss. Rapid signer adaptation for isolated sign language recognition. In *V4HCI*, 2006.

[30] U. von Agris, J. Zieren, U. Canzler, B. Bauer, and K.-F. Kraiss. Recent developments in visual sign language recognition. *Univers. Access Inf. Soc.*, 6(4):323–362, 2008.

[31] Y. Wang, X. Ni, and F. Jian. Discriminative training methods of HMM for sign language recognition. *CAAI Transactions on Intelligent Systems*, 1:80–84, 2007.

[32] G. Zhao and M. Pietikäinen. Dynamic texture recognition using volume local binary patterns. *European Conference on Computer Vision*, 2006.

[33] J. Zieren and K.-F. Kraiss. Robust person-independent visual sign language recognition. In *Proceedings of the 2nd Iberian Conference on Pattern Recognition and Image Analysis*, volume LNCS, 2005.