Learning Partially-Observed Hidden Conditional Random Fields for Facial Expression Recognition

Kai-Yueh Chang[1,2] Tyng-Luh Liu[1] Shang-Hong Lai[2]
[1]Institute of Information Science, Academia Sinica, Taiwan
[2]Department of Computer Science, National Tsing Hua University, Taiwan

Abstract

This paper describes a novel graphical model approach to seamlessly coupling and simultaneously analyzing facial emotions and the action units. Our method is based on the hidden conditional random fields (HCRFs) where we link the output class label to the underlying emotion of a facial expression sequence, and connect the hidden variables to the image frame-wise action units. As HCRFs are formulated with only the clique constraints, their labeling for hidden variables often lacks a coherent and meaningful configuration. We resolve this matter by introducing a partially-observed HCRF model, and establish an efficient scheme via Bethe energy approximation to overcome the resulting difficulties in training. For real-time applications, we also propose an on-line implementation to perform incremental inference with satisfactory accuracy.

1. Introduction

For humans, the making of a particular facial expression is a *continuous* and often *short* event, typically triggered by the associated emotion and put together through a series of muscle motions (see Figure 1). These subtleties have caused designing computer vision algorithms to automatically detect and recognize facial expressions a challenging task. To tackle this problem, most of the existing techniques, *e.g.*, [2, 14] have converged to investigate the *action units* (AU) of the Facial Action Coding System (FACS) proposed by Ekman and Friesen [7]. In this work, our primary goal is to establish a new graphical model approach of which classifying facial expressions and identifying action units can be elegantly coupled and simultaneously analyzed. As a result, the framework can lead to a more effective implementation for both technical and practical concerns. (The term "facial expressions" hereafter is restricted to the six basic ones, including `joy`, `sadness`, `surprise`, `anger`, `fear`, and `disgust`.)

Instead of recognizing the facial expression per image frame, we consider casting the problem as a classification task over an image sequence. That is, our aim is to determine the class label specifying the emotion of a given sequence. Meanwhile, observe that knowing the combinations of action units (especially those in the *peak* images) generally provides useful evidence for distinguishing different emotions. It is therefore insightful to know what action units are activated in each facial image. To this end, we consider *hidden conditional random fields* (HCRF) [16] for facial expression recognition, and establish a useful connection between the hidden states of a CRF [11] and the action units. A key distinction between our framework and HCRF is that learning the proposed graphical model utilizes information from some *partially-observed* hidden states. We will show that such a deviation usually yields better predictions for the sequence and the hidden-state labels, with the price of requiring a more delicate learning/training process.

Figure 1. From left to right, these face images show a transition from *neutral* to *peak* for the making of a `joy` facial expression.

1.1. Related work

An extensive review on facial expression analysis can be found in Pantic and Rothkrantz [15]. While the survey is a bit outdated, it still provides a comprehensive overview on the related topics. In what follows, we briefly describe techniques that deal with facial activity, and then discuss those focusing on predicting the emotion of a facial expression.

On analyzing facial activity and action units, Donato *et al*. [6] show that the Gabor wavelet representation and the independent component analysis are useful for classifying action units. They conclude with a surmise that combining motion and gray-level information may give the best facial expression recognition performance. Kapoor *et al*. [10] have constructed a shape model of the upper face, and used the model parameters as the inputs to SVMs for recognizing action units within the upper face. Different from

978-1-4244-3991-1/09/$25.00 ©2009 IEEE

[10], Bartlett *et al*. [2] consider applying SVMs to the Gabor wavelet coefficients of a face image. In [17], Tian *et al*. describe a neural network approach that facial expressions are analyzed based on a set of *permanent* facial features (brows, eyes, mouth) and *transient* facial features (furrows). For better inferring the action units, Tong *et al*. [18] have utilized a dynamic Bayesian network (DBN) to model the spatial and temporal relationships among the action units.

To classify facial expressions into a set of basic emotion classes, Pantic and Rothkrantz [14] propose a rule-based system to code the emotions via action units. Zhang and Ji [24] instead establish a DBN model to correlate the relationships between the emotions and the action units. Alternatively, there are methods that are formulated to directly recognize the basic emotions without drawing on action units. Cohen *et al*. [4] consider the tree-augmented-naive Bayes (TAN) classifier to learn the dependencies between the facial emotions and the *motion units*. In [2], Bartlett *et al*. report that in their experiments the best results on classifying facial expressions into basic emotions are achieved by using SVMs with feature selection through AdaBoost [8].

Among the preceding techniques [2, 4, 14, 24] on linking facial expressions to emotions, the emotion class label is *image frame-wise* predicted (despite that their formulation may use temporal information). This may not be reasonable, as the making of a facial expression is a transition over a sequence of image frames. For example, it would be unrealistic and difficult to tell whether the facial expression associated with the third image in Figure 1 is `joy` even by relating to the second image. Studies, *e.g*., [1, 3] from a psychology viewpoint have also supported the *sequence-wise* analysis can achieve better recognition results.

Yang *et al*. [21, 22] and Zhao and Pietikäinen [25] have described techniques to extract features from a whole image sequence, and to sequence-wise classify the emotion. As the sequence lengths of a facial expression are generally different, the so-called *dynamic features* accounting for the distribution of temporal patterns are proposed in Yang *et al*. [21] to handle such variations. However, when processing facial expression sequences, not only is the time span variable, but the change of the *magnitude* is non-linear. Learning only the temporal distribution may not well address the full complexity of the difficulty. And the need to have access to a whole sequence further prevents these approaches from being generalized to supporting real-time applications.

2. Image Representation via Boosting

Our formulation classifies sequences of facial expression into six emotion categories: `joy` (2), `sadness` (3), `surprise` (4), `anger` (5), `fear` (6) and `disgust` (7). For the convenience of implementation, we also have an additional category called `neutral` (1). The numbers in the brackets are the emotion class labels.

Let D be the training data. $(\mathbf{s}, y) \in D$ denotes that $\mathbf{s} = \{I_1, \ldots, I_e\}$ is a sequence of facial-expression images and $y \in \{2, \ldots, 7\}$ is its emotion class label. (e symbolizes *ending*, and its value could vary from different $\mathbf{s}$.) As each sequence $\mathbf{s}$ starts from a neutral status and ends at a peak status of the underlying expression (see Figure 1), we can thus produce two labeled images from $\mathbf{s}$: I_1 will be labeled as `neutral` (1), and I_e will have label y inherited from the sequence.

With the (pairwise) labeled images from D, we are ready to construct the image representation. It is desirable to have a representation based on image features that are *stable* and have certain *invariant* properties. The nowadays popular *interest points*, *e.g*., [12] appear to meet the requirements. Nevertheless, the *bag-of-features* representation is somewhat awkward for incorporating into a classification framework. To relax the restriction, we next describe a scheme to transfer a bag-of-features description into a feature vector.

2.1. Interest point descriptor

We re-scale each labeled image to 100×100 pixels, and use the SIFT keypoint detector [12] to generate interest points. Instead of the SIFT descriptor, we find that encoding an interest point with Gabor responses leads to better classification results for our application. Specifically, we construct a bank of Gabor filters at 8 orientations and 9 spatial frequencies (4 to 64 pixels per cycle at $1/2$ octave steps), and perform image convolution through the filter bank. We also attach the coordinates of an interest point to the Gabor response. Altogether, each interest point is described by a descriptor vector of dimension 74 ($= 8 \times 9 + 2$).

2.2. Image feature vector

Let $\{J_i\}_{i=1}^M$ be the set of (pairwise) labeled images, N_i be the number of interest points detected from J_i, and $N = \sum_{i=1}^M N_i$ be the total number of interest points. Since not all interest points are discriminative, and in practice N is too large to work with, we consider AdaBoost for interest point selection, and build our image representation based on the selected, informative ones. We carry out the interest point selection in seven different runs in that the training images are labeled from seven emotion categories. As the procedure in each run is exactly the same, it suffices to explain how it is done for a particular emotion category.

We begin by dividing the training images into positive and negative ones, and define a "distance function" d to measure the *irrelevance* of interest point k to image J_i by

$$d(k, J_i) = \min_{\ell \in \{1, \ldots, N_i\}} \|\mathbf{g}^k - \mathbf{g}_i^\ell\|, \qquad (1)$$

where $\mathbf{g}^k$ and $\mathbf{g}_i^\ell$ are the interest-point descriptor vectors.

At iteration t of running the AdaBoost algorithm, our criterion for choosing a good interest point is as follows.

Observe that, through (1), an interest point k gives rise to two distributions: one for the positive images of $\{J_i\}$ and the other for the negative ones. Treating each interest point as a weak classifier would produce a weighted misclassification error $\epsilon_t(k)$. It follows that the selected interest point at iteration t satisfies

$$k_t^* = \arg\min_k \epsilon_t(k). \tag{2}$$

Suppose that upon the completion of AdaBoost, K interest points have be chosen. ($K = 60$ in all our experiments.) Then, repeatedly performing the interest point selection by setting the target class to each of the emotion categories in turn would yield a set of $n = 7 \times K$ interest points, denoted as $\{k_1^*, \ldots, k_n^*\}$. To construct a feature vector $\mathbf{x}$ (of dimension n) for an arbitrary image I from its bag-of-features representation, we first detect the interest points of I, and then define the ith component of $\mathbf{x}$ by

$$x_i = d(k_i^*, I). \tag{3}$$

3. Partially-observed HCRFs

The HCRF model has been applied to object recognition [16] and gesture recognition [20]. In essence, the main idea behind HCRFs is to enrich CRFs by adding hidden states to capture complex dependencies or implicit structures in the training samples. The effect can be achieved by using more hidden variables, or by increasing the number of possible hidden states. Either way would lead to a graphical model with a large number of hidden-state configurations.

Unlike other graphical models with hidden states, HCRFs lack an explicit formulation (such as the transition probabilities in HMMs) on correlating the hidden variables other than the *clique* relations. Under such a general setting, it is difficult to foresee useful regularities from the hidden-state outputs by HCRFs. In our experiments we observe that applying HCRFs to image sequences of similar appearances may give rise to rather different hidden-state configurations.

Our use of HCRFs for facial expression recognition has a good analogy here. While recognizing the underlying emotion is our goal, uncovering the action units in each image frame turns out to be crucial for making the prediction. Also, in our training data, the action unit information is already provided in the peak (last) image of each sequence. These two aspects of consideration have prompted us to develop a new generalization for HCRFs—by introducing partially-observed hidden state variables. As we will explain that the modification does not affect the graph structure (see Figure 2), and requires no extra data labeling.

3.1. Energy function and data likelihood

In a partially-observed HCRF model, the hidden variables of a training sequence $\mathbf{s}$ are divided by $\mathbf{h} = \mathbf{h}_o \cup \mathbf{h}_u$,

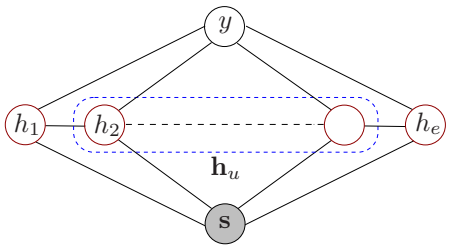


Figure 2. The shaded node corresponds to an observed sequence $\mathbf{s}$. h_1 and h_e are the starting and the ending hidden variables respectively, and $\mathbf{h}_u$ includes the remaining hidden variables.

where $\mathbf{h}_o$ and $\mathbf{h}_u$ denote the hidden variables whose (discrete) state values are respectively observed or unknown during training. Analogous to [16], the conditional probability of class label y and hidden variables $\mathbf{h}$ is given by

$$p(y, \mathbf{h}_o, \mathbf{h}_u \,|\, \mathbf{s}; \boldsymbol{\theta}) = \frac{\exp\{-E(y, \mathbf{h}_o, \mathbf{h}_u, \mathbf{s}; \boldsymbol{\theta})\}}{\sum_{y', \mathbf{h}'_o, \mathbf{h}'_u} \exp\{-E(y', \mathbf{h}'_o, \mathbf{h}'_u, \mathbf{s}; \boldsymbol{\theta})\}} \tag{4}$$

where $\boldsymbol{\theta}$ includes the parameters of the probabilistic model, and E is the *energy function*. It implies that

$$p(y, \mathbf{h}_o \,|\, \mathbf{s}; \boldsymbol{\theta}) = \sum_{\mathbf{h}_u} p(y, \mathbf{h}_o, \mathbf{h}_u \,|\, \mathbf{s}; \boldsymbol{\theta}). \tag{5}$$

And the data log-likelihood of $D = \{(\mathbf{s}^{(i)}, y^{(i)}, \mathbf{h}_o^{(i)})\}$, with partially-observed hidden states, is given by

$$L(\boldsymbol{\theta}) = \sum_i \log p(y^{(i)}, \mathbf{h}_o^{(i)} \,|\, \mathbf{s}^{(i)}; \boldsymbol{\theta}) - \frac{\|\boldsymbol{\theta}\|^2}{2\sigma^2} \tag{6}$$

where we have assumed a zero-mean Gaussian prior on $\boldsymbol{\theta}$. Learning a partially-observed HCRF model can now be accomplished by solving

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} L(\boldsymbol{\theta}). \tag{7}$$

In our implementation, *scaled conjugate gradient* [13] is used to find $\hat{\boldsymbol{\theta}}$ in (7). We emphasize that the partially-observed information about the hidden variables is provided only in training. Hence probability inference with the proposed model is exactly the same as with a regular HCRF. That is, given a new test sequence $\mathbf{s}$, we have

$$p(y \,|\, \mathbf{s}; \boldsymbol{\theta}) = \sum_{\mathbf{h}} p(y, \mathbf{h} \,|\, \mathbf{s}; \boldsymbol{\theta}). \tag{8}$$

3.2. Approximation with Bethe free energy

By far it may appear that adding partially-observed hidden variables to the learning of HCRFs is straightforward. The definition of $L(\boldsymbol{\theta})$ in (6) indicates that solving (7) requires evaluating $p(y, \mathbf{h}_o \,|\, \mathbf{s}; \boldsymbol{\theta})$ for each training sequence. When applying the belief propagation algorithm, the joint probability whose variables belong to a clique can be directly approximated by the corresponding belief. In our

case, the variables y and $\mathbf{h}_o$ most likely do not form a clique. Evaluating (5) requires to explicitly compute

$$\sum_{\mathbf{h}_u} \exp\{-E(y, \mathbf{h}_o, \mathbf{h}_u, \mathbf{s}; \boldsymbol{\theta})\} \tag{9}$$

and

$$\sum_{y', \mathbf{h}'_o, \mathbf{h}'_u} \exp\{-E(y', \mathbf{h}'_o, \mathbf{h}'_u, \mathbf{s}; \boldsymbol{\theta})\}, \tag{10}$$

which are both *intractable*. To resolve the difficulty, we consider using *Bethe free energy* [23] to approximate the *Helmhotz free energy*, which is simply the negative log of the partition function $Z(T)$ in the *Boltzmann's Law*:

$$p(\mathbf{x}) = \frac{1}{Z(T)} \exp\{-E(\mathbf{x})/T\}, \tag{11}$$

where E is the energy function and T is known to be the temperature. After running belief propagation, the Bethe free energy can be efficiently computed from the potential functions and the beliefs of the nodes and cliques.

The two quantities (9) and (10) can be thought as the partition functions of two particular graphical models: (9) is for the graph that the labels/states of the nodes y and $\mathbf{h}_o$ are given, and (10) is for the graph that all labels are unknown. In Figure 2, $\mathbf{h}_o = \{h_1, h_e\}$. Since the Bethe free energy is a function of the *beliefs*, it can be readily computed with belief propagation. Consequently, we can well approximate the values of (9) and (10).

4. Facial Expression Recognition

We are now in a position to lay out how facial expression recognition is done via inference with the learned graphical model, and also describe how to extend our method to process on-line image streams.

4.1. HCRFs for recognizing facial expressions

Let $(\mathbf{s} = \{I_i\}, y)$ denote a labeled sequence of facial expression images, and $\mathbf{x}_i$ be the feature vector of image I_i. Also let $\{h_1, h_2, \ldots, h_e\}$ be the set of hidden variables, each of which assumes a possible label corresponding to a combination of action units. In learning a partially-observed HCRF model, the labels of h_1 and h_e will be provided. Thus, according our notations, $\mathbf{h}_o = \{h_1, h_e\}$ and $\mathbf{h}_u = \{h_2, \ldots, h_{e-1}\}$. An illustration of such a graphical model is shown in Figure 2. (Note that the nodes with respect to h_1 and h_e are not adjacent, and not belong to the same clique.) Since there are three different types of cliques describing the relationships among the state and the observation nodes, three kinds of feature functions are considered in the definition of energy function E:

$$\begin{aligned} E(y, \mathbf{h}_o, \mathbf{h}_u, \mathbf{s}; \boldsymbol{\theta}) &= \sum_j \phi(\mathbf{s}, j) \cdot \theta_1(h_j) + \sum_j \theta_2(y, h_j) \\ &+ \sum_{(j,k)\in\mathcal{E}} \theta_3(y, h_j, h_k), \end{aligned} \tag{12}$$

where $\mathcal{E}$ is the set of edges linking the hidden nodes, and $\phi(\mathbf{s}, j)$ represents the observation at node j. In our formulation, we exploit the temporal information by setting $\phi(\mathbf{s}, j) = [\mathbf{x}_{j-1}^T, \mathbf{x}_j^T - \mathbf{x}_{j-1}^T]^T$.

4.2. Real-time incremental inference

By *incremental inference*, we are to design a message-passing scheme such that when processing an upcoming image frame I_t, all the previous inference results before time instant t will be available for making the new inference. Motivated by this desirable property, we define the message sent from hidden node h_{t-1} to h_t as

$$m_t(y, h_t; \boldsymbol{\theta}) = \sum_{h_{1:t-1}} \exp\{-E(y, h_{1:t}, \mathbf{x}_{1:t}; \boldsymbol{\theta})\} \tag{13}$$

where $h_{1:t}$ and $\mathbf{x}_{1:t}$ stand for $h_1, \ldots, h_t$ and $\mathbf{x}_1, \ldots, \mathbf{x}_t$, respectively. $m_t(y, h_t; \boldsymbol{\theta})$ can be further rewritten as

$$\sum_{h_{t-1}} \exp\{-E(y, h_{t-1:t}, \mathbf{x}_{t-1:t}; \boldsymbol{\theta})\}\, m_{t-1}(y, h_{t-1}; \boldsymbol{\theta}) \tag{14}$$

where according to (12)

$$\begin{aligned} E(y, h_{t-1:t}, \mathbf{x}_{t-1:t}; \boldsymbol{\theta}) &= \phi(\mathbf{s}, t) \cdot \theta_1(h_t) + \theta_2(y, h_t) \\ &+ \theta_3(y, h_t, h_{t-1}). \end{aligned} \tag{15}$$

From (14), we can derive an implementation for performing incremental inference. Namely, making inference at time t is done by evaluating

$$p(y \mid \mathbf{x}_{1:t}; \boldsymbol{\theta}) = \frac{\sum_{h_t} m_t(y, h_t; \boldsymbol{\theta})}{\sum_{y', h'_t} m_t(y', h'_t; \boldsymbol{\theta})}. \tag{16}$$

Indeed the foregoing scheme is not limited to processing on-line streaming data. Owing to the chain structure (among hidden variables), when the proposed incremental inference is applied to a whole image sequence $\mathbf{s}$, the resulting probability $p(y \mid \mathbf{s}; \boldsymbol{\theta})$ is the same as that by simultaneously considering all image frames. However, the labeling of hidden variables $h_t, t = 1, \ldots, e$ may be different due to that now only the information related to the preceding hidden nodes is available for each prediction. We will discuss further details regarding this matter in the next section.

In our formulation of incremental inference, the message $m_t(y, h_t; \boldsymbol{\theta})$ at time t carries all the previous inference results. This property is preferable when there would be only

one facial expression from the on-line streaming data. In practice such a restriction is not reasonable. To relax the limitation, we need some mechanism to ensure that probability inference of the current facial expression can be made without including the effects from the previous expressions. One way to achieve this is to detect the conclusion of an expression and reset the recognition system. However, there still lacks an efficient way for detecting the ending of an expression peak in real-time applications [5]. Here we discuss what conditions can assure the HCRF framework to make inference without considering the past information. Assume that frame t is the start of an expression. From (14) and (15), we denote those terms related to the past by

$$u_t(y, h_t; \boldsymbol{\theta}) = \sum_{h_{t-1}} \exp\{-\theta_3(y, h_t, h_{t-1})\}\, m_{t-1}(y, h_{t-1}; \boldsymbol{\theta}). \tag{17}$$

As is implied in Section 3, we can compute the Bethe free energy for each y. If (17) yields the same value for all h_t given any y, the information prior to frame t causes no effect at all since the message from $t-1$ acts as a uniform distribution. To facilitate this condition, we consider

$$f(x) = x^q, \text{ for } x > 0 \text{ and } 0 \le q < 1. \tag{18}$$

When f is repeatedly applied to any real value, the outcome will converge to 1. We call the parameter q in (18) the *pruning factor* that controls the rate to approach 1. At frame t, we "relax" the past information $u_t(y, h_t; \boldsymbol{\theta})$ by $f(u_t)$ and approximate $m_t(y, h_t; \boldsymbol{\theta})$ by

$$f(u_t(y, h_t; \boldsymbol{\theta})) \times \exp\left(\phi(\mathbf{s}, t) \cdot \theta_1(h_t) + \theta_2(y, h_t)\right). \tag{19}$$

Whenever a new expression starts, one can set the value of q to be close to 0 (to disregard the past information), and otherwise to be close to 1. In our experiments, the values are 0.1 and 0.9, respectively. The reason we choose 0.9 instead of 1 is mainly because such a tactic can result in a more noticeable drop in the inference probability when an expression is completed and image frames with neutral expression are reached. Nevertheless, detecting the end of an expression is still a hard problem. In our experiments, we use a heuristic way to decide. We compute the difference between the values $\max_{y,r;1\le r\le 5} p(y \mid \mathbf{x}_{1:t-r-1}; \boldsymbol{\theta})$ and $\max_y p(y \mid \mathbf{x}_{1:t-1}; \boldsymbol{\theta})$. If the difference is larger then 0.05, we say that a new expression starts and set $q = 0.1$.

5. Experimental Results and Discussion

We test our method with three sets of experiments. The first is to compare the sequence-wise classification outcomes derived by the partially-observed HCRF model (PO-HCRF for abbreviation) and other implementations. The second is to investigate the effects of using an extensive set of hidden labels to account for all action unit combinations in the training dataset. And for the last, we demonstrate that satisfactory real-time recognition performances can be achieved via the on-line incremental inference.

5.1. Dataset

The facial expression database used in our experiments is Cohn and Kanade's DFAT-504 dataset [9]. It contains 486 sequences produced from 97 subjects. There are about 1 to 9 sequences for each subject. In this dataset, the action unit information is provided only for the last frame (peak image) of each sequence, and occasionally it may not be sufficient for identifying the emotion category. We have labeled 392 sequences by referencing Table 2 in Zhang and Ji [24].

5.2. Sequence-wise classification

In this set of experiments, we are to demonstrate the efficiency of using PO-HCRFs to analyze sequences of facial expressions. We begin by setting the total possible hidden labels/states to 14, and further consider two cases. For case one, the hidden labels range from 1 to 7 (1 will be reserved for `neutral`), and each corresponds to some combination(s) of action units. And for the other case, the meaningful labels are from 1 to 9. In both cases the remaining labels are not explicitly defined so that it leaves some degree of freedom for the training process since certain image frames are hard to be labeled. The purpose of adding two more hidden labels is to enable our system to distinguish whether (i) a `joy` face is with mouth open (*i.e.*, AU25), and (ii) a `fear` face is with a "shock" (AU2: outer brow raiser or AU5: upper lid raiser). (See Table 1 for details.)

Besides comparing the results derived by HCRFs and PO-HCRFs, we have implemented the method of Bartlett *et al*. [2] for image sequences. It is done by applying their classifier to each image frame, and the label of a sequence is decided by a majority vote. (Note that those classified as `neutral` are not counted.) For a more insightful study, we also consider adapting their method by using our feature selection scheme. In Table 2, we report the recognition accuracies. The notations PO-HCRF7 and PO-HCRF9 indicate the number of meaningful hidden labels used in their implementation. Examples of illustration on the labeling results are provided in Table 1. A confusion matrix by PO-HCRF9 is given in Table 3. Overall, the experimental results show that PO-HCRF can achieve better accuracy rates, and output more coherent hidden labels.

Indeed the fact that all test sequences start with a neutral frame is not required by our method. To verify this claim, we apply PO-HCRF to all the last half sequences, and obtain a slightly better recognition rate, 93.11%. Two such examples are plotted in Figure 3, where the broken-line graphs show the probabilities of different emotions for the given (last half) facial expression sequences.

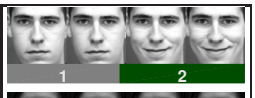
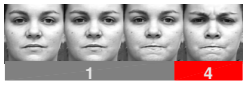
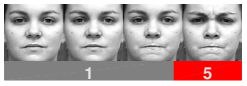
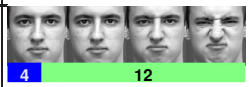
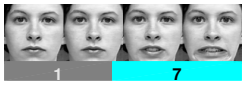
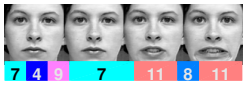
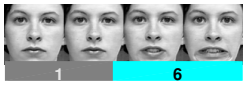
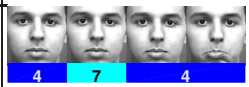
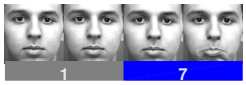
Emotion	Example	Basic AUs	Hidden label		HCRF	PO-HCRF7	PO-HCRF9
joy		6+12	2	2			
		6+12+25	2	3			
surprise		1+2+5+25+27	3	4			
anger		4+7+17+23+24	4	5			
disgust		4+9+17	5	6			
fear		1+20+25+2	6	7			
		1+20+25+5	6	7			
		1+20+25	6	8			
sadness		1+15+17	7	9			

Table 1. Hidden labels vs. emotions for PO-HCRF7 and PO-HCRF9 are shown in the first four columns. Note that the set of action units with respect to a hidden label is not the only possible combination. The remaining columns include illustrations of the labeling results from our sequence-wise classification experiments. The four face images shown in each example are the 1st, the $\frac{1}{3}$rd, the $\frac{2}{3}$rd, and the last ones of the sequence. The color bar and the numbers signify the hidden labels of the images, while gray color is for `neutral`.

Bartlett *et al.* [2]	Our feature	HCRF	PO-HCRF7	PO-HCRF9
84.69%	87.50%	86.48%	91.33%	92.86%

Table 2. Accuracy rates for sequence-wise recognition.

	joy	sadness	surprise	anger	fear	disgust
joy	98.0%	0.0%	0.0%	0.0%	2.0%	0.0%
sadness	0.0%	97.5%	0.0%	2.5%	0.0%	0.0%
surprise	0.0%	1.4%	98.6%	0.0%	0.0%	0.0%
anger	2.8%	22.2%	0.0%	69.4%	2.8%	2.8%
fear	7.0%	1.8%	1.8%	1.8%	87.7%	0.0%
disgust	2.2%	0.0%	0.0%	6.7%	2.2%	88.9%

Table 3. Confusion matrix of PO-HCRF9.

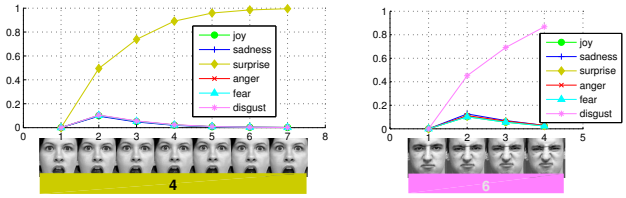


Figure 3. Examples of last half sequence labeling from Table 1.

5.3. More on the labeling results

Interesting observations can be inferred from the foregoing experiments. First, a facial expression typically does not progress "linearly." For example, in a `joy` sequence as in Figure 4a, it would be unnatural to use interpolation based on the neutral and peak images to approximate the middle ones. In this case, the lip corner pulls up first (AU12) and then the mouth opens gradually (AU25). Our labeling results show a good match for such face actions. Second, a more dramatic example is given in Figure 4b where hidden label 8 for `fear` appears in labeling a `joy` sequence. This can be explained in terms of action units. In some facial expressions of `fear`, the mouth corner is pulled up (AU12), as in Figures 4c and 4d. Furthermore, the basic set of the action units for `fear` comprises AU1 (inner brow raiser), AU20 (lip stretcher) and AU25 (lips part). The last two, AU20 and AU25, may also happen in a `joy` sequence.

In Figure 4b, the two labeling results derived by the implementations for sequences and incremental inference are shown. Although, from (16), the two approaches would output the same class probability at each time instant, the labeling for the hidden variables can be different. The distinction is caused by that labeling with incremental inference cannot reference information beyond its current image frame.

5.4. All action unit combinations

Among our selected 392 training sequences, there are 15 action units (see Figure 5) occurred most frequently, and in total 100 combinations from them appeared in all of the last

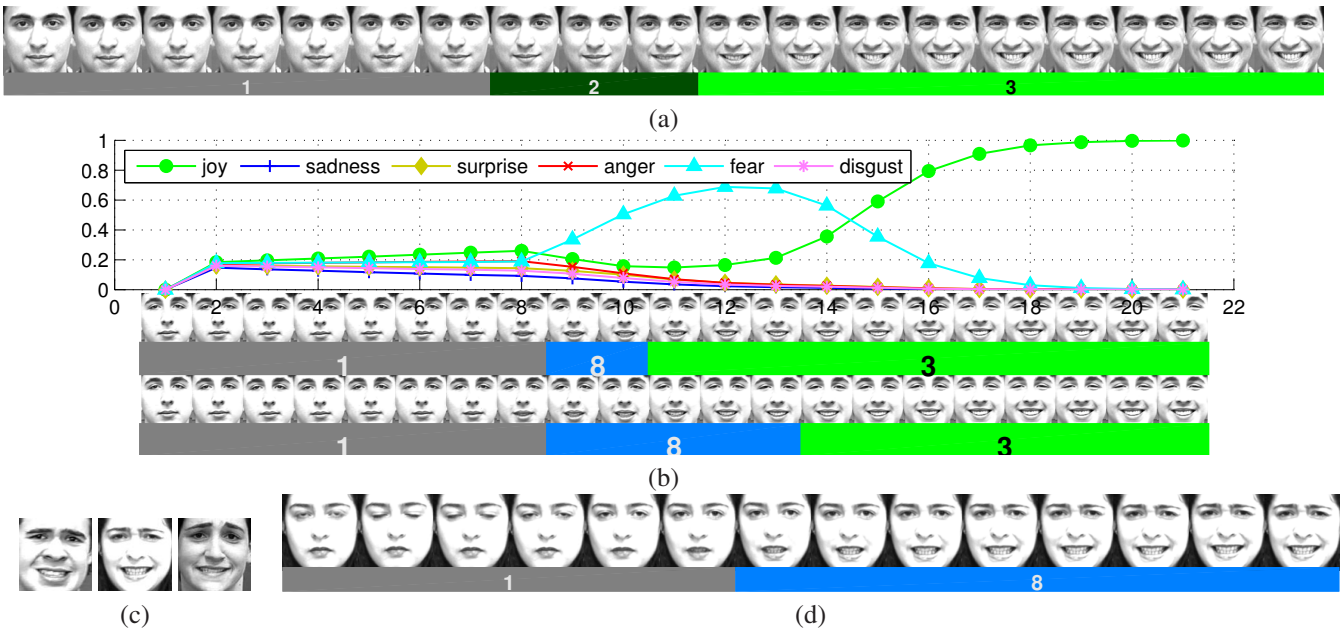
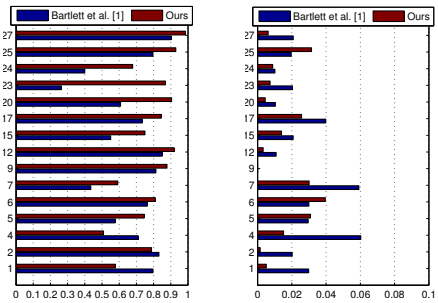


Figure 4. (a) A `joy` sequence. (b) The plot for the inference probabilities for the six emotion c1asses. Below the plot are the hidden-node labeling results respectively derived by processing the whole sequence (top) and by incremental inference (bottom). In both results, despite some of the images have been labeled as `fear`, the output emotion class label is still correct. (c) and (d) Some examples of the `fear` faces with action unit AU12 (mouth corners pull up), which also could occur in a `joy` facial expression.

image frames. Thus, we use an extensive set of 100 hidden labels, and carry out PO-HCRF100 to achieve a five-fold cross-validation rate of 90.05%.

We consider the labeling results at the first and the last frames to evaluate the recognition rate and the false alarm rate for each action unit (see Figure 5). Most of our results are better than or comparable to those in [2] except for AU1 and AU4. The main reason for the degradation is due to that the distributions of AU1 and AU4 are dominated by those of other action units for the `fear` and `sadness` sequences. Finally, we note that the better action unit recognition rates in Tong *et al.*'s work [18, 19] are derived by using action unit information from all training image frames, while in our method only the two end frames are used.



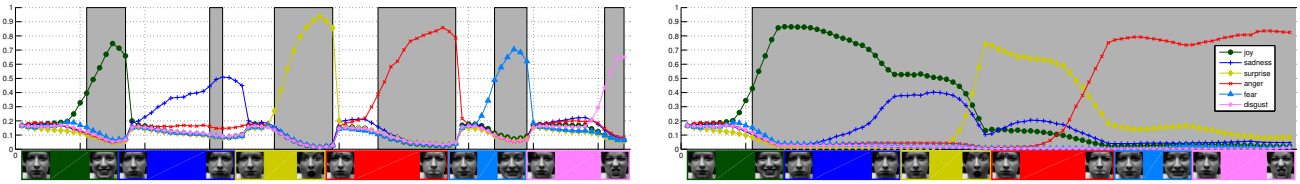
(a) Recognition rate (b) False alarm rate

Figure 5. Evaluation for classifying each action unit.

5.5. Implementation for real-time applications

Five-fold cross-validation is adopted to estimate the on-line recognition accuracy by our real-time implementation. At each fold, we first run PO-HCRF9 and PO-HCRF100 separately on each training sequence, and frame-wise derive the inference probabilities for the six emotion classes (like the plot in Figure 4b). We also compute the *entropy* and its first derivative at each time instant. The six probabilities and the two entropy-related quantities form a feature vector of dimension 8. Suppose now we want to learn a decision function for the emotion of `joy` via the Perceptron algorithm. The positive/negative data are those image frames from the `joy`/non-`joy` sequences that the inference probability for `joy` in their feature vector is the largest. The remaining cases can be analogously learned.

For testing, we perform incremental inference by treating a test sequence as an image stream. At each time instant, the decision function corresponding to the emotion class that currently has the highest inference probability will be applied, and a decision on the emotion label will be made if the response is positive. With this setting, the accuracy rate we achieve is 80.10% with 9.18% false alarm rate for PO-HCRF9 and 80.36% with 8.93% false alarm rate for PO-HCRF100. For justifying the advantage of using the pruning factor q in our on-line implementation, we further *simulate* image streams by concatenating all the sequences of the same person in the training dataset. An example of such comparison results is illustrated in Figure 6.



(a) With the pruning factor. (b) Without the pruning factor.

Figure 6. The effect of the pruning factor. The plotted inference probabilities suggest how likely the subject is making the expressions. In the bottom, each color section with faces indicates the period of the sequence comes from which expression and for each period only the neutral and peak faces are displayed. Whenever a time instant is in the gray areas, our system will issue that the subject is making the expression currently with the highest probability.

Acknowledgements

This work is supported in part by NSC grants 95-2221-E-001-031 and 97-2221-E-001-019.

References

[1] Z. Ambadar, J. Schooler, and J. Cohn. Deciphering the enigmatic face: The importance of facial dynamics to interpreting subtle facial expressions. *Psychological Science*, 2005.

[2] M. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, and J. Movellan. Recognizing facial expression: Machine learning and application to spontaneous behavior. In *CVPR*, pages II: 568–573, 2005.

[3] J. Bassili. Emotion recognition: The role of facial movement and the relative importance of upper and lower areas of the face. *Journal of Personality and Social Psychology*, 37:2049–2058, 1979.

[4] I. Cohen, N. Sebe, A. Garg, L. Chen, and T. Huang. Facial expression recognition from video sequences: Temporal and static modeling. *CVIU*, 91(1-2):160–187, July 2003.

[5] F. de la Torre, J. Campoy, Z. Ambadar, and J. Cohn. Temporal segmentation of facial behavior. In *ICCV*, pages 1–8, 2007.

[6] G. Donato, M. Bartlett, J. Hager, P. Ekman, and T. Sejnowski. Classifying facial actions. *PAMI*, 21(10):974–989, October 1999.

[7] P. Ekman and W. Friesen. The facial action coding system: A technique for the measurement of facial movement. In *Consulting Psychologists*, 1978.

[8] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Proc. European Conf. Computational Learning Theory*, pages 23–37, Barcelona, Spain, 1995.

[9] T. Kanade, J. Cohn, and Y. Tian. Comprehensive database for facial expression analysis. In *AFGR*, pages 46–53, 2000.

[10] A. Kapoor, Y. Qi, and R. Picard. Fully automatic upper facial action recognition. In *International Workshop on Analysis and Modeling of Faces and Gestures*, pages 195–202, 2003.

[11] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, pages 282–289. Morgan Kaufmann, San Francisco, CA, 2001.

[12] D. Lowe. Object recognition from local scale-invariant features. In *ICCV*, pages 1150–1157, 1999.

[13] I. Nabney. *Netlab: Algorithms for Pattern Recognition*. Springer-Verlag London Ltd, 2004.

[14] M. Pantic and L. Rothkrantz. An expert system for multiple emotional classification of facial expressions. In *Proc. IEEE Conference of Tools with Artificial Intelligence*, pages 113–120, 1999.

[15] M. Pantic and L. Rothkrantz. Automatic analysis of facial expressions: The state of the art. *PAMI*, 22(12):1424–1445, December 2000.

[16] A. Quattoni, M. Collins, and T. Darrell. Conditional random fields for object recognition. In L. Saul, Y. Weiss, and L. Bottou, editors, *NIPS 17*, pages 1097–1104. MIT Press, Cambridge, MA, 2005.

[17] Y. Tian, T. Kanade, and J. Cohn. Recognizing action units for facial expression analysis. *PAMI*, 23(2):97–115, February 2001.

[18] Y. Tong, W. Liao, and Q. Ji. Inferring facial action units with causal relations. In *CVPR*, pages II: 1623–1630, 2006.

[19] Y. Tong, W. Liao, and Q. Ji. Facial action unit recognition by exploiting their dynamic and semantic relationships. *PAMI*, 29(10):1683–1699, October 2007.

[20] S. Wang, A. Quattoni, L. Morency, D. Demirdjian, and T. Darrell. Hidden conditional random fields for gesture recognition. In *CVPR*, pages II: 1521–1527, 2006.

[21] P. Yang, Q. Liu, X. Cui, and D. Metaxas. Facial expression recognition using encoded dynamic features. In *CVPR*, 2008.

[22] P. Yang, Q. Liu, and D. Metaxas. Boosting coded dynamic features for facial action units and facial expression recognition. In *CVPR*, pages 1–6, 2007.

[23] J. Yedidia, W. Freeman, and Y. Weiss. Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Trans. on Information Theory*, 51(7):2282–2312, 2005.

[24] Y. Zhang and Q. Ji. Active and dynamic information fusion for facial expression understanding from image sequences. *PAMI*, 27(5):699–714, May 2005.

[25] G. Zhao and M. Pietikäinen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *PAMI*, 29(6):915–928, June 2007.