Recognition of visual speech elements using adaptively boosted hidden Markov models

Foo, Say Wei; Lian, Yong; Dong, Liang

2004

Foo, S. W., Lian, Y., & Dong, L. (2004). Recognition of visual speech elements using adaptively boosted hidden Markov models. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(5), 693-705.

<https://hdl.handle.net/10356/91495>

<https://doi.org/10.1109/TCSVT.2004.826773>

© 2006 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE. This material is presented to ensure timely dissemination of scholarly and technical work. Copyright and all rights therein are retained by authors or by other copyright holders. All persons copying this information are expected to adhere to the terms and constraints invoked by each author's copyright. In most cases, these works may not be reposted without the explicit permission of the copyright holder.

Downloaded on 26 Apr 2023 16:24:47 SGT

Recognition of Visual Speech Elements Using Adaptively Boosted Hidden Markov Models

Say Wei Foo, *Senior Member, IEEE*, Yong Lian, *Senior Member, IEEE*, and Liang Dong

Abstract—The performance of automatic speech recognition (ASR) system can be significantly enhanced with additional information from visual speech elements such as the movement of lips, tongue, and teeth, especially under noisy environment. In this paper, a novel approach for recognition of visual speech elements is presented. The approach makes use of adaptive boosting (AdaBoost) and hidden Markov models (HMMs) to build an AdaBoost-HMM classifier. The composite HMMs of the AdaBoost-HMM classifier are trained to cover different groups of training samples using the AdaBoost technique and the biased Baum–Welch training method. By combining the decisions of the component classifiers of the composite HMMs according to a novel probability synthesis rule, a more complex decision boundary is formulated than using the single HMM classifier. The method is applied to the recognition of the basic visual speech elements. Experimental results show that the AdaBoost-HMM classifier outperforms the traditional HMM classifier in accuracy, especially for visemes extracted from contexts.

Index Terms—Adaptive boosting (AdaBoost), automatic lip reading, hidden Markov model (HMM), visual speech processing.

I. INTRODUCTION

THE technique of retrieving speech content from visual clues such as the movement of the lips, tongue, and teeth is commonly known as automatic lip reading. Recently, the multimedia signal processing community has shown increasing interest for research on this area. It has been shown in [2]–[7], [9], [10] that the performance of a purely acoustic-based speech recognition system will improve with additional input from the visual speech elements, especially when the speech signal has low signal-to-noise ratio (SNR). Visual speech processing can also be applied to areas such as speaker verification, multimedia telephony for the hearing impaired, cartoon animation, and video games.

In 1984, Petajan developed probably the first visual speech processor [2]. In this system, the distance of geometric measures among different mouth shapes was computed for identifying the visual representations of word productions. In 1993, Goldschen extended Petajan's design by using the hidden Markov model (HMM) as the visual classifier [39].

Subsequent researches on implementing visual speech processing/visual audio integration include Neural Network [40], Time-Delayed Neural Network (TDNN) [55], [56], fuzzy logics [57], and Boltzmann zippers [41]. Among the various techniques for visual speech processing studied so far, the HMM holds the greatest promise due to its capabilities in modeling and analyzing temporal processes. In Goldschen's system, HMM classifiers were explored for recognizing a closed set of TIMIT sentences [39], [42]. In 1990, Welch *et al.* explored audio-to-visual mapping using HMM for building speech-driven models [8]. Silsbee and Bovik applied HMM to identify isolated words [43], [44]. Tomlinson *et al.* suggested a cross-product HMM topology, which allows asynchronous processing of visual signals and acoustic signals [6]. Luetttin *et al.* used HMMs with an early integration strategy for both isolated digit recognition and connected digit recognition [10]. In recent years, coupled HMM, product HMM and factorial HMM have been explored for audio-visual integration [45]–[48]. Details of the HMM-based visual speech processing techniques can be found in [13]–[15], [51].

Most of the HMM-based visual-only systems reported in literature take an individual word as the basic recognition unit and an HMM is trained to model it. Such an approach works well with limited vocabulary such as digit set [10], [13] or a small number of AVletters [12], isolated words, or nonsense words [59], but it is difficult to extend such methods to cover a large vocabulary because a great number of samples have to be collected to train all the possible word models that may appear in the speech. A text-independent speech recognition system, on the other hand, not only is able to identify words within its vocabulary, but is also able to recognize new words. To make a visual speech recognition system text-independent, one approach is to use subword visual speech units. In this paper, visemes are chosen as the subword units. By modeling and recognizing visemes, the system may be further trained to recognize words by breaking them into a sequence of visemes.

In visual speech processing, the smallest visibly distinguishable unit of visual speech element is commonly referred to as viseme [16]. It indicates a short period of lip movement that may be repeated in different articulations. Like phonemes, which are the basic building blocks of sound of a language, visemes are the basic constituents for the visual representations of words. The relationship between phonemes and visemes is a many-to-one mapping. For example, although phonemes /b/, /m/, and /p/ are acoustically distinguishable sounds, they are grouped into one viseme category as they have similar sequence of mouth shapes and are visually confusable. An early viseme grouping was suggested by Binnied *et al.* [49]

Manuscript received April 30, 2003; revised September 3, 2003.

S. W. Foo is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 (e-mail: eswfoo@ntu.edu.sg).

Y. Lian is with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 119260 (e-mail: eleliany@nus.edu.sg).

L. Dong is with the Digital Systems and Applications Lab, Department of Electrical and Computer Engineering, National University of Singapore, Singapore 119260 (e-mail: engp0564@nus.edu.sg).

Digital Object Identifier 10.1109/TCSVT.2004.826773

and was applied to some identification experiments such as [50]. Viseme groupings in [16] are obtained by analyzing the stimulus–response matrices of the perceived visual signals. The MPEG-4 multimedia standards adopted the same viseme grouping strategy for face animation, in which 14 viseme groups are included [22]. However, different groupings may be adopted to fulfill specific requirements [52].

In the acoustic speech processing domain, a number of speech databases, such as TIMIT and YOHO, developed under sponsorship from the United States government agency, have been widely adopted by the research community. The TIMIT corpus of read speech has been designed to provide speech data for the acquisition of acoustic-phonetic knowledge and for the development and evaluation of automatic speech recognition systems. The YOHO database is a corpus designed for speaker verification purposes. In visual speech processing, although some audio-visual speech databases have been proposed in recent years, they are not widely accepted by the multimedia community [53], [54], [59]. The data used in most visual speech experiments are independently recorded.

A viseme consists of a time sequence of lip shapes. When extracted from spoken sentences, the same phonetic sound may have slightly different set of lip shapes affected by the adjoining sounds/visemes. Thus, visemes demonstrate polymorphism under various contexts. The traditional single-HMM classifier, e.g., the maximal likelihood (ML) model trained with the Baum–Welch method, is most of the time incompetent to cover symbols that vary too much from the norm. To model the lip motion with enough accuracy for extracting speech information, adoption of multiple HMMs becomes necessary. In this paper, a novel multiple-HMM classifier using the adaptive boosting (AdaBoost) technique is proposed for the classification of visemes. The “boosted” classifier comprises a number of HMMs with each of them focusing on certain training samples. By suitably combining the decisions made by the component HMMs, a more complex decision boundary can be established to deal with the variations (polymorphism) of visemes. Experiments are carried out to identify the independently produced visemes and those extracted from various contexts. As identification of the visemes, especially those extracted from contexts, from merely visual signals is a relatively fresh research area, there are few previous experimental results for reference. In this paper, the comparison is made between the AdaBoost-HMM classifier and single-HMM classifier. While identifying independently produced visemes, the false rejection rates (FRRs) of the two classifiers are not significantly different. However, the FRRs of the boosted classifier are much lower than those of the single HMM classifier in identifying the visemes extracted from contexts. The comparison, thus, proves that the AdaBoost-HMM provides better adaptability and accuracy than single HMM in dealing with the variations.

The organization of this paper is as follows. Section II describes the structure of the proposed system. A brief review of the HMM principles and AdaBoosting method is given in Sections III and IV, respectively. The description of the procedures of HMM boosting is presented in Section V. Since the system deals with a practical problem, some strategies that

Fig. 1. Block diagram of a viseme recognition system.

facilitate the implementation are also presented in the paper. The performance of the system in recognition of visemes is described in Section VI and the conclusion is given in Section VII.

II. OVERVIEW OF THE PROPOSED SYSTEM

Visual speech processing may be carried out at two stages. In the first stage, the basic visual speech elements such as visemes are identified; in the second stage, recognition of connected words is performed. In HMM-based classifiers, the first stage is usually realized via HMMs, where a viseme is characterized with one HMM. For continuous speech, the connected-visemes are usually identified by constructing HMM chains using exhaustive means such as level building [18]. In this paper, discussion is focused on the first stage, where visemes are the basic symbols to be identified.

The block diagram of a traditional HMM classifier is shown in Fig. 1. In a system of identifying K visemes, K classifiers are included. The input to the system is the vector sequence of an unknown viseme production, denoted by x^T . In the traditional approach, each subclassifier is a single HMM that is trained with the samples of a specific viseme. Each HMM determines the probability of occurrence of the unknown viseme. By comparing the probabilities $P_1, P_2, \dots, P_K$, a decision is made about the identity of the unknown viseme.

Traditional approaches for optimizing the viseme classifier focuses mainly on improving the performance of the individual HMM, e.g., setting proper HMM parameters, maximizing the probabilities of the given samples, minimizing the cross-entropy, and so on. For the proposed AdaBoost-HMM classifier, a new system of management is employed where multiple HMMs are adopted as the subclassifier instead of a single HMM for each viseme. The block diagram of the structure of a typical subclassifier of the proposed system is presented in Fig. 2. The subclassifier of the k th viseme is comprised of L_k HMMs, where L_k is the number of boosting rounds (or the number of the composite HMMs). The outputs from the L_k HMMs $P(x^T|\theta_1), P(x^T|\theta_2), \dots, P(x^T|\theta_{L_k})$ are processed according to a desired probability synthesis rule, where $P(x^T|\theta_i)$ ($i = 1, 2, \dots, L_k$) denotes the conditional probability of occurrence of x^T given the HMM model θ_i . The synthesized likelihood $\bar{P}(x^T|\Theta_k)$ is then submitted to

Fig. 2. Block diagram of an adaptively boosted multiple-HMM subclassifier.

the decision module to determine the identity of the unknown viseme.

III. REVIEW OF HMMs

The HMM is essentially a division of a process into a number of discrete states. While using an HMM to analyze a stochastic process, the observation sequence, say, $o^T = (o_1, o_2, \dots, o_t, \dots, o_T)$ where o_t is the t th symbol appeared in the sequence, is assumed to be emitted from a sequence of hidden states $s^T = (s_1, s_2, \dots, s_T)$, where $s_i \in S^N$ and $S^N = \{S_1, S_2, \dots, S_N\}$ is the state set of the HMM. Each state maintains a probability function or probability density function to indicate the likelihood of emitting certain symbols. The states are interconnected with each other by some state transition coefficients, which indicate the likelihood of a state repeating itself or transiting to another state. The relationship between the explicit process (observations), the hidden process (states) and the HMM is depicted in Fig. 3, where $P(S_j|S_i)$ ($i, j = 1, 2, \dots, N$) is the likelihood of transition from S_i to S_j and $P(o_t|S_i)$ ($t = 1, 2, \dots, T$) is the likelihood of emitting o_t in state S_i . If the output of an HMM takes discrete and finite values, e.g., from a finite symbol set $O^M = \{O_1, O_2, \dots, O_M\}$, the HMM is called discrete; if the output takes infinite continuous values, the HMM is called continuous. For ease of subsequent discussion on AdaBoost-HMM, the mathematical model of discrete HMMs is summarized in the following paragraphs.

Assume that $O^M = \{O_1, O_2, \dots, O_M\}$ is the set of discrete symbol alphabet and $S^N = \{S_1, S_2, \dots, S_N\}$ is the set of states. An N -state M -symbol HMM $\theta(\pi, A, B)$ is determined with the following three components.

- 1) The probabilities of the initial states $\pi = [\pi_i]_{1 \times N} = [P(s_1 = S_i)]_{1 \times N}$ ($1 \leq i \leq N$), where s_1 is the first state in the state chain.
- 2) The state transition matrix $A = [a_{ij}]_{N \times N} = [P(s_{t+1} = S_j | s_t = S_i)]_{N \times N}$ ($1 \leq i, j \leq N$), where s_{t+1} and s_t are the $t+1$ th and the t th states.
- 3) The symbol emission matrix $B = [b_{ij}]_{N \times M} = [P(o_j | S_i)]_{N \times M}$ ($1 \leq i \leq N, 1 \leq j \leq M$).

 Fig. 3. Relation between the observation sequence and the state sequence of an HMM with N states.

For the input sequence o^T , the likelihood $P(o^T|\theta)$ is measured by means of the forward process [17]. The optimal hidden states s^T are revealed through Viterbi matching [17]. Training of the HMM serves to determine the parameters to satisfy certain criterion function such as the likelihood $P(o^T|\theta)$ or the mutual information [18], [27]. If an HMM is properly trained, the states may well reveal the features of the process investigated. Thus, training of the HMM is usually the most important aspect in the construction of an HMM classifier.

IV. REVIEW OF ADAPTIVE BOOSTING

The AdaBoost algorithm is a method for converting a weak classifier into a strong one. This method maintains a distribution of weights over the training set. A new classifier is trained by weighted samples and the final decision is made by summing up the subdecisions of all the subclassifiers. Assume that in a two-class identification problem (Class 1 versus Class 2), the training set is comprised of R samples $\{y_1, d_1\}, \{y_2, d_2\}, \dots, \{y_R, d_R\}$. y_i is the observed data and $d_i = \{-1, +1\}$ is the label identifier where $d_i = -1$ denotes Class 1 and $d_i = +1$ denotes Class 2. The process of adaptive boosting involves of a series of rounds ($t = 1, 2, \dots, T$) of weight adjusting and classifier training [19]. Let $D_t(i)$ stand for the weight that is assigned to the i th training sample in

Fig. 4. Steps of adaptive boosting algorithm.

the t th boosting round and D_t denote the set of the weights $\{D_t(1), D_t(2), \dots, D_t(R)\}$. The steps of adaptive boosting are presented below.

- 1) Initially, assign the weight $D_1(i) = 1/R$ ($i = 1, 2, \dots, R$) to the R training samples $\{y_1, d_1\}, \{y_2, d_2\}, \dots, \{y_R, d_R\}$. Note that $D_1(i)$ ($i = 1, 2, \dots, R$) follows a uniform distribution.
- 2) Train the t th classifier θ_t in the t th boosting round using the distribution D_t , starting with $t = 1$.
- 3) Formulate the hypothesis, $h_t(y_i) \rightarrow \{-1, +1\}$ ($i = 1, 2, \dots, R$) and calculate the error $\varepsilon_t = P(h_t(y_i) \neq d_i)$ for θ_t .
- 4) Calculate $w_t = (1/2) \ln((1 - \varepsilon_t)/\varepsilon_t)$, which weights the importance of the classifier θ_t .
- 5) Update the distribution: $D_{t+1}(i) = (D_t(i)/Z_t) \times \begin{cases} e^{-w_t}, & \text{if } h_t(y_i) = d_i \\ e^{w_t}, & \text{if } h_t(y_i) \neq d_i \end{cases}$, where Z_t is a normalization factor to make D_{t+1} a distribution.

The procedures of adaptive boosting are also presented as a flow chart in Fig. 4. Steps 2–5 are repeated until the error rate of the classifier exceeds 0.5 or after a given number of training epochs. A series of classifiers $\theta_1, \theta_2, \dots, \theta_T$ and weights $w_1, w_2, \dots, w_T$ are obtained at the end of the above procedures. In this paper, the T subclassifiers, together with the weights assigned to them, are looked as an integral entity and are referred to as an AdaBoost classifier. For an unknown input y , the subdecisions made by the T composite classifiers are synthesized using $H(y) = \text{sign}(\sum_{t=1}^T w_t h_t(y))$. If $H(y) < 0$, $y \in \text{Class 1}$, otherwise $y \in \text{Class 2}$.

The objective of boosting is to minimize the training error $\varepsilon(H) = (1/R) \sum_{i=1}^R [1 : H(y_i) \neq d_i]$ of the final hypothesis. Schapire and Singer proved that $\varepsilon(H)$ is bounded as follows [33]:

$$\varepsilon(H) \leq \frac{1}{R} \sum_{i=1}^R \exp[-d_i f(y_i)] \quad (1)$$

where $f(y_i) = \sum_{t=1}^T w_t h_t(y_i)$. By unraveling the recursive definition of D_t , we have

$$\frac{1}{R} \sum_{i=1}^R \exp[-d_i f(y_i)] = \prod_{t=1}^T Z_t \quad (2)$$

where

$$Z_t = \sum_{i=1}^R D_t(y_i) \exp[-w_t d_i h_t(y_i)]. \quad (3)$$

Equation (1) suggests that the training error can be reduced most rapidly by choosing w_t and h_t at each round to minimize Z_t . In the case of binary hypotheses, this leads to the choice of w_t in (4), which is adopted in Step 4 of the above-mentioned boosting steps [32]

$$w_t = \frac{1}{2} \ln \left(\frac{1 - \varepsilon_t}{\varepsilon_t} \right). \quad (4)$$

If the training error of θ_t (the classifier obtained at the t th boosting round) is less than 0.5, say, $\varepsilon_t = (1/2) - \gamma_t$ ($\gamma_t > 0$), it is also proved in [20] that the training error is bounded as in (5)

$$\begin{aligned} \varepsilon(H) &\leq 2^T \prod_{t=1}^T \sqrt{\varepsilon_t(1 - \varepsilon_t)} \\ &= \prod_{t=1}^T \sqrt{1 - 4\gamma_t^2} \leq \exp \left(-2 \sum_{t=1}^T \gamma_t^2 \right). \end{aligned} \quad (5)$$

Equation (5) shows that if only the individual classifier has a classification error less than 0.5 (or, equivalently, a classification rate greater than 0.5), the overall error rate should decrease exponentially [20]. The boosted classifier may generate a hypothesis with an arbitrary low rate of error as boosting continues.

V. ADABOOSTING HMM

Boosting is carried out during the training phase of the composite HMMs. For the K class identification problem, the purpose of HMM AdaBoosting is to train a set of HMMs that represent or span the distribution of the training samples. In the application of AdaBoosting strategy to the construction of a multi-HMM classifier, the following two important issues arise: 1) the choice of base training algorithm and 2) the measurement of classification error. These are discussed in detail in the following paragraphs.

A. Base Training Algorithm

The most popular training algorithm for HMM is the Baum–Welch algorithm [18]. The Baum–Welch method makes use of ML estimation. The parameters are adjusted toward the direction of maximizing the probability $P(x^T | \theta)$ where x^T is the training sample. Since the expectation-maximization (EM) iterations are applied in the training procedures, the method has the advantage of ease of implementation and speed of convergence. However, the parameters of the HMM obtained from the Baum–Welch estimation are solely determined by the “correct” samples. The trained HMM is, therefore, not guaranteed to discriminate similar samples well.

An alternative to the Baum–Welch training algorithm is the maximum mutual information (MMI) estimation [27]. This method is a discriminative training method because it increases the *a posteriori* probabilities of the model corresponding to the training data. However, the analytical solution to the MMI criterion function is difficult to realize, the implementation of MMI estimation is, hence, tedious.

As mentioned in Section IV, adaptive boosting has loose requirements on the selection of base classifiers. As long as the training error of the individual classifier is less than 0.5, the training error of the AdaBoost classifier will drop. The same requirements apply to HMMs. If the training error of the composite HMMs is forced to be less than 0.5, the error rate of the AdaBoost-HMM classifier will also decrease as boosting continues. Besides this, since each of the multiple HMMs has to be individually trained, reduction of the computational load per HMM is also an important consideration. Considering these two factors, the Baum–Welch algorithm, which is less computationally intensive, is adopted.

Assume that $X_k = \{x_1^k, x_2^k, \dots, x_{R_k}^k : d_k\}$ are R_k training samples (sequences) of Class d_k , where $x_l^k = (o_1^{k,l}, o_2^{k,l}, \dots, o_{T_l}^{k,l})$ ($l = 1, 2, \dots, R_k$) is a T_l -length observation sequence and $o_i^{k,l}$ ($i = 1, 2, \dots, T_l$) is the i th symbol appeared in the sequence. For the N -state M -symbol HMM $\theta(\pi, A, B)$ mentioned in Section III, we define the forward variables $\alpha_t^{k,l}(i) = P(o_1^{k,l}, o_2^{k,l}, \dots, o_t^{k,l}, s_t = S_i | \theta)$ and backward variables $\beta_t^{k,l}(i) = P(o_{t+1}^{k,l}, o_{t+2}^{k,l}, \dots, o_{T_l}^{k,l} | s_t = S_i, \theta)$ for x_l^k , the parameters of the HMM are then estimated through the following EM recursion [18]:

$$\bar{a}_{ij} = \frac{\sum_{l=1}^{R_k} \frac{1}{P_l} \sum_{t=1}^{T_l-1} \alpha_t^{k,l}(i) a_{ij} b_j(o_{t+1}^{k,l}) \beta_{t+1}^{k,l}(j)}{\sum_{l=1}^{R_k} \frac{1}{P_l} \sum_{t=1}^{T_l-1} \alpha_t^{k,l}(i) \beta_{t+1}^{k,l}(j)} \quad (6a)$$

$$\bar{b}_j(O_m) = \frac{\sum_{l=1}^{R_k} \frac{1}{P_l} \sum_{\substack{t=1 \\ s.t. o_t^{k,l}=O_m}}^{T_l-1} \alpha_t^{k,l}(j) \beta_t^{k,l}(j)}{\sum_{l=1}^{R_k} \frac{1}{P_l} \sum_{t=1}^{T_l-1} \alpha_t^{k,l}(j) \beta_t^{k,l}(j)} \quad (6b)$$

where O_m is the m th symbol in the symbol set and $P_l = P(x_l^k | \theta)$. In (6b), $\sum_{\substack{t=1 \\ s.t. o_t^{k,l}=O_m}}^{T_l-1} \alpha_t^{k,l}(j) \beta_t^{k,l}(j)$ is the sum of the products of the forward and backward variables when the t th observed symbol $o_t^{k,l} = O_m$, i.e., $\alpha_t^{k,l}(i) = P(o_1^{k,l}, o_2^{k,l}, \dots, o_t^{k,l} = O_m, s_t = S_i | \theta)$. In the above-mentioned strategy, all the samples are treated equally. If weight D_l is assigned to the l th sample x_l^k , then (6) becomes

$$\bar{a}_{ij} = \frac{\sum_{l=1}^{R_k} \frac{D_l}{P_l} \sum_{t=1}^{T_l-1} \alpha_t^{k,l}(i) a_{ij} b_j(o_{t+1}^{k,l}) \beta_{t+1}^{k,l}(j)}{\sum_{l=1}^{R_k} \frac{D_l}{P_l} \sum_{t=1}^{T_l-1} \alpha_t^{k,l}(i) \beta_{t+1}^{k,l}(j)} \quad (7a)$$

$$\bar{b}_j(O_m) = \frac{\sum_{l=1}^{R_k} \frac{D_l}{P_l} \sum_{\substack{t=1 \\ s.t. o_t^{k,l}=O_m}}^{T_l-1} \alpha_t^{k,l}(j) \beta_t^{k,l}(j)}{\sum_{l=1}^{R_k} \frac{D_l}{P_l} \sum_{t=1}^{T_l-1} \alpha_t^{k,l}(j) \beta_t^{k,l}(j)} \quad (7b)$$

For the above equations, Arslan and Hansen [21] proved that weighting of the training samples does not violate the convergence property of the maximum likelihood training. A local maximum point of $P(X_k | \theta)$ will be attained after a sufficient number of training epochs. Since different samples are treated differently in estimating the parameters, (7) has the potential to be applied to HMM AdaBoosting. In this paper, the above-mentioned training strategy shall be referred to as the biased Baum–Welch estimation.

B. Cross-Validation for Error Estimation

Unlike the Neural Network or other classifiers, HMM gives a probabilistic measure rather than a definite Boolean result. The decision about the identity of the input is usually obtained by comparing the probabilities measured of all HMMs.

Assume that at a certain boosting round, Class d_l ($l = 1, 2, \dots, K$) has L_l subclassifiers (HMMs)— $\Theta_l = \{\theta_1^l, \theta_2^l, \dots, \theta_{L_l}^l\}$. For a given training sample $x_i^k \in X_k$, the probabilities of x_i^k , given all the HMMs, are computed and compared with one another. The HMM that gives the maximum likelihood is chosen as the identity of x_i^k

$$\theta^* = \arg \max_{\theta} P(x_i^k | \theta_j^l) \quad \forall l = 1, 2, \dots, K, \quad j = 1, 2, \dots, L_l. \quad (8)$$

The decision made in this way is a *one versus the rest* classification. If the correct model scores greater likelihood than the others, the result is correct; otherwise, an error occurs. As a result, the following hypothesis is made upon an HMM classifier in d_k , e.g., θ_p^k ($1 \leq p \leq L_k$):

$$h_p^k(x_i^k) = \begin{cases} 1, & \text{if } P(x_i^k | \theta_p^k) > P(x_i^k | \theta_q^j) \\ & \forall j = 1, 2, \dots, K; \quad j \neq k; \quad q = 1, 2, \dots, L_j \\ -1, & \text{otherwise.} \end{cases} \quad (9)$$

The training error of θ_p^k is estimated by summarizing the weighted hypotheses over all the training samples in X_k .

$$\varepsilon_p^k = \frac{D(x_i^k) E(h_p^k(x_i^k) = -1)}{R_k} = \sum_{\substack{i=1 \\ s.t. h_p^k(x_i^k)=-1}}^{R_k} \frac{D(x_i^k)}{R_k}. \quad (10)$$

It is shown in (9) that the error rate not only depends on the classifier itself but also relates to the other classifiers. The HMM obtained at boosting round t , e.g., θ_t^k , will, therefore, influence the error rate of all the HMMs trained at the previous boosting rounds, e.g., θ_τ^j ($j = 1, 2, \dots, K, \tau < t$). AdaBoosting requires that the composite classifiers have an error rate less than 0.5. As a result, not only the recently boosted HMM but also all the existing HMMs have to be validated. Clearly, the computations involved in calculating and comparing the probabilities are intensive. In our system, the following measures are taken to facilitate the processing.

Fig. 5. Data structure for implementing error estimation.

As illustrated in Fig. 5, each class, say, d_k , maintains a maximum probability array. The elements in the array are the R_k greatest probabilities of the training samples $X_k = \{x_1^k, x_2^k, \dots, x_{R_k}^k\}$ that are scored by the HMMs of Class d_j ($\forall j \neq k$). These maximum probabilities are denoted as $P_{\max}(x_1^k | \bar{\theta}), P_{\max}(x_2^k | \bar{\theta}), \dots, P_{\max}(x_{R_k}^k | \bar{\theta})$ in the figure, where $\bar{\theta}$ denotes any HMM of the class other than d_k .

At boosting round $L_k + 1$, after a new HMM of Class d_k is trained, say, $\theta_{L_k+1}^k$, the probabilities of all the training samples of Classes $d_1, d_2, \dots, d_K$, given $\theta_{L_k+1}^k$ are computed. For Class d_l as an example, if $P(x_i^l | \theta_{L_k+1}^k) > P_{\max}(x_i^l | \bar{\theta})$ ($i = 1, 2, \dots, R_l$), $P_{\max}(x_i^l | \bar{\theta})$ is replaced with $P(x_i^l | \theta_{L_k+1}^k)$. The following hypothesis, which is concluded from (9), is made for a training sample of Class d_k , say, x_i^k .

$$h_{L_k+1}^k(x_i^k) = \begin{cases} 1, & \text{if } P(x_i^k | \theta_{L_k+1}^k) > P_{\max}(x_i^k | \bar{\theta}) \\ i = 1, 2, \dots, R_k \\ -1, & \text{otherwise.} \end{cases} \quad (11)$$

The training error of $\theta_{L_k+1}^k$ is then computed using (10). In this way, the training error of any composite HMM, say, θ_t^l ($t = 1, 2, \dots, L_l$), can be easily obtained by comparing $P(x_i^l | \theta_t^l)$ ($i = 1, 2, \dots, R_l$) with the corresponding value in its maximum likelihood array— $P_{\max}(x_i^l | \bar{\theta})$.

If the error rates of $\theta_{L_k+1}^k$ and the existing HMMs are all less than 0.5, $\theta_{L_k+1}^k$ is retained as a qualified boosted classifier; otherwise, $\theta_{L_k+1}^k$ is discarded. The above-mentioned strategy provides an easily programmable approach for evaluating and computing the training error of the AdaBoost-HMM classifiers. Because the error rates are computed by comparing the probabilities scored by the individual HMMs, the above-mentioned procedure shall be referred to as cross-validation.

C. Steps of the HMM AdaBoosting Algorithm

The step-by-step procedures of HMM AdaBoosting for a K -class problem are given below.

- 1) For training set of Class d_k ($k = 1, 2, \dots, K$) with R_k samples $X_k = \{x_1^k, x_2^k, \dots, x_{R_k}^k\}$, initially, assign a uniform distribution $D_1(x_j^k) = 1/R_k$ ($\forall k = 1, 2, \dots, K, j = 1, 2, \dots, R_k$) to $x_1^k, x_2^k, \dots, x_{R_k}^k$. The boosting token k is initialized to be equal to 1.
- 2) Train a new HMM for the k th class θ_t^k using the biased Baum–Welch algorithm with the distribution $D_t(x_j^k)$.

- 3) Formulate the binary hypothesis $h_t^k(x_j^k) \rightarrow \{-1, +1\}$ for θ_t^k using (9). The error rate of $\theta_t^k - \varepsilon_t^k$, and the error rates of all the existing HMMs of other classes θ_t^j ($j = 1, 2, \dots, K, j \neq k, l = 1, 2, \dots, R_j$) are estimated and verified using the cross-validation and (10). If the new model is valid, go on to Step 4; otherwise, the boosting token is passed to the next class $k = \begin{cases} k+1, & \text{if } k < K \\ 1, & \text{if } k = K \end{cases}$ and then Step 2 is repeated.
- 4) Calculate $w_t^k = (1/2) \ln((1 - \varepsilon_t^k)/\varepsilon_t^k)$ for θ_t^k , where ε_t^k is the error rate. If the error rate of some existing HMM, say, θ_t^j , changes, the corresponding $w_t^j = (1/2) \ln((1 - \varepsilon_t^j)/\varepsilon_t^j)$ is also recomputed.
- 5) Update the distribution: $D_{t+1}(x_j^k) = ((D_t(x_j^k))/Z_t) e^{-w_t^k h_t^k(x_j^k)}$ ($h_t^k(x_j^k) = 1$ or -1), where Z_t is the normalization factor.

The procedure terminates when boosting for all the classes are completed. Assume that Θ_k is the AdaBoost-HMM classifier for Class d_k ($k = 1, 2, \dots, K$) that has been boosted for L_k rounds. Then Θ_k consists of L_k HMMs— $\theta_1^k, \theta_2^k, \dots, \theta_{L_k}^k$ and L_k weights— $w_1^k, w_2^k, \dots, w_{L_k}^k$. The normalized log likelihood of an observed sequence x^T given Θ_k is defined as follows:

$$\bar{P}(x^T | \Theta_k) = \frac{1}{L_k} \sum_{t=1}^{L_k} \log [w_t^k P(x^T | \theta_t^k)]. \quad (12)$$

The final decision is made by comparing the $\bar{P}(x^T | \Theta_k)$ for all the classes ($k = 1, 2, \dots, K$). The one that gives the maximum probability is chosen as the identity of x^T , as follows:

$$ID(x^T) = \arg \max_k \bar{P}(x^T | \Theta_k), \quad (1 \leq k \leq K). \quad (13)$$

D. Conceptual Interpretation of HMM AdaBoosting

The improvement of the performance of the AdaBoost-HMM classifiers can be interpreted conceptually as follows. Given enough training samples (the number of samples does not have to be very large but they should cover the entire sample space), the boosted HMMs can be trained to cover the widely spread samples. Assume that $X_k = \{x_1^k, x_2^k, \dots, x_{R_k}^k\}$ are training samples of Class d_k . In the first round of boosting, the samples are treated equally ($D_1(x_j^k) = (1/R_k) \forall k = 1, 2, \dots, K, j = 1, 2, \dots, R_k$) and the HMM obtained θ_1^k is a normal ML model. Most of the samples, say, $X_{k,\text{ideal}} = \{x_1^{k,\text{ideal}}, x_2^{k,\text{ideal}}, \dots, x_{R_{k,\text{ideal}}}^{k,\text{ideal}}\}$, where $X_{k,\text{ideal}} \subset X_k$ and $R_{k,\text{ideal}}$ is the number of the samples in $X_{k,\text{ideal}}$, have greater likelihood values such that $P(x_i^{k,\text{ideal}} | \theta_1^k) > P(x_i^{k,\text{ideal}} | \theta_1^j)$ ($\forall j = 1, 2, \dots, K, j \neq k, i = 1, 2, \dots, R_{k,\text{ideal}}$). These samples are referred to as the ideal samples because the normal ML model is able to identify them correctly. However, the likelihood values of some other samples $X_{k,\text{hard}} = \{x_1^{k,\text{hard}}, x_2^{k,\text{hard}}, \dots, x_{R_{k,\text{hard}}}^{k,\text{hard}}\}$, where $X_{k,\text{hard}} \subset X_k$ and $R_{k,\text{hard}}$ is the number of the samples in $X_{k,\text{hard}}$, determined by $\theta_1^k - P(x_i^{k,\text{hard}} | \theta_1^k)$ ($i = 1, 2, \dots, R_{k,\text{hard}}$) may be smaller than that determined by incorrect HMMs. These samples are called hard samples

Fig. 6. Conceptual illustration: the decision boundaries formed by a single HMM (left) and an AdaBoost-HMM (right).

because the ML model is not able to identify them correctly. As boosting continues, new HMMs $\theta_2^k, \theta_3^k, \dots, \theta_{L_k}^k$ tend to bias toward these hard samples. For example, if θ_t^k is obtained at Round t , the scored likelihood of a certain group of hard samples $P(x_i^{k, \text{hard}} | \theta_t^k)$ increases as t increases. The cost is that the likelihood of some ideal samples will normally decrease. By synthesizing the likelihood according to (12), the decision boundaries formed will properly cover both the ideal samples and hard samples. Fig. 6 gives conceptual illustrations of the difference between a single HMM classifier and an AdaBoost-HMM classifier. A single HMM may identify the ideal samples with good accuracy but may be unable to take into consideration the hard or outlier samples as well. After boosting, a more complex decision boundary is formed. The hard samples can then be classified with good credibility.

The above boosting strategy strictly observes the principles of AdaBoosting. The convergence of the training error is, thus, guaranteed. The decrease in error rate after each round of boosting for HMM classifier may not be as fast as that of other types of classifiers such as Neural Network. The reason is that the error rate of an HMM is obtained by comparing the likelihood value determined by HMM in the current round of boosting and those in the previous rounds, as explained in Section V-B. With the propagation of HMMs, the classification error of the composite HMM will normally increase and hence the error of the AdaBoost-HMMs decreases at a low speed. Our experiments also reveal that the error surface with respect to the boosting round is generally smooth but sometimes may have bumps on it. The error rate, however, demonstrates the tendency to decrease as boosting continues.

For different classes, the number of rounds the boosted HMM can pass cross-validation are different and hence different classes may have different number of HMMs. However, the unequal number of HMMs for different classes does not affect the final decision because $\bar{P}(x^T | \Theta_k)$ in (12) is normalized with the number of HMMs in a class.

Another aspect of the proposed strategy that should be stressed is that the error rate computed by (10) only accounts for part of the classification error. It indicates the samples in X_k being misclassified into some wrong category d_j ($j \neq k$). This kind of error is normally referred to as false rejection rate (FRR) or Type II error. Another source of the error indicates the samples of other classes X_j ($j \neq k$) are erroneously accepted

by Θ_k . This portion of error is referred to as false acceptance rate (FAR) or Type I error. However, FAR is not used in the proposed HMM AdaBoosting algorithm because it cannot work with the biased Baum–Welch estimation. As illustrated in (7), the biased Baum–Welch algorithm only uses the correct training samples for parameter estimation. An erroneously accepted sample cannot be applied to train the parameters of the HMMs. FRR can weight the correct training samples in (7) as it is an indicator of the goodness-of-fit of the correct samples, while FAR, which is a statistical measure for the incorrect samples (irrelevant to the correct samples), cannot be applied for the proposed method. This is why FAR is not computed in the boosting steps given in Section V-C. If other base training algorithm, such as MMI estimation, is applied in HMM boosting, where the erroneously accepted samples are also used to train the parameters of the HMM, both FRR and FAR may be considered.

VI. PERFORMANCE OF THE ADABOOST-HMM CLASSIFIER

The AdaBoosted HMM is applied to identify the basic visual speech elements—visemes. Based on MPEG-4 Multimedia standards, the visemes are grouped into 14 categories as shown in Table I. Each viseme corresponds to several phonemes and phoneme-like productions [22], [23].

Since the computational load involved in boosting HMM using the Baum–Welch algorithm is proportional to the number of input frames, the sampling rate of the lip movement should be kept low. It is found that a sampling rate of 50 frames per second is sufficient to capture the continuous movement of the lips.

As mentioned in Section I, there may be significant variation in the samples of a given viseme taken from different contexts. The prior and posterior visemes will both influence the temporal features of the target viseme to a certain extent. For example, the visual representations of the phoneme /ai/ are very different in the words *hide* [haid] and *right* [rait]. Such polymorphism of the viseme indicates that its samples may have a spread-out distribution. Traditional single-HMM classifier is not capable of identifying the samples with spread-out distribution, while AdaBoost-HMM classifier, which has better clustering ability via incorporating multiple HMMs, holds the promise of identifying visemes under various contexts.

TABLE I
VISEMES DEFINED IN MPEG-4 MULTIMEDIA STANDARDS

Viseme Number	Phonemes	Examples
0	none	(silence and relax)
1	p, b, m	push, bike, milk
2	f, v	find, voice
3	T, D	think, that
4	t, d	teach, dog
5	k, g	call, guess
6	tS, dZ, S	check, join, shrine
7	s, z	set, zeal
8	n, l	note, lose
9	r	read
10	A:	jar
11	e	bed
12	I	tip
13	Q	shock
14	U	good

A. Image Processing and Lip Tracking for Visual Speech Analysis

For the experiments, the raw data are video frames that describe the movement of the lips during articulation. The lip images are sampled at 50 frames per second. Typical duration of visemes ranges from 0.3 to 1 s. The following steps are taken to convert the lip shapes into tractable feature vectors.

Human lips are of darker color than the color of the surrounding skin. There are distinguishable borders between the lips and the surrounding skin. A proper threshold can, therefore, be set for segmentation based on, say, the red, green, blue (RGB) parameters or the hue, saturation, value (HSV) parameters of the image. Hue-saturation parameters are used in our system because they are relatively insensitive to changes in the absolute brightness. The RGB-to-HSV conversion algorithm and the lip region detection strategy described in [30] and [31] are adopted in our experiments. First, the red-dominant area of the image is detected by means of red exclusion [31]. The approximate lip region is located as illustrated in Fig. 7(b). Next, the raw image is transformed to a modified hue image with RGB-to-HSV transformation proposed in [30]. As illustrated in Fig. 8, the hue factors of the lip region and the remaining lip-excluded image account for different portions of the histogram. A proper threshold is extracted from the hue histogram of the entire image, which usually corresponds to a local minimum point (valley) in the histogram as shown in the figure. The hue image is then compared with the chosen threshold and a binary image is obtained. Finally, logical AND operation of the binary image and the approximate lip region is performed. The lip region is then segmented as shown in Fig. 7(c).

Further processing is based on the deformable templates algorithm [24]–[26]. This approach maintains a geometric template with dynamic contours to fit an elastic object. The template is controlled by some parameters and is drawn near to the boundaries of the target in the tracking process. The contours of the lips are simple compared with other moving objects, such as human body or sign language. The requirement on the selection of the dynamic contours that build the template is, thus, not stringent. It is commonly agreed that many polynomial curves, such as parabolas or elliptical curves can well match the boundaries of the lip [13], [24]. In our experiments, the parameter-

ized template is built up with eight Bezier curves as illustrated in Fig. 7(d). Such a template is easily programmable because Bezier curve is provided in many programming languages, such as C++ or Java. The results of our lip-tracking experiments also show that Bezier curves are able to characterize the shape of the lip with good accuracy [28], [60]. The process of fitting the template to the lip image is to adjust the parameters (usually the control points) of the template to minimize certain energy function. As illustrated in Fig. 7(d), the points in the small circles are control points of the template. The energy function takes four components, defined as follows:

$$E_{\text{lip}} = -\frac{1}{R_1} \int_{R_1} H(z) dz \quad (14a)$$

$$E_{\text{edge}} = -\frac{1}{C_1 + C_2} \times \int_{C_1 + C_2} |H^+(z) - H(z)| + |H^-(z) - H(z)| dz \quad (14b)$$

$$E_{\text{hole}} = -\frac{1}{R_2} \int_{R_2} H(z) dz \quad (14c)$$

$$E_{\text{inertia}} = \|T_{t+1} - T_t\|^2 \quad (14d)$$

where R_1 , R_2 , C_1 , and C_2 are areas and contours as illustrated in Fig. 7(d). $H(z)$ is a function of the hue of the given pixel z , $H^+(z)$ is the hue function of the closest right-hand-side pixel, and $H^-(z)$ is that of the closest left-hand-side pixel. T_{t+1} and T_t are the matched templates at time $t+1$ and t . $\|T_{t+1} - T_t\|$ indicates the Euclidean distance between the two templates. The overall energy of the template E is the linear combination of the components, as defined as follows:

$$E = c_1 E_{\text{lip}} + c_2 E_{\text{edge}} + c_3 E_{\text{hole}} + c_4 E_{\text{inertia}}. \quad (15)$$

While tracking the boundaries of the lips from the image sequence, the parameters of the template are updated to minimize E in a number of epochs. A detailed discussion on the deformable template method is presented in [24]. The matched template and the actual lip boundary are depicted in Fig. 7(d) and (e). It can be seen that the matched template is symmetric and smooth, and is, therefore, easy to process.

From the matched lip template, ten geometric parameters are extracted as shown in Fig. 7(f). These features indicate the thickness of various parts of the lips, the positions of some key points and the curvatures of the lips. They are chosen as they uniquely determine the shape of the lips and best characterize the movement of the lips. The geometric features so extracted undergo principal component analysis (PCA) to reduce the dimensionality from ten to seven dimensions. The collected feature vectors are finally clustered into groups using K -means algorithm. For the experiments carried out in this paper, 128 clusters (code words) are used in the vector database (code book). They are taken to encode the input observation sequences and build the symbol set $\{O_1, O_2, \dots, O_{128}\}$ of the discrete HMMs.

B. Viseme Classifier

Our study on lip dynamics shows that the motion of the lips can be partitioned into three phases during viseme production.

Fig. 7. (a) Original image. (b) Lip localization. (c) Segmented lip area. (d) Parameterized lip template. (e) Actual lip shape. (f) Geometric measures extracted out of the lip template. 1: Thickness of the upper bow. 2: Thickness of the lower bow. 3: Thickness of the lip corner. 4: Position of the lip corner. 5: Position of the upper lip. 6: Position of the lower bow. 7: Curvature of the upper-exterior boundary. 8: Curvature of the lower-exterior boundary. 9: Curvature of the upper-interior boundary. 10: Curvature of the lower-interior boundary. [The curvatures are defined as the height of the control points (the middle points of the curves) relative to the starting points of the Bezier curves.]

Fig. 8. Separation of the lip region out of the entire image using hue distribution. (a) Histogram of the hue component of the entire image. (b) Histogram of the hue component of the lip region. (c) Histogram of the hue component of the lip-excluded image.

The first is the initial phase, starting from a closed mouth or the end of the last viseme production to the beginning of sound production of the viseme investigated. During this interval, the sound of the target viseme is not fully articulated and the lips are characterized with sharp changes. The next phase is the articulation phase, which is the time when the sound is produced. The third phase is the end phase when the mouth restores to the relaxed state or transits to the production of the next viseme.

Fig. 9 illustrates the three phases and the corresponding acoustic signal when the phonetic sound /u/ is uttered.

Among these phases, the articulation phase is the most important for recognition because this is where the major difference among visemes lies and it is relatively independent to the context. The initial phase and end phase are transitional phases. Based on the fact that there are three phases in viseme production, three-state left-right HMM structure as shown in Fig. 10 is adopted in the proposed system.

Using this structure, the state transition matrix A has the form

$$A = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 \\ 0 & a_{22} & a_{23} & 0 \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where the fourth state is a null state that indicates the end of the viseme production. The initial values of the coefficients in matrices A and B are set according to the statistics of the three phases. Given a viseme sample, the approximate initial phase, articulation phase, and end phase are manually segmented from the image sequence and the acoustic signal (an example is given in Fig. 9), and the number of frames of each phase is counted. Based on the number of the frames for the phases, a set of forward probabilities (the probabilities of one phase transiting to the next phase) and iteration probabilities (the probabilities of the phase repeating) are computed [28]. These probabilities are then adopted as the initial values of $a_{i,i}$ and $a_{i,i+1}$. For example, if the duration of state S_i is T_i , the initial value of $a_{i,i}$ is $T_i/(T_i + 1)$, and the initial value of $a_{i,i+1}$ is $1/(T_i + 1)$ as they maximize $a_{i,i}^{T_i} a_{i,i+1}$. Matrix B is initialized in a similar manner. If symbol O_j appears $T(O_j)$ times in state S_i , the initial value of b_{ij} is $T(O_j)/T_i$. For such an arrangement, the states of the HMM are physically associated with the actual phases of viseme

Fig. 9. Three phases of viseme production (upper) and the waveform of the sound produced (lower). (a) Initial phase. (b) Articulation phase. (c) End phase.

Fig. 10. Three-state left-right viseme model.

production and hence are referred to as the initial state, articulation state and end state. The HMMs configured in this way are adopted as the HMMs for the subsequent adaptive boosting.

C. Experimental Results

Experiments are carried out to assess the performance of the AdaBoost-HMM in recognition of visemes. Comparison is also made with the performance of the single HMM classifiers.

In the first experiment, the speaker is asked to produce one sound/viseme in Table I at a time. Video recording is made and vector sequences corresponding to the visual speech elements are extracted as described in Section VI-A. The visemes obtained in this way are referred to as isolated visemes because they are independently produced. They start from a closed mouth and end with a closed mouth.

The speaker is asked to repeat each isolated viseme 140 times. The 140 vector sequences obtained are divided into two groups: 40 of them are used as the training samples and the other 100 are used as the testing samples. For each isolated viseme, an AdaBoost-HMM classifier that consists of 15–20 HMMs is trained with the proposed strategy given in Section V-C. According to (13), for a testing sample x^T of Class d_k , a correct classification is made if $ID(x^T) = d_k$. The classification error (FRR) of the AdaBoost-HMM classifier Θ_k is then computed using

$$FRR(\Theta_k) = 1 - \frac{\text{number of correctly classified samples of } d_k}{\text{number of all the testing samples of } d_k}. \quad (16)$$

The classification errors of the AdaBoosted viseme classifiers are listed in Table II. A single-HMM classifier is also trained for each isolated viseme using the same 40 training samples. The single-HMM classifier is an ML HMM as the Baum–Welch estimation is adopted in training. The classification errors of the single-HMM classifiers are computed using (16) and are also listed in Table II for comparison.

TABLE II
CLASSIFICATION ERRORS (FRR) OF THE SINGLE-HMM CLASSIFIER AND THE ADABOOST-HMM CLASSIFIER IN RECOGNITION OF ISOLATED VISEMES

Viseme Categories	Classification Error (FRR)	
	Single-HMM Classifier	AdaBoost-HMM Classifier
1 p, b, m	13%	8%
2 f, v	4%	5%
3 T, D	11%	4%
4 t, d	35%	17%
5 k, g	24%	9%
6 tS, dZ, S	10%	10%
7 s, z	4%	4%
8 n, l	19%	20%
9 r	18%	7%
10 A:	1%	4%
11 e	8%	11%
12 I	1%	4%
13 Q	7%	10%
14 U	7%	9%
Average	11.6%	8.7%

From Table II, it can be seen that both the single-HMM classifiers and AdaBoost-HMM classifiers can identify the isolated visemes with reasonable accuracy. An average classification error below 20% is obtained with either approach. The classification error is lower for the vowels as the movement of the lips is more. The performance of the AdaBoost-HMM classifiers and that of the single-HMM classifiers are not significantly different. The samples obtained for the isolated visemes demonstrate good homogeneity as they are independently produced.

In the second experiment, the speaker is asked to produce different words containing the sound/visemes. The target viseme are then isolated from the video sequence. For example, the samples of the viseme of /t,d/ are extracted from words such as *teach*, *bundle*, and *lad*. The visemes obtained in this way are referred to as context-dependent visemes. For each viseme, 200 samples are drawn, with 100 for training and the other 100 for testing. The corresponding vector sequences are then determined from the images and a series of boosted HMMs are trained for each viseme with the proposed strategy.

As mentioned in Section V-B, the training error of an AdaBoost-HMM classifier [which is the error obtained from (10)] shows tendency of decreasing as boosting continues. To illustrate such change, the training errors of four AdaBoost-HMM classifiers and the t th HMM (the HMM trained

Fig. 11. Rate of training error versus boosting round. Viseme classifiers of (a) $/e/$. (b) $/s, z/$. (c) $/T, D/$. (d) $/t, d/$.

in the t th boosting round) of the classifiers are presented in Fig. 11. Taking Fig. 11(a) as an example, the training error of the AdaBoost-HMM classifier decreases from 0.27 to 0.13 for the 20 rounds while the training error of the t th HMM increases from 0.27 to 0.45. This is so as the composite HMM biases more and more to the outlier samples as boosting continues.

For the experiments carried out, the boosted subclassifier for each viseme consists of 12 to 20 HMMs. The training errors and classification errors (FRR) of the testing samples are listed in Table III together with the corresponding errors using the single-HMM classifiers. It is observed that the accuracy of recognition is significantly improved (smaller error rate) using AdaBoost-HMM classifier.

Compared with the classification errors listed in Table II, the classification accuracy of the single-HMM classifiers decreases dramatically in identifying context-dependent visemes. The classification rates of some consonants are even less than 50%. The reason underlying the high identification error is the distribution of the samples. As the samples of a context-dependent viseme are extracted from various words (contexts), the “shapes” of the samples of even the same viseme are different. In statistics jargon, the samples of a context-dependent viseme demonstrate a spread-out distribution. The traditional single-HMM classifiers cannot cover such a distribution well. However, the classification accuracy can be greatly improved with the application of AdaBoost-HMM classifiers. As illustrated in Table III, although the classification errors are larger compared with those listed in Table II, an average recognition accuracy of 70%–80% is still attainable, which is about 16% better than the single-HMM classifiers. The improvement can be attributed to the fact that a more complex decision boundary is formulated using the AdaBoost-HMM classifier than using the single-HMM classifier. Therefore, the AdaBoost-HMM classifiers can better cover the spread-out distribution for both the testing samples and the training samples. This is validated by

TABLE III
TRAINING ERRORS AND CLASSIFICATION ERRORS (FRR) OF THE SINGLE-HMM CLASSIFIERS AND THE ADABOOST-HMM CLASSIFIERS

Viseme Categories	Single-HMM Classifier		AdaBoost-HMM Classifier	
	Training Error	Classification Error	Training Error	Classification Error
1 p, b, m	14%	20%	6%	15%
2 f, v	15%	27%	11%	25%
3 T, D	24%	34%	10%	18%
4 t, d	39%	50%	17%	19%
5 k, g	17%	17%	16%	16%
6 tS, dZ, S	17%	21%	5%	9%
7 s, z	39%	45%	13%	17%
8 n, l	40%	79%	22%	33%
9 r	21%	54%	22%	37%
10 A:	14%	18%	5%	5%
11 e	27%	33%	13%	7%
12 I	9%	10%	0%	2%
13 Q	10%	35%	4%	11%
14 U	4%	9%	1%	7%
Average	21%	32%	10%	16%

the experimental results. If the context-dependent visemes are looked as isolated visemes distorted by adjoining visemes, it is concluded that the AdaBoost-HMM classifiers provide better robustness on identifying visemes than single-HMM classifiers.

D. Computational Load

The computations involved in AdaBoosting-HMM are estimated as follows. For the three-state 128-symbol discrete HMM applied in the experiments, with 100 training samples ranging from 15 to 50 frames for each viseme, about 10^6 computations (computations include multiply and division) are required to train an HMM with the biased Baum–Welch estimation. Most of the computations are for calculating the forward variables and backward variables. For example, if the length of a training sample is 30 frames (average length), about 1200 multiplies are required to build a probability trellis to compute the forward variables and backward variables [18]. Assume that all the 100 training samples have the same length of 30 frames, about 1.2×10^5 computations are required to compute these variables. The forward variables and backward variables have to be computed more than once because EM iterations are taken in the Baum–Welch estimation. If 10 iterations are used (in our experiments, the probability scored for the training samples becomes stable after about 10 iterations), 1.2×10^6 computations are required. The number of computations involved in estimating the state transition coefficients and symbol emission probabilities is small compared with this quantity. As a result, it is reasonable to conclude that computations of the order of 10^6 are required to train a single-HMM classifier. The computations can be completed in less than 10 s using 900-MHz Pentium III.

For AdaBoost-HMM classifier comprising 20 HMMs, approximately 2×10^7 computations are required for training the classifier. The additional number of computations including error estimation using cross-validation and calculation of weights is small compared with the number of computations required for the Baum–Welch estimation. The total number of computations is, thus, approximately 2×10^7 . This is a modest amount of computations for the modern computers or signal processor chips. The average time for training the

AdaBoost-HMM classifier is about 2 min using 900-MHz Pentium III.

The computational load in the recognition phase is far less than that in the training phase. For the single-HMM classifiers, the likelihood of an input sequence is computed using the forward process. If the input sequence is 30 frames in length, about 1200 multiplies are performed. As the identity of the input sequence is determined out of 14 viseme categories, the total number of computations will be approximately 1.7×10^4 . It only takes 1 s using 900-MHz Pentium III. For the AdaBoost-HMM classifiers, the input sequence is evaluated by a total of 280 (20×14) HMMs, where about 3.4×10^5 multiplies are carried out. The number of computations involved in probability synthesis is small compared with this quantity. It takes 4–5 s to determine the identity of the input sequence using the 900-MHz Pentium III computer.

VII. CONCLUSION

The adaptively boosted HMM classifier proposed in this paper is a type of multi-HMM classifiers. It is specially suited to improve the accuracy of identifying time series with spread-out distribution. The AdaBoosting technique serves to train a group of undemanding HMMs, in which the training error is less than 0.5. The decisions made by the composite HMMs are synthesized and a more complex decision boundary is formulated than using single HMM.

In this paper, the AdaBoost-HMM classifier is applied to identify visemes in visual speech processing. Experimental results show that such a classifier is able to achieve an average improvement of 16% on recognition accuracy over the traditional single HMM classifier in identifying context-dependent visemes. The complexity of computational load to train the boosted viseme classifier is approximately twenty times that of the load of training a single-HMM viseme classifier when the Baum–Welch estimation is used.

The proposed method can readily be extended to many other situations especially when the observed data have spread-out distribution, for example, speech recognition, handwriting recognition, and speaker identification.

REFERENCES

- [1] H. McGurk and J. MacDonals, "Hearing lips and seeing voices," *Nature*, vol. 264, pp. 746–748, Dec. 1976.
- [2] E. D. Petajan, "Automatic lipreading to enhance speech recognition," Ph.D. dissertation, Univ. Illinois, Urbana-Champaign, 1984.
- [3] S. Morishima, S. Ogata, K. Murai, and S. Nakamura, "Audio-visual speech translation with automatic lip synchronization and face tracking based on 3D head model," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 2, May 2002, pp. 2117–2120.
- [4] A. Adjoudani and C. Benoit, "On the integration of auditory and visual parameters in an HMM-based ASR," in *Speechreading by Humans and Machines*, ser. NATO ASI Series, D. G. Stork and M. E. Hennecke, Eds., 1996, pp. 461–472.
- [5] P. L. Silsbee and A. C. Bovik, "Computer lipreading for improved accuracy in automatic speech recognition," *IEEE Trans. Speech Audio Processing*, vol. 4, pp. 337–351, Sept. 1996.
- [6] M. Tomlinson, M. Russell, and N. Brooke, "Integrating audio and visual information to provide highly robust speech recognition," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 2, 1996, pp. 821–824.
- [7] T. Chen and R. R. Rao, "Audio-visual integration in multimodal communication," *Proc. IEEE*, vol. 86, pp. 837–852, May 1998.
- [8] W. J. Welsh, A. D. Simon, R. A. Hutchinson, and S. Searby, "A speech-driven 'talking-head' in real time," in *Proc. Picture Coding Symp.*, Cambridge, MA, 1990, pp. 7.6–1–7.6–2.
- [9] K. E. Finn and A. A. Montgomery, "Automatic optically-based recognition of speech," *Pattern Recognit. Lett.*, vol. 8, no. 3, pp. 159–164, 1988.
- [10] J. Luetttin, N. A. Thacker, and S. W. Beet, "Speechreading using shape and intensity information," in *Int. Conf. Spoken Language Processing*, 1996, pp. 58–61.
- [11] J. K. Baker, "The Dragon system—An overview," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-23, pp. 24–29, Feb. 1975.
- [12] I. Matthews, T. F. Cootes, J. A. Bangham, S. Cox, and R. Harvey, "Extraction of visual features for lipreading," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, pp. 198–213, Feb. 2002.
- [13] T. Chen, "Audiovisual speech processing," *IEEE Signal Processing Mag.*, vol. 18, pp. 9–21, Jan. 2001.
- [14] D. G. Stork and M. E. Hennecke, "Speechreading: An overview of image processing, feature extraction, sensory integration and pattern recognition techniques," in *2nd Int. Conf. Automatic Face Gesture Recognition*, Oct. 1996, pp. XVI–XXVI.
- [15] A. J. Goldschen, O. N. Garcia, and E. D. Petajan, "Continuous automatic speech recognition by lipreading," in *Computational Imaging and Vision*, 1997, ch. 14, pp. 321–343.
- [16] E. Owens and B. Blazek, "Visemes observed by hearing impaired and normal hearing adult viewers," *J. Speech Hear. Res.*, vol. 28, pp. 381–393, 1985.
- [17] L. R. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*, ser. Signal Processing Series. Englewood Cliffs, NJ: Prentice-Hall, 1993.
- [18] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, pp. 257–286, Feb. 1989.
- [19] R. E. Schapire, "A brief introduction to boosting," in *16th Int. Joint Conf. Artificial Intelligence*, 1999, pp. 1401–1405.
- [20] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, Aug. 1997.
- [21] L. M. Arslan and J. H. L. Hansen, "Selective training in hidden Markov model recognition," *IEEE Trans. Speech Audio Processing*, vol. 7, pp. 46–54, Jan. 1999.
- [22] A. M. Tekalp and J. Ostermann, "Face and 2-D mesh animation in MPEG-4," *Signal Processing: Image Communication—Special Issue on MPEG-4*, vol. 15, pp. 387–421, Jan. 2000.
- [23] Overview of the MPEG-4 Version 1 Standard [Online]. Available: <http://www.am.hhi.de/mpeg-video/standards/mpeg-4.htm>
- [24] A. Yuille and P. Hallinan, "Deformable templates," in *Active Vision*, A. Blake and A. Yuille, Eds. Cambridge, MA: MIT Press, 1992, pp. 21–38.
- [25] T. Coianiz, L. Torresani, and B. Caprile, "2D deformable models for visual speech analysis," in *Speech Reading by Humans and Machines*, ser. NATO ASI Series, D. G. Stork and M. E. Hennecke, Eds., 1996, pp. 391–398.
- [26] M. E. Hennecke, K. V. Prasad, and D. G. Stork, "Using deformable templates to infer visual speech dynamics," in *28th Asilomar Conf. Signals, Systems and Computers*, vol. 1, 1994, pp. 578–582.
- [27] L. R. Bahl, P. F. Brown, P. V. de Souza, and R. L. Mercer, "Maximum mutual information estimation of hidden Markov model parameters for speech recognition," in *IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 1, Apr. 1986, pp. 49–52.
- [28] S. W. Foo and L. Dong, "Recognition of visual speech elements using hidden Markov models," in *3rd IEEE Pacific Rim Conf. Multimedia*, vol. LNCS 2532, Dec. 2002, pp. 607–614.
- [29] S. W. Foo and E. G. Lim, "Speaker recognition using adaptively boosted classifiers," *IEICE Trans. Inform. Syst.—Special Issue on Speech Information Processing*, vol. E86-D, no. 3, pp. 474–482, Mar. 2003.
- [30] X. Z. Zhang, C. Broun, R. M. Mersereau, and M. A. Clements, "Automatic speechreading with applications to human-computer interfaces," *EURASIP J. Appl. Signal Processing—Special Issue on Joint Audio-Visual Speech Processing*, pp. 1228–1247, 2002.
- [31] T. W. Lewis and D. M. W. Powers, "Lip feature extraction using red exclusion," in *Pan-Sydney Workshop Visualization, Selected Papers*, vol. 2, 2000, pp. 61–67.
- [32] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," in *11th Annu. Conf. Computational Learning Theory*, 1998, pp. 80–91.
- [33] R. E. Schapire, "Theoretical views of boosting and applications," in *10th Int. Conf. Algorithmic Learning Theory*, 1999, pp. 13–25.

- [34] C. Binnie, A. Montgomery, and P. Jackson, "Auditory and visual contributions to the perception of consonants," *J. Speech Hear. Res.*, vol. 17, pp. 619–630, 1974.
- [35] W. H. Sumby and I. Pollack, "Visual contributions to speech intelligibility in noise," *J. Acoust. Soc. Amer.*, vol. 26, pp. 212–215, 1954.
- [36] B. Dodd and R. Campbell, *Hearing by Eye: The Psychology of Lip-Reading*. London, U.K.: Lawrence Erlbaum, 1987.
- [37] R. Campbell, "Seeing brains reading speech: A review and speculations," in *Speech Reading by Humans and Machines*, ser. NATO ASI Series, D. G. Stork and M. E. Hennecke, Eds., 1996, pp. 115–133.
- [38] D. Reisberg, J. McLean, and A. Goldfield, "Easy to hear, but hard to understand: A lip-reading advantage with intact auditory stimuli," in *Hearing by Eye*, B. Dodd and R. Campbell, Eds. London, U.K.: Lawrence Erlbaum, 1987, pp. 97–113.
- [39] A. J. Goldschen, "Continuous automatic speech recognition by lipreading," Ph.D. dissertation, George Washington Univ., Washington, DC, 1993.
- [40] B. P. Yuhas, M. H. Goldstein, and T. J. Sejnowski, "Integration of acoustic and visual speech signals using neural networks," *IEEE Commun. Mag.*, vol. 27, pp. 65–71, Nov. 1989.
- [41] D. G. Stork and H. L. Lu, "Speechreading by Boltzmann zippers," in *Proc. Workshop Machines That Learn*, Snowbird, UT, 1996.
- [42] A. J. Goldschen, O. N. Garcia, and E. Petajan, "Continuous optical automatic speech recognition by lipreading," in *28th Asilomar Conf. Signals, Systems, and Computers*, vol. 1, 1994, pp. 572–577.
- [43] P. L. Silsbee and A. C. Bovik, "Visual lipreading by computer to improve automatic speech recognition accuracy," Comput. Vision Res. Ctr., Univ. Texas, Austin, TX, TR-93-02-90, 1993.
- [44] —, "Medium vocabulary audiovisual speech recognition," in *New Advances and Trends in Speech Recognition and Coding*, 1993, pp. 13–16.
- [45] X. Zhang, R. M. Mersereau, and M. A. Clements, "Audio-visual speech recognition by speechreading," in *14th Int. Conf. Digital Signal Processing*, vol. 2, 2002, pp. 1069–1072.
- [46] G. Gravier, G. Potamianos, and C. Neti, "Asynchrony modeling for audio-visual speech recognition," presented at the *Int. Conf. Human Language Technology*, 2002.
- [47] A. V. Nefian, L. Liang, X. Pi, X. Liu, and K. Murphy, "Dynamic Bayesian networks for audio-visual speech recognition," *EURASIP J. Appl. Signal Processing*, vol. 2002, no. 11, pp. 1274–1288, 2002.
- [48] S. Dupont and J. Luetttin, "Audio-visual speech modeling for continuous speech recognition," *IEEE Trans. Multimedia*, vol. 2, pp. 141–151, Sept. 2000.
- [49] C. Binnie, A. Montgomery, and P. Jackson, "Auditory and visual contributions to the perception of consonants," *J. Speech Hear. Res.*, vol. 17, pp. 619–630, 1974.
- [50] G. W. Greenwood, "Training partially recurrent neural networks using evolutionary strategies," *IEEE Trans. Speech Audio Processing*, vol. 5, pp. 192–194, Mar. 1997.
- [51] J. J. Williams and A. K. Katsaggelos, "An HMM-based speech-to-video synthesizer," *IEEE Trans. Neural Networks*, vol. 13, pp. 900–915, July 2002.
- [52] S. Morishima, S. Ogata, K. Murai, and S. Nakamura, "Audio-visual speech translation with automatic lip synchronization and face tracking based on 3-D head model," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 2, May 2002, pp. 2117–2120.
- [53] C. C. Chibelushi, S. Gandon, J. S. D. Mason, F. Deravi, and R. D. Johnston, "Design issues for a digital audio-visual integrated database," *IEE Colloq. Integrated Audio-Visual Processing*, no. 1996/213, pp. 7/1–7/7, Nov. 1996.
- [54] G. Potamianos, F. Cosatto, H. P. Graf, and D. B. Roe, "Speaker independent audio-visual database for bimodal ASR," in *Proc. ESCA Workshop Audio-Visual Speech Processing*, Sept. 1997, pp. 65–68.
- [55] D. G. Stork, G. Wolff, and E. Levine, "Neural network lipreading system for improved speech recognition," in *Proc. Int. Joint Conf. Neural Network*, 1992, pp. 285–295.
- [56] C. Bregler and S. Omohundro, "Nonlinear manifold learning for visual speech recognition," in *Proc. IEEE Int. Conf. Computer Vision*, 1995, pp. 494–499.
- [57] P. Silsbee and A. Bovik, "Computer lipreading for improved accuracy in automatic speech recognition," *IEEE Trans. Speech and Audio Processing*, vol. 4, pp. 337–351, Sept. 1996.
- [58] A. Adjoudani and C. Benoit, "On the integration of auditory and visual parameters in an HMM-based ASR," in *Speech Reading by Humans and Machines*, ser. NATO ASI Series, D. G. Stork and M. E. Hennecke, Eds., 1996, pp. 461–471.

- [59] M2VTS Multimodal Face Database Release 1.00 [Online]. Available: <http://www.tele.ucl.ac.be/PROJECTS/M2VTS/m2fdb.html>
- [60] S.W. Foo, Y. Lian, and L. Dong, "A simplified Viterbi matching algorithm for word partition in visual speech processing," in *Proc. 4th Eur. Workshop Image Analysis for Multimedia Interactive Services*, vol. S5, Apr. 2003.
- [61] S.W. Foo and L. Dong, "A boosted multi-HMM classifier for recognition of visual speech elements," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 2, Apr. 2003, pp. 285–288.

Say Wei Foo (SM'91) received the B.Eng. degree in electrical engineering from the University of Newcastle, Australia, in 1972, the M.Sc. degree in industrial and systems engineering from the University of Singapore in 1979, and the Ph.D. degree in electrical engineering from Imperial College London, University of London, London, U.K., in 1983.

From 1972 to 1973, he was with the Electrical Branch, Lands and Estates Department, Ministry of Defense, Singapore. From 1973 to 1992, he was with the Electronics Division of the Defense Science Organization, Singapore, where he conducted research and carried out development work on security equipment. From 1992 to 2001, he was the Associate Professor with the Department of Electrical and Computer Engineering, National University of Singapore. In 2002, he joined the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. He has authored and coauthored over 80 published articles. His research interests include speech signal processing, speaker recognition, and musical note recognition.

Yong Lian (S'91–M'95–SM'99) received the B.Sc. degree from the School of Management, Shanghai Jiao Tong University, China, in 1984 and the Ph.D. degree from the Department of Electrical Engineering, National University of Singapore, Singapore, in 1994.

He was with the Institute of Microcomputer Research of Shanghai Jiao Tong University, Brighten Information Technology Ltd., SyQuest Technology International, and Xyplex Inc. from 1984 to 1996. He joined the National University of Singapore in 1996, where he is currently an Associate Professor in the Department of Electrical and Computer Engineering. His research interests include digital filter design, VLSI implementation of high-speed digital systems, biomedical instrumentation, and RF IC design.

Dr. Lian received the 1996 IEEE Circuits and Systems Society's Guillemain-Cauer Award for the best paper published in IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—PART II: ANALOG AND DIGITAL SIGNAL PROCESSING. He currently serves as an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—PART II: ANALOG AND DIGITAL SIGNAL PROCESSING and has been an Associate Editor for *Circuits, Systems and Signal Processing* since 2000. He serves as the Secretary of the IEEE Circuits and Systems Society Biomedical Circuits and Systems Technical Committee and is a Member of the Digital Signal Processing Technical Committee.

Liang Dong received the B.Eng. degree in electronic engineering from Beijing University of Aeronautics and Astronautics, China, in 1997, the M.Eng. degree in circuit and system from the Second Academy of China Aerospace, China, in 2000. He is currently working towards the Ph.D. degree at the National University of Singapore.

His research interests include speech recognition, image processing, and video processing.