VISUAL SPEECH RECOGNITION VIA LIP CONTOUR TRACKING AND HAUSDORFF DISTANCE BASED MATCHING

A DISSERTATION

Submitted in partial fulfillment of the requirements for the award of the degree of

INTEGRATED DUAL DEGREE

(Bachelor of Technology & Master of Technology)

in

ELECTRONICS AND COMMUNICATION ENGINEERING

(With Specialization in Wireless Communication)

By

ABHISHEK YADAV

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE - 247 667 (INDIA)

JUNE, 2011

CANDIDATE'S DECLARATION

I hereby declare that the work, presented in this dissertation report, entitled **"VISUAL SPEECH RECOGNITION VIA LIP CONTOUR TRACKING AND HAUSDORFF DISTANCE BASED MATCHING,"** being submitted in fulfillment of partial requirements for the award of the degree of Integrated Dual Degree (Bachelor of Technology and Master of Technology) in Electronics and Communication Engineering with Specialization in Wireless Communication, in the Department of Electronics and Computer Engineering, Indian Institute of Technology, Roorkee is my original work. The results submitted in this dissertation report have not been submitted for the award of any other Degree or Diploma.

Date:

Place:

(Abhishek Yadav)

IDD ECW- V year
Enrolment # 061201
ECE Deptt.
IIT Roorkee

CERTIFICATE

This is to certify that the statement made by the candidate is correct to the best of my knowledge and belief. This is to certify that this dissertation entitled, **"VISUAL SPEECH RECOGNITION VIA LIP CONTOUR TRACKING AND HAUSDORFF DISTANCE BASED MATCHING,"** is an authentic record of candidate's own work carried out by him under my guidance and supervision. He has not submitted it for the award of any other degree.

Date:

Place:

(Dr. Debashis Ghosh)
Associate Professor,
ECE Deptt., IIT Roorkee

ACKNOWLEDGEMENTS

First of all, I would like to express my deep sense of respect and gratitude towards my guide **Prof. Debashis Ghosh**, who has been the guiding force behind this work. I am greatly indebted to him for his constant encouragement and invaluable advice in every aspect of my academic life. I consider it my good fortune to have got an opportunity to work with such a wonderful person.

I would like to thank all faculty members and staff of the Department of Electronics and Computer Engineering, IIT Roorkee for their generous help in various ways for the completion of this thesis.

I am greatly indebted to all my batch mates, who have helped me with ample moral support and valuable suggestions. Most of all I would like to thank my family. Finally, I would like to extend my gratitude to all those persons who directly or indirectly contributed towards this work.

ABHISHEK YADAV

ABSTRACT

Frequently there are instances where speech is corrupted by the surrounding noises. This puts a limitation on the efficiency of an audio-only speech recognition system. This drives us to think of alternative sources of information from the speaker. Lip movements satisfy the condition of being a source of what the speaker said as well as inertness from the environmental noises. Also, visual information extraction for visual speech recognition would be required in case the user is unable to speak.

This thesis attempts to achieve the task of visual speech recognition. First motivation came from making the process of lip reading completely automatic. As a result, there comes a need to first of all identify the skin region of the person in the video. Once we are able to accomplish that, we get the face boundary from where we attempt to obtain the location of eyes of the speaker. The coordinates of eyes leads us to a region between the nose and the upper lip. This leads us to the implementation of the process called lip-tracking, giving us the lip shapes in all the frames that comprise a video. Once we are at the stage of comparing videos, we attempt to extract key frames from a video since that helps in reducing the amount of calculations which would have to be accomplished otherwise when comparing all the frames of the videos. After extracting the key frames, we compare the set of key frames of different videos using a modified form of Hausdorff distance.

Table of Contents

List of Figures

Chapter 1 Introduction

1.1 Speech Recognition

Automatic recognition of speech by machine has been a goal of research for more than four decades. However, in spite of the glamour of designing an intelligent machine that can recognize the spoken word and comprehend its meaning, and in spite of the enormous research efforts spent in trying to create such a machine, we are far from achieving the desired goal of a machine that can understand spoken discourse on any subject by all speakers in all environments[19].

The earliest attempts to devise systems for automatic speech recognition by machine were made in 1950s, when various researchers tried to exploit the fundamental ideas of acoustic-phonetics. In the 1960s, several fundamental ideas in speech recognition surfaced and were published. However, the decade started with several Japanese laboratories entering the recognition arena and building special-purpose hardware as part of their systems. In the 1970s, speech-recognition research achieved a number of significant milestones. The area of isolated word or discrete utterance recognition became a viable and usable technology. Another milestone of the 1970s was the beginning of a longstanding, highly successful group effort in large vocabulary speech recognition at IBM in which researchers studied three distinct tasks over a period of almost two decades.

Just as isolated word recognition was a key focus of research in the 1970s, the problem of connected word recognition was a focus of research in the 1980s. Here the goal was to create a robust system capable of recognizing a fluently spoken string of words based on matching a concatenated pattern of individual words.

Speech research in the 1980s was characterized by a shift in technology from template-based approaches to statistical modelling methods – especially the hidden Markov model approach. Although the methodology of hidden Markov modelling (HMM) was well known and understood in a few laboratories, it was not until wide spread publication of the methods and theory of HMMs, in the mid-1980s, that the technique became widely applied in virtually every speech-recognition research laboratory in the world.

Another "new" technology that was reintroduced in the late 1980s was the idea of applying neural networks to problems in speech recognition. Neural networks were first introduced in the 1950s, but they did not prove useful initially because they had many practical problems. In the 1980s, however, a deeper understanding of the strengths and limitations of the technology was obtained, as well as the relationships of the technology to classical signal classification methods.

Finally, the 1980s was a decade in which a major impetus was given to large vocabulary, continuous-speech–recognition systems.

1.2 The Visual Component in Speech Recognition

There are cases where the environment in which a speaker says something in noisy or there is hearing impairment and, in such conditions, the audio speech recognition system would not perform as expected. Knowing that face movements also provide cues, precisely visemes, extracting such information from the speaker can improve the recognition efficiency of the system.

According to [16], the benefit gained from the visual, facial cues has been quantitatively estimated to be equivalent to an increase of 8-10 dB in the signal-to-noise ratio when speech sentences are presented in a noise background [17]. This observation suggests that, if the acoustic inputs to conventional speech recognition systems could be augmented by data about the visible speech gestures, an enhanced-performance, audio-visual recognition system should be possible. Indeed, one of the challenges of speech technology is to be able to provide robust and accurate automatic systems capable of operating successfully in a wide range of environments, including those where high levels of noise and vibration may be encountered. Aircraft cockpits are one example of a demanding environment in which reliable automatic speech recognition is becoming an important requirement.

Thus, incorporating visual information provides a solution to improving the recognition procedure.

But accomplishing such a task would involve many difficulties. As according to [18], there are however many hurdles that must be overcome before commercial audio-visual speech recognition systems will become a reality. Such systems must be capable of tracking the lips (inner or outer contour, or both) and reasoning about the presence/absence and position of the teeth and tongue on unconstrained speakers who may be moving around and nodding or rotating their heads. Such systems should also be robust to variations in lighting and shadowing. Furthermore, in order to provide accurate recognition, they must yield visual features capable of discriminating among the various recognition Units (words, phonemes, tri-phones). The extracted visual features must also be intelligently integrated with the acoustic features, presumably in proportion to the information content of each channel. And of course, all of this should be accomplished in real-time or near real-time in order that the users of such systems are not unduly put off.

First of all, it is the face of the speaker which needs to be extracted before we could go for extracting visual cues from the video.

Two traditional ways for face detection are the geometric, feature-based and the template matching. These techniques are computationally very demanding and cannot handle large variations in face images [1].

The technique which has been used here does not have such computational demands. It focuses on extracting the skin pixels of a frame in a video. For accomplishing this, we first need to choose a color space in which we would be doing all our processing. Details of choosing a color space has been discussed in the 2nd Chapter of this thesis.

Once we know the boundaries of the speaker's face, we attempt lip tracking of the video after locating the mouth region. The contours of the lip shape for each frame form the representations of different visemes. For the recognition part, key frames extracted from these contours are used for calculating a similarity measure called Hausdorff distance.

1.3 Motivation

In acoustically noisy environments, it would be relatively very difficult for the audio-only speech recognition system to perform at par with the efficiency for which it was designed. This leads us to think of other alternatives which could serve as information source while the speaker speaks together with the condition that the information source to be relied on should not get affected with acoustic noise.

One of the candidates that satisfy these conditions is the **lip movement.** Moreover, in situations where the person has a disability of not being able to speak, lip movements provide the basis of extracting information from the speaker. This served as the motivation for the work presented in this thesis.

1.4 Thesis Organization

First part of Chapter 2 deals with the task of face detection/location of the speaker. Not knowing where the speaker might be in the video which we need to analyze, there arise a need to get the boundaries of the speaker's face which precedes all other processing required for lip extraction. The task is accomplished by differentiating the skin-tone pixels from the non-skin-tone pixels. This requires choosing a color space out of the number of color spaces in which we could operate. Appropriate color transformation is applied as explained[1].

Once we have accomplished the task of face detection, we concentrate on narrowing down to the mouth region of the speaker using an eye location algorithm. The location of eyes help us narrowing down to the mouth region[14].

In Chapter 3, the algorithm applied for lip tracking has been discussed. The region narrowed down to in the previous chapter (the mouth region) serves as the starting point for the process of lip tracking in this algorithm. A new form of active contour is the feature here[2]. It is known as the *"jumping snake"* since it *jumps* after each iteration to a new position. Once it rests on the lip

contour (for the first frame of the video), we proceed towards the process of corners detection and model fitting (cubic curves here) [2]. After that, key points representing the lip are tracked in the subsequent frames which gives us the representation of the viseme that the video represents.

Chapter 4 discusses the algorithms[3] applied for getting the key frames of a video (so that to reduce the computational complexity of comparing videos for recognition) and a modified form of Hausdorff distance which is a measure of similarity (or dissimilarity) between to-be-compared shapes and thus helps in determining which reference video is closest to the test video.

Chapter 2 Face Detection and Eye Localization

2.1 Face Detection

There are two traditional classes of techniques applied to the recognition of digital images of frontal views of faces under roughly constant illumination [10]. The first technique is based on the computation of a set of ***geometrical features*** from the picture of a face. This was the first approach toward an automated recognition of faces. The second class of techniques is based on ***template matching.***

Geometric, Feature-Based Matching: A face can be recognized even when the details of the individual features (such as eyes, nose, and mouth) are no longer resolved. The remaining information **is,** in a sense, purely geometrical and represents what is left at a very coarse resolution. The idea is to extract relative position and other parameters of distinctive features such as eyes, mouth, nose, and chin.

Template Matching: In the simplest version of template matching, the image, which is represented as a bi dimensional array of intensity values, is compared using a suitable metric (typically the euclidean distance) with a single template representing the whole face. There are, of course, several, more sophisticated ways of performing template matching. For instance, the array of grey levels may be suitably preprocessed before matching. Several full templates per each face may be used to account for the recognition from different viewpoints. Still another important variation is to use, even for a single viewpoint, multiple templates.

A rather different and more complex approach is to use a single template together with a qualitative prior model of how a generic face transforms under a change of viewpoint. The deformation model is then heuristically built into the metric used by the matching measure.

But the above techniques are computationally very demanding and cannot handle large variations in face images[1].

Categorizing face detection methods based on the representation used reveals that detection algorithms using holistic representations have the advantage of finding small faces or faces in poor-quality images, while those using geometrical facial features provide a good solution for detecting faces in different poses. A combination to holistic and feature-based approaches is a promising approach to face detection as well as face recognition[1]. Motion and skin-tone color are useful cues for face detection[1].

2.2 Choosing a Color Space for Face Detection

According to [11], research has been performed on the detection of human skin pixels in color images and on the discrimination between skin pixels and "non-skin" pixels by use of various statistical color models. As an example, ad hoc skin color models have been used as a preprocessor in analyzing large image databases; other researchers have used skin color models such as the single Gaussian model, a Gaussian mixture density model or histograms.

Recently, a comprehensive and detailed analysis of skin and non-skin color models was implemented by use of a very large database of skin and non-skin pixels manually extracted from the WorldWideWeb : the comparative performance of histogram models and of Gaussian mixture density models with the EM algorithm was analyzed for the standard 24-bit RGB color space, and histogram models were found to be slightly superior to Gaussian mixture models in terms of skin pixel classification performance for that color space.

In most experiments, skin pixels are acquired from a limited number of people under a limited range of illumination conditions. A relative robustness to changes in illumination conditions is achieved if a color space efficiently separating the chrominance from the luminance in the original color image is used. This implies a dimensionality reduction.

Normalized r-g chrominance space has often been used for face detection specifically because it reduces the sensitivity of the segmentation to changes in illumination. Other chrominance-luminance spaces that have been commonly used are the perceptually plausible HSV (or HSI) space or the hardware-oriented YIQ or YES spaces. The selection of a suitable chrominance space is an important task, because the shape of the skin and nonskin distributions depends on the chrominance space. Two important criteria are : 1) how well a given chrominance model can describe complex-shaped distributions in a given space, and 2) the amount of overlap between the skin and non-skin distributions in that space.

2.3 Implementation of Face Detection

Modeling skin color requires choosing an appropriate color space and identifying a cluster associated with skin color in this space. It has been observed that the normalized red-green (rg) space[5-1] is not the best choice for face detection [20], [21]. Based on Terrillon et al.'s [20] comparison of nine different color spaces for face detection, the tint-saturation-luma(TSL) space provides the best results for two kinds of Gaussian density models (unimodal and a mixture of Gaussians). We adopt the YC_bC_r space since it is perceptually uniform [22], is widely used in video compression standards (e.g., MPEG and JPEG) [23], and it is similar to the TSL space in terms of the separation of luminance and chrominance as well as the compactness of the skin cluster. Many research studies assume that the chrominance components of the skin-tone color are independent of the luminance component [24], [25], [26], [27]. However, in practice, the

skin-tone color is nonlinearly dependent on luminance. Detecting skin tone based on the cluster of training samples in the CbCr subspace results in many false positives[1]. And the case for the subspace of (Cb/Y) – (Cr/Y) results in many false negatives[1]. Thus, the YCbCr color space is nonlinearly transformed[1] to make the skin cluster luma-independent. Piecewise linear boundaries are fitted to the skin cluster. The transformed space enables a robust detection of dark andlight skin tone colors. In such a case, more skin-tone pixels with low and high luma are detected in the transformed subspace than in the CbCr subspace[1].

2.3.1 The Nonlinear Transformation:

In the YC_bC_r color space, we can regard the chroma (C_b and C_r) as functions of the luma(Y) : $C_b(Y)$ and $C_r(Y)$. Let the transformed chroma be $C_b'(Y)$ and $C_r'(Y)$. The skin color model is specified by the centers (denoted as $\underline{C_b}(Y)$ and $\underline{C_r}(Y)$) and spread of the cluster (denoted as $WC_b(Y)$ and $WC_r(Y)$) and is used for computing the transformed chroma[1].

$$C_i'(Y) = \begin{cases} (C_i(Y) - C_i(Y)) * \frac{Wc_i}{Wc_i(Y)} + C_i(K_h), Y < K_l \text{ or } K_h < Y \\ C_i(Y) \qquad\qquad\qquad\qquad , K_l < Y < K_h \end{cases} \tag{2.1}$$

$$Wc_i(Y) = \begin{cases} WLc_i + (Y - Y_{min}) * \frac{Wc_i - WLc_i}{K_l - Y_{min}}, Y < K_l \\ WHc_i + (Y_{max} - Y) * \frac{Wc_i - WHc_i}{Y_{max} - K_h}, K_h < Y' \end{cases} \tag{2.2}$$

$$\underline{C_b}(Y) = \begin{cases} 108 + (K_l - Y) * \frac{10}{K_l - Y_{min}}, Y < K_l \\ 108 + (Y - K_h) * \frac{10}{Y_{max} - K_h}, K_h < Y \end{cases} \tag{2.3}$$

$$\underline{C_r}(Y) = \begin{cases} 154 - (K_l - Y) * \frac{10}{K_l - Y_{min}}, Y < K_l \\ 154 + (Y - K_h) * \frac{22}{Y_{max} - K_h}, K_h < Y \end{cases} \tag{2.4}$$

where C_i in (1) and (2) is either C_b or C_r, Wc_b=46.97, WLc_b=23, WHc_b=14, Wc_r=38.76, WLc_r=20, WHc_r=10, K_l=125 and K_h=188. These parameter values have been estimated [1] from training samples of skin patches. Y_{min} and Y_{max} are the minimum and the maximum values of the luminance in all of the training samples.

The elliptical model for the skin tones in the transformed $C_b'C_r'$ space is described by the below two equations

$$\frac{(x-eC_x)^2}{a^2} + \frac{(y-eC_y)^2}{b^2} = 1 \tag{2.5}$$

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{pmatrix} * \begin{pmatrix} C_b' - C_x \\ C_r' - C_Y \end{pmatrix} \tag{2.6}$$

where C_x=109.38, C_y=152.02, Θ = 2.53(in radian), eC_x=1.6, eC_y=2.41, a=25.39, b=14.03 have been computed as given in [1] from the skin cluster in $C_b'C_r'$ space.

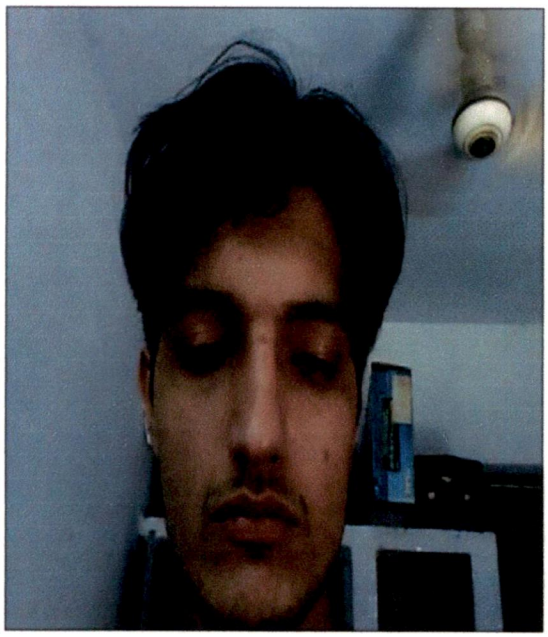

Figure 2-1 (a) : A frame from a video

Figure 2-1 (b) : Result of face detection for figure (a)

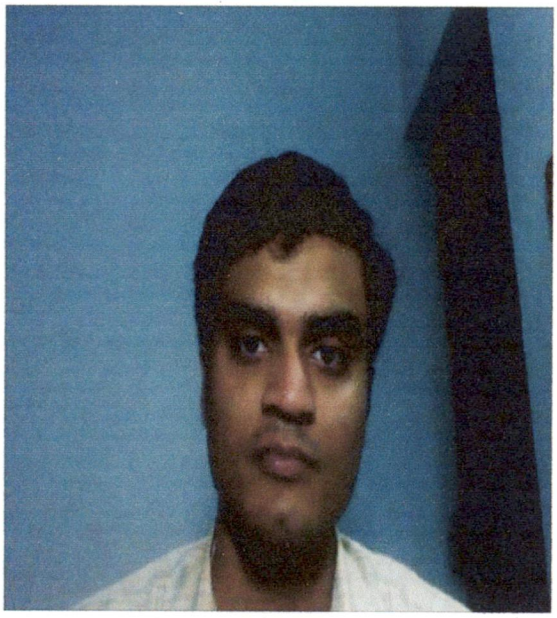

Figure 2-2 (a) : A frame from a video.

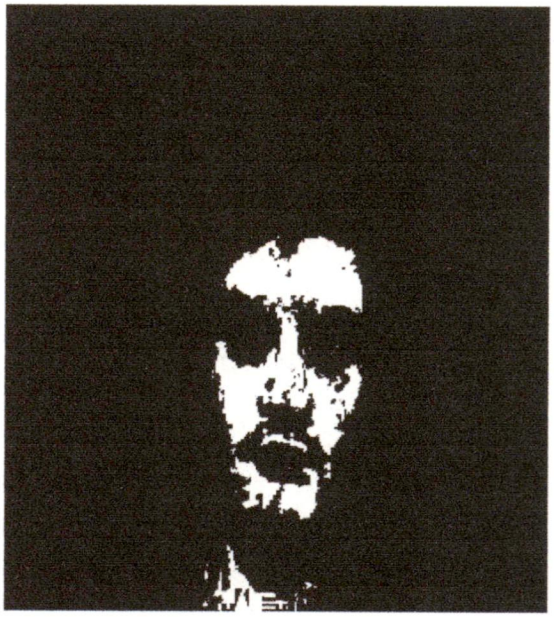

Figure 2-2 (b) : Result of face detection for figure (a).

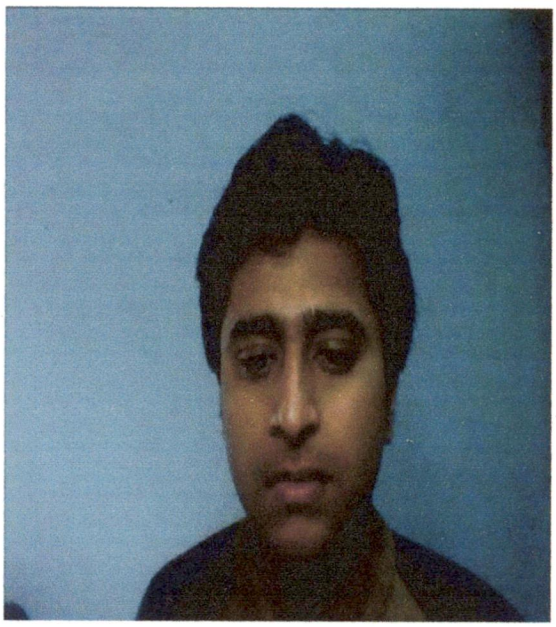

Figure 2-3 (a) : A figure from a video.

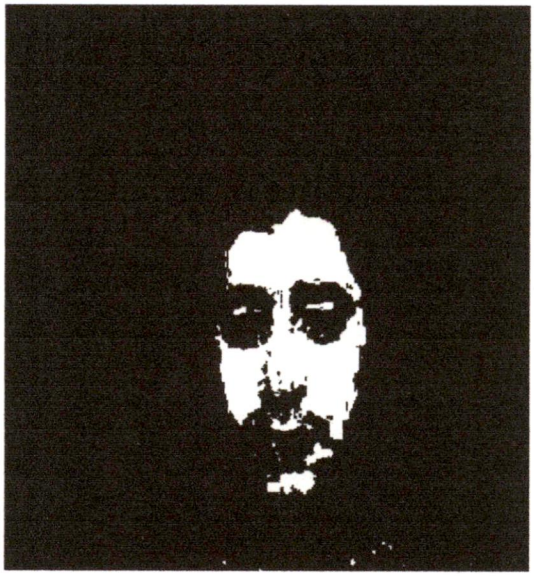

Figure 2-3 (b) : Result of face detection for figure (a).

2.4 Eye localization

In the previous sections, we have been able to accomplish getting the skin pixels of the speaker in the video. Now our purpose in this chapter would be to extract the location of eyes of the speaker from where we could get the region between the nose and the upper lip. This is so because the distance of mouth to the two eyes center is about 1.2 times of the distance between two eyes center[14].

According to [15], human faces have a special pattern that is usually different from the patterns of background objects in face images. The grayscales of the pupil and the iris of an eye are usually lower than those of the skin near the eye and those of the white of the eye; therefore, if we can find an appropriate threshold value to segment a face image, the eyes can be separated from other facial features and background objects in the segmented face image (i.e., the binary image, where, if the grayscale of a pixel is more than or equal to the threshold value, the grayscale of the pixel will be set to be 1 (white pixel); otherwise, it will be set to be 0 (black pixel).

The connected components (black pixels) in the segmented face image are called a block. To locate eyes from an appropriately segmented face image, a determination criterion of eye location, established in [15] by the priori knowledge of geometrical facial features, has been followed. The following[15] are the conditions which should be satisfied for the block to be selected as an eye block :

• The distance between the geometrical centers of the two eye blocks should be within a certain range of pixel number such as from 15 pixels to 45 pixels in a face image with size 120×160.

• There are no other blocks in a certain area under each eye.

• The vertical distance difference between the geometrical centers of the two eye blocks is not more than a certain number of pixels.

• The size (the pixel number) in each eye block is limited in a certain range.

• There is no other block between the two eye blocks.

• The proportion of height to length in the rectangular bounding box around each eye block is limited in a certain range.

• Any block connected with or very close to the four edges of face images is not an eye block.

As an example, the following is the result of applying the above procedure (in the next figure, only the centre of each eye block selected has been displayed) :

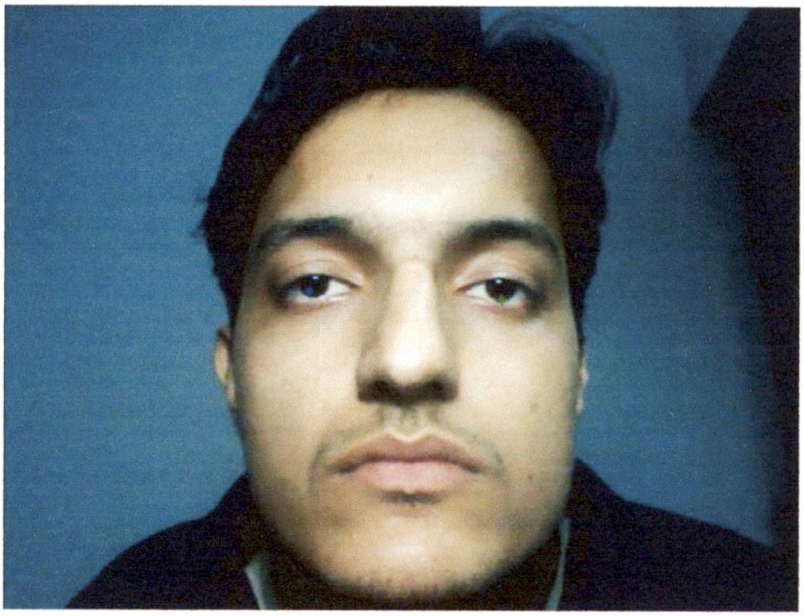

Figure 2-4 : The blue-coloured and the green-coloured (on the eyeball) are the centre points of the blocks that were selected as eye candidates.

Now since we have got the eye centres, we can easily narrow down to the region between the nose and the upper lip using the distance between the two points shown in the above figure. A point that would be this distance below the center of the line joining these two points would fall

in the region between the nose and the upper lip of the speaker. This point serves as the starting point of the algorithm implemented in the next chapter to achieve lip tracking.

Chapter 3 LIP TRACKING

During the last few years, many techniques have been proposed to achieve lip segmentation. Some of them use only low-level spatial cues such as color and edges. Zhang [12] uses hue and edge information to achieve mouth localization and segmentation. There is no shape or smoothness constraint, so the segmentation is often very rough, which makes this method unsuitable for applications that require a high level of accuracy, such as lip reading or clone synthesis. In [13], a linear discriminant analysis (LDA) is used to separate the lip pixels from the skin pixels and thus to extract the lip contour. Even if the LDA is followed by a smoothing operation, the resulting segmentation is often noisy. Because of their ability to take smoothing and elasticity constraints into account, the "snakes" [28] have been widely applied to lip segmentation [29]–[31]. They can give quite good results, but most of the time the tuning of parameters is very difficult to achieve, and the snakes often converge to wrong results when the initial position is far from the lip edges. Moreover, the mouth corners' detection is generally difficult because they are located in low gradient areas, which leads to rough final contours. Some authors propose to detect the mouth corners by a specific algorithm [30] and to keep them still during the snake convergence. This improves accuracy, but it does not address the problem of parameters adjustment.

To make segmentation more robust and realistic, *a priori* shape knowledge has to be used. By designing a global shape model, boundary gaps are easily bridged and overall consistency is more likely to be achieved. This supplementary constraint ensures that the detected boundary belongs to possible lip shape space. For example, active shapes models (ASMs) can be used [32], but they need a large training set to cover a high variability range of lip shapes. Moreover, the images of this training set have to be cautiously calibrated. The face orientation and the lighting conditions have to be constant, otherwise the ASM method leads to unreliable results. To avoid a restricting training step, a parametric description can be used to design models. As introduced by Yuille [33], a parametric deformable template is a parameterized mathematical model used to track the movement of a given object. In our case, the lip shape is approximated by a set of curves which is uniquely described by some parameters. Several parametric models have already been proposed. Tian [34] uses a simple three-states geometric model made of parabola. The color and shape information is used to know which model to use: mouth tightly closed, closed, or open. Then, four keypoints are used to draw the model. The position of the model is generally good, but it does not fit the boundary with accuracy because only symmetrical parabolic shapes can be generated. To make the model more flexible, other authors propose to use two parabola instead of one for the upper boundary [35] or to use quartics instead of parabola [36]. This improves accuracy, but the models are still limited by their rigidity, particularly in the case of an asymmetric mouth.

In the model discussed in [2], which has been implemented, there is more flexibility as cubic curves have been used. The model is positioned by several keypoints located on the mouth

boundary, and it is fitted by using edge information. Interframe tracking has been used to enhance the speed and the robustness of the segmentation.

In [2], the keypoints are detected in the first frame itself. The model is a quasi-automatic method that only requires the manual selection of a single point located above the mouth. This point is used as a seed for a new kind of active contour: the "*jumping snake.*" Unlike classic snakes, its parameters are easy to choose, and its convergence is ensured even if the initial seed is far away from the mouth. The model is flexible enough to reproduce the specificities of very different lip shapes. This enables accurate segmentations, even in the challenging case of an asymmetric mouth.

3.1 SKIN AND LIP COLOR ANALYSIS

In ***RGB*** space, skin and lip pixels have quite different components. For both, red is prevalent. Moreover there is more green than blue in the skin color mixture and for lips these two components are almost the same. Skin appears more yellow than lips because the difference between red and green is greater for lips than for skin[37]. Hulbert and Poggio [6] proposes a pseudo hue definition that exhibits this difference. It is computed as follows :

$$h(x,y) = \frac{R(x,y)}{R(x,y) + G(x,y)} \tag{3.1}$$

where R(x,y) and G(x,y) are respectively the red and the green components of the pixel ***(x,y).*** Unlike usual hue, pseudo hue is bijective. It is higher for lips than for skin[7] (next figure).

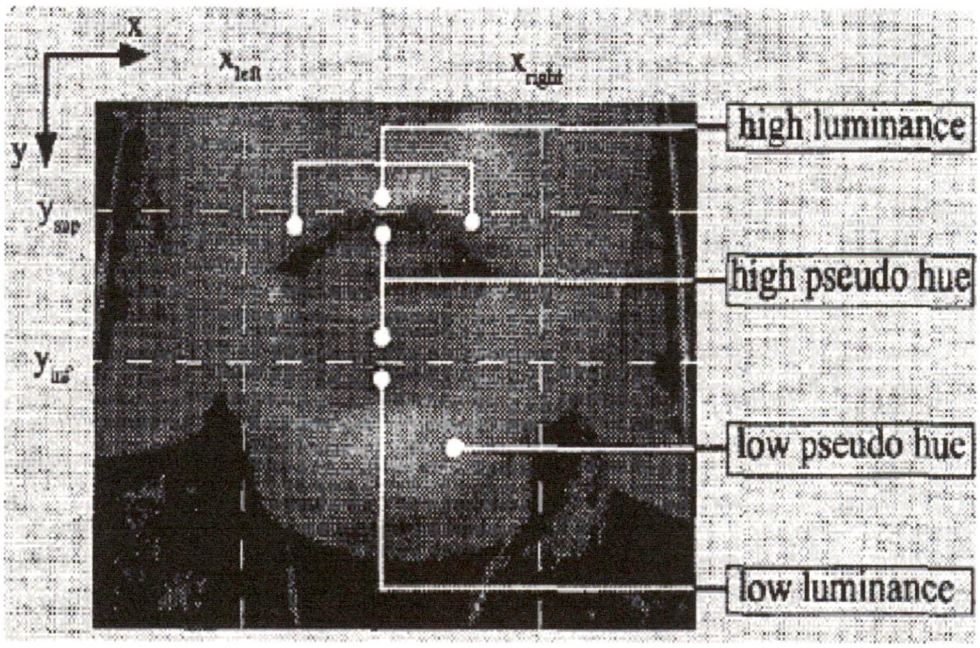

Figure 3-1 : The mouth region characteristics

Intensity is also a good cue to be taken into account. In general, light comes from above the speaker. Then the top frontier of the upper lip is very well illuminated while the upper lip itself is in the shadow. At the opposite, the bottommost lip is in the light while its central lower boundary is in the shadow. very well illuminated while the upper lip itself is in the shadow. At the opposite, the bottommost lip is in the light while its central lower boundary is in the shadow. To combine color and luminance information, [37] introduces the "hybrid edges" $\mathbf{R}_{sup}(x,y)$ and $R_{inf}(x,y)$, computed as follows(vectors are written in bold) :

$$\boldsymbol{R}_{sup}(x,y) = \nabla[h_N(x,y) - L_N(x,y)] \tag{3.2}$$

$$R_{inf}(x,y) = \nabla_y[h_N(x,y) + L_N(x,y)] \tag{3.3}$$

where $L_N(x,y)$ and $h_N(x,y)$ are respectively the pseudo hue and the luminance of pixel (x,y), normalized between 0 and 1 on the mouth region. $\nabla[.]$ is the gradient operator, and $\nabla_y[.]$ is its vertical component. R_{inf} is a scalar which is very negative for the lower central boundary of the mouth. $\mathbf{R}_{sup}$ is a vector whose norm has a high value on the upper lip boundary.

3.2 The Jumping Snake Algorithm

Active contours, or snakes, have proved their efficiency in many segmentation problems. Since their introduction by Kaas *et al.* [28], many improvements have been proposed in the literature. But none of them has totally removed the two major weak points of the snakes: the choice of parameters and the high dependence on the initial position. The method presented in [2] helps to address these problems.

To find the upper mouth boundary, [2] introduces a new kind of active contour that is called "jumping snake" because its convergence is a succession of jumps and growth phases [38]. It is initialized with a seed S^0 that can be located quite far away from the final edge. The seed is, as in [2], put manually above the mouth but since we have been able to reach the region between the nose and the upper lip of the speaker, we can get this seed as explained in chapter 2 (figure below) :

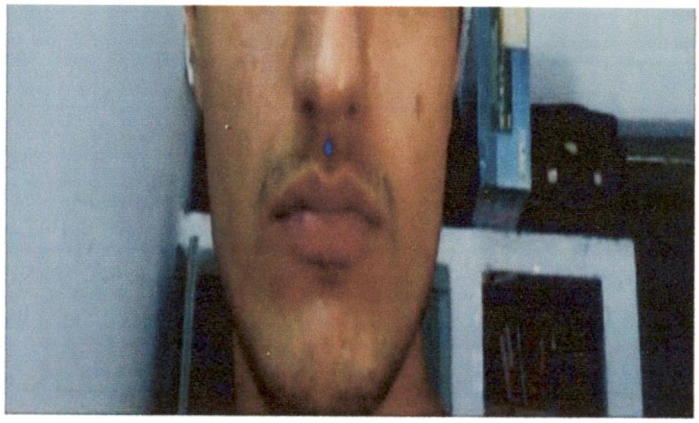

Figure 3-2 : Initial seed S^0 marked with blue colour.

The snake grows from this seed until it reaches a predetermined number of points. This growth phase is quite similar to the growing snake proposed by Berger and Mohr [39], in the sense that the snake is initialized with a single point and is progressively extended to its endpoints. Then, the seed "jumps" to a new position that is closer to the final edge. The process stops when the size of the jump is smaller than a threshold.

In [37], "hybrid edge" has been introduced that combines color and luminance information. It is computed as follows (vectors and matrixes are written in bold):

$$\boldsymbol{R}_{top}(x,y) = [h_N(x,y) - L_N(x,y)] \qquad (3.4)$$

where $h_N(x,y)$ and $L_N(x,y)$ are respectively the pseudo hue and the luminance of the pixel at the location (x,y), normalized between 0 and 1. ∇ is the gradient operator. The pseudo hue, introduced by Hulbert and Poggio [40], is less noisy than the usual hue and is higher for lips than for skin, as shown in [16-2]. It is computed as follows:

$$h(x,y) = \frac{R(x,y)}{R(x,y)+ G(x,y)} \tag{3.5}$$

where R(x,y) and G(x,y) are the red and green components of the pixel located at (x,y). The hybrid edge $\mathbf{R_{top}}$ exhibits the top frontier of the mouth much better than the classic gradients of luminance or pseudo-hue do. It is used to guide the *jumping snake* toward the upper lip edge.

During the growth phase, left and right endpoints are added to the snake. They are located at a constant horizontal distance, denoted Δ , from the previous point. Moreover, the search area is restricted to the angular sector $[\Theta_{inf}, \Theta_{sup}]$.

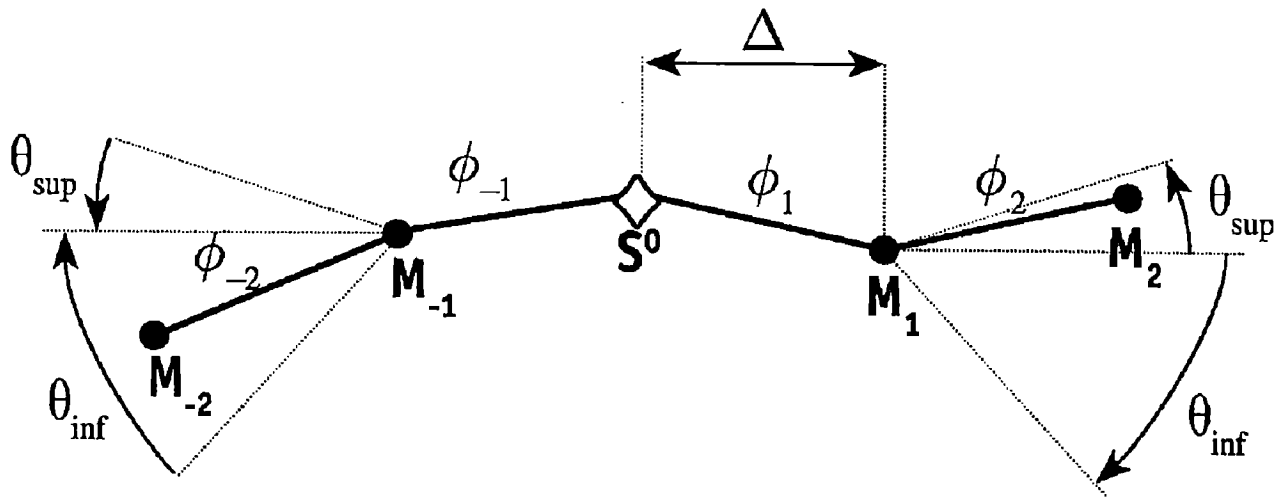

Figure 3-3 : From seed S^0 (◊), the snake is extended by adding left and right endpoints(•). The $\mathbf{R_{top}}$ mean flows Φ_i through each segment have to be maximized.

The best left and right endpoints, denoted $M_{-(i+1)}$ and M_{i+1} , are found in this area by maximizing the $\mathbf{R_{top}}$ mean flow through the end segments $M_{-(i+1)}M_{-i}$ and $M_i M_{i+1}$ (see figure above). These two mean flows can be written as follows:

$$\Phi_{i+1} = \frac{\int_{M_i}^{M_{i+1}} \boldsymbol{Rtop \cdot dn}}{|\boldsymbol{M_i M_{i+1}}|} \tag{3.6}$$

$$\Phi_{-i-1} = \frac{\int_{M_{-i-1}}^{M_{-i}} \boldsymbol{Rtop \cdot dn}}{|\boldsymbol{M_{-i-1} M_{-i}}|} \tag{3.7}$$

where **dn** is the vector orthogonal to the segment. The maximizations of Φ_{i+1} and $\Phi_{-(i+1)}$ are achieved by a systematic computation over a small set of candidates located in the search area.

When the snake reaches a predetermined number of points=2n+1, the growth stops and the position of the new seed S^1 is computed. This is the jump phase of the jumping snake algorithm.

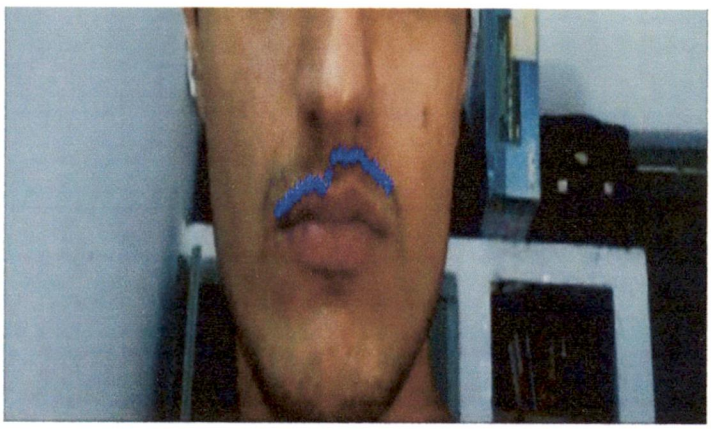

Figure 3-4 : Here, the snake has reached a predetermined number of points (=2n+1). This is the end of the first iteration.

Let $\{M_{-N},\ldots,M_{-1},S^o,M_1,\ldots,M_N\}$ be the points of the snake and let $\{\Phi_{-N},\ldots, \Phi_{-1},\Phi_1,\ldots, \Phi_N\}$ be the mean flows through the segments. The new seed S^1 has to get closer to high gradient regions, i.e., high mean flow segments. We consider S^1 is the barycentre of S^o and the points which are in

the highest gradient regions. If $\{i_1,\ldots,i_N\}$ are the indices associated with the N highest mean flows, then the vertical position of S^1 can be written as follows :

$$Y_{S1} = \frac{1}{2}\left(Y_{S0} + \frac{\sum_{k=1}^{N} \Phi_{i_k} \cdot y(i_k)}{\sum_{k=1}^{N} \Phi_{i_k}}\right) \tag{3.8}$$

where $y(i_k)$ is the vertical position of the point M_{ik}. The horizontal position X_{S1} of the seed is kept constant.

Then, a new snake grows from this new seed until it reaches the predetermined length and "jumps" again. This growth-jump process is repeated until the jump's amplitude becomes smaller than one pixel. Typically, four or five jumps are needed to achieve the convergence of the snake. In its final position, it lies on the upper lips boundary(as shown in the figures below) :

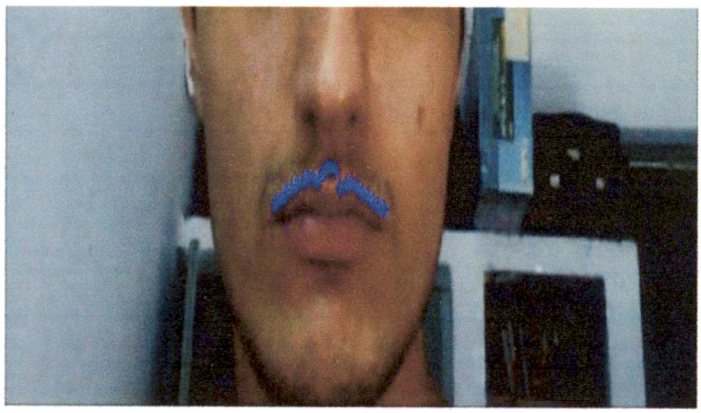

Figure 3-5 : At the end of 2nd iteration.

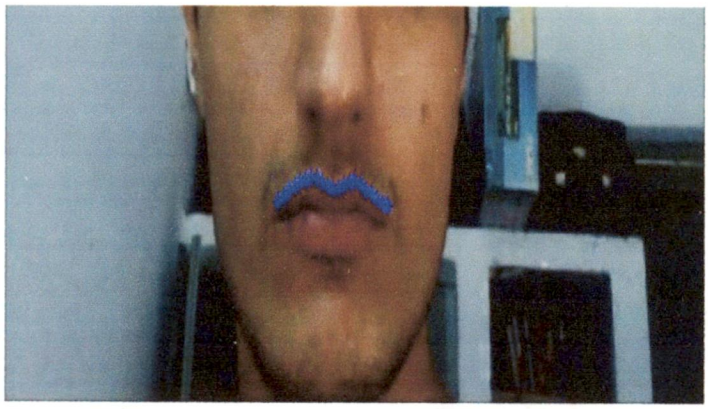

Figure 3-6 : At the end of 6th iteration(iteration ends here).

Unlike classic snakes, the choice of the jumping snake parameters (Δ, Θ_{inf}, Θ_{sup}, N) is easy and intuitive. If there is no strong edge in the neighborhood of the snake, the overall directions of its left and right parts are dependent on the choice of Θ_{inf}, and , the angular limits of the search area. When $\Theta_{inf} = \Theta_{sup}$, the snake tends to be horizontal. If $|\Theta_{inf}| < |\Theta_{sup}|$, then the two branches tend to go upwards. At the opposite, they tend to go downwards when $|\Theta_{inf}| > |\Theta_{sup}|$. Here, the initial seed S^o is above the mouth and the snake has to fall down to get closer to the upper mouth boundary. Then we choose $|\Theta_{inf}| > |\Theta_{sup}|$.

The horizontal distance Δ has a direct influence on the accuracy of the snake final position. For a small value of Δ, the detected upper edge is very detailed. However, for a given length of the snake, the computational cost is higher because more points have to be computed. On the other hand, a high value of Δ leads a rough estimation, but the convergence is achieved quicker. So, the choice of Δ is a compromise between speed and accuracy. The last parameter N gives the number of points of the snake. For a given Δ, a high value N of leads to a long snake.

3.3 Upper and lower key points detection

The keypoints give important cues about the lip shape. They are used as fulcra for the computation of the model. We use six principal key points(figure below from [2]) :

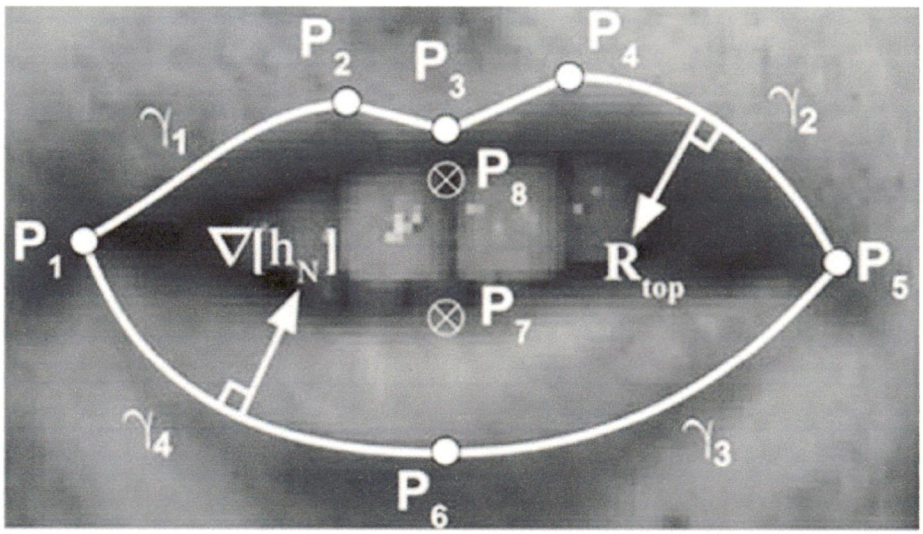

Figure 3-7 : Four cubics γ_1, γ_2, γ_3, γ_4 passing by six key points.

the right and left mouth corners (P_1 and P_5), the lower central point (P_6) and the three points of the Cupid's bow (P_2,P_3, and P_4). The three upper points are located on the estimated upper lip boundary resulting from the jumping snake algorithm. P_2 and P_4 are the highest points on the left and right of the seed. P_3 is the lowest point of the boundary between P_2 and P_4.

According to [2], the point P_6 is found by analyzing $\bar{V}_y[h]$, the one-dimensional gradient of the pseudo hue along the vertical axis passing by P_3(figure below from [2]) :

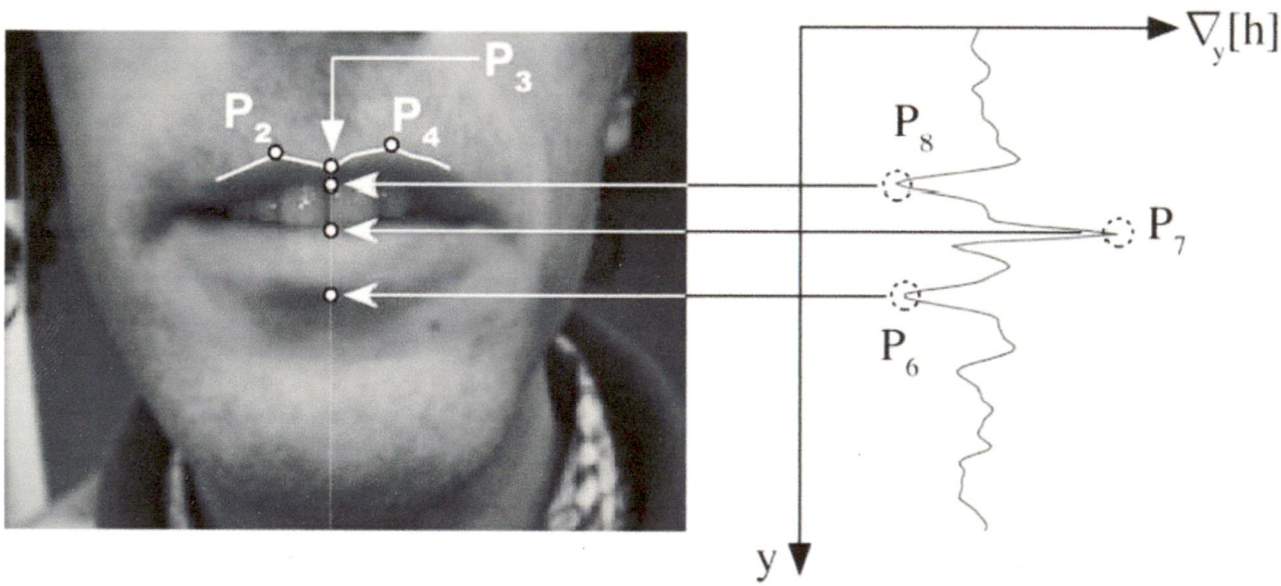

Figure 3-8 : Way of finding P_6 as described in [2].

As this was attempted, the results didn't come as expected. So, an attempt was made to get the lower lip using $\mathbf{R}_{top}$. Similar to locating the seed for the upper lip, we can located the seed here for the lower lip. The results are as shown :

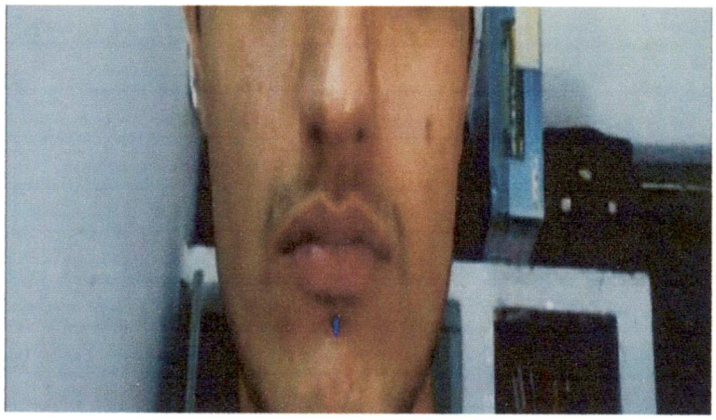

Figure 3-9 : Initial seed for the lower lip.

As for the upper lip, mean flows are maximized.

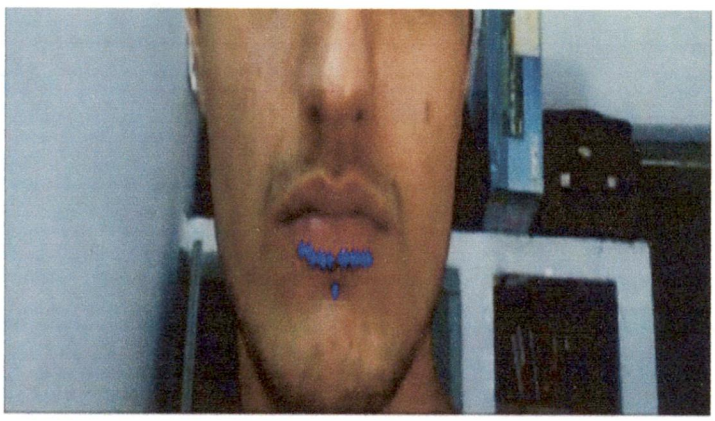

Figure 3-10 : Lower coordinates after 1st iteration.

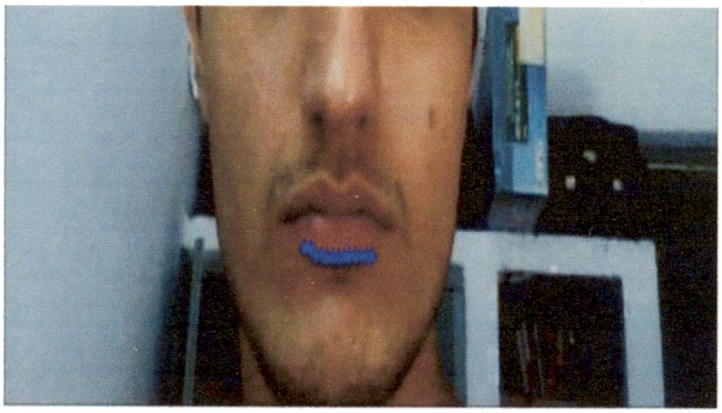

Figure 3-11 : Lower coordinates after 5th iteration.

3.4 Contour Extraction

Several parametric models for the lip boundary have been proposed. Tian [34] uses a model made of two parabolas. It is very easy to compute, but it is too simple to fit the edges with accuracy(figure (a)) :

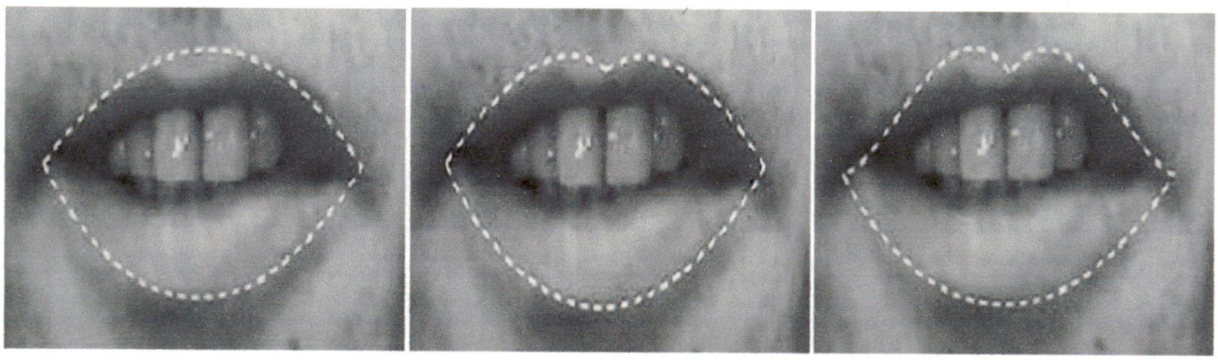

Figure 3-12 (a,b,c) : Models, respectively, with two parabolas, three parabolas, and quartics.

Other authors propose to use two parabolas instead of one for the upper boundary [35] or to use quartics instead of parabolas [36]. This improves accuracy, but the model is still limited by its rigidity, particularly in the case of asymmetric mouth shape(figures (b) and (c)).

The model used in [2] is flexible enough to reproduce the specificities of very different lip shapes and is composed of five independent curves. Each one of them describes a part of the lip boundary. In [2], each cubic has a null derivative at key points P_2,P_4 or P_6. As an example, γ_1 has a null derivative on P_2 .

3.4.1 Mouth corners and model fitting :

The model fitting and the mouth corners detection are linked[2] . A cubic curve is uniquely defined if its four parameters are known. Here, each curve passes by, and has a null derivative on points P_2,P_4 or P_6 . These considerations bring two constraint equations that decrease the number of parameters to be estimated from four to two for each cubic. So, only two more points of each curve are needed to achieve the fitting. These missing points are chosen in the most reliable parts of the boundary, i.e., near P_2, P_4 or P_6 . Now it should be possible to compute the curves γ_i passing by them and to find the mouth corners where these curves intersect. However, this direct and intuitive method provides not very accurate results. The reliable points used to compute the model are much too close to each other. A very small displacement of one of them leads to a completely different curve.

The method which has been proposed in [2] is that certain number of candidate points for P_1 and P_5 are chosen along the curve of minimum luminance(L_{min}) points, one each for a column of pixels between the upper and lower lip boundaries. It has been supposed in [2] that the corner points lie on this line. So, the fitting is achieved by finding the corners that give the best couple of curves.

To know if a curve fits well to the lip boundary, an edge criterion is used in [2]. If the upper curves γ_1 and γ_2 fit perfectly to the edge, they are orthogonal to the $\mathbf{R}_{top}$ gradient field. On the other hand, the curves γ_3 and γ_4 have to be orthogonal to the $\nabla[h_N]$ gradient field. $\Phi_{top,i}$ and $\Phi_{low,i}$, the mean flows through the upper and lower curves, are calculated as follows :

For i={1,2} :

$$\Phi_{top,i} = \frac{\int \boldsymbol{R_{top}} \bullet \boldsymbol{dn}}{\int \boldsymbol{ds}} \qquad (3.8)$$

both the integrals are over the curve Φ_i .

For i={3,4} :

$$\Phi_{low,i} = \frac{\int \nabla[h_N] \bullet \boldsymbol{dn}}{\int \boldsymbol{ds}} \tag{3.9}$$

both the integrals are over the curve Φ_i .

Here, **dn** and **ds** are the vector orthogonal to the segment and the curvilinear abscissa, respectively. In the implementation, p = 5 possible positions along L_{min} were considered for each point P_1 and P_5. The best position gives a high $\Phi_{top,i}$ and a very negative $\Phi_{low,i}$. The expression is

$$\Phi^k_{total} = \Phi^k_{top,normalized} - \Phi^k_{low,normalized}, \quad k \in \{1,2,\ldots,p\} \tag{3.10}$$

where

$$\Phi^k_{top,normalized} = (\Phi^k_{top} - min_{j \in \{1,\ldots,n\}}(\Phi^j_{top}))/(max_{j \in \{1,\ldots,n\}}(\Phi^j_{top}) - min_{j \in \{1,\ldots,n\}}(\Phi^j_{top})) \tag{3.11}$$

Φ_{top}^{k} and Φ_{low}^{k} are associated with the tested corner number k. $\Phi_{top,normalized}^{k}$ and $\Phi_{low,normalized}^{k}$ are their normalized values over the whole tested set. Φ_{total}^{k} is high, the corner position is reliable because the corresponding curves fit well to the lip boundaries. Thus, the boundaries and the corners are found in a single operation. The maximum of Φ_{total}^{k} gives the position of the corner along L_{min}.

As an example, the following three contours were estimated (i.e their respective Φ_{total} were estimated) in an attempt to correctly identify the corner point P_1 (next figure) :

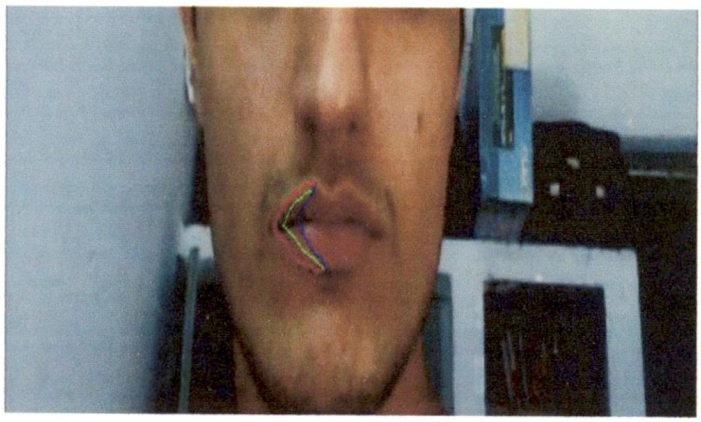

Figure 3-13 : Φ_{total} associated with each curve (they differ in colours) are : red = 1, green = -0.1991, blue = -0.0279.

As we can see from the above figure, we get the location of P_1 as shown next :

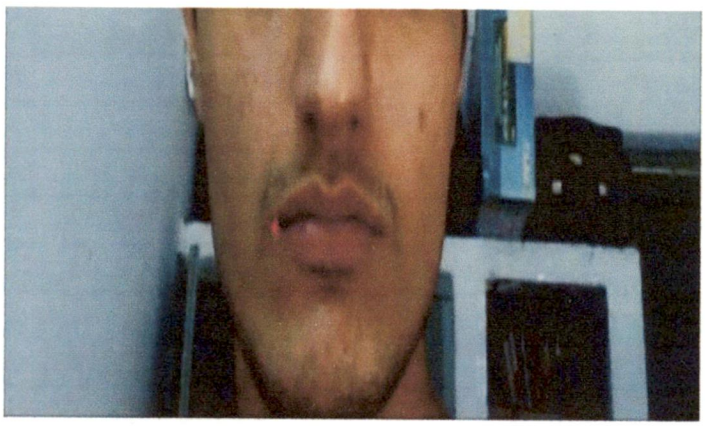

Figure 3-14 : Point marked red is chosen as the corner point P1 (from the candidate contours shown before).

3.5 Keypoints Tracking

To increase the robustness and the speed of the segmentation,the keypoints are tracked from one image to the other. Their positions have been obtained in [2] using a variant of the Kanade-Lucas algorithm[42] adapted to the particular geometry of the mouth.

The neighborhoods of the points being tracked are assumed to have only translation movements from image I to next image J as follows[2] :

$$J(x,y) = I(x-\alpha, y-\beta) + n(x,y) \tag{3.12}$$

where $(\alpha, \beta)^T$ are the components of the displacement vector **d** and n(x,y) is the noise level for the pixel (x,y). I(x,y) and J(x,y) are scalars, for example, the luminance value of the pixel (x,y).

The vector **d** is chosen to minimize the residue factor ε, computed on the neighborhood window W, around the pixel (x,y), as follows :

$$\varepsilon = \iint [I(\boldsymbol{x} - \boldsymbol{d}) - J(\boldsymbol{x})]^2 \, \omega(\boldsymbol{x}) d\boldsymbol{x} \tag{3.13}$$

the integral being evaluated over the window W. Here, $\mathbf{x} = (x,y)^T$ and $\omega(\boldsymbol{x})$ is a weighting function usually constant and equal to 1 [2].

The resolution of above equation[41] leads to the following 2x2 linear system of equations :

$$\boldsymbol{Gd}=\boldsymbol{e} \tag{3.14}$$

where

$$G = \iint \boldsymbol{g}(\boldsymbol{x}) \boldsymbol{g}^T(\boldsymbol{x}) \, \omega(\boldsymbol{x}) d\boldsymbol{x} \tag{3.15}$$

$$\boldsymbol{e} = \iint \big(I(\boldsymbol{x}) - J(\boldsymbol{x})\big) g(\boldsymbol{x}) \, \omega(\boldsymbol{x}) d\boldsymbol{x} \tag{3.16}$$

$$\boldsymbol{g}^T = \left(\frac{\partial I(\boldsymbol{x})}{\partial \boldsymbol{x}} \quad \frac{\partial I(\boldsymbol{x})}{\partial \boldsymbol{y}} \right) \tag{3.17}$$

where the integrals involved in **G** and **e** are computed over the window W.

Some of the results are shown below :

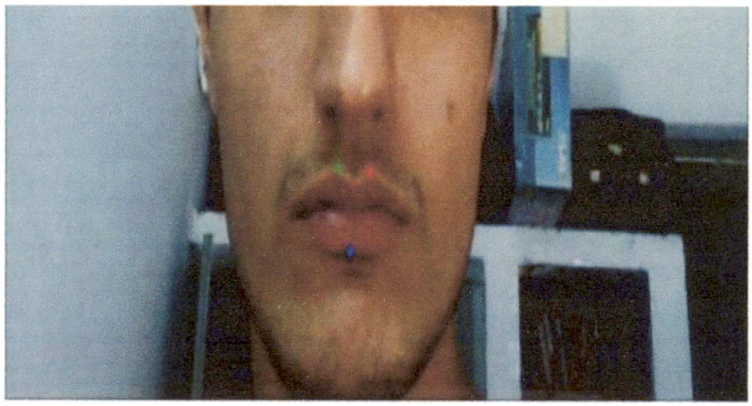

Figure 3-15 : Keypoints from the first frame.

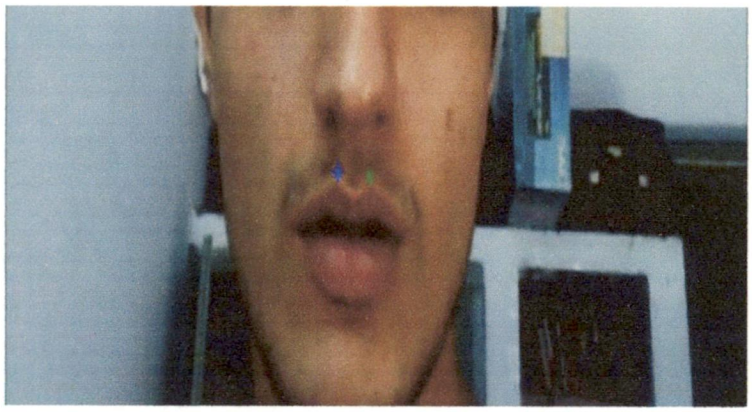

Figure 3-16 : Predicted location of keypoints P_2 and P_4 (for the 30^{th} frame).

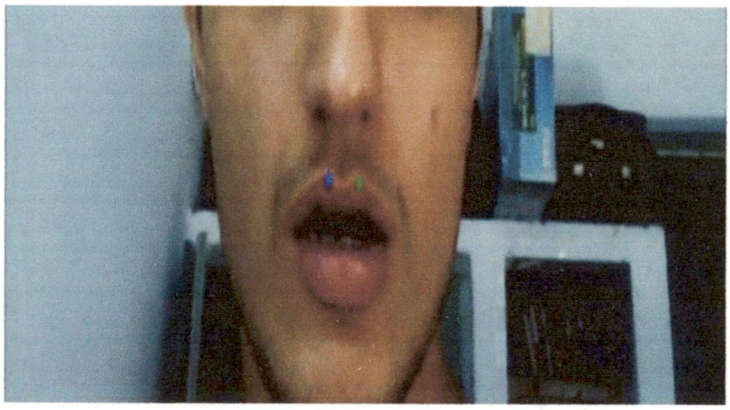

Figure 3-17 : Predicted location of keypoints P_2 and P_4 (for the 80^{th} frame).

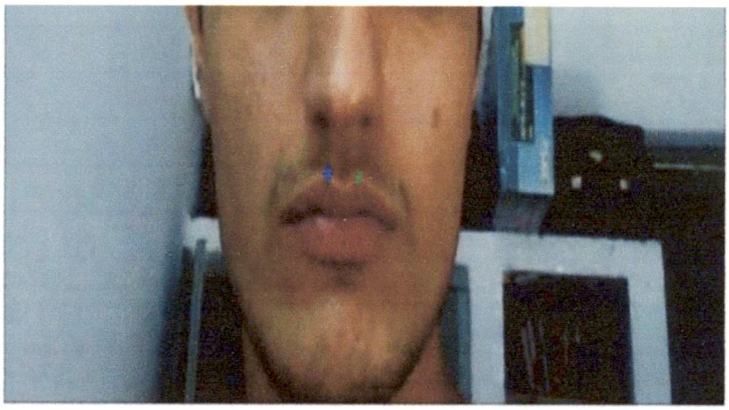

Figure 3-18 : Predicted location of keypoints P_2 and P_4 (for the last frame).

Chapter 4 Key Frames Extraction and Dissimilarity measure

4.1 Cumulative Directed Divergence

For comparing videos in visual speech recognition, [3] suggests that, rather than comparing all of the frames of the videos to be compared, key frames could be extracted from each frame and compare the set of key frames using the modified Hausdorff distance. The technique used in [3] to extract key frames from each video is cumulative directed divergence.

Let p and q be probability density functions. In the terminology of S. Kullback[8], the directed divergences of p and q are the two integrals in

$$B\int q(x)\,log\left(\frac{q(x)}{p(x)}\right)dx + C\int p(x)\,log\left(\frac{p(x)}{q(x)}\right)dx \qquad (4.1)$$

the sum, with B = C= 1, is the divergence.

The commonly used video indexing methods utilize histogram comparisons, because extraction of histograms is computationally efficient compared with the motion based methods[43]. Most common algorithms using histogram comparison include histogram difference[44], Euclidean metric[45] and directed divergence[3].

The divergence measure is defined by the sum of directed divergences[47]. The directed divergences of histograms are expressed as[4] :

$$\sum_j H_{t+1}(j)\log\left(\frac{H_{t+1}(j)}{H_t(j)}\right) + \sum_j H_t(j)\log\left(\frac{H_t(j)}{H_{t+1}(j)}\right) \qquad (4.2)$$

where $H_t(j)$ signifies the histogram in the j th bin($0 \le j \le 255$), with the subscript t denoting the t th frame and bin signifying the gray level range of the histogram representation[4].

The key frames are detected if the directed divergence value between the current frame and the previous key frame is larger than the given threshold[3]. The extracted key frames can be used for matching video sequences with a very low computational load[48].

Following four keyframes were extracted out of a video of 120 frames :

Figure 4-1 : (clockwise from top-left) : 1st frame, 37th frame, 60th frame, 77th frame.

4.2 Modified Hausdorff distance

4.2.1 Hausdorff distance

A central problem in pattern recognition and computer vision is determining the extent to which one shape differs from another. Pattern recognition operations such as correlation and template matching and model-based vision methods can all be viewed as techniques for determining the difference between shapes. It is important for shape comparison functions to obey metric properties.

Given two finite point sets A = $\{a_1,\ldots,a_p\}$ and B = $\{b_1,\ldots,b_p\}$, the Hausdorff distance is defined as [9] :

$$H(A,B) = max(h(A,B),h(B,A)) \quad (4.3)$$

Where the directed measure $h(A,B) = \max_{a \in A}\min_{b \in B}\|a - b\|$ with $\|\bullet\|$ denoting the norm on the points of A and B [49].

The function h(A,B) is called the directed Hausdorff distance from A to B. It identifies the point a ϵ A that is farthes from any point of B and measures the distance from a to its nearest neighbor in B (using the given norm ||.||), that is, h(A,B) in effect ranks each point of A based on its distance to the nearest point of B and then uses the largest ranked such point as the distance(the most mismatched point of A). Intuitively, if h(A,B) = d, then each point of A must be within distance of d of some point of B, and there also is some point of A that is exactly distance d from the nearest point of B(the most mismatched point).

The Hausdorff distance H(A,B) is the maximum of h(A,B) and h(B,A). Thus it measures the degree of mismatch between two sets by measuring the distance of the point of A that is farthest from any point of B and vice versa. Intuitively, if the Hausdorff distance is d, then every point of A must be withing a distance d of some point of B and vice versa. Thus, the notion of resemblance encoded by this distance is that each member of A be near some member of B and vice versa. Unlike most methods of comparing shapes, there is no explicit pairing of points of A with points ofB (for example, many points ofA may be close to the same point of B).

4.2.2 Video sequence matching using the modified Hausdorff distance :

For matching between video sequences, [3] employs the modified Hausdorff distance measure.

In [3], the modified Hausdorff distance D(S,R) is given by

$$D(S,R) = max[min_{r \in R}\{d(s_1,r)\}, min_{r \in R}\{d(s_2,r)\},\ldots, min_{r \in R}\{d(s_n,r)\}] \quad (4.4)$$

Where $S=\{s_1,\ldots,s_n\}$ represents the set of key frames for the query sequence and $R=\{r_1,\ldots,r_m\}$ signifies the set of key frames for matching sequences, with n and m denoting the total numbers of elements in sets S and R respectively[50].

Using the above form of modified Hausdorff distance, we compare a test video with the reference database of the visemes and results show that the Hausdorff distance is least for that stored viseme which represents the test viseme.

Chapter 5 Conclusions and Future Work

This thesis being aimed at extracting visual information for aiding speech recognition, we have been able to, first of all, extract the location of the speaker via face detection algorithm. The algorithm uses skin-tone color. The difficulty of detecting the low-luma and high-luma skin tones have been circumvented using a nonlinear transform to the YC_bC_r color space. Skin regions are detected over the entire image.

After achieving this task, we aimed to reach the mouth region of the speaker since that serves as the starting point to the algorithm of lip tracking (Chapter 3) implemented. For doing so, we aimed at extracting the location of both eyes of the speaker, which lead us to the mouth region.

The previous result led us to the implementation of the lip tracking algortithm. The algorithm employs a new kind of active contour : the "*jumping snake*". It can be initialized relatively far away from the final contour when compared with classic snakes. Using an "hybrid edge", accurate lip boundary localization in the first image of a video sequence is ensured [2]. Then a cubic-curves model is used to fit the outer lips boundary. It's high flexibility enables very realistic results [2]. To achieve segmentation in the following images, an interframe tracking of keypoints is used.

For comparing videos, firstly, key frames have been extracted using directed divergence [3]. This reduces the complexity related to the comparison of different video sequences. A modified form of Hausdorff distance is finally used to measure the dissimilarity between video sequences.

As an extension to the work presented in this thesis, we can implement an algorithm which could detect phoneme boundaries in the audio. This would give us the intra-word and inter-word viseme boundaries. These visemes could be recognized using the algorithms discussed in this thesis and as a result, we could develop a visual speech recognition system which could detect not only what is stored in it's database but any possible combination of the visemes stored.

REFERENCES

[1] Rein-Lien Hsu, Mohamed Abdel Mottaleb, and K. Anil Jain, "Face Detection in Color Images." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:696–706, 2002.

[2] Nicolas Eveno, Alice Caplier, Pierre-Yves Coulon, "Accurate and Quasi-Automatic Lip Tracking", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol 14, NO. 5, May 2004.

[3] Sang Hyun Kim, Rae-Hong Park, "An Efficient Algorithm for Video Sequence Matching Using the Modified Hausdorff Distance and the Directed Divergence" *IEEE Transactions on Circuits and Systems for Video Technology*, Vol 12, NO. 7, July 2002.

[4] Sang Hyun Kim, Rae-Hong Park, "A Novel Approach to Video Sequence Matching Using Color and Edge Features with the Modified Hausdorff Distance".

[5] Liang Tao, Hua-bin Wang, "Detecting and Locating Human Eyes in Face Images Based on Progressive Thresholding" , *Proc. IEEE Conf. Robotics and Biomimetics* pp. 445-449, Dec 2007

[6] A. Hulbert, T. Poggio, "Synthesizing a Color Algorithm From Examples", *Science,* Vol239, pp 482-485, 1998.

[7] N.Eveno, A. Caplier, P.Y. Coulon, "A New Color Transformation for Lips Segmentation", *IEEE Workshop* on *Multimedia Signal Processing (MMSP 'Ol),* Cannes, France, 2001.

[8] S. Kullback, Information Theory and Statistics. New York: Dover, 1957.

[9] D. P. Huttenlocher,G.A. Klanderman, andW. J. Rucklidge, "Comparing images using the Hausdorff distance," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 850–863, Sept. 1993.

[10] R. Brunelli and T. Poggio, "Face Recognition: Features vs. Templates," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 15, no. 10, pp. 1042-1052, Oct. 1993.

[11] J.C. Terrillon, M.N. Shirazi, H. Fukamachi, and S. Akamatsu, "Comparative Performance of Different Skin Chrominance Models and Chrominance Spaces for the Automatic Detection of Human Faces in Color Images," *Proc. IEEE Int'l Conf. Face and Gesture Recognition,* pp. 54-61, 2000.

[12] X. Zhang, R. M. Mersereau, M. A. Clements, and C. C. Broun, "Visual speech feature extraction for improved speech recognition," *Proc. ICASSP*, 2002, pp. 1993–1996.

[13] A. Nefian, L. Liang, X. Pi, L. Xiaoxiang, C. Mao, and K. Murphy, "A couple HMM for audio-visual speech recognition," *Proc. ICASSP*, 2002, pp. 2013–2016.

[14] Zhao Quanyou, Pan Baochang, Zheng Shenglin, Liang Jian, "A New Facial Key Features Location Algorithm in Color Images," *Proc. ICSP, 2008, pp. 932-936.*

[15] Liang Tao, Hua-bin Wang, "Detecting and Locating Human Eyes in Face Images Based on Progressive Thresholding," *Proc. IEEE Int'l Conf. Robotics and biomimetics,* pp. 445-449, Dec. 2007, Sanya, China.

[16] N.M. Brooke, "Using the visual component in automatic speech recognition," *ICSLP 96, Fourth International Conference on Spoken Language*, 1996.

[17] MacLeod, A., Summerfield, A.Q "Quantifying the contribution of vision to speech perception in noise", *British Journal of Audiology,* 21, 131-141, 1987.

[18] Kaucic, R.; Reynard, D.; Blake, A., "Real-time lip trackers for use in audio-visual speech recognition ," *IEE Colloquium on Integrated Audio-Visual Processing for Recognition, Synthesis and Communication (Digest No: 1996/213),* vol., no., pp.3/1-3/6, 28 Nov 1996.

[19] Lawrence Rabiner, Biing-Hwang Juang, *Fundamental of Speech Recognition,* Pearson Edu, 2007.

[20] C. Kotropoulos, A. Tefas, and I. Pitas, "Frontal Face Authentication Using Morphological Elastic Graph Matching," *IEEE Trans. Image Processing,* vol. 9, pp. 555-560, Apr. 2000.

[21] E. Saber and A.M. Tekalp, "Frontal-View Face Detection and Facial Feature Extraction Using Color, Shape and Symmetry Based Cost Functions," *Pattern Recognition Letters,* vol. 19, pp. 669-680, 1998.

[22] K.M. Lam and H. Yan, "An Analytic-to-Holistic Approach for Face Recogniton Based on a Single Frontal View," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 7, pp. 673-686, July 1998.

[23] C. Garcia and G. Tziritas, "Face Detection Using Quantized Skin Color Regtions Merging and Wavelete Packet Analysis," *IEEE Trans. Multimedia,* vol. 1, no. 3, pp. 264-277, Sept. 1999.

[24] M. Jones and J.M. Rehg, "Statistical Color Model with Application to Sking Detection," *Technical Report Series,* Cambridge Research Laboratory, Dec. 1998.

[25] B. Menser and M. Brunig, "Locating Human Faces in Color Images with Complex Background," *Intelligent Signal Processing and Comm. Systems,* pp. 533-536, Dec. 1999.

[26] E. Saber and A.M. Tekalp, "Frontal-View Face Detection and Facial Featuer Extraction Using Color, Shape and Symmetry Based Cost Functions," *Pattern Recognition Letters,* vol. 19, pp. 669-680, 1998.

[27] K. Sobottka and I. Pitas, "A Novel Method for Automatic Face Segmentation, Facial Feature Extraction and Tracking," *Signal Processing: Image Comm.,* vol. 12, pp. 263-281, 1998.

[28] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Comput. Vis.*, vol. 1, no. 4, pp. 321–331, Jan. 1988.

[29] D. Terzopoulos and K. Waters, "Analysis and synthesis of facial image sequences using physical and anatomical models," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 569–579, June 1993.

[30] P. Delmas, P.-Y. Coulon, and V. Fristot, "Automatic snakes for robust lip boundaries extraction," in *Proc. ICASSP*, 1999, pp. 3069–3072.

[31] P. S. Aleksic, J. J. Williams, Z. Wu, and A. K. Katsaggelos, "Audiovisual speech recognition using MPEG-4 compliant visual features," *EURASIP J. Appl. Signal Processing*, vol. , pp. 1213–1227, Sept. 2002.

[32] J. Luettin, N. A. Tracker, and S. W. Beet, "Active Shape Models for Visual Speech Feature Extraction," Univ. of Sheffield, Sheffield, U.K., Electronic System Group Rep. 95/44, 1995.

[33] A. Yuille, P. Hallinan, and D. Cohen, "Feature extraction from faces using deformable templates," *Int. J. Comput. Vis.*, vol. 8, no. 2, pp. 99–111, 1992.

[34] Y. Tian, T. Kanade, and J. Cohn, "Robust lip tracking by combining shape, color and motion," in *Proc. ACCV*, 2000, pp. 1040–1045.

[35] T. Coianiz, L. Torresani, and B. Caprile, "2D deformable models for visual speech analysis," *NATO Advanced Study Institute: Speech reading by Man and Machine*, pp. 391–398, 1995.

[36] M. E. Hennecke, K. V. Prasad, and D. G. Stork, "Using deformable templates to infer visual speech dynamics," in *Proc. 28th Annu. Asilomar Conf. Signals, Systems, and Computers*, 1994, pp. 578–582.

[37] N. Eveno, A. Caplier, and P. Y. Coulon, "Key points based segmentation of lips," in *Proc. ICME*, 2002, pp. 125–128.

[38] N. Eveno, A. Caplier, and P. Y. Coulon, "Jumping snake and parametric model for lip segmentation," in *Proc. ICIP*, Barcelona, Spain, 2003, pp. 867–870.

[39] M. O. Berger and R. Mohr, "Toward autonomy in active contour models," in *Proc. ICPR*, 1990, pp. 847–851.

[40] A. Hulbert and T. Poggio, "Synthesizing a color algorithm from examples," *Science*, vol. 239, pp. 482–485, 1998.

[41] C. Tomasi and T. Kanade, "Detection and Tracking of Point Features," Carnegie Mellon Univ., Tech. Rep. CMU-CS-91-132, 1991.

[42] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. IJCAI'81*, Vancouver, BC, Canada, 1981, pp. 674–679.

[43] A. Akutsu, Y. Tonomura, H. Hashimoto, and Y. Ohba, "Video indexing using motion vectors," in *Proc. SPIE Conf. Visual Communications and Image Processing*, Boston, MA, Nov. 1992, vol. 1818, pp. 1522-1530.

[44] U. Gargi, R. Kasturi, and S. H. Strayer, "Performance characterization of video-shot-change detection methods," *IEEE Trans. Circuits Syst. Video Technol.*, vol. CSVT-10, pp. 533-544, Feb. 2000.

[45] B. S. Manjunath, J.-R. Ohm, V. V. Vasudevan, and A. Yamada, "Color and texture descriptors," *IEEE Trans. Circuits Syst. Video Technol.*, vol. CSVT-11, pp. 703- 715, June 2001.

[46] S. H. Kim and R.-H. Park, "A novel approach to scene change detection using a cross entropy," in *Proc. IEEE Int. Conf. Image Processing*, vol. 3, Vancouver, Canada, Sept. 2000, pp. 937–940.

[47] M. R. Naphade, M. M. Yeung, and B.-L. Yeo, "A novel scheme for fast and efficient video sequence matching using compact signatures," in *Proc. IS&T/SPIE Conf. Storage and Retrieval for Media Databases 2000*, vol. 3972, San Jose, CA, Jan. 2000, pp. 564–572.

[48] D. P. Huttenlocher,G.A. Klanderman, andW. J. Rucklidge, "Comparing images using the Hausdorff distance," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 850–863, Sept. 1993.

[49] S. H. Kim and R.-H. Park, "An efficient algorithm for video sequence matching using the Hausdorff distance and the directed divergence," in *Proc. SPIE Conf. Visual Communications and Image Processing 2001*, vol. 4310, San Jose, CA, Jan. 2001, pp. 754–761.