Islamic University of Gaza
Engineering Collage
Computer department

Visual Speech Recognition

Ikrami A. Eldirawy

Supervisor:

Dr. Wesam Ashour.

In Fulfillment

of the Requirements for the Degree

Masters of Computer Engineering

1432 H
May 2011

ACKNOWLEDGMENT

I deeply thank my supervisor for all of his efforts to support me during my work, his immense support not only in advice, guidance, and inspiration, but also, he was one of the main reasons that support me to complete this thesis when I have suspecting in my desire to do that.

All my respect for Professor Ibrahim Abuhaiba. The person who teaches me how to keep patient to get my aim, despite he may not know that I learned that from him, also he teaches me how to study, all of that beside the invaluable information that I got as I attended in his lectures.

To my family, my father, my mother, the rose of my life my wife, Eng. Mona, and to my little two daughters, they brings the light into my heart and mind. Thank you all for your existence in my life, which is a big reason that gives me the power to complete this work.

Above all, I thank Allah for blessing me with all these resources, favors and enabling me to complete this thesis.

Table of Contents

First Part: Lip Localization

Second part: Speech Recognition

First part: Template Method

Second part: Visual Speech Recognition Algorithm

List of Figures

List of Tables

List of Abbreviations

AAM: Active Appearance Models.

ADI: Abstract Difference Image.

ASM: Active Shape Models.

ASR: Automatic speech recognition.

CLAHE: Contrast Limited Adaptive Histogram Equalization.

CVC: Consonant-Vowel-Consonant.

DIC/DDIT: Digital Image Correlation and Differential Digital Image Tracking.

DV: displacement vector.

DVF: displacement vector field.

DWT: discrete wavelet transform.

EM: Expectation Maximization.

E-MRF: EM and Extended-Markov Random Field.

FK-NN: Fuzzy K-Nearest Neighborhood.

GMM: Gaussian Mixture Model.

HCI: Human-Computer Interfacing.

HMM: Hidden Markov Model.

HSI: hue-saturation Intensity color system.

HSL: Hue Saturation Lightness.

HSV: Hue Saturation Value.

K-NN: K-Nearest Neighborhood.

MAP: Maximum A posteriori Probability.

MLP: Multi-Layer Perceptron.

NN: Neural Network.

PDAF: Probabilistic Data Association Filter.

PF: Particle Filter.

RBF: Radial Basis Function.

RGB: Red, Green, Blue color system.

SGF: Statistical Geometric Features.

SOFM: Self-Organizing Feature Map.

TDNN: Time-Delayed Neural Network.

TSL: Tint Saturation Lightness.

V-RS: Visual-Recognition System.

التعرف على الكلام بواسطه الحركات المرئيه

مقدم من

اكرامي عبد الحليم الديراوي

الملخص

إهتم الباحثون في السنوات الأخيرة بمجال التعرف على الكلام المنطوق من خلال الصور المرئيه أكثر من السابق، ولم تحظى اللغة العربية بهذا الإهتمام لذلك بدأنا البحث في هذا المجال.

التعرف على الكلام من خلال الصوت يهتم بالخصائص الصوتية للإشارة، وهذا يتطلب أن تكون الإشارة واضحة، وهذا تم مناقشته في الباب الثاني.

في هذه الأطروحة تم التركيز على بعض الخصائص المستخرجة من سلسلة الصور المتوالية، هذه الخصائص تُدرس وتُصنف بإستخدام طرق عديدة، وأهم هذه الخصائص هي الحركة، أي الحركة الخاصة بشفاه المتكلم.

حيث تستطيع بعض الخوارزميات أن تتعامل مع حركة الشفاه المستخرجة من سلسلة الصور المتوالية، بحيث تتعرف على الكلمة المنطوقة، ولكن يسبق هذه الخطوة إستخراج صورة الفم نفسها من الصورة الكاملة، وسنقدم في هذه الأطروحة طريقة جديدة لفصل صورة الشفاه.

أحيانا لا تكفي حركة الشفاة للتعرف على الكلام المنطوق، ولكننا نحتاج إلى خصائص أخرى، ولذلك قدمت هذه الأطروحة خوارزمية متكاملة من أجل إستخراج حركة الشفاه بإستخدام طريقة تجميع الفروقات المجردة (ADI) من سلسلة الصور المتتالية للفم، وتم دعم هذه الطريقة باستخدام (correlation) لتصحيح المواضع النسبية للصور المتتالية وبعضها، والخوارزمية تستخدم أيضا مجموعة الخصائص المستخرجه الثابتة (HU) لوصف صورة تجميع الفروقات المجردة، وكذلك تستخدم ثلاث طرق مختلفة للتعرف على الكلمات، وأيضا تستخدم طريقة تسمى (CLAHE) كتقنية ترشيح لمعالجة مشاكل الاضاءة في الصور.

وتم في هذه الأطروحة التعرف على عشرة كلمات من اللغة العربيه وهي الكلمات من الرقم "واحد" الى الرقم "عشرة".

الخوارزمية في هذه الاطروحة بنيت على تجميع تفاصيل الاختلافات في متوالية من الصور للتعرف على الكلمة، وحققت نسبة نجاح 55.8%، وهي كافية في حالة تم اندماج هذه الخوارزمية في نظام تكاملي للتعرف على الكلام من خلال الصوت والصورة.

الكلمات المفتاحية: التعرف على الكلام المنطوق من خلال الصور، التعرف على الكلام المنطوق بدون صوت، التعرف على الكلام المنطوق من خلال حركات الشفاه، استخراج صورة الفم من الصورة الكاملة.

Visual speech recognition

By

Ikrami A. Eldirawy

Abstract

In recent years, Visual speech recognition has a more concentration, by researchers, than the past. Because of the leakage of the visual processing of the Arabic vocabularies recognition, we start to search in this field.

Audio speech recognition concerned with the acoustic characteristic of the signal, but there are many situations that the audio signal is weak of not exist, and this will be a point in Chapter 2. The visual recognition process focuses on the features extracted from video of the speaker. These features are to be classified using several techniques. The most important feature to be extracted is motion. By segmenting motion of the lips of the speaker, an algorithm has manipulate it in such away to recognize the word which is said. But motion segmentation is not the only problem facing the speech recognition process, segmenting the lips itself is an early step in the speech recognition process, so, to segment lips motion we have to segment lips first, a new approach for lip segmentation is proposed in this thesis. Sometimes, motion feature needs another feature to support in recognition the spoken word. So in our thesis another new algorithm is proposed to use motion segmentation by using the Abstract Difference Image from an image series, supported by correlation for registering images in the image series, to recognize ten words in the Arabic language, the words are from "one" to "ten" in Arabic language.

The algorithm also uses the HU-Invariant set of features to describe the Abstract Difference Image, and uses a three different recognition methods to recognize the words.

The CLAHE method as a filtering technique is used by our algorithm to manipulate lighting problems.

Our algorithm based on extracting the differences details from a series of images to recognize the word, achieved an overall results 55.8%, it is an adequate result for our algorithm when integrated in an audio-visual system.

Keywords: visual speech recognition, voiceless speech recognition, mouth segmentation, lip motion based speech recognition.

Chapter One

Introduction

1.1: Speech Recognition

In our daily communication humans identify speakers based on a variety of attributes of the person which include acoustic cues, visual appearance cues and behavioral characteristics (such as characteristic gestures, lip movements). In noisy environments such as bus stop, stock market or office, much of the speech information is retrieved from the visual clues.

In the past, machine implementations of person identification have focused on single techniques relating to audio cues alone (speaker recognition), visual cues alone (face identification, iris identification) or other biometrics.

Automatic speech recognition (ASR), referred to as automatic lip-reading, introduces new and challenging tasks compared to traditional audio-only ASR.

ASR uses the images sequence segmented from the video of the speaker's lips, which is the technique of decoding speech content from visual clues such as the movement of the lip, tongue and facial muscles.

ASR has recently attracted significant interest [1, 2, 3]. Much of this interest is motivated by the fact that the visual modality contains some complementary information to the audio modality[4], as well as by the way that human fuse audio-visual stimulus to recognize speech[5, 6]. Not surprisingly, ASR has been shown to improve traditional audio-only ASR performance over a wide range of conditions

Human lip reading

Speech generally is multi-modal in nature [1]. The human speech perception system fuses both acoustic and visual cues to decode speech produced by a talker. So, lip reading started when human started to know the language. In [7] the researchers found that lip information can lead to good improvement of human’s perception of speech, this improvement goes better in a noisy environment.

1.2: Computer Machine Interface

Human-Computer Interfacing (HCI) is an advance technology. While it could not be a simple process any more for researchers, it makes the life of the users more comfortable and easier. HCI using lip reading is the most advance system to communicate. During speaking, the system has the voice signals, and the sequence of images. This thesis will focus on establishing a visual-recognition system (V-RS) based on lip images sequence. This system could not be a standalone system, because it needs the acoustic part system. V-RS contribute a wide area of researches in the recent decay, as the need for robust system to work in different conditions like: crowd, weak voice, large distance between system and user. The system in this thesis relies on two principles: first is that the camera is fixed somewhere to capture the face region. The second: the speaker has to record his vocabularies several times before using the system, because our system is a user-dependant system.

1.3: Lip Segmentation, Tracking

Lip localization allow us to find the basic and major points on mouth area, that will be used later with the lip tracking to extract necessary information about lip contour and shape. To detect lip region much systems use a skin-color model that first applied in the system to locate the face region. The lips image is then extracted from the face image. Other systems use an early lip detection stage. These systems use techniques like watershed segmentation technique to detect lip region directly.

Lip contours detection and tracking has been extensively studied in recent years because it can significantly improve the performance of an automatic speech recognition and face recognition systems, especially in noisy environments.

In Chapter 2, many methods are shown to clarify how the lip segmentation and tracking are done in real life.

1.4: Visual Feature Extraction

Lip boundary tracking allows accurate determination of a region of interest from which we construct pixel-based features that are robust to variation in scale and translation. Motivated by computational considerations, we need to select a subset of the pixels in the centre of the inner mouth area that was found to capture sufficient details of the appearance of the teeth and tongue for assisting in the discrimination of spoken words.

In general, when the input data to an algorithm is too large to be processed and it is suspected to be notoriously redundant (much data, but not much information) then the input data will be transformed into a reduced representation set of features (also named features vector).

Transforming the input data into the set of features is called feature extraction. If the features extracted are carefully chosen it is expected that the features set will extract the relevant information from the input data in order to perform the desired task using this reduced representation instead of the full size input.

The preceding paragraph is a general speaking on the feature extraction process and its importance. In image processing, as a specific speaking, the importance is growing, because of the huge amount of data in each image which may simply results in a huge amount of error results. To understand this, imagine when using the raw pixels, how it is difficult to keep each pixel fixed during analysing a series of images, despite the registration methods we may use.

Visual features usually extracted from the original images of the user face and consist of shape parameters that describe the lip boundary which describe the mouth area. Sometimes we need pre-processing techniques before extraction of the features, such as: Enhancement the images, or extraction of the motion parameters, which comes before the features extraction process in our work.

1.5: Motion as another Feature to Detect

Motion analysis long used to be a specialized research area that had not much to do with general image processing. This separation had two reasons. First, the techniques used to analyze motion in image sequences were quite different. Second, the large amount of storage space and computing power required to process image sequences made image sequence analysis available only to a few specialized institutions that could afford to buy the expensive specialized equipment. Both reasons are no longer true, because of the general progress in image processing. The more advanced methods used in motion analysis no longer differ from those used for other image processing tasks. The rapid progress in computer hardware and algorithms makes the analysis of image sequences now feasible even on standard personal computers and workstations.

Therefore we treat motion in this thesis as just another feature that can be used to identify, characterize, and distinguish objects, or understand scenes, or understand words. Motion is indeed a powerful feature.

Image sequence analysis allows us to recognize and analyze dynamic processes. Thus the many capabilities become available for scientific and engineering applications including the study of flow, transport, biological growth processes from the molecular to the ecosystem level, and of course, speech recognition, in short, everything that causes temporal changes or makes them visible in our world is a potential subject for image sequence analysis.

The analysis of motion is still a challenging task and requires some special knowledge. Therefore we discuss the basic problems and principles of motion analysis in Chapter 2. There we run on the various techniques related works for motion determination. As in many other areas of image processing, the literature is swamped with a multitude of approaches.

Motion segmentation and image segmentation

Object segmentation aims to decompose an image to objects and background. In many computer vision algorithms this decomposition is the first fundamental step. It is an essential building block for robotics, inspection, metrology, video surveillance, video indexing, traffic monitoring, speech recognition and many other applications. A great number of researchers have focused on the segmentation problem and this testifies the relevance of this topic. However, despite the vast literature, performances of most of the algorithms still fall far behind human perception.

Almost all the information that can be extracted from a single image has been used for object segmentation: textures (or more in general, statistical descriptors), edges, colors, etc.

The list of approaches is fairly long: typical segmentation techniques such as region growing, splitting and merging, watershed, histogram based algorithms [8], or neural networks [9], active contours [10], graph partitioning [11], and level sets [12, 13], are some of the most famous.

At the end, in the motion segmentation field, an object can be defined as anything that is of interest for further analysis with the final aim of object segmentation, but this object is considered to be moving. So in this thesis, the mouth pronouncing a number between one, to ten in Arabic is an object we want to segment. This gives a reasonable reason for taking in Section 1.3 about the lip tracking.

Thus, the way the object represented by the algorithms could be used as a classification parameter. However, the literature on motion segmentation is wide and using only this criterion would lead to a classification where techniques based on very different principles are grouped together. In order to make the overview easier to read and to create a bit of order, the approaches will be divided into categories which represent the main principle underlying algorithm.

This division is not meant to be tight; in fact some of the algorithms could be placed in more than one category. The groups identified are:

- Image Difference
- Statistical, further divided into:
    - Maximum A posteriori Probability (MAP)
    - Particle Filter (PF)
    - Expectation Maximization (EM)
- Optical Flow
- Wavelets
- Layers
- Factorization

More related works are proposed in Chapter 2. This thesis focuses on the Image Difference Method, study it in more details, use the main idea of this method, and suggest possible improvements.

1.6: Participations in the Big World

In this thesis, we propose more than one topic to research, all are related to the automatic speech recognition based on lip information, and movements. We research in the localization methods and proposed a new technique to localize the lips as an early stage, without localizing the face first. The main research topic in this thesis is detecting motion feature from a series of lip images, we propose a new algorithm depending on the ADI technique, improve the ADI method, and support it with several methods such as: correlation, filtering. Finally we used the Hu set of features to represent the motion image and chosen five of them. The last research topic in this thesis is to find the best way to compare between the speaker's words to recognize what word he says.

In the practical experiments we consider the Arabic language as the main language to work. The dataset is the numbers between “one” to “ten” in the Arabic language. Later in Chapter 4 we will describe in details this dataset.

1.7: Thesis Organization

This thesis is organized as follows: Chapter 1 is the introduction. Chapter 2 describes related work. Chapter 3 describes the lip localization techniques, and the improvements that we propose to get a robust technique. Also Chapter 3 discusses the main aim of the thesis, which is lib motion detection to recognize what speaker says. In Chapter 3, also we discuss some main techniques that are used to extract features. Finally, Chapter 4 gives the training and testing results and their analysis. Conclusion and future works are in Chapter 5.

Chapter Two

Related Works

In recent years, together with the investigation of several acoustic noise reduction techniques, the study of systems that combine the audio and visual features emerged as an attractive solution to speech recognition under less constrained environments. In this chapter we show extensive efforts that researchers have in the field of ASR. We cover a lot of these efforts to know how they work, and to know the topics related to ASR. After this coverage we will know that the following categories are related to ASR: lip localization and tracking, motion detection, visual feature detection, and finally classification techniques. It is important to indicate that the use of visual information needs fast and accurate lip tracking algorithms to achieve efficient results. When we investigate the lip localization and tracking algorithms, found much algorithms that segment and track the face and lip, and much of them define geometric features including the height and width of the lips automatically. In the next subsections, we show, in a glance, some previous work that manipulates these points.

2.1: Human Lip Reading

Conditions that affect human lip reading are mentioned and studied in [14], which include illumination, distance from the speaker, detection of teeth and tongue. Another factor stated in [14] that using frontal views of the speaker is better than using other view angles and give a higher accuracy of lip reading. In general, it found in many researches that contribution of visual information to speech perception has been affected by a wide variety of conditions: noise [15], highly complex sentences [16], conflicting auditory and visual speech [17, 18], and with asynchronous auditory and visual speech information [19]. Under all these conditions, improvement to speech perception was observed. The question now is why the improvement of speech perception by lip reading happens? The answer can be found in [15] as follows: visual speech predominantly provides information about the place of articulation of the spoken sounds. Human observers may thus pay attention to the correct signal source. Alternative reasons are in [20, 21] that movements of the articulators naturally accompany the production of speech sound. Human observers use these two sources of speech information from an early age and thus they can fuse the two types of information quickly and accurately.

A detailed study on the relationship between visual speech and acoustic speech was carried out in [5]. The famous "McGurk effect" indicates that human perception of speech is bimodal in nature. When human observers were presented with conflicting audio and visual stimuli, the perceived sound may not exist in either modality. For example, when a person heard the sound /ba/ but saw the speaker saying /ga/, the person might perceive neither /ga/ nor /ba/. Instead, what he perceived was /da/. Table 1.1 gives some examples of the McGurk effect.

Table 2.1: Examples of the McGurk effect

Audio	Visual	Perceived
ba	ga	da
pa	ga	ta
ma	ga	na

Human lip reading development took a wide range of research efforts since 1980s. Since that date, the researchers started to focus on this branch of science, "speech processing".

2.2: Machine-Based Lip Reading

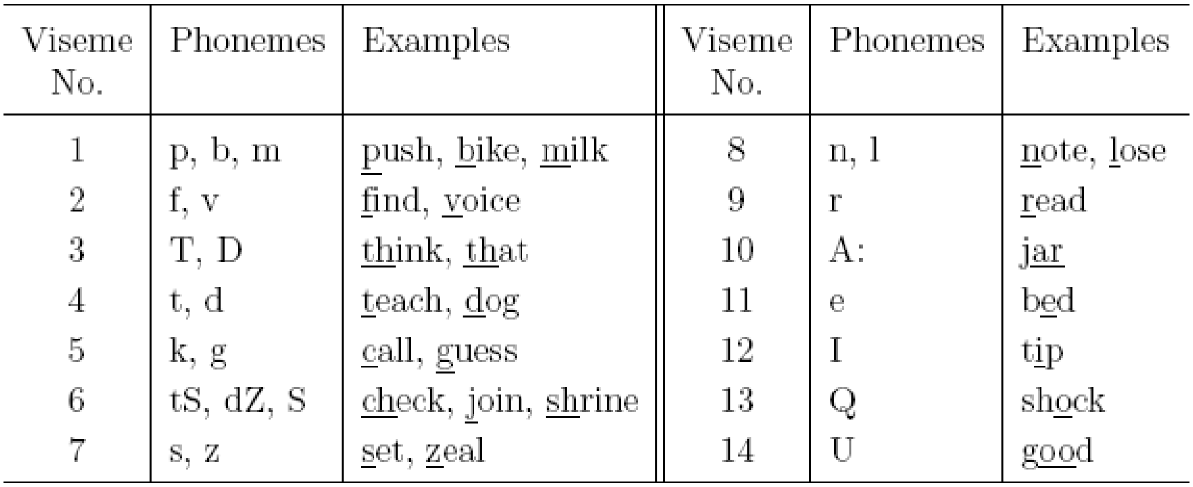
To convert the captured videos to speech information, the following processing must be undertaken: image processing, feature extraction, sequence modeling/identification, speech segmentation, grammar analysis and context analysis. If any of the composite processing modules malfunctions, the overall performance of lip reading becomes unreliable. In addition, the above mentioned processing units are inter-dependent. The individual processing units should have the ability to respond to the feedback from the other units. The difficulties involved in machine-based lip reading are even more enormous if the distinct features of lip dynamics are considered. An important factor is that the movement of the lip is slow compared with the corresponding acoustic speech signal. The low frequency feature of the lip motion indicates that the amount of information conveyed by the visual speech is very much smaller than that by the speech sound. Another important factor, which is the variations between consecutive frames of visual images are small while such variation is important for recognition because they serve as the discriminative temporal features of visual speech. Also, the visual representations of some phonemes are confusable. It is commonly agreed that the basic visual speech elements in

English, which are called visemes (the concepts about viseme are explained in detail in Section 2.3), can be categorized, for English language, into 14 groups, while there are 48 phonemes used in acoustic speech. For example, phonemes /s/ and /z/, in English of course belong to the same viseme group. As a result, even if a word is partitioned into the correct viseme combination, it is still not guaranteed that the correct word can be decoded. An important note is that the visemes are easily distorted by the prior viseme and posterior viseme. The temporal features of a viseme can be very different under different contexts. As a result, the viseme classifiers have stricter requirement on the robustness than the phoneme classifiers. Although there are many difficulties in machine-based lip reading, it does not mean that efforts made in this area are not worthwhile. Many experiments proved that even if a slight effort was made toward incorporation of visual signal, the combined audio-visual recognizer would outperform the audio-only recognizer [22, 1, 23]. Some speech sounds, which are easily confused in the audio domain such as "b" and "v", "m" and "n", are distinct in the visual domain [6]. These facts indicate that the information hidden in visual speech is valuable. In addition, many potential applications of visual speech such as in computer-aided dubbing, speech-driven face animation, visual conferencing and tele-eavesdropping stimulate the interest of researchers. With the aid of modern signal processing technologies and computing tools, lip reading became a feasible research area and much inspiring work has been done on the theoretical aspects and applications of automatic lip reading. According to the order of the implementation of lip reading, the previous work concentrated on the following three aspects: 1) Lip tracking, 2) Visual features processing, 3) Language processing. In our research we will only focus on the first two aspects.

2.3: Viseme

In visual speech domain, the smallest visibly distinguishable unit is commonly referred to as viseme. A viseme is a short period of lip movement that can be used to describe a particular sound. Like phonemes which are the basic building blocks of sound of a language, visemes are the basic constituents for the visual representations of words. The relationship between phonemes and visemes is a many-to-one mapping.

Table 2.2: Visemes defined in MPEG-4 Multimedia Standards.



Viseme No.	Phonemes	Examples	Viseme No.	Phonemes	Examples
1	p, b, m	push, bike, milk	8	n, l	note, lose
2	f, v	find, voice	9	r	read
3	T, D	think, that	10	A:	jar
4	t, d	teach, dog	11	e	bed
5	k, g	call, guess	12	I	tip
6	tS, dZ, S	check, join, shrine	13	Q	shock
7	s, z	set, zeal	14	U	good

For example, although phonemes /b/, /m/, /p/ are acoustically distinguishable sounds, they are grouped into one viseme category as they are visually confusable, i.e. all are produced by similar sequence of mouth shapes. An early viseme grouping was suggested by Binnied *et al* in 1974 [16] and was applied to some identification experiments such as [24]. Viseme groupings in [25] are obtained by analyzing the stimulus-response matrices of the perceived visual signals. The recent MPEG-4 Multimedia Standards adopted the same viseme grouping strategy for face animation, in which fourteen viseme groups are included [26]. Unlike the 48 phonemes in English [27], the definition of viseme is not uniform in visual speech. In the respective researches conducted so far, different groupings may be adopted to fulfill specific requirements [28]. This fact may cause some confusion on evaluating the performance of viseme classifiers.

MPEG-4 is an object-based multimedia compression standard and plays an important role in the development of multimedia techniques.

The fourteen visemes defined in MPEG-4 are illustrated in Table 2.2. It is observed that each viseme corresponds to several phonemes or phoneme-like productions. Note that some consonants are not included in the table as their visual representations are chiefly determined by their adjoining phonemes and diphthongs are also not included as their visual representations are assumed to be combinations of the visemes illustrated in the table. The visemes are liable to be distorted by their context. For example, the visual representations of the vowel /ai/ are very different when extracted from the words *hide* and *right*. A viseme thus demonstrates polymorphism under different contexts.

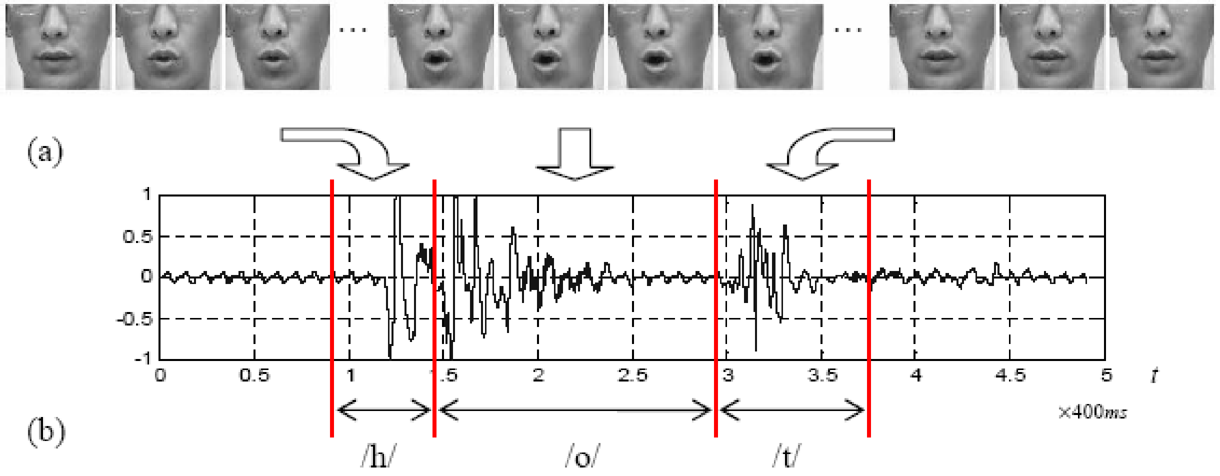


Figure 2.1: Segmentation of a viseme out of word production. (a) Video clip. (b) Acoustic waveform of the production of the word hot.

2.4: Lip Segmentation, and Tracking

The purpose of lip tracking is to provide an informative description of the lip motion. The raw input data to the lip reading system are usually video clips that indicate the production of a phoneme, word or sentence. The most direct means is to gather the color information of all the pixels of the image and feed them into the recognition modules. Actually, this was done by Yuhas *et al* [29]. The advantage of this approach is that there is no information loss during recognition. However, two disadvantages of the method. First, the computations involved in processing the entire frame are intolerable. Second, this method is very sensitive to the change of illumination, position of the speaker's lips and camera settings. The initial attempts on lip feature extraction were chiefly individual-image-oriented methods. By analyzing the color distribution of the image, the lip area was segmented by some image processing techniques. To improve the accuracy of image segmentation, image smoothing, Bayes thresholding, morphological image processing, and "eigenlip" method were all used [30-32]. These approaches treated the video as a series of independent images. The geometric measures extracted from one frame were not relevant to the other frames. The individual-image-oriented approaches had the advantage of easy implementation and many mature image processing techniques could be adopted. However, the features obtained in this way might not be accurate enough and the continuity was not good. Much of the recent work in visual analysis has centered on deformable models. The snake-based methods fit into this category. Snake was first proposed by Kass *et al* [33]. It allows one to parameterize a closed contour by minimizing an energy function that is the sum of the internal energy and external energy. The internal energy acts to keep the contour smooth while the external energy acts to attract the snake to the edges of the image. The curves used as “snakes” can be Bsplines [34, 35], single-span quadratics and cubic splines, e.g. Bezier curves [36]. Further researches were carried out to improve the performance of the snakes such as the robustness, continuity or viscosity. For example, surface learning [37, 38] and flexible appearance models [39] were adopted in snake-fitting. Deformable template algorithm is another deformable model approach. The method was proposed by Yuille *et al* [40] and was applied to capture lip dynamics by Hennecke *et al* [41]. Like snakes, the deformable templates also give an energy function for parameter adjustment. Besides this, it provides a parameterized model that imposes some constraints on the tracking process. The prior knowledge about the tracked object is revealed by the initial settings of the template. When applied to lip tracking, the templates that describe the lip contour may be simple, e.g. several parabolas [42]. Many researchers have used deformable templates to achieve good results in visual speech recognition. Several extensions to the method have also been studied. Kervrann *et al* suggested incorporating Kalman filtering techniques into deformable templates [43]. The method was also extended from 2D to 3D by Lee *et al* [44].

The two deformable model approaches mentioned above are continuous-image-oriented methods. In the tracking process, the relation between continuous frames is taken into consideration. As a result, the geometric features obtained demonstrate good continuity for continuous flow of images.

Other lip tracking approaches include Active Shape Models (ASMs) and Active Appearance Models (AAMs). The ASM was first formulated in 1994 [45] and was introduced to lip reading by Luettin *et al* [46]. The ASM is a shape-constrained iterative fitting algorithm. The shape constraint comes from the use of a statistical shape model which is called point distribution model. In the ASM tracking process, the conventional iterative algorithm [45], simplex algorithm [47] or multi-resolution image pyramid algorithm [48] could be applied. The AAM was proposed by Cootes *et al* [49]. It is a statistical model of both shape and gray-level appearance. The fitting process of AAM is largely similar to that of the ASM where iterations were implemented to minimize the difference between the target image and the image synthesized by the current model parameters. Like the deformable models, ASM and AAM approaches also focus on the changes between consecutive images. As a result, the features extracted also demonstrate good continuity. The ultimate goal of visual speech processing is to decode speech content from the lip motion. Lip tracking accomplishes the first half of the task, in which the raw image sequence is converted into tractable feature vector sequence. Subsequent processing will be carried out to extract the information conveyed by the decoded feature vectors.

2.5: Feature Processing and Extraction: Motion Vector

In general it reflects the image changes due to motion during a time interval. This process has several usages. One of them is to determine optical flow that corresponds with observed motion field.

Optical flow computation is based on two assumptions:

1. The observed brightness of any object point is constant over time.
2. Nearby points in the image plane move in a similar manner (the velocity smoothness constraint).

Implementations of using optical flow are motion detection, object segmentation, and compression.

The simplest way to compute the apparent motion is accumulative difference image (ADI), to store changes from one frame to the next.

There are few studies incorporating the pure lip motion as the visual feature, some references use this motion for identification. As an example, in 2000, Frischholz and Dieckmann [74] developed a commercial multimodal approach, BioID, for a model-based face classifier, a VQ-based voice classifier, and an optical flow based lip movement classifier for verifying persons. Lip motion and face images were extracted from a video sequence and the voice from an audio signal.

In [75] again used the lip motion for identification persons. It uses a dense uniform grid of size 64X40 on the intensity lip image. This grid definition allows us to analyze the whole motion information contained within the rectangular mouth region and it has proven, in [75], its identification performance. hierarchical block are used for matching.

2.6: Automatic Lip Reading versus Speech Recognition

The literature on automatic lip reading is fairly limited compared with that on speech recognition. However, because visual speech and acoustic speech have much in common, some techniques that have achieved success in acoustic speech recognition can be applied to visual speech recognition with some modifications. These techniques/tools include Time Warping, Neural Network, Fuzzy Logic and Hidden Markov Models (HMM). Early lip reading systems only used some simple pattern recognition strategies as the designer might face severe hardware speed limitations. In some cases, a major goal of the research was simply to demonstrate the feasibility of the concept. Some scholars consider Petajan as the first researcher that systematically investigated machine-based lip reading. In his design, linear time warping and some distance measures were used for recognition [30]. Later, Mase and Pentland also applied linear time warping approach to process the feature vector sequences [50]. Although these studies laid emphasis on the time warping aspect of visual speech, the linear time warping is not an appropriate technique to process natural speech because the temporal features of natural speech are far from linear.

Dynamical time warping was used in a later version of Petajan's lip reading system [51]. With further consideration on the non-linear features of visual speech, some improvement on the recognition accuracy was observed. The Neural Network (multi-layer perceptron, MLP) was first applied to lip reading by Yuhas *et al* [52]. However, the MLP is not flexible enough for processing time sequences. In 1992, Time-Delayed Neural Network (TDNN) was explored by Stork *et al* [53]. The inputs to Stork's system were dots of the raw image. Such a design made full use of the information conveyed by the video and was computationally expensive. The recognition results of Stork's system were better than that of time warping but were sensitive to the changes of the environment. Some improved TDNN designs were proposed and further experiments were conducted by Cosi *et al* [54] and Movellan [55].

Neural Network (NN) is a classical tool of pattern recognition. It has been intensively studied for more than half a century. From primitive McCulloch-Pitts's neuron model to today's MLP with millions of neurons, the theoretical aspects and applications of NN developed very fast. There are many types of NN and training strategies available for various requirements such as MLP [56, 3], Support Vector Machines [57, 58], Radial Basis Function (RBF) [59], TDNN [60,61] and Self-Organizing Feature Maps (SOFMs) [62]. As a result, NN-based lip reading is also a promising research area. Another powerful tool for visual speech recognition is Hidden Markov Models (HMMs). The basic theory of HMM was published in a series of papers by Baum and his colleagues in the late 1960s and early 1970s. The process generated by HMMs has been widely studied in statistics. It is basically a discrete-time bivariate parametric process: the underlying process is a finite-state Markov chain; the explicit process is a sequence of conditionally independent random variables for a given state chain. HMM was first applied to lip reading by Goldschen in 1993 [63]. In Goldschen's system, HMM classifiers were explored for recognizing a closed set of TIMIT sentences. Because of its good performance and speed of computation, HMM was extensively applied to the subsequent lip reading systems for recognizing isolated words or non-sense words, consonant-vowel-consonant (CVC) syllables [64], digit set [65, 66] and [67]. In the mean time, HMM-related techniques have advanced greatly. Tomlinson *et al* suggested a cross-product HMM topology, which allows asynchronous processing of visual signals and acoustic signals [68]. Luettin *et al* used HMMs with an early integration strategy for both isolated digit recognition and connected digit recognition [69]. In recent years, coupled HMM, product HMM and factorial HMM are explored for audio-visual integration [3,70-72]. Details of the HMM-based visual speech processing techniques can be found in [73] and [74].

In addition to the techniques mentioned above, fuzzy logic was also applied to visual speech processing. In 1993, Silsbee presented a system that combined an acoustic speech recognizer and a visual speech recognizer with fuzzy logic [65].

Chapter 3 and Section 2.8 will show in details the methods to extract the motion vector, and show how to process the motion vector to get the best and easy way to recognize speech.

2.7: Feature Processing and Extraction: HU-Invariant Feature Set

Face or lips is a kind of texture, so if we want to speak about the feature extraction in the lips image processing, we have to know how to deal with texture. Before that, we have to define the texture word. Texture is actually a very hard to defined concept, often attributed to human perception, as either the feel or the appearance of object. Everyone has their own interpretation as to the nature of texture; there is no mathematical definition for texture. By way of reference, let us consider one of the dictionary definitions Oxford (1996):

texture n., & v.t. 1. n. arrangement of threads etc. in textile fabric. characteristic feel due to this; arrangement of small constituent parts, perceived structure, (of skin, rock, soil, organic tissue, literary work, etc.); representation of structure and detail of objects in art; . . .

If we change ‘threads’ for ‘pixels’ then the definition could be applied to images (except for the bit about artwork). Essentially, texture can be what we define it to be. By way of example, analysis of remotely sensed images is now a major application of image processing techniques. In such analysis, pixels are labeled according to the categories of a required application, such as whether the ground is farmed or urban in land-use analysis, or water for estimation of surface analysis, or, in our case, skin and motion in speech recognition. Skin is a kind of texture, while motion is a series of changing texture images.

In this section we are going to show in general how researchers deal with texture images. Approaches, in general, are divided into three categories, two of them are main categories, and one category is a composite of the two main categories: structural approaches, statistical approaches, and combination approaches.

The first one, **structural approaches**: which generates the Fourier transform of the image and then to group the transform data in some way to obtain a set of measurements, such as: entropy, energy, and inertia.

Liu’s features, in [77], are chosen in a way aimed to give Fourier transform-based measurements good performance in noisy conditions. Naturally, there are many other transforms and these can confer different attributes in analysis. The wavelet transform is very popular, see [78] and [79]. Other approaches use the Gabor wavelet [80, 81, 82], Gabor has the greater descriptional ability, but a penalty of greater computational complexity[83]. Markov random fields [84, 85]. [86] Includes use of Fourier, wavelet and discrete cosine transforms for texture characterization.

These approaches are structural in nature: an image is viewed in terms of a transform applied to a whole image as such exposing its structure.

Second approach is the **statistical approaches**: The most famous statistical approach is the co-occurrence matrix. [87]. It remains popular today, by virtue of good performance. The co-occurrence matrix contains elements that are counts of the number of pixel pairs for specific brightness levels, when separated by some distance and at some relative relation. Again, we need measurements that describe these matrices. We shall use the measures of entropy, inertia and energy.

Grey level difference statistics (a first-order measure) were later added to improve descriptional capability [88]. Other statistical approaches include the statistical feature matrix [89] with the advantage of faster generation

Combination approaches: The previous approaches have assumed that we can represent textures by purely structural, or purely statistical description. One approach [90](Chen, 1995) suggested that texture combines geometrical structures with statistical ones and has been shown to give good performance in comparison with other techniques. The technique is called Statistical Geometric Features (SGF), reflecting the basis of its texture description.

Some of the previous approaches are invariant to translation, and some are invariant to scaling, while the others are invariant to rotation. Here we need invariant features to both translation and scaling, we will find this in moments, HU-set of invariant features, which belongs to the statistical approaches category, these moments are invariant under translation, changes in scale, and also rotation. Moment invariants have become a classical tool for object recognition during the last 30 years. They were firstly introduced to the pattern recognition community by Hu [91], who employed the results of the theory of algebraic invariants [92] and derived his seven famous invariants to the rotation. Since then, numerous works have been devoted to the various improvements and generalizations of Hu's invariants and also to its use in many application areas. Dudani [93] and Belkasim [94] described their application to aircraft silhouette recognition, Wong and Hall [95], Goshtasby [96] and Flusser and Suk [97] employed moment invariants in template matching and registration of satellite images, Mukundan [98,99] applied them to estimate the position and the attitude of the object in 3-D space, Sluzek [100] proposed to use the local moment invariants in industrial quality inspection and many authors used moment invariants for character recognition [94,101-104]. Maitra [105] and Hupkens [106] made them invariant also to contrast changes, Wang [107] proposed illumination invariants particularly suitable for texture classification, Van Gool [108] achieved photometric invariance and Flusser et al. [109,110] described moment invariants to linear filtering.

2.8: A brief Review of Different Motion Segmentation Techniques

As stated in the introduction there are many methods to segment motion. In this thesis we chose to state the properties of the general attributes of the methods of motion segmentation, and then some of these methods are briefly delivered before starting the discussion of our method, “the accumulative different method”.

2.8.1: Attributes

The following list highlights the most important attributes of any motion segmentation algorithm.

Feature-based or Dense-based.

In **feature-based methods**, the objects are represented by a limited number of points like corners or salient points or any features that can represent the object. Most of these methods rely on computing a homography corresponding to the motion of a planar object [111].

Features represent only part of an object hence the object can be tracked even in case of partial occlusions. On the other hand, grouping the features to determine which of them belong to the same object is the main drawback of these approaches [112]. A comprehensive survey on feature-based methods can be found in [113].

Dense methods do not use only some points but compute a pixel-wise motion [111]. The result is a more precise segmentation of the objects but the occlusion problem becomes harder to solve [111]. Dense methods could be further divided in three subgroups:

1- **Primitive geometric shapes**: in this case an object is composed by many simple geometric shapes (ellipses and rectangles) connected together by joints. The motion is modeled by affine or projective transformations. Although, in theory this representation can be used also for non-rigid objects it is more naturally suited for rigid and articulated cases [114].

2- **Object silhouette and contour:** the contour is the representation of an object by means of its boundary. The region inside the contour is called silhouette. These approaches are particularly suited for non-rigid objects [114].

3- **Skeletal models: the** skeleton can be extracted from the object silhouette by the medial axis transform, as for the geometric shapes it can be used for the rigid and the articulated cases [114].

Occlusions: it is the ability to deal with occlusions.
Occlusion is the hidden parts of image or object under consideration, disappearance of these parts due to the scene conditions, or the capturing problems.
Multiple objects (S will stand for static camera): it is the ability to deal with more than one object in the scene. This includes moving camera and non-static objects. However, some algorithms can deal with multiple objects but with the static camera constraint.

Spatial continuity: this means that the algorithm models spatial continuity, i.e. each pixel is not considered as a single point but the information provided by its neighborhood is taken into account. For a motion segmentation algorithm is normal to exploit temporal continuity but not all of them use the spatial continuity.

Temporary stopping: it is the ability to deal with objects that stop temporarily.

Robustness: it is the ability to deal with noisy images.

Sequentiality: it is the ability to work incrementally, this means for example that the algorithm is able to exploit information that were not present at the beginning of the sequence.

Missing data: it is the ability to deal with missing data, i.e. features that appear and disappear. This can be due to presence of noise, occlusions or interesting points that are not in scene for the whole sequence.

Non-rigid object: it is the ability to deal with non-rigid objects.

Camera model: it says, if it is required, which camera model is used.

Furthermore, if the aim is to develop a generic algorithm able to deal in many unpredictable situations, there are some algorithm features that may be considered as a drawback. Specifically:

Prior knowledge (X): any form of prior knowledge that may be required. For example in case a method is able to segment multiple objects it may require the knowledge of the number of moving objects or the shape of the objects.
Training (T): some algorithms require a training step.
But the last two cases are not a drawback in some or specific cases.

2.8.2: Main Techniques

Statistical methods

Statistic theory is widely used in the motion segmentation field. In fact, motion segmentation can be seen as a classification problem where each pixel has to be classified as background or foreground. Statistical approaches can be further divided depending on the framework used.
Common frameworks are Maximum A posteriori Probability (MAP), Particle Filter (PF) and Expectation Maximization (EM). Statistical approaches provide a general tool that can be used in a very different way depending on the specific technique.

MAP is based on Bayes rule:

$$P(\omega_j|x) = \frac{p(x|\omega_j)P(\omega_j)}{\sum_{i=1}^{c} p(x|\omega_i)P(\omega_i)} \qquad 2.1$$

Where x is the object to be classified (usually the pixel), $\omega_1 \ldots \omega_c$ are the c classes (usually background or foreground), $P(\omega_j|x)$is the "a posteriori probability", $p(x|\omega_j)$ is the conditional density, $P(\omega_j)$ is the "a priori probability" and

$$\sum_{i=1}^{c} p(x|\omega_i)P(\omega_i) \qquad 2.2$$

is the “density function".

MAP classifies x as belonging to the class ω_i which maximizes the “a posteriori probability".

Rasmussen and Hager (2001) [115] use a MAP framework, namely they use the Kalman Filter (and the Probabilistic Data Association Filter, PDAF) to predict the most likely location of a known target in order to initialize the segmentation process. They also discuss an extension for tracking multiple objects but without estimating the number of them. Depending on the noise level, the authors use different cues for tracking such as color, shape, textures and edges. The choice of which information has to be used is not automatized yet.

Cremers and Soatto (2005) [116] use level sets incorporating motion information. The idea is to avoid the computation of the motion field which is usually inaccurate at the boundaries; instead they jointly estimate the segmentation and the motion model for each region by minimizing a functional. The algorithm is based on a geometric model of the motion, once the motion of objects deviates from the model hypothesis the segmentation gradually degrades. A main limitation in this model is that it is based on the assumption that objects do not change their brightness throughout time. This assumption is often violated especially in cluttered background. This method is able to deal with multiple objects but it requires knowing the maximum number of objects and it has problems with new objects entering the scene in locations very far from the evolving boundary.

Shen, Zhang, Huang and Li (2007) [117] also use the MAP framework to combine and exploit the interdependence between motion estimation, segmentation and super resolution. The authors observed that when the scene contains multiple independently moving objects the estimated motion vectors are prone to be inaccurate around the boundaries and occlusion regions, thus the reconstructed high-resolution image contains artifacts. On the other hand, a sub-pixel accuracy would facilitate an accurate motion field estimation and hence a better segmentation.

They propose a MAP formulation to iteratively update the motion fields and the segmentation fields along with the high-resolution image. The formulation is solved by a cyclic coordinate descent process that treats the motion, the segmentation and the high-resolution image as unknown and estimates them jointly using the available data.

Another widely used statistical method is PF. The main aim of PF is to track the evolution of a variable over time. The basis of the method is to construct a sample-based representation of the probability density function. Basically, a series of actions are taken, each of them modifying the state of the variable according to some model. Multiple copies of the variable state (particles) are kept, each one with a weight that signifies the quality of that specific particle.

An estimation of the variable can be computed as a weighted sum of all the particles. The PF algorithm is iterative, each iteration is composed by prediction and update. After each action the particles are modified according to the model (prediction), then each particle weight is re-evaluated according to the information extracted from an observation (update). At every iteration, particles with small weights are eliminated [118].

Rathi, Vaswani, Tannenbaum and Yezzi (2007) [119] unify some well known algorithms for object segmentation using spatial information, such as geometric active contours [11] and level sets [13], with the PF framework. The particle filter is used to estimate the conditional probability distribution of a group action (Euclidean or affine) and the contour at each time. The algorithm requires the knowledge of the objects shape in order to deal with major occlusion. Exploiting a fast level set implementation [120] the authors claim that the algorithm can be used in near real-time speeds (no further indications are provided).

EM is also a frequently exploited tool. The EM algorithm is an efficient iterative procedure to compute the Maximum Likelihood (ML) estimate in presence of missing or hidden data. In ML the aim is to estimate the model parameter(s) for which the observed data is most likely to belong to. Each iteration of the EM algorithm consists of the E-step and the M-step. In the E-step, using the conditional expectation the missing data is estimated. On the other hand, in the M-step the likelihood function is maximized. Convergence is assured since the algorithm is guaranteed to increase the likelihood at each iteration [121].

Stolkin, Greig, Hodgetts and Gilby (2008) [122] present a new algorithm which uses EM and Extended-Markov Random Field (E-MRF). The authors developed this method for poor visibility environment, specifically for underwater. The algorithm merges the observed data (the current image) with the prediction derived from prior knowledge of the object being viewed in order to track the camera trajectory. The merging step is driven by the E-MRFs within a statistical framework. Furthermore, E-MRFs are particularly useful in case of very noisy images.

The importance of the observed image rather than the predicted model is decided by means of two weights. This allows to have an ad hoc behavior depending on the degree of noise: if the visibility is good it is desirable to rely on observed data, while if it the visibility is bad it is necessary to make greater use of predicted information. This latter feature is an interesting ability to adapt to the different conditions but in this implementation, the parameters are selected once and they do not change dynamically if the conditions change.

Wavelets

Another group of motion segmentation algorithms that we have identified is the one based on wavelets. These methods exploit the ability of wavelets to perform analysis of the different frequency components of the images, and then study each component with a resolution matched to its scale. Usually wavelet multi-scale decomposition is used in order to reduce the noise and in conjunction with other approaches, such as optical flow, applied at different scales.

Wiskott (1997) [123] combines Gabor and Mallat wavelet transform to overcome the aperture problem. The former transform is used to estimate the motion field and roughly cluster the image, while the latter is used to refine the clustering. The main limitation of this model is that it assumes that the objects only translate in front of the camera.

A different approach is presented by Kong, Leduc, Ghosh and Wickerhauser (1998) [124] where the motion segmentation algorithm is based on Galilean wavelets. These wavelets behave as matched filters and perform minimum mean-squared error estimations of velocity, orientation, scale and spatio-temporal positions. This information is finally used for tracking and segmenting the objects. The authors claim that the algorithm is robust; it can deal with temporary occlusions and by tuning a threshold it can estimate the number of moving objects in the scene.

Optical flow

The optical flow is defined as the “flow” of gray values at the image plane. This is what we observe. Optical flow and motion field are only equal if the objects do not change the irradiance on the image plane while moving in a scene. Two classical examples where the projected motion field and the optical flow are not equal were given by Horn [125]. The first is a spinning sphere with a uniform surface of any kind. Such a sphere may rotate around any axes through its center of gravity without causing an optical flow field. The counterexample is the same sphere at rest illuminated by a moving light source. Now the motion field is zero, but the changes in the gray values due to the moving light source cause a non-zero optical flow field.

At this point it is helpful to clarify the different notations for motion with respect to image sequences, as there is a lot of confusion in the literature and many different terms are used. *Optical flow* or *image flow* means the apparent motion at the image plane based on visual perception and has the dimension of a velocity. We denote the optical flow with $\boldsymbol{f} = [f1, f2]^T$. If the optical flow is determined from two consecutive images, it appears as a *displacement vector* (*DV*) from the features in the first to those in the second image. A dense representation of displacement vectors is known as a *displacement vector field* (*DVF*) $\boldsymbol{s} = [s1, s2]^T$. An approximation of the optical flow can be obtained by dividing the DVF by the time interval between the two images. It is important to note that optical flow is a concept inherent to continuous space, while the displacement vector field is its discrete counterpart. The *motion field* $\boldsymbol{u} = [u1,u2]^T = [u,v]^T$ at the image plane is the projection of the 3-D physical motion field by the objects onto the image plane.

Like image difference, (OF) is an old concept greatly exploited in computer vision. It was first formalized and computed for image sequences by Horn and Schunck in the 1980 [126]. However, the idea of using discontinuities in the optical flow in order to segment moving objects is even older, in [126] there is a list of older methods based on this idea but they all assume the optical flow is already known. Since the work of Horn and Schunck, many other approaches have been proposed. In the past the main limitation of such methods was the high sensitivity to noise and the high computational cost. Nowadays, thanks to the high process speed of computers and to improvements made by research, OF is widely used. As an example, Zhang, Shi, Wang and Liu (2007) [127] propose a method to segment multiple rigid-body motions using Line Optical Flow [128]. Despite classical point optical flow algorithms, the line optical flow algorithm can work also when the moving object has a homogeneous surface, provided that the object edges can be identified and used as straight lines. The limitation of this method is that is only able to deal with rigid motion because it requires straight lines in order to compute the optical flow. This approach uses K-means in order to build the final clusters hence it assumes the number of moving objects (i.e. the number K of clusters) is known a priori.

Layers

The key idea of layers based techniques is to understand which are the different depth layers in the image and which objects (or which part of an articulated object) lay on which layer. This approach is often used in stereo vision as it is easier to compute the depth distance. However, without computing the depth it is possible to estimate which objects move on similar planes. This is extremely useful as it helps to solve the occlusion problem.

Kumar, Torr and Zisserman (2008) [111] propose a method for learning a layered representation of the scene. They initialize the method by first finding coarse moving components between every pair of frames. They divide the image in patches and find the rigid transformation that moved the patch from frame j to frame j+1. The initial estimate is then refined using αβ-swap and α-expansion algorithms [129].

The method perform very well in term of quality of the segmentation and is able to deal with occlusions and non-rigid motion. The authors also reported one sequence with moving (translating) camera and static background. The algorithm is able to segment the scene correctly in presence of occlusions thanks to the fact that it can deal with different camera depths. Unfortunately, there is no information regarding the performances with moving camera and non static background. The main drawback of this method is its complexity. Furthermore the accuracy of the model is paid with the difficulty to find weights that provide good results for any kind of motion.

Factorization methods

Since Tomasi and Kanade (1992) [130] introduced a factorization technique to recover structure and motion using features tracked through a sequence of images, factorization methods have become very popular especially thanks to their simplicity. The idea is to factorize the trajectory matrix W (the matrix containing the position of the P features tracked throughout F frames) into two matrices: motion M and structure S. If the origin of the world coordinate

This algorithm works for one static object viewed from a moving camera which is modeled with the simplest affine camera model: the orthographic projection. Despite the fact that this method gives the 3D structure of the object and the motion of the camera, it has evident limits: it cannot really segment (it assumes that the features belong to the same object), it can deal only with a single rigid object, it is very sensitive to noise and it is not able to deal with missing data and outliers. However, it was the first method of this family and the solution is mathematically elegant. Some approaches were proposed with the aim of improving this original solution [131-133]. Other pointed out some weaknesses of the original methods.

Anandan and Irani (2002) [134] point out that SVD is powerful at finding the global solution associated least-square-error minimization problem only when x and y positional errors in the features are uncorrelated and identically distributed. This is rarely the case in real images.

Hence, they propose a method that is based on transforming the raw-data into a covariance-weighted data space, where the components of noise in the different directions are uncorrelated and identically distributed. In the new space they apply an SVD factorization. This method can only account frame dependent 2D affine deformations in the covariance matrices.

Okatani and Koichiro Deguchi (2007) [135] present another factorization approach. They use the Gauss-Newton method (also known as alternation technique), originally proposed by Wiberg [136], applied to the matrix factorization problem. The idea is to separate the variables of a problem into two sets and estimate them alternatively similarly to what is done in the EM algorithm. Hartley and Schafializky, 2003, [137] introduce a normalization step, updated at each iteration, between the two sets. This specific alternation technique is also known as Power Factorization.

Even if none of the previous methods can be directly used for segmentation, they started a new path for structure from motion based on factorization which eventually leaded to some solutions also for the segmentation problem. It follows a review on factorization methods that can be used to solve the motion segmentation problem. A comprehensive survey on factorization methods, with more details of some of these techniques, could be found in [138].

Costeira and Kanade [139], use the same factorization technique of [130] and then they compute the shape interaction matrix Q which, among other properties, it has zero entries if the two indexes represent features belonging to different objects, non-zero otherwise. Hence, the algorithm focuses on finding the permutation of the interaction matrix that gives a block diagonal matrix structure. Unfortunately, this process is time consuming and in presence of noise the interaction matrix may have non-zero values even when it should have zero entries.

Ichimura and Tomita (2000) [140] estimate the rank r of the trajectory matrix in order to guess the number of moving objects. Then, they perform the QR decomposition of the shape interaction matrix and they select the r basis of the shape space which gives the segmentation among those features. Finally, the remaining features are also segmented by using the orthogonal projection matrix. Again, in presence of noise it is difficult to estimate the exact rank of the trajectory matrix.

Kanatani and Matsunaga (2002) [141] use the geometric information criterion (AIC) defined in [142] in order to evaluate whether two clouds of points should be merged or not. Doing so they can segment and estimate without thresholds the number of moving objects. However, this technique works well when it is free of noise.

All of the previous techniques assume that the objects have independent motions.

Zelnik-Manor and Irani (2003) [143] study the degeneracy in case of dependent motion. They propose a factorization method that consists in building an affinity matrix by using only the dominant eigenvector and estimating the rank of the trajectory matrix by studying the ratio between the eigenvalues. Sugaya and Kanatani (2004) [144] study the degeneracy but in the geometric model. This leads to a two-steps segmentation algorithm: a multi-stage unsupervised learning scheme first using the degenerate motion model and then using the general 3-D motion model.

Zelnik-Manor and Irani (2004) [145] present a dual approach to the traditional factorization technique. Traditionally, factorization approaches provide spatial clustering by grouping together points moving with consistent motions. They propose a temporal clustering by grouping together frames capturing consistent shapes. The advantages are the smaller quantity of tracked points required and the smaller dimensionality of the data. They also show that any of the existing algorithms for columns factorization can be transformed in rows factorization.

Del Bue, Llado and Agapito (2007) [146] evaluate an automatic segmentation algorithm to segment rigid motion from non-rigid 2D measurements acquired by a perspective camera. The
RANSAC algorithm [147] is used to estimate the fundamental matrix every pair of frames and to group the points into rigid and non-rigid.

Image Differences Based Methods

Cavallaro, Steiger and Ebrahimi (2005) [112] reinforce the motion difference using a probability-based test in order to change the threshold locally. As previously explained, this first step allows a coarse map of the moving objects. Each blob is then decomposed into non-overlapping regions. From each region spatial and temporal features are extracted. The spatial features used are: color (in the CIE lab space), texturedness and variance. The temporal features are the displacement vectors from the optical flow computed via block matching. The idea is that spatial features are more uncertain near edges, whereas temporal features are more uncertain on uniform areas but the union of the two should guarantee a more robust behavior. The tracking is performed by minimization of the distance between features descriptors and is also responsible for merging or splitting the regions. This technique is capable of dealing with multiple objects, occlusion and non-rigid objects. Unfortunately, the region segmentation stage is based on an iterative process which makes the algorithm time consuming. Another limitation is due to the initialization performed when a group of objects enter in the scene, in such cases the algorithm assigns them a unique label rather than treating them as separated objects.

Cheng and Chen (2006) [148] exploit the wavelet decomposition in order to reduce the typical noise problem of image difference based approaches. They compute the image difference on the low frequency sub-image of the third level of the discrete wavelet transform (DWT). On the extracted blobs they perform some morphological operations and extract the color and some spatial information. In this way each blob is associated with a descriptor that is used to track the objects through the sequence. The authors focused on segmenting and tracking human being hence they paid particular attention in selecting features that could help in this specific task; specifically they introduced some prior on the model of a human being silhouette. Doing so they can successfully track humans but of course the method loses its generality. Furthermore, no motion compensation or statistical background is built which makes the method not suitable for moving camera applications. Li, Yu and Yang (2007) [149] use image difference in order to localize moving objects. The noise problem is attenuated by decomposing the image in non-overlapping blocks and working on its average intensity value. They also use an inertia compensation to avoid losses in the tracking when the object stops temporarily. Simultaneously, the watershed algorithm is performed in order to extract the gradient of the image. Finally, the temporal and the spatial information are fused together so that the coarse map is refined using an anisotropic morphological dilation which follows the gradient of the image. This technique deals successfully with the temporary stopping problem, but its main drawbacks are the high number of parameters to be tuned and the inability to deal with moving camera.

Colombari, Fusiello and Murino (2007) [150] propose a robust statistic to model the background. For each frame a mosaic of the background is back-warped onto the frame and a binary image is obtained indicating for each pixel whether it belongs to a moving object or not. The binary image is cleaned and regions are merged. Exploiting temporal coherence the blobs are tracked through the sequence. This technique is able to deal with occlusions, appearing and disappearing objects and non-rigid motions. The background mosaic is done offline and in order to recover the motion of the camera is necessary to extract many features in the non-moving area.

Chapter Three

The Proposed Algorithm

Lip Localization and Words Recognition

This chapter is divided into two parts, the first part is to use an algorithm to segment the face then the lips based on the skin model method, a change to a part of this algorithm is done. The second part in this chapter is a new algorithm to detect a ten words in the Arabic language, from “one” to “ten” numbers.

First Part: Lip Localization

3.1: Overview

In computer vision, the detection of skin color in images is a very useful and popular technique for detecting and tracking humans. It has been found that skin color from all ethnicities clusters tightly in hue-saturation (HS)-space [151].

A lot of face or skin color modelling techniques have been proposed. Examples on these techniques: pattern recognition based strategies, region based methods, and features based methods. One of the feature based face detection methods uses skin color as a detection cue, which gained strong popularity.

Skin color has been widely used as a first primitive in human image processing systems. Problems of automatic face detection using skin color have been successfully solved, and a number of systems use skin detection to drive real time face/people trackers. Research has focused on different ways of representing the distribution of skin tone within a chosen color space using various statistical models, such as histograms and Gaussian mixtures; this then facilitates discrimination between skin and non-skin pixels.

The most popular algorithm for face localization is the use of color information, whereby estimating areas with skin color is often the first important step of such strategy. Therefore, skin color classification has become an important task. Much of the research in skin color based face localization and detection is based on RGB, HSI and YCbCr color spaces.

The accuracy of this classification depends on the scene background color distribution. A classifier can be trained off-line by having a human identify skin and non-skin pixels. Changing illumination conditions, which alter the distribution of skin color in the image over time, complicates the classification task. There are successful applications of adaptive skin color models that track the color distribution over time. The two main approaches use histograms [152], and mixtures-of-Gaussians adapted using Expectation-Maximisation [153].

There are a lot of various color-spaces with different properties such as:

- RGB
- Normalized RGB
- Hue Saturation Intensity (HSI)
- Hue Saturation Value (HSV)
- Hue Saturation Lightness (HSL)
- Tint Saturation Lightness (TSL)
- YCbCr Color Space

3.2: RGB Color Space

The RGB color space consists of three components: red, green and blue. The spectral components of these colors are combined together to produce a resultant color. The RGB model can be represented by a 3-dimensional cube with red, green and blue at the corners on each axis. Black is at the origin. White is at the opposite end of the cube. The grey scale follows the line from black to white. For example, we can get full red color by putting RGB components to (255,0,0). The RGB model simplifies the design of computer graphics systems but is not ideal for all applications. The red, green and blue color components are highly correlated.

Unfortunately, the RGB color space is not well suited to the task of skin recognition as the color and brightness are coupled together in this color space. In addition, slight changes in illumination (e.g. shadows on the skin) affect the values for red, green and blue components. Therefore, the normalized RGB and then ignoring blue component can be used to reduce the effects of brightness on the representation of color in the RGB color space [154].

In this thesis, the normalized RGB is used for skin color-space, as the normalized RGB components reduce the effect of brightness and are easily obtained through the following equations:

- $r = R/(R + G + B)$ (3.1)
- $g = G/(R + G + B)$ (3.2)

- $b = B/(R + G + B)$ (3.3)

Where R is Red component and r is normalized Red component.
G is Green component and g is normalized Green component.
B is Blue component and b is normalized Blue component.

The conversion from RGB to normalized rg is very efficient. Normalized rg color space reduces the sensitivity to illumination changes while staying very close to the usual component (RGB). The skin color variance is also much less in the normalized rg color space than in RGB color space [154]. In addition, a multivariate Gaussian distribution in a normalized RG color space can model the skin color.

As the sum of the three components R,G and B is known and equal one, the third component b is redundant and can be omitted.

Crowley and Coutaz [154] said that one of the simplest algorithms for detecting skin pixels is to use skin color algorithm. The perceived human color varies as a function of the relative direction to the illumination. The pixels for skin region can be detected using a normalized color histogram, and can be further normalized for changes in intensity by dividing by luminance. And thus an [R, G, B] vector is converted into an [R, G] vector of normalized color that provides a fast method to detect skin. This detects the skin color region that localizes face. As in [33], the output is a face-detected image that is from the skin region. This algorithm fails when there is some more skin region like legs, arms, etc.

3.3: Gaussian Mixture Model (GMM)

Gaussian mixture model, one of the parametric skin distribution modelling techniques, was used in this thesis to model face skin color. Mixture models are a semi-parametric alternative to non-parametric histogram and provide greater flexibility and precision in modelling the underlying statistics of sample data. They are able to provide a smooth approximation and tighter constrains in assigning data to trained color space regions. Therefore, using GMM can obtain good results in pixel based skin color classification.

In the normalized RG space color, the distribution of skin colors is modelled with a multivariate Gaussian mixture model (GMM), thereby the parameters of the Gaussian mixture model are estimated using the standard Expectation- Maximization (EM) algorithm. That is, each skin color value is viewed as a realization from a Gaussian mixture model consisting of Gaussian components.

To get the GMM loglikelihood value for each pixel, which represents the skin color, we need to go through the following equations [153]:

$$r = \frac{R}{R+G+B} \quad \text{where r is the normalized red component} \tag{3.4}$$

$$g = \frac{G}{R+G+B} \quad \text{where g is the normalized green component} \tag{3.5}$$

$$p(r \mid m) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-0.5(\frac{(r-\mu)^2}{\sigma^2})} \tag{3.6}$$

$$p(g \mid m) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-0.5(\frac{(g-\mu)^2}{\sigma^2})} \tag{3.7}$$

$$p(X \mid m) = p(r \mid m) * p(g \mid m) \tag{3.8}$$

The loglikelihood value for one pixel is:

$$p(X / skincolor) = \log(\sum_{1}^{m} Wm * p(X \mid m)) \tag{3.9}$$

Where: W is the mixture weight.

μ is the mean vector

σ^2 is the variance, or in general, σI is the covariance matrix

X is the pixel with the normalized components r, and g

m is the percentage of the skin pixels class to the sum of the all pixels

Note: weight, μ and σ are given in a file contains the trained values.

The GMM likelihood function is a non-linear function; therefore the direct estimation is not possible. Thus, the model training is performed with well-known iterative technique called the

Expectation Maximization (EM) algorithm, which assumes the number of component m to be known beforehand. The EM algorithm starts with a pre-defined initial model, a new model is computed in the maximization step, the maximization step computes the parameters that maximize the old model to get the new model. The new model will be taken as the starting point for the next iteration until some convergence threshold is reached.

We can scale the loglikelihood values to be between 0 and 255 (to obtain grey level) by using the following equation:

$$New\_Value = \left(\frac{LoglikelihoodValue - \min \log likelihood}{\max \log likelihood - \min \log likelihood} \right) * 255 \qquad (3.10)$$

Figure 3.1 shows the original image and Figure 3.2 shows the loglikelihood image after scaling the values between 4 and maximum loglikelihood value (which almost around 4) and put the other values (below 4) to zeros.

Pseudocode of the GMM skin model:

```
Input: all possible skin pixel colors, a wide range of non-skin color pixels
Output: the minimum and maximum probabilities of any pixel to be a skin or not, this values are normalized using equation 3.9
begin
Use Expectation Maximization (EM) algorithm to get the probabilities
end
```

3.4: How to Detect the Face Region?

Figure 3.3 shows the loglikelihood image after scaling the values between 3 and maximum loglikelihood value and put the other values (below 3) to zeros.

Figure 3.4 shows the loglikelihood image after scaling the values between 1 and maximum loglikelihood value and put the other values (below 1) to zeros.

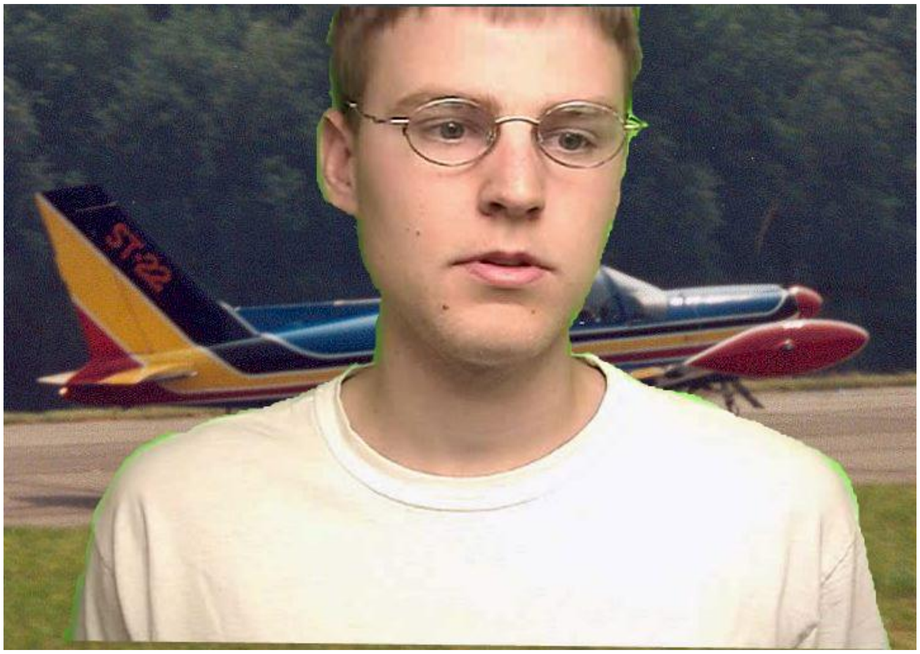


Figure 3.1: original image of a man with a complex background.

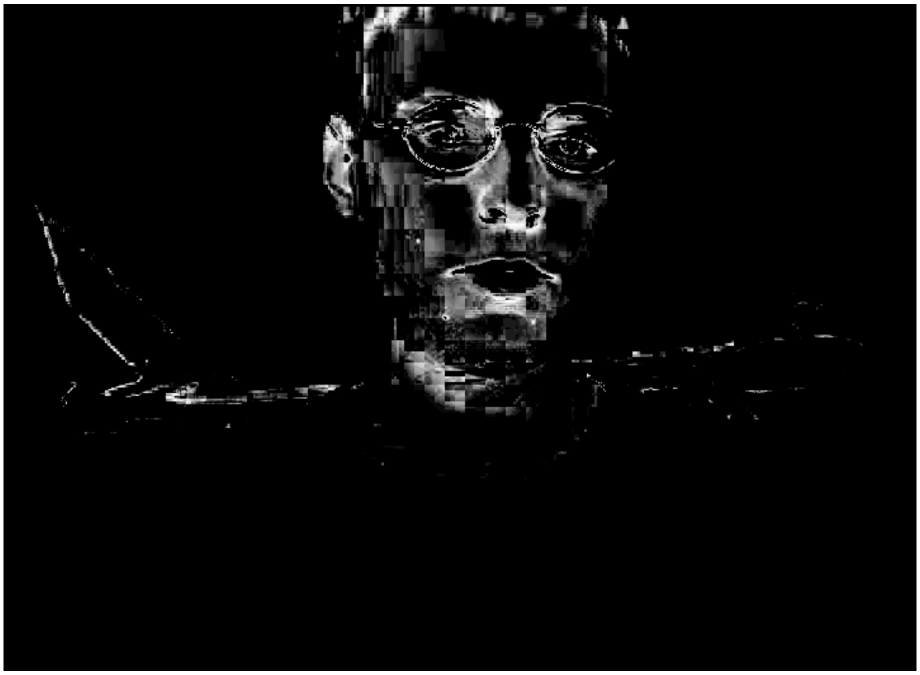


Figure 3.2: loglikelihood image (4-max. loglikelihood value, the pixel with the value between 4 and maximum are scaled depending on its probability to be a skin pixel, the other pixels are set to zero)

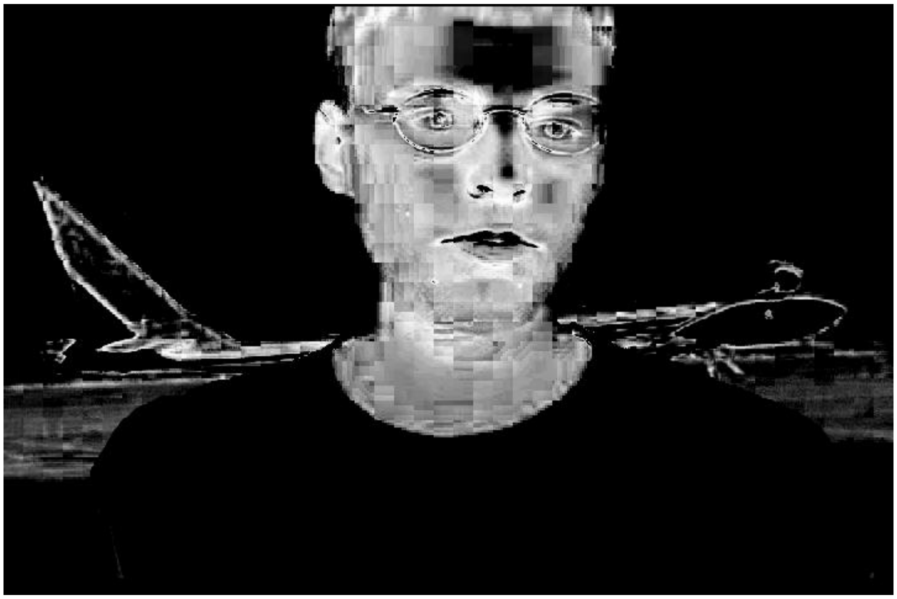


Figure 3.3: loglikelihood image (3-max. log likelihood value, the pixel with the value between 3 and maximum are scaled depending on its probability to be a skin pixel, the other pixels are set to zero)

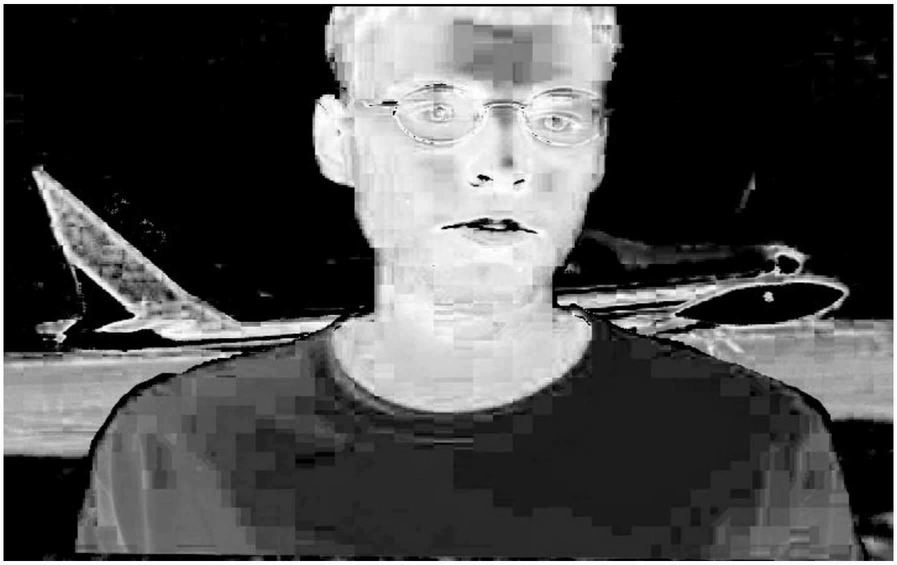


Figure 3.4: loglikelihood image (1-max. log likelihood value, the pixel with the value between 1 and maximum are scaled depending on its probability to be a skin pixel, the other pixels are set to zero)

From images in Figures 3.2, 3.3 and 3.4, we can find that the log likelihood values for the skin color always have the highest log likelihood values. Thus it is now possible to detect the face region just by applying the border tracking, which is explained in Section 3.5 .

Note: In this thesis, we provide an index counter (called e.g. bb) start from the maximum log likelihood value. If the face border is not detected at this value, the index bb is decremented by one step, this process is repeated until the face is detected.

3.4.1: How to Detect the Mouth Region?

3.4.1.1: Conditioned Tracking Border Method

Detection of mouth region starts after detecting the face region.

Figure 3.5, shows log likelihood images after setting all the log likelihood values above zero (or above bb) to 255 and the others to zeros (to convert to binary image).

Note: bb, explained in the previous section, is the index counter used for face detection.

From the images in Figure 3.5, we can notice that the mouth region (or lip corner pair) usually appears in the binary image associated by applying the loglikelihood technique for the detected face image. Thus it is possible to detect the mouth just by applying the tracking border technique, which is discussed in the next section.

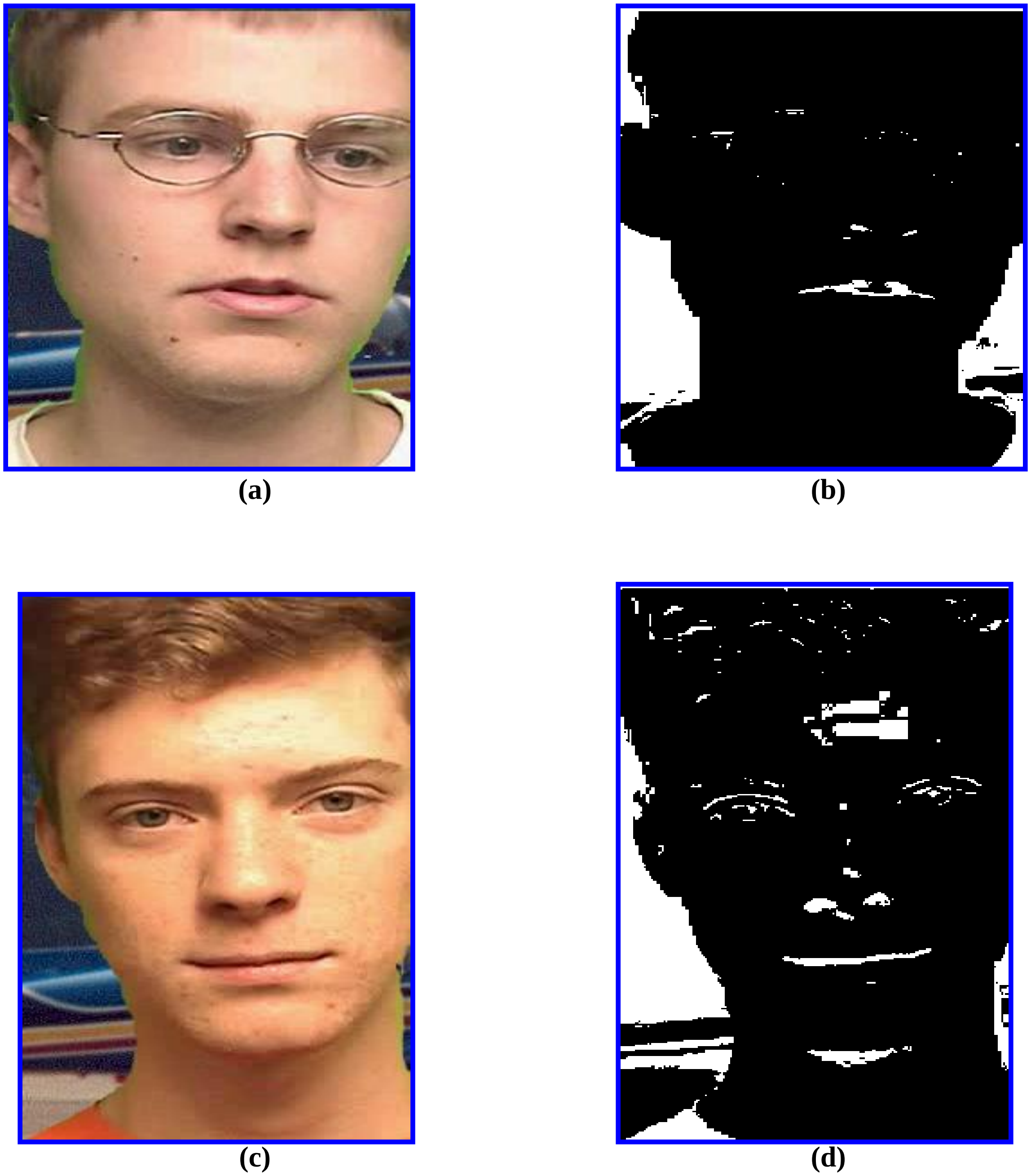
Before applying the border tracking, see Section 3.5, to detect the mouth, we need to know the following important steps:

1- We want to search for the mouth in the detected face images.

2- It is better to start scanning after skipping the upper 1/3 (or ¼) region to avoid noises that may occur due to hair, cap or anything on the head.

3- To avoid detecting eyes instead of mouth, we can start scanning from the middle region. Or for more accuracy we can specify the width of the shape that we search for (lip width). In this case, and according to the face geometry, we can choose the width that equal to the face width divided by 3, by this we can confirm that if anything is detected, this will be not eye and not nose, and therefore in usual it is the mouth.

Sometimes the mouth width is lower than the face width divided by three and hence nothing will be detected, thus to avoid this problem, we have a counter. This counter starts from the value that equal face width divided by three, and this counter decreases automatically until the lip is detected. The idea here is that we know, from the face geometric, the mouth width is always bigger than the eyes width and the eyes width will not reach the face width divided by three.



(a)

(b)

(c)

(d)

Figure 3.5: log likelihood binary images, (a) and (b) are an original image and its binary loglikelihood, (c) and (d) are another pair of original and binary images

3.4.1.2: The New Method

Modified tracking border and mouth template based method

As in the previous method, mouth detection starts after face detection. In this method we use two stages in detecting the mouth region, like in the last method. The second stage, tracking border technique see Section 3.5, is same as the last method. But the first stage assumes an average mouth, or a template, the idea of this mouth came from the modelling techniques, but it is not a model, it is a real mouth image, we prefer to name it an average mouth because it has to carry the common properties of mouths, or may, if possible, be an old mouth of the person that we are going to segment his mouth, note that we mean by the 'old mouth' that an already segmented mouth which belongs to the same speaker.

This technique almost known as template matching[155] is a technique in digital image processing for finding small parts of an image which match a template image. It can be used in manufacturing as a part of quality control[156], a way to navigate a mobile robot[157], or as a way to detect edges in images[158].

In general, template matching can be subdivided into two approaches: feature-based and template-based matching. The feature-based approach uses the features of the search and template image, such as edges or corners, as the primary match-measuring metrics to find the best matching location of the template in the source image. The template-based, or global approach, uses the entire template with generally a sum-comparing metric (using SAD, SSD, cross-correlation, etc.) that determines the best location by testing all or a sample of the viable test locations within the search image that the template image may match up to

Feature-based approach

If the template image has strong features, a feature-based approach may be considered; the approach may prove further useful if the match in the search image might be transformed in some fashion. Since this approach does not consider the entirety of the template image, it can be more computationally efficient when working with source images of larger resolution. On the other side, as in the alternative approach, template-based may require searching potentially large amounts of points in order to determine the best matching location[159].

Template-based approach

For templates without strong features, or when the bulk of the template image constitutes the matching image, a template-based approach may be effective. As aforementioned, since template-based matching may potentially require sampling of a large number of points, it is possible to reduce the number of sampling points by reducing the resolution of the search and template images by the same factor and performing the operation on the resultant downsized images (multiresolution, or pyramid, image processing), providing a search window of data points within the search image so that the template does not have to search every viable data point, or a combination of both.

Here in this thesis, it assumes, or imagine, that the face is divided to parts, each equals to the average mouth and then we compare these parts with the average mouth. The closest part will be assumed as the region of interest, which means that it may be one possible probability out of three. 1- The part is the desired mouth exactly, good work. 2- the part is a small part of the mouth, and need to enlarge the part and start the second stage. 3- the part is larger than the mouth, this happens if the mouth is not an old mouth, not belong to the current speaker, and need only to start the second stage. Finally it appears from studying the three probabilities, we need to enlarge the part by a defined number of pixels in the x-y axis, and then start the second stage.

Another advantage here that we did not do a full cross-correlation between the face and the template, but the method is to jump in steps, the length of one step equals to the length of the template. The aim of these steps is to make a rough estimation of the location of the real mouth in the face.

3.4.2: Face Geometric Method

Another general method, which depends on the geometry of the face. This method assumes that if height and width of a head is known, then it is possible to find the locations of eyes and mouth according to some measurements and calculations as illustrated in Figure 3.6 [160].

Based on reference [160], if we divide the detected face into regions then the characteristic of region that contains eyes will be similar to that in Figure 3.7 and the characteristic of region that contains mouth will be similar to that in Figure 3.7.

Now it is possible to detect the mouth region just by dividing the head into regions and then detect the region that contains the eyes and the second that contains the mouth through knowing certain characteristic shown in Figure 3.7.

The final note in this method that we propose briefly this method as an alternative way to segment the face, but we does not use it in our thesis.

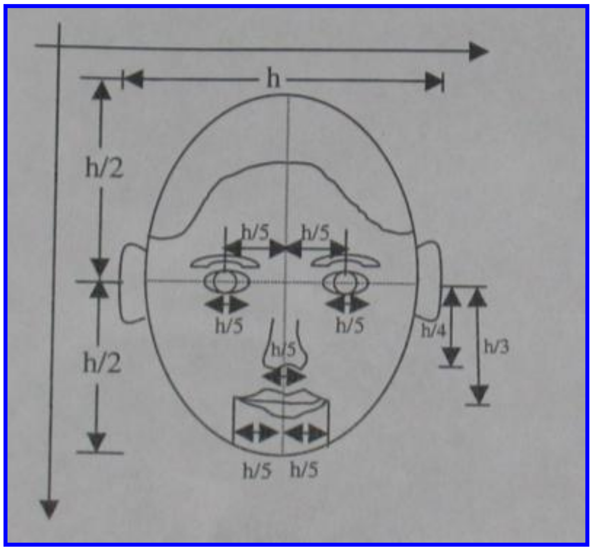


Figure 3.6: Head geometry

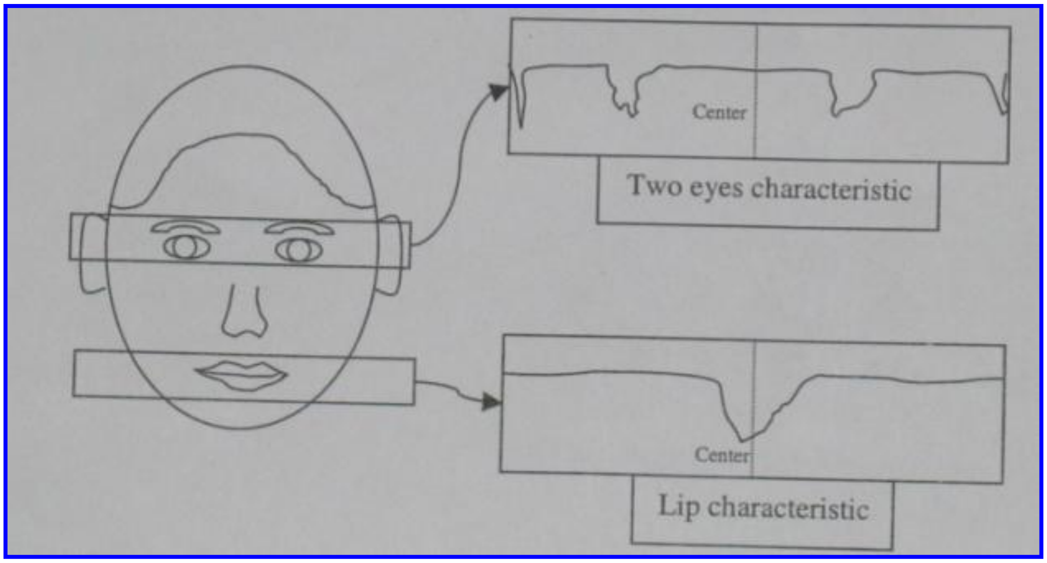


Figure 3.7: Region characteristics of eyes and mouth

3.5: Border Tracking

Border tracking is used in this chapter, see Sections 3.4 and 3.4.1, to detect the border of the face and the mouth. It is a segmentation method that follows the border around the region until it returns to the original point. It starts by scanning the image row by row and stops when white pixel, start pixel, (for binary image) is detected. Then it continuously scans the pixel's neighbourhood and proceeds to the next connected neighbour pixel when the neighbour pixel is also white. This will end when it reached back to the start pixel. Moving from one pixel to another depends on a special lookup table shown in Figure 3.8.

Border tracking algorithm can be implemented through the following steps:

1. Scan the image row by row until finding a white pixel. This white pixel holds the value 1 (or 255) in the binary image and is called start pixel.
2. Move from the start pixel to one of its neighbourhood pixels according to Figure 3.8
3. Repeat step 2 until return back to the start pixel
4. If the required border is not detected, continue searching for the next start pixel and repeat all the previous steps.

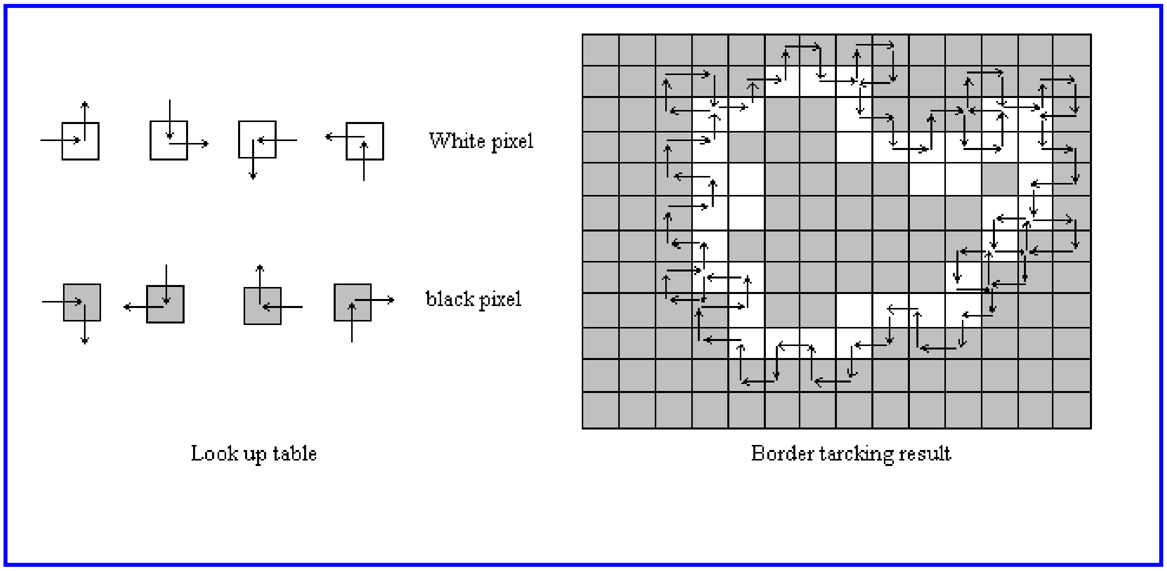


Figure 3.8: Border tracking technique

3.6: Median Filtering

This filter is applied before all the above processes to be done to remove any salt or pepper due to the capturing process.

Median filters are very effective in removing salt and pepper and impulse noise while retaining image details because they don’t depend on values, which are significantly different from typical values in the neighbourhood.

Median filters work in successive image windows in a fashion similar to linear filters. However the process is no longer a weighted sum. To define the median filter more precisely, define a window of x*y, say 3x3, neighbourhood about each pixel of the image [i, j], then we can apply the median filter at each pixel [i, j] through the following steps:

1- Sort the pixels, in the x*y window in addition to the center [i, j], in ascending order by grey level.
2- Select the middle value of the pixels as the new value for center pixel [i, j].

This process is illustrated in Figure 3.9. In general, an odd-size neighbourhood is used for calculating the median. However, if the number of pixels is even, the median is taken as the average of the middle two pixels after sorting. Figure 3.10, shows the result of median filter on an image.

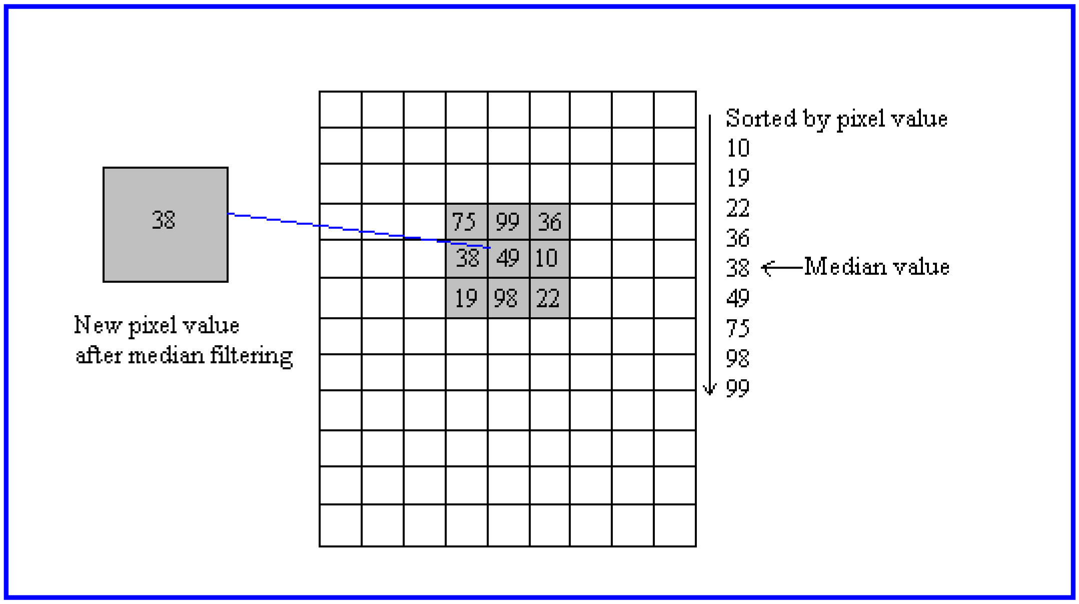
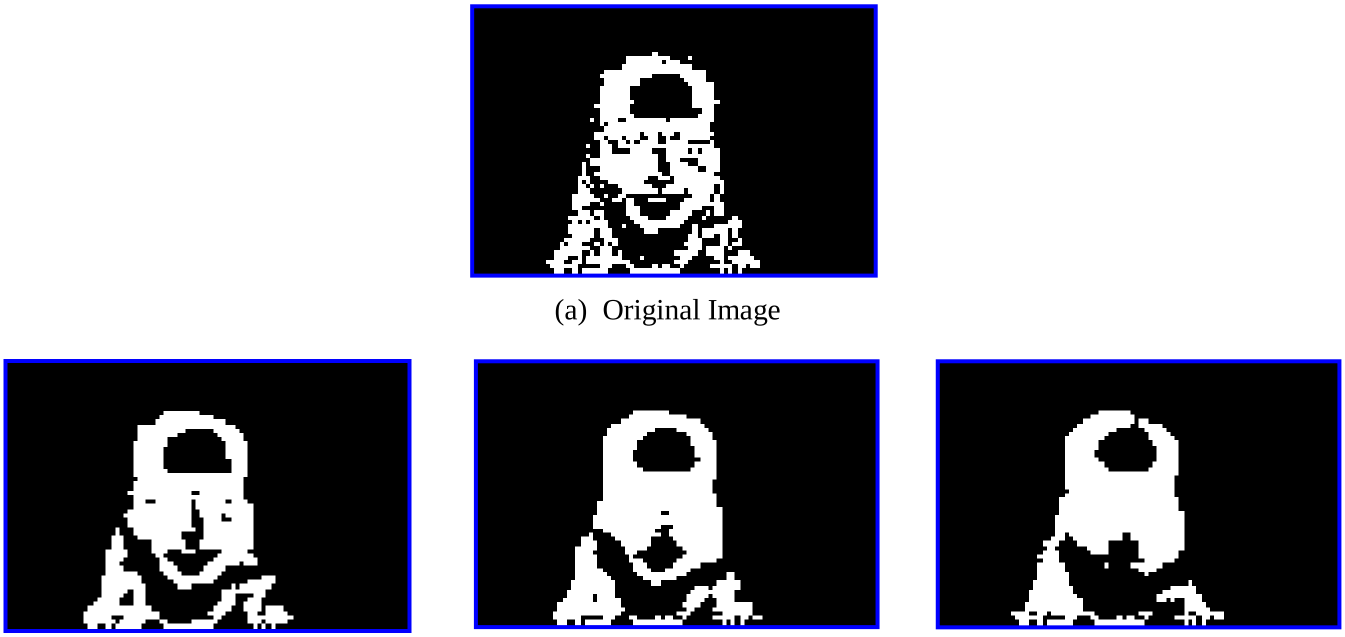


Figure 3.9: Median filter using 3x3 neighbourhood



(a) Original Image

(b) 3x3 median filter 5x5 median filter 7x7 median filter

Figure 3.10: median filters

Pseudocode of Face and mouth segmentation:

Input: The original image to be segmented and the parameters output from the GMM model

Output: the face image, the mouth image

```
begin
Use Loglikelihood to convert the input image to a loglikelihood image.
Threshold the loglikelihood image to get a binary image.
Use the median filter to remove any pepper and salt noise to recover the pepper gaps, see figure 3.12.
Use border tracking to segment face area from the binary image.
Use border tracking with the conditions in Section 3.4.1.1 to segment mouth region, or
Use border tracking after applying the method in Section 3.4.1.2 to segment the mouth region.
end
```

Finally in the previous sections of this chapter we discussed the important methods and techniques that provide the ability to detect and localize the lip region. The main idea here is to model the skin color region by applying the Gaussian mixture model (GMM) and calculating log likelihood values. As log likelihood calculations need a long time for the whole image (720x480) we need to down sample the image before the processing to save time and computational cost. A new method in this chapter was proposed to detect the mouth in the segmented face, the method is template approach based. That method showed a good results. To generalize that method to use a general image with complex background, we need more experiments to know how is the percentage of success when the image of the face come with a complex background.

Second part: Speech Recognition

3.7: General Rules

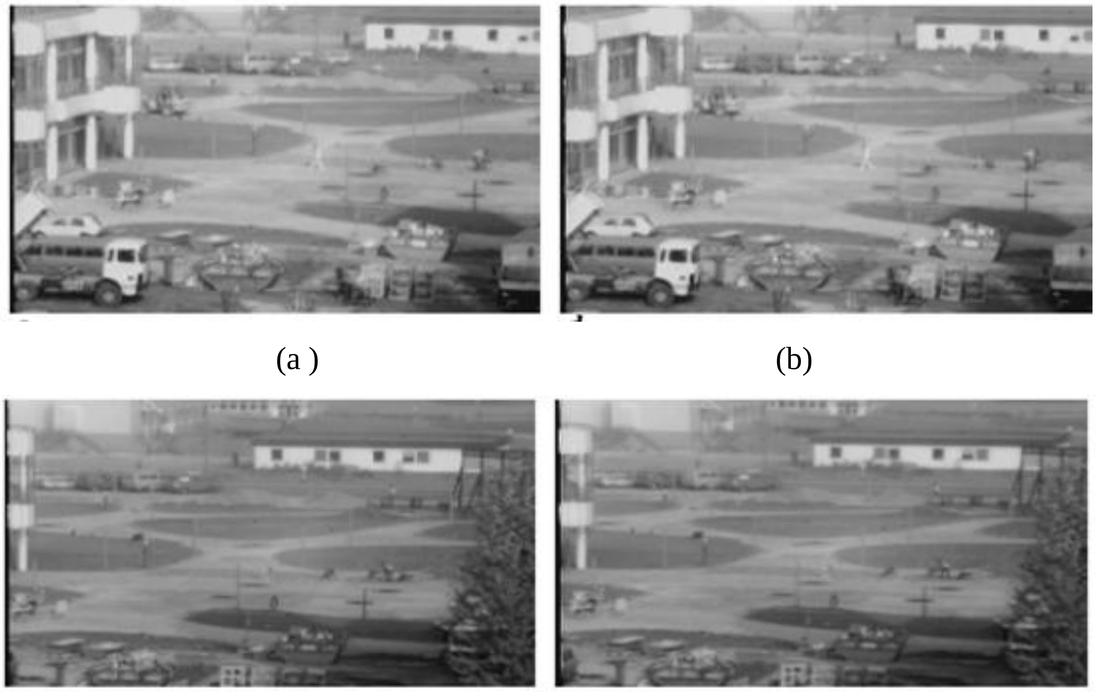
Because of the general progress in image processing, the more advanced methods used in motion analysis no longer differ from those used for other image processing tasks. The rapid progress in computer hardware and algorithms makes the analysis of image sequences now feasible even on standard personal computers and workstations.

Therefore we treat motion in this thesis as just another feature to understand scenes. Motion is indeed a powerful feature.

The following subsections give a general overview of image sequence analysis.

3.7.1: General Motion Analysis and Detection

Intuitively motion is associated with changes. Thus starting the discussion on motion analysis by observing the differences between two images of a sequence were a good start. Figure 3.11(a) and (b) show an image pair. There are differences between the left and right images which are not evident from direct comparison. However, if we subtract one image from the other, the differences immediately become visible in Figure 3.12(a),



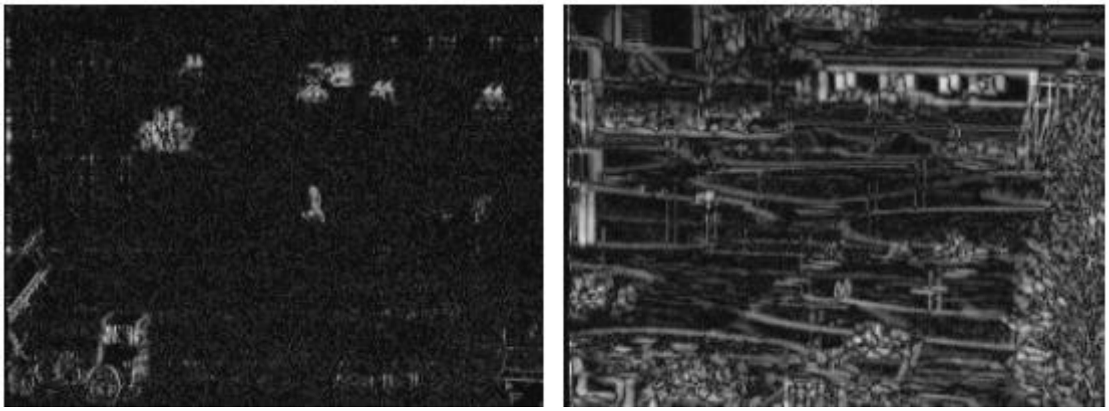
(a)

(b)

(c)

(d)

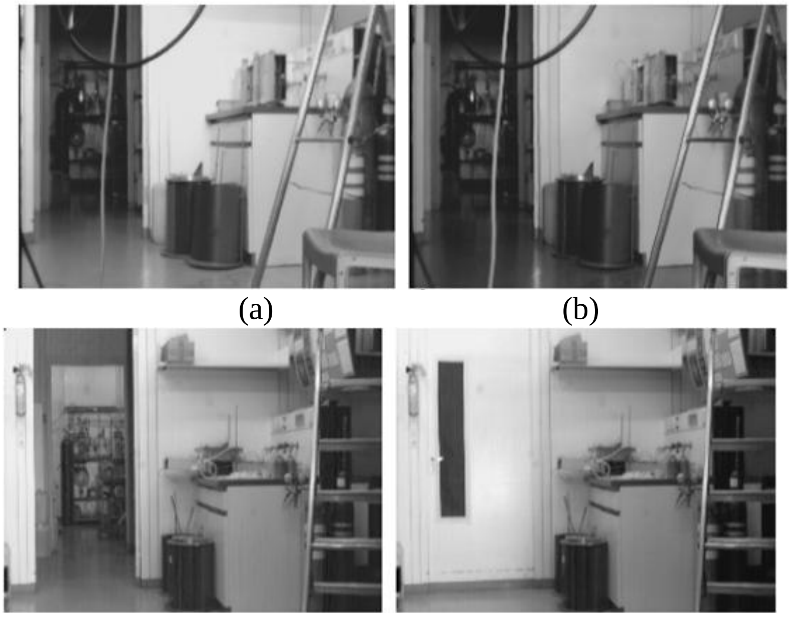
Figure 3.11: Two pairs of images. It is hard to notice the differences from the left to the right images



(a) (b)

Figure 3.12: The differences between: (a) Images a and b in Figure 3.11,
(b) Images c and d in Figure 3.11

Consequently, we can detect motion only in the parts of an image that show gray value changes. This simple observation points out the central role of spatial gray value changes for motion determination. So far we can sum up our experience with the statement that motion might result in temporal gray value changes. Unfortunately, the reverse conclusion that all temporal gray value changes are due to motion is not correct. At first glance, the pair of images in Figure 3.13(a) and (b) look identical.



(a) (b)

(c) (d)

Figure 3.13: a to d Two pairs of images from an indoor lab scene. Again,
changes cannot be seen between the left and right images

Yet, the difference image in Figure 3.14(a) reveals that some parts in the upper image are brighter than the lower. This is obviously shown because the illumination has changed and make the difference image bright. Another pair of images in Figure 3.13 (c, d) shows a much more complex scene, although we did not change the illumination. We just closed the door of the lab. Of course, we see strong gray value differences where the door is located. The gray value changes extend to the floor close to the door and to the objects located to the right of the image in Figure 3.14(b).

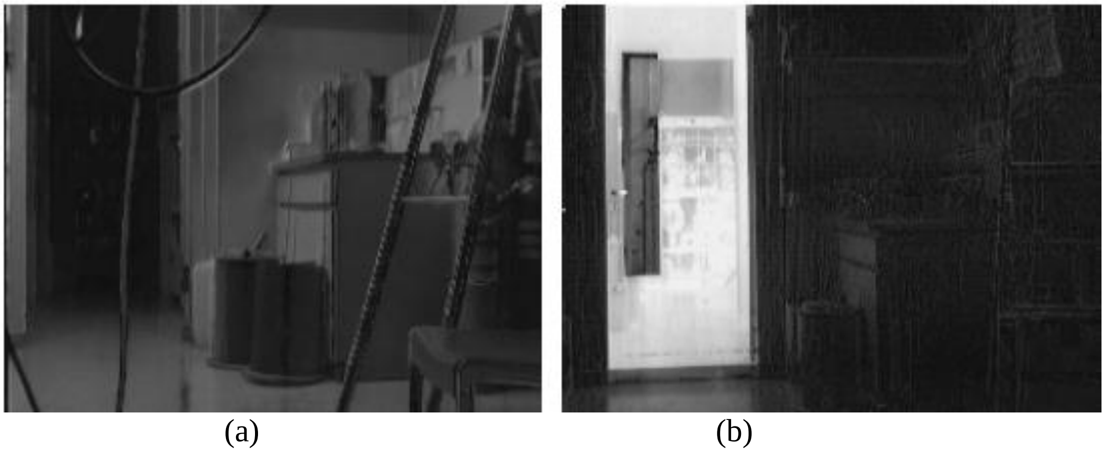

(a) (b)

Figure 3.14: Difference between (a) images a and b in Figure 3.13, (b) images c and d in Figure 3.13

As we close the door, we also change the illumination in the proximity of the door, especially below the door because less light is reflected into this area.

The aperture problem

We will not cover all the special problems that face the motion segmentation, but here we propose an example on these problems, which is the aperture problem.

The problem caused by motion of homogeneous contour, which is locally ambiguous.

Figure 3.15(a) shows that the grating appears to be moving down and to the right, perpendicular to the orientation of the bars. But it could be moving in many other directions, such as only down, or only to the right. It is impossible to determine the motion direction unless the ends of the bars become visible in the aperture. If we described the motion from one image to another by a *displacement vector*, or briefly, *DV*, we cannot determine the displacement unambiguously. The displacement vector might connect one point of the edge in the first image with any other point of the edge in the second image Figure 3.15(a). We can only determine the component of the DV normal to the edge, while the component parallel to the edge remains unknown. This ambiguity is known as the *aperture problem*.

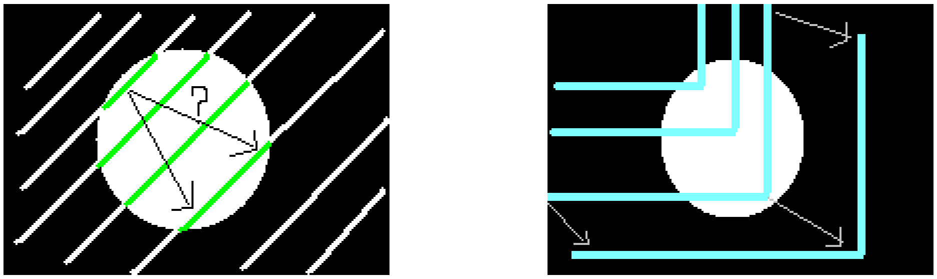


Figure 3.15 (a) An ambiguous motion (b) An unambiguous motion

An unambiguous determination of the DV is only possible if a corner of an object is within the mask of our operator Figure 3.15(b). This emphasizes that we can only gain sparse information on motion from local operators.

3.7.2: Motion and Space-Time Images

The analysis of motion from only two consecutive images is plagued by serious problems, as we see in Section 3.7.1, The Aperture Problem.

The question arises, whether these problems, or at least some of them, can be overcome if we extend the analysis to more than two consecutive images. With two images, we get just a "snapshot" of the motion field. We do not know how the motion continues in time. We cannot observe how parts of objects appear or disappear as another object moves in front of them.

Or we cannot see the mouth when open and when close. Note that in some words, the mouth is opening before closing, and do the opposite in another word. So if the mouth do the same movements, but open first, it will be very different of closing first

In this section, we consider the general idea of image sequence analysis in a multidimensional space spanned by one time space coordinates. Consequently, we speak of a *space-time image*, a *spatiotemporal image*, or simple the ***xt*** space.

We can think of a three-dimensional space-time image as a stack of consecutive images which may be represented as an *image cube* as shown in Figure 3.16. At each visible face of the cube we map a cross section in the corresponding direction. Thus an *xt* slice is shown on the top face and a *yt* slice on the right face of the cube. The front face shows the last image of the sequence. In a space-time image a pixel is extended to a *voxel*, by adding a time element to its coordinates, then the voxel coordinate is Δx, Δy, and Δt.

The previous imagination will help us to understand the idea of our work in this thesis, we consider this section is the key to imagine what we did.

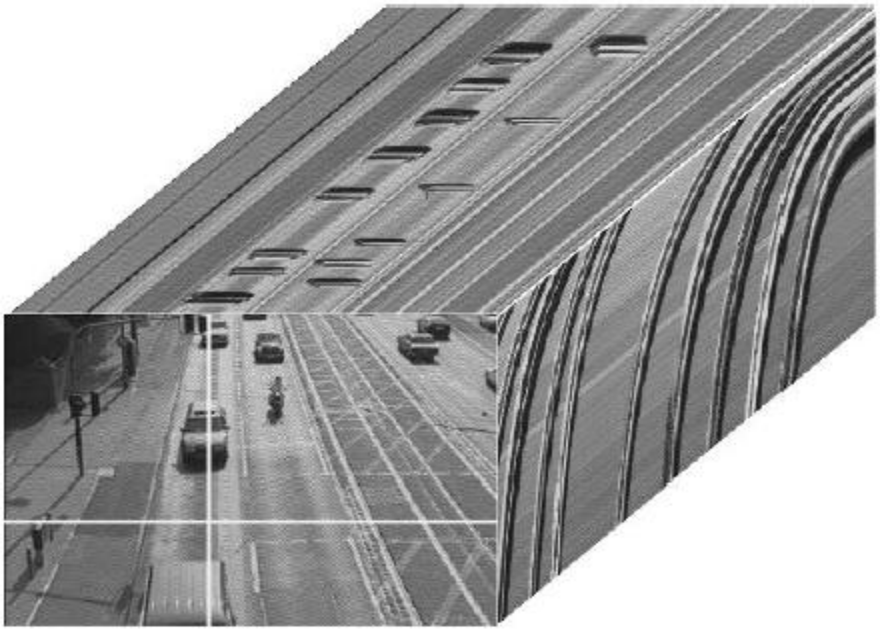


Figure 3.16: A 3-D image sequence demonstrated with a traffic scene represented as an image cuboid. The time axis runs into the depth, pointing towards the viewer. On the right side of the cube a yt slice marked by the vertical white line in the xy image is shown, while the top face shows an xt slice marked by the horizontal line "from Jahne [161]".

3.8: The New Algorithm

The new algorithm has to manipulate a speaker word. That word has to be in Arabic language, and among "one" to "ten" in the Arabic numbers. The block diagram of the system is in Figure 3.17, the figure shows the stages that our system will do. In this part of chapter we will suppose that the input to the algorithm is the series of mouth images, which is the second stage in figure 3.17.

As stated previously in the chapter we notice that an easy way is the Abstract Difference Image methods, ADI, which gives a direct impression of the motion, but it needs a cooperative method to make it more powerful.

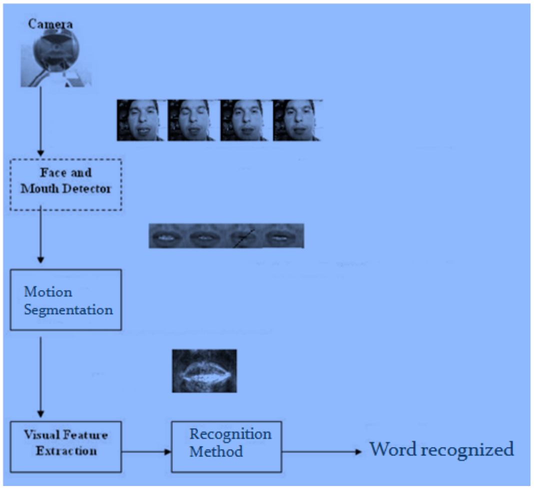


Figure 3.17: The block diagram of the system, which recognize ten numbers in the Arabic language

In the following subsections we will show how the researchers used this method, and how they developed it to make it usable. And in the same time we will show the architecture of our algorithm, how we arranged the ideas, and how we used the principles to build up our algorithm. First we will show how the ADI used in segmenting the pronounced word in its abstract manner. This will be the subject of the first subsection.

Section 3.8.1 also contains an improvement, which propose an idea on how to detect the first frame in the sequence that carries the word pronunciation. Also Section 3.8.1contains the comparing techniques which used to take the decision about the incoming word. The second section shows the principle of correlation, and how we use it together with the ADI to build our algorithm.

Our algorithm has been described briefly in the previous paragraph. This description was the main backbone of the algorithm. The final stage is an improvement to the result by using a filtering technique. This technique will be described briefly in a filtering subsection.

After completing the segmentation of the motion of the pronounced word, the result is, finally, an image that holds the coarse map of the temporal changes. This map is the goal of our motion segmentation process. It is an image that holds the sum of all motion in the sequence of images of the pronounced word. Now we have to compare this map to other maps, these maps present the numbers between “one”, to “ten”, in Arabic language, and be classified in a way that the number one, for example, is stored in a specified, and known, place in a dataset. All the numbers also are stored in the dataset in the same way. To complete the comparison process, first we apply the motion segmentation algorithm to the whole dataset to extract the maps, and then extract the feature vectors.

The feature vectors are a summarization of the image map, in this thesis we choose the 7 Hu-invariant set of features, and 5 have been chosen among them. The choice process depends on a simple algorithm that will be clarified later in a subsection. These features are stored, and be ready for later comparisons with the incoming speaker inputs.

3.8.1: Using Abstract Difference Image Method

Image difference is one of the simplest and most used techniques for detecting changes. it consists of taking the difference of intensity between every corresponding two pixels in every two successive frames, these differences are summed. The result is a coarse map of the temporal changes. An example of an image sequence and the image difference result is shown in Figure 3.18. Despite its simplicity, this technique cannot be used in its basic version because it is really sensitive to noise. Moreover, when the camera is moving, the whole image is changing and, if the frame rate is not high enough, the result would not provide any useful information. However, there are few techniques based on this idea. It follows a brief presentation of some of the recent techniques that improved the basic concept of image difference.

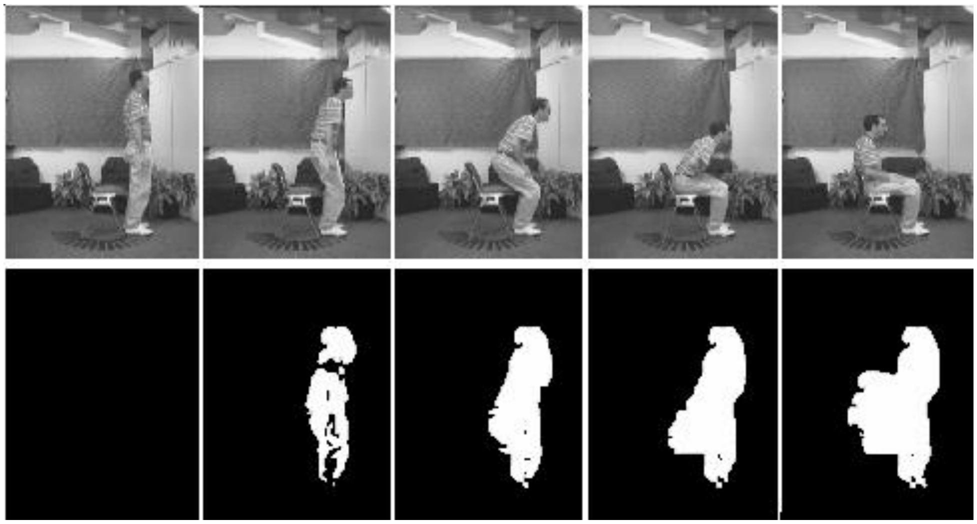


Figure 3.18: Example of image sequence and its image difference result. Sequence taken from [162]

In this section, after introducing the ADI technique, we have to explain that we use the plain ADI. This will be the basic step in our algorithm. The final result is passed along five stages.

First, filtering the image series. Second is to correlate each two images in the sequence that expresses the word, and this will be the topic of the next subsection. Third is to take the difference between each images then sum all the changes, which is the idea of the ADI itself, Fourth, the features detection process. Fifth, the comparing process, and all these topics will come in the later subsections.

The pseudocode describes the algorithm is in the results chapter.

The next subsections will explain the next steps in our algorithm.

3.8.2: Correlation Step

Principle

The correlation is one of the most common and most useful statistics. A correlation is a single number that describes the degree of relationship between two variables.

If we have a series of n measurements of X and Y written as x_i and y_i where i = 1, 2, ..., n, then the sample correlation coefficient can be used to estimate the population pearson correlation r between X and Y. The sample correlation coefficient is written as:

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i-\overline{x})(y_i-\overline{y})}{(n-1)s_x s_y} = \frac{\sum_{i=1}^{n}(x_i-\overline{x})(y_i-\overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\overline{x})^2 \sum_{i=1}^{n}(y_i-\overline{y})^2}} \qquad 3.11$$

Where x_i and $y_i$are sample points in X and Y, $\bar{x}$ and $\bar{y}$ are the sample means of X and Y, and s_x , s_y are the sample standard deviations of X and Y.

When X and Y is two images then, r_{xy} is the correlation between two images, when r_{xy} is maximum then the maximum similarity between the two images exists, we uses the cros-correlation at this point to register the two images.

In general, Digital Image Correlation and Differential Digital Image Tracking (DIC/DDIT) is an optical method that employs tracking and image registration techniques for accurate 2D and 3D measurements of changes in images. This is often used to measure deformation (engineering), displacement, and strain, but it is widely applied in many areas of science and engineering. One of its common applications is for measuring the motion of an optical mouse.

In motion segmentation, the correlation technique is an approach which originates from analyzing the displacement between two consecutive images. To find a characteristic feature from the first image within the second, we take the first image $g(t_1) = g_1$ and compare it with the second image $g(t_2) = g_2$ within a certain search range. Within this range we search for the position of optimum similarity between the two images. When do we regard two features as being similar? The similarity measure should be robust against changes in the illumination. Thus we regard two spatial feature patterns as equal if they differ only by a constant factor α which reflects the difference in illumination.

In other words, we need to maximize the *cross-correlation coefficient*, it is zero for totally dissimilar (orthogonal) patterns, and reaches a maximum of one for similar features.

The first step is to introduce a window function *w* into the definition of the cross-correlation coefficient. This window is moved around the image to compute the local cross-correlation coefficient.

In our algorithm, the window image is the next image. So the correlation is done on each two successive frame. After the two successive frames are correlated, then, the second frame takes the position of the first frame in the next iteration, and a new frame from the series considered as the second frame of the next iteration. The correlation process to get maximum similarity. This process has to be done before taking the difference in the sequence. why? This is to get rid of the small movements of the speaker. These movements make the ADI changes more complex, and hard to get any impression of the real facial movements while talking.

3.8.3: Image Enhancement and Filtering

Image processing modifies pictures to improve them (enhancement, restoration). In this thesis, we propose that we do not need restoration, we suppose that the imaging device is noise free. Image enhancement is very important technique for many applications. Image enhancement improves the quality (clarity) of images for human viewing. Removing blurring and noise, increasing contrast, and revealing details are examples of enhancement operations. Image enhancement approaches in general fits into two categories: spatial domain methods and frequency domain methods. The meaning of spatial domain is that the image domain, or image plan. That is means the methods manipulate the pixels of the image directly. On the other side, the frequency domain methods depend on the Fourier transform, first, it compute the Fourier transform, then manipulate it, finally compute back the resulting image from the Fourier transform.

There is no general theory of image enhancement. When a user looks at an image is processed, he is the only judge of how well is the results of that enhancement process. So the evaluation of the 'goodness' of an image is a relative process, which depends on the user, “evaluator”. It is easier to give a precise definition of 'goodness' when the enhancement is one of several processes in a certain application, or algorithm. In this case, which is the same case of what happens in this thesis, the judge will be the results of the application. To explain this more, when the application is lip detection, while keeping all other processes not changed and changing only the image enhancement process, if the results are better, then the image enhancement process is better, and so on.

processing in spatial domain has two methods : point methods, also named histogram statistics, and the second methods are the neighborhood operations. These methods are usually faster in computer implementation as compared to frequency filtering methods that require computation of Fourier transform for frequency domain representation. Sometimes the frequency domain methods give a better results if a prior information about the frequency component of noise is available, a model for noise, so it is better, as opinion of the author of this thesis, for restoration process. A histogram of an image provides information about the intensity distribution of pixels in the image. The histogram is expressed in terms of the probability of occurrence of gray levels as in the following equation :

$$P(r_i) = n_i/n \qquad \text{for } i= 0, 1, 2, \ldots\ldots\ldots., L-1 \qquad 3.12$$

Where L is the number of gray levels in the image, r_i is the ith gray level in the image, n_i is the number of occurrences of the gray level r_i in the image.

Several methods depend on the histogram statistics, named histogram transformations, such as: histogram equalization. Sometimes the need to modify the histogram to get some specific results, and some methods tend to break the histogram into slices and make local modifications in that slices, one of these methods is the contrast-limited adaptive histogram equalization, which used in this thesis.

In this thesis, is important to manipulate the effect of lighting. The variations in light conditions are hard to be controlled. So in this section we try to avoid, as possible, these conditions. We do not need to manipulate shadows, despite that our method, CLAHE, manipulates it. We consider the shadows has not a strong effect because the deal only with the mouth area reduces shadows appearance.

Image filtering is a process by which we can enhance (or otherwise modify, warp, and mutilate) images, so it is a kind of image enhancement. This process allows you to apply various effects on photos. There are many types of filtering methods, which may be two dimensional filtering, or three dimensional filtering. Note that the filtering approaches is that the methods in which processing is done using gray levels of pixel neighborhoods. These approaches is not in the focus of this thesis.

3.8.4: Features Detection Process

After motion has been segmented, and converted to an ADI image. This image has to be described, or summarized. In general, image description involves two methods: first, represents the image using the external characteristics. Second, represents the image in terms of its internal characteristics (the pixels comprising the image). Choosing the representation scheme depends on the task we want to do. An external representation is chosen when the primary focus is on shape characteristics. An internal representation is selected when the primary focus is on regional properties, such as color and texture. In this thesis, we need the representation scheme to be robust against translation, and scaling. This leads us to use the moments, which have this features.

Feature detection is the most important stage in our thesis. Without this stage, we have to deal with a huge amount of raw information that is conflicting, which easily lead us to false results.

An important factor in choosing the feature is its resistance to scale, rotation, and translation.

In our thesis we have chosen the Hu-set of invariant features which is defined as the following:

$$\mu_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(x-\bar{x})^{p}(y-\bar{y})^{q}f(x,y)\,dxdy \qquad 3.13$$

where $\bar{x} = \frac{M_{10}}{M_{00}}$ and $\bar{y} = \frac{M_{01}}{M_{00}}$ are the components of the centroid.

If $f(x,y)$ is a digital image, then the previous equation becomes

$$\mu_{pq} = \sum_x \sum_y (x-\bar{x})^p (y-\bar{y})^q f(x,y) \qquad 3.14$$

The central moments of order up to 3 are:

$$\mu_{00} = M_{00} \qquad 3.15$$

$$\mu_{01} = 0 \qquad 3.16$$

$$\mu_{10} = 0 \qquad 3.17$$

$$\mu_{11} = M_{11} - \bar{x}M_{01} = M_{11} - \bar{y}M_{10} \qquad 3.18$$

$$\mu_{20} = M_{20} - \bar{x}M_{01} \qquad 3.19$$

$$\mu_{02} = M_{02} - \bar{y}M_{01} \qquad 3.20$$

$$\mu_{21} = M_{21} - 2\bar{x}M_{11} - \bar{y}M_{20} + 2\bar{x}^2 M_{01} \qquad 3.21$$

$$\mu_{12} = M_{12} - 2\bar{y}M_{11} - \bar{x}M_{02} + 2\bar{y}^2 M_{10} \qquad 3.22$$

$$\mu_{30} = M_{30} - 3\bar{x}M_{20} + 2\bar{x}^2 M_{10} \qquad 3.23$$

$$\mu_{03} = M_{03} - 3\bar{y}M_{02} + 2\bar{y}^2 M_{01} \qquad 3.24$$

It can be shown that:

$$\mu_{pq} = \sum_m^p \sum_n^q \binom{p}{m}\binom{q}{n}(-\bar{x})^{(p-m)} (-\bar{y})^{(q-n)} M_{mn} \qquad 3.25$$

Central moments are translational invariant only so seven features derived of the central moment as follows:

$$
\begin{aligned}
I_1 &= \eta_{20} + \eta_{02} \\
I_2 &= (\eta_{20} - \eta_{02})^2 + (2\eta_{11})^2 \\
I_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\
I_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\
I_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + \\
&\quad (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\
I_6 &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \\
I_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + \\
&\quad (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2].
\end{aligned}
\tag{3.26}
$$

The previous moments are invariant under translation, changes in scale, and rotation, and frequently used and named "The Hu set of Invariant Features".

Five of seven hu are used. Experiments show that this is the best to describe images in our application. The choice based on the trial and error. Experiment 4.4 in Chapter 4 is repeated on the principle of all the probabilities. We try all the probabilities of the Hu-set, on each probability we do the whole stages of the recognition process. Finally, after completing the first stage of Chapter 4, which ends by Experiment 4.6, then Experiment 4.6 repeated again for all the probabilities of the seven Hu-set invariant features, we got the same results.

Chapter four

Experimental Results

In this chapter we divide the experiments into two parts: in the first part we examine the template method to modify the mouth detection methods that we use to segment the mouth. The second part concerned in the new algorithm for visual speech recognition which proposed in the previous chapter.

The dataset is composed of ten speakers, two females and eight males, each one pronounce the words from “one” to “ten” in the Arabic language fifteen times. In the first part we use the fifteen records for each speaker to try the segmentation algorithm. While in the second part, we use ten records for training and five records for testing, for every speaker.

To collect the dataset as a series of images. This series is taken as a video by using the camera of a Nokia N73 mobile with resolution to 3.2-megapixels and uses Carl Zeiss lens. After that the video is converted to a series of images by using the program Ulead video studio11.

The experiments in this chapter done on a Lenovo laptop, dual core, 2.1GHz each, 3MB ram, 500G hard disk, system is windows 7 32bit

First part: Template Method

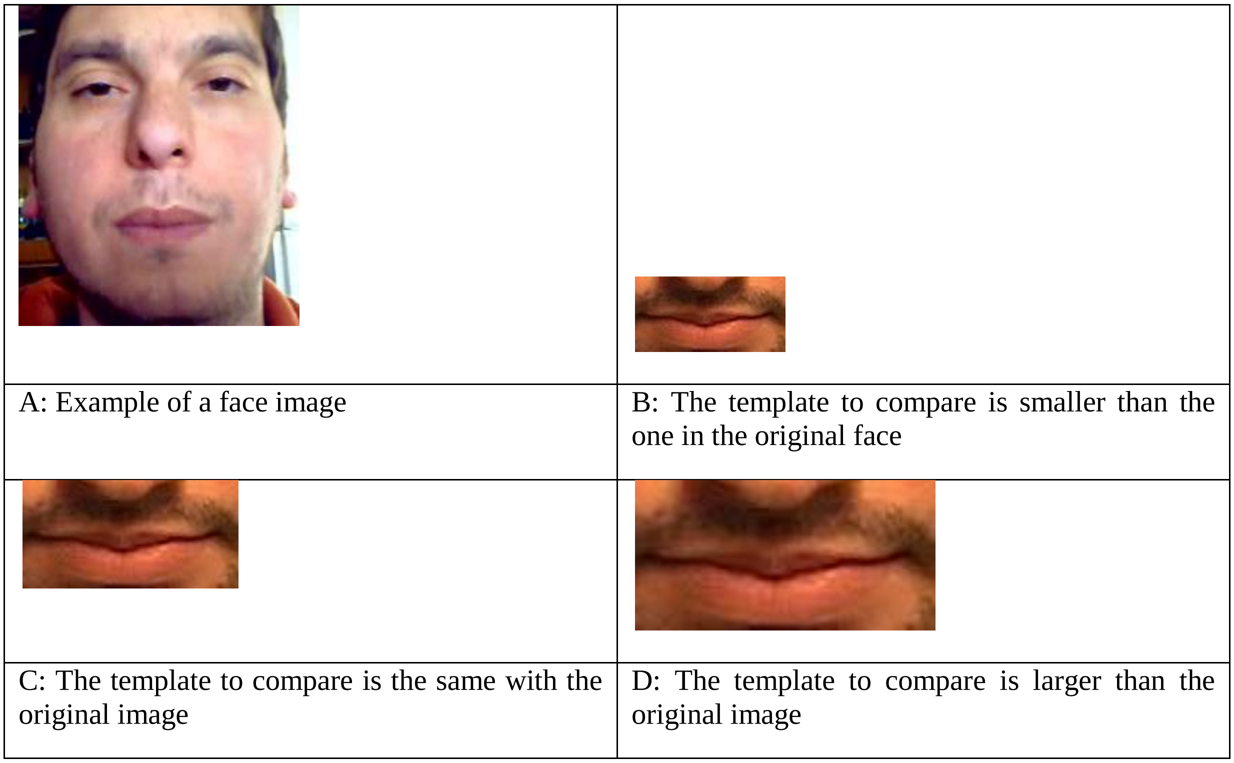
4.1: Experiments

Experiment 4.1

In this experiment we try to cover all cases that may occur when the template dataset does not contain an already segmented mouth of the current speaker, “old mouth”, the cases will be as the following:

1- The template size is the same with the mouth image in the face of the current speaker.
2- The template size is larger than the size of the mouth image in the face of the current speaker.
3- The template size is smaller than the size of the mouth image in the face of the current speaker.

In Table 4.1 we could see that we used all the words in the dataset as input to test our algorithm, the 'Larger than' row show the result using our algorithm. This row shows the results using a mouth templates which is not the same with the mouths in original images. The name of the row corresponds to that the template mouth is larger than the dataset mouths, just like in case 2 above.



A: Example of a face image

B: The template to compare is smaller than the one in the original face

C: The template to compare is the same with the original image

D: The template to compare is larger than the original image

Figure 4.1: Example illustrates the face and different templates that may be compared with it to modify the process of localization of the mouth, note that the templates are not belong to the speaker.

The same mentioned in the above paragraph could be applied to the next row, 'smaller', except that the template mouth is smaller than the original images in the dataset, which is case 3.

The last row, 'Nearly same', means the results with the template mouth is the same size to the dataset images, and except for that, the row is the same with the previous two rows, which is case 1.

Table 4.1: The results of detected mouth region of Experiment 4.1 that the average mouse not belong to the speaker, note that the overall number of each word is 150, which is divided into 15 image for each speaker.

Size of template image/ with respect to the original image	The word	one	two	three	four	five	six	seven	eight	nine	ten	Total percentage
Larger than	(true/false) Number of lips detected	120	120	150	130	150	130	150	120	130	140	89.33%
smaller		100	120	140	130	140	120	140	140	140	140	87.33%
Nearly same		130	130	140	130	140	130	140	130	150	150	91.33%

The results shown in the Table 4.1 is expected, the case 1, which is the last row in the table, has the higher percentage, because when the template mouth is the same with the tested image mouth, then the correlation will give the precise location of the mouth to be segmented.

Figure 4.1 shows an example of a speaker and a three cases templates that used in this experiment.

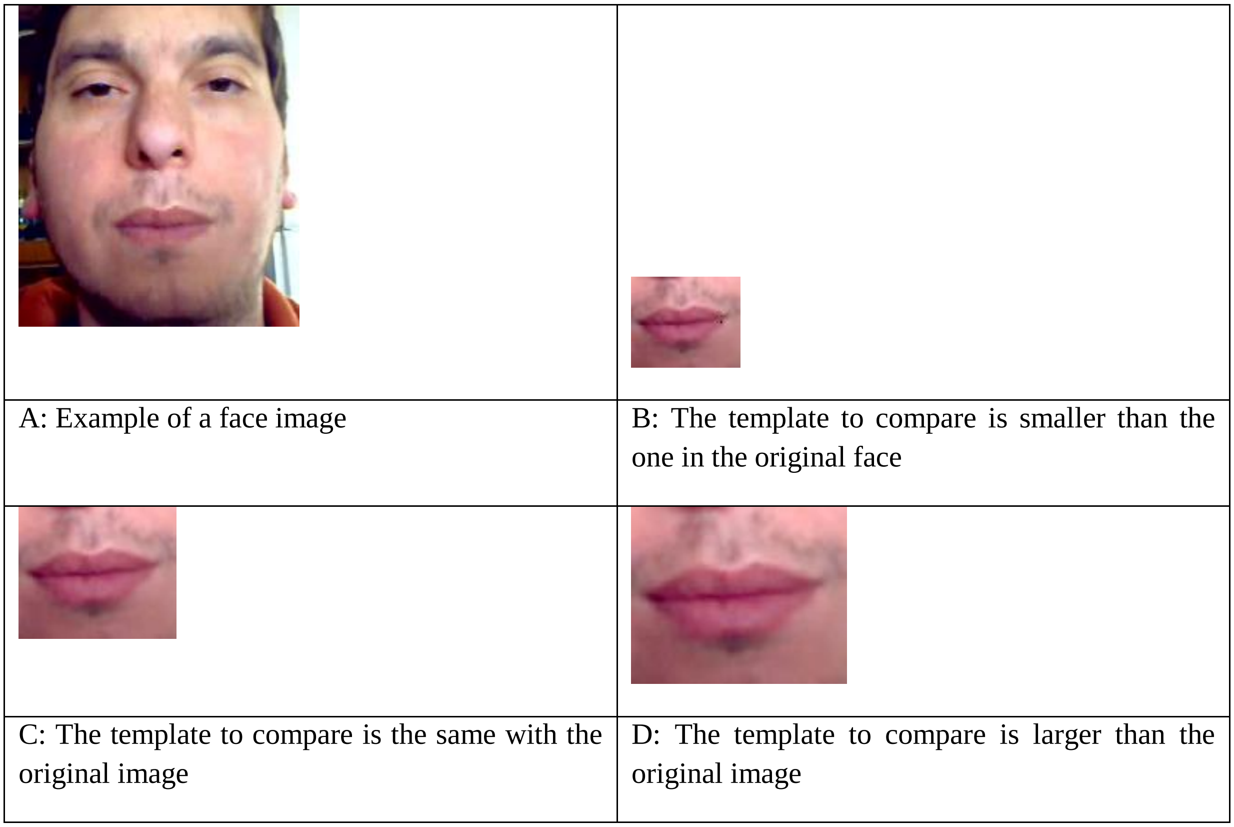
Another suggestion is made. That instead of using the face image we could use the original image and bypass the stage of finding the face to find the mouth in it. This method raises the capability of making the rough estimation of the mouth on any image. See Figure 4.3. this will be the subject of Experiment 4.3

Experiment 4.2

Again we try to cover all of the previous three cases, which stated in Experiment 4.1. The only difference between this experiment and the previous one is that in this experiment we consider the mouth template belongs to the current speaker, “old mouth”. Also, this may be the face of the speaker that uses the system segment his mouth in previous time. This may be an option in the application to save his “Early segmented mouth” to the template dataset.

Table 4.2 is the same structure with Table 4.1, with the difference of the results. In this table we could note that the result is better than that in Table 4.1, this is expected because when the template mouth is the same with the one in the dataset to be segmented, then the correlation will give a precise result.

Finally, Figure 4.2 shows the template mouths that belong to the speaker, that is the subject of Experiment 2 the three cases templates are shown, and one of the dataset face images is shown also.



A: Example of a face image

B: The template to compare is smaller than the one in the original face

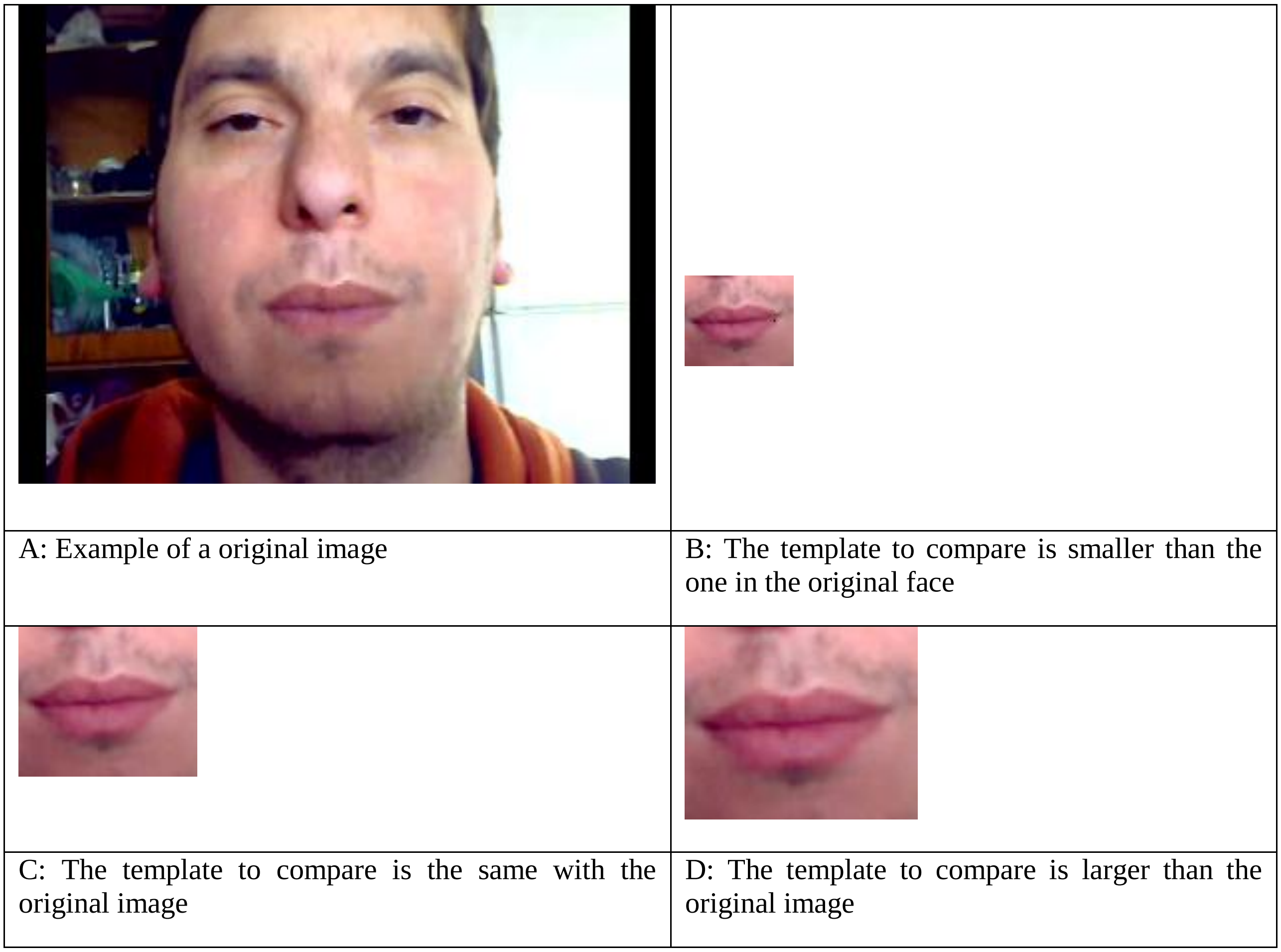
C: The template to compare is the same with the original image

D: The template to compare is larger than the original image

Figure 4.2: Example illustrates the face and different templates that may be compared with it to modify the process of localization of the mouth, note that the templates are belong to the speaker.

Table 4.2: The results of Experiment 4.2 that the average mouse is belong to the speaker, note that the overall number of each word is 150, which is divided into 15 images for each speaker.

Size of template image	The word	one	two	three	four	five	six	seven	eight	nine	ten	Total percentage
Larger than	(true/false)Number of lips detected	130	150	150	140	150	140	150	140	140	140	95.33%
smaller		150	150	150	140	150	140	150	140	140	140	96.67%
Nearly same		150	150	150	150	150	150	150	150	150	140	99.33%



A: Example of a original image

B: The template to compare is smaller than the one in the original face

C: The template to compare is the same with the original image

D: The template to compare is larger than the original image

Figure 4.3: using the original image instead of the face image and bypass the face detection stage

Experiment 4.3

In Figure 4.3 we used the original image with a complex background, before applying the skin-model, in this experiment we used the template of the same speaker in the original image that contains a complex background, and this experiment achieved a good result which is shown in table 4.3.

Table 4.3: The results of Experiment 4.3 that the average mouse is belong to the speaker, using an image with a complex background, note that the overall number of each word is 150, which is divided into 15 images for each speaker.

Size of template image	The word	one	two	three	four	five	six	seven	eight	nine	ten	Total percentage
Larger than	(true/false)Number of lips detected	140	140	140	150	150	150	140	140	140	140	95.33%
smaller		150	150	150	150	150	140	140	150	140	130	96.67%
Nearly same		150	140	150	150	150	150	150	150	150	140	98.66%

Time complexity:

For the Experiments 4.1 and 4.2

One image segmenting using this method take on average 212 m second

One image segmenting using the original method, skin model and conditioned border tracking take on average 234 m second

For Experiment 4.3

One image segmenting using this method take on average 164 m second

Note that in Experiment 4.3 we do not need the skin model but a direct estimation of the mouth is done then we use border tracking method to refine the results.

Notes:

1- Time complexity computed by measuring the time for every speaker alone, then we take the average value.

2- For specific speaker we measure the time for all of his images, then we take the average value, the true and false results are included in the computation.

Second part: Visual Speech Recognition Algorithm

This part of our thesis is composed of two stages: stage one is comparing the recognition methods, training, and preparing dataset. Stage two is to setup the system, and testing it. In the stage one many experiments are done, each experiment acts as a stage, or part of the algorithm. The division of the algorithm into parts clarifies the development of the algorithm and show the importance of each part as it has been added to the algorithm. Parts of the system are abbreviated in the following pseudo code:

Input: a dataset, as described above, each word in the dataset composed of a series of images

Stage One Output: preparing the ten records for each speaker in the dataset, each recognition method will mark each word with one of the ten words from “one” to “ten” in Arabic

Stage Two Output: test the five records for each speaker against the ten records that prepared in stage one

Procedure:

```
Begin
 1-  Enhancement and filtering using CLAHE
 2-  Registration by correlation.
 3-  Motion segmentation by ADI technique.
 4-  Features extracting using the HU-invariant set of features.
 5-  Classification and comparing of the words.
End
```

Note that the five steps are done completely in the two stages.

Recognition methods that used for recognizing the words in the dataset are: average, K-NN, Fuzzy K-NN. In the second stage, we are going to do two things, first to setup the system, and clarify how to use it. The second is to test the recognition method that succeeded in stage one, this will show the practical results when using the system by applying the recognition method which is succeeded in stage one.

4.2: Stage One

4.2.1: Training and Preparing the Dataset

There are many reasons to do the training experiments, one of these reasons is to verify which method is better to be selected. Another reason is to explain some problems that face the system, which is responsible for keeping the final results low.

Before starting this stage, it is important to show the experimental setup:

An important notes: the steps 3, 4,5 of the algorithm are contained in all the first three experiments, the steps 3,4,5 are the subject of Experiment 4.4, while the two other parts are added in the later experiments, in Experiment 4.5 step 2 is added, but not step 1. The final Experiment 4.6 contains all the steps of the algorithm.

In all the recognition methods, the recognition method pick out one word among the 1500 words, it remove the word from the dataset, then compares this word with the remainder of the dataset to test if the algorithm recognizes it correctly or not, in the average method the algorithm compares with the average of each number. If the method recognized the word correctly, then it increases an accumulative 'good' counter. If the result is false, then it increases the 'error'. The true result percentage, which is the result of dividing the 'good' counter over the sum of ‘good’ + ‘error’, is the number of times that the algorithm made a correct recognition. While the error percentage, which is the result of dividing the 'error' counter over the sum of ‘good’ + ‘error’ is the number of times that the algorithm made a wrong recognition. Note that the result shown in the result tables is the true percentage.

4.2.2: Preparing the Dataset

we have to note that a training step is composed of preparing the already saved dataset, which initially is only a series of images not more, so in the preparing step this series of images are converted to a five-dimension input vectors to the three methods to test it. Note that also the average of each repeated word is computed by taking the average ADI of the word ADIs then extracting its features, for example, if the word 'one' is repeated 15 times for each speaker, then compute the average of the 15 'one's ADIs.

4.2.3: Experiments

Experiment 4.4

As a basic step, an experiment on the accumulative difference image (ADI) is done to get the result without any cooperative methods. As stated early, the plain ADI is a poor method to segment the motion from a series of images. This is to show the enhancements in the results after completing each stage of the algorithm. This gradating in the experiments gives us a clear idea on the exact enhancement, which every stage in the algorithm adds.

The series of the Arabic ‘one’ is shown in Figure 4.4 (a), we will talk about the marks on this figure later.

Now we are ready to test the second step in our algorithm, which is segmenting the motion for all the words in our dataset. We start by this step to discover the limitation of using this method alone. The algorithm, as shown in the previous chapter, is divided into stages, first stage is summing the differences in each series of images to get the ADI image. In this experiment, we will not register, nor correlate, nor filter, the images.

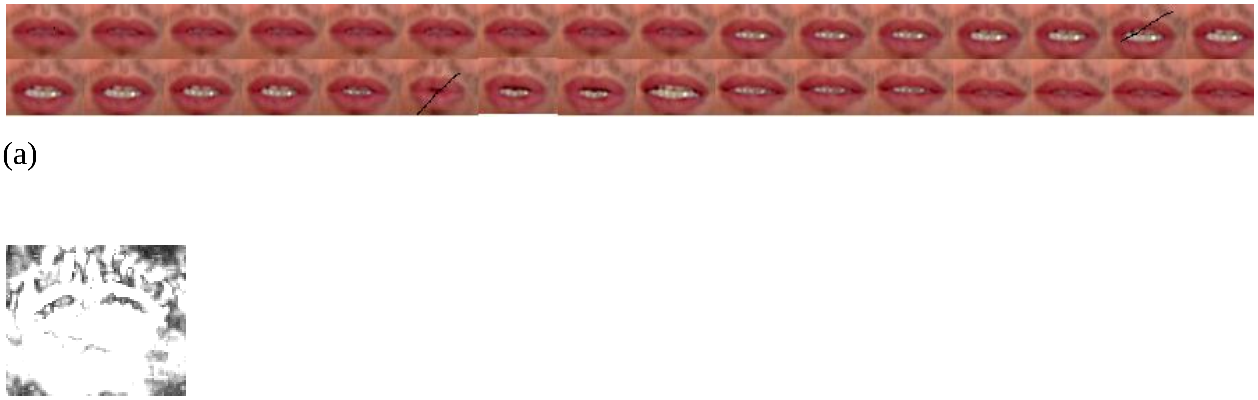
This stage will be done first on the whole dataset to get the ADI image for all the numbers in the dataset. After that the algorithm will compute the HU-invariant set of features for all the ADI images in the dataset and save it in a text file. Note that also in this experiment, as we said above, we will not filter the ADI image, this will be done in Experiment 4.6 to see the difference in the results due to enhancement.

Finally, the process now is looking like just comparing vectors of five dimensions.

In this experiment, the K-NN, the fuzzy K-NN, and average recognition methods are used. 1000 five-dimension vectors are used as an input to these recognition methods. Each recognition method outputs some details about which words are recognized correctly, and which are not, these details will be shown in a result table in every experiment. Note that the other 500 records are left for the testing stage.

A brief description to each recognition method is in the recognition methods section.

Figure 4.4 (b) shows the resulted ADI image of summing the differences of the series in Figure 4.4 (a). The reader now will notice simply that the image is almost saturated in many parts. This is happened because the images are almost not registered. That is coming from the fact that we cannot compel the speaker to stay fixed front the camera and do not make any little movements, which makes differences in a few pixels.



(a)

(b)

Figure 4.4: (a) The series of the Arabic word ‘one’, (b) The ADI of (a)

Recognition methods:

To compare and recognize spoken Arabic numbers, we use three methods based on Euclidian distance, these methods will be explained shortly.

Depending on the Euclidian distance we use three methods for recognition: the averaging method, K-nearest neighborhoods, and fuzzy K-nearest neighborhoods. Experimentally the averaging method gives the best results. The reason of this is averaging. The averaging method depends on the average of the class, not on a specific member(s) like in KNN or in FK-NN. This will be clarified in more details at the end of stage one.

The Average method: it simply computes the average ADI, and then it computes the distance between the input instance and each average ADI. After that it takes the minimum distance for recognition. Finally the instance is classified as the word associated with the minimum distance.

The K-Nearest Neighborhood method: Find the k closest vectors and collect their classifications. Then the input vector will be classified as the most dominant class in the k vectors.

The K-Fuzzy Nearest Neighborhood method: In many data sets, there are a number of attributes that are completely unrelated to the classification process. When this happens, these attributes must have a lower priority. In this method the priority is expressed as a weight. Then the distance will be calculated by using the following equation:

$$d\ (P, Q) = \sqrt{\left(\sum_i w_i (p_i - q_i)^2\right)}$$

Where W is a vector of weights.

This is not related to our case. In our case, the weight is relevant to the rank of the k-nearest neighborhoods. Suppose k= 3, then the first vector have the highest weight. Number three has the lowest weight, while number two has the middle weight. Now if the three nearest are all different, the input unknown vector will has the class of the first vector.

At the end when comparing the content of the dataset, the results are like in Tables 4.4, 4.5, 4.6:

Table 4.4: Results of Experiment 4.4 for a male speaker

Method	The word	one	two	three	four	five	six	seven	eight	nine	t en	Total percentage
The average	(true/false)Number of words classified	7/10	4/10	2/10	5/10	5/10	2/10	5/10	4/10	3/10	0/10	37%
K-NN		6/10	2/10	1/10	7/10	4/10	1/10	2/10	3/10	0/10	1/10	27%
Fuzzy K-NN		4/10	4/10	2/10	4/10	4/10	2/10	4/10	4/10	0/10	0/10	28%

Table 4.5: Results of Experiment 4.4 for a female speaker

Method	The word	one	two	three	four	five	six	seven	eight	nine	ten	Total percentage
The average	(true/false)Number of words classified	8/10	4/10	4/10	4/10	5/10	3/10	2/10	2/10	1/10	2/10	35%
K-NN		6/10	2/10	1/10	8/10	4/10	2/10	2/10	4/10	0/10	1/10	30%
Fuzzy K-NN		4/10	4/10	2/10	4/10	4/10	2/10	4/10	5/10	0/10	0/10	29%

Table 4.6: Results of Experiment 4.4 for all the dataset, "for the ten speakers, 10 records for each speaker"

Method	The word	one	two	three	four	five	six	seven	eight	nine	ten	Total percentag
The average	(true/false) Number of words classified	80/100	37/100	40/100	40/100	50/100	30/100	20/100	20/100	9/100	20/100	31%
K-NN		60/100	20/100	9/100	80/100	40/100	20/100	18/100	38/100	0/100	10/100	25%
Fuzzy K-NN		40/100	40/100	20/100	40/100	40/100	20/100	40/100	42/100	0/100	0/100	30%

Tables 4.4, 4.5, and 4.6 show the results of Experiment 4.4. Each table consists of three rows, the first row is for the average recognition method, the second, for the K-NN recognition method, while the last row is used for the Fuzzy K-NN recognition method. Every row shows the result, of one of the recognition methods, for each word separately, for example, Table 4.4, the first column of the result in the table corresponds for the word "one", because there are ten instances of the word "one" saved in the dataset for the male speaker, as stated in Section 4.2.1 then the result is shown as a percentage of the 'good' counter divided by ten, 7/10 in Table 4.4.

The last column in the table shows the overall percentage, the percentage resulted in dividing the counter 'good' by 100, 100 is the number of words contained in the speaker section of the dataset, which consists of ten speakers as shown previously .

The above paragraph shows the structure of Table 4.4, which is the same with the structure of the Tables 4.4, 4.5, 4.6.

From Table 4.4 we cans see that the best result is achieved by the average method. This is an expected result, because it sums all the properties of the already existing words in the dataset. While the two other algorithms, in our case, may accumulate the error to give a mistaken results.

Table 4.5 shows the same results but for a female speaker, while Table 4.6 shows the overall results of the ten speakers in the dataset.

Experiment 4.5: Adding the Correlation Step

In this experiment, we do a correlation between every two images in the series to correct the coordinate system of the second image with respect to the previous image, this means that transforming the two images into one coordinate system, this process is shown in Chapter 3, and it is called the registration between two images. This makes the method powerfully independent to the little movements that made by the speaker while pronouncing the word. This method simply registers all the images with the first image before subtracting. Note that the other steps in this experiment are the same as in Experiment 4.4. Table 4.7 shows the results of Experiment 4.5.

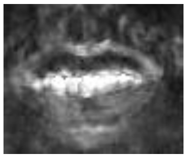


Figure 4.5: ADI extracted after using correlation for the image series in Figure 4.4(a).

As you can see in Figure 4.5 the ADI image extracted after the correlation process is more visible than that one without correlation. The moving lips look very visible. It's clearly noted that the area surrounding the lips is moving. These complex movements in lips and in the surrounding area are hard to be discovered without the correlation process.

The enhancement of the results is clearly shown in Table 4.9.

As shown in Table 4.9, the result of the average recognition method achieved 44.6% success rate, while the K-NN and the Fuzzy K-NN achieved 36.8%, 44.8% success rate respectively. From these results, the enhancement is 13% between the average method in the two Experiments 4.5, and 4.4.

Table 4.7: The results after using the correlation to register the series. Results are for a male speaker.

Method	The word	one	two	three	four	five	six	seven	eight	nine	ten	Total percentage
The average	(true/false)Number of words classified	8/10	7/10	4/10	3/10	5/10	2/10	7/10	7/10	1/10	0/10	44%
K-NN		8/10	6/10	2/10	5/10	5/10	0/10	6/10	6/10	1/10	1/10	40%
Fuzzy K-NN		8/10	7/10	2/10	4/10	6/10	2/10	4/10	6/10	1/10	1/10	41%

Table 4.8: The results after using the correlation to register the series. Results are for a female speaker.

Method	The word	one	two	three	four	five	six	seven	eight	nine	ten	Total percen tage
The average	(true/false) Number of words classified	9/10	7/10	6/10	3/10	5/10	2/10	5/10	7/10	1/10	0/10	45%
K-NN		9/10	4/10	4/10	5/10	5/10	0/10	6/10	6/10	1/10	1/10	41%
Fuzzy K-NN		9/10	7/10	2/10	4/10	6/10	2/10	4/10	6/10	1/10	1/10	42%

The process registering by correlating the images makes the changes in the images series is clearer to detect. The next section will focus on the filtering as a technique to manipulate some problems facing the enhancements of the results. The analysis section will focus on more details about the results.

Table 4.9: The results after using the correlation to register the series, for the ten speakers in the dataset

Method	The word	one	two	three	four	five	six	seven	eight	nine	ten	Total percentage
The average	(true/false)Number of words classified	90/100	55/100	45/100	40/100	55/100	25/100	65/100	65/100	3/100	3/100	44.6%
K-NN		78/100	61/100	18/100	49/100	45/100	5/100	30/100	64/100	9/100	9/100	36.8%
Fuzzy K-NN		82/100	50/100	25/100	44/100	61/100	15/100	42/100	66/100	15/100	8/100	40.8%

Experiment 4.6: Enhancement in the ADI Image

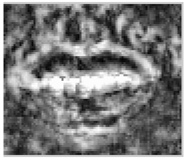


Figure 4.6: The ADI image after correlation and filtering the series of images.

In this experiment we try to avoid the variation of light, this enhancement is applied on the series of images before starting to register the images. Contrast-Limited Adaptive Histogram normalization or Equalization filter (CLAHE). Before speaking about CLAHE method we have to define the histogram equalization which enhances the contrast of images by transforming the values in an intensity image so that the histogram of the output image approximately matches a specified histogram (uniform distribution by default).

Now CLAHE performs contrast-limited adaptive histogram equalization. Unlike histogram equalization, it operates on small data regions (tiles) rather than the entire image. Each tile's contrast is enhanced so that the histogram of each output region approximately matches the specified histogram (uniform distribution by default). The contrast enhancement can be limited in order to avoid amplifying the noise which might be present in the image. This overall process is hoped to enhance the results under lighting conditions, it's also remove shadows despite we do not need that, because there are no shadows in the input images.

Now the only difference between the image in Figure 4.5 and the image in Figure 4.6 is that the filtering process is done and the effect is clearly appeared in the image in Figure 4.6 We notice that in Figure 4.6 the lightening is more distributed in a wider area.

Table 4.12 shows that the filtering technique could give a little but important enhancements in the results. Note that the average recognition method in this experiment achieved 44.9% success rate with respect to 44.6% in the Experiment 4.5, that’s mean a 0.3% enhancement in the result. Of course this enhancement depends on the light conditions that surround the speaker.

Table 4.10: The results after filtering the series of images. results of Experiment 4.6 for a male speaker.

Method	The word	one	two	three	four	five	six	seven	eight	nine	ten	Total percentage
The average	(true/false)Number of words classified	9/10	4/10	9/10	5/10	5/10	4/10	1/10	5/10	3/10	5/10	50%
K-NN		9/10	3/10	5/10	6/10	5/10	0/10	6/10	6/10	1/10	1/10	42%
Fuzzy K-NN		8/10	5/10	4/10	4/10	5/10	2/10	4/10	5/10	1/10	1/10	39%

Table 4.11: The results after filtering the series of images. Results of Experiment 4.6 for a female speaker.

Method	The word	one	two	three	four	five	six	seven	eight	nine	ten	Total percentage
The average	(true/false)Number of words classified	7/10	4/10	9/10	5/10	5/10	1/10	1/10	4/10	3/10	1/10	40%
K-NN		7/10	4/10	5/10	6/10	5/10	1/10	5/10	7/10	1/10	1/10	42%
Fuzzy K-NN		8/10	4/10	4/10	4/10	5/10	2/10	4/10	5/10	3/10	1/10	40%

Table 4.12: The results after filtering the series of images. Results of Experiment 4.6 for the ten speakers in the dataset.

Method	The average	K-NN	Fuzzy K-NN
The word	(true/false)Number of words classified		
one	90/100	78/100	74/100
two	40/100	63/100	42/100
three	46/100	38/100	29/100
four	41/100	49/100	46/100
five	54/100	40/100	62/100
six	35/100	15/100	13/100
seven	60/100	10/100	41/100
eight	67/100	70/100	64/100
nine	8/100	9/100	15/100
ten	8/100	9/100	8/100
Total percentage	44.9%	38.1%	39.4%

4.2.4: Assembling the Parts

The last question remains is: how could we determine the start frame? To answer this question, we have to define the most distinct frame. The definition depends on the observations: first, every word, digit in our case, starts with a closed mouth frame, the silence stage, the end of the word also is a closed mouth frame. The second observation is that every time we have to talk we must open the mouth, so the most opened mouth frame is the most distinct frame from the start frame, and the start frame is the first frame when the mouth starts opening, the most distinct frame is marked in Figure 4.7.

Simply in our algorithm, we measure the changes of the successive images, when the change exceeds some threshold, then we consider the threshold's frame is the most distinct frame. After determining this frame, then we go back for twenty frames to define the start frame, and go forward another twenty frames to define the end frame, the result which is the input for all the previous experiments is shown in Figure 4.7.

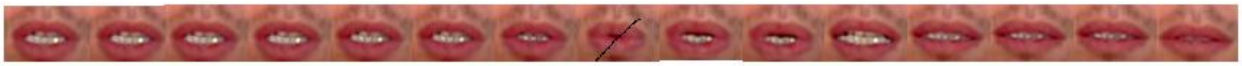
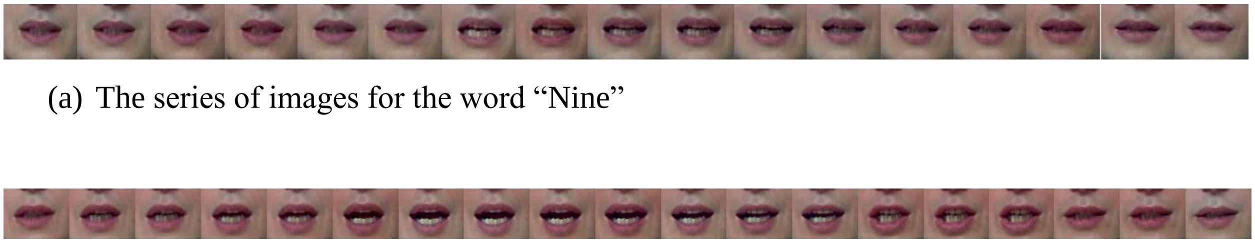


Figure 4.7: the number “one” in the Arabic language as an input example of the first three Experiments, 4.4, 4.5, and 4.6, the most distinct frame are marked in this figure.

4.2.5: Analysis

In this section we are going to clarify one of the reasons that give bad results. One of the most important reasons is shown in Figure 4.8, this figure shows two series of images, Figure 4.8(a) is for the number nine in Arabic, while Figure 4.8(b) is for number ten. This figure shows that the two numbers are almost close. There are some differences between them. One of the differences shown is the rank of the images, that is, if we conserve the arrangement of the images and giving each one a rank number, the images in the two series are nearly similar but do not have the same rank number. For example, the image number seven in the ‘nine’ series is the same with the image number four in the “ten” series. Again, the image number eleven in the ‘nine’ series is almost the same with the one with rank 13 in image series “ten”.



(a) The series of images for the word “Nine”

(b) The series of images for the word “Ten”

Figure 4.8: Two series for two different words show the reasons distort the results.

4.3: Stage Two

Setup the system, and testing it

As a clarification example on the system, in Figure 4.9, a Skelton of a dataset contains two speakers, the first have a three records, while the other have four records.

Note that Figure 4.9 is for clarification only because our dataset consists of ten speakers, each has a fifteen records, ten are used in stage one, and the other five are used in stage two.

As we saw in the test step the best results were for the average method.

This means that we have to compute the average of all instances of the ADI images that belong to the same word. Depending on that average, every time the speaker wants to use the system, he has to say his word, the program has to measure the distance to all the average images, compare these distances, give the speaker word the title with the smallest distance.

In this step, we do a test experiment.

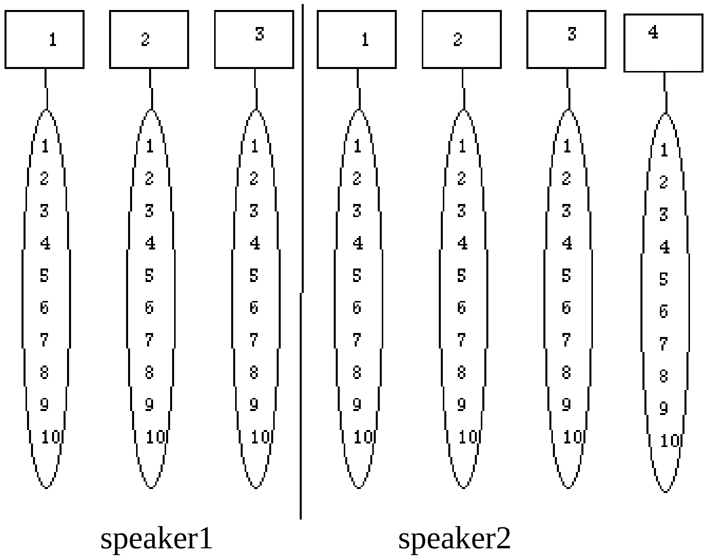


Figure 4.9: A dataset consists of two speakers, speaker1 has three records, speaker2 has a four records, the record in an ellipse consists of the words 'one' to 'ten' in the Arabic language.

Experiment 4.7:

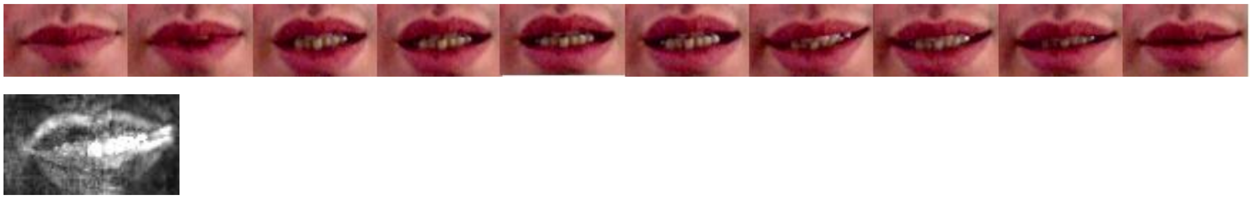


Figure 4.10: The word “six” as an example of the input of the Experiment 4.6.

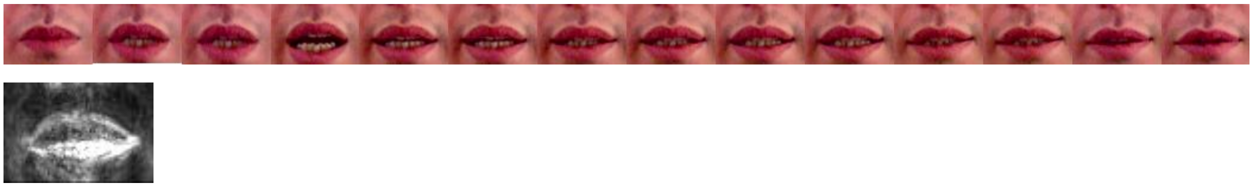


Figure 4.11: The word “two” as an input example of Experiment 4.6.

Five records for every speaker are used in this experiment. The speakers in this experiment pronounce the ten words, from “one” to “ten” in Arabic language, as a new input. At every time the speaker says a word, the program should recognize it immediately. After completing the ten words, we have to check each word result, if it was true, false, and then, write down the overall final result in a table, such as the one in Table 4.13. The three rows in Table 6.4 correspond to the three methods: the average, the K-NN, and the Fuzzy K-NN recognition methods. The result in any interior cell in the table shows a result from the recognition method which titles that row, for the word which titles that column. The last column in the table shows the overall percentage of success rate. We used our algorithm to extract the ten AD, then it extracts the HU-invariant set of features from the resulted image. These features have to be compared with the speaker record. We want to use the averages in the comparisons, so after preparing the ten words, we will use the average method to recognize them based on the averages that already exist in the dataset.

Table 4.13: The result of testing Experiment 4.7, for a male speaker from the dataset.

Method	The word	one	two	three	four	five	six	seven	eight	nine	ten	Total percentage
The average		4/5	5/5	3/5	2/5	4/5	1/5	2/5	2/5	2/5	2/5	54%
K-NN		4/5	4/5	3/5	2/5	0/5	3/5	2/5	2/5	2/5	1/5	46%
Fuzzy K-NN		3/5	4/5	4/5	3/5	3/5	2/5	3/5	2/5	1/5	1/5	52%

Table 4.14: The result of testing Experiment 4.7, for a female speaker from the dataset.

Method	The word	one	two	three	four	five	six	seven	eight	nine	ten	Total percentage
The average		4/5	5/5	3/5	4/5	4/5	1/5	3/5	2/5	2/5	2/5	60%
K-NN		3/5	3/5	3/5	3/5	4/5	0/5	3/5	2/5	2/5	0/5	46%
Fuzzy K-NN		4/5	4/5	4/5	3/5	4/5	2/5	3/5	2/5	0/5	1/5	54%

Table 4.15: The result of testing Experiment 4.7, for the ten speakers in the dataset.

Method	The word	one	two	three	four	five	six	seven	eight	nine	ten	Total percentage
The average		45/50	42/50	39/50	41/50	44/50	13/50	29/50	18/50	5/50	3/50	55.8%
K-NN		43/50	43/50	36/50	42/50	49/50	9/50	30/50	5/50	5/50	1/50	52.6%
Fuzzy K-NN		30/50	40/50	40/50	30/50	30/50	20/50	30/50	20/50	10/50	10/50	52%

In Table 4.13, we show the average recognition result for the ten speakers using the prepared dataset. In this result we see reasonable results that comply with the results in stage one. In the average method, the bad results started to show up in the last three numbers, the reason for this is stated in the analysis section, this achieved an overall 55.8% recognition rate. While in the Fuzzy K-NN, the word "four" the detection rate was low. In spite that this word, the word "four", achieved good results in the stage one, this comes from the bad capturing conditions for this word. Same comments for the K-NN method. This is a glance that the average recognition method is more robust for the different circumstances. As an example of the problems that face this experiment is the word "six", which is shown in Figure 4.10, we could see clearly that there is a problem in capturing the video. This problem comes from that the speaker shifts his face during the capturing process. The speaker could notice that this problem makes all the methods to fail in detecting the word. This problem is not found in the Figure 4.11 which for the word "two", in this figure we notice that the ADI is very visible, this is a direct result of clear series of images. The reader could see that in this series the speaker commitment to pronounce the word while he was fixed.

Time complexity:

After executing all part of the algorithm, to convert one word from a series of images to a feature vector, the process take 1.5351366 second

Notes:

1- Time complexity computed by measuring the time for every speaker alone, then we take the average value.

2- For specific speaker we measure the time for each of his records, from “one” to “ten”, then we take the average value.

Chapter five

Conclusion and Future Works

In this chapter we will introduce the conclusion from this thesis to summarize the work in section 5.1, later in Section 5.2 future works are proposed as ideas to develop what we did.

5.1: Conclusion remarks

In recent days speech recognition systems getting more important. It is used in a lot of area. For example, some newer cellular phones include speech recognition. Another example is Windows Speech Recognition in Windows Vista empowers users to interact with their computers by voice. It was designed for people who want to significantly limit their use of the mouse and keyboard due to various conditions.

In our thesis we focus on the visual speech recognition section, Visual speech in itself does not contain sufficient information for speech recognition, but by combining visual and audio speech systems are able to achieve better performance than what is possible with audio-only ASR.

Our work depends on the fact that Visual speech information mainly contained in the motion of visible articulators such as lips, and tongue, so we start our work by focusing on the lip area segmentation and tracking techniques. In this field we proposed a new algorithm based on template matching method to segment the lip area, we got adequate results for this algorithm.

In speech recognition we invent a new algorithm for the visual part, in this part we did much work in motion detection based on a series of images taken for the speaker, including ADI supported by registration by correlation, feature extraction, and recognition methods. We do some experiments to compare between the methods of recognition.

5.2: Future Works

Find a way to consider the arrangement of the frames of the sequence:

A problem face our thesis is to rank the frames, so some researches needed to cover this point. An idea to solve this is by using a scaling technique, which means, despite there is no difference in the summed changes, we multiply the change between every two successive frames by a certain factor, this factor depends on the rank of frames. This will force the sum of changes to depend on the rank of the frames.

More experiments on using the whole face. Instead of using only the mouth region:

Some papers state that the face expression may give a more clear idea about the speech, like the jaw motion, we thing this point deserves a research.

We need a method to vote depending on the three results of the three recognition methods to make a final decision:

There are several methods to sum the results of the three recognition methods, we need more research on this point to benefit from the three methods together. This is a point we are looking forward to do some work.

References

[1] D.G. Stork and M.E. Hennecke, Speech reading by Humans and Machines, Berlin, Germany: Springer, 1996.

[2] P. Teissier, J. Robert-Ribes, Schwartz, J.-L., and A. Guerin-Dugue, " Comparing models for audiovisual fusion in a noisy-vowel recognition task," IEEE Transactions on Speech and Audio Processing, 7(6):629-642, 1999.

[3] S. Dupont and J. Luettin, "Audio-visual speech modeling for continuous speech recognition," IEEE Transactions on Multimedia, 2(3):141-151, 2000.

[4] D.W. Massaro and D.G. Stork, "Speech recognition and sensory integration," American Scientist, 86(3):236-244. 1998.

[5] H. McGurk and J.W. MacDonald, "Hearing lips and seeing voices," Nature, 264:746-748, 1976 .

[6] A.Q. Summerfield, "Some preliminaries to a comprehensive account of audio-visual speech perception," In Dodd, B. and Campbell, R. (Eds.), Hearing by Eye: The Psychology of Lip-Reading. Hillside, NJ: Lawrence Erlbaum Associates, pp. 97-113. 1987 .

[7] W. H. Sumby and I. Pollack, "Visual contributions to speech intelligibility in noise," Journal of the Acoustical Society of America, 26:212–215, 1954.

[8] R. C. Gonzalez and R. E. Woods, Digital Image Processing, 3rd Edition. Upper SaddleRiver, NJ, USA: Prentice-Hall, Inc., 2006.

[9] F. G. Smith, K. R. Jepsen, and P. F. Lichtenwalner, "Comparison of neural network and Markov random field image segmentation techniques," in Proceedings of the 18th Annual Review of progress in quantitative nondestructive evaluation, vol. 11, 1992, pp. 717-724.

[10] A. Blake and M. Isard, Active Contours, Springer, 1998.

[11] J. Shi and J. Malik, "Normalized cuts and image segmentation," in CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97). Washington, DC, USA: IEEE Computer Society, 1997, p. 731.

[12] J. Sethian, "Level set methods and fast marching methods: Evolving interfaces in computational geometry," 1998.

[13] D. Cremers, M. Rousson, and R. Deriche, "A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape, " Int. J. Comput. Vision,vol. 72, no. 2, pp. 195-215, 2007.

[14] K. Neely. "Effect of visual factors on the intelligibility of speech, " Journal of the Acoustical Society of America, 28(6):1275–1277, 1956.

[15] C. Binnie, A. Montgomery, and P. Jackson, "Auditory and visual contributions to the perception of consonants, " Journal of Speech Hearing and Research, 17:619–630, 1974.

[16] D. Reisberg, J. McLean, and A. Goldfield, "Easy to hear, but hard to understand:A lipreading advantage with intact auditory stimuli, " In B. Dodd and R. Campbell, "Hearing by Eye, " pages 97–113. Lawrence Erlbaum Associates, 1987.

[17] K. P. Green and P. K. Kuhl, "The role of visual information in the processing of place and manner features in speech perception, " 45(1):32–42, 1989.

[18] D. W. Massaro, "Integrating multiple sources of information in listening andreading, " In Language perception and production. Academic Press, New York.

[19] R. Campbell and B. Dodd, "Hearing by eye, " Quarterly Journal of Experimental Psychology, 32:85–99, 1980.

[20] B. Dodd, "Lipreading in infants: Attention to speech presented in and out of synchrony, " Cognitive Psychology, 11:478–484, 1979.

[21] P. K. Kuhl and A. N. Meltzoff, "The bimodal perception of speech in infancy, " 218:1138–1141, 1982.

[22] B. Dodd and R. Campbell, "Hearing by Eye: The Psychology of Lipreading, " Lawrence Erlbaum, London, 1987.

[23] B. Dodd, R. Campbell and D. Burnham, "Hearing by eye II : advances in the psychology of speechreading and auditory-visual speech, " Psychology Press, Hove, East Sussex, UK, 1998.

[24] G. W. Greenwood, "Training partially recurrent neural networks using evolutionary strategies, " *IEEE Trans. Speech and Audio Processing*, 5(2):192–194, 1997.

[25] E. Owens and B. Blazek, "Visemes observed by hearing impaired and normal hearing adult viewers, " *Journal of Speech Hearing and Research*, 28:381–393, 1985.

[26] A. M. Tekalp and J. Ostermann, "Face and 2-D mesh animation in MPEG-4, " *Signal Processing: Image Communication, Special Issue on MPEG-4*, 15:387–421, Jan. 2000.

[27] L. R. Rabiner and B. H. Juang, *Fundamentals of speech recognition.,* Prentice Hall International Inc., 1993. Signal Processing Series.

[28] S. Morishima, S. Ogata, K. Murai and S. Nakamura, "Audio-visual speech translation with automatic lip synchronization and face tracking based on 3-D head model, " In *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, volume 2, pages 2117–2120, May 2002.

[29] B. P. Yuhas, M. H. Goldstein and T. J. Sejnowski, "Integration of acoustic and visual speech signals using neural network, " *IEEE Communication Magazine*, pages 65–71, 1989.

[30] E. D. Petajan, "*Automatic lipreading to enhance speech recognition*, " PhD thesis, University of Illinois at Urbana-Champaign, 1984.

[31] K. Prasad, D. Stork and G. Wolff, "Preprocessing video images for neural learning of lipreading. Technical Report, " Technical Report CRC-TR-93-26, Ricoh California Research Center, 1993.

[32] S. Lucey, S. Sridharan and V. Chandran, "Initialized eigenlip estimator for fast lip tracking using linear regression, " In *International Conference on Pattern Recognition*, pages 182–185, Barcelona, Sep. 2000.

[33] M. Kass, A. Witkin and D. Terzopoulus, "Snakes: Active contour models" In *International Journal of Computer Vision*, 1(4):321–331, 1988.

[34] M. U. Ramos Sanchez, J. Matas and J. Kittler, "Statistical chromaticity-based lip tracking with b-splines, " In *International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages 2973–2976, Munich, Germany, Apr. 1997.

[35] R. Cipolla and A. Blake, "The dynamic analysis of apparent contours, " In *3rd International Conference on Computer Vision*, pages 616–623, 1990.

[36] Hyewon Pyun, Hyun Joon Shin, Tae Hoon Kim and Sung Yong Shin, "Realtime facial expression capture for performance-driven animation. Technical Report, " Technical Report CS-TR-2001-167, Computer Science Department, Korea Advanced Science and Technology, 2001.

[37] C. Bregler and S. M. Omohundro, "Surface learning with applications to lipreading, " In J. D. Cowan, G. Tesauro and J. Alspector, editors, *Advances in Neural Information Precessing Systems 6*. Morgan Kaufmann Publishers, San Francisco, CA, 1994.

[38] C. Bregler and Y. Konig, "Eigenlips for robust speech recognition, " In *International Conference on Acoustics, Speech, and Signal Processing*, pages 669–672, 1994.

[39] A. Lanitis, C. Taylor and T. Cootes, "Automatic tracking, coding and reconstruction of human faces using flexible appearance models, " *IEE Electronic Letters*, 30:1578–1579, 1994.

[40] A. L. Yuille and P. Hallinan, "Deformable templates, " In Andrew Blake and Alan Yuille, editors, *Active Vision*, pages 21–38. The MIT Press, 1992.

[41] M. E. Hennecke, K. V. Prasad and D. G. Stork, "Using deformable templates to infer visual speech dynamics, " In *The 28th Asilomar Conf. on Signals, Systems and Computers*, pages 578–582, 1994.

[42] A. L. Yuill,e P. Hallinan and D. S. Cohen, "Feature extraction from faces using deformable templates, " International Journal of Computer Vision, 2(8):99–112, 1992.

[43] C. Kervrann and F. Heitz, "A hierarchical markov modeling approach for the segmentation and tracking of deformable shapes, " Graphical Models and Image Processing, 60(3):173–195, May 1998.

[44] MunWai Lee and Surendra Ranganath, "Pose-invariant face recognition using a 3D deformable model, " Pattern Recognition, 36:1835–1846, 2003.

[45] T. F. Cootes, A. Hill, C. J. Taylor and J. Haslan, "Use of active shape models for locating structures in medical images, " Image and Vision Computing, 12(6):355–365, 1994.

[46] J. Luettin and N. A. Thacker, "Speechreading using probabilistic methods, " Computer Vision and Image Understanding, 65(2):163–178, Feb. 1997.

[47] J. A. Nelder and R. Mead, "A simplex method for function optimization, " Computing Journal, 7(4):308–313, 1965.

[48] T. F. Cootes, C. J. Taylor and A. Lanitis, "Multi-resolution search using active shape models, " In The 12th International Conference on Pattern Recognition, pages 610–612, 1994.

[49] T. F. Cootes, C. J. Taylor and G. J. Edwards, "Active appearance models, " IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(6):681–685, Jun. 2001.

[50] K. Mase and A. Pentland, "Lip reading: Automatic visual recognition of spoken words, " In Optical Society of America Topical Meeting on Machine Vision, pages 1565–1570, 1989.

[51] E. D. Petajan, B. J. Bischoff, D. A. Bodoff and N. M. Brooke, "An improved automatic lipreading system to enhance speech recognition. Technical Report, " Technical Report TM 11251-871012-11, Bell Labs, 1987.

[52] B. P. Yuhas, M. H. Goldstein, T. J. Sejnowski and R. E. Jenkins, "Neural network models of sensory integration for improved vowel recognition, " Proceedings of the IEEE, 78(10):1658–1668, 1990.

[53] D. G. Stork, G. Wolff and E. P. Levine, "Neural network lip-reading system for improved speech recognition, " In International Joint Conference on Neural Network, pages 285–295, 1992.

[54] P. Cosi, M. Dugatto, E. Magno Caldognetto, K. Vagges, G. A. Mian and M. Contolini, "Bimodal recognition experiments with recurrent neural networks, " In International Conference on Acoustics, Speech, and Signal Processing, volume 2, pages 553–556, 1994.

[55] J. R. Movellan, "Visual speech recognition with stochastic networks, " In D. Touretzky G. Tesauro and T. Leen, editors, Advances in Neural Information Processing Systems, volume 7, pages 851–858. The MIT Press, Cambridge, MA, 1995.

[56] A. Hagen and A. Morris, "Comparison of HMM experts with MLP experts in the full combination multi-band approach to robust ASR, " In Int. Conf. Spoken Language Processing, volume 1, pages 345–348, China, 2000.

[57] M. Gordan, C. Kotropoulos and I. Pitas, "A support vector machine-based dynamic network for visual speech recognition applications, " EURASIP Journal on Applied Signal Processing, (11):1248–1259, 2002.

[58] G. Potamianos, C. Neti, J. Luettin and I. Matthews, "Audio-visual automatic speech recognition: An overview, " In G. Bailly E. Vartikiotis-Bateson and P. Perrier, editors, Audio-Visual Speech Processing, pages 121–148. The MIT Press, 2003.

[59] K. Saenko, K. Livescu, J. Glass and T. Darrell, "Production domain modeling of pronunciation for visual speech recognition, " In IEEE Int. Conf. Acoustics, Speech, and Signal Processing, volume 3, US, March 2005.

[60] S. Choi, H. Hong, H. Glotin and F. Berthommier, "Multichannel signal separation for cocktail party speech recognition: a dynamic recurrent network, " Neurocomputing, 49(1-4):299–314, 2002.

[61] T. TobeIyt, N. Tsurutat and M. Amamiyat, "On-line speech-reading system for japanese language, " In 9th International Conference on Neural Information Processing, volume 3, pages 1188–1193, 2002.

[62] T. Chen J. S. Kim, "Segmentation of image sequences using sofm networks, " 15th Int. Conf. Pattern Recognition, volume 3, pages 3877–3880, 2000.

[63] A. J. Goldschen, "Continuous Automatic Speech Recognition by Lipreading, " PhD thesis, Dept. of Electrical Engineering and Computer Science, George Washington University, 1993.

[64] A. Adjoudani and C. Benoit, "Audio-visual speech recognition compared across to architectures, " In 4th European Conference on Speech Communication and Technology, volume 2, pages 1563–1566, Madrid, 1995.

[65] P. L. Silsbee and A. C. Bovic, "Visual lipreading by computer to improve automatic speech recognition accuracy, " Technical Report TR-93-02-90, University of Texas Computer and Vision Research Center, Austin, TX, 1993.

[66] P. L. Silsbee and A. C. Bovic, "Medium vocabulary audiovisual speech recognition, " In NATO ASI New Advances and Trends in Speech Recognition and Coding, pages 13–16, 1993.

[67] I. Matthews, T. F. Cootes, J. A. Bangham, S. Cox and R. Harvey, "Extraction of visual features for lipreading, " IEEE Trans. on Pattern Analysis and Machine Intelligence, 24(2):198–213, Feb. 2002.

[68] M. Tomlinson, M. Russell and N. Brooke, "Integrating audio and visual information to provide highly robust speech recognition, " In IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, volume 2, pages 821–824, 1996.

[69] J. Luettin, N. A. Thacker and S. W. Beet, "Speechreading using shape and intensity information, " In Int. Conf. on Spoken Language Processing, pages 58–61, 1996.

[70] Xiaozheng Zhang, R. M. Mersereau and M. A. Clements, "Audio-visual speech recognition by speechreading, " In 14th Int. Conf. on Digital Signal Processing, volume 2, pages 1069 –1072, 2002.

[71] G. Gravier, G. Potamianos and C. Neti, "Asynchrony modeling for audiovisual speech recognition, " In Int. Conf. of Human Language Technology, USA, 2002.

[72] A. V. Nefian, L. Liang X. Pi, X. Liu and K. Murphy, "Dynamic Bayesian networks for audio-visual speech recognition, " EURASIP Journal of Applied Signal Processing, 002(11):1274–1288, 2002.

[73] Tsuhan Chen, "Audiovisual speech processing, " IEEE Signal Processing Magazine, pages 9–21, Jan. 2001.

[74] J. J. Williams and A. K. Katsaggelos, "An HMM-based speech-to-video synthesizer, " IEEE Trans. Neural Networks, 13(4):900–915, Jul. 2002.

[75] R. Frischholz and U. Dieckmann, "BiolD: A multimodal biometric identification system," Computer,vol.33,no.2, pp.64-68,Feb,2000.

[76] H. Cetingul, Y. Yemez, E. Erzin and A. Tekalp, "Robust lipmotion features for speaker identification," Proceedings of the International Conference on Acoustics, Speech and Signal Processing 2005 (ICASSP ’05), vol. I, 2005, pp. 509–512.

[77] S. S. Liu and M. E. Jernigan, "Texture Analysis and Discrimination in Additive Noise," CVGIP, 49, pp. 52–67, 1990.

[78] Laine, A. and Fan, J., "Texture Classification via Wavelet Pattern Signatures," IEEE Trans. on PAMI, 15(11), pp. 1186–1191, 1993.

[79] Lu, C. S., Chung, P. C. and Chen, C. F., "Unsupervised Texture Segmentation via Wavelet Transform," Pattern Recog., 30(5), pp. 729–742, 1997.

[80]Bovik, A. C., Clark, M. and Geisler, W. S., "Multichannel Texture Analysis using Localized Spatial Filters," IEEE Trans. on PAMI, 12(1), pp. 55–73, 1990.

[81]Jain, A. K. and Farrokhnia, F., "Unsupervised Texture Segmentation using Gabor Filters," Pattern Recog., 24(12), pp. 1186–1191, 1991.

[82]Daugman, J. G., "High Confidence Visual Recognition of Persons using a Test of Statistical Independence," IEEE Trans. on PAMI, 18(8), pp. 1148–1161, 1993.

[83]Pichler, O., Teuner, A. and Hosticka, B. J., "A Comparison of Texture Feature Extraction using Adaptive Gabor Filtering, Pyramidal and Tree Structured Wavelet Transforms," Pattern Recog., 29(5), pp. 733–742, 1996.

[84]Gimmel’farb, G. L. and Jain, A. K., "On Retrieving Textured Images from an Image Database," Pattern Recog., 28(12), pp. 1807–1817, 1996.

[85]Wu, W. and Wei, S., "Rotation and Gray-Scale Transform-Invariant Texture Classification using Spiral Resampling, Subband Decomposition and Hidden Markov Model," IEEE Trans. on Image Processing, 5(10), pp. 1423–1434, 1996.

[86]Randen, T. and Husoy, J. H., "Filtering for Texture Classification: a Comparative Study," IEEE Trans. on PAMI, 21(4), pp. 291–310, 2000

[87]Haralick, R. M., Shanmugam, K. and Dinstein, I., "Textural Features for Image Classification," IEEE Trans. on Systems, Man and Cybernetics, 2, pp. 610–621, 1973.

[88]Weska, J. S., Dyer, C. R. and Rosenfeld, A., "A Comparative Study of Texture Measures for Terrain Classification," IEEE Trans. on SMC, SMC-6(4), pp. 269–285, 1976.

[89] Wu, C. M. and Chen, Y. C., "Statistical Feature Matrix for Texture Analysis," CVGIP:Graphical Models and Image Processing, 54, pp. 407–419, 1992.

[90] Chen, Y. Q., Nixon, M. S. and Thomas, D. W., "Texture Classification using Statistical Geometric Features," Pattern Recog., 28(4), pp. 537–552, 1995.

[91] M.K. Hu, "Visual pattern recognition by moment invariants, " IRE Trans. Inform. Theory 8 (1962) 179-187.

[92] G.B. Gurevich, "Foundations of the Theory of Algebraic Invariants, " Noordhoff, Groningen, The Netherlands, 1964.

[93] S.A. Dudani, K.J. Breeding and R.B. McGhee, "Aircraft identification by moment invariants, " IEEE Trans. Computer. 26 (1977) 39-45.

[94] S.O. Belkasim, M. Shridhar and M. Ahmadi, "Pattern recognition with moment invariants: a comparative study and new results, " Pattern Recognition 24 (1991) 1117-1138.

[95] R.Y. Wong and E.L. Hall, "Scene matching with invariant moments, " Comput. Graphics Image Process. 8 (1978) 16-24.

[96] A. Goshtasby, "Template matching in rotated images, " IEEE Trans. Pattern Anal. Mach. Intell. 7 (1985) 338-344.

[97] J. Flusser and T. Suk, "A moment-based approach to registration of images with affine geometric distortion, " IEEE Trans. Geosci. Remote Sensing 32 (1994) 382-387.

[98] R. Mukundan and K.R. Ramakrishnan, "An iterative solution for object pose parameters using image moments, " Pattern Recognition Lett. 17 (1996) 1279-1284.

[99] R. Mukundan and N.K. Malik, "Attitude estimation using moment invariants, " Pattern Recognition Lett. 14 (1993) 199-205.

[100] A. Sluzek, "Identification and inspection of 2-D objects using new moment-based shape descriptors, " Pattern Recognition Lett. 16 (1995) 687-697.

[101] F. El-Khaly and M.A. Sid-Ahmed, "Machine recognition of optically captured machine printed arabic text, " Pattern Recognition 23 (1990) 1207-1214.

[102] K. Tsirikolias and B.G. Mertzios, "Statistical pattern recognition using efficient two-dimensional moments with applications to character recognition, " Pattern Recognition 26 (1993) 877-882.

[103] A. Khotanzad and Y.H. Hong, "Invariant image recognition by Zernike moments, " IEEE Trans. Pattern Anal. Mach. Intell. 12 (1990) 489-497.

[104] J. Flusser and T. Suk, "Affine moment invariants: A new tool for character recognition, " Pattern Recognition Lett. 15 (1994) 433-436.

[105] S. Maitra, "Moment invariants, " Proc. IEEE 67 (1979) 697-699.

[106] T.M. Hupkens and J. de Clippeleir, "Noise and intensity invariant moments, " Pattern Recognition 16 (1995) 371-376.

[107] L. Wang and G. Healey, "Using Zernike moments for the illumination and geometry invariant classification of multispectral texture, " IEEE Trans. Image Process. 7 (1998) 196-203.

[108] L. van Gool, T. Moons and D. Ungureanu, "Affine/photometric invariants for planar intensity patterns, " Proceedings of the Fourth ECCV'96, vol. 1064, Lecture Notes in Computer Science, Springer, Berlin, 1996, pp. 642-651.

[109] J. Flusser, T. Suk and S. Saic, "Recognition of blurred images by the method of moments, " IEEE Trans. Image Process. 5 (1996) 533-538.

[110] J. Flusser and T. Suk, "Degraded image analysis: an invariant approach, " IEEE Trans. Pattern Anal. Mach. Intell. 20 (1998) 590-603.

[111] M. P. Kumar, P. H. Torr, and A. Zisserman, “Learning layered motion segmentations of video," International Journal of Computer Vision, vol. 76, no. 3, pp. 301-319, 2008.

[112] A. Cavallaro, O. Steiger, and T. Ebrahimi, "Tracking Video Objects in Cluttered Background, " IEEE Transactions on Circuits and Systems for Video Technology, vol. 15, no. 4,pp. 575-584, 2005.

[113] P. H. S. Torr and A. Zisserman, "Feature based methods for structure and motion estimation, " in Workshop on Vision Algorithms, 1999, pp. 278-294.

[114] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey, " ACM Comput. Surv., vol. 38, no. 4, 2006.

[115] C. Rasmussen and G. D. Hager, "Probabilistic data association methods for tracking complex visual objects, " IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 6, pp. 560-576, 2001.

[116] D. Cremers and S. Soatto, "Motion competition: A variational approach to piecewise parametric motion segmentation, " International Journal of Computer Vision, vol. 62, no. 3, pp. 249-265, May 2005.

[117] H. Shen, L. Zhang, B. Huang, and P. Li, "A map approach for joint motion estimation, segmentation, and super resolution, " IEEE Transactions on Image Processing, vol. 16, no. 2, pp. 479-490, 2007.

[118] I. Rekleitis, "Cooperative localization and multi-robot exploration, " PhD in Computer Science, School of Computer Science, McGill University, Montreal, Quebec, Canada, 2003.

[119] N. Vaswani, A. Tannenbaum, and A. Yezzi, "Tracking deforming objects using particle filtering for geometric active contours, " IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 8, pp. 1470-1475, 2007.

[120] Y. Shi and W. C. Karl, "Real-time tracking using level sets, " in Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2. Washington, DC, USA: IEEE Computer Society, 2005, pp. 34-41.

[121] S. Borman, "The expectation maximization algorithm - a short tutorial, " Jul. 2004.

[122] R. Stolkin, A. Greig, M. Hodgetts, and J. Gilby, "An em/e-mrf algorithm for adaptive model based tracking in extremely poor visibility, " Image and Vision Computing, vol. 26, no. 4, pp. 480-495, 2008.

[123] L. Wiskott, "Segmentation from motion: Combining Gabor- and Mallat-wavelets to overcome aperture and correspondence problem, " in Proceedings of the 7th International Conference on Computer Analysis of Images and Patterns, G. Sommer, K. Daniilidis, and J. Pauli, Eds., vol. 1296. Heidelberg: Springer-Verlag, 1997, pp. 329-336.

[124] M. Kong, J.-P. Leduc, B. Ghosh, and V. Wickerhauser, "Spatio-temporal continuous wavelet transforms for motion-based segmentation in real image sequences, " Proceedings of the International Conference on Image Processing, vol. 2, pp. 662-666 vol.2, 4-7 Oct 1998.

[125] B. K. Horn, *Robot Vision*. MIT Press, Cambridge, MA, 1986.

[126] B. K. Horn and B. G. Schunck, "Determining optical flow, " Cambridge, MA, USA, Tech. Rep., 1980.

[127] J. Zhang, F. Shi, J. Wang, and Y. Liu, "3d motion segmentation from straight-line optical flow, " in Multimedia Content Analysis and Mining, 2007, pp. 85-94.

[128] O. Faugeras, R. Deriche, and N. Navab, "Information contained in the motion field of lines and the cooperation between motion and stereo, " International Journal of Imaging Systems and technology, vol. 2, pp. 356-370, 1990.

[129] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts, " in International Conference on Computer Vision, 1999, pp. 377-384.

[130] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: a factorization method, " International Journal of Computer Vision, vol. 9, no. 2, pp. 137-154, 1992.

[131] C. J. Poelman and T. Kanade, "A paraperspective factorization method for shape and motion recovery, " IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 3, pp. 206-218, 1997.

[132] T. Morita and T. Kanade, "A sequential factorization method for recovering shape and motion from image streams, " in Proceedings of the 1994 ARPA Image Understanding Workshop, vol. 2, November 1994, pp. 1177 - 1188.

[133] P. Chen and D. Suter, "Recovering the missing components in a large noisy low-rank matrix: application to sfm, " IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 8, pp. 1051-1063, Aug. 2004.

[134] P. Anandan and M. Irani, "Factorization with uncertainty, " International Journal of Computer Vision, vol. 49, no. 2-3, pp. 101-116, 2002.

[135] T. Okatani and K. Deguchi, "On the wiberg algorithm for matrix factorization in the presence of missing components, " in International Journal of Computer Vision, vol. 72, no. 3, pp. 329-337, 2007.

[136] T. Wiberg, "Computation of principal components when data are missing, " in Proceedings of the Second Symposium of Computational Statistics, Berlin, 1976, pp. 229-236.

[137] R. Hartley and F. Scha_alizky, "Powerfactorization: 3d reconstruction with missing or uncertain data, " in Australia-Japan Advanced Workshop on Computer Vision, 2003.

[138] J. Carme, "Missing data matrix factorization addressing the structure from motion problem, " PhD in Computer Science, Universitat Aut_onoma de Barcelona, 2007.

[139] J. P. Costeira and T. Kanade, "A multibody factorization method for independently moving objects, " in International Journal of Computer Vision, vol. 29, no. 3, pp. 159-179, 1998.

[140] N. Ichimura and F. Tomita, "Motion segmentation based on feature selection from shape matrix, " Systems and Computers in Japan, vol. 31, no. 4, pp. 32-42, 2000.

[141] K. Kanatani and C. Matsunaga, "Estimating the number of independent motions for multi body motion segmentation, " in Proceedings of the Fifth Asian Conference on Computer Vision, vol. 1, Jan 2002, pp. 7-12.

[142] K. ichi Kanatani, "Statistical optimization and geometric visual inference, " in AFPAC '97: Proceedings of the International Workshop on Algebraic Frames for the Perception-Action Cycle. London, UK: Springer-Verlag, 1997, pp. 306-322.

[143] L. Zelnik-Manor and M. Irani, "Degeneracies, dependencies and their implications in multibody and multisequence factorizations, " Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. II-287-93 vol.2, 18-20 June 2003.

[144] Y. Sugaya and K. Kanatani, "Geometric structure of degeneracy for multi-body motionsegmentation, " in Statistical Methods in Video Processing, 2004, pp. 13-25.

[145] L. Zelnik-Manor and M. Irani, "Temporal factorization vs. spatial factorization, " in European Conference on Computer Vision, vol. 2. Springer Berlin / Heidelberg, 2004, pp. 434-445.

[146] A. D. Bue, X. Llado, and L. Agapito, "Segmentation of rigid motion from non-rigid 2d trajectories, " in Pattern Recognition and Image Analysis. SpringerLink, 2007, pp. 491-498.

[147] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, " Commun. ACM, vol. 24,no. 6, pp. 381-395, 1981.

[148] F.-H. Cheng and Y.-L. Chen, "Real time multiple objects tracking and identification based on discrete wavelet transform, " Pattern Recognition, vol. 39, no. 6, pp. 1126-1139, 2006.

[149] R. Li, S. Yu, and X. Yang, "Efficient spatio-temporal segmentation for extracting moving objects in video sequences, " IEEE Transactions on Consumer Electronics, vol. 53, no. 3,pp. 1161-1167, Aug. 2007.

[150] A. Colombari, A. Fusiello, and V. Murino, "Segmentation and tracking of multiple video objects, " Pattern Recognition, vol. 40, no. 4, pp. 1307-1317, 2007.

[151] Crowley, J. and Berard, F., "Multi-Modal Tracking of Faces for Video Communication, " Proc. IEEE CVPR., pp. 640-647, Puerto Rico. 1997.

[152] G. Bradski, "Computer vision face tracking for use in a perceptual user interface, " in *Intel Technology Journal*, 2nd Quarter, 1998.

[153] Y. Raja, S. J. McKenna, and S. Gong, "Tracking and segmenting people in varying lighting conditions using colour, " In IEEE International Conference on Face & Gesture Recognition, pages 228-233, Nara, Japan, 1998.

[154] Crowley, J. L. and Coutaz, J., "Vision for Man Machine Interaction, " Robotics and Autonomous Systems, Vol. 19, pp. 347-358 (1997)

[155] R. Brunelli, Template Matching Techniques in Computer Vision: Theory and Practice, Wiley, ISBN 978-0-470-51706-2, 2009.

[156] Aksoy, M. S., O. Torkul, and I. H. Cedimoglu. "An industrial visual inspection system that uses inductive learning, " Journal of Intelligent Manufacturing 15.4 (August 2004): 569(6). Expanded Academic ASAP. Thomson Gale.

[157] Kyriacou, Theocharis, Guido Bugmann, and Stanislao Lauria, "Vision-based urban navigation procedures for verbally instructed robots, " Robotics and Autonomous Systems 51.1 (April 30, 2005): 69-80. Expanded Academic ASAP. Thomson Gale.

[158] WANG and CHING YANG, "Edge Detection Using Template Matching, " Ph.D. Duke University, 1985, 288 pages; AAT 8523046

[159] Li, Yuhai, L. Jian, T. Jinwen and X. Honbo, "A fast rotated template matching based on point feature, " Proceedings of the SPIE 6043 (2005): 453-459. MIPPR 2005: SAR and Multispectral Image Processing.

[160] Munevver Kokuer, "Model-Based Coding for Human Imagery, " Ph.D. Thesis, University of Essex, Department of electronic systems engineering, UK, January 1994.

[161] B. Jahne., *Handbook of Digital Image Processing for Scientific Applications*. CRC Press, Boca Raton, FL, 1997.

[162] A. Bobick and J. Davis, "An appearance-based representation of action, " IEEE International Conference on Pattern Recognition, pp. 307-312, 1996.