International Journal of Innovative Computing, Information and Control
Volume **9**, Number **8**, August **2013**

ICIC International ©2013 ISSN 1349-4198
pp. 3159–3172

VISION-BASED HYPOTHESIS INTEGRATION FOR INNER AND OUTER LIP CONTOUR DETECTION

Mau-Tsuen Yang and Zhen-Wei You

Department of Computer Science and Information Engineering
National Dong-Hwa University
No. 1, Sec. 2, Da Hsueh Rd., Shoufeng, Hualien 97401, Taiwan
mtyang@mail.ndhu.edu.tw

Received June 2012; revised October 2012

Abstract. *Vision-based lip contour detection is a challenging problem because lip and skin colors are similar, and the boundary between the lip and skin is usually ambiguous. We propose a real-time lip contour extraction algorithm by integrating several simple classifiers. Because the visual properties (concave-convex, shadow, illumination, and surface normal) of various parts of the lip contour vary considerably, we divided the whole lip contour into four parts (outer-upper, outer-lower, inner-upper, and inner-lower) to capture the specific characteristics of each part. The color/edge features and spatial-temporal consistency were exploited to make several simple hypotheses of lip contour pixels. For each lip contour part, a strong classifier was built by combining a set of hypotheses based on the AdaBoost algorithm to distinguish the lip contour from non-contour pixels. A deformable lip shape model was applied for fitting the lip contour by searching model parameters that maximize the classification scores along the contour. We compared the proposed algorithm with the Active Contour Model and Active Shape Model. The experiments show that both inner and outer lip contours can be detected and traced efficiently and reliably. The proposed lip contour extraction algorithm has potential for use in several fields, such as speech recognition and language learning.*

Keywords: Lip contour extraction, Lip image segmentation, Lip tracking, AdaBoost

1. **Introduction.** Lip shapes and motions convey valuable information during conversations. Summerfield [22] showed that the presence of lips increases word intelligibility considerably, especially in noisy conditions. In addition to communications between humans, extracting lip contours from a video stream has excellent potential in several human-machine interface applications, such as speaker authentication, facial expression analysis, lip reading, and audio-visual speech recognition. The extracted lip parameters can also be used to drive talking heads in avatar animation, video conferencing, and online language learning. Moreover, lip features and contours provide vital cues in video compression and audio to video synchronization.

There are two types of lip contours to be considered: the inner lip contour and the outer lip contour. It is difficult to extract the outer lip contour because the lip and skin colors are similar. Consequently, the boundary between the lip and skin is usually ambiguous. Extracting the inner lip contour is even more challenging because the inner lip region is not always visible, and the color inside the inner lip region varies because of the motion of the teeth and tongue. We propose an algorithm to extract both types of lip contours from an image sequence based on hypothesis integration using AdaBoost. A lip shape model composed of five quadratic polynomials was used to describe the actual lip contour to find the optimal lip shape parameters that approximate the real lip contours by training and integrating a set of simple rules (or classifiers). In the *training stage* (as shown in Figure

1(a)), the color/edge information and the spatial-temporal consistency are extracted to make several simple hypotheses. The AdaBoost algorithm is subsequently used to train the weak hypothesis weights and combine them into a strong classifier. In the *extraction stage* (as shown in Figure 1(b)), a score map is calculated based on the trained strong classifier. The optimal set of lip shape parameters with the highest classification score is subsequently found using a maximization search. Because each part of a lip contour has different visual properties (concave-convex, shadow, illumination, and surface normal), a complete lip contour is divided into four parts (outer-upper, outer-lower, inner-upper, and inner-lower), and a strong classifier is trained individually for each of these lip parts.

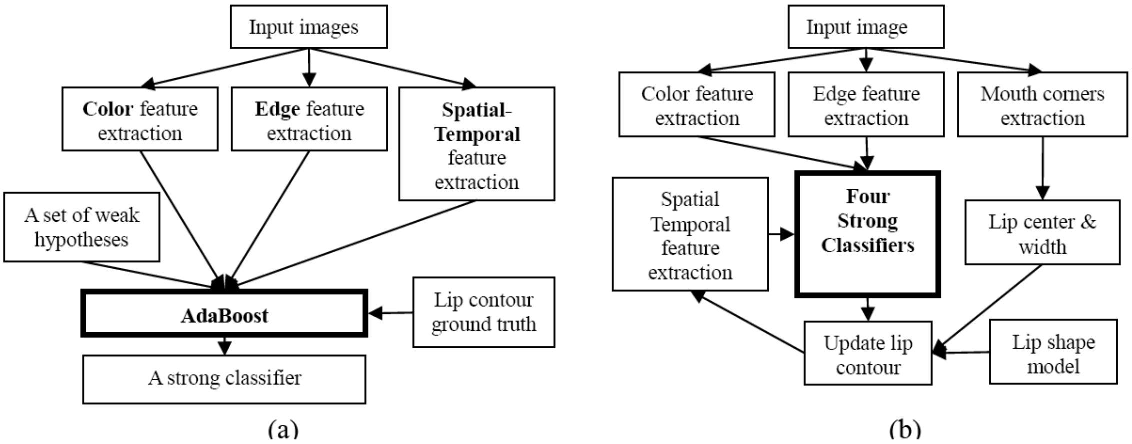


FIGURE 1. The flowchart of the proposed lip contour extraction system: (a) the training stage and (b) the extraction stage

The proposed approaches are significant in several aspects. First, we propose a novel approach to extract lip contours using AdaBoost by combining a set of simple hypotheses to build a strong classifier. The experimental results show that the proposed approach achieves slightly superior performance to ACM and ASM; moreover, it runs at least two times faster than ACM and ASM. Second, multiple heterogeneous features, including color, edge, and spatial-temporal consistency, were exploited to make simple hypotheses. The use and fusion of these hypotheses resulted in a robust system under diverse circumstances. Third, we propose a symmetric inner and outer lip contour model that can be determined using only nine parameters. Each lip contour is further divided into four parts to exploit the specific characteristics of each part. Fourth, the proposed lip contour extraction was successfully implemented in a language learning application called VEC3D, in which remote learners can communicate through avatars with live voice and synchronized lip animations.

The remainder of this paper is organized as follows: Section 2 provides reviews of related works; Section 3 introduces a lip shape model used to approximate the actual lip contours; Section 4 presents the proposed lip contour extraction algorithm; Section 5 presents the training of strong classifiers based on the AdaBoost algorithm; Section 6 provides the experimental results; Section 7 presents a potential application of language learning; and finally, Section 8 offers a conclusion.

2. **Related Works.** Several lip segmentation algorithms based only on color information have been proposed. Zhang et al. [14] used hue and edge cues to segment lips. Eveno et al. [5] transformed the RGB color space into a chromatic curve map to efficiently discriminate lips from skin. These methods performed segmentation pixel-wise without shape constraints, which resulted in unstable segmentations. To consider the shape constraint,

a number of methods [1,4,12,17] extracted lip contours based on Snakes. Snakes [7] are Active Contour Models (ACM) that dynamically alter the shape to fit edges in an image. A problem with Snakes is that the edges between the lip and skin are usually unclear. To address this problem, Wakasugi et al. [13] defined a term called separability, which considers multidimensional distributions. The separability can yield superior edges in ambiguous regions between the lip and skin pixels. Another problem with Snakes is error accumulations. To address this problem, a number of methods used the Active Shape Model (ASM) [9], which uses prior shape information as a constraint to limit model deformation. ASM can yield accurate results with the known shape information of a target object; however, a large amount of training data is required. Cootes et al. proposed the Active Appearance Model (AAM) [2], which uses the appearance and the shape information in the target object. Instead of modeling the whole object region, Cristinacce and Cootes [23] modeled a set of local feature templates, called the Constrained Local Model (CLM), to locate a set of feature points.

A few hybrid approaches combining different cues have been developed for lip contour extraction. Leung et al. [8,18] proposed a fuzzy c-means with shape function (FCMS) that considered distances in color and spatial space, and an ellipse model was used to provide a rough lip shape hint for clustering. Dansereau et al. [3] used the Markov Random Field (MRF) to segment lips. They proposed an energy function based on edge information and spatial consistency. Segmentation was performed by minimizing the energy function. Yang et al. [16] analyzed three dynamic probability maps based on color, shape, and edge features, respectively. These probability maps were used to extract both inner and outer lip contours using a grid-based gradient-ascent approach. Saeed and Dugelay [19] combined edge-based and region-based ACM detection using simple AND/OR logical operators to compensate for the weakness of a standalone approach.

A number of studies explored possibilities to estimate the 3D shape of human lips from a 2D video stream. Basu et al. [20] proposed a mesh-based method, whereas Gastelum et al. [21] proposed a particle-based approach to model 3D lips. Three-dimensional lip modeling captures and reflects realistic lip motions in space. However, 3D reconstruction is computationally costly and frequently requires human interaction. Therefore, they are unsuitable for real-time applications.

Current lip contour extraction methods based on color segmentation usually label pixels as either lip or skin according to their distance or similarity in color space. However, labeling is a difficult task because of dynamic illumination, the changing appearance of the inner lip region, and the presence of shadows or specular reflection. To address these problems, we extracted lip contour pixels, instead of pixels inside or outside lips, by combining several simple hypotheses. AdaBoost [6] is an algorithm that is used for constructing a strong classifier as a linear combination of weak hypotheses. We used the AdaBoost algorithm to train and integrate the pixel classifiers for the extraction of lip contour pixels.

3. Lip Shape Model.

To simulate the movement of natural lips, five quadratic polynomials were used to model the lip contour (three for the outer lip contour and two for the inner lip contour) [16]. Suppose that both the inner and outer lip contours share the same centroid p_c located at (x_c, y_c). The upper lip contour y_{ou} of the outer lip model is described by two horizontally symmetrical curves y_1 and y_2:

$$\begin{cases} y_1(x) = cu_o\left[1 - \frac{(x-x_c+b)^2}{(\delta_o-b)^2}\right] + y_c \\ y_2(x) = cu_o\left[1 - \frac{(x-x_c-b)^2}{(\delta_o-b)^2}\right] + y_c \end{cases} \tag{1}$$

where b is an offset that controls the horizontal displacement of two curves, y_1 and y_2. δ_o is the distance between the centroid p_c and the outer mouth corners (the intersection of y_{ou} and y_{ol}). cu_o is the maximal distance from the centroid p_c to the curve y_{ou}. The complete upper lip contour y_{ou} of the outer lip model was set to the maxima of y_1 and y_2. Similarly, suppose that cl_o is the maximal distance from the centroid p_c to the curve y_{ol}, the lower lip contour y_{ol} of the outer lip model is modeled as

$$\begin{cases} y_{ou}(x) = \max(y_1(x), y_2(x)) \\ y_{ol}(x) = -cl_o\left[1 + \dfrac{(x - x_c)^2}{\delta_o^2}\right] + y_c \end{cases} \tag{2}$$

Furthermore, the inner lip model with the upper lip contour y_{iu} and the lower lip contour y_{il} is represented as

$$\begin{cases} y_{iu}(x) = cu_i\left[1 - \dfrac{(x - x_c)^2}{\delta_i^2}\right] + y_c \\ y_{il}(x) = -cl_i\left[1 + \dfrac{(x - x_c)^2}{\delta_i^2}\right] + y_c \end{cases} \tag{3}$$

where δ_i is the distance between the centroid p_c and the inner mouth corners (the intersection of y_{iu} and y_{il}). cu_i and cl_i are the maximal distances from the centroid p_c to curves y_{iu} and y_{il}, respectively. In summary, a vector $\lambda = \{x_c, y_c, cu_o, cl_o, \delta_o, b, cu_i, cl_i, \delta_i\}$ represents the complete lip shape model.

4. Lip Contour Extraction.

Lip contour is extracted by combining several simple hypotheses. Each hypothesis h compares features k in the feature vector K, which consists of the color, edge, and spatial-temporal information in the neighboring pixels. A strong classifier H is constructed by combining these weak classifiers h to estimate the probability of a pixel belonging to the lip contour.

4.1. Color features/hypotheses.

Several feature images are extracted from an input lip image I. First, the normalized red image r, normalized green image g, and gray level image u are computed. The color difference between r and g is subsequently calculated to obtain the feature image π. For a lip pixel, the normalized red intensity is usually larger than the green intensity. Thus, the feature π emphasizes the lip region. The Sobel edge detector is applied to images r, u, and π to obtain feature images E_r, E_u, and E_π, respectively. Teeth usually cause difficulty in the lip detection process. To address the teeth problem, an edge difference image E'_u is calculated by subtracting E_r from E_u to suppress the teeth edges inside the inner lip region. Figure 2 shows a number of these feature images.

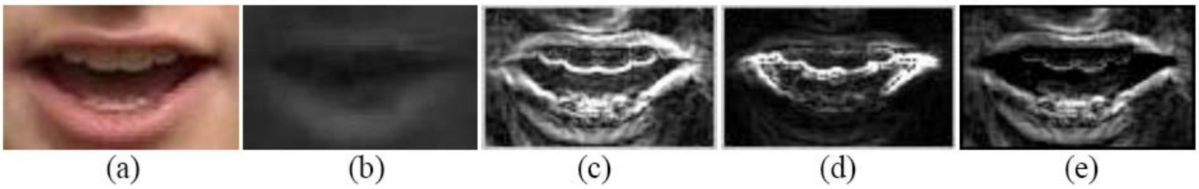


Figure 2. Some images obtained from preprocessing: (a) input lip image I; (b) color difference image $\pi = r - g$; (c) E_u, the edge image of u; (d) E_r, the edge image of r and (e) edge difference image $E'_u = E_u - E_r$

The colors of lip contour pixels are generally unstable. Thus, the relationship between neighboring pixels is used to determine the possibility for a pixel to be on the lip contour. For each pixel $p(x, y)$, several features k_i are retrieved from neighboring pixels to form a q-dimensional feature vector $K(p) = \{k_1, k_2, \ldots, k_q\}$. Because the primary lip axis usually lies horizontally, we focus mainly on analyzing neighboring pixels in the vertical direction. For example, $k_1 = r_u = r(x, y-1)$ is the normalized red intensity of the neighboring pixel above p, and $k_2 = r_l = r(x, y+1)$ is the normalized red intensity of the neighboring pixel below p. Table 1 shows a number of other features in vector K.

TABLE 1. List of some features of a pixel $p(x, y)$

Index	ID	Description
k_1	r_u	$r(x, y+1)$, r: normalized red channel
k_2	r_l	$r(x, y-1)$
k_3	g_u	$g(x, y+1)$, g: normalized green channel
k_4	g_l	$g(x, y-1)$
k_5	π_u	$\pi(x, y+1)$, $\pi = r - g$
k_6	π_l	$\pi(x, y-1)$
k_7	u_u	$u(x, y+1)$, u is a gray level image
k_8	u_l	$u(x, y-1)$
k_9	Eu'_u	$E'_u(x, y+1)$, $E' = E - E_r$
k_{10}	Eu'_l	$E'_u(x, y-1)$
k_{11}	Er_u	$E_r(x, y+1)$, E_r is the sobel edge image of r
k_{12}	Er	$E_r(x, y-1)$
k_{13}	$E\pi_u$	$E_\pi(x, y+1)$, E_π is the sobel edge image of π
k_{14}	$E\pi_l$	$E_\pi(x, y-1)$
$k_{15} - k_{22}$	ec_i	Edge direction feature
k_{23}	d_{pre}	Spatial-temporal consistency feature

With the feature vector K, several simple hypotheses (or weak classifiers) are made for lip contour pixels. The feature vector $K(p)$ is sent to each weak classifier h_j to acquire a classification score. A pixel with a higher score is more likely to be on the lip contour. Each weak classifier h_j can be defined using the following general equation:

$$h_j(K) = \begin{cases} +1 & \text{if } \beta_j f_j(K) < \beta_j \theta_j \quad \beta_j = \{+1, -1\} \\ -1 & \text{otherwise} \end{cases} \tag{4}$$

where β_j controls the direction of the comparison, θ_j is the threshold, and $f_j(K)$ retrieves the participating features for the weak classifier. If the input feature vector K satisfies the hypothesis, h_j returns +1; otherwise, it returns −1.

Each weak hypothesis is made by comparing feature elements in K. For example, the first hypothesis based on a comparison between the color features is proposed as $r_l < r_u$, i.e., $k_1 < k_2$, or

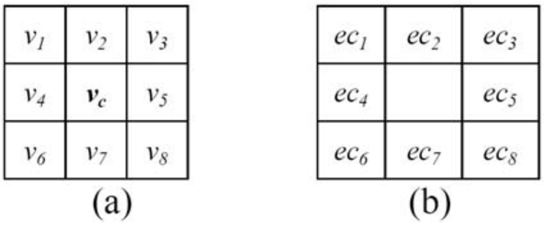
$$f(K) = k_1 - k_2 < \theta \tag{5}$$

where the threshold θ is equal to 0 in this case. Most of our weak hypotheses were designed with $\theta = 0$ to improve the robustness under changing illumination. A number of other weak hypotheses are shown in Table 2.

4.2. **Edge and spatial-temporal features/hypotheses.** In addition to the color information comparison, several other hypotheses are made to exploit the edge information and the spatial-temporal consistency. To exploit the edge information, the consistency of

TABLE 2. List of some hypotheses

Index	Description
h_1	$\beta_1 r_l < \beta_1 r_u$
h_2	$\beta_2 g_l < \beta_2 g_u$
h_3	$\beta_3 \pi_l < \beta_3 \pi_u$
h_4	$\beta_4 u_l > \beta_4 u_u$
h_5	$\beta_5 Eu'_l < \beta_5 Eu'_u$
h_6	$\beta_6 Er_l < \beta_6 Er_u$
h_7	$\beta_7 E\pi_l < \beta_7 E\pi_u$
h_8	$\beta_8(ec_2 * ec_7) < 0$
h_9	$\beta_9(ec_1 * ec_8) < 0$
h_{10}	$\beta_{10}(ec_3 * ec_6) < 0$
h_{11}	$d_{pre} < \theta_{11}$



v_1	v_2	v_3
v_4	$\mathbf{v}_c$	v_5
v_6	v_7	v_8

(a)

ec_1	ec_2	ec_3
ec_4		ec_5
ec_6	ec_7	ec_8

(b)

FIGURE 3. Edge features of a pixel p located at the center of the mask: (a) edge directions v_i and (b) edge consistency ec_i

the edge direction is used to design a set of edge hypotheses. As shown in Figure 3, vector ν_c represents the Sobel edge direction of a pixel p, and $\{\nu_1, \nu_2, \ldots, \nu_8\}$ represents the edge directions of the neighboring pixels. Let $\{ec_1, ec_2, \ldots, ec_8\}$ represent the edge consistency of the neighboring pixels that are defined as the inner product of the neighboring edge directions:

$$ec_i = v_i \cdot v_c \quad 1 \leq i \leq 8 \tag{6}$$

Three hypotheses based on the edge consistencies across the pixel p were designed as

$$ec_1 \times ec_8 < 0, \quad ec_2 \times ec_7 < 0, \quad ec_3 \times ec_6 < 0 \tag{7}$$

The edge consistencies across the lip contour have different signs to ensure that the pixels on the lip contour generally satisfy these hypotheses. Conversely, these hypotheses are likely to fail for pixels that originate from noise or homogeneous regions.

Because the lip contours move smoothly in an image sequence captured at a real-time frame rate, the displacement of corresponding lip contour pixels between consecutive frames is limited to a certain range. To exploit this spatial-temporal consistency, a new feature d_{pre} is defined as the distance between p and the nearest contour pixel in the last frame. A new hypothesis based on the spatial-temporal consistency was designed as $d_{pre} < \theta_j$.

4.3. **Strong classifiers and score maps.** If a pixel p satisfies most of the weak hypotheses, the pixel is possibly a lip contour pixel. The final pixel classifier H linearly combines a set of weak classifiers $h^{(t)}$ with respective weight $\alpha^{(t)}$:

$$H(K) = \sum_{t=1}^{T} \alpha^{(t)} h^{(t)}(K) \tag{8}$$

The hypothesis weight $\alpha^{(t)}$ determines the importance of the t-th hypothesis and is trained by the AdaBoost algorithm, which is introduced in Section 5. Because the characteristics of each part of the lip contour differ, we divided the whole lip contour into four parts (outer-upper y_{ou}, outer-lower y_{ol}, inner-upper y_{iu}, and inner-lower y_{il}). Four classifiers (H_{ou}, H_{ol}, H_{iu}, and H_{il}) were established, and four score maps (S_{ou}, S_{ol}, S_{iu}, and S_{il}) were computed for these lip contour parts, respectively: $S_i(p) = H_i(K(p)) \quad i \in \{ou, ol, iu, il\}$.

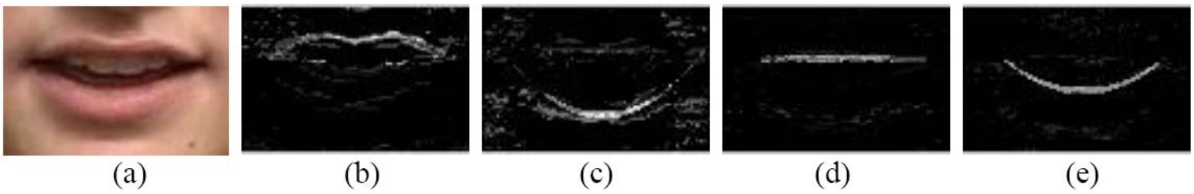


Figure 4. Score maps for different lip contour parts: (a) input lip image I; (b) outer upper lip score map S_{ou}; (c) outer lower lip score map S_{ol}; (d) inner upper lip score map S_{iu} and (e) inner lower lip score map S_{il}

Examples of score maps for these lip contour parts are shown in Figure 4. These score maps provide valuable hints for the extraction of lip contour pixels. To find the optimal-fit lip contour parameters, four goal functions were defined by

$$F_i(\lambda) = \sum_{x=x_c-\delta_o}^{x_c+\delta_o} S_i(p(x, y_i(x))) \quad i \in \{ou, ol, iu, il\} \tag{9}$$

To increase the speed of the optimal-fit lip contour parameter search for a set λ that maximizes the goal functions, we reduced the search dimension by sequentially finding the lip contour parts in a fixed order. First, x_c, y_c, and δ_o were anchored by detecting lip corners using a simple process, as described in the following paragraph. Second, cu_o and b were fixed by approximating the outer upper lip contour based on the goal function F_{ou}. Third, cl_o was determined by extracting the outer lower lip contour based on the goal function F_{ol}. Fourth, cu_i and δ_i were determined by detecting the inner upper lip contour based on the goal function F_{iu}. Finally, we found cl_i that optimally fits the inner lower lip contour based on the goal function F_{il}:

$$\begin{cases} \lambda_{ou} = \max \underset{cu_o, b}{\arg} F_{ou}(\lambda) & cu_o \in [1, y_{top}] \quad b \in \left[0, \frac{\delta_o}{2}\right] \\ \lambda_{ol} = \max \underset{cl_o}{\arg} F_{ol}(\lambda) & cl_o \in [1, y_{bottom}] \\ \lambda_{iu} = \max \underset{cu_i, \delta_i}{\arg} F_{iu}(\lambda) & cu_i \in [0, cu_o) \quad \delta_i \in [1, \delta_o) \\ \lambda_{il} = \max \underset{cl_i}{\arg} F_{il}(\lambda) & cl_i \in [1, cl_o) \end{cases} \tag{10}$$

Two lip corners were detected by horizontally scanning the normalized red edge image E_r. The scan was applied twice: from left to center and from right to center. In each scan, the first touched clear end point is used as the lip corner. The extracted lip corners, $p_r(x_r, y_r)$ and $p_l(x_l, y_l)$, can be used to update the lip center (x_c, y_c) and the width of the outer lips (δ_o):

$$x_c = \frac{(x_r + x_l)}{2}, \quad y_c = \frac{(y_r + y_l)}{2}, \quad \delta_o = \frac{(x_r - x_l)}{2} \tag{11}$$

5. **Classifiers Training Using the AdaBoost Algorithm.** We used an iterative AdaBoost algorithm with T iterations to obtain the weights for the hypotheses. In iteration t, the hypothesis $h^{(t)}$ that optimally discriminates the lip contour from non-contour pixels was chosen and assigned a high hypothesis weight $\alpha^{(t)}$. At the end, T most important hypotheses were selected, and the remaining hypotheses were discarded. To train a strong classifier using AdaBoost, a set of training samples was derived by

$$(K_1, G_1), \ldots, (K_m, G_m) \quad G_i \in \{+1, -1\} \tag{12}$$

where m is the total number of the training samples, G is the manually produced ground truth in that $+1$ represents a lip contour pixel, and -1 represents a non-contour pixel. The sample weight D_i representing the importance of the i-th sample is initiated as a constant and updated over time. Because the numbers of positive and negative samples in our sample set differed, the sample weights D_i were initialized as

$$D_i^{(1)} = \begin{cases} \frac{1}{2P} & \text{if } G_i = +1 \\ \frac{1}{2N} & \text{if } G_i = -1 \end{cases} \tag{13}$$

where P is the number of positive samples, and N is the number of negative samples. A set of hypotheses (as shown in Table 2) was subsequently trained iteratively. In iteration t, the optimal hypothesis $h^{(t)}$ with minimal classification error $\varepsilon^{(t)}$ was chosen:

$$\begin{cases} h^{(t)} = \arg\min\limits_{h_j} \sum\limits_{i=1}^{m} D_i^{(t)} [G_i \neq h_j(K_i)] \\ \varepsilon^{(t)} = \sum\limits_{i=1}^{m} D_i^{(t)} \left[G_i \neq h^{(t)}(K_i)\right] \end{cases} \tag{14}$$

The overall error ε for each hypothesis h is weighted according to the sample weights D. If the minimal error $\varepsilon^{(t)}$ is greater than 0.5, none of the hypotheses can make an optimal decision. Consequently, the training must be terminated and the hypotheses must be redesigned. Otherwise, the weight $\alpha^{(t)}$ of the hypothesis $h^{(t)}$ is updated as

$$\alpha^{(t)} = \frac{1}{2} \ln \frac{1 - \varepsilon^{(t)}}{\varepsilon^{(t)}} \tag{15}$$

Generally, hypotheses that yield lower classification errors must obtain higher weights. Therefore, the sample weights for the next iteration are updated as

$$D_i^{(t+1)} = \frac{D_i^{(t)} e^{-\alpha^{(t)} G_i h^{(t)}(K_i)}}{Z_t} \tag{16}$$

where Z_t is a normalization factor that is chosen to ensure that the sum of D_i is equal to one. This update rule reduces the weights of samples that have already been correctly classified, and increases the weights of samples that cannot be correctly classified in this round. Consequently, the forthcoming iterations focus more on the hard classified samples instead of the easy classified samples.

6. **Experimental Results.** The proposed lip extraction system was tested on five video sequences with various speakers from the CUAVE (Clemson University Audio Visual Experiments) database [11]. These videos contain 24-bit color images with 720 × 480 resolution in MPEG-2 compression format. The length of each video sequence is approximately 10s (300 frames). Fifty lip images (approximately 300,000 pixels) with ground truth lip contour were trained using the AdaBoost algorithm. Sample pixels retrieved from various sub-regions were used to train classifiers for various lip contour parts. Table 3 shows the top 10 hypotheses for each strong classifier after the training. For the lip contour extraction, an initial bounding box of 100 × 60 in size was provided manually at

Table 3. Top 10 weak classifiers by AdaBoost for each lip contour part

Outer upper lip contour.		**Outer lower lip contour**		**Inner upper lip contour**		**Inner lower lip contour**	
Hypothesis	Weight	Hypothesis	Weight	Hypothesis	Weight	Hypothesis	Weight
$r_l > r_u$	0.37	$u_l < u_u$	0.27	$g_l > g_u$	0.52	$\pi_l > \pi_u$	0.55
$g_l < g_u$	0.32	$\pi_l < \pi_u$	0.22	$Eu'_l < Eu'_u$	0.30	$Eu'_l > Eu'_u$	0.35
$d_{pre} < 5$	0.21	$g_l > g_u$	0.17	$\pi_l < \pi_u$	0.28	$d_{pre} < 5$	0.35
$ec_1 * ec_8 < 0$	0.19	$d_{pre} < 5$	0.16	$ec_1 * ec_8 < 0$	0.17	$r_l < r_u$	0.27
$ec_2 * ec_7 < 0$	0.17	$Eu'_l < Eu'_u$	0.14	$ec_3 * ec_6 < 0$	0.16	$E\pi_l < E\pi_u$	0.19
$E\pi_l < E\pi_u$	0.17	$\pi_l < \pi_u$	0.07	$r_l < r_u$	0.15	$ec_3 * ec_6 < 0$	0.18
$Eu'_l > Eu'_u$	0.16	$r_l > r_u$	0.07	$E\pi_l < E\pi_u$	0.12	$ec_2 * ec_7 < 0$	0.17
$\pi_l > \pi_u$	0.15	$ec_3 * ec_6 > 0$	0.07	$ec_2 * ec_7 < 0$	0.07	$g_l > g_u$	0.16
$Er_l > Er_u$	0.14	$ec_1 * ec_8 < 0$	0.06	$Er_l < Er_u$	0.07	$u_l > u_u$	0.13
$u_l < u_u$	0.14	$ec_2 * ec_7 > 0$	0.05	$d_{pre} < 5$	0.07	$ec_1 * ec_8 < 0$	0.12

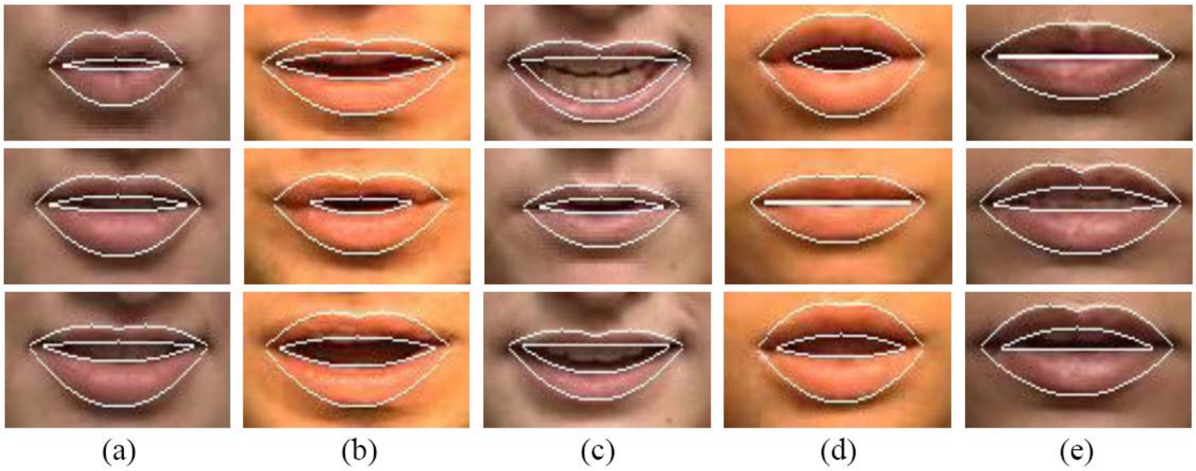


Figure 5. The extracted lip contours on five different speakers in CUAVE database. Three images in a column come from the same speaker at different time instants. (a) Speaker 1; (b) Speaker 2; (c) Speaker 3; (d) Speaker 4 and (e) Speaker 5.

the first frame. Figure 5 shows examples of the extracted lip contours from five videos with various speakers.

According to the MPEG-4 facial animation standard [10], mouth motion is described using 20 facial animation parameters (FAP). Each FAP indicates the movement of a feature point on a human face, and is defined as the relative distance between a feature point and the corresponding point on a neutral face that refers to an expressionless face. Sixty-eight FAPs were divided into 10 groups. Group 8 consists of 10 FAPs that control the motion of 10 outer lip feature points, and Group 2 consists of another 10 FAPs that control the motion of eight inner lip feature points. Generally, each FAP is related to the movement of a feature point, except that two FAPs are defined for each inner lip corner to model both horizontal and vertical motion. To map the extracted lip contour parameters to the FAP, a number of feature points are located at extreme points and intersection points of the extracted lip contours. Other feature points are evenly placed between these extreme points and intersection points along the extracted lip contours. Each FAP is determined by measuring the horizontal or vertical displacement of the feature point regarding either mouth width (MW) or mouth-nose separation (MNS). The FAP detection errors are calculated based on the mean squared error between the extracted lip FAPs and the real lip FAPs:

$$\begin{cases} AE_O = \dfrac{\sum\limits_{i=1}^{frame_{num}} \sum\limits_{j=1}^{ap_O} \left(\phi_j^{(i)} - \psi_j^{(i)}\right)^2}{frame_{num} \times ap_O} \\ AE_I = \dfrac{\sum\limits_{i=1}^{frame_{num}} \sum\limits_{j=1}^{ap_I} \left(\phi_j^{(i)} - \psi_j^{(i)}\right)^2}{frame_{num} \times ap_I} \end{cases} \quad (17)$$

where AE_O and AE_I are the FAP detection errors for the outer and inner lip contours, respectively. Similarly, ap_O and ap_I are the number of FAPs in the outer and inner lip contours, respectively. In the i-th frame, $\phi_j^{(i)}$ and $\psi_j^{(i)}$ are the j-th FAPs for the extracted and real lip contours, respectively. The detection errors are further normalized by

$$\begin{cases} NAE_O = \frac{\sqrt{AE_O}}{1024} \times 100\% \\ NAE_I = \frac{\sqrt{AE_I}}{1024} \times 100\% \\ NAE = \frac{\sqrt{(NAE_O + NAE_I)/2}}{1024} \times 100\% \end{cases} \quad (18)$$

Table 4 shows a comparison of the FAP detection errors of three strong classifiers that consider the top 10, 7, and 3 hypotheses, respectively. The detection errors decreased as the number of integrated hypotheses increased. However, integrating more than 10 hypotheses does not help because the top 10 hypotheses dominate the integration, and the weightings of the remaining hypotheses are weak. Therefore, the following experiments using the proposed method were conducted with the top 10 hypotheses. Table 5 shows a comparison of the detection errors for outer and inner lip contours. Generally, the detection errors for inner lip contours are more significant because of the complexity of the region inside the inner lip contours. In the sequence of Speaker 2, the detected inner lip contour was unstable because the teeth and tongue appeared and disappeared frequently in the sequence.

Table 4. FAP detection errors (NAE) of three strong classifiers that combine the top 10, 7, or 3 weak hypotheses respectively

	Speaker 1	Speaker 2	Speaker 3	Speaker 4	Speaker 5	Avg.
Top 10	6.75	9.56	5.89	5.38	6.45	6.81
Top 7	7.34	9.93	5.97	5.73	7.71	7.34
Top 3	9.14	9.80	8.42	8.88	7.67	8.78

Table 5. FAP detection errors (NAE) for outer and inner lip contours

	Speaker 1	Speaker 2	Speaker 3	Speaker 4	Speaker 5	Avg.
Outer lip (NAE_O)	7.88	7.12	4.47	5.35	5.87	6.14
Inner lip (NAE_I)	5.38	11.48	7.03	5.42	6.98	7.26

For comparison, the two main approaches, ACM and ASM, were implemented and applied on the same five video sequences in the CUAVE database. Generally, the traditional edge-based ACM is sensitive to uneven illuminations and teeth appearances; therefore, we used the up-to-date ACM fusion approach [19], which uses and combines edge and region features. An active contour consists of 10 control points on the outer lip contour and eight control points on the inner lip contour. These control points actively move over time to reduce internal and external energy, and the lip contours are deformed dynamically.

We also compared the performance of the proposed method with a standard ASM. In the implementation of ASM, 50 lip images were trained using Principle Component Analysis (PCA) and tracked by an iterative scheme [9]. The 50 trained lip images were the same as those used in the training of the proposed method, and each trained lip image was manually marked with 10 feature points on the outer lip contour and eight feature points on the inner lip contour.

Because ACM and ASM tracking is performed on feature points instead of FAPs, we defined another detection error based on feature points for comparison:

$$\begin{cases} FE_O = \dfrac{\sum\limits_{i=1}^{frame_{num}} \sum\limits_{j=1}^{fp_o} \left\| \Phi_j^{(i)} - \Psi_j^{(i)} \right\|}{frame_{num} \times fp_O} \\ FE_I = \dfrac{\sum\limits_{i=1}^{frame_{num}} \sum\limits_{j=1}^{fp_I} \left\| \Phi_j^{(i)} - \Psi_j^{(i)} \right\|}{frame_{num} \times fp_I} \end{cases} \tag{19}$$

where FE_O and FE_I are the feature point detection errors for the outer and inner lip contours, respectively. Similarly, fp_O and fp_I are the number of feature points in the outer and inner lip contours, respectively. In the i-th frame, $\Phi_j^{(i)}$ and $\Psi_j^{(i)}$ are the coordinates of j-th feature points for the extracted and real lip contours, respectively. The operator $\| \cdot \|$ represents Euclidean distance.

Table 6 shows a comparison of the performance of ACM fusion [19], ASM, and the proposed method. As shown in the table, ACM fusion and ASM can achieve similar performance. The video of Speaker 4 is an exception, in that ASM outperforms ACM fusion by the guidance of the constrained mouth shapes, whereas ACM is trapped in a local minimum in the latter part of the video. The table also shows that the proposed method can yield slightly superior detection compared with ASM, except for the sequence of Speaker 2. Further investigation of the sequence of Speaker 2 revealed that ACM and ASM outperformed the proposed method when the captured lip contours were not horizontally symmetric. Although asymmetric lips are uncommon, this is the limitation of the proposed method because our search was based on a symmetric lip contour model.

Table 6. Feature point detection errors (FE) for ACM fusion, ASM, and the proposed method

	Speaker 1	Speaker 2	Speaker 3	Speaker 4	Speaker 5	Avg.
ACM fusion [19]	5.25	4.83	4.78	5.51	4.98	5.07
ASM	5.42	4.69	4.73	4.37	5.45	4.93
Proposed method	4.73	6.69	4.12	3.77	4.52	4.77

ASM and the proposed method require training with ground truth in advance. After the training, the three methods were initialized by specifying mouth shapes manually at the first frame. In the extraction stage, the processing speed occasionally varied slightly because of the differences of dynamic image contents. Generally, the frame rate of ACM fusion is approximately 21 frames per second (fps), whereas the frame rate of ASM tracking is approximately 25fps. The processing speed of our system is at least two times faster than that of ACM and ASM, and is sufficient for most real-time applications.

As an alternative to a video stream in a database, the proposed lip detection algorithm can use live images captured by a webcam as the input. The proposed lip extraction system was tested on six video sequences with various members from our laboratory. These videos contain 24-bit color images with 640 × 480 resolution at 25fps. The length

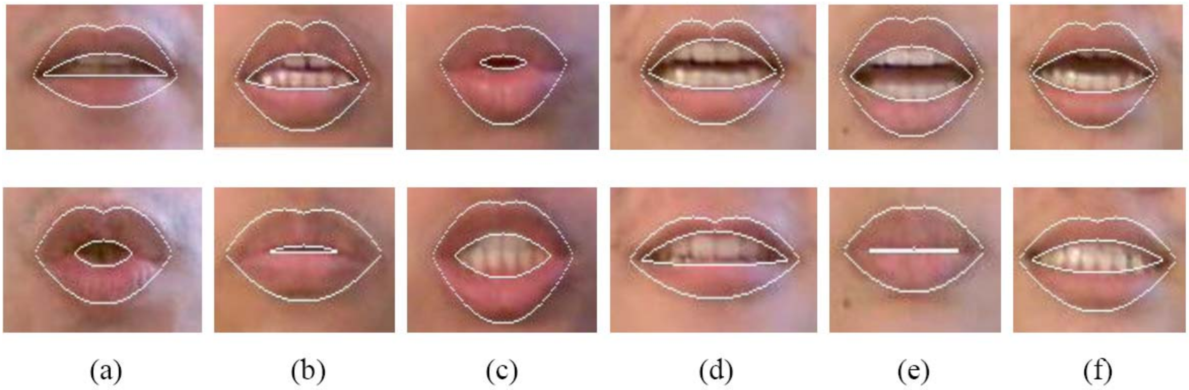


Figure 6. The extracted lip contours on six different members in our lab. The videos are captured using an off-the-shelf webcam. Three images in a column come from the same member at different time instants. (a) Member 1; (b) Member 2; (c) Member 3; (d) Member 4; (e) Member 5 and (f) Member 6.

of each video sequence is approximately 20s. Figure 6 shows the extracted lip contours from examples that are not included in the CUAVE database. Because the classifier training by Adaboost was performed in advance, only the score map calculation and lip parameter search were performed online, and the proposed lip contour extraction was applied to live video without a noticeable delay. The proposed lip extraction can run at 60fps on a standard personal computer (PC) with an Intel Core i7-3770 3.4 GHz CPU and pre-captured video stored on the hard disk. The experimental results show that the proposed method can extract inner and outer lip contours accurately and efficiently.

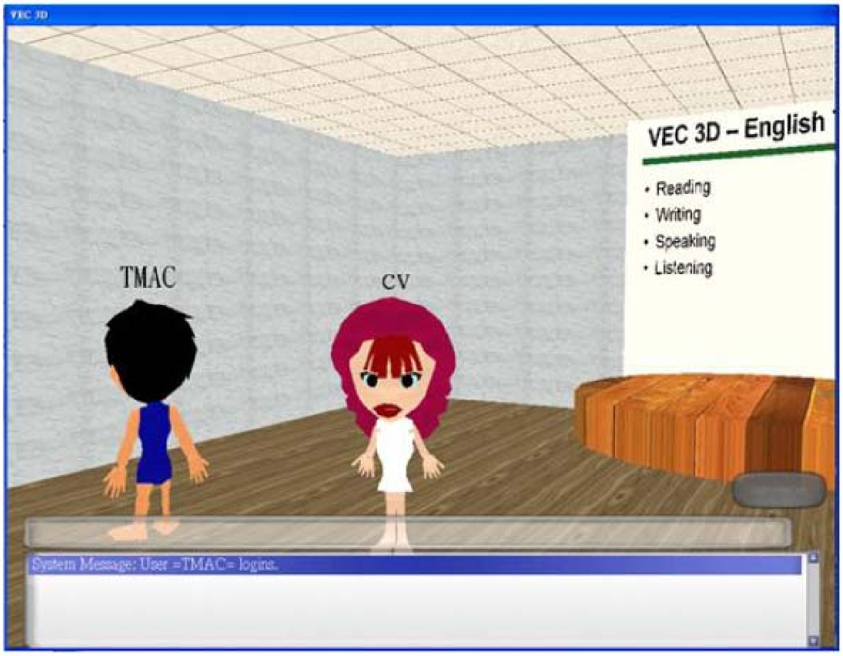
7. **Applications.**

The proposed vision-based lip extraction is useful in several human-machine interface applications, such as speaker authentication, facial expression analysis, lip reading, and audio-visual speech recognition. As a case study, the proposed lip contour extraction algorithm was successfully applied in a language-learning application called VEC3D (3D Virtual English Classroom [15]). VEC3D is a virtual campus-like environment in which students and teachers can log in to learn or teach English online. Live voice communication and virtual mouth rendering were implemented in VEC3D. First, the input lip images were captured using an off-the-shelf web-camera with a resolution of 320 × 240 at 30fps. The inner and outer lip contour parameters were subsequently extracted using the proposed algorithm. As shown in Figure 7, the extracted lip parameters were sent to remote clients to render the mouth of a virtual avatar in VEC3D realistically and dynamically.

In language learning, the lip shape and motion are vital clues for the training of both oral expression and listening comprehension. With the ability of real-time lip contour extraction and animation, VEC3D empowers online language learners to listen to the live voice of a speaker and view the synchronized lip shape and motion. Moreover, online learners can remain anonymous without revealing their faces through live video.

8. **Conclusion.**

We propose a real-time lip extraction algorithm that can reliably and efficiently extract both outer and inner lip contours. The color/edge information and spatial-temporal consistency were extracted and used to make weak hypotheses to discriminate lip contour from non-contour pixels. We divided the whole lip contour into four parts (outer-upper, outer-lower, inner-upper, and inner-lower) to capture the characteristics of each part of the lip contour. The AdaBoost algorithm was used to linearly integrate



System Message: User =TMAC= logins.

FIGURE 7. The virtual avatar with animated lips in VEC3D

the weak hypotheses into a strong classifier for each lip contour part. A deformable lip shape model was subsequently used to approximate the outer and inner lip contours based on the classification scores.

The limitations of the proposed system are non-symmetric mouths and teeth confusion, especially on the inner lower lip contour. A more complex lip shape model should be designed to manage a non-symmetrical mouth. The ambiguity caused by teeth could be mitigated by adding more hypotheses or removing the teeth in preprocessing.

Acknowledgments. This work was supported by the National Science Council, Taiwan.

REFERENCES

[1] P. Aleksic, J. Williams, Z. Wu and A. Katsaggelos, Audio-visual speech recognition using MPEG-4 compliant visual features, *EURASIP Journal on Applied Signal Processing*, vol.2002, pp.1213-1227, 2002.

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

[3] R. Dansereau, C. Li and R. Goubran, Lip feature extraction using motion, color, and edge information, *IEEE International Workshop on Haptic, Audio and Visual Environments and Their Applications*, pp.1-6, 2003.

[4] P. Delmas, P. Coulon and V. Fristot, Automatic snakes for robust lip boundaries extraction, *ICASSP*, pp.3069-3072, 1999.

[5] N. Eveno, A. Caplier and P. Coulon, Key point based segmentation of lips, *IEEE International Conference on Multimedia and Expo*, vol.2, pp.125-128, 2002.

[6] Y. Freund and R. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, *Computational Learning Theory: Eurocolt*, pp.23-37, 1995.

[7] M. Kass, A. Witkin and D. Terzopoulos, Snakes: Active contour models, *International Journal on Computer Vision*, vol.1, no.4, pp.321-331, 1988.

[8] S. Leung, S. Wang and W. Lau, Lip image segmentation using fuzzy clustering incorporating an elliptic shape function, *IEEE Transactions on Image Processing*, vol.13, no.1, pp.51-62, 2004.

[9] T. Cootes, C. Taylor, D. Cooper and J. Graham, Active shape models – Their training and application, *Computer Vision and Image Understanding*, vol.61, no.1, pp.38-59, 1995.

[10] MPEG Video Group, ISO/IEC JTC1/SC29/WG11 N2502, FDIS of ISO/IEC 14496-2, *Generic Coding of Aaudio-Visual Objects: Part 2 – Visual*, (MPEG-4), 1998.

[11] E. Patterson, S. Gurbuz, Z. Tufekci and J. Gowdy, Moving-talker, speaker-independent feature study and baseline results using the CUAVE multimodal speech corpus, *EURASHIP Journal of Applied Signal Processing*, no.11, pp.1189-1201, 2002.

[12] D. Terzopoulos and K. Waters, Analysis and synthesis of facial image sequences using physical and anatomical models, *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol.25, pp.569-579, 1993.

[13] T. Wakasugi, M. Nishiura and K. Fukui, Robust lip contour extraction using separability of multidimensional distributions, *IEEE International Conference on Automatic Face and Gesture Recognition*, pp.415-420, 2004.

[14] X. Zhang, R. Mersereau, M. Clements and C. Broun, Visual speech feature extraction for improved speech recognition, *ICASSP*, pp.1993-1996, 2002.

[15] Y. Shih and M. Yang, A collaborative virtual environment for situated language learning using VEC3D, *Journal of Educational Technology and Society*, vol.11, no.1, pp.56-68, 2008.

[16] M. Yang, Z. You and Y. Shih, Lip contour extraction for language learning in VEC3D, *Machine Vision and Applications*, vol.21, no.1, pp.33-41, 2009.

[17] X. Liu, Y. Cheung, M. Li and H. Liu, A lip contour extraction method using localized active contour model with automatic parameter selection, *International Conference on Pattern Recognition*, 2010.

[18] S. Wang, W. Lau and S. Leung, Automatic lip contour extraction from color images, *Pattern Recognition*, vol.37, pp.2375-2387, 2004.

[19] U. Saeed and J. Dugelay, Combining edge detection and ergion segmentation for lip contour extraction, *Lecture Notes in Computer Science*, vol.6169, pp.11-20, 2010.

[20] S. Basu, N. Oliver and A. Pentland, 3D lip shapes from video: A combined physical-statistical model, *Speech Communication*, vol.26, pp.131-148, 1998.

[21] A. Gastelum, M. Krueger, J. Marquez, G. Gimel'farb and P. Delmas, Automatic 3D shape segmentation and modelling, *International Conference on Image and Vision Computing*, 2008.

[22] Q. Summerfield, Lipreading and audio-visual speech perception, *Philosophical Transactions: Biological Sciences*, vol.335, pp.71-78, 1992.

[23] D. Cristinacce and T. Cootes, Automatic feature localisation with constrained local models, *Pattern Recognition*, vol.41, pp.3054-3067, 2008.