International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:3, No:4, 2009

Automatic Lip Contour Tracking and Visual Character Recognition for Computerized Lip Reading

Harshit Mehrotra, Gaurav Agrawal and M.C. Srivastava

Abstract—Computerized lip reading has been one of the most actively researched areas of computer vision in recent past because of its crime fighting potential and invariance to acoustic environment. However, several factors like fast speech, bad pronunciation, poor illumination, movement of face, moustaches and beards make lip reading difficult. In present work, we propose a solution for automatic lip contour tracking and recognizing letters of English language spoken by speakers using the information available from lip movements. Level set method is used for tracking lip contour using a contour velocity model and a feature vector of lip movements is then obtained. Character recognition is performed using modified k nearest neighbor algorithm which assigns more weight to nearer neighbors. The proposed system has been found to have accuracy of 73.3% for character recognition with speaker lip movements as the only input and without using any speech recognition system in parallel. The approach used in this work is found to significantly solve the purpose of lip reading when size of database is small.

Keywords—Contour Velocity Model, Lip Contour Tracking, Lip Reading, Visual Character Recognition.

I. Introduction

LIP reading is a complex art of observation to understand speech by visually interpreting clues from lip movements, eyes, gestures and tongue along with any knowledge available from context of discussion and language used. A lip reading system is used along with speech recognition system due to multi-modal nature of speech perception. The phenomenon demonstrating an interaction between hearing and vision in speech perception is known as *McGurk - MacDonald effect* [1]. Lip reading systems have several applications [2] which range from defense applications to use in car navigation systems and also as an alternative to input devices in computer.

One of the most fundamental tasks of any lip reading system is lip contour detection and tracking which requires accurate lip segmentation. Several methods have been proposed in the past for lip segmentation and lip contour detection. These can be classified as follows:
a) *Model based methods* - like active shape models [3]-[5], active appearance models [5, 6] and snakes: active contour models [7]. A model generally describes an object using certain landmark points which can be generated either manually or automatically [8, 9]. An active shape model describes the modes of variation in the shape of an object in terms of the Eigen values and Eigen vectors of a covariance matrix obtained by the difference of each shape in training database from mean shape. An active appearance model is used along with active shape model to describe the variation in gray level of the object using a normalized derivative profile of gray levels near the landmark points. Active contour models are based on minimization of cost functions which may involve computationally expensive algorithms. The performance of model based techniques is not satisfactory when applied on real world images due to little difference between lip and skin color.
b) *Color based methods* [10, 11] aim to increase the discrimination between lip and skin color using a transformation based on RGB color space. These methods are computationally efficient but do not yield satisfactory results on images with weak color contrast.
c) *Level Set based methods* were introduced by Osher and Sethian [12] to capture fronts moving with curvature dependent speeds. The level set represents a closed curve as zero level of a two dimensional embedding function and evolves with curvature dependent speed which is determined by the image content [13, 14]. Lip contour is given by the zero level set when the iterative procedure of level set evolution is complete.

When lip contour is obtained on first frame, next step involved is to track the contour on subsequent frames so as to obtain a numerical representation of lips, since such representations facilitate processing and statistical analysis. Object tracking algorithms largely fall into two categories [15]: -
a) *Filtering and Data Association algorithms* are usually computationally complex and require prior information about the object. They involve use of Kalman filter or Particle filter for object tracking. Y. M. Kim [16] proposed the use of Scale Invariant Feature Transform (SIFT) features [17] and Kalman filter [18] for object tracking using Gaussian distribution to model the motion of the object. In this method, SIFT features of an object are stored which are then used to correct the location of the object predicted by using a Kalman filter. This method is computationally expensive since the SIFT features of an object have to be computed on each frame to perform

Corresponding Author: Harshit Mehrotra is with Department of Electronics and Communication Engineering, Jaypee Institute of Information Technology University, Noida 201 307, U.P., INDIA. *Phone +919871330472, (e-mail: hmehrotra.86@gmail.com, harshit.mehrotra@yahoo.co.in)* .
Gaurav Agrawal is with Department of Electrical Engineering, Indian Institute of Technology Kanpur, Kanpur, INDIA. (email: gagrawal@iitk.ac.in).
M.C. Srivastava is professor at Department of Electronics and Communication Engineering, Jaypee Institute of Information Technology University, (email: mc.srivastava@jiit.ac.in).

a match on next frame.
b) *Target Representation and Localization algorithms* are usually computationally efficient and involve methods like blob detection, optical flow, mean shift tracking, contour tracking and visual feature matching. Ying-li Tian et al. [19] and J. Chen et al. [20] proposed the use of a combination of lip color, shape, motion and structure information of mouth for lip tracking. A multi-state mouth model is used in [19] using different lip templates for each lip state: open, relatively closed and tightly closed. The technique proposed in [20] has the advantage to detect the lip feature points automatically and recover the feature points lost during tracking process. Accuracy of these methods depends on how well the object has been modeled.

Several methods have been proposed in the past for lip reading systems. A standard minimum distance classifier is used in [21, 22] for visual recognition of pre-stored labeled utterances of words. This system is extended to a continuous speech reading system using discrete HMM (Hidden Markov Model) by Goldschen [23]. A HMM similarity metric and a clustering algorithm are then used to determine distinct viseme groups. Pentland and Mase [24] used optical flow to estimate the velocities of left, top, right and bottom points of lip contour in video of mouth movement and then used principal component analysis and standard minimum distance classifier on three and four digit phrases. A probabilistic method of matching for lip reading digits is discussed in [25]. In this method a probabilistic sequence matching function automatically segments a test video and matches the most likely sequence digits recognized in the test sequence.

In this paper we propose a solution for automatic lip contour tracking and for recognizing characters of English language after lip contour is accurately detected on first frame. The method proposed in [14] for lip segmentation and lip contour detection is used in the present work for locating eye corner and center points to obtain an approximate location of lip region. Level set method is then employed to obtain lip contour by minimizing a cost function. The lip contour is enclosed within a rectangle and a velocity model is designed to obtain the position of this rectangle on next frame. The new position of rectangle gives a good approximation to begin the level set method so as to obtain the new lip contour. After lip contour is accurately tracked on all the frames, a feature vector is obtained to numerically represent lip movements for statistical analysis. Feature vectors for all the training videos of characters are stored in a database with properly labeled classes. While testing the system, feature vector of the test sample are obtained during run-time and distance weighted k nearest neighbor algorithm is used to find the best match.

The manuscript has been organized as follows: Lip contour tracking is discussed in Section II. Proposed method for visual character recognition is given in Section III. Experimental results are discussed in Section IV. Major conclusions of the work are presented in Section V followed by an appendix at the end.

II. Lip Contour Tracking

In present work, the input to lip reading system for training and testing purpose consists of a video with unobstructed view of speaker lip movements. Lip contour is first detected on the initial frame using the technique proposed in [14] which involves level set method. A level set may be defined as a set of points where the value of a function is constant. Since the boundary of object is given by zero level set, the value of this constant is taken as zero. The numerical representation of zero level set (Γ) in two dimensions is described by

$$\Gamma = \{(x, y) \mid \phi(t, x, y) = 0\} \tag{1}$$

where ϕ represents a two dimensional embedding function. The rate of change of function ϕ is zero on a level set because the value of ϕ is same at all the points on level set. Hence, a level set evolves in a direction normal to gradient (where rate of change is maximum) which is governed by a partial differential equation known as *Hamilton-Jacobi equation* [12] represented by (2), with V representing the velocity function, determined by the image content.

$$\frac{\partial \phi}{\partial t} + V * | \bigtriangledown \phi| = 0 \tag{2}$$

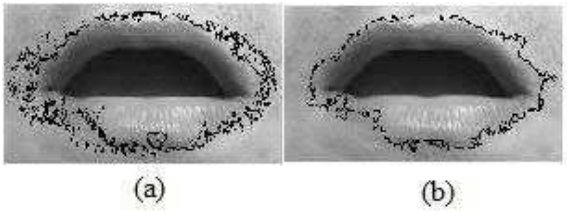

Fig. 1. Lip Contour (a) Isocontours formed during level set evolution superimposed on lip image. (b) Lip contour obtained after removing small contours.

Lip contour obtained by level set method is shown in Fig. 1. It can be observed from Fig. 1(a) that a large number of small contours are formed along with a large contour during level set evolution due to changing topology. All these contours represent points of same height and are called *isocontours*. These small contours are removed to obtain accurate lip contour as shown in Fig. 1(b).

Small contours can be eliminated by identifying the following:
a) A_n representing a set of all those points which lie on n^{th} contour represented by C_n.

$$A_n = \{(x_i, y_j) | (x_i, y_j) \in C_n\} \tag{3}$$

b) L_n representing the length of each contour is then obtained, which is given by cardinality of set A_n, i.e. L_n = $|A_n|$giving the number of elements in a set.

c) All the contours other than one with maximum length are then deleted.

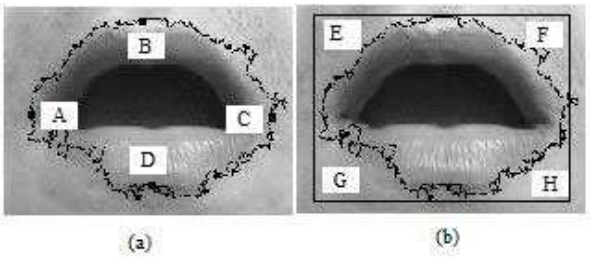

Fig. 2. (a) Four characteristic points obtained on lip contour. (b) A rectangle circumscribing lip contour.

To track lip contour four characteristic points A(X_1,Y_1), B(X_2,Y_2), C(X_3,Y_3) and D(X_4,Y_4) are obtained as shown in Fig. 2(a). These points are chosen such that they completely describe a circumscribing rectangle. The coordinates of these points, with N representing the number of points on the contour, are as follows:

$$X_1 = \min_{i,j=1toN}\{x_i \,|\, (x_i, y_j) \in C\} \tag{4}$$

$$X_3 = \max_{i,j=1toN}\{x_i \,|\, (x_i, y_j) \in C\} \tag{5}$$

$$X_2 = X_4 = \frac{X_1 + X_3}{2} \tag{6}$$

$$Y_1 = \{y_j \,|\, x_i = X_1 \, and \, (x_i, y_j) \in C\} \tag{7}$$

$$Y_3 = \{y_j \,|\, x_i = X_3 \, and \, (x_i, y_j) \in C\} \tag{8}$$

It is observed from (6) that the x coordinates of points B and D are same, but these points lie on opposite side of line joining points A and C. So, for x_i = X_2 there will be two corresponding values of y_j, one less than Y_1 and other greater than Y_1. Therefore,

$$Y_2 = \{y_j \,|\, x_i = X_2 \, and \, (x_i, y_j) \in C \, and \, y_j < Y_1\} \tag{9}$$

$$Y_4 = \{y_j \,|\, x_i = X_4 \, and \, (x_i, y_j) \in C \, and \, y_j > Y_1\} \tag{10}$$

The lip contour so obtained is inscribed within a rectangle as shown in Fig. 2(b) with coordinates of rectangle as E(X_1,Y_2), F(X_3,Y_2), G(X_1,Y_4) and H(X_3,Y_4). These coordinates are obtained from the coordinates of characteristic points chosen on the contour. To obtain lip contour on next frame, position of its circumscribing rectangle is first approximated. It should be noted that the approximate location of circumscribing rectangle may not be same as its actual location. This approximation is done by using a velocity model. It is assumed that the lip contour does not move with more than a maximum velocity which can be different for horizontal and vertical directions. This is due to vertical motion of lips being more significant than horizontal motion. Further, higher the frame rate, lower will be the maximum velocity and vice-versa. This method does not require exact contour velocity because the approximate circumscribing rectangle only gives a starting contour from which exact lip contour is obtained by an iterative procedure.

Let the matrix X_n = [x_1, y_1, x_2, y_2]^T represent the left-top and right-bottom coordinates of circumscribing rectangle on n^{th} frame. Then the corresponding coordinates of approximate circumscribing rectangle on $(n+1)^{th}$ frame represented by X_{n+1} are given by

$$X_{n+1} = \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \end{bmatrix}_{n+1} = \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \end{bmatrix}_{n} + \begin{bmatrix} -V_H \\ -V_V \\ V_H \\ V_V \end{bmatrix} \tag{11}$$

where velocity model V_M = [$-V_H$, $-V_M$, V_H, V_M]^T and V_H and V_V are the maximum velocity of lip contour in horizontal and vertical direction respectively.

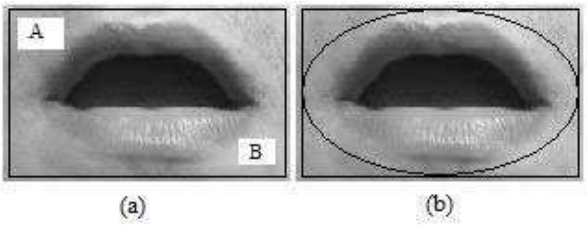

Fig. 3. (a) Approximate circumscribing rectangle of lip contour on $(n+1)^{th}$ frame with coordinates of points A and B as (x_1-V_H,y_1-V_V) and (x_2+V_H,y_2+V_V). (b) Ellipse inscribed within rectangle as starting contour for level set evolution.

Fig.3 (a) shows approximate circumscribing rectangle obtained on next frame. Exact lip contour on this frame is obtained by numerically solving (2) by finite difference method on a Cartesian grid. This is achieved as follows:
a) A Cartesian grid of size $L \times W$ is designed, where L and W are respectively the length and width of rectangle. The x and y coordinates on this grid vary from 1 to L and 1 to W respectively.

b) A 2D embedding function ϕ, described in (1) is initialized such that its zero level set gives a starting lip contour, assumed as an ellipse as shown in Fig. 3(b). The exact lip contour is given by zero level set of ϕ through an iterative procedure. The ellipse with coordinates of center (u_1,v_1) = (L/2,W/2) and a = L/2, b = W/2 representing respectively the length of semi-major and semi-minor axis of ellipse, is described by

$$\frac{(u-u_1)^2}{a^2} + \frac{(v-v_1)^2}{b^2} = 1 \tag{12}$$

Let f(u,v) be a function which is zero for all points (u,v) lying on ellipse. Therefore,

$$f(u,v) = \frac{(u-u_1)^2}{a^2} + \frac{(v-v_1)^2}{b^2} - 1 = 0 \tag{13}$$

$$\phi(t,x,y)|_{t=0} = \frac{(x-x_1)^2}{a^2} + \frac{(y-y_1)^2}{b^2} - 1 \tag{14}$$

It should be noted that $\phi(t,x,y)$ is zero only when point (x,y) lies on an ellipse and not for all points on the Cartesian grid. Fig. 4 gives a 3D plot of ϕ where ϕ(x,y) represents height at that point above xy plane.

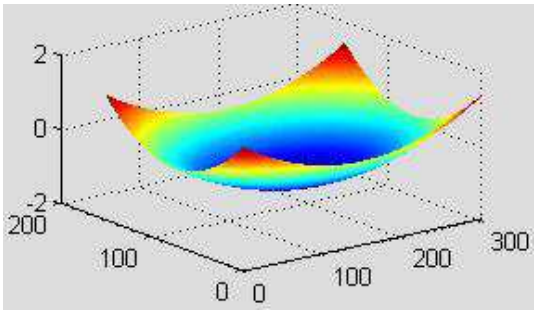

Fig. 4. 3D plot of 2D embedding function ϕ

c) The level set evolves in an iterative way in a direction normal to gradient such that a cost function is minimized. The cost function with G_σ representing a Gaussian kernel of standard deviation σ and I representing segmented lip image, may be taken as

$$g = 1/(1 + |\bigtriangledown G_\sigma * I|^2) \tag{15}$$

This cost function will be minimized as it is inversely proportional to gradient and acts as an edge stopping function with gradient being maximum at object edges.

d) The value of embedding function ϕ_{t+1} at any instant (t+1) may be expressed as

$$\phi_{t+1} = \phi_t + l\frac{d\phi}{dt} \tag{16}$$

where ϕ_t is value at instant t and *l* is the gradient descent step size. The velocity function described by (2) can be expressed in terms of *curvature* κ as

$$\frac{\partial\phi}{\partial t} = g\kappa|\bigtriangledown\phi| + g|\bigtriangledown\phi| \tag{17}$$

where κ may be expressed [26] as

$$\kappa = \bigtriangledown.\frac{\bigtriangledown\phi}{|\bigtriangledown\phi|} = \frac{\phi_{xx}\phi_y^2 - 2\phi_x\phi_y\phi_{xy} + \phi_{yy}\phi_x^2}{(\phi_x^2+\phi_y^2)^{\frac{3}{2}}} \tag{18}$$

The proof of (17) is given in Appendix. It should be noted that *t* in above equations is *artificial time* or *virtual time* denoting process evolution and not actual time elapsed in the process. A small change $d\phi$ with respect to small time interval dt can be expressed as

$$\frac{d\phi}{dt} = g\kappa|\bigtriangledown\phi| + g|\bigtriangledown\phi| + \frac{\partial g}{\partial x}\frac{\partial\phi}{\partial x} + \frac{\partial g}{\partial y}\frac{\partial\phi}{\partial y} \tag{19}$$

As the function ϕ evolves with time, the shape of its zero level set changes from an ellipse (shown in Fig. 3(b)) to exact lip contour. However, some numerical instabilities like very sharp or flat shapes may occur that may lead to computational inaccuracies and hence improper output. A traditional way to avoid this problem is to use re-initialization function so as to reshape ϕ periodically after a small number of iterations. Sussman, Smereka and Osher [27] expressed the re-initialization function as

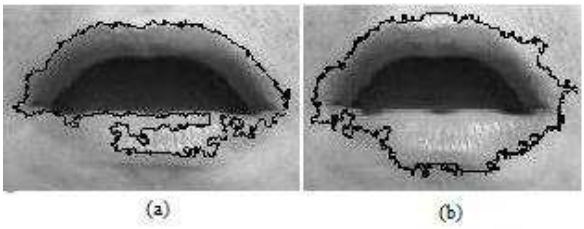

Fig. 5. Output lip contour obtained (a) without re-initialization (b) with re-initialization.

$$\frac{\partial\phi}{\partial t} = S(\phi_0)(1 - |\bigtriangledown\phi|) \tag{20}$$

where $S(\phi_0)$, a sign function using initial value of ϕ (denoted by ϕ_0) may be expressed as

$$S(\phi_0) = \frac{\phi_0}{\sqrt{\phi_0^2 + (\Delta x)^2}} \tag{21}$$

Later, Peng et al. [28] suggested S(ϕ) as

$$S(\phi) = \frac{\phi}{\sqrt{\phi^2 + (\Delta x)^2|\bigtriangledown\phi|^2}} \tag{22}$$

It is a better choice especially when ϕ_0 is a poor estimate of signed distance, i.e., when $|\bigtriangledown\phi_0|$is far away from 1. Fig. 5 shows comparitive results for lip contour obtained when (a) no re-initialization function is used (b) when lip contour is periodically reinitialized during evolution. It can be observed that periodical re-initialization of ϕ removes numerical instabilities and results in proper output. Fig. 6 shows a 3D plot of ϕ (a) before re-initialization and (b) after re-initialization.

Evolution of ϕ automatically stops when its zero level set reaches lip boundary because cost function is then minimized. This stopping condition can be computationally determined when the change in contour between two consecutive iterations is less than convergence criterion. In present work, this is taken as $|\phi_{t+1}|$- $|\phi_t| \leq$ 0.05. When lip contour is accurately obtained on this frame, its characteristic points and circumscribing rectangle are obtained and the same procedure is followed to obtain lip contour on next frame.

III. Visual Character Recognition

Visual character recognition is classified as a type of pattern recognition problem which requires statistical feature vector describing lip movements to determine the character spoken by speaker. In the present work, we propose a solution to this problem using distance weighted k-nearest neighbor (k-NN) algorithm using memory based learning. This is the simplest type of all machine learning algorithms as it does not need to learn a global model. Moreover, there is no explicit training

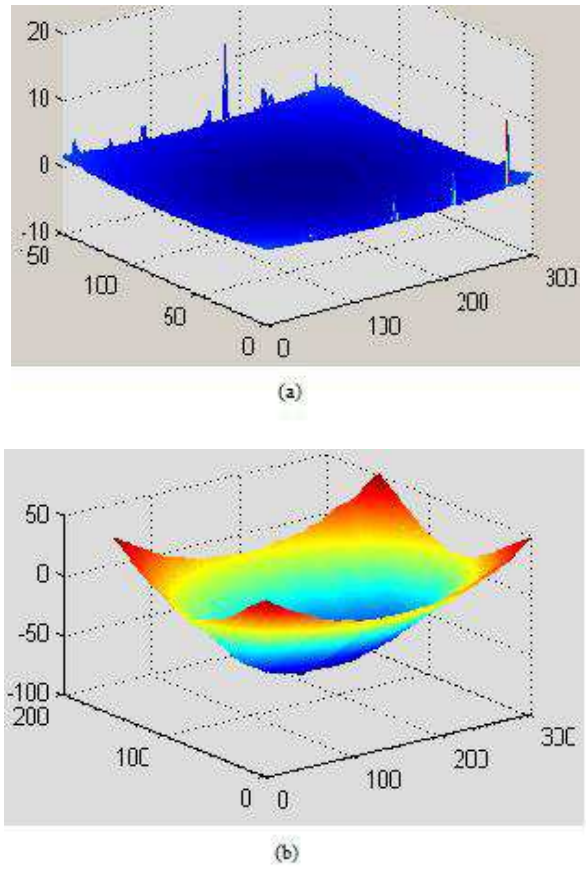

Fig. 6. 3D plot of ϕ (a) before re-initialization and (b) after re-initialization.

step required. The training phase of the algorithm only consists of storing the feature vectors and class labels of the training samples in the database. The feature vector of test video is obtained during run time and the database is searched to find the best match.

A. Feature Vector

Lip reading requires a numerical representation of lips to facilitate processing and statistical analysis. Feature vector in general is an n-dimensional vector of numerical features that represent some object. Once lip contour is obtained on a frame, position of eight reflective markers is calculated on the contour on each frame of the video as shown in Fig. 7. Points 1, 3, 5 and 7 are the four characteristic points on the lip contour as obtained in sec. 2. Remaining four reflective markers are chosen as the mid points of two neighboring reflective markers. These reflective markers are transformed into four component feature vector. A feature vector thus signifies the distance between reflective markers. These markers are so chosen because the feature vector obtained can explain all possible variations in lip movements for lipreading: Second component of feature vector (FV2) gives vertical separation between markers on upper and lower lip and represents mouth opening, FV4 gives horizontal separation between left and right corner points on lips, FV1 gives vertical separation between markers in north-west and south-west quadrants and FV3 is analogous for this distance in north-east and south-east quadrants. Feature vector on each frame is divided by length of approximate circumscribing rectangle (shown in Fig. 3(a)) so as to avoid the effect of distance of speaker from camera. Feature vector is calculated on each frame for all the training videos and is stored in the database with properly labeled class.

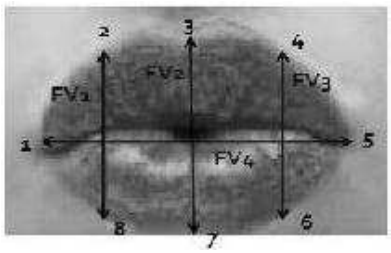

Fig. 7. Reflective markers and Feature Vectors on lips.

B. k Nearest Neighbor

k-NN [29] is a type of memory based learning in which an object is classified based on the number of votes of its neighbors. The object is assigned to the class most common amongst its k nearest neighbors, where k is a positive integer. The training phase of the algorithm consists of storing the feature vectors and class labels of the training samples. Feature vector of the test sample are calculated in run-time. Euclidean distance from the new vector to all stored vectors are computed using (23) and k closest samples are selected. k-NN in general can use any other distance metric like *Manhattan distance* or *Hamming distance*.

$$Dist(C_1, C_2) = \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{4}(FV_{ij}(C_1) - FV_{ij}(C_2))} \quad (23)$$

where C_1 and C_2 are respectively the test sample and training sample, N represents the total number of frames in a video, FV_{ij} represents the j^{th} component of feature vector on i^{th} frame. The test sample is assigned the class with highest frequency amongst k nearest neighbors. K must be an odd number so as to avoid voting ties. Large value of k is used when the training data is large and gives better probability estimates.

In Fig. 8, if k = 3 test sample (?) is classified to the second class (B) because there are 2 B's and only 1 A inside the inner circle. If k = 5 it is classified to first class (3 A's vs. 2 B's inside the outer circle). A major drawback of this technique is that the prediction of new vector is dominated by classes with more frequent examples because more samples of these classes tend to come up in the k nearest neighbors. A solution to this problem is to use *distance weighted k nearest neighbor.* This takes into account the distance of each k nearest neighbors while predicting the class of new vector.

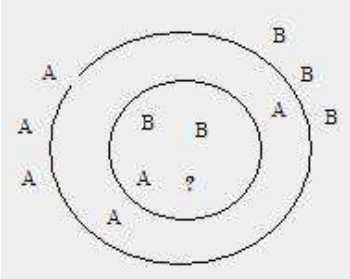

Fig. 8. Example of k-NN classification with test sample (?) classified either to the first class A or to the second class B.

C. Distance Weighted k Nearest Neighbor

This approach is used to predict the class of a test sample by assigning more weight to nearer neighbor. The weight given to a training sample is inversely proportional to the Euclidean Distance between the training sample and test case.

$$W_k = \frac{1}{Dist(C_k, C_{test})} \tag{24}$$

Class labels of training samples are then arranged in decreasing order of weight and k highest weighted samples are selected. The total weight assigned to a class is equal to sum of weights of all samples belonging to that class present in the k samples selected as shown in Fig. 9. An advantage of using this approach is that it assigns more weight to nearer neighbors. However, large training sets require lot of memory and hence run time costs scale with training data size.

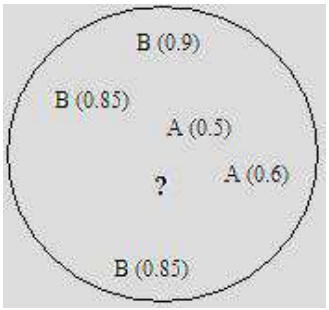

Fig. 9. Example of distance weighted k-NN classification where numbers in bracket denote Euclidean distance from test sample.

The test sample (?) should be classified either to the first class A or to the second class B. When k = 5 it is classified to the second class (B) using simple k-NN classification. Using weighted k-NN, weight of class A = 1/0.5 + 1/0.6 = 3.66, weight of class B = 1/0.8 + 1/0.85 + 1/0.9 = 3.537. So, test sample (?) is assigned to class A because of higher weight.

IV. Results and Discussion

Lip reading system proposed in this paper has been implemented in MATLAB Ver. 7.0 on Intel Core 2 Duo CPU @2.53 GHz and 2 GB RAM. Videos of size 320x240 have been captured for training and testing the system using 2.0 megapixel autofocus Logitech Quickcam. Each video is captured for duration of two seconds at a frame rate of 15 frames/sec with JPEG image quality of 90% (i.e. 10% compression).

Fig. 10 shows the segmented lip region with initializing ellipse and exact lip contour obtained for two different mouth states: closed and wide open. It is observed from this figure that lip contour may be accurately obtained for both the extreme mouth states using this approach. Fig. 11 shows various stages of contour evolution. It can be noticed from the figure that the curve evolves from an ellipse and stops at lip boundary to give exact lip contour.

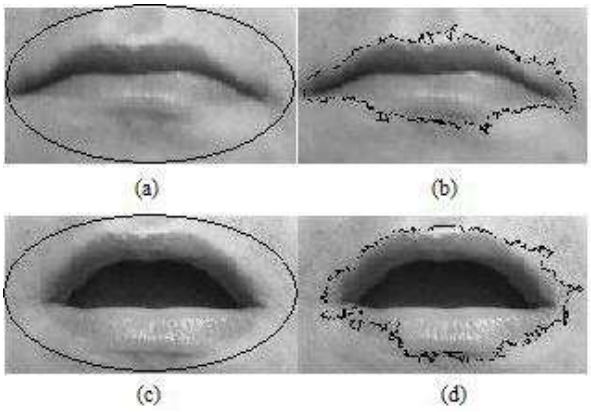

Fig. 10. Lip segmentation and contour detection: (a) and (c) show segmented lip region and initializing ellipse when mouth is closed and open, (b) and (d) show lip contour obtained in the two cases.

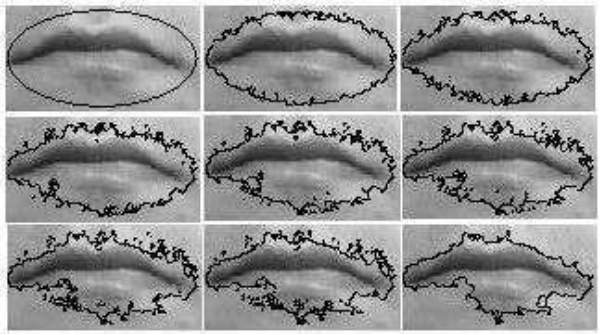

Fig. 11. Lip contour at various stages of evolution.

The step size (l) should be initialized to a small value so that the contour does not jump lip boundary. Similarly, the function ϕ should be re-initialized after small number of iterations so that numerical instabilities do not continue further. Experiments for lip contour detection have been performed for different values of step size and number of iterations before re-initialization. We observed from the output obtained that lip contour deteriorates significantly with increasing step size and number of iterations before re-initialization as expected. Fig. 12 gives lip contour obtained for various step sizes. Effect of number of iterations before re-initialization can be observed from Fig. 13.

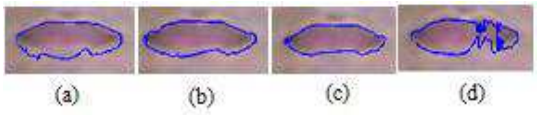

Fig. 12. Lip contour obtained for various step sizes (a) 0.5 (b) 1 (c) 2 (d) 3.

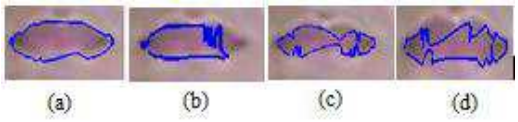

Fig. 13. Lip contour obtained for step size 1 and number of iterations (a) 10 (b) 20 (c) 50 (d) 100.

In present work, step size is taken as 1 and the surface is re-initialized after every 10 iterations. Standard deviation σ of Gaussian kernel is taken as 1.2 for smoothing image. Lip contour maximum velocity in horizontal and vertical direction is chosen as $V_H = 5pixels/frame$ and $V_V = 10pixels/frame$ respectively. Therefore, velocity model obtained is V_M = [-5 -10 5 10]^T. Fig. 14 shows lip contour obtained on consecutive frames of a video.

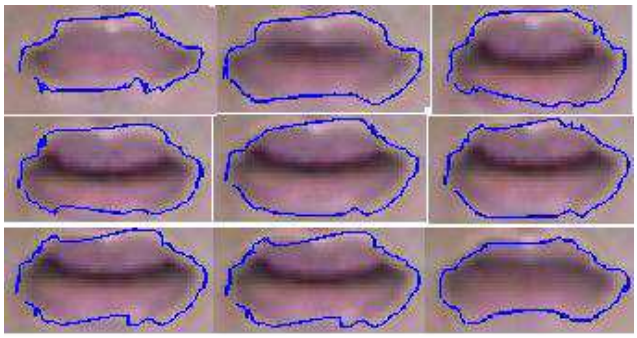

Fig. 14. Lip contour obtained on nine consecutive frames of a video.

Feature vectors are obtained on all frames for each training video and are stored in the database. The training set consists of five examples of each character spoken by different speakers (three males and two females). Fig. 15 shows a graphical representation of feature vectors of vowels. It can be observed from this figure that variation in feature vector of a character is similar even for two different speakers. Moreover, feature vectors of two different characters pronounced by same speaker differ significantly. For example: FV4 of character 'A' remains constant for some time and then increases slightly towards the end. On the other hand, FV4 of character 'O' remains constant throughout because there is very little or no vertical motion of lips while pronouncing 'O'. Similarly, FV1, FV2 and FV3 of character 'A' remain constant in the beginning, increase in the middle and finally decrease in the end. Therefore, we conclude that reflective markers (shown in Fig. 7) chosen in this work correctly represent variation in feature vector of characters and can distinguish between characters.

Experiments for visual character recognition have been performed for various values of k for three different speakers (two males and one female). Results of these experiments are given in Table 1-5 with numbers in bracket denoting the error occurred. The error computed as inverse of weight assigned to a class (as explained in section 3.3) may be represented as

$$Error = \frac{1}{Weight_{class}} = \frac{1}{\sum \frac{1}{Dist}} \tag{25}$$

It is observed from these tables that error occuring during character recognition decreases as k increases. Moreover, using higher value of k makes system less succeptible to noise. Therefore, higher value of k should be chosen to minimize error. In present work, k is taken as 9 and maximum acceptable error limit is set to 0.3. Although tables with k = 5, 7 and 9 provide same accuracy, k = 9 is preferred to provide least error for character recognition. Fig. 16 shows graphical representation of feature vector of test sample (A) and best matching character.

TABLE I
VISUAL CHARACTER RECOGNITION WITH K = 1

Test	Speaker 1	Speaker 2	Speaker 3
A	A (0.3845)	A (0.5498)	A (0.7233)
E	E (0.6269)	**A (0.5928)**	E (0.6161)
I	**A (0.3998)**	**E (0.5636)**	I (0.6908)
O	**U (0.3697)**	**E (0.5643)**	**U (1.1407)**
U	U (0.3551)	**E (0.6821)**	**E (0.6732)**

TABLE II
VISUAL CHARACTER RECOGNITION WITH K = 3

Test	Speaker 1	Speaker 2	Speaker 3
A	A (0.2082)	**E (0.3403)**	**U (0.4152)**
E	E (0.2176)	**A (0.5928)**	E (0.3352)
I	**A (0.2157)**	**E (0.5636)**	I (0.3567)
O	**U (0.2059)**	**E (0.5643)**	**U (1.1407)**
U	U (0.2163)	**E (0.6821)**	**E (0.6732)**

TABLE III
VISUAL CHARACTER RECOGNITION WITH K = 5

Test	Speaker 1	Speaker 2	Speaker 3
A	A (0.1522)	A (0.3108)	A (0.4105)
E	E (0.2176)	E (0.3455)	E (0.2456)
I	**A (0.2157)**	**E (0.3318)**	I (0.3587)
O	O (0.2059)	O (0.3740)	**U (0.6024)**
U	U (0.2163)	**E (0.4135)**	U (0.3869)

Accuracy(%) = $\frac{CorrectMatch}{Totalnumberof Samples} \times 100$
For k = 1: Accuracy = $\frac{7}{15} \times 100$ = 46.66%.
For k = 3: Accuracy = $\frac{5}{15} \times 100$ = 33.33%.
For k = 5: Accuracy = $\frac{11}{15} \times 100$ = 73.33%.
For k = 7: Accuracy = $\frac{11}{15} \times 100$ = 73.33%.
For k = 9: Accuracy = $\frac{11}{15} \times 100$ = 73.33%.

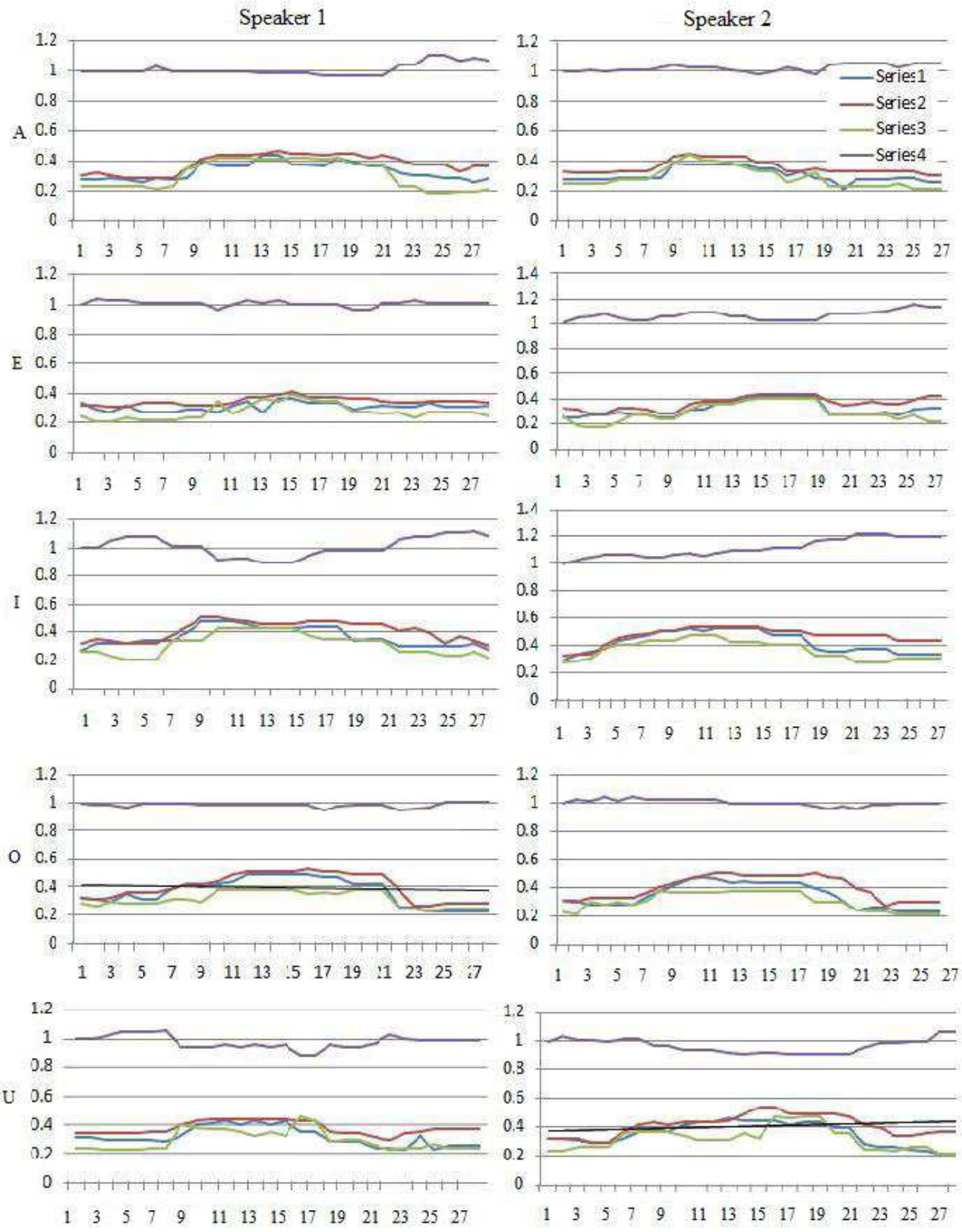

Fig. 15. Plot of normalized feature vectors obtained on each frame for vowels pronounced by one male and one female (Series 1, 2, 3 and 4 represent FV1, 2 , 3 and 4 respectively).

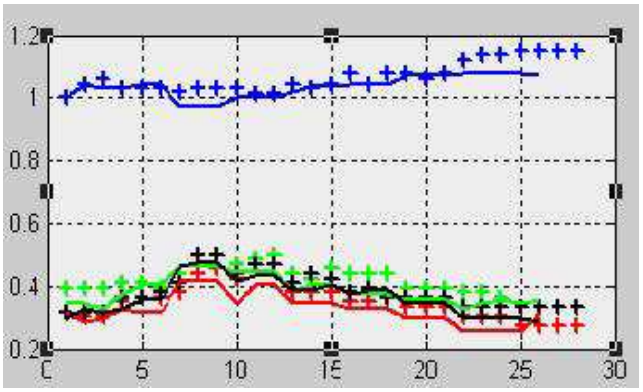

Fig. 16. Graphical representation of feature vector of test sample (A) represented by straight line and best matching character represented by '+'.

TABLE IV
VISUAL CHARACTER RECOGNITION WITH K = 7

Test	Speaker 1	Speaker 2	Speaker 3
A	A (0.1522)	A (0.2182)	A (0.4105)
E	E (0.2176)	E (0.2480)	E (0.2456)
I	**A (0.1558)**	**E (0.3318)**	I (0.3567)
O	O (0.1553)	O (0.3252)	**U (0.4154)**
U	U (0.1547)	**E (0.4135)**	U (0.3724)

TABLE V
VISUAL CHARACTER RECOGNITION WITH K = 9

Test	Speaker 1	Speaker 2	Speaker 3
A	A (0.1522)	A (0.2182)	A (0.2911)
E	E (0.2176)	E (0.2480)	E (0.2456)
I	**A (0.1558)**	**A (0.2652)**	I (0.2819)
O	O (0.1553)	O (0.2526)	**U (0.4154)**
U	U (0.1547)	**E (0.3071)**	U (0.2673)

V. CONCLUSION

In this paper, we have proposed a solution for automatic lip contour tracking using level set evolution with re-initialization function after lip contour is accurately obtained on initial frame. Visual character recognition is performed using distance weighted k nearest neighbor algorithm. In this method, more training examples of a class do not affect the output of the system. Experimental results show that the feature vectors obtained in this work can distinguish between characters. Moreover, the method proposed for character recognition will significantly solve the purpose of lip reading when the size of database is small. This method can even be employed for recognition of words as long as the database is small. However, as the size of database increases, artificial neural network can be used with greater computational efficiency. The accuracy of this system can be significantly improved by using a speech recognition system in parallel.

APPENDIX

The basic idea behind level set method is to use surface evolution to track motion of a planar curve rather than tracking the curve movement itself. The velocity function (described by (2)) governing evolution of ϕ is designed from image data such that the curve stops at lip boundary. For this purpose, the velocity function is divided into two components as V = V_A + V_G [30]. The first component V_A, known as *advection term*, is independent of geometry of curve. The second term V_G depends on the geometry of level set function ϕ. To obtain this component, we consider motion of curve only in self-generated velocity field [31]. Eq. 2 can be represented in vector form as

$$\frac{\partial \phi}{\partial t} + \vec{V_G}. \bigtriangledown \phi = 0 \tag{A.1}$$

Component V_G can further be expressed as sum of two components, i.e., $\vec{V_G} = V_n\vec{N}+V_t\vec{T}$, where V_n and V_t are components of velocity in normal and tangential direction respectively. Since $\vec{N}$ and $\bigtriangledown\phi$ point in the same direction, the tangential component $\vec{T}.\bigtriangledown\phi$ becomes zero. Substituting $\vec{N} = \bigtriangledown\phi/| \bigtriangledown \phi|$ gives equation of curve evolution in self-generated velocity field as

$$\frac{\partial \phi}{\partial t} + V_n| \bigtriangledown \phi| = 0 \tag{A.2}$$

We consider the motion of curve by mean curvature in self-generated velocity field. In this case, the velocity of curve in normal direction is proportional to its curvature [31] and is assumed as $\vec{V_n} = -a\kappa\vec{N}$ where κ denotes curvature and *a* is a constant. It should be noted that the curve shrinks when $a > 0$ and expands when $a < 0$.

The velocity function including both *advection term* and *self-generated field* is represented as $V = V_A - a\kappa$. This velocity function is multiplied by cost function g (described in (15)). This is done so that contour velocity becomes close to zero in regions of high image gradient. In regions of constant intensity, cost function is close to unity thereby not affecting the velocity function. Substituting these values in (2), we obtain

$$\frac{\partial \phi}{\partial t} = (-V_A + a\kappa)g| \bigtriangledown \phi| \tag{A.3}$$

In present work, the variables V_A and *a* are initialized as -1 and 1 respectively so as to obtain the velocity function proposed in [30] for a shrinking surface. The evolution equation now reduces to the form

$$\frac{\partial \phi}{\partial t} = (1 + \kappa)g| \bigtriangledown \phi| = g\kappa| \bigtriangledown \phi| + g| \bigtriangledown \phi| \tag{A.4}$$

The above equation can be substituted in expression of dϕ/dt to obtain (19). This eq. helps us to obtain the change in evolving surface.

REFERENCES

[1] H. McGurk, J. MacDonald, "Hearing lips and seeing voices," J. Nature, vol. 264(5588), 1976, pp 746-748.
[2] W.C. Yau. (2009). Computer-based lip-reading using motion templates [Online]. Available: http://www.ieeevic.org/events/getdetails.php?id=234(URL)
[3] T.F. Cootes, A. Hill, C.J. Taylor, J. Haslam, "The use of active shape models for locating structures in medical images," J. Image Vis. Comput. vol. 12 Issue 6, 1994, pp 355-366.
[4] T.F. Cootes, C.J. Taylor, D.H. Cooper, J. Graham, "Active shape models-Their training and application," J. Comput. Vis. Image Underst. vol 61 Issue 1, 1995, pp 38-59.
[5] I. Matthews, T.F. Cootes, J.A. Bangham, S. Cox, R. Harvey, "Extraction of visual features for lipreading," IEEE Trans. Pattern Anal. Mach. Intell. vol. 24 Issue 2, pp 198-213, 2002.
[6] T.F. Cootes, G.J. Edwards, C.J. Taylor, "Active appearance models," in Proc. European Conf. Comput. Vis., June 1998, pp. 484-498.
[7] M. Kass, A. Witkin, D. Terzopoulos, "Snakes: Active contour model," Int. J. Comput. Vis. vol. 1, 1987, pp 321-331.
[8] A. Souza, J.K. Udupa, "Automatic landmark selection for active shape models- Medical imaging: Image Processing", in Proc. of the SPIE, Vol. 5747,2005, pp. 1377-1383.
[9] K. Domijan, S. Wilson, "A Bayseian method for automatic landmark detection in segmented images," in Proc. of the workshop on Mach. Learning Techniques for Processing Multimedia Content, Bonn Germany, 2005.
[10] N. Eveno, A. Caplier, P.Y. Coulon, "New color transformation for lips segmentation," in Proc. IEEE 4th Workshop Multimedia Signal Proc., France, 2001, pp. 3-8.
[11] T. Wark, S. Sridharan, V. Chandran, "An approach to statistical lip modeling for speaker identification via Chromatic Feature Extraction," in Proc. 4th Intl. Conf. Pattern Recognition., Brisbane, Australia, 1998, pp. 123-125.
[12] S. Osher, J.A. Sethian, "Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations," J. Comput. Phys. vol. 79 Issue 1, 1988, 12-49.
[13] A. Sayeed Md. Sohail, P. Bhattacharya, "Automated lip contour detection using the level set segmentation method," Int. Conf. on Image Anal. and Proc. (ICIAP 2007), pp. 425-430.
[14] H. Mehrotra, G. Agrawal, M.C. Srivastava, "Automatic lip contour extraction using level set evolution, 3rd Int. Conf. on Information Proc., (ICIP 2009) Bangalore, to be published.
[15] Wikipedia [Online], Available: http://en.wikipedia.org/wiki/Video_tracking (URL)
[16] Y.M. Kim, Object tracking in video sequence,[Online], Available: http://www.stanford.edu/∼jinhae/CS229_report.pdf(URL).
[17] D.G. Lowe, "Distinctive image features from scale-invariant keypoints," Int. J. Comput. Vis. vol.60 Issue 2, 2004, pp 91-110.
[18] G. Welch, G. Bishop, "An introduction to Kalman filter," Technical report: TR95-041, July 24, 2006. [Online], Available: http://www.cs.unc.edu/∼welch/media/pdf/kalman_intro.pdf(URL).
[19] Y. Tian, T. Kanade, J.F. Cohn, "Robust lip tracking by combining shape, color and motion," in Proc. of the 4th Asian Conf. on Comput. Vis. (ACCV'00), Jan, 2000, pp. 1040-1045.
[20] J. Chen, Y. Laprie, M.O. Berger, "A robust lip tracking system for acoustic to articulatory inversion," The 6th IASTED Int. Conf. on Signal and Image Proc., August 2004, USA.
[21] E.D. Petajan, "Automatic lipreading to enhance speech recognition," in Proc. of the IEEE Communication Society Global Telecommunications Conference", November 26-29, 1984, Atlanta, Georgia.
[22] E.D. Petajan, B. Bischoff, D. Bodoff, "An improved automatic lipreading system to enhance speech recognition," ACM SIGCHI-88, 19-25 (1988).
[23] A.J. Goldschen, O.N. Garcia, E. Petajan, "Continuous optical automatic speech recognition by lipreading," 28th Annual Asilmomar Conference on Signals, Systems, and Computer, 1994.
[24] A. Pentland, K. Mase, "Lip reading: Automatic visual recognition of spoken words," in Proc. Image Understanding and Mach. Vis., Optical Society of America, June 12-14 (1989).
[25] S. Pachoud, S. Gong, A. Cavallaro, "Macro-cuboid based probabilistic matching for lip-reading digits," IEEE Computer Society Conf. on Comput. Vis. and Pattern Recognition (CVPR), USA, June 2008
[26] Y. Qu, P.A. Heng, T.T. Wong, "Image segmentation using the level set method," in: Deformable Models II: Theory and Biomaterial Applications, J.S. Suri, A. Farag, Ed, Springer, 2007,pp. 95-122.
[27] M. Sussman, P. Smereka, S. Osher, "A level set approach for computing solutions to incompressible two-phase flow," J. Comput. Phys. vol 114, 1994, pp 146-159.
[28] D. Peng, B. Merriman, S. Osher, H. Zhao, M. Kang, "A PDE based fast local level set method," J. Comp. Phys. Vol. 155, 1999, pp. 410-438.
[29] Wikipedia Online, Available: http://en.wikipedia.org/wiki/K-nearest_neighbor_algorithm(URL).
[30] R. Malladi, J.A. Sethian, B.C. Vemuri, "Shape modeling with front propagation: A level set approach," IEEE Trans on Pattern Anal. and Mach. Intell., Vol. 17, 1995, pp. 158-175.
[31] S. Osher, R.P. Fedkiw, "Motion involving mean curvature," in: Level set methods and dynamic implicit surfaces, Springer, 2002, pp. 41-46.