ÉCOLE CENTRALE MARSEILLE
AIX-MARSEILLE UNIVERSITÉ

Titre:

TRAITEMENT DU SIGNAL POUR LA RECONNAISSANCE DE GESTES ET APPLICATION A UNE INTERFACE HOMME MACHINE SANS CONTACT

THÈSE

pour obtenir le grade de DOCTEUR

délivré par l'ÉCOLE CENTRALE MARSEILLE

École Doctorale : Physique et Sciences de la Matière
Mention : Optique, Photonique et Traitement d'Image

Effectuée à INSTITUT FRESNEL
Présentée et soutenue publiquement par:

Nabil BOUGHNIM

le 19 Septembre 2013

Directeur de thèse : Pr. Salah BOURENNANE
Co-directeur de thèse : Dr. Caroline FOSSATI

JURY :

Rapporteurs:	Pr. Latifa HAMAMI	École Nationale Polytechnique d'Alger
	Pr. Yide WANG	École polytechnique de l'université de Nantes
Examinateurs:	Pr. Rachid OUTBIB	Université Aix-Marseille
	Pr. Salah BOURENNANE	École Centrale Marseille
	Dr. Caroline FOSSATI	École Centrale Marseille
	Dr. Julien MAROT	Université Aix-Marseille
	Dr. Frederic GUERAULT	Intui-sense Technologies, Aubagne

ANNEE : 2013

ÉCOLE CENTRALE MARSEILLE
UNIVERSITÉ AIX-MARSEILLE

N° assigned by library
2013AIX.....

Title:

SIGNAL PROCESSING FOR GESTURE RECOGNITION AND APPLICATION TO A TOUCHLESS HUMAN MACHINE INTERFACE

THESIS
to obtain the degree of DOCTOR

issued by: ÉCOLE CENTRALE MARSEILLE

Doctoral school : Physics and material Sciences
Discipline : Optics, Photonics and Image Prossesing

Carried out at INSTITUTE FRESNEL
Presented and defended publicly by :

Nabil BOUGHNIM

on September 19, 2013

Thesis advisor : Pr. Salah Bourennane
Co-advisor : Dr. Caroline Fossati

COMMITEE:

Reviewers:	Pr. Latifa Hamami	École Nationale Polytechnique d'Alger
	Pr. Yide Wang	École polytechnique de l'université de Nantes
Examiners:	Pr. Rachid Outbib	Université Aix-Marseille
	Pr. Salah Bourennane	École Centrale Marseille
	Dr. Caroline Fossati	École Centrale Marseille
	Dr. Julien Marot	Université Aix-Marseille
	Dr. Frederic Guerault	Intui-sense Technologies, Aubagne

YEAR : 2013

Abstract

THIS thesis is devoted to the development of computer vision methods which must be robust to variations due to acquisition conditions and processing in real-time in applicative contexts.

The objective is to create a touchless human-machine interface (HMI). At first, we describe the various problems which are specific to the existing databases. At the same time we present the principal postures that compose the dictionary of gestures which we retained. This leads us to conclude that we need to create our own database. In a second phase, we are interested in a gesture recognition system that can be decomposed into 3 steps: detection, characterization and recognition.

In the detection step we mentioned two types of detection methods: one for static gestures and the second for dynamic gestures (movements), we adapt optical flow techniques to hand detection. This adaptation allows us to extend the detection of static gestures regardless of the color of the skin and track the trajectory of the hand in a video stream.

The characterization step commits in transforming an image into a set of signals which characterizes a clearly defined posture by its contour. We notice that a hand contour is generally non star-shaped, so we apply the methods adapted from array processing to this type of contours which have given previously convincing results. We propose a new signature which involves the generation of signals. We describe the generation of different signals and we show the various invariance properties of this new characterization method.

The proposed signature is a sparse matrix of considerable size, hence our proposal to apply principal component analysis (PCA) to reduce the dimension of matrix signature. We also reduce the dimension of the test vocabulary set, through a first rejection test based on a geometric criterion (the isometric rate). The basic principles of the recognition step are as follows: a learning phase permits to define a set of reference signatures. In the subsequent test phase, the signature obtained from the tested images is compared with the reference signatures.

We present recognition results obtained with dimension reduction by PCA and by adopting the Euclidean and Mahalonobis distances. Comparative methods are also considered: we discuss the advantages and limitations of our methods, the recognition rate and the computational load.

Keywords: Hand posture; gesture recognition; classification algorithm; principal component analysis; biometrics; array processing; optical flow; hand database; human-computer interaction.

Résumé

CETTE thèse est consacrée au développement des méthodes de vision par ordinateur robustes aux variations dues aux conditions pratiques et exploitable en temps réel dans des contextes applicatifs.

L'objectif est de créer une interface homme-machine sans contact. Dans un premier temps, nous décrivons les différents problèmes spécifiques aux bases de données existantes et les principaux postures qui vont servir pour construire et fixer le dictionnaire de gestes qui nous avons retenu. Ce qui nous à conduit à conclure à la nécessité de créer notre propre base de données. Dans un deuxième temps, nous nous sommes intéressés au système de reconnaissance gestuelle qui peut être décomposé en 3 étapes : la détection, la caractérisation et la reconnaissance.

Dans l'étape de détection nous avons mentionné deux types de détection: la première pour les gestes statiques et la seconde pour les gestes dynamique (mouvements), nous montrons l'adaptation des techniques de flux optique pour la détection de la main. Cette adaptation nous permet d'étendre la détection de gestes statiques indépendamment de la couleur de la peau et de suivre la trajectoire de la main dans le flux vidéo.

L'étape de caractérisation consiste à transformer une image en un ensemble de signaux qui caractérise une posture clairement définie par son contour et qui permet de comparer ces critères avec des critères de postures stockées et définis à l'étape d'apprentissage. Nous notons que le contour de la main peut être un contour non étoilé, par conséquent, nous appliquons des méthodes de traitement d'antenne qui ont déjà donné de bons résultats pour ce type de contours.

Nous détaillons la génération de différents signaux et nous montrons les différentes propriétés d'invariance de cette nouvelle méthode de caractérisation. La signature proposée est une matrice creuse de taille considérable, d'où nous avons proposé d'appliquer l'analyse en composantes principales (PCA) pour réduire la dimension des données. Nous réduisons également la dimension de l'ensemble de vocabulaire de test à travers un premier rejet basé sur le critère géométrique (taux isométrique).

Nous présentons les résultats de la reconnaissance obtenus avec réduction de dimension par PCA et en adoptant les distances euclidienne et de Mahalonobis, et nous les comparons avec d'autres méthodes. Finalement, nous discutons les avantages et les limites de nos méthodes ainsi que le taux de reconnaissance et le temps de calcul.

Mots clé: Posture de la main; reconnaissance des gestes; algorithme de classification, analyse en composantes principales; biométrie, traitement d'antenne, flux optique, interaction homme-machine.

Acknowledgements

THIS work in this thesis was carried out at GSM Group (Groupe Signaux Multidimensionnels) of Fresnel Institute, in Marseille, France. I am greatly indebted to many people, who have provided endless support and help.

First, I wish to express my sincerest gratitude to the "Conseil régional Provence Alpes Côte d'Azur (PACA)", and the firm "Intui-Sense Technologies". They have offered me the great opportunity to work in this thesis and start a new step in my life.

Coming to the work's environment, I am deeply grateful to my supervisors Prof. Salah Bourennane and Dr. Caroline Fossati who welcomed me to the GSM team and gave me their significant guidance and constant encouragement during these three years. Their vast knowledge and unfailing trust have been incomparably useful in shaping my attitude toward research. Salah Bourennane managed to sustain my initiatives at all opportune moments, his unmatched enthusiasm and above all his unfailing trust made me feel confident about my work. His rigorous way of considering theoretical issues were very profitable.

I would like to take this opportunity to express my sincere gratitude to Dr. Julien Marot. I am grateful for having had the privilege of working with him on a number of technical results presented in his dissertation and elsewhere, he provided me with valuable advice and practical help, concerning my research investigations or for submitting articles for publication in journal or conference papers. Thanks to his ready assistance, I could progress constantly during my thesis and complete this research work.

Special gratitude goes to Prof. Latifa Hamami from École Nationale Polytechnique d'Alger and Prof. Yide Wang from Ecole Polytechnique de l'Université Nantes, for honoring me while constructively reviewing my thesis. I also wish to thank Prof. Rachid Outbib from Aix-Marseille University, for presiding over my PhD jury, as well as for his interest in my work and his constructive remarks and suggestions. I specially thank Dr. Frédéric Guérault for being present at my PhD talk and also for his remarks and suggestions during the three years of my thesis.

I would also like to thank the researchers of GSM team at Fresnel Institute. I would like to thank Dr. Mouloud Adel, Dr. Mohammad-Ali Khalighi, Dr. Stéphane Derrode for their support of my work. Having in mind the memorable time I have spent in GSM group, many thanks go to all my lab colleagues for their sincere help, both former and current ones, including Riad Khelifi, Sylvain Jay, Xuefeng Liu, Alexis Cailly, lin Tao, Valérian Némesin et al.

Finally, I am extremely grateful to my parents who live in my home country Tunisia. Adjacently, I am deeply grateful to my wife Sonia who has always supported me and given birth to my pearl ESSIL whose are my happiness and my motivation.

Contents

Introduction

General context

THE subject of our research concerns the conception and the development of methods of computer vision for hand gesture recognition. Our work is inserted in the design of a human-machine interface which aims transforming a classical screen in an interface without contact and at allowing the use of the finger as a pointing device. The hand gestures are a natural and intuitive way of communication which allow humans to interact with their environment. They permit to designate or manipulate objects, to enhance the speech, or to communicate basically in a noisy environment. They can also represent a language in its own right with sign language. Gestures can have a different signification depending on the language and culture : the sign languages in particular are specific to each culture.
Thinking on what to use as gestures or postures is necessary, to ensure that users can intuitively realize them, or with a limited period of learning. What gestures should you use? Are they easy to reproduce? To what actions are they intuitively associated? These are the questions that should be asked while building a gesture database.
In general, the gesture is assimilated to all the movements of a body part. The hand gesture is both a means of action, perception and communication.

For Cadoz [25], the gesture is one of the richest way of communication. Thus, in the field of Human-Machine Interfaces (HMI), the hand can be used to point (to replace the mouse), to manipulate objects (for augmented or virtual reality), or to communicate with a computer through gestures. Compared to the affluence of information conveyed by hand gestures, the possibilities of communication with computers are reduced today with the mouse and keyboard. The man-machine interaction is currently based on the WIMP (Window, Icon, Menu, Pointing device) paradigm that presents the functional basis for a computer graphical interface.

The majority of operating systems are based on this concept, with a pointing device, usually a mouse, which allows to interact with graphical elements such as windows, icons and menus, we can say with a more intuitive way than the textual interface (command line). Using hand gestures, the interface becomes perceptual (PUI 3).
The gesture recognition systems first used electronic gloves with sensors providing the hand position and angles of the finger joints. But these gloves are expensive and

bulky, hence the growing interest for the methods of computer vision. Indeed, with the technological progress and the apparition of cheap cameras, it is now possible to develop systems of gesture recognition based on computer vision, running in real time. However, the hand being a complex organ, deformable, having a many degrees of liberty in the joints, it is difficult to recognize its form images without some limits and priors. Indeed, human beings can naturally perform a very large number of different gestures.

With the development of acquisition technologies and gesture recognition techniques, many application domains have emerged :

- Recognition of sign language.
- The Virtual reality, where the hand is used to manipulate virtual objects and trigger actions, or navigate within a virtual environment.
- The Augmented reality, where the physical world is increasing with virtual information, for example by a retro-projection.
- The Multimodal applications, combining gesture with other means of communication, such as speech or facial expressions.
- The Coding and the transmission of gestures with low output for Tele-conference.
- The biometry, for the recognition of persons with the hand form.

Subject of research and industrial context

We aim at developing computer vision methods that meet specific criteria, in an applied context. Indeed, computer vision offers many possibilities, but some solutions are not suitable for our application, mainly because of a lack of robustness to the actual conditions or too much complexity to be implemented in real time.

For economic and hygiene reason, this project is based on the development of a touchless human-machine interface, and allows to transform a classic screen into a tactile one. Manipulating an interface without having to touch it reduces the maintaining costs, generalizes its use thanks to hygiene standards, and makes interaction more convivial. The industry collaboration behind this thesis has guided the choice of the materials used, the selected methods and constraints to solve.

The thesis was conducted in the context of a project funded by the PACA region (Provence-Alpe-Cote d'Azur) and the firm Intui-sense technologies. Intui Sense provides interactive solutions with intuitive interfaces based on innovative touchless technologies for retail applications, in particular for the vending industry.

Among the constraints imposed and to resolve, we cite:

- cameras of low-cost types,
- processing in real-time,
- methods must be robust to acquisition conditions,
- constraints imposed on users should be minimal,
- presence of other objects in the field of camera,
- treatment of fast and slow motion.

A study of the literature on the field is necessary and will allow us to analyze the different approaches and choose the most suitable approach to our application. We then propose techniques to implement the various steps of a gesture recognition system, decomposed according to the following scheme:

1. for each frame:
    - detection and segmentation of the hand,
    - extraction of features representing the posture of the hand,
    - extraction of the center of the hand,
    - recognition of gestures from a predefined set (dictionary).
2. for the video stream:
    - track of the center of the hand to determine its trajectory

Organization of the manuscript

This manuscript is organized in two parts.

1. A first part consists of two chapters:
    - Chapter 1 presents the state of the art of the whole process of gesture recognition process. This chapter is divided into three sections: section 1.2 that includes various methods and techniques used for the detection and segmentation, section 1.3 is devoted to the aspect of characterization and extraction of features that can well describe the same posture with different transformations, section 1.4 shows the different techniques used in the literature to discriminate and distinguish different classes.
    - In Chapter 2 we describe the different problems of existing databases and different postures which allows us to build and fix our dictionary of gestures. We deduce from these issues the necessity to create our own database with our postures. It seems very important to us to conduct this project properly forward.

2. The second part consists of four chapters:

    - Chapter 3 is devoted to the tools of image processing adopted from array processing. We remind in section 3.2 the detection of straight contours in images and in section 3.3 we extend these methods to circular contours. Section 3.4 shows the technique to determine a blurred contour. In the last section (3.5) we discuss the adaptation of these methods of array processing to distorted circular contours. We focus on the characterization of star-shaped contours which are strongly distorted. Noticing that the hand contour is approximately circular and very distorted, we decided to include these methods in the characterization of hand postures. However, it has been necessary to adapt these techniques because none of them handles the case of non star-shaped contours.
    - Chapter 4 is devoted to the definition of a new feature extraction method for hand postures. We propose a new signature which involves the generation of signals. We detail how the different signals are generated and we prove the different properties of this new characterization method. Finally, we explain the technique of dimension reduction with PCA and its relevance.
    - In Chapter 5 we define the optical flow technique, which is used for tracking and smoothness and we prove the adaptation of this technique for the detection of the hand. This adaptation allows us to extend the detection to colored people hand, which was not treated yet.
    - The final chapter (6) contains the different results and the whole process of our algorithm. We detail the different preprocessings used to improve the different process steps. We present the results obtained with the new approaches used for the recognition and we compare them with other methods. Eventually we discuss the advantages and limitations of our methods as well as the recognition rate and the computation load.

We finalize the manuscript by a general conclusion, as well as further prospects.

Part I

State of the art and hand database

CHAPTER 1 State of the art

1.1 Introduction of the chapter

WITH the development of computer systems and their ever growing embedded presence into our daily life, the question of convenient and natural types of human-computer interaction becomes crucial. If user-computer relationships have already evolved in that sense, going from cumbersome text-based command lines to dedicated devices such as mouse or pen, they still remain restrictive. One way to simplify the means of interacting with computers consists in using hand gesture interfaces.

Two ways exist to turn hand gestures understandable by computers. The first one relies on the use of extra sensors, such as magnetic ones or data gloves. If these instruments often help in collecting accurate information, they also act as a brake upon free movements. The load of cables connected to the computer, induced by this approach, indeed hinders the ease of the user interaction. A less intrusive solution resorts to vision-based systems. Even though it is difficult to intend a generic interface using this technique, this approach has many appealing advantages. The most interesting among these is undoubtly the naturalness of interaction, which results in a much more intuitive communication between human and computers. Many application domains take interest in gesture interaction, one can quote among others : computer games development, virtual reality, robot control or sign language interpretation.

Systems that employ hand driven Human-machine interfaces (HMI) interpret hand gestures and postures in different modes of interaction depending on the application domain. Previous works have concentrated on hand gesture classification [19, 115], where gesture command is based on slow movements with large amplitude (see for instance in [115] the twelve types of hand gestures). To our knowledge, future applications should concern the classification of hand posture, for the purpose of automated sign language decoding for instance. Contrary to hand gesture, hand posture describes the hand shape and not its movement.

A hand can exhibit a great variety of postures, and it is extremely difficult to recognize all possible configurations of the hand starting from its projection on a 2-D image. Indeed, some parts of the hand can be hidden. It is necessary to consider subsets of postures depending on the application. Different technologies have been developed in order to recognize gestures. It is therefore difficult to achieve a state of the exhaustive art of the field. We try, in this chapter, to present a state of the art of some approaches based on computer vision in the context of Human-Machine Interfaces. Generally, a gesture recognition system can be decomposed in several steps: detection, characterization and recognition. The questions that arise here, and for which we have responded in different sections are the following: how can we detect the hand in any scene? How can we characterize the hand numerically? What methods are used to classify or rather recognize the type of posture?

1.2 Hand detection

A hand is the source of a wide variety of postures. Different devices allow interaction with a computer through the hand (mouse, data gloves, touches screens, ...). However, these devices have some limitations. Moreover, the scientific and technical developments offer new possibilities of interaction, more natural and intuitive, based on gestural channel. There are many applications such as the augmented or virtual reality, the recognition of sign language, the control articulated arms, or the biometrics. One of the more developed applications consists on making an interactive surface. In detection step we can distinguish two main categories of gestures: static gestures and dynamic gestures.

1.2.1 Static gestures recognition

The basic aim of this step is to optimally prepare the image obtained from a camera in order to extract the features in the next step. How an optimal result looks like depends mainly on the next step, since some approaches only need an approximate bounding box of the hand, whereas others need a properly segmented hand region in order to get the hand silhouette. In general, some regions of interest, that will be subject of further analysis in the next step, are searched in this phase.

The most commonly used technic to determine the regions of interest is skin color detection. A previously created probabilistic model of skin-color is used to calculate the probability of each pixel to represent some skin. Thresholding then leads to the coarse regions of interest. Analysis of the skin color is used to detect the face and hands. Indeed, Jones and Rehg [63] have shown that skin color has a characteristic distribution in certain color spaces, and that this property can be used to segment regions of skin color, regions are delimited by contours.

A rule of thumb about contour characterization methods such as Fourier descriptors [19, 34] is that they require a binary image I, possibly noise-free. The same constraint holds in the frame of our work. To perform hand contour detection, some classical preprocessing methods have been applied in previous works [15, 16, 19, 34]: the YC_bC_r mapping, using the YC_bC_r space, which consists of a luminance component (Y) and two chrominance (C_b and C_r) and the selection of the C_b component, emphasize the hand surface with respect to the background. The transformation is linear with the RGB space. The non-moving background is then removed, by substraction of a frame where the hand is not present.

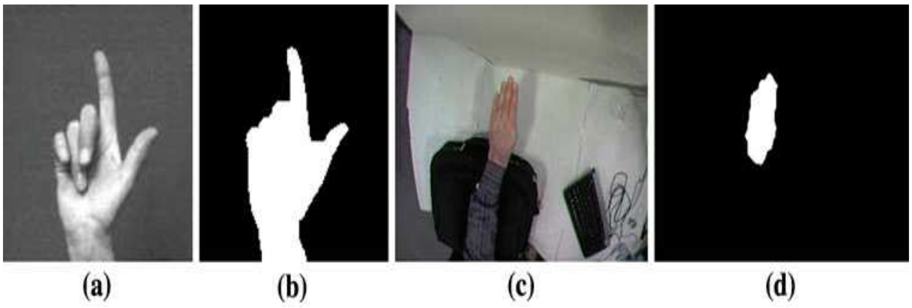


Figure 1.1 — Hand segmentation examples: (a) and (b), on a gray image from the Triesch database, with threshold; (c) and (d) from internal database, with thresholds on C_b and C_r.

There are many other color spaces, the most used are RGB, HSV and YCbCr. Phung *et al.* [89] compared the performance of these spaces and they found out that the results are very similar, regardless of the color space. Thus, the choice of a color space must be depending on the format of the images and any pre-treatment. Some further analysis could for example involve the size or perimeter of the located regions in order to exclude regions such as the face.

In [99], Soriano *et al.* propose a dynamic skin color model, for a segmentation purpose. Their method copes with changes in illumination. However, their method is applied to faces and not to hands. In [112], a set of relevant grey level values are selected from chromatic histograms to segment face. To create a chromatic histogram, an HSI mapping is performed, and a 2-D map of the couples (H,S) for each pixel is computed. The chromatic histogram exhibits the advantage of being insensitive to scaling, and rotation. However, authors must combine the chromatic histogram with the prior knowledge of the approximate shape of faces to detect them. The main drawback of YC_bC_r or HSI mappings is that they do not handle hands of colored people.

Yet another interesting approach is to use a previously acquired image of the background, substracting it from the image with the posture, as proposed in [95]. Based on perimeter lengths, the hand region can then be extracted.

1.2.2 Dynamic gestures recognition

A dynamic gesture corresponds to a time variation in the shape and the position of the hand. The first challenge is to locate temporally the realization of a gesture, that is to say, to determine the start and end of the gesture. A gesture is divided into three stages: a preparatory phase, gesture, and withdrawal phase. A major difficulty arises from the variation of the period of execution of a same gesture. It is therefore necessary to perform temporally normalization of the duration of the observations.

The Dynamic Time Warping (DTW) compares two temporally sequences of different lengths, stretching or reducing their length, implying that the beginning and end of the gesture are determined. Darrell and Pentland [36] use this method: gestures are modeled by scores of correlation with a set of models, which are accumulated to form a signature. The Dynamic Time Warping allows comparing signatures.

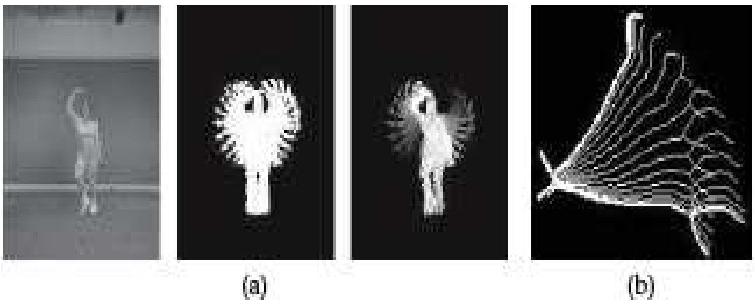


Figure 1.2 — Dynamic gestures: (a) MEI and MHI [14], and (b) signature of a dynamic gesture by superimposing the skeletons of sequence images [59].

Bobick and Davis [14] use temporal models for the recognition of human movement: the "image of the motion energy" (MEI), and the "image of the movement history "(MHI). These images are formed by the accumulation of motions of each pixel over a time window (see fig 1.2 (a)). The images are described with the invariants of Hu, and gestures are classified using the Mahalanobis distance. Ionescu *et al.* [59] propose a method for dynamic gesture recognition based on skeletons. Static signatures of the beginning and the end of gestures are calculated with a Histogram of Oriented Gradient. The dynamic signature is obtained by superimposing the skeletons of sequence images (see fig 1.2 (b)). Zhu *et al.* [115] segment the hand with the color, associated with motion detection.

The spatio-temporal representation of a gesture is made with motion estimation based on a parametric model and a description the shape of the hand with the geometrical moments. After a temporal normalization with a method of linear sampling, the recognition is performed with a distance with models that were learned previously. In their application, 12 gestures are used to navigate with a panoramic view. Kong and Ranganath [68] use a hierarchical approach to recognize 3d trajectories, periodic or not (see figure 1.3). The detection of periodicity is based on Fourier analysis. The trajectories are then recognized with a variant of the ACP.

The Hidden Markov Models (HMM) have been successfully used for long time in the field of speech recognition. By analogy, they have been used for gesture recognition and interpretation of sign language, first with data gloves (Braffort [23]), then with computer vision where different models have been developed. Among the first studies in this field, Starner and Pentland [100, 101] use the HMM for the recognition of 40 signs from the American Sign Language (ASL), with a single camera.

The features used are the center of the hand and elliptical bounding box, obtained with the principal axes. Marcel *et al.* [74] propose a hybrid approach between HMM

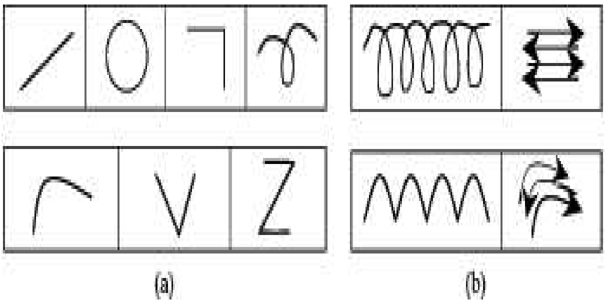


Figure 1.3 — 3D trajectories [68] : (a) non-periodic and (b) periodic.

and neural networks, called "Input-Output Hidden Markov Models", to recognize four gestures in using the center of gravity of the hand. Wilson and Bobick [109] propose a HMM parametric form, to estimate the direction of movement in a pointing gesture. Vogler and Metaxas [105, 106] propose the "Parallel HMM " to model separately the left and right hands, and to recognize 53 gestures of American Sign Language, continuously.

Sato *et al.* [85, 93] track a monitoring of the hand and the fingertips, in two dimensions, for Enhanced Desk system. An infra-red camera facilitates the detection of the hands, and then each finger tip is detected by correlation with a circle, and followed with a Kalman filter. The thumb is detected to differentiate a "handling" mode from a "symbolic gesture" mode. The symbolic gestures recognition is based on HMM with 12 different gestures (see figure 1.4). Similarly, Martin and Durand [79] use HMM for handwriting recognition in 2D, with letters from an alphabet.

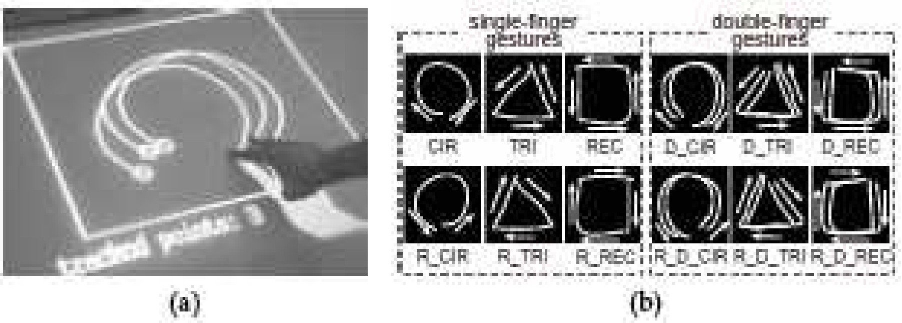


Figure 1.4 — The EnhancedDesk system [85] : (a) track multiple finger tips, and (b) trajectories recognized by HMM.

1.3 Hand characterization

In this section, we focus on the extraction of a vector or matrix of features to represent the shape of the hand. Since the appearance of the hand in an image can vary greatly depending on the perspective, for the same configuration, we seek euclidean transformations (translation, rotation, scaling), which represent most of the changes we face.

In order to characterize an object (various hand postures) that can appear at different scales and orientations, descriptors which are invariant to these transformations must be used.

The descriptors can be divided into four classes: the global descriptors that work on the entire image, the semi-local descriptors that work on a set of sub-images representing cuts of the complete image, the local descriptors that combine interest points detection and characterization of the neighborhood of each detected keypoint and the geometric descriptors that utilize low level features to express object shape. In the following paragraphs, we detail some descriptors for each class.

1.3.1 Global approach

• **Zernike moments** [66] are built around a family of complex polynomials forming an orthogonal basis, defined in the unit circle. This orthogonal basis can reduce the redundancy between the moments. Standardizations can turn these descriptors invariant to transformations involving rotations, translations and scaling.

$$A_{mn} = \frac{m+1}{\pi} \sum_x \sum_y I(x,y) V^*_{mn}(x,y) \tag{1.1}$$

Where $x^2 + y^2 \leq 1, m = 0, 1, 2..., \infty$ is the moment's order and n is an integer respecting the following conditions:

$$\begin{cases} m - |n| & \text{is an even number} \\ |n| \leq m & \end{cases}$$

The Zernike descriptor is among the most used in the literature (see equation (1.1)). It is built from a set of Zernike polynomials. This set is complete and orthonormal inside the unit circle.

$$V_{mn}(r,\theta) = R_{mn}(r) e^{jn\theta} \tag{1.2}$$

with (r,θ) defined on the unit disk, and $R_{mn}(r)$ is the radial polynomial.

$$R_{mn}(r) = \sum_{s=0}^{\frac{m-|n|}{2}} (-1)^s \frac{(m-s)!}{s!(\frac{m+|n|}{2} - s)!(\frac{m-|n|}{2} - s)!)} r^{m-2s} \tag{1.3}$$

The Zernike moments have shown their performance in terms of robustness to noise and near zero value in redundancy of information. Modules of Zernike moments are invariant to rotation. To obtain the translational invariance and scaling, the images are normalized using the moments of order 0 and 1. According to Kumar and Singh [69], it is sufficient for the recognition to the moments of order 2 to 15, which represent 70 moments. The major drawback of Zernike moments is their elevated computational load. Various methods have been proposed (Hwang and Kim [56]) to allow faster computation times. Chong *et al.* [32] compare different methods available and offer to calculate the moments up to order 24 in 50 milliseconds instead of 1,10 seconds using the direct method for a binary image of 50×50 pixels.

• **Hu moments** [52], compound a family of invariants which have been used for a long time for recognition. The knowledge of the center of gravity (x_G, y_G) of the region is required to calculate the centered moments, u_{pq}:

$$u_{pq} = \sum_{(x,y)\in I} (x - x_G)^p (y - y_G)^q I(x, y) \tag{1.4}$$

The centered moments are invariant to translations. To obtain invariance to scaling factor, normalized moments are calculated:

$$\eta_{pq} = \frac{u_{pq}}{u_{pq}^{\gamma}} \text{ with } \gamma = \frac{p+q}{2} + 1, \forall\, p + q \geq 2 \tag{1.5}$$

Using normalized moments up to order 3, we can calculate the seven Hu moment invariants:

$$I_1 = \eta_{20} + \eta_{02} \tag{1.6}$$

$$I_2 = (\eta_{20} + \eta_{02})^2 + 4\eta_{11}^2 \tag{1.7}$$

$$I_3 = (\eta_{30} + 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \tag{1.8}$$

$$I_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} - \eta_{03})^2 \tag{1.9}$$

$$\begin{aligned} I_5 &= (\eta_{30} + 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ &+ (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \end{aligned} \tag{1.10}$$

$$I_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})] \tag{1.11}$$

$$\begin{aligned} I_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ &+ (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \end{aligned} \tag{1.12}$$

The first six features characterize the shape with invariance to translation, rotation and scaling. The seventh invariant distinguishes symmetrical shapes.

• **Fourier descriptors** (FD) were known thanks [35, 88]. They are extensively used for the characterization and shape classification for a closed contour, as they allow a good representation of shapes and have interesting invariance properties. FD are calculated from the coefficients of the Fourier transform of the contour . Fourier descriptors have been usually used for gesture recognition [31, 71, 84] as one component of a complete system of recognition. Thus, the performance of the FD has not been analyzed in detail, and independently of other system components. In general, in existing work, complex signature is used, as well as the module of the Fourier coefficients (FD1). The second family of descriptors (FD2) has not been used for gesture recognition.

FD are calculated on the contour of the hand region, extracted from the segmented image. Points of this contour can be represented with various signatures (complex coordinates, central distance, curvature, cumulative angular function) [113]. We consider the case of closed planar curves under the action of Euclidean transformations. If $\gamma_1(l)$ and $\gamma_2(l)$ denote the respective arclength parametrization of two closed contour objects, having the same shape and different poses, we can write [30,31]:

$$\gamma_2(l) = ae^{j\theta}\gamma_1(l + l_0) + b \tag{1.13}$$

with a the scale factor, θ the rotation angle, b the translation and l_0 the difference starting between description points , $l_0 \in [0, L]$ with L the length of the contour. The scale invariance is obtained by normalizing the arc-length parametrization with an equal length of 1, leading to $l_0 \in [0, 1]$. The translation invariance is given by describing the contours according to their center of mass.

Before calculating the Fourier Transform, with the Fast Fourier Transform (FFT), shape is first sampled to a fixed number of points. In general, object shape and model shape can have different sizes. Consequently, the number of data points of the object and model representations will also be different. For matching purposes, the shape boundary or the shape signature of objects and models must be sampled to have the same number of data points. The sampling process not only normalizes the size of shapes but also has the effect of smoothing the shape. The smoothing eliminates the noise in the shape boundary and the small details along the shape boundary as well, what may be a drawback in a hand posture recognition method.

The number of resolution levels at which the shape signature will be decomposed is determined by the length of the shape boundary. By varying the number of sampled points, the accuracy of the shape representation can be adjusted. The larger the number of sampled points, the more details in the representation of the shape; consequently, the matching result will be more accurate. In contrast, a smaller number of sampled points reduce the accuracy of the matching results but improve the computational efficiency.

There are generally three methods of normalization : (i) equal points sampling; (ii) equal angle sampling; (iii) equal arc-length sampling. Assuming N is the total number of candidate points to be sampled along the shape boundary, the equal angle sampling selects candidate points spaced at equal angle $\theta = 2\frac{\pi}{N}$.
The equal points sampling method selects candidate points spaced at equal number of points along the shape boundary. The space between two consecutive candidate points

is given by P/N, where P is the total number of boundary points. The equal arc-length sampling method selects candidate points spaced at equal arc length along the shape boundary.

The space between two consecutive candidate points is given by L/N, where L is the perimeter of the shape boundary. Among the three sampling methods, the equal arc-length sampling method apparently achieves the best equal space effect, because the use of arc length as parameter in the signature achieves the unit speed of motion along the shape boundary [87].

We use the complex coordinates, each point M_i of the shape contour is represented by a complex number z_i, with N the number of points of the contour:

$$\forall i \in [0, N-1], M_i(x_i, y_i) \Leftrightarrow z_i = x_i + jy_i \tag{1.14}$$

This number must be chosen as a compromise between a reliable description of the shape, with enough details, and shape smoothing, which eliminates the finest details more subject to noise. Therefore, we choose the equal arc-length sampling to normalize the sizes of the shapes. For each shape, we select 64 candidate points with equal arc-length space between them. Another factor is the computation time, which increases with the number of points. For computational efficiency of the fast Fourier transform, the number of points is chosen to be a power of two. Hence, the Fourier transform leads to N Fourier coefficients C_k :

$$C_k(\gamma) = \sum_{i=0}^{N-1} z_i e^{-j\frac{2\pi ik}{N}}, k = 0, ..., N-1. \tag{1.15}$$

In the frequency domain, Eq.(1.13) and Eq.(1.15) gives:

$$C_k(\gamma_2) = e^{j\theta} e^{j\frac{2\pi kl_0}{N}} C_{k(\gamma_1)} + b\delta_k. \tag{1.16}$$

where δ_k is the Kronecker delta. The first coefficient C_0 is discarded because it contains only the position of the hand shape. Rotation of the shape affects only the phase information, thus rotation invariance of the Fourier descriptors is achieved by taking the magnitude of coefficients. Scale invariance is achieved by dividing coefficients by the magnitude of the second coefficient, C_1. Starting point invariance is also achieved by taking the magnitude, as a change of the starting point affects only the phase. So, Eq.(1.16) can be written as follows:

$$C_k(\gamma_2) = e^{j\theta} e^{j\frac{2\pi kl_0}{N}} C_{k(\gamma_1)}, k = 0, ..., N-1. \tag{1.17}$$

A common way to obtain FD which are invariant to similarities is to take the magnitude of Fourier coefficients [35, 88]. Then, we obtain the $N-2$ FD1 coefficients:

$$I_k(\gamma) = \frac{|C_k(\gamma)|}{|C_1(\gamma)|}, k = 2, ..., N-1. \tag{1.18}$$

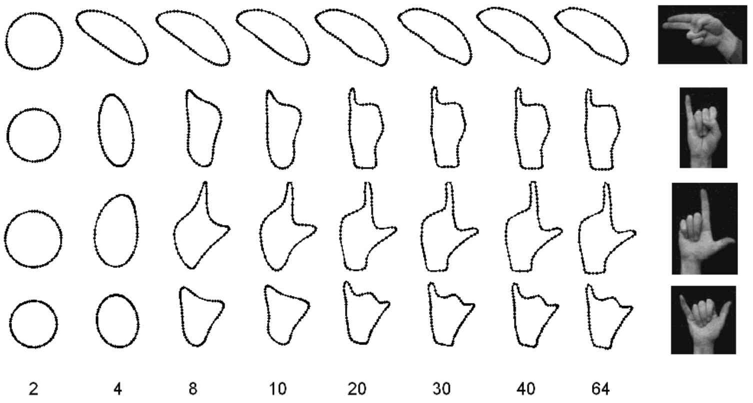


Figure 1.5 — Examples of reconstruction as a function of the cutoff frequency, with an initial contour sampled with $N = 64$ points

However, this set of invariants is not complete as it does not hold the phase information of the shape. The completeness of a set of invariant features (FD2) expresses the fact that two objects have the same shape if and only if they have the same set of features. A set of features which is complete but not stable is proposed in [35]. Stability means that a small distortion of the shape does not induce a noticeable divergence in the values of invariant features. The complete and stable set of invariant descriptors is defined by [43]:

$$I_{k_0}(\gamma) = |C_{k_0}(\gamma)|, \text{ for } k_0 \text{ such that } C_{k_0}(\gamma) \neq 0, \tag{1.19}$$

$$I_{k_1}(\gamma) = |C_{k_1}(\gamma)|, \text{ for } k_1 \neq k_0 \text{ such that } C_{k_1}(\gamma) \neq 0, \tag{1.20}$$

$$I_k(\gamma) = \frac{C_k(\gamma)^{k_0-k_1} C_{k_0}(\gamma)^{k_1-k} C_{k_1}(\gamma)^{k-k_0}}{I_{k_0}(\gamma)^{k_1-k-p} I_{k_1}(\gamma)^{k-k_0-q}} \tag{1.21}$$

with $p, q \in \mathbb{R}_+$ and $k1 \leq k0$.

For experiments, in order to simplify the expression of $I_k(\gamma)$, following [43], we take $k_0 = 2, k_1 = 1, p = q = 0.5$.

Notice that the cepstral descriptors can be investigated as used in speech recognition front-ends to enhance the robustness [45].

Figure 1.5 shows that the low frequency coefficients contain information on the general form of the shape and the high frequency coefficients contain information on the finer details of the shape. We can notice that with more than 20 coefficients the hand shape is well reconstructed.

• **legendre moments**: Any shape may theoretically be characterized by its set of regular moments. However, this kind of description is information redundant and prone to numerical instability. A better representation is obtained by using an orthogonal basis [102], such as Legendre polynomials. Assuming, without loss of generality, that the image domain is $[-1,1] \times [-1,1]$, the (p,q)-th order normalized Legendre moment is defined as:

$$\lambda_{p,q} = C_{pq} \iint_{\Omega_{ix}} P_p(x) P_q(y)\, \mathrm{d}x dy, \tag{1.22}$$

with normalizing constant: $C_{p,q} = (2p+1)(2q+1)/4$. The $p-th$ order Legendre polynomial is given by:

$$P_p(x) = \frac{1}{2^p p!} \frac{d^p}{dx^p} (x^2-1)^p \quad , \quad x \in [-1,1]. \tag{1.23}$$

Legendre polynomials generalize regular moments in the sense that the monomial $x^p y^q$ is replaced by an orthogonal polynomial $P_p(x)P_q(x)$ of the same order. Moreover, if we rewrite $P_p(x)$ as:

$$P_p(x) = \sum_{k=0}^{p} a_{pk} x^k, \tag{1.24}$$

then we come up with a simple relationship between Legendre moments and normalized central regular moments:

$$\lambda_{p,q} = C_{pq} \sum_{u=0}^{p} \sum_{v=0}^{p} a_{pu} a_{qv} \eta_{u,v}. \tag{1.25}$$

Any reference shape, discretized on a sufficiently fine grid, can be described by the vector of its central normalized Legendre moments up the order $N : \lambda_{p,q}^{ref}, p+q \leq N$.

This description inherits scale and translation invariance from normalized central moments. The invariance to rotation may be proved but it is not the purpose of this work. For the complexity of computation (order to ensure scale, translation and rotation invariance), this method can be considered more CPU consuming compared to other descriptors of global approaches like fourier descriptors.

1.3.2 Semi-local Approach

• **Histogram of Oriented Gradient (HOG) descriptors** are features widely used by the object detection and object recognition community. They have been shown to be distinctive and robust under small affine transformations and illumination changes. They are constructed by dividing the image into a dense grid of uniformly spaced cells and then computing the orientation histograms of the image gradient values on each cell. The illumination and contrast changes are taken into account by local normalization of the gradient strengths which requires grouping the cells together into larger, spatially-connected blocks.
The HOG descriptor is then the vector of the components of the normalized cell histograms for all the block regions. Dalal *et al.* [82] have proposed Histogram of Oriented

Gradients in the case of human detection. They have also been used for hand posture recognition [38] and gesture recognition [64].

1.3.3 Local Approach

• **The Scale Invariant Feature Transform (SIFT)** is a well known local descriptor created in 1999 by Lowe [72], allowing to detect and extract features which are invariant to rotation and scale and robust to some variations of illuminations, viewpoints and noise. The SIFT descriptor is computed in four steps. The two first stages correspond to the choice of keypoints, first identifying potential interest points that are scale and rotation invariant and then rejecting the ones that have low contrast and stability. The two last stages correspond to the descriptor vector computation, assigning one or more orientations to each selected keypoint based on local image gradient directions and using a 4*4 location Cartesian grid to compute the gradient on each location bin on the patch around the keypoint.
The SIFT descriptor gives good results in the case of object recognition when it can find relevant keypoints. It has been used by Wang *et al.* [107] for hand posture recognition with the objective of human-robot interaction.

• **Speeded Up Robust Feature (SURF)** was first presented by Bay *et al* in 2006 [10]. Partly inspired by the SIFT descriptor, SURF also consists in interesting points localization followed by feature descriptors computation. In both cases, the output is a representation of the neighborhood around an interest point as a descriptor vector. SURF is based on the distribution of first order Haar wavelet responses [49]. One of the principal advantages of SURF is to be several times faster than SIFT while having more discriminative power. It uses the integral images to simplify and to accelerate the computations. Yielding a lower dimensional feature descriptor, it reduces the time for feature computation and matching. In [39], a fast multi-scale feature detection, SURF-inspired, and a description method for hand gesture recognition is proposed.

1.3.4 Geometrical Approach

• **Varied Form Descriptor (Var)**. Full reconstruction of the hand is not essential for gesture recognition. Many approaches have instead used the extraction of low-level image measurements for that purpose [83]. Being fairly robust to noise, these characteristics can be extracted quickly. In this approach we created a geometry-based feature vector by gathering simple geometrical characteristics described hereunder:

$$Isometric\ rate = \frac{hand's\ perimeter^2}{hand's\ area\ \ \times 4 \times \pi} \tag{1.26}$$

$$Lengthening = \frac{radius\ of\ the\ biggest\ hand\ inscribed\ circle}{radius\ of\ the\ smallest\ hand\ circumscribed\ circle} \tag{1.27}$$

$$Concavity = \frac{perimeter\ of\ the\ hand's\ convex\ hull}{hand's\ perimeter} \tag{1.28}$$

$$Elongation = \frac{major\ axis\ of\ the\ hand's\ smallest\ elliptical\ hull}{minor\ axis\ of\ the\ hand's\ smallest\ elliptical\ hull} \tag{1.29}$$

1.3.5 Comparative Study

Collumeau *et al.* [33] assess that the geometrical approach Var and the geometry-based global approach Hu moments perform best (see table 1.1) but require a segmentation step prior to their computation. They are followed by keypoint-based local methods (SIFT, SURF) whose performance is slightly enhanced by the segmentation step. HOG proved to be especially dependant on the correct framing of the hand, performing poorly when facing a large background-enclosed hand but achieving second best recognition rate when the hand is well-framed. Although less improved than Hu moments by the segmentation step, HOG's performance nevertheless suffers from its lack. Zernike moments come last with the smallest recognition rate.

These results outline the worthiness of simple, geometrical descriptors for describing a single object, namely the user's hand, displayed in various configurations. Predominance of such descriptors conveying the hands shape will therefore focus future research on descriptors whose relevance have been established when dealing with shapes.

	'Gray-level hand and background'	**'Gray-level hand on black background'**	**'Binary object'**
ZER	**21.1**	**24.9**	**25.6**
HU	**19.7**	**52.5**	**68.1**
HOG	**33.2**	**44.3**	**38.2**
SIFT	**58.1**	**60.3**	**63.5**
SURF	**51.5**	**60.1**	**66.8**
VAR	**-**	**-**	**76.4**

Table 1.1 — Mean recognition rates obtained over the 4 speakers with images presenting palmar aspect [33]

Bourennane *et al.* [19] have shown that Fourier descriptors (FD1) outperforms Hu moments for all deformations (see table 1.2), they notice that Hu moment invariants and Zernike's moment invariants are calculated on the global image space. It has been shown that the values of Hu's moment invariants and Zernike's moment invariants are sensitive to noise [19].

The FD1 outperforms the other shape descriptors in terms of discrimination between visually close gestures. Either moment invariants or Fourier coefficients are computed from the segmented hand posture. When the postures lead to similar segmentation results, some details of the hand contour are smoothed, and both moment invariants and Fourier coefficients are affected.

		'HU'	'Zernicke'	'FD1'	'FD2'
Learning set	:	**38.9**	**81.5**	**81.5**	**80.3**
Test set	:	**37.1**	**74.9**	**77.8**	**77.0**
Cross-validation	:	**30.5**	**76.7**	**77.0**	**76.2**

Table 1.2 — Recognition rates (%) with Triesch database and Euclidean distance For FD1, 6 invariant features are used, and 4 for FD2 [19].

1.4 Hand posture classification

The classification represents the task of assigning a feature vector or a set of features to some predefined classes in order to recognize the hand gesture. In previous years several classification methods have been proposed and successfully tested in different recognition systems. In general, a class is defined as a set of reference features that were obtained during the training phase of the system or by manual feature extraction, using a set of training images. Therefore, the classification mainly consists of finding the best matching reference features for the features extracted in the previous phase. The classification consists in maximizing or minimizing a discriminant function $d_i(x)$ between a vector of measurements x and the N classes of gestures. For example, in the case of a function to be minimized, such as a distance, we look for the class C such that: $C = \underset{i \in [1,N]}{argmin}\ (d_i(x))$.

The classification is usually performed with a distance, or methods such as nearest neighbors. The number of images used for learning is an important factor for classification.

Chen *et al.* [31] use the FD and motion analysis to recognize dynamic gesture with Hidden Markov Models (HMM).
Wah Ng and Ranganath [84] use the FD and Radial-Basis Function (RBF) as classifier-type to recognize five postures. They then propose to recognize fourteen dynamic gestures, some of which are made with both hands, with HMM or neural networks.

The Adaboost classifier, short for Adaptive Boosting, is a machine learning algorithm, formulated by [41]. It is a meta-algorithm, and can be used in conjunction with many other learning algorithms to improve their performance. AdaBoost is adaptive in the sense that subsequent classifiers built are tweaked in favor of those instances misclassified by previous classifiers. AdaBoost is sensitive to noisy data and outliers. In some problems, however, it can be less susceptible to the overfitting problem than most learning algorithms.

Caplier *et al.* [27] use of Hu moments and a neural network "Multi-layer perceptron" to classify eight gestures made by three people.

The Euclidean distance is the "ordinary" distance between two points that one would measure with a ruler, and is given by the Pythagorean formula. By using this formula as distance, Euclidean space (or even any inner product space) becomes a metric space. The Euclidean distance between the measurement vector x and the class i is defined by:

$$d_{E,i}(\mathbf{x}) = \sqrt{(x-\mu_i)^T(x-\mu_i)} \tag{1.30}$$

with μ the mean vector of class i. This is the usual metric for calculating a distance between the invariants vectors I_k of contours γ_1 and γ_2.

$$d_E(\gamma_1,\gamma_2) = \sqrt{\sum_k |I_k(\gamma_1) - I_k(\gamma_2)|^2} \tag{1.31}$$

Bayesian classification is based on Bayes' theorem:

$$p(C_i|x) = \frac{p(x|C_i)p(C_i)}{p(x)} \tag{1.32}$$

with:

$p(C_i|x)$ the posterior probability of the class C_i knowing that the measurement vector is x,
$p(x|C_i)$ the conditional probability of x, knowing that the class C_i,
$p(C_i)$ the prior probability of the class C_i
$p(x)$ the conditional probability of measurement vector x

$$p(x) = \sum_{i=1}^{N} p(x|C_i)p(C_i) \tag{1.33}$$

In this case, the discriminator function is given by the maximum *a posteriori*:

$$d_i(x) = p(C_i|x) \tag{1.34}$$

the Bayes Theorem (Eq. 1.32) can be rewritten as follows [78]:

$$d_i(x) = d_{M,i}(x) + \log(\Lambda_i) \tag{1.35}$$

with $d_M(x)$ the Mahalanobis distance:

$$d_{M,i}(x) = (x - \mu_i)^T \Lambda_i^{-1} (x - \mu_i) \tag{1.36}$$

the Mahalanobis distance appears as an Euclidean distance weighted by the inverse of the covariance matrices for each class.

The K-nearest neighbors classification method uses the feature-vectors gathered in the training to find the K nearest neighbors in a n-dimensional space. The training mainly consists of the extraction of (possible well discriminable) features from training images, which are then stored for later classification. Due to the use of distance measuring such as the euclidian or manhattan distance, the algorithm performs relatively slowly in higher dimensional spaces or if there are many reference features. In [114], an approximate nearest neighbors classification was proposed, which provides a better performance.

Support vector machines (SVM) are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis. The basic SVM takes a set of input data and predicts, for each given input, which of two possible classes forms the output, making it a non-probabilistic binary linear classifier. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other.

An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

The SVM is based on kernels that allow optimal separation of points into sets. The solution is optimal in the sense that the margin between the hyperplane and vectors of each class of the learning data is maximum. Also, SVM solve the problem of non-linearly separable data by projecting the data into a space of higher dimension. This projection is done with a polynomial kernel, Gaussian or hyperbolic.

Bourennane *et al.* [19] prove that the results are significantly better when using the Bayesian classifier on the Triesch database (100% see Table 1.3). For their internal database, with Fourier descriptors (6 invariants), the recognition rates also increase, in comparison with Euclidean distance, and results are similar for the three classifiers with a small advantage for k-NN.

The Hidden Markov Model (HMM) classifiers belong to the class of trainable classifiers. An HMM represents a statistical model, in which the most probable matching gesture-class is determined for a given feature vector, based on the training data. In order to train the HMM, a Baum-Welch re-estimation algorithm, which adapts the internal states of the HMM according to some feedback concerning the accuracy, was

		'BAYES'	'SVM'	'K-NN'	'EUCL'
Triesch, test set	:	**100**	**89.1**	**93.3**	**77.8**
Internal database, learning set	:	**99.9**	**99.9**	**100**	**96.8**
Internal database, test set	:	**84.7**	**84.2**	**87.9**	**83.9**

Table 1.3 — Recognition rates (%) with Triesch database and FD1, 6 invariant features are used, and different classifiers: Bayesian classifier (BAYES), support vector machine (SVM), k-nearest neighbors (k-NN) and Euclidean distance (EUCL) [19].

used.

The Multi Layer Perceptron (MLP) classifier is based on a neural network. Therefore, MLPs represent a trainable classifier (similar to Hidden Markov Models). They use three or more layers of neurons that are all connected. During the training phase, the weights of the connections between the neurons are adapted, based on the feedback that describes the difference between the output and the expected result.

1.5 Conclusion of the chapter

In this chapter, we reviewed several existing methods for supporting vision-based human-computer interaction based on the recognition of hand gestures. The provided review covers research work related to all three individual subproblems of the full problem, namely detection, characterization and recognition or classification.

In the detection step we mentioned two types of detection: detection for static gestures and detection for dynamic gesture:

- The detection of postures (static gestures) is usually based on the color of the hand. This detection is very limited in the case where there is a background of the same color as the hand or if there are other objects in the scene which also have the same color. As it is known, to make a detection based on the color of the hand, we must have prior knowledge and it will be limited if we try to extend it to all users.

- Concerning dynamic gestures we have mentioned several methods as HMM, DTW or a method based on the skeleton. These methods give satisfactory results but sometimes have a large computational time or are limited to a small and although accurate dictionary of gestures.
  For these reasons, a new method of detection must be found, or we have to combine existing methods to overcome these limitations and make the detection, which is the major step in our process, very reliable.

The purpose of characterization, or features extraction, is to transform an image into a signature which characterizes a clearly defined a contour of posture and which permits to compare, in the next step of process, test postures with references postures stored and characterized in learning step. However, we have seen that the characterization needs to validate properties of invariance (rotation, translation, scale factor), and we must be able bijectively reconstruct the image from these signals. We mention many methods such as descriptors or geometric methods but also local or semi-local methods. As fast as possible, the main objective is to find and combine the methods that give the best results and faster and which also discriminates very close postures, thinking in this sense is highly essential.

Classification is an important step in our process, it is often based on the criterion of distance (Euclidian, Bayesian, KNN) or on geometric criteria (SVM), but it will be very difficult to implement if the feature vector or the characteristic matrix has many parameters, or if there's multiple classes. So our choice will be set according to the number of parameters that characterizes our gesture but also by the complicity of classifying and computational time. That's why we will perform the dimension reduction and decrease the number of classes. This seems to be a good strategy to use the easiest and fastest classifier.

CHAPTER 2 Hand database

2.1 Introduction of the chapter

GESTURES are an important modality for human-machine communication, and robust gesture recognition can be an important component of assistive environments and human-computer interfaces in general. A key problem in recognizing gestures is that the appearance of a gesture can vary widely depending on variables such as the person performing the gesture, or the position and orientation of the camera. For example, the same handshape can look very different in different images, depending on the 3D orientation of the hand and the viewpoint of the camera. Similarly, in the domain of sign language recognition, the appearance of a sign can vary depending on the person performing the sign and the distance from the camera. This database-based framework is applied to two different gesture recognition domains.

The first domain is handshape categorization. Handshapes can hold important information about the meaning of the gesture, for example in sign languages, or about the intent of an action, for example in manipulative gestures or in virtual reality interfaces. A large database of tens of thousands of images is used to represent the wide variability of handshape appearance. A key advantage of the database is that it provides a very natural way to characterize the appearance of each handshape class. Furthermore, databases containing tens or hundreds of thousands of images representing several people can ensure a learning more consistent to the reality.

The second gesture recognition domain where we apply the proposed approach is recognition of signs in American Sign Language (ASL).

2.2 Various hand databases

According to the literature, and best of our knowledge, there are a few publicly available gesture image databases. Athitsos and Sclaroff [7] published a database for hands posed in different gestures. The database contains more than 107000 images. Despite the fact that the database covers 26 gestures and has ground truth tables, the images actually present only the edges of the hands. Tests for algorithms that are not based on edges are not feasible. Athitsos also contributed to the creation of an American

Sign Language (ASL) video sequence database. These videos present the upper body part of a person signaling short texts in ASL. The videos were recorded at a rate of 60 frames per second. Some frames present the hands in a small scale and they are sometimes blurred. It is also difficult to cluster sets of hands where the gesture is of a certain type. There are images from 4 different cameras. This database would be suitable for testing detection algorithms, but it would be difficult to use those images for training.

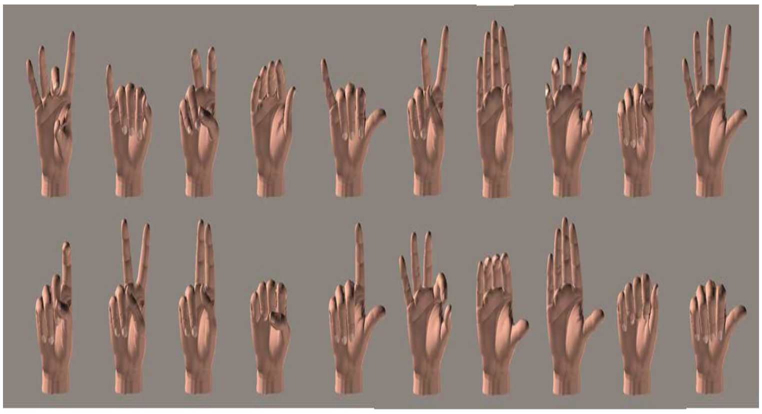


Figure 2.1 — Exemple of ASL postures

In handshape recognition for ASL database, there are 20 postures to recognize as shown on Fig. 2.1. For the evaluation of hand tracking methods in sign language recognition systems a database has been prepared. The RWTH-BOSTON-Hands database is a subset of the RWTH-BOSTON-104 videos with additional annotation of the signer's hand positions. The positions of both hands have been annotated manually in 15 videos. 1119 frames in total are annotated.

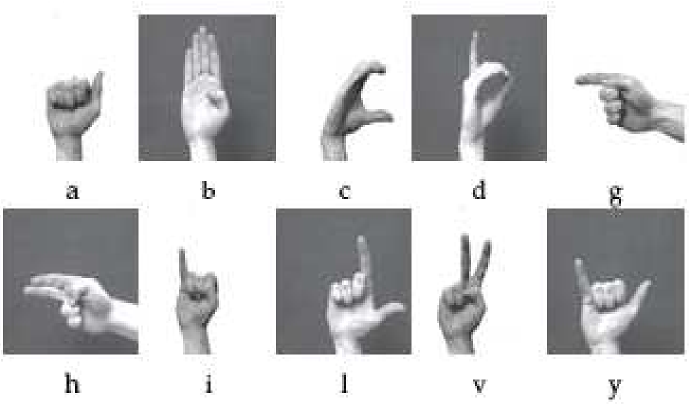


Figure 2.2 — The gestures base of Triesch and von der Malsburg [103]

There are also some other databases that are not specifically related to gesture but are particularly related to the subject.

The gestures base of Triesch and von der Malsburg [103] is a base of reference used in several studies, and made available on Internet. It contains 10 hand postures (Fig. 2.2), realized by 24 people and in front of different backgrounds (white, black and complex). Pictures are in gray level, PGM format, and in 128 × 128 size.

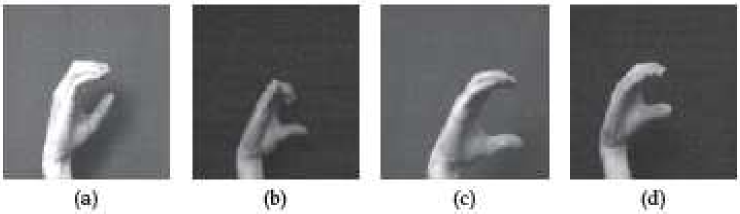


Figure 2.3 — Exemple of image with gestures "c" in Triesch base[103]

We can use sets of pictures with black and white backgrounds, but it's always a white hand. The variation of the form of the gestures in terms of size, translation and rotation is very limited. However, the form of the hand of different users can be very variable (see figure 2.3).

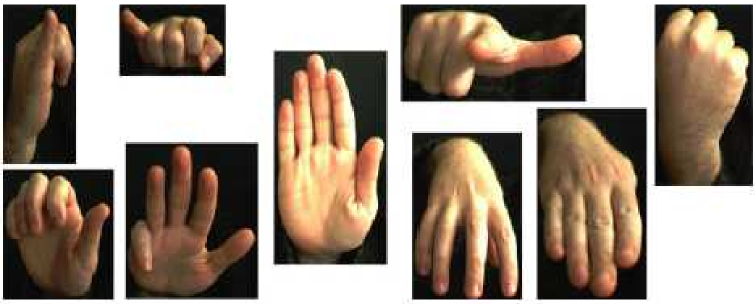


Figure 2.4 — Some images of the samples from the Massey Hand Gesture Database

The Massey Hand Gesture Database is an image database containing a number of hand gesture and hand posture images. The database has been developed by the authors to evaluate their methods and algorithms for real-time gesture and posture recognition. It is posted on the web with the hope of assisting other researchers investigating in the related domains. At this stage, the Database includes about 1500 images of different hand postures, in different lighting conditions. The data was collected by a digital camera mounted on a tripod from a hand gesture in front of a dark background, and in different lighting environments, including normal light and dark room with artificial light. Together with the original images there is a clipped version of each set of images that contains only the hand image. The maximum resolution of the images is 640 x 480 with 24 bit RGB color. So far, the database contains material gathered from 5 different individuals.

2.3 Proposed hand database

As new hand detection and gesture recognition algorithms are being developed, the use of features such as color, size, and shape of the favorite object are more likely to be used. Currently available databases are either for special purposes, or suffer from the lack of the desired features (e.g. not being in color or exhibiting a very small size of the samples). Previous works show that color is one of the important features in body tracking [31, 33, 34, 39, 113]. Color can be found to be invariant to changes in size, orientation and sometimes occlusion. In addition, according to Moore's law, every 18 month the processing speed and available memory size of processors double. So, possibly in the near future, using samples with higher details would be preferred by researchers.

Different gesture bases have several limitations: the number of images is small, the angle of view, the size and orientation of the hand is always the same, the images are grayscale and contain solely a hand without any other object in the background. Or even the database is not accessible. The common point of these bases is the use of white hands only. Our goal was to create a man-machine interface that applies everywhere (non-uniform background), for any kind of hand (adult or child, male or female, white or color).
This requires a much better developed database, but also a database where you can combine static and dynamic recognition, with simple but various postures, which are easily achievable by any user. Thus, to achieve a more realistic test base which could be the closest to our HMI configuration, we established our own database.

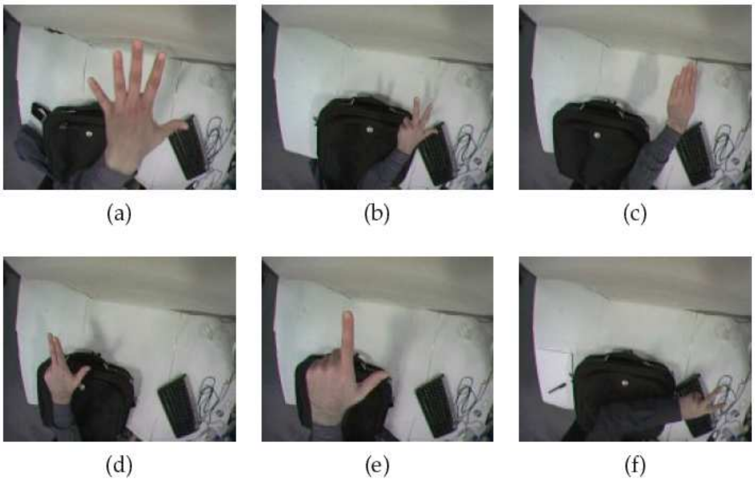


Figure 2.5 — Examples of images in Simon Conseil's database

At first it was decided to use the database from Simon Conseil [34], although it is limited to white hands (see figure 2.5), but it allows, first, to test the effectiveness of our characterization method before expanding them to other types of hands. Also, with this database we can compare the different methods of shape description with our method of characterization and deduce the contribution and performance as well as the limitations of our signature. the most important in the choice of database is to choose postures that are adapted to industrial applications, but also simple, practical and realizable by any user.

This base is inspired by the 8 postures of Cued Speech presented by Caplier *et al* [27]. The Cued Speech is a language which differs from the language of signs, and which aims at facilitating lip reading for deaf and hard of hearing (see figure 2.6).

However, postures "5" and "8" have been added to Cued Speech database to assess the performance of the methods we propose.

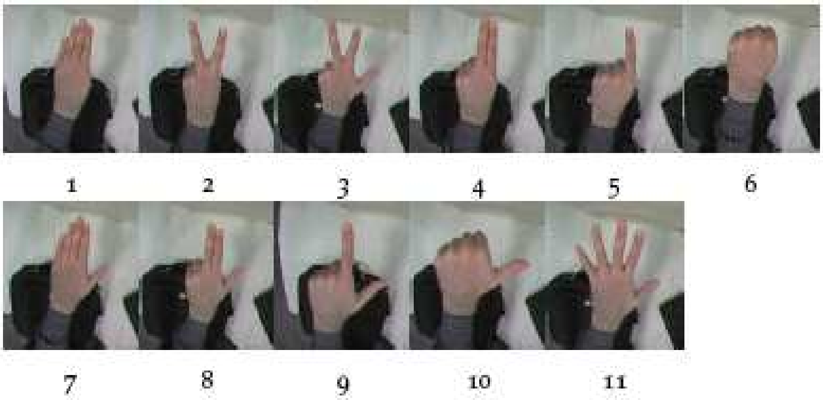


Figure 2.6 — The 11 postures of our database

This database is available within GSM group,where is performed by this thesis, and was built with a monoscopic video acquisition system. The video sequences were then split into images, to be processed separately. This database is composed of 11 postures performed by 18 persons (1000images/personne/posture) which represents roughly 200,000 images.

Once of the relevance of this database is validated, it can be extended to colored hands just by introducing new image in the learning base. The hand contours characterization is performed out of a binary image which includes only the contours. So the generalization and extension of the algorithm to a database including colored hands will mainly the first images preprocessing steps.

2.4 Conclusion of the chapter

One of the typical applications for an image database is to use it as a training set for learning algorithms. The same database could also be used for the testing phase, but it is more convenient to perform tests with real images acquired separately from a different person.

In this chapter we are interested in various existing databases, that are either intended for recognition of hands or for the language of signs where different gestures are used. it appears that these databases are limited by their format, the color of the hand, or even the number of stored images. So we decided to create our own database with postures which are easy to produce by all users, which will be for us a universal database without forgeting to compare our method with references databases as the Triech database.

Part II

Recognition process and results

CHAPTER 3 Array processing models and methods adapted to contour detection

3.1 Introduction of the chapter

CONTOUR detection is an important step in image processing. After a low-level processing such as denoising, it permits to enhance fitting lines, and the interest is to delimitate structures of interest such as roads, buildings, vehicles, etc. A large amount of methods have been proposed to characterize either parametrized or free-form contours. The most common method is still the derivative approach with linear filtering. Many derivative filters have been studied and used to compute the intensity gradient of gray-level images: Roberts, Sobel, Prewitt or Canny operators [26]. Other approaches have followed, such as mathematical morphology, Markov random fields, surface models, histogram automatic threshold [86].
General contours are called free-form. Detecting them is the purpose for instance of snakes [65] which have been improved in various ways such as Gradient Vector Flow [110, 111]. This type of method makes a single contour evolve while ensuring an attach to the image gradient, but also a control of the properties of the snake such as elasticity. Free-form contour detection is also the purpose of levelset [8, 29, 58]. Levelsets exhibit the advantage of retrieving multiple contours, in particular blurred contours for some specific version [29]. It is however well-known that an elevated number of parameters must be tuned and that they rely on an optimization strategy which is sensitive to initialization.

Very simple contours which are therefore encountered in many applications can be characterized by a few parameters: straight lines with orientation and offset, or circles with center coordinates and radius. The Hough transform for instance [37, 51, 67] was proposed under different versions, to retrieve straight lines. The generalized Hough transform (GHT) provides an estimation of the circle center coordinates when their radius is known [9, 57]. But in this chapter we concentrate on original methods for the detection of linear-like or circular-like contours. These methods rely on the array processing paradigm. This chapter is logically divided into four parts, starting from the first issue of this framework, the estimation of straight lines with a linear antenna, proposed in [6] in the early nineties, and concluding with the estimation of highly distorted star-shaped contours [61], which inspired the method for the characterization of hand contours exposed further in this manuscript. In between, we also present the estimation of circular contours and blurred contours.

3.2 Straight contour retrieval

3.2.1 Data model, generation of the signals out of the image data

To adapt array processing techniques to distorted curve retrieval, the image content must be transcripted into a signal. This transcription is enabled by adequate conventions for the representation of the image, and by a signal generation scheme[2, 5]. Once a signal has been created, array processing methods can be used to retrieve the characteristics of any straight line. Let I be the recorded image (see Fig. 3.1(a)).

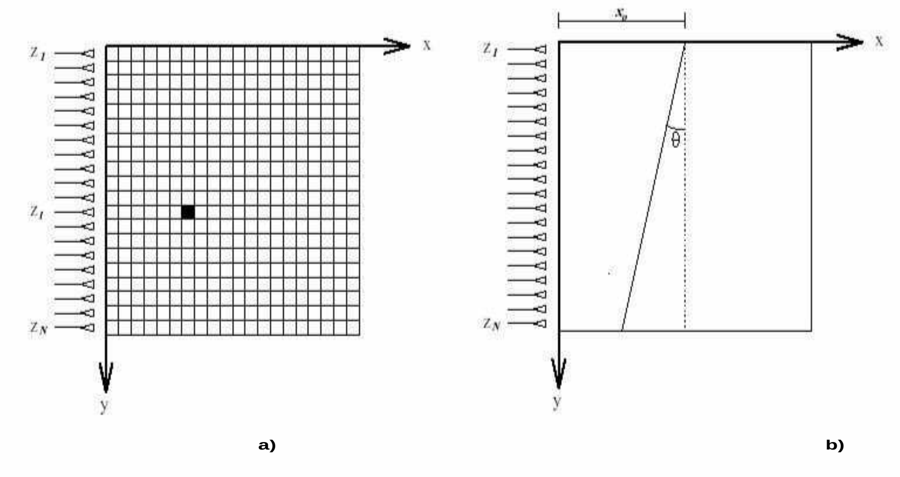


Figure 3.1 — The image model (see [5]): (a) The image-matrix provided with the coordinate system and the rectilinear array of N equidistant sensors, (b) A straight line characterized by its angle θ and its offset x_0.

We consider that I contains d straight lines and an additive uniformly distributed noise. The image-matrix is the discrete version of the recorded image, compound of a set of $N * C$ pixel values. A formalism adopted in [6] allows signal generation, by the following computation:

$$z(i) = \sum_{k=1}^{C} I(i,k)exp(-j\mu k), \ i = 1, \ldots, N \tag{3.1}$$

where $\{I(i,k);\ i \in\ \{1,\ldots,N\};\ k \in\ \{1,\ldots,C\}\}$ denote the image pixels. Eq. (3.1) simulates a linear antenna: each row of the image yields one signal component as if it were associated with a sensor. The set of sensors corresponding to all rows forms a linear antenna. We focus in the following on the case where a binary image is considered. The contours are composed of 1-valued pixels also called "edge pixels", whereas 0-valued pixels compose the background. When d straight lines, with parameters angle $\{\theta_k\}$ and offset x_{0k} $(k = 1,\ldots,d)$, are crossing the image, and if the image contains noisy outlier pixels, the signal generated on the i^{th} sensor, in front of the i^{th} row, is [6]:

$$z(i) = \sum_{k=1}^{d} exp(j\mu(i-1)tan(\theta_k))exp(-j\mu x_{0k}) + n(i) \tag{3.2}$$

where μ is a propagation parameter [3] and $n(i)$ is due to noisy pixels on the i^{th} row. Defining: $a_i(\theta_k) = exp(j\mu(i-1)tan(\theta_k)),\ s_k = exp(-j\mu x_{0k})$, Eq. (3.2) becomes:

$$z(i) = \sum_{k=1}^{d} a_i(\theta_k)s_k + n(i),\ \ i = 1,\cdots,N \tag{3.3}$$

Grouping all terms in a single vector, Eq. (3.3) becomes: $\mathbf{z} = \mathbf{A}(\theta)\mathbf{s} + \mathbf{n}$, with $\mathbf{A}(\theta) = [\mathbf{a}(\theta_1),\cdots,\mathbf{a}(\theta_d)]$ where $\mathbf{a}(\theta_k) = [a_1(\theta_k), a_2(\theta_k),\cdots,a_N(\theta_k)]^T$, with $a_i(\theta_k) = exp(j\mu(i-1)tan(\theta_k)),\ i = 1,\ldots,N$, superscript ^{T} denoting transpose. SLIDE (Subspace-based LIne DEtection) algorithm [6] uses TLS-ESPRIT (Total-Least-Squares Estimation of Signal Parameters via Rotational Invariance Techniques) method to estimate the angle values. To estimate the offset values, the "extension of the Hough transform" [67] can be used. It is limited by its high computational cost and the large required size for the memory bin. [20, 22] developed another method. This method remains in the frame of array processing and reduces the computational cost: A high-resolution method called MFBLP (Modified Forward Backward Linear Prediction) [20] is associated with a specific signal generation method, namely the variable parameter propagation scheme [3]. The formalism introduced in that section can also handle the case of straight edge detection in gray-scale images [4].

3.2.2 Angle estimation, overview of the SLIDE method

The method for angles estimation falls into two parts: the estimation of a covariance matrix and the application of a total least squares criterion.

Numerous works have been developed in the frame of the research of a reliable estimator of the covariance matrix when the duration of the signal is very short or the number of realizations is small. This situation is often encountered, for instance, with seismic signals. To cope with it, numerous frequency and/or spatial means are computed to replace the temporal mean. In this study the covariance matrix is estimated by using the spatial mean [46]. From the observation vector we build K vectors of length M with $d < M \leq N - d + 1$. In order to maximize the number of sub-vectors we choose $K = N + 1 - M$. By grouping the whole sub-vectors obtained in matrix form, we obtain: $\mathbf{Z}_K = [\mathbf{z}_1, \cdots, \mathbf{z}_K]$, where $\mathbf{z}_l = \mathbf{A}_M(\theta)\mathbf{s}_l + \mathbf{n}_l, \quad l = 1, \cdots, K$. Matrix $\mathbf{A}_M(\theta) = [\mathbf{a}_M(\theta_1), \cdots, \mathbf{a}_M(\theta_d)]$ is a Vandermonde type one of size $M \times d$. Signal part of the data is supposed to be independent from the noise; the components of noise vector $\mathbf{n}_l$ are supposed to be uncorrelated, and to have identical variance. The covariance matrix can be estimated from the observation sub-vectors as it is performed in [5]. The eigen-decomposition of the covariance matrix is, in general, used to characterize the sources by subspace techniques in array processing. In the frame of image processing the aim is to estimate the angle θ of the d straight lines. Several high-resolution methods that solve this problem have been proposed [92]. SLIDE algorithm is applied to a particular case of an array consisting of two identical sub-arrays [4]. It leads to the following estimated angles [4]:

$$\hat{\theta}_k = \tan^{-1}[\frac{1}{(\mu * \Delta)} Im(\ln(\frac{\lambda_k}{|\lambda_k|}))], \tag{3.4}$$

where $\{\lambda_k,\ k = 1, \ldots, d\}$ are the eigenvalues of a diagonal unitary matrix that relates the measurements from the first sub-array to the measurements resulting from the second sub-array. Parameter μ is the propagation constant, and Δ is the distance between two sensors. TLS-ESPRIT method used by SLIDE provides the estimated parameters in closed-form, in opposite to the Hough transform which relies on maxima research [67]. Offset estimation exploits the estimated straight lines angles.

3.2.3 Offset estimation

The most well-known offset estimation method is the "Extension of the Hough Transform" [96]. Its principle is to count all pixel aligned on several orientations. The expected offset values correspond to the maximum pixel number, for each orientation value. The second proposed method remains in the frame of array processing: it employs a variable parameter propagation scheme [2, 3, 4] and uses a high resolution method. This high resolution "MFBLP" method relies on the concept of forward and backward organization of the data [46, 90, 104]. A variable speed propagation scheme [3, 4], associated with "MFBLP" (Modified Forward Backward Linear Prediction) yields offset values with a lower computational load than the Extension of the Hough Transform. The basic idea in this method is to associate a propagation speed which is different for each line in the image [4]. By setting artificially a propagation speed that linearly depends on row indices, we get a linear phase signal. When the first orientation value is considered, the signal received on sensor i ($i = 1, \cdots, N$) is then:

$$z(i) = \sum_{k=1}^{d_1} exp(-j\tau x_{0k}) exp(j\tau(i-1)tan(\theta_1)) + n(i) \tag{3.5}$$

d_1 is the number of lines with angle θ_1. When τ varies linearly as a function of the line index the signal vector $\mathbf{z}$ contains a modulated frequency term. Indeed we set $\tau = \alpha(i-1)$.

$$z(i) =$$

$$\sum_{k=1}^{d_1} exp(-j\alpha(i-1)x_{0k})exp(j\alpha(i-1)^2 tan(\theta_1)) + n(i) \tag{3.6}$$

This is a sum of d_1 signals that have a common quadratic phase term but different linear phase terms. The first processing consists in obtaining an expression containing only linear terms. This goal is reached by dividing $z(i)$ by the non zero term $a_i(\theta_1) = exp(j\alpha(i-1)^2 tan(\theta_1))$. We obtain then:

$$w(i) = \sum_{k=1}^{d_1} exp(-j\alpha(i-1)x_{0k}) + n^{'}(i), \tag{3.7}$$

The resulting signal appears as a combination of d_1 sinusoids with frequencies :

$$f_k = \frac{\alpha x_{0k}}{2\pi}, \ \ k = 1, \cdots, d_1. \tag{3.8}$$

Consequently, the estimation of the offsets can be transposed to a frequency estimation problem. Estimation of frequencies from sources having the same amplitude was considered in [104]. In the following a high resolution algorithm, initially introduced in spectral analysis, is proposed for the estimation the offsets.

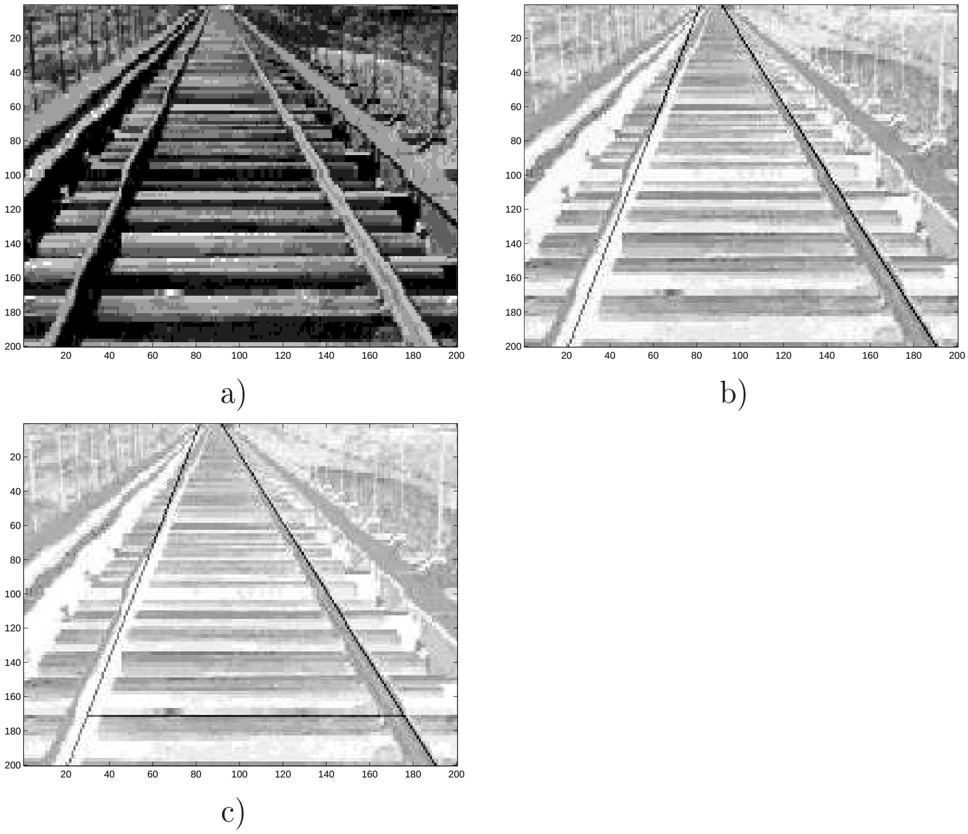
After adopting our signal model we adapt to it the spectral analysis method called modified forward backward linear prediction (MFBLP) [104] for estimating the offsets: We consider d_k straight lines with given angle θ_k, and apply the MFBLP method. We consider d_k straight lines with given angle θ_k, and apply the MFBLP method, to the vector $\mathbf{w}$.

An outline of the method is as follows: 1) For a N-data vector $\mathbf{w}$, form matrix $\mathbf{Q}$ of size $2*(N-L) \times L$, where $1 \leq L \leq N-1$. The j^{th} column $\mathbf{q}_j$ of $\mathbf{Q}$ is defined by:
$\mathbf{q_j} = [w(L-j+1), ..., w(N-j), w^*(j+1), ..., w^*(N-L+j)]^T$.
Then build a length $2*(N-L)$ vector:
$\mathbf{h} = [w(L+1), ..., w(N), w^*(1), ..., w^*(N-L)]^T$. Calculate the singular value decomposition of $\mathbf{Q}$: $\mathbf{Q} = \mathbf{U\Lambda V}^H$.
2) Form a matrix $\mathbf{\Sigma}$, setting to 0 the $L-1$ smallest singular values contained in $\mathbf{\Lambda}$.
3) Form vector $\mathbf{g}$ from the following matrix computation: $\mathbf{g} = [g_1, g_2, ..., g_L]^T = -\mathbf{V} * \mathbf{\Sigma}^\sharp * \mathbf{U}^H * \mathbf{h}$ where $\mathbf{\Sigma}^\sharp$ is the pseudo-inverse of $\mathbf{\Sigma}$.
4) Determine the roots of polynomial function H, where $H(\gamma) = 1 + g_1\gamma^{-1} + g_2\gamma^{-2} + ... + g_L\gamma^{-L}$.
5) One zero of H is located on the unit circle. The complex argument of this zero is the frequency value; according to Eq. (3.5) this frequency value is proportional to the radius, the proportionality coefficient being $-\alpha$.

More details about MFBLP method applied to offset estimation are available in [22]. MFBLP estimates the values of f_k, $k = 1, \cdots, d_k$. According to Eq. (3.8) these frequency values are proportional to the offset values, the proportionality coefficient being $-\alpha$. The main advantage of this method comes from its low computational load. Indeed the complexity of the variable parameter propagation scheme associated with MFBLP is much less than the complexity of the Extension of the Hough Transform as soon as the number of non zero pixels in the image increases. This algorithm enables the characterization of straight lines with same angle and different offset.

3.2.4 Exemplification of the straight line retrieval methods

We propose an application of our method in the case of robotic vision. Fig. 3.2 is a photography taken by a camera and transmitted to the automatic command of a vehicle moving on the railway. This vehicle is used in particular for servicing of railways, *i.e.* for the replacement of the parallel crosspieces. The vehicle, when moving along the railway, determines first the position of the rails from the obtained picture. Then, the position of the nearest crosspiece is detected.



a)

b)

c)

Figure 3.2 — (a) - Image transmitted to the automatic command of a vehicle that is moving on a railway for the servicing of the railways. (b) Detection of the rails for the progress of the vehicle. (c) Localization of the first crosspiece that the vehicle has to replace. The process is iterated crosspiece after crosspiece: photography, detection of the rails and detection of the next crosspiece.

3.3 Retrieval of circular contours

Signal generation upon a linear antenna yields a linear phase signal when a straight line is present in the image. While expecting circular contours, we associate a circular antenna with the processed image. By adapting the antenna shape to the shape of the expected contour, we aim at generating linear phase signals.

3.3.1 Problem setting and virtual signal generation

Our purpose is to estimate the radius of a circle, and the distortions between a closed contour and a circle that fits this contour. We propose to employ a circular antenna that permits a particular signal generation and yields a linear phase signal out of an image containing a quarter of circle. In this section, center coordinates are supposed to be known, we focus on radius estimation, center coordinate estimation is explained further. Fig. 3.3(a) presents a binary digital image I. The object is close to a circle with radius value r and center coordinates (l_c, m_c). Fig. 3.3(b) shows a sub-image extracted from the original image, such that its top left corner is the center of the circle. We associate this sub-image with a set of polar coordinates (ρ, θ), such that each pixel of the expected contour in the sub-image is characterized by the coordinates $(r + \Delta\rho, \theta)$, where $\Delta\rho$ is the shift between the pixel of the contour and the pixel of the circle that roughly approximates the contour and which has same coordinate θ. We seek for star-shaped contours, that is, contours that can be described by the relation: $\rho = f(\theta)$ where f is any function that maps $[0, 2\pi]$ to $\mathbb{R}_+$. The point with coordinate $\rho = 0$ corresponds then to the center of gravity of the contour.

Generalized Hough transform estimates the radius of concentric circles when their center is known. Its basic principle is to count the number of pixels that are located on a circle for all possible radius values. The estimated radius values corresponds to the maximum number of pixels.

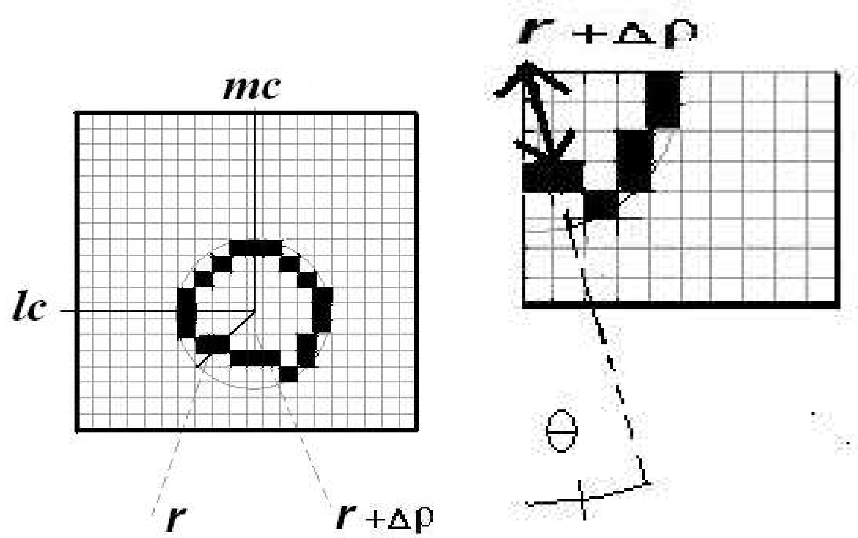


Figure 3.3 — (a) Circular-like contour, (b) Bottom right quarter of the contour and pixel coordinates in the polar system (ρ, θ) having its origin on the center of the circle. r is the radius of the circle. $\Delta\rho$ is the value of the shift between a pixel of the contour and the pixel of the circle having same coordinate θ.

Contours which are approximately circular are supposed to be made of more than one pixel per row for some of the rows and more than one pixel per column for some columns (see Fig. 3.3a)). Therefore, we propose to associate a circular antenna with the image which leads to linear phase signals, when a circle is expected. The basic idea is to obtain a linear phase signal from an image containing a quarter of circle (such as in Fig. 3.3b)). To achieve this, we use a circular antenna. The phase of the signals which are virtually generated on the antenna is constant or varies linearly as a function of the sensor index. A quarter of circle with radius r and a circular antenna are represented on Fig. 3.4.

The antenna is a quarter of circle centered on the top left corner, and crossing the bottom right corner of the sub-image. Such an antenna is adapted to the sub-images containing each quarter of the expected contour (see Fig. 3.4). In practice, the extracted sub-image is possibly rotated so that its top left corner is the estimated center. The antenna has radius R_a so that $R_a = \sqrt{2}N_s$ where N_s is the number of rows or columns in the sub-image. When we consider the sub-image which includes the right bottom part of the expected contour, the following relation holds: $N_s = max(N - l_c, N - m_c)$ where l_c and m_c are the vertical and horizontal coordinates of the center of the expected contour in a cartesian set centered on the top left corner of the whole processed image (see Fig. 3.3). Coordinates l_c and m_c are estimated by the method proposed in [2], or the one that is detailed later in this chapter.
Signal generation scheme upon a circular antenna is the following: the directions adopted for signal generation are from the top left corner of the sub-image to the corresponding sensor. The antenna is composed of S sensors, so there are S signal components.

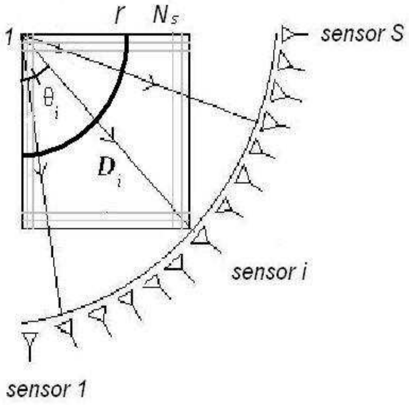


Figure 3.4 — Sub-image, associated with a circular array composed of S sensors

Let us consider D_i, the line that makes an angle θ_i with the vertical axis and crosses the top left corner of the sub-image. The i^{th} component ($i = 1, \ldots, S$) of the signal $\mathbf{z}$ generated out of the image reads:

$$z(i) = \sum_{\substack{l,m=1 \\ (l,m)\in D_i}}^{l,m=N_s} I(l,m) exp(-j\mu\sqrt{l^2+m^2}), \tag{3.9}$$

The integer l (resp. m) indexes the lines (resp. the columns) of the image. j stands for $\sqrt{-1}$. μ is the propagation parameter [4]. Each sensor indexed by i is associated with a line D_i having an orientation $\theta_i = \frac{(i-1)\cdot\pi/2}{S}$.

In Eq. (3.9), the term $(l, m) \in D_i$ means that only the image pixels that belong to D_i are considered for the generation of the i^{th} signal component. Satisfying the constraint $(l, m) \in D_i$, that is, choosing the pixels that belong to the line with orientation θ_i, is done in two steps: let *setl* be the set of indexes along the vertical axis, and *setm* the set of indexes along the horizontal axis.

If $\theta_i \leq \pi/4$, $setl = [1 : N_s]$ and $setm = \lfloor [1 : N_s] \cdot \tan(\theta_i) \rfloor$.

If $\theta_i \geq \pi/4$, $setm = [1 : N_s]$ and $setl = \lfloor [1 : N_s] \cdot \tan(\pi/2 - \theta_i) \rfloor$.

Symbol $\lfloor \cdot \rfloor$ means integer part.

The minimum number of sensors that permits a perfect characterization of any possibly distorted contour is the number of pixels that would be virtually aligned on a circle quarter having radius $\sqrt{2}N_s$. Therefore, the minimum number S of sensors is $\sqrt{2}N_s$.

3.3.2 Proposed method for radius estimation

In the most general case there exists more than one circle for one center. We show how several possibly close radius values can be estimated with a high-resolution method. For this, we use a variable speed propagation scheme towards the circular antenna. We propose a method for the estimation of the number d of concentric circles, and the determination of each radius value. For this purpose we employ a variable speed propagation scheme [4]. We set $\mu = \alpha(i - 1)$, for each sensor indexed by $i = 1, \ldots, S$. From Eq. (3.9), the signal received on each sensor is:

$$z(i) = \sum_{k=1}^{d} exp(-j\alpha(i-1)r_k) + n(i), \;\; i = 1, \ldots, S \tag{3.10}$$

where $r_k, k = 1, \ldots, d$ are the values of the radius of each circle, and $n(i)$ is a noise term that can appear because of the presence of outliers. All components $z(i)$ compose the observation vector $\mathbf{z}$. TLS-ESPRIT method is applied to estimate $r_k, k = 1, \ldots, d$, the number of concentric circles d is estimated by MDL criterion. The estimated radius values are obtained with TLS-ESPRIT method, which also estimated straight line orientations (see section 3.2.2). A further section is dedicated to the estimation of one-pixel wide nearly circular distorted contours. Let us now concentrate on 'blurred' contours, that is, contours which are composed of more than one pixel.

3.3.3 Linear antenna for the estimation of circle center parameters

Usually, an image contains several circles which are possibly not concentric and have different radii (see Fig. 3.5). To apply the proposed method, the center coordinates for each feature are required. To estimate these coordinates, we generate a signal with constant propagation parameter upon the image left and top sides. The l^{th} signal component, generated from the l^{th} row, reads: $z_{lin}(l) = \sum_{m=1}^{N} I(l,m)exp(-j\mu m)$, where μ is the propagation parameter. The non-zero sections of the signals, as seen at the left and top sides of the image, indicate the presence of features. Each non-zero section width in the left (respectively the top) side signal gives the height (respectively the width) of the corresponding expected feature. The middle of each non-zero section in the left (respectively the top) side signal yields the value of the center l_c (respectively m_c) coordinate of each feature.

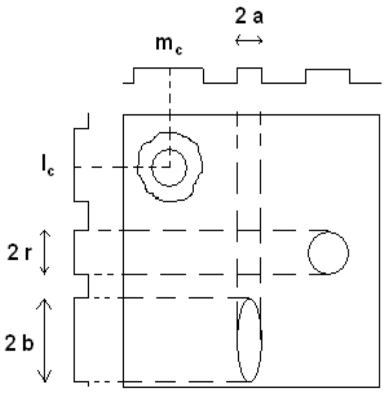


Figure 3.5 — Nearly circular or elliptic features. r is the circle radius, a and b are the axial parameters of the ellipse.

3.3.4 Exemplification of the circle characterization method

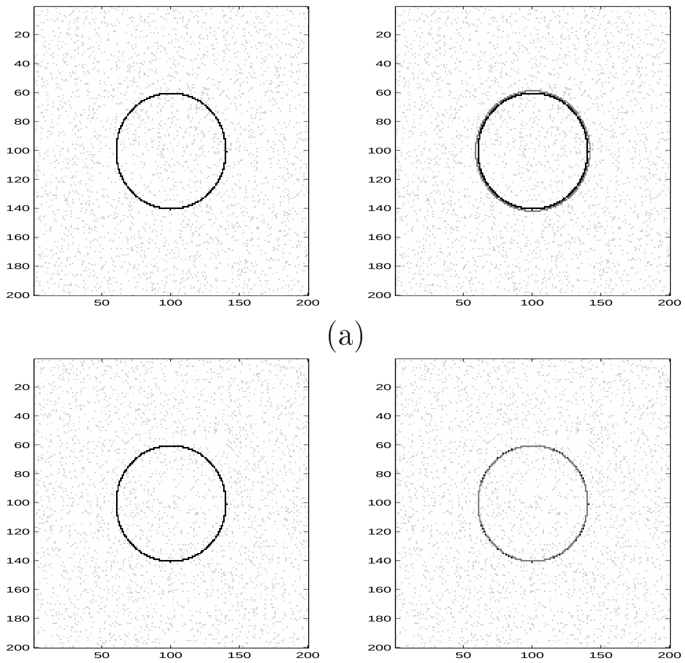
In Fig. 3.6, we exemplify the proposed method and the Hough tranform [67] on the same type of hand-made image containing a single circle. In both cases, the image is impaired with an additive Gaussian noise, with mean 0.02 and standard deviation 0.009, on 20% of the pixels.

The error on the radius value is 0.1 for the proposed method and 0.05 on the Hough transform. In both cases, this error is less than 1 pixel.

3.4 Blurred contour retrieval

3.4.1 Problem statement

In this subsection, we provide the models that we adopt for the image and the blurred contours therein. We remind a specific technique to generate a signal out of the image. Let $I(l,m)$ be an $N \times N$ recorded image (see Fig. 3.7(a) or Fig. 3.7(b)). We assume



(a)

(b)

Figure 3.6 — One circle: radius estimation by the proposed method and GHT: (a) Processed image and result with our method, (b) Processed image and result with GHT.

that $I(l,m)$ is compound of either several blurred linear contours or one blurred circular contour, and an additive uniformly distributed noise, whose gray level values follow a Gaussian distribution. A linear-like contour is supposed to have main orientation θ. We define its center offset x_0 as the distance between the top left corner of the image and the pixel with maximum gray level value $I_{\max}$ in the first row (see Fig. 3.7(a)). The spread of the contour is characterized by the parameter σ, and we define the parameter G such that $I_{\max} = \frac{G}{\sqrt{2\pi}\sigma}$. The value of G depends on the number of bits which are used to encode the image. When d blurred linear contours are present, they are defined by the set of parameters
$\{\theta_k,\ x_{0k},\ \sigma_k,\ k = 1,\ldots,d\}$. A circular-like contour is supposed to have center coordinates $\{l_c; m_c\}$. The pixels with value $\frac{G}{\sqrt{2\pi}\sigma}$ compound a circle with center coordinates $\{l_c; m_c\}$ and radius r_0. In both cases the gray level values of the pixels decrease gradually aside the set of pixels with value $\frac{G}{\sqrt{2\pi}\sigma}$. Blurred linear contours have width $2X_f$. A circular-like contour has width $2r_f$.

To set the link between image data representation and sensor array processing methods, an array of sensors is associated with the image [6, 76], as previously explained in this manuscript. Fig. 3.8 represents the linear and circular arrays associated with an image containing a blurred contour. The shape of the array is adequately chosen, considering the shape of the expected contour. To retrieve linear-like contours, the array sensors are supposed to be placed in front of each row (or each column) of the image [6] (see Fig. 3.8(a)). To retrieve circular-like contours, the array sensors are supposed to be placed along a quarter of circle centered on the center point of the

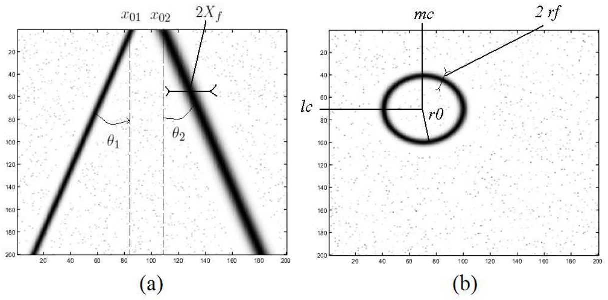


Figure 3.7 — Contour models: (a) blurred contours characterized by main orientations θ_1, θ_2, offsets x_{01} and x_{02}, and width $2X_f$; (b) blurred circular contour characterized by center coordinates $\{l_c; m_c\}$, radius r_0, and width $2r_f$.

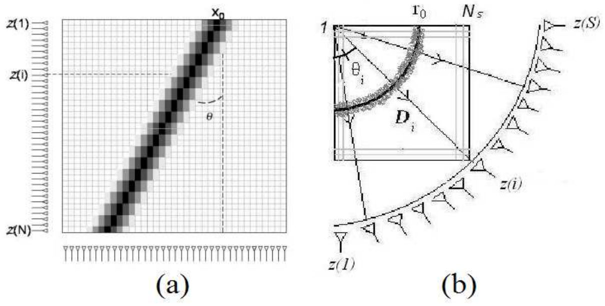


Figure 3.8 — Signal generation: (a) linear antenna for the generation of signal components $z(1)$, $z(2), \ldots,$ $z(N)$ on left and bottom sides, blurred linear-like contour with orientation θ and offset x_0; (b) sub-image of size $N_S \times N_S$ circular antenna [76] for the generation of signal components $z(1)$, $z(2), \ldots,$ $z(S)$ with i^{th} sensor at angular position θ_i and associated direction of generation D_i, blurred quarter of circle

expected circle [76]. The intuition behind this choice is to adapt the antenna shape to the expected contour shape and get similar signal models. How to choose between linear or circular antenna is explained in subsection 3.4.2. Fig. 3.8(b) shows part of Fig. 3.7(b), which is selected to perform circle characterization. The top left corner of Fig. 3.8(b) coincides with the center of the blurred circular contour. The antenna is compound of S sensors, each one related to the angular position $\theta_i = \frac{(i-1)\cdot\pi/2}{S}$, and to the direction of generation D_i.
In the case where linear-like contours are expected, we adopt the signal generation scheme proposed in [6] and exposed previously in the manuscript (see Eq. (3.1)). Pixels along one row yield one signal component. Let i be any of the row indices ($i = 1, \ldots, N$). The i^{th} row yields the signal component $z(i)$ as in Eq. (3.1). The signal components form the following signal vector: $\mathbf{z} = [z(1), z(2), \ldots, z(N)]^T$. In the case where circular-like contours are expected, an adequate signal generation process adapted to a quarter of the image also yields signal components. Pixels along the direction of generation D_i ($i = 1, \ldots, S$) yield the i^{th} signal component $z(i)$ (see Fig. 3.8(b)) which reads as in Eq. (3.9).

The signal components form the following signal vector: $\mathbf{z} = [z(1), z(2), \ldots, z(S)]^T$. The propagation parameter is further adapted so that the signal vector fits an array processing model.

3.4.2 Signal models

In this section, we derive exponential signal models for both linear blurred contours and circular blurred contours, and show that both contour types share the same signal model. To get a model for the signals generated, we first need a model for the contours, that is, or equivalently for their grey level values. We assume the gray level values $I(l,m)$ evolve aside a central position of the contour as an exponential function of the pixel position (see Fig. 3.7(a) and Fig. 3.7(b)). For linear-like contours:

$$I(l,m) = \frac{G}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}, \tag{3.11}$$

where $x = m - (x_0 - (l-1)tan(\theta))$. For circular-like contours, we get:

$$I(l,m) = \frac{G}{\sqrt{2\pi}\sigma} e^{-\frac{(\sqrt{(l-l_c)^2+(m-m_c)^2}-r_0)^2}{2\sigma^2}} \tag{3.12}$$

Referring to Eqs. (3.11) and (3.12), $\frac{G}{\sqrt{2\pi}\sigma}$ is the maximum gray level value. We expect that the exponential distribution, for instance a Gaussian distribution, of the gray level values in both cases facilitates the transfer of array processing methods to the considered parameter estimation issue.

Linear blurred contour

Firstly, we assume that the image contains only one blurred contour of width $2X_f$, main orientation θ, offset x_0, and spread parameter σ. Referring to Eqs. (3.1) and (3.11), the signal generated on the i^{th} sensor is expressed as:

$$\begin{array}{rcl} z(i) & = & \frac{G}{\sqrt{2\pi}\sigma} \sum_{x=1}^{X_f} e^{-j\mu(x_0+x-(i-1)tan(\theta))} e^{-\frac{x^2}{2\sigma^2}} \\ & + & \frac{G}{\sqrt{2\pi}\sigma} \sum_{x=1}^{X_f} e^{-j\mu(x_0-x-(i-1)tan(\theta))} e^{-\frac{x^2}{2\sigma^2}} \\ & + & \frac{G}{\sqrt{2\pi}\sigma} e^{-j\mu(x_0-(i-1)tan(\theta))} \end{array} \tag{3.13}$$

That is:

$$\begin{array}{l} z(i) = \frac{G}{\sqrt{2\pi}\sigma} \sum_{x=-X_f}^{X_f} e^{-j\mu(x_0+x-(i-1)tan(\theta))} e^{-\frac{x^2}{2\sigma^2}} \\ = \frac{G}{\sqrt{2\pi}\sigma} e^{-j\mu x_0} e^{j\mu(i-1)tan(\theta)} \sum_{x=-X_f}^{X_f} e^{-j\mu x} e^{-\frac{x^2}{2\sigma^2}} \end{array} \tag{3.14}$$

If σ is small enough compared to the number of columns in the image, we can turn the considered discrete calculation into a continuous case calculation. The intuition behind this approximation is that the values of the term $e^{-\frac{x^2}{2\sigma^2}}$ decrease rapidly when x increases, that is, when we get far from the pixels with gray level value $\frac{G}{\sqrt{2\pi}\sigma}$. Therefore a summation between $-X_f$ and X_f can be approximated as a summation between $-\infty$ and $+\infty$. A deeper study of this approximation is proposed in [60] for blurred circular contours. Eq. (3.14) becomes:

$$z(i) \approx$$

$$\frac{G}{\sqrt{2\pi}\sigma}\ e^{-j\mu x_0} e^{j\mu(i-1)tan(\theta)} \int_{x=-\infty}^{+\infty} e^{-j\mu x} e^{-\frac{x^2}{2\sigma^2}} dx \tag{3.15}$$

A general formula provides the equality:

$$\int_{x=-\infty}^{+\infty} e^{-ax^2+jbx} dx = \sqrt{\frac{\pi}{a}} e^{-\frac{b^2}{4a}} \tag{3.16}$$

Referring to Eq. (3.16), it is easy to express Eq. (3.15) by

$$z(i) = G\ e^{-j\mu x_0} e^{j\mu(i-1)tan(\theta)} e^{-\frac{\mu^2\sigma^2}{2}} \tag{3.17}$$

Eq. (3.17) is the signal received on the i-th sensor if one blurred contour is present. Secondly, we consider the case where the image contains:

- d blurred contours, with orientations θ_k, offsets x_{0k}, and spread parameters σ_k ($k = 1, \ldots, d$);
- uniformly distributed noise pixels, whose gray level values follow a Gaussian distribution.

The expression of the signal received by i^{th} sensor becomes:

$$z(i) = G\ \sum_{k=1}^{d} e^{-j\mu x_{0k}} e^{j\mu(i-1)tan(\theta_k)} e^{-\frac{\mu^2{\sigma_k}^2}{2}} + n(i) \tag{3.18}$$

where $n(i)$ is a noise term originated by the noise pixels during the signal generation process. It has been shown that this noise follows a Gaussian distribution [6]. We notice that, when σ tends to 0, Eq. (3.18) is equal to the equation obtained in the case of a one-pixel wide contour (refer to [6]). The signal components in Eq. (3.18) follow an array processing signal model, involving source amplitudes and steering vectors. Equation (3.18) can be expressed as:

$$z(i) = \sum_{k=1}^{d} s(k) c_i(\theta_k) + n(i) \tag{3.19}$$

For this we define:

1. the source amplitude associated with the k-th contour as: $s(k) = \frac{G}{\sqrt{2\pi}\sigma}\ e^{-j\mu x_{0k}} \sum_{x=-X_f}^{X_f} e^{-j\mu x} e^{-\frac{x^2}{2{\sigma_k}^2}}$, $k = 1, \cdots, d$. When the continuous approximation holds, the source amplitude components are expressed as:

$$s(k) = Ge^{-j\mu x_{0k}} e^{-\frac{\mu^2{\sigma_k}^2}{2}} \tag{3.20}$$

2. the steering vector associated with the k-th contour as: $\mathbf{c}(\theta_k) = [c_1(\theta_k), c_2(\theta_k), \cdots, c_N(\theta_k)]^T$, with $c_i(\theta_k) = e^{j\mu(i-1)\tan(\theta_k)}$.

In a matrix form, we get:

$$\mathbf{z} = \mathbf{C}(\theta)\mathbf{s} + \mathbf{n} \tag{3.21}$$

where $\mathbf{C}(\theta) = [\mathbf{c}(\theta_1), \mathbf{c}(\theta_2), \ldots, \mathbf{c}(\theta_d)]^T$, $\mathbf{s} = [s(1), s(2), \ldots, s(d)]^T$, and $\mathbf{n} = [n(1), n(2), \ldots, n(S)]^T$.

Extension to a circular blurred contour

In the case of blurred circular contours, it was shown in [60] that we get an array processing signal model if, instead of the fixed parameter μ, we choose a parameter which depends on the sensor index $\mu = \alpha(i-1)$, where α is a constant. As shown in [60], a circular blurred contour with spread parameter σ which is small enough yields the following signal components:

$$z(i) = exp(-j\alpha(i-1)r_0)exp(-\frac{\sigma^2\alpha^2(i-1)^2}{2}). \tag{3.22}$$

We notice that, contrary to Eq. (3.17), Eq. (3.22) contains a quadratic term, which is the modulus of each signal term. If we account for noise and consider the signal terms $z^{'}(i)$ such that:

$$z^{'}(i) = \frac{z(i)}{|z(i)|} = exp(-j\alpha(i-1)r_0) + n(i) \tag{3.23}$$

we get the following expression:

$$\mathbf{z}^{'} = \mathbf{c}(r_0) + \mathbf{n} \tag{3.24}$$

with $\mathbf{z}^{'} = \left[z^{'}(1), \ldots, z^{'}(S-1)\right]^T$, $\mathbf{c}(r_0) = [1, exp(-j\alpha r_0), \ldots, exp(-j\alpha(S-1)r_0)]^T$, and $\mathbf{n} = [n(1), \ldots, n(S-1)]^T$ being the noise vector. In the next subsection, we set the link between linear-like contours and circular-like contours: we propose a common signal model for both types of contours.

Common signal model

The notations above permit to express the signal generated out of the image in a matrix form:

$$\mathbf{z} = \mathbf{C}(\iota)\mathbf{s} + \mathbf{n} \tag{3.25}$$

where:
$\mathbf{z} = [z(1), z(2), \ldots, z(N_S)]^T$,
and $\mathbf{C}(\iota) = [\mathbf{c}(\iota_1), \mathbf{c}(\iota_2), \cdots, \mathbf{c}(\iota_d)]$. In the case of linear-like contours, $N_S{=}N$, and in the case of circular-like contours, $N_S{=}S$. Vector $\mathbf{n} = [n(1), n(2), \ldots, n(N_S)]^T$ represents noise resulting from possibly present outlier pixels. For linear blurred contours, $\mathbf{s} = [s(1), s(2), \cdots, s(d)]^T$, and $\mathbf{C}(\iota) = \mathbf{C}(\theta)$. For circular blurred contours, $\mathbf{s}$ is a scalar: $\mathbf{s} = 1$, and $\mathbf{C}(\iota) = \mathbf{c}(r_0)$.

Estimation of prior information needed for contour characterization

The proposed method is entirely blind. We propose to distinguish between line and circle with two linear antennas, placed aside the image on the left or the bottom side. A threshold value is applied to the generated signals to get rid of noise. When the signals received on both antennas exhibit first and last components which are zero-valued, one or several circles are present. Their center coordinate l_c (resp. m_c) are the middle of the non-zero sections of the signal generated on the left (resp. bottom) array. If only the left (resp. bottom) array signal contains zero sections, at least one nearly horizontal (resp. vertical) line is present. If no array signal contain zero sections, a diagonal line is present. If a horizontal line is present, the signal generated on the bottom array is further used instead of the signal generated on the left array. The number of lines is estimated by MDL (minimum description length) criterion, as explained in the following.

3.4.3 Subspace-based methods for the estimation of contour parameters

In this section, we adapt subspace-based methods coming from array processing to estimate some of the parameters of blurred contours. Firstly, we seek for linear blurred contours: a subspace-based method and Fourier processing provide orientations and offsets $\{\theta_k,\ x_{0k},\ k = 1, \ldots, d\}$. Secondly, we seek for a circular blurred contour, and a subspace-based method provides the radius r_0.

Linear blurred contours

We adapt a subspace-based method coming from array processing to retrieve the main orientation of the contour, and apply Fourier processing to retrieve its center offset.

• **Estimation of the blurred contour orientation** Equation (3.25) is exactly analogous to an array processing equation [94]. Therefore, an array processing method can be applied to the signals generated from the image. However, we do not afford several signal snapshots, and an array processing method such as MUSIC [94] cannot be directly applied. We have to simulate artificially multiple signal snapshots out of a single sample array data by splitting the array (of length N) into smaller overlaying sub-arrays (of length M). This is called spatial smoothing technique. For more information about spatial smoothing, refer to [6, 76]. We get P snapshots, where P is such that: $M = N - P + 1$. From the observation vector $\mathbf{z}$ we obtain P overlapping sub-vectors. By grouping all sub-vectors obtained in matrix form, we obtain:

$$\mathbf{Z}_P = [\mathbf{z}_1, \cdots, \mathbf{z}_P] \tag{3.26}$$

The covariance matrix of all sub-vectors of Eq. (3.26) is defined by:

$$\mathbf{R}_{zz} = \mathbf{Z}_P {\mathbf{Z}_P}^H \tag{3.27}$$

MDL criterion, when applied to $\mathbf{R}_{zz}$, provides the number of dominant eigenvalues of $\mathbf{R}_{zz}$, which is equal to the number of contours d [6]. We estimate the parameters θ_k, $k = 1, \ldots, d$ through the maxima of the pseudo spectrum $\mathrm{F}(\theta)$ [94]:

$$\mathrm{F}(\theta) = \frac{1}{\| \mathbf{c}^H(\theta) \cdot \mathbf{U}_2 \|^2} \tag{3.28}$$

where θ is the parameter upon which the optimization is done, and $\mathbf{c}(\theta)$ is a model for the signal subspace vectors: $\mathbf{c}(\theta) = [c_1(\theta), c_2(\theta), \cdots, c_M(\theta)]^T$, with $c_i(\theta) = e^{j\mu(i-1)\tan(\theta)}$. Matrix $\mathbf{U}_2$ columns span the noise subspace of the data: it is composed of the $M-d$ columns of the covariance matrix $\mathbf{R}_{zz}$ associated with its $M-d$ smallest eigenvalues. We notice that a constraint on M and P with respect to the number of expected sources is the following: $M > d$ and $P \geq d$ (to get a full rank covariance matrice). From $M = N - P + 1$ we also get: $M \leq N - d + 1$.

• **Estimation of the blurred contour offset** The estimation of the offset parameters of linear contours falls into two steps: first, an approximation is made to get a rough value of the offsets, which is needed to estimate the spread parameters. Supposing we have at disposal the spread parameters (whose estimation is presented further in this chapter), it is possible to get a more accurate estimate of the contour offsets. They are first grossly estimated, and then the accurate estimate is retrieved with the knowledge of the spread parameters.

Once the orientation values are known, the offset values can be estimated by variable speed generation scheme [21] and TLS-ESPRIT algorithm [6]. We set $\mu = \alpha(i-1)$. Eq. (3.18) becomes:

$$z(i) = \; G\,\Sigma + n(i) \tag{3.29}$$

with $\Sigma =$

$$\sum_{k=1}^{d} e^{-j\alpha(i-1)x_{0k}} e^{j\alpha(i-1)^2 tan(\theta_k)} e^{-\frac{(\alpha(i-1))^2 {\sigma_k}^2}{2}}$$

Then, each contour is considered successively. We can consider for instance the first orientation θ_1. As θ_1 value has been estimated, we can divide $z(i)$ by the term $e^{j\alpha(i-1)^2 tan(\theta_1)}$. We obtain:

$$w(i) = z(i)/e^{j\alpha(i-1)^2 tan(\theta_1)} =$$

$$G\; e^{-j\alpha(i-1)x_{01}} e^{-\frac{(\alpha(i-1))^2 {\sigma_1}^2}{2}} + n'(i) \tag{3.30}$$

where $n'(i)$ is a noise term resulting from the influence of noisy pixels and all but the first contour. At this point, the value of σ_1 is not known and we propose an approximation which permits to get a gross estimate of x_{01} without the prior knowledge of σ_1. If the propagation parameter α is chosen such that $\alpha(i-1) << 1$, $\forall\, i = 1, \ldots, N$, we can adopt the following approximation:

$$w(i) \approx \tilde{w}(i) = \; G\; e^{-j\alpha(i-1)x_{01}} + n(i) \tag{3.31}$$

The signal $\tilde{\mathbf{w}} = [\tilde{w}(1), \tilde{w}(2), \ldots, \tilde{w}(N)]^T$ can be analysed by Fourier transform, which provides the estimated offset value $\hat{x_{01}}$:

$$\hat{x_{01}} = \underset{x_{01}}{argmax}(|FT(\tilde{\mathbf{w}})|) \tag{3.32}$$

where FT denotes Fourier transform. The term $\underset{x_{01}}{argmax}$ means that we seek for the value of x_{01} which maximizes $|FT(\tilde{\mathbf{w}})|$. The division process of Eq. (3.30) and the Fourier analysis of Eq. (3.32) are repeated for each value $k = 1, \ldots, d$. Fourier analysis is fast and easy to implement. At this point a gross estimate of the offset values is available, which will be used to estimate the spread parameter values σ_k, $k = 1, \ldots, d$. The estimation of the spread parameters out of the grossly estimated offset values is explained further in this chapter. Let's assume that all spread values are available, and avoid the approximation of Eq. (3.31).
Starting from the expression of $w(i)$ in Eq. (3.30), we derive the signal $\omega(i)$, $i = 1, \ldots, d$:

$$\begin{aligned} \omega(i) &= w(i)/(e^{-\frac{(\alpha(i-1))^2 {\sigma_1}^2}{2}}) \\ &= G\ e^{-j\alpha(i-1)x_{01}} + n'(i) \end{aligned} \tag{3.33}$$

where $n'(i)$ is a noise term resulting from the influence of all but the first contour. The signal components $\omega(i)$ form the signal vector
$\boldsymbol{\omega} = [\omega(1), \omega(2), \ldots, \omega(N)]^T$ which can be analysed by Fourier transform to provide the estimate $\hat{x_{01}}$ of the offset value:

$$\hat{x_{01}} = \underset{x_{01}}{argmax}(|FT(\omega)|) \tag{3.34}$$

The division processes performed in Eqs. (3.30) and (3.33) are applied d times, that is, for each contour, to retrieve the refined estimates $\hat{x_{0k}}$, $k = 1, \ldots, d$.

Circular blurred contours: estimation of the radius

At this point the center coordinates $\{l_c; m_c\}$ are known (see subsection 3.4.2). From Eq. (3.24), we notice that the problem of radius estimation is similar to the retrieval of harmonics in several signal processing fields such as radar, sonar, communication. The resulting signal appears as a single sinusoid with unitary amplitude and frequency:

$$f = -\alpha r_0/2\pi \tag{3.35}$$

MFBLP method (Modified Forward-Backward Linear Prediction), which was previously presented in the manuscript, in subsection (3.2.3), is adequate for frequency retrieval from coherent signals, in particular signals with unitary amplitude. We adapt it to the signal vector $\mathbf{z}'$ (see Eq. (3.24)) to estimate the radius of a single circle. To reduce the computational load of radius estimation, and on condition that still one circle is solely expected, the Fourier transform with adequate frequency can yield the radius value.

3.4.4 Optimization strategy for spread parameter estimation of the blurred contours

In this subsection we propose least-square criteria which involve the generated signals and either the signal model of Eq. (3.25) for linear contours, or the signal model of Eq. (3.22) for circular contours. The proposed optimization strategy should provide the spread parameter σ for either each of the blurred linear contours or for the blurred circular contour.

Linear blurred contours

The contour orientations estimated by MUSIC algorithm are used to compute the steering matrix $\mathbf{C}(\theta)$ (see Eq. (3.25)). The source vector $\mathbf{s}$ depends not only on the offset parameters x_{0k} $(k = 1, \ldots, d)$, but also on the spread parameters σ_k $(k = 1, \ldots, d)$. Therefore we propose to retrieve the components of the source vector $\mathbf{s}$, through the following criterion minimization:

$$\hat{\mathbf{s}} = \underset{\mathbf{s}}{argmin}(||\mathbf{z} - \mathbf{Cs}||^2) \tag{3.36}$$

where $||.||$ represents the norm induced by the usual scalar product of $\mathbb{C}^N$. It is easy to show that the density function of the measurement noise is Gaussian if the noise pixels are identically distributed over the image [6]. Therefore, the above least-squares problem provides the maximum likelihood estimate for the source vector. We remind that the relationship between the source vector components and the spread parameter values is given by (see Eq. (3.20)):

$$s(k) = f(\sigma_k) = G\ e^{-j\mu x_{0k}} e^{-\frac{\mu^2 {\sigma_k}^2}{2}} \tag{3.37}$$

We denote by $\sigma = [\sigma_1, \ldots, \sigma_d]^T$ the vector containing all spread parameter values, and by $\mathbf{f}(\sigma) = [f(\sigma_1), \ldots, f(\sigma_d)]^T = [s(1), \ldots, s(d)]^T$ the source vector. We denote by $\hat{\sigma} = [\hat{\sigma_1}, \ldots, \hat{\sigma_d}]^T$ the vector containing the estimates of all spread parameter values. From Eqs. (3.36) and (3.37), we get:

$$\hat{\sigma} = \underset{\sigma}{argmin}(||\mathbf{z} - \mathbf{Cf}(\sigma)||^2) \tag{3.38}$$

which can be expressed as:

$$\hat{\sigma} = \underset{\sigma}{argmin}(J_{line}(\sigma)) \tag{3.39}$$

where J_{line} denotes the criterion to be minimized. To solve Eq. (3.38) and minimize criterion J_{line}, we adopt a recurrence loop to modify recursively the vector $\hat{\sigma}$. The series vectors are obtained from the relation

$$\hat{\sigma}^q \rightarrow \mathbf{f}(\hat{\sigma}^q) \rightarrow J_{line}(\hat{\sigma}^q), \forall\ q \in \mathbb{N} \tag{3.40}$$

When q tends to infinity, the criterion J_{line} tends to zero and $\hat{\sigma}_k^q = \sigma_k$, $\forall\ k = 1, \ldots, d$. The criterion J_{line} presented in Eq. (3.39) is a Lipschitz continuous function of the vector of variables σ and therefore fulfils the requirements of the DIRECT (DIviding RECTangles) method [62]. Therefore, to carry out this recurrence loop, we can adopt the robust DIRECT optimization method [62]. DIRECT method is initialized by $\hat{\sigma}^0$, and a research space which is an acceptable interval for each value. Vector $\hat{\sigma}^0$ and the research space are *a priori* fixed by the user. The main property of DIRECT is that it is able to obtain the global minimum of a function. DIRECT normalizes the research space in a hypercube and evaluates the solution which is located at the center of this hypercube. Then, some solutions are evaluated and the hypercube is divided into smaller cubes, supporting the zones where the evaluations are small. When the required number of iterations $q = It$ is reached, DIRECT provides the estimated vector of spread parameters $\hat{\sigma}^{It} = [\sigma_1, \sigma_2, \ldots, \sigma_d]$.

Extension to a blurred circular contour

In the case of a blurred circular circle, we propose the following algorithm: we start from the signal $\mathbf{z} = [z(1), z(2), \ldots, z(S)]^T$ whose components $z(i)$ are defined in Eq. (3.22). The value of r_0 is known at this point, and can be used to obtain the signal components $z''(i)$ defined as follows: $z''(i) = z(i)/exp(-j\alpha(i-1)r_0)$. Let's then denote by $\mathbf{z}''_{model}$ the signal whose components are defined by $z''_{model}(i) = exp(-\frac{\sigma^2\alpha^2(i-1)^2}{2})$, and let's denote by $\mathbf{z}''_{image}$ the signal whose components are defined by: $z''_{image}(i) = z(i)/exp(-j\alpha(i-1)r_0)$ and obtained from the signal components $z(i)$ generated out of the image. With these notations, the spread parameter σ can be estimated as follows:

$$\hat{\sigma} = \underset{\sigma}{argmin}(||\mathbf{z}''_{image} - \mathbf{z}''_{model}||^2) \tag{3.41}$$

which can be expressed as:

$$\hat{\sigma} = \underset{\sigma}{argmin}(J_{circle}(\sigma)) \tag{3.42}$$

where J_{circle} denotes the criterion to be minimized. Contrary to the case of linear blurred contours described in subsection 3.4.4, the global optimization method DIRECT [62] is not adequate to minimize the criterion J_{circle} presented in Eq. (3.42). An advanced well-known local minimizer is adapted: the Nelder-Mead Simplex Method [70]. It is meant to minimize a scalar-valued nonlinear function of n real variables. It is then adequate to minimize the criterion $J_{circle}(\sigma)$, which constitutes a nonlinear function of the parameter σ. Nelder-Mead method involves four scalar parameters: the coefficients of reflection (ρ_{NM}), expansion (χ_{NM}), contraction (γ_{NM}), and shrinkage (σ_{NM}).

3.4.5 Exemplification of the blurred contour retrieval methods

In the following experiment, we analyse an image including two linear blurred contours, with different spread values (see Fig. 3.9). The image has size 400 × 400. The center offsets of the two blurred contours are $x_{01} = 200$ and $x_{02} = 170$, and the main orientation of two contours are $\theta_1 = -18°$ and $\theta_2 = 18°$. The spread values are $\sigma_1 = 8$ and $\sigma_2 = 1$.

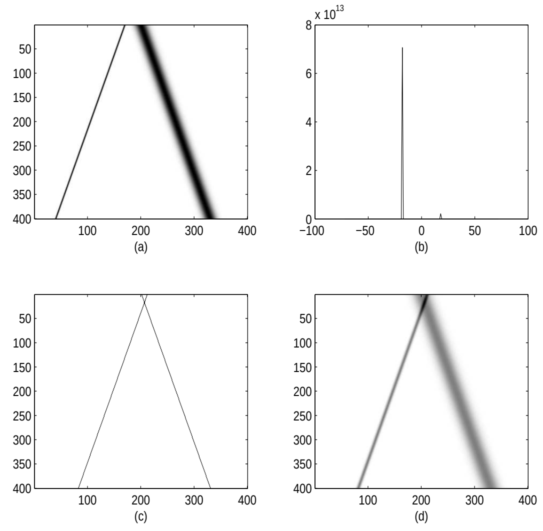


Figure 3.9 — Blurred linear contours: (a) processed image with a blurred contour and a one-pixel wide contour; (b) pseudo spectrum when MUSIC algorithm is exploited; (c) center contours; (d) final result

The estimated orientations of the blurred contours are $\hat{\theta_1} = -18°$ and $\hat{\theta_2} = 18°$. The offsets are estimated as $\hat{x_{01}} = 200.5$ and $\hat{x_{02}} = 211$ pixels. The estimated spread parameters are $\hat{\sigma_1} = 10.9$ and $\hat{\sigma_2} = 2.4$. Fig 3.9(b) shows that the contour with low spread value is hardly detected by MUSIC algorithm. The dominating influence of the most blurred contour in the generate signals of Eq. (3.26) also explains the slight bias (41 pixels) obtained on the offset of the least blurred contour.

We present a result obtained from an image of size 200 × 200 pixels (see Fig. 3.10), containing a blurred circle. The experimental conditions and expected values for the blurred circular contour are as follows: the center coordinates are $\{l_c, m_c\} = \{70, 60\}$; the radius is $r_0 = 45$ pixels; the spread value is $\sigma = 5$. The proposed methods yield the following estimated parameters out of the generated signals: the estimated center coordinates are $\left\{\hat{l_c}, \{l_c, m_c\}\, m_c\right\} = \{70, 60\}$, the estimated radius value is $\hat{r_0} = 45.4$ pixels, and the estimated spread value is $\hat{\sigma} = 5.6$. As a comparative method we chose Chan and Vese levelset algorithm. As expected, this method manages to focus on the blurred contour boundaries, but it does not characterize the blur, contrary to the proposed method.

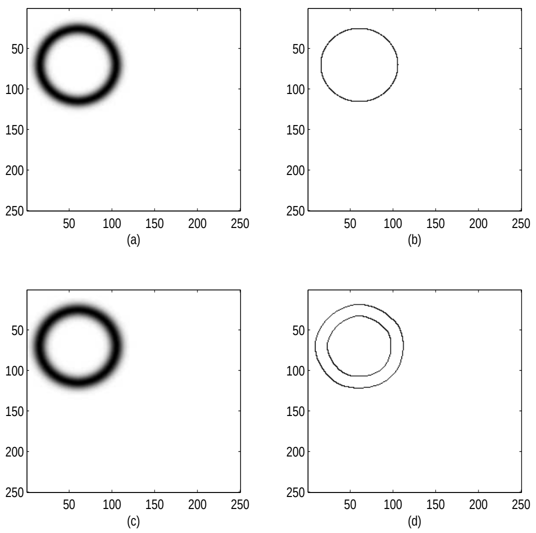


Figure 3.10 — Blurred circular contours: (a) initial image; (b) initialisation circle; (c) final result; (d) results by Chan and vese method

3.5 Retrieval of distorted contours

3.5.1 Nearly rectilinear contour retrieval

We keep the same signal generation formalism as for straight line retrieval. The more general case of distorted contour estimation is proposed. The reviewed method relies on constant speed signal generation scheme, and on a recursive optimization method.

Initialization of the proposed algorithm

To initialize our recursive algorithm, we apply SLIDE algorithm, which provides the parameters of the straight line that fits the best the expected distorted contour. In this section, we consider only the case where the number d of contours is equal to one. The parameters angle and offset recovered by the straight line retrieval method are employed to build an initialization vector $\mathbf{x}_0$, containing the initialization straight line pixel positions:

$$\mathbf{x}_0 = [x_0, x_0 - \tan(\theta), \ldots, x_0 - (N-1)\tan(\theta)]^T$$

Fig. 3.11 presents a distorted curve, and an initialization straight line that fits this distorted curve.

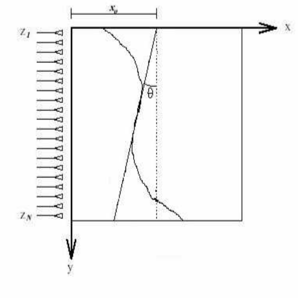


Figure 3.11 — A model for an image containing a distorted curve

Distorted curve: proposed algorithm

We aim at determining the N unknowns $x(i),\ i = 1, \dots, N$ of the image, forming a vector $\mathbf{x}_{input}$, each of them taken into account respectively at the i^{th} sensor:

$$z(i) = exp(-j\mu x(i)),\ \forall\ i = 1, \dots, N \tag{3.43}$$

The observation vector is

$$\mathbf{z}_{input} = [exp(-j\mu x(1)), \dots, exp(-j\mu x(N))]^T \tag{3.44}$$

We start from the initialization vector $\mathbf{x}_0$, characterizing a straight line that fits a locally rectilinear portion of the expected contour. The values $x(i),\ i = 1, \dots, N$ can be expressed as: $x(i) = x_0 - (i-1)\tan(\theta) + \Delta\ x(i),\ i = 1, \dots, N$ where $\Delta\ x(i)$ is the pixel shift for row i between a straight line with parameters θ and x_0 and the expected contour. Then, with k indexing the steps of this recursive algorithm, we aim at minimizing

$$J(\mathbf{x}_k) = ||\mathbf{z}_{input} - \mathbf{z}_{estimated\ for\ \mathbf{x}_k}||^2 \tag{3.45}$$

where $||.||$ represents the $\mathbb{C}^N$ norm. For this purpose we use fixed step gradient method: $\forall k \in \mathbb{N}: \quad \mathbf{x}_{k+1} = \mathbf{x}_k - \lambda\nabla(J(\mathbf{x}_k))$, λ is the step for the descent. At this point, by minimizing criterion J (see Eq. (3.45)), we find the components of vector $\mathbf{x}$ leading to the signal $\mathbf{z}$ which is the closest to the input signal in the sense of criterion J. Choosing a value of μ which is small enough (see Eq. (3.1)) avoids any phase indetermination. A variant of the fixed step gradient method is the variable step gradient method. It consists in adopting a descent step which depends on the iteration index. Its purpose is to accelerate the convergence of gradient. A more elaborated optimization method based on DIRECT algorithm [62] and spline interpolation [75] can be adopted to reach the global minimum of criterion J of Eq. (3.45). This method is applied to modify recursively signal $\mathbf{z}_{estimated\ for\ \mathbf{x}_k}$: at each step of the recursive procedure vector $\mathbf{x}_k$ is computed by making an interpolation between some "node" values that are retrieved by DIRECT. The interest of the combination of DIRECT with spline interpolation comes from the elevated computational load of DIRECT. Details about DIRECT algorithm are available in [62]. Reducing the number of unknown values retrieved by DIRECT reduces drastically its computational load. Moreover, in

the considered application, spline interpolation between these node values provides a continuous contour. This prevents the pixels of the result contour from converging towards noisy pixels. The more interpolation nodes, the more precise the estimation, but the slower the algorithm.
After nearly linear contours, we focus on nearly circular contours.

3.5.2 Nearly circular contour retrieval

To retrieve the distortions between an expected star-shaped contour and a fitting quarter of circle, we work successively on each quarter of circle, and retrieve the distortions between one quarter of the initialization circle and the part of the expected contour that is located in the same quarter of the image. As an example, in Fig. 3.3, The right bottom quarter of the considered image is represented in Fig. 3.3(b).

The optimization method that retrieves the shift values between the fitting circle and the expected contour is the following:
A contour in the considered sub-image can be described in a set of polar coordinates by: $\{\rho(i), \theta(i),\ i = 1, \ldots, S\}$. We aim at estimating the S unknowns $\rho(i),\ i = 1, \ldots, S$ that characterize the contour, forming a vector:

$$\rho = [\rho(1), \rho(2), \ldots, \rho(S)]^T, \tag{3.46}$$

The basic idea is to consider that ρ can be expressed as: $\rho = [r + \Delta\rho(1), r + \Delta\rho(2), \ldots, r + \Delta\rho(S)]^T$ (see Fig. 3.3), where r is the radius of a circle that approximates the expected contour. The parameters $\Delta\rho(1), \ldots, \Delta\rho(S)$ can be estimated by a gradient-type algorithm or DIRECT combined with spline interpolation, as was performed in [77]. However, these two methods exhibit limitations when the considered contour is highly distorted. The computational load required by gradient is elevated, and the regularity constraints on spline interpolation prevent from providing to the distortions their actual shape. Hence the method proposed in [61], which is summarized in the next subsection.

3.5.3 Highly distorted star-shaped contour retrieval

In this subsection, we consider star-shaped contours. On the one hand, this is a limiting model because for one angle value in a polar set of parameters, there must be only one pixel of the contour. On the other hand, this allows the distortion amplitudes to be as elevated as possible, as soon as the contour remains in the processed image. The signal generation method is still based on virtual sensors placed along a circular antenna, but the formula providing the signal components is slightly different.

Problem formulation

Assume that a closed circular contour is in an $N \times N$ recorded image $I_{l,m}$ (see Fig. 3.12). The most simple star-shaped contour is the circle. A circle is supposed to have center coordinates (l_c, m_c) and radius r. Note that, for a binary image, $I_{l,m} = 1$ on the contour and $I_{l,m} = 0$ otherwise. The signal component for a given sensor i is generated by the pixels in every D_i direction as follows:

$$z_i = \underset{(l,m)\in D_i}{\sum_{l=1}^{N_s}\sum_{m=1}^{N_s}} I_{l,m}\sqrt{l^2+m^2},\ i = 1,\cdots,S \tag{3.47}$$

where N_s is the maximum number of rows and columns in the sub-image. The signal components form the signal vector $\mathbf{z} = [z_1, z_2, \ldots, z_S]^T$.

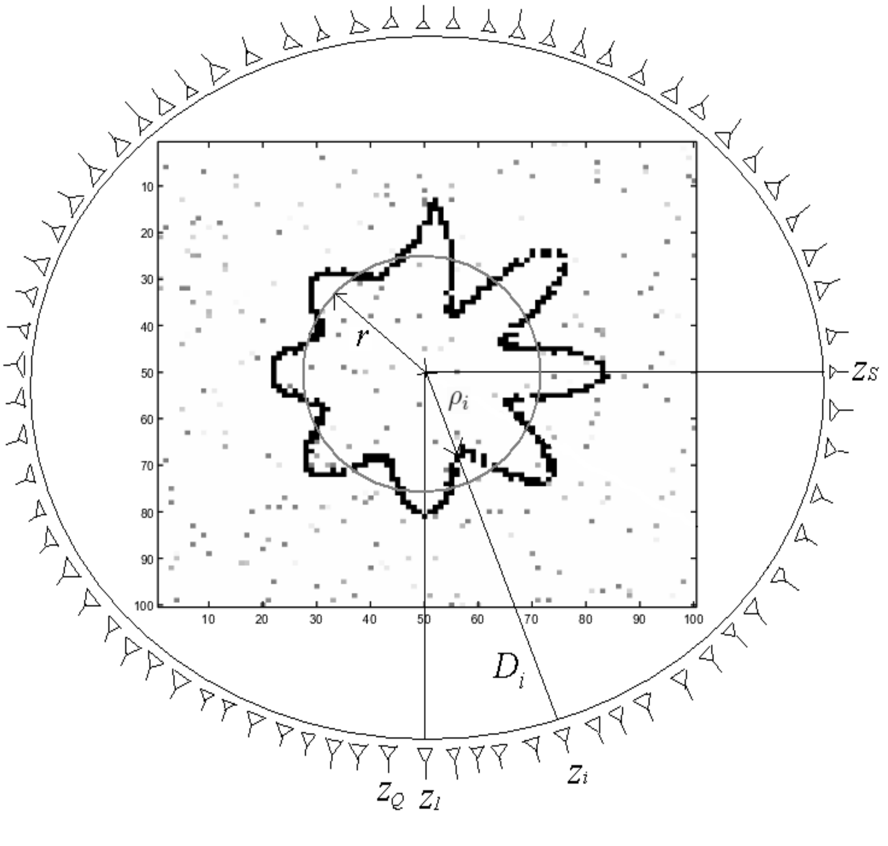


Figure 3.12 — A model for an image containing a highly distorted circle

The considered signal generation process requires the knowledge of the center coordinates (l_c, m_c). We explain in subsection 3.3.3 how to estimate these center coordinates. When a single one-pixel wide circular contour with radius r is present, the signal components read:

$$z_i = r, \quad i = 1, \ldots, S \tag{3.48}$$

When a distorted nearly circular contour is considered, the signal components read:

$$z_i = r + \Delta\rho(i), \quad i = 1, \ldots, S \tag{3.49}$$

In the rest of the subsection, we denote $\Delta\rho(i)$ as x_i, $i = 1, \ldots, Q$.

From the signals $\mathbf{z} = [z_1, z_2, \cdots, z_Q]^T$ of Eq. (3.49), we wish to retrieve the radius value r, and the oscillations x_i, $i = 1, \cdots, Q$, in particular from contours presenting a strong concavity. Without loss of generality, we define r as the mean value of the components z_i $i = 1, \ldots, Q$. r is estimated as:

$$r = \bar{z} \tag{3.50}$$

where $\bar{z}$ is defined as: $\bar{z} = \frac{1}{S}\sum_{i=1}^{S} z_i$. Then, we can compute:

$$x_i = z_i - r, \; i = 1, \ldots, Q \tag{3.51}$$

The values x_i, $i = 1, \cdots, Q$ are exactly the edge oscillation values in the case where the image is not impaired with noise. If the image is impaired with uniformly distributed noise, the computation of Eq. (3.51) provides signal components x_i, $i = 1, \ldots, Q$ which are impaired by random noise, due to the influence of random noise pixels on the signal generation process. Therefore, we seek for a method which retrieves the oscillations of possibly strongly concave contours, and which is robust to noise. For this, we propose in the following a model for edge oscillations x_i, $i = 1, \cdots, Q$. We will further adapt an advanced damped frequency retrieval method to characterize the edge oscillations, in accordance with the proposed model.

Edge oscillations modelled as damped sinusoids

For the edge oscillations of a star-shaped contour, the pixel coordinates in a polar representation are supposed to follow a generalized version of the sinusoidal model, that is, K damped sinusoidal components, each of which has respective amplitude, frequency and damping factor. So we model the edge oscillations as follows:

$$x_i = \sum_{k=1}^{2K} a_k e^{j\phi_k} e^{(-d_k + j\omega_k)(i-1)} = \sum_{k=1}^{2K} c_k w_k^{(i-1)}, \quad i = 1, \ldots, Q \tag{3.52}$$

where $j = \sqrt{-1}$. In Eq. (3.52), x_i represents the oscillation magnitude for $i = 1, \ldots, Q$, a_k is amplitude of the k-th sinusoidal component, d_k its damping factor, ω_k its angular frequency, and ϕ_k its initial phase. Note that damping factor d_k may be negative. In this case, the amplitude of k-th component grows with index i. $c_k = a_k e^{j\phi_k}$ is the complex-valued amplitude of k-th component, and $w_k = e^{(-d_k + j\omega_k)}$.

The observed signal segment $\mathbf{x} = [x_1, x_2, \ldots, x_Q]^T$ is entirely characterized by the parameters a_k, d_k, ω_k, ϕ_k, $k = 1, \ldots, 2K$. The number K of sinusoidal components can be estimated by MDL criterion [108].

We then have to determine the parameters cited above by applying a variant of the parameter estimatorFirstly, we rearrange the signal segment $\mathbf{x}$ in a Hankel matrix with $L \times M$ as follows:

$$\mathbf{X} = \begin{bmatrix} x_1 & x_2 & \dots & x_M \\ x_2 & x_3 & \dots & x_{M+1} \\ \vdots & \vdots & & \vdots \\ x_L & x_{L+1} & \dots & x_Q \end{bmatrix} \tag{3.53}$$

where L, K, and Q are related by: $L \geq 2K$, $M \geq 2K$ and $Q = L + 2K - 1$.

Then, by implementing the Vandermonde Decomposition (VD) for Hankel data matrix of Eq. (3.53) with rank of $2K$, $\mathbf{X}$ can be written as:

$$\mathbf{X} \overset{\text{VD}}{=} \mathbf{SCT}^T,$$

where $(\cdot)^T$ denotes matrix transposition, $\mathbf{C} = diag(c_1, c_2, \dots, c_{2K})$,

$$\mathbf{S} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ w_1^1 & w_2^1 & \dots & w_{2K}^1 \\ \vdots & \vdots & & \vdots \\ w_1^{L-1} & w_2^{L-1} & \dots & w_{2K}^{L-1} \end{bmatrix},$$

$$\mathbf{T} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ w_1^1 & w_2^1 & \dots & w_{2K}^1 \\ \vdots & \vdots & & \vdots \\ w_1^{M-1} & w_2^{M-1} & \dots & w_{2K}^{M-1} \end{bmatrix}.$$

According to the shift-invariant property in column space,

$$\mathbf{S}^L = \mathbf{S}^F \mathbf{Z}, \tag{3.54}$$

where $\mathbf{S}^L$ is a matrix containing all but the first row of $\mathbf{S}$, and $\mathbf{S}^F$ is a matrix containing all but the last row of $\mathbf{S}$. $\mathbf{Z}$ is a diagonal matrix whose nonzero terms depend on the expected parameters. By performing SVD, $\mathbf{X}$ can be decomposed as:

$$\mathbf{X} \overset{\text{SVD}}{=} \begin{bmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & \mathbf{0} \\ \mathbf{0} & \Sigma_2 \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^H \\ \mathbf{V}_2^H \end{bmatrix} \tag{3.55}$$

where $(\cdot)^H$ is the Hermitian transposition, Σ_1 contains the largest $2K$ singular values of $\mathbf{X}$ and Σ_2 the $L - 2K$ singular values of $\mathbf{X}$. The matrices $\mathbf{U}_1$ and $\mathbf{V}_1^H$ contain the first $2K$ left and right singular vectors, and their dimension is $L \times 2K$ and $M \times 2K$, respectively. Because the rank of $\mathbf{X}$ is $2K$, all values of Σ_2 are null. Therefore, we can express $\mathbf{X}$ as:

$$\mathbf{X} = \mathbf{U}_1 \Sigma_1 \mathbf{V}_1^H, \tag{3.56}$$

and we get the following equation from Eq. (3.54) by orthogonal basis transformation.

$$\mathbf{U}_1^F \mathbf{Z}^u = \mathbf{U}_1^L \tag{3.57}$$

where $\mathbf{U}_1^F$ contains all but the last row of matrix $\mathbf{U}_1$, $\mathbf{U}_1^L$ contains all but the first row of matrix $\mathbf{U}_1$, and $\mathbf{Z}^u$ is a similarity transform of $\mathbf{Z}$. The damping factors d_k and frequencies ω_k $(k = 1, \ldots, 2K)$ of the exponential sinusoidal model (see Eq. (3.52)) are estimated from the eigenvalues of $\mathbf{Z}^u$. Then we substitute these estimated d_k and ω_k in Eq. (3.52) and compute the least-squares solution of the N linear equations. Finally, the amplitude a_k and phase ϕ_k of each component are determined from the magnitude and angle of c_k in Eq. (3.52). According to these estimated parameters, we can reconstruct the contour with oscillations. The pixel coordinates in the contour are given as:

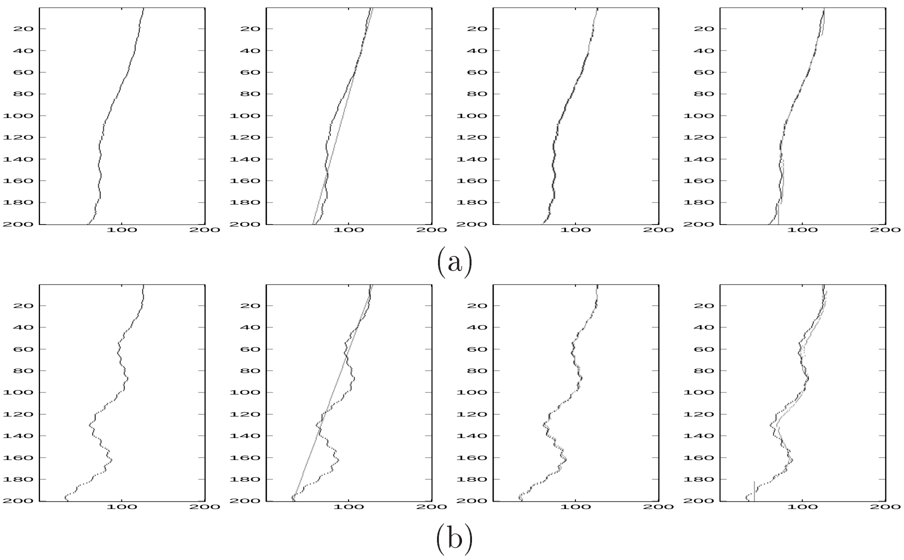
$$\rho_i = r + \hat{x}_i, \quad i = 1, \cdots, Q$$

where $\hat{x}_i$ is initial estimation of x_i, $i = 1, \cdots, Q$. We now afford the values of the contour distortions, for any angle coordinate θ_i. We also afford, r, the radius of the fitting circle. With the knowledge of the center, whose estimation is the purpose of subsection 3.3.3, we reconstruct perfectly the expected contour.

3.5.4 Exemplification of the distorted contour retrieval methods

We consider two approximately linear distorted contours, with different distortion amplitude. These contours are the ones of Figs. 3.13(a) and (b). The pixel of the least and most distorted contours, and their estimation by the proposed method and by GVF [111] are drawn on Figs. 3.13(a) and (b).

We now consider highly distorted approximately circular contours. We denote by $ME_{\mathbf{x}}$ the mean error between actual and estimated radial coordinate oscillations. In some cases, due to the acquisition conditions or the image quantization, the continuous form of contour edge is not perfect. It is therefore very interesting to evaluate the robustness of the proposed method to pixel location errors. We produce test images by initially creating a star-shaped contour (see Fig. 3.14 (a)); and then adding pixel displacement by modifying the actual pixel radial coordinates with a Gaussian random variable with mean value 0 and standard deviation 1 (see Fig. 3.14(b)). We assume there exists equally distributed random noise in the image, with mean value 0 and standard deviation 10^{-2}. Referring to Figs. 3.14 (d)-(i), when the proposed method is applied, the mean error is $ME_{\mathbf{x}} = 1.61$ when small random displacements are added; and $ME_{\mathbf{x}} = 1.86$ when larger random displacements are added. When Gradient method is applied, the mean error value is increased dramatically from $ME_{\mathbf{x}} = 1.78$ to $ME_{\mathbf{x}} = 2.25$. When GVF is applied, the mean error value is increased from $ME_{\mathbf{x}} = 1.90$ to $ME_{\mathbf{x}} = 2.42$. So, Figs. 3.14 (d)-(i) show that the proposed method



(a)

(b)

Figure 3.13 — (a) Least distorted contour: initialization, results obtained (b) Most distorted contour: initialization, results obtained by the proposed method and GVF respectively.

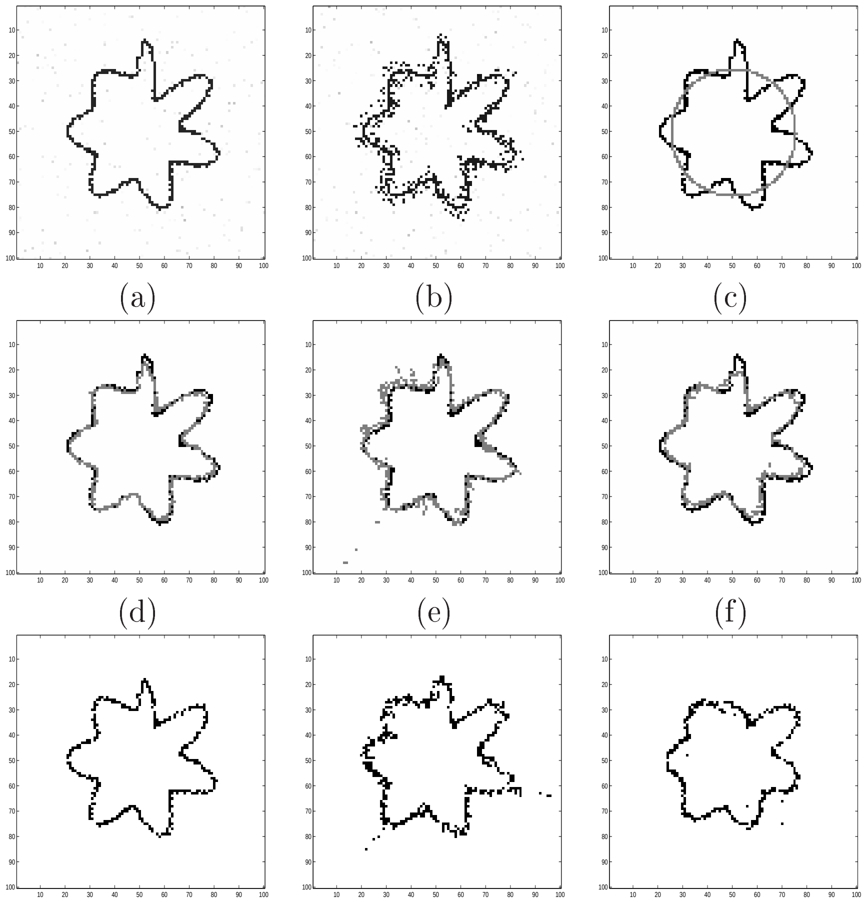
is not sensitive to the random pixel displacements, contrary to Gradient method and GVF method. This is due to the fact that the proposed method processes the signal generated from the image as a whole, providing parameters of interest, whereas Gradient method and GVF are local methods, which may focus on random pixels.

This type of contour, though being rigourously star-shaped, makes us think about the outside borders of hands, captured on video frames. In the next sections of this manuscript, we will show how this intuition yields a specific signature inspired by the signal generation methods presented above.

3.6 Conclusion of the chapter

This chapter presents an overview of an original approach of contour detection which has been proposed during the past years. Array processing signal models and methods have been adapted to various aspects of contour detection. Originally, this approach consisted in considering a contour as a wavefront and the image background as a propagation medium [6]. In this framework, a signal generation scheme along the rows of the image yields signal components. Each row is associated with a virtual sensor, and the whole set of sensors forms a uniform linear antenna. This approach was extended to circles, by adapting the shape of the antenna [76], and choosing radial directions for the generation of signal components.

An extension of these methods, inspired from real-world issues, was proposed thereafter: it consists in characterizing blurred contours. Blur can indeed occur because of de-focus, transmission media inhomogeneities, etc. We reminded what are the principles of characterization of either linear of circular blurred contours. An outline of the



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 3.14 — (a) processed image: $\overline{\kappa} = 2.7\ 10^{-3}$, with small edge perturbation and noise $(0, 10^{-2})$; (b) processed image, with large edge perturbation and noise $(0, 10^{-2})$; (c) initialization of the methods for both processed images; (d-f) superposition processed and result obtained on 'a' by the proposed method ($ME_{\mathbf{x}} = 1.58$), Gradient method ($ME_{\mathbf{x}} = 1.78$), and GVF method ($ME_{\mathbf{x}} = 1.90$); (g-i) result obtained on 'b' by the proposed method ($ME_{\mathbf{x}} = 1.86$), Gradient method ($ME_{\mathbf{x}} = 2.25$), and GVF method ($ME_{\mathbf{x}} = 2.42$).

proposed blurred contour estimation methods is as follows:

- find out the mean position of the pixels of the contour:

  For blurred linear contours:

  – choose μ as a constant value, and estimate the orientations θ_k $(k = 1, \ldots, d)$ through Eq. (3.28);

  – choose μ as a variable value $\mu = \alpha(i-1)$, and estimate the offsets x_{0k} $(k = 1, \ldots, d)$ through Eq. (3.32), for each orientation value.

  For blurred circular contours: choose μ as a variable value $\mu = \alpha(i-1)$, and estimate the radius r_0 by determining the roots of the polynomial function H;

- estimate the spread parameters σ_k $(k = 1, \ldots, d)$ by DIRECT optimization method (see Eq. (3.39) for linear contours) or Nelder-Mead method (see Eq. (3.42) for circular contour);
- obtain a refined estimation of x_{0k} $(k = 1, \ldots, d)$, knowing σ_k values (linear contours, see Eq. (3.34)).

The methods dedicated to straight line estimation and circle retrieval were extended to distorted linear contours and distorted circular contours. For this, a pixel shift term was introduced in the model which is followed by the signal generated on the uniform linear antenna or the circular antenna. In the case of linear contours, an optimization method, based either on gradient [21] or on the combination of DIRECT and spline interpolation [76]. Table 3.1 provides the directions for signal generation, the parameters which characterize the initialization contour and the distortions when either linear or circular contours are expected.

	Straight	Circular
Direction for signal generation	row i	$\mathbf{D}_i$
Initialization parameters	θ, x_0	r, center
Pixel shift	$\Delta x(i)$	$\Delta\rho(i)$

Table 3.1 — Similarities between nearly straight and nearly circular distorted contour estimation

A summary of the estimation nearly rectilinear distorted contour is given as follows:

- Signal generation with constant parameter on linear antenna, using Eq. (3.1);
- Estimation of the parameters of the straight lines that fit each distorted contour (see subsection 3.5.1);
- Distortion estimation for a given curve, estimation of $\mathbf{x}$, applying gradient algorithm to minimize a least squares criterion (see Eq. 3.45).

The optimization method based on gradient or DIRECT combined with spline interpolation yield satisfactory results when the distortions are of low amplitude. In the case of any star-shape contour, with either low amplitude or high amplitude distortions, a method proposed in [61] is preferable. It models the pixel radial shifts as damped sinusoids. A method dedicated to the estimation of the damp factor, the frequency and the phase shift of multiple sinusoids was adapted in [61]. It permits to retrieve the contour distortions with a computational load which is independent from the distortion amplitude, contrary to the optimization methods which were proposed previously. The proposed method for star-shaped contour estimation is summarized as follows:

- Variable speed propagation scheme upon the proposed circular antenna : Estimation of the number of circles by MDL criterion, estimation of the radius of each circle fitting any expected contour (see Eqs. (3.9) and (3.10)) or the axial parameters of the ellipse;

- Estimation of the radial distortions, in polar coordinate system, between any expected contour and the circle or ellipse that fits this contour. In the case of low amplitude distortions, either the gradient method or the combination of DIRECT and spline interpolation may be used to minimize a least-squares criterion. In the case of star-shape contours with possibly large distortions, a damped sinusoid characterization method is adapted to the signals generated on the circular antenna.

Now, the methods presented in this chapter cope with either linear, or star-shape contours. The results presented above while exemplifying the methods for strongly distorted star-shape contours lead to an intuition: this kind of methods could be adapted to hand contour characterization. However, we will show further in this manuscript that, although this intuition is justified, a completely new signal generation method is necessary to characterize hand contours, which are most often non star-shape. This is the purpose of a next chapter of this manuscript.

CHAPTER 4 Novel signature for hand characterization

4.1 Introduction of the chapter

HAND characterization appears to be a necessary and important step in the hand recognition procedure. Several methods have proven successful and have given promising results but they are applied on a reduced base of postures. Thinking in this direction is more essential than ever because existing descriptors based for instance on moments exhibit drawbacks.

From the comments provided in section 1.3, it appears that a new characterization method is now required. It must ensure maximum discrimination between the postures that are very close, it must also ensure the properties of invariance such as rotation, translation and the scale factor. Finally it must guarantee the consistency between the reconstructed image (with the vector or matrix characterization) and the initial image.

With the experience of the GSM team in the field of antenna treatment and the transfer of array processing to image processing using the tools of signal processing (see section 3), we managed to find a new method of characterization, but the questions that arise are as follows:
how could antenna tools processing be adapted to the generation of a discriminative hand signature? how does this method guaranteed the invariance properties? And finally, what are the required preprocessings which permit to respect the conditions of use of this novel signature?

4.2 Signature generation

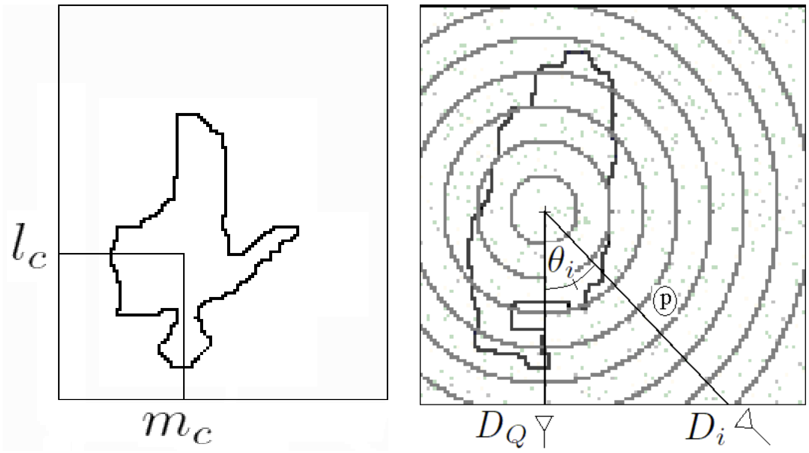
A planar object shape can be characterized through two-dimensional moment invariants, obtained for instance with Hu [53], Zernike [28, 66], or Legendre [40] moments. One-dimensional moment invariants can also be used as signatures to characterize contours, for instance Fourier descriptors [30, 88], which are obtained by Fourier transform of the arclength parametrization, in complex coordinates, of a closed contour. The image scan in [98] provides a contour signature as a matrix involving the contour polar coordinates.

An equivalent descriptor called shape context descriptor is presented in [44] as a compact human pose representation. The processed image is divided into different ranges of radius and angle values. Each range couple compounds a bin. Counting the number of pixels in each bin yields a 2-D histogram. The main drawback of such a descriptor is that it does not provide a 1-pixel precision: it is impossible to distinguish between the pixels of a given bin, so details which are smaller than the bins are skipped. And, the more accurate the description, the smaller the regions, but the higher the computational load and the storage place. On the contrary, we propose a contour signature which offers a resolution of one pixel.

The proposed novel scan is inspired from [98] but also from [61, 77]. In [77] and [61], an image scan is proposed to characterize star-shaped contours. In a system of polar coordinates with adequately chosen pole, a contour is star-shaped if the radial coordinates (ρ) of its pixels are function of their angular coordinates (θ): $\rho = f(\theta)$. In the general case, hand contours are not star-shaped: it is impossible to find a pole for which the relation $\rho = f(\theta)$ holds for all contour pixels. That is why we seek for a characterization method which handles non-star-shaped contours.

The proposed method for contour characterization splits the image into several rings centered on a reference point. The requirements on the location of this reference point are low, contrary to the condition imposed by the method in [61]. With this characterization method, we aim at distinguishing very similar postures with a computational load which is lower than what the generally used Fourier descriptors would require.
The image I^c, denoted by I in the following for convenience, is supposed to have size $N \times N$, and its pixels are referred to, starting from the top left corner of the image, as $I_{l,m}$ (see Fig. 4.1.a). The 1-valued pixels compound the expected contour. The contour pixels are located in a system of polar coordinates with pole $\{l_c, m_c\}$ (see Fig. 4.1.a).

Contrary to the methods proposed in [61], where the center must be chosen in such a way that the contour is star-shaped, the computation of the center coordinates is not essential. For instance, this pole can be the center of mass obtained in the previous section. What we call signature in this thesis is a set of data which characterizes the corresponding contour. The novel signature that we propose in this thesis is based on the generation of signals out of an image. As in [61], a circular array of sensors is associated with the image. The sensor array is supposed to be placed along a circle centered on the pole $\{l_c, m_c\}$. The number of sensors is denoted by Q and one sensor corresponds to one direction for signal generation D_i, which makes an angle θ_i with the vertical axis. See for instance the i^{th} and the Q^{th} sensors in Fig. 4.1.b. The other sensors are not represented for sake of clarity.



(a) (b)

Figure 4.1 — Image and edge model (a); signal generation process (b).

The method proposed in [61] is valid only for contours exhibiting at most one pixel for one direction D_i. We wish to overcome this limitation and characterize non star-shaped contours, because the hand contours considered in this thesis are mostly non star-shaped. To separate the influence of each pixel located along a given direction D_i, we no longer generate one 1-D signal, but a number P of 1-D signals on the antenna. Each signal corresponds to one 'ring' represented on Fig. 4.1.b.

We assume that, for each direction D_i, there is only one pixel in each of the P intervals. P differs from one direction D_i to another. Its maximum theoretical value is, for instance, $\frac{N}{\sqrt{2}}$, if $l_c = N/2$ and $m_c = N/2$. In these conditions also, the value of Q should not exceed $\sqrt{2}\pi N$: it is sufficient to take into account all pixels of a given interval p. So, we generate P signal vectors for each direction D_i. For the p^{th} interval ($p = 1, \ldots, P$) and the direction D_i ($i = 1, \ldots, Q$), the signal component $z_{p,i}$ is computed as follows:

$$z_{p,i} = I_{l_{p,i},m_{p,i}} \sqrt{(l_{p,i} - l_c)^2 + (m_{p,i} - m_c)^2} \tag{4.1}$$

The components $z_{p,i}$ ($p = 1, \ldots, P$, $i = 1, \ldots, Q$) can be grouped into a matrix $\mathbf{Z}$ of size $P \times Q$:

$$\mathbf{Z} = \begin{bmatrix} z_{1,1} & z_{1,2} & \cdot & \cdot & z_{1,Q} \\ z_{2,1} & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ z_{P,1} & \cdot & \cdot & \cdot & z_{P,Q} \end{bmatrix} \tag{4.2}$$

where several $z_{p,i} = 0$

$$\mathbf{Z} = \begin{bmatrix} 0 & z_{1,2} & \cdot & 0 & z_{1,Q} \\ z_{2,1} & \cdot & \cdot & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot & \cdot & 0 \\ z_{P,1} & 0 & \cdot & \cdot & z_{P,Q} \end{bmatrix} \tag{4.3}$$

All columns of $\mathbf{Z}$ should have the same number of rows, so for the directions D_i which cross less than P intervals, 0-valued components are set in $\mathbf{Z}$ for the corresponding indices i. If the width of the intervals is chosen such that there is at most one pixel per direction D_i and per interval, this matrix permits to reconstruct exactly the contour: it contains the radial coordinates of the contour in the system of pole $\{l_c, m_c\}$.

However the purpose of the signature is not obligatorily to reconstruct exactly the contour: it should characterize a contour so that all postures can be distinguished. Also, the signature should be invariant to rotation. To ensure this, the components $z_{p,i}$ of a given interval p are sorted. As a consequence, all non-zero values of the p^{th} row of $\mathbf{Z}$, issued by contour points, are turned as the last components of the p^{th} row. This method differs from the method proposed in [15], where the images were straightened up through several rotations and the maximization of the hand Feret's diameter in the horizontal direction. This process was much more time consuming.

Before getting the image I which is fed to the method of characterization, we apply some adequate preprocessings.

From the initial processed image, we select the smallest subimage containing the expected contour. This subimage is called "enclosing box". The enclosing box is obtained in the following way: the image content is projected onto the left and the bottom sides (it could be also the right and the top sides). We get two signals, $\mathbf{z}^{left}$ and $\mathbf{z}^{bottom}$, from this projection: Their components are obtained as follows: $z_l^{left} = \sum_{m=1}^{N} I_{l,m}$ $l = 1, \cdots, N$ and $z_m^{bottom} = \sum_{l=1}^{N} I_{l,m}$ $m = 1, \cdots, N$. For each signal, a non-zero section indicates the presence of the expected feature. The l and m indices of the non-zeros sections yield a box enclosing the contour. Extracting this box reduces the computational load of the signature generation.

Eventually, through the following remarks (•) we can assess that the rows of matrix $\mathbf{Z}$ compose a complete set of invariant features:

- They describe entirely the hand contour: the rows of matrix $\mathbf{Z}$ compose a complete set of invariant features when only couple (p, i) correspond to only one pixel.
  Fig. 4.2 illustrates this by showing a segmented hand posture (see Fig. 4.2(a)),

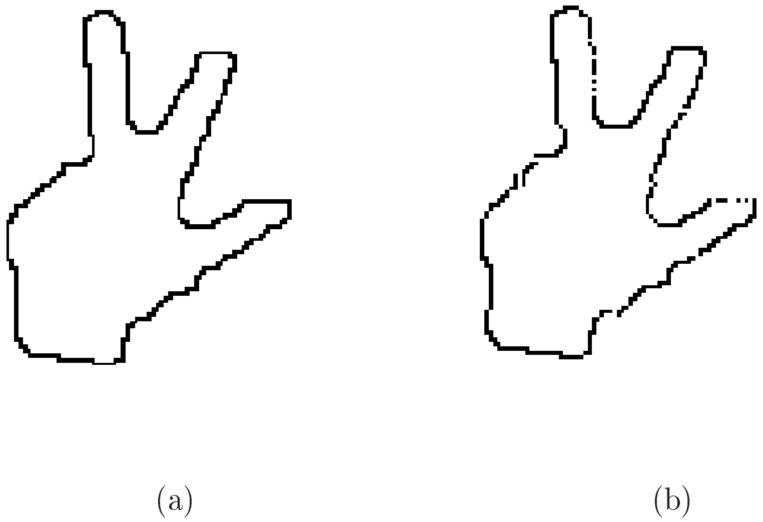


Figure 4.2 — Segmented contour (a); contour reconstructed from the signature **Z** (b).

and the contour which is reconstructed out of its signature **Z** (see Fig. 4.2(b)).

- They are invariant to translation: the box which encloses the contour is blindly estimated, whatever the hand position in the initial image.

- They are invariant to scaling: whatever the size of the subimage (small number of pixels if the camera is far from the hand, large number of pixels if the camera is near to the hand), the number of intervals for the radial coordinate values P is always the same. Also, the number of directions for signal generation is always the same. As a consequence, the size of matrix Z will be constant, whether the user's hand is near to or far from the camera. This makes the method invariant to scaling. Hence, the signature depends on the shape of the hand, not on its size.

- They are invariant to rotation: whatever the initial orientation of the hand, straightening up the hand contour makes the proposed method invariant to rotation.

These invariance properties permit to use the proposed contour signature (matrix **Z**) as for hand posture classification purpose.

4.2.1 Dimensionality reduction and Bayesian distance computation

Let's consider H classes of hand postures. For the purpose of hand posture classification, Euclidean and Bayesian distances are used in [15]. We will compare the results obtained with Euclidean and Bayesian distances. We vectorize any matrix $\mathbf{Z}$ characterizing a posture into a $P.Q$ vector $\mathbf{x}$. For each class h, a subset of hand photographs is available. The H subsets compose the learning set. This set was created by an expert who knows exactly what position his fingers should have to fit each posture in Fig. 2.6. Let $\mathbf{X}_h$ be the matrix whose columns are the vectors $\mathbf{x}_{n_h}$, $n_h = 1, \ldots, M_h$ obtained from the images belonging to class h. It is obvious from Fig. 4.1.b that, the higher P and Q, the more details we keep in the signature $\mathbf{Z}$, and the more accurate the hand posture classification method involving this signature.

However, for large values of P and Q, $\mathbf{X}_h$ exhibits a large number of rows, and it is a sparse matrix. The principles of posture classification are as follows: a test set is created from persons who are not the expert. We aim at associating a label with any image chosen from the test set. This label is one of the 11 postures presented in section 2.3. To improve the recognition rate with respect to the work presented in [15], we propose in the following to reduce the number of candidates for a posture and, in subsection 4.2.1, to reduce the dimensionality of matrix $\mathbf{X}_h$ obtained from the learning set.

For a classification purpose, two main distances may be chosen: the Euclidean distance and the Bayesian (Mahalanobis) distance. Let $\mathbf{x}^c_{n_h}$, $n_h = 1, \ldots, M_h$ denote the columns of $\mathbf{X}^c_h$. The mean invariant vector is computed as $\boldsymbol{\mu}_h = \frac{1}{M_h} \sum_{n_h=1}^{M_h} \mathbf{x}^c_{n_h}$, and the covariance matrix is computed as $\boldsymbol{\Lambda}_h = \frac{1}{M_h} \sum_{n_h=1}^{M_h} (\mathbf{x}^c_{n_h} - \boldsymbol{\mu}_h)(\mathbf{x}^c_{n_h} - \boldsymbol{\mu}_h)^T$, for each class $h = 1, \ldots, H$. Even if there are small variations from one posture provided by the expert to another, these variations are smoothed through the computation of the mean invariant vector $\boldsymbol{\mu}_h$. Any image coming from the test set and characterized by vector $\mathbf{x}$ is classified by minimizing the Mahalanobis distance applied to the compressed vector $\mathbf{U}_h^T\mathbf{x}$:

$$\mathcal{D}_m = (\mathbf{U}_h^T\mathbf{x} - \boldsymbol{\mu}_h)^T \boldsymbol{\Lambda}_h^{-1} (\mathbf{U}_h^T\mathbf{x} - \boldsymbol{\mu}_h) \tag{4.4}$$

Computing the Bayesian distance involves, as shown in Eq. (4.4), the inversion of the covariance matrix $\boldsymbol{\Lambda}_h$. This is not the case for the Euclidean distance which is then easier to implement than the Bayesian distance, but the Bayesian distance usually provides better classification results, which has been verified in the frame of hand posture recognition in [34].
Consequently, we propose to use the Bayesian distance. To enable the inversion of matrix $\boldsymbol{\Lambda}_h$, and thereby the computation of this distance, $\boldsymbol{\Lambda}_h$ should not exhibit a too large dimension. That is why we perform dimensionality reduction of the data, with principal component analysis (PCA).

Let K ($K < P.Q$) be the number of dominant singular values in $\mathbf{X}_h$. Let $\mathbf{U}_h$ be the matrix whose columns are the K singular vectors associated with the K largest singular values of $\mathbf{X}_h$. Each singular vector corresponds to a mode of variation of the considered hand posture of class h, and its corresponding eigenvalue is related to the variance specified by the eigenvector.

In [81], such a data compression is also performed on human motion descriptors. In [81], each singular vector reflects a natural mode of variation of human gait. In our case each singular vector reflects a natural mode of variation of presenting the hand in the desired posture in front of the camera. The compressed version of the data is obtained by: $\mathbf{X}_h^c = \mathbf{U}_h^T \mathbf{X}_h$, where ^{T} denotes transpose. With this compressed version of the data, we obtain a lower-dimensional representation of reference hand postures which is more suitable to describe any test posture: in [81], each dimension on the PCA space describes a natural mode of variation of human motion, in the case of hand posture, each dimension describes a natural mode of variation of how the user presents its hand in front of the camera.

Dimensionality reduction permits to reduce the computational load dedicated to matrix inversion in Eq. (4.4): matrix $\mathbf{\Lambda}_h$ was computed from the compressed data and has low $K \times K$ dimensionality. This also prevents from inverting an ill-conditioned matrix. For sake of comparaison, the proposed signature can be also exploited with Euclidian distance, computed as follows: $||\mathbf{U}_h^T \mathbf{x} - \boldsymbol{\mu}_h||$, where $||.||$ denotes Frobenius norm.

4.3 Pre-selection of best posture candidates

Through a careful look at the dictionary of posture (see Fig. 2.6), we can distinguish two large categories of postures. To characterize these categories, we introduce a isometric rate, denoted by S, which involves the geometric hand criterion computed from I^f and the length of the hand contour, computed from I^c. S is the hand contour length divided by the hand surface. In practice, we compute the isometric rate as $S = \frac{hand's\ perimeter^2}{hand's\ area\ \times 4 \times \pi}$. Postures 2, 3, 7, 8, 9 and 11 exhibit a high sphericity criterion, and postures 1, 4, 5, 6, and 10 a low isometric rate.

Our purpose is then to pre-select one of these two large categories of postures, and to look for the reference posture which is the closest to the test image posture inside of this category. For this, we compute the distance $\mathcal{D}_m$ of Eq. (4.4) with respect to a low number of reference postures, which are pre-selected from the dictionary by considering the isometric rate.

The criterion S is computed for all images of each class in the learning set. Then we choose the following criterion: $|S_t - S_h|$ where S_t is the isometric rate for the test image and S_h the mean isometric rate for all images of class h in the learning set. We select the 6 classes (about half of the total number of reference postures) which yield the minimum criterion value. They compose a new dictionary with a reduced number of candidates, and distance $\mathcal{D}_m$ of Eq. (4.4) is computed only six times to perform classification.

4.4 Conclusion of the chapter

We propose a novel signature for the characterization of hand postures. This signature is made of several 1-D signals. Each signal contains radial coordinates of the pixels in an image region which has the shape of a ring. This signature permits to reconstruct the corresponding contour with a precision of one pixel.

By applying some preprocessing, we ensure that this signature forms a complete set of features which are invariant to translation, scaling and rotation. This makes this signature fit for hand posture recognition, we facilitate the classification step with dimensionality reduction by PCA because we reduce the size of characteristic matrix to K x K.

The new matrix can be used to improve classification and learning steps. In the learning step we should represent all user(adult, child,male, female, left hand, right hand and color hand) to calculate the referent matrix which can be used in classification.

CHAPTER 5 Optical Flow

5.1 Introduction of the chapter

WHILE detecting hand contours, the diversity of users is one of the constraints to solve. Indeed, the detection and recognition must be carried out for all hands (white or colored, with or without gloves), and we found that most of the methods used for the detection step are based on the skin color. In [91] for instance, the authors use green-colored gloves to detect easily a moving hand. In [99], Soriano *et al.* propose a dynamic skin color model, for a segmentation purpose. Their method copes with changes in illumination. However, their method still relies on relevant color properties of the skin. No result is presented concerning dark skins or hands wearing gloves. In [80], the authors modelled their object colors as a Gaussian mixture and recursively adapted the mean, covariance and prior probabilities of each Gaussian cluster. In [112], a set of relevant grey level values are selected from chromatic histograms to segment faces. To summarize these approaches, either the user affords a prior knowledge of the scene and the target or he assumes that the hand is white.

On the contrary, we aim at detecting the contour of a hand, whatever its color. Thinking in this direction leads us to look for other methods that allow us to solve this problem. A promising method for the detection of hands, whatever their color, consists in adapting optical flow (as used in Fig. 5.1). It appears to us as a reliable technique especially because we combine static and dynamic hand recognition.
Therefore, questions arise while implementing and using this method. They are considered successively in sections 5.2, 5.4, 5.3 of this chapter: what are the conditions and assumptions required to use the optical flow algorithm? How to adapt the optical flow for the recognition of hand postures ? What is the efficacy of this technique for the determination of hand movements in any scene ?

5.2 Definition and conditions of use

Optical flow is the pattern of motion, as it appears to a camera, of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer (an eye or a camera) and the scene. The concept of optical flow was introduced by the

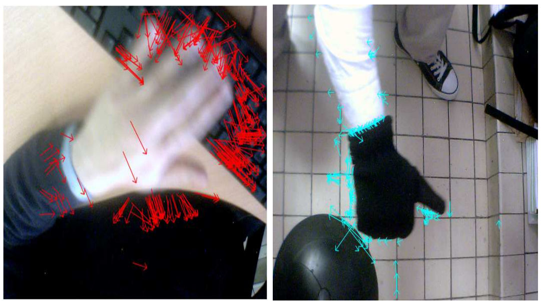


Figure 5.1 — Example of motion detection with optical flow.

American psychologist James J. Gibson in the 1940s to describe the visual stimulus provided to animals moving through the world. As already mentioned, optical flow may often want to assess motion between two frames (or a sequence of frames) without any other prior knowledge about the content of those frames. Typically, the motion itself is what indicates that something interesting is going on.

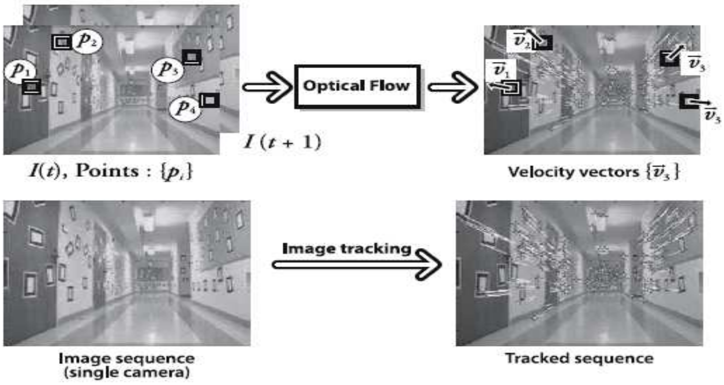


Figure 5.2 — Optical Flow

Movement, characterized by optical flow, has been exploited by roboticists, who use optical flow techniques (including motion detection, luminance, motion encoding, and stereo disparity measurement) for image processing and control of navigation.
As already mentioned, optical flow may often want to assess motion between two frames (or a sequence of frames) without any other prior knowledge about the content of those frames. A result that can be obtained by optical flow is illustrated in Figure 5.2.

The principles of optical flow are as follows: if color images are considered, a conversion to one channel is done. For instance, we can select the Cr component of the $YCbCr$ representation, but this is valid only when white hand are considred. We also can retain only the luminance component from the HSL (Hue, Saturation, Lightness) representation of the RGB image. We can associate some kind of velocity with each pixel in the frame or, equivalently, some displacement that represents the distance a pixel has moved between the previous frame and the current frame. It associates a velocity with every pixel in an image. There exist two approaches to calculate the

optical flow.
The first approach is the dense technique which tries to match large windows around each pixel of an image to another, as the algorithm of Horn and Schunk [50]. This algorithm was developed in 1981; it puts aside the hypothesis of constancy of brightness by minimizing the regularized Laplacian of optical flow velocity components. This turns as a valid one the hypothesis of smoothness constraint on the velocities. Also, there exists a whole class of similar algorithms in which the image is divided into small regions called blocks [11, 55].

These blocks are generally square and may overlap. These algorithms attempt to divide the two previous and current images in blocks and then calculate the movement of these blocks. Such algorithms are of great interest in many video compression techniques and in computer vision. Black and Anadan have created dense optical flow techniques [12, 13] that are often used in movie production, where, for the sake of visual quality, the movie studio is willing to study in detail the flow information, in practice the movement of the actors or objects. The block-matching algorithms operate on aggregates of pixels, not on individual pixels.

If the overlap between blocks is very important, the returned images of "flow" are usually of a lower resolution than the input images. Algorithms of this approach have superior quality but are slow and cannot be applied in real time and cannot resolve the case of large displacements. In practice, calculating dense optical flow is not easy. Let's consider the motion of a white sheet of paper. Many of the white pixels in the previous frame will simply remain white in the next. Only the edges may change, and even then only those orthogonal to the direction of motion. Hence the idea of creating a sparse optical flow, developed originally in [73].
The second approach is a popular sparse tracking technique, Lucas-Kanade (LK) optical flow. This version of optical flow relies on some means of specifying beforehand the subset of points that are to be tracked. If these points have certain desirable properties, such as the "corners", then the tracking will be relatively robust and reliable. The LK algorithm [73], as originally proposed in 1981, was an attempt to produce dense results. However, because the method is easily applied to a subset of the points in the input image, it has become an important sparse technique. The LK algorithm can be applied in a sparse context because it relies only on local information that is derived from some small windows surrounding each of the points of interest. This contrasts with the intrinsically global nature of the Horn and Schunck algorithm.

The basic idea of the Lucas-Kanade algorithm is based on three assumptions (see Fig. 5.3):
• *Brightness constancy*: A pixel from the image of an object in the scene does not change in appearance as it (possibly) moves from frame to frame. For grayscale images (LK can also be done in color), this means we assume that the grey level of a pixel does not change as it is tracked from frame to frame.

• *Temporal persistence or ”small movements”*: The image motion of a surface patch changes slowly in time. In practice, this means the temporal increments are fast enough, relative to the scale of motion in the video sequence, to prevent the object from moving much from frame to frame.
• *Spatial coherence*: Neighboring points in a scene belong to the same surface, have similar motion, and project to nearby points on the image plane.

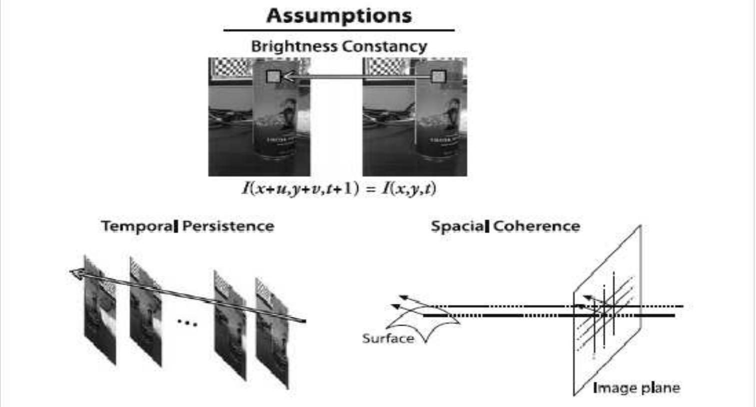


Figure 5.3 — Assumptions behind Lucas-Kanade optical flow

As mentioned above, the disadvantage of using small local windows in Lucas-Kanade approch is that large motions can move points outside of the local window and thus become impossible for the algorithm to find. Indeed large and non-coherent motions are often observed in practice. The key idea in the Lucas-Kanade approach is to avoid this problem, by tracking first over larger spatial scales, by using an image pyramid and then by refining the initial motion velocity assumptions by working its way down the levels of the image pyramid until it arrives at the raw image pixels.
Hence, this problem led to the development of the "pyramidal" LK algorithm, which tracks an object starting from the highest level of an image pyramid (lower detail resolution) and working down to lower levels (finer detail resolution). Thus we minimize the violations of our motion assumptions and we can track faster and longer motions. This more elaborated function is known as "pyramidal Lucas-Kanade" optical flow and is illustrated in Figure 5.4. Hence, tracking along the resolution levels as downhill along pyramids allows large motions to be characterized by local windows.

In the following section, we detail the initial purpose of optical flow, which is originally meant to characterize movements.

5.3 Optical Flow: an algorithm originally dedicate to trajectory detection

The first and most common application of optical flow is to track a target between two frames. Motion estimation and video compression have been the most common

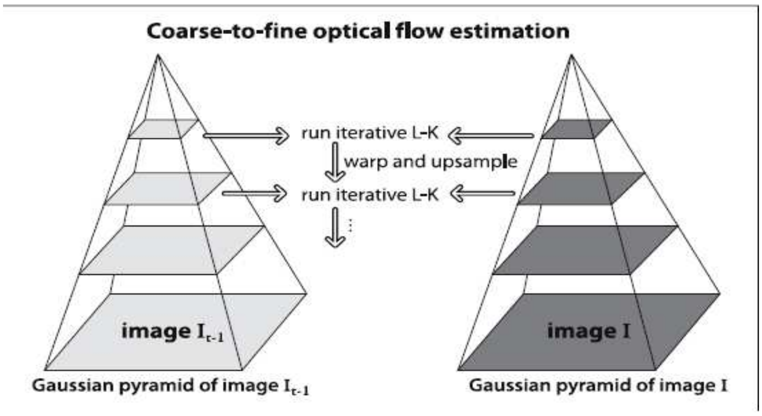

Coarse-to-fine optical flow estimation
run iterative L-K
warp and upsample
run iterative L-K
image I_{t-1}
image I
Gaussian pyramid of image I_{t-1}
Gaussian pyramid of image I

Figure 5.4 — Pyramid Lucas-Kanade optical flow

application fields of optical flow. A direct application of optical flow consists in tracking a hand in a video sequence. Starting from the moving points, as represented in Fig. 5.1, which are essentially part of the hand contours, but may also be outliers, we aim at finding some representative points of the hand. For this, we first remove outliers: we suppress the points which contain at least one extreme coordinate: these outliers are the nearest to the image corners. The center of mass of the remaining points is considered as the most representative to locate the hand.
Therefore, studying the overall trajectory of the hand is equivalent to studying the trajectory of this representative point. However, we notice that this method is not sufficient to characterize the hand shape, and thereby the hand posture itself. The moving points provided by optical flow compose part of the hand contour points. A method must be found to get a continuous hand contour. We address this issue in section 5.4.

5.4 Optical flow adapted as a contour detection method

A promising method for the detection of hands, whatever their color, consists in adapting optical flow. Indeed, as it is based on movement properties and not on intrinsic grey level values, optical flow may characterize indistinctly white-skin hands and colored-skin hands. Moreover, optical flow attracts the interest of the image processing community, showing its adaptability. It has been recently improved to cope with dense optical flow fields by integrating rich descriptors [24], and to face discontinuities on motion boundaries [47]. We wish to adapt this method to segmentation purposes. Our idea is to take profit from the information provided by optical flow to isolate a target which is moving in the scene, namely the hand.

There are many kinds of local features that one can track. If we pick a point on a large blank wall then it won't be easy to find that same point in the next frame of a

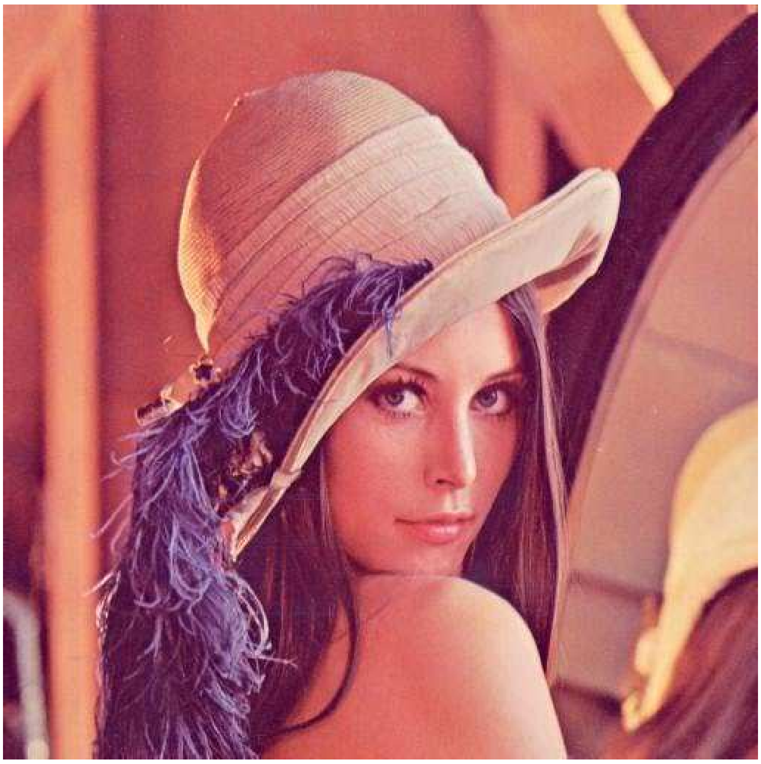


Figure 5.5 — Selection of good features without prior knowledge: 'Lena' image under study

video. If all points on the wall are identical or even very similar, then we won't have much luck tracking that point in subsequent frames. On the other hand, if we choose a point that is unique then we have a good chance of finding that point again. In practice, the point or feature we select should be unique, or nearly unique, and should be parameterizable in such a way that it can be compared to other points in another image. Or if we consider that the hand color and the background color are different, we are certain that the hand contour by itself represents good points to track, and this feature limits properly the region of the hand. This permits to highlight the main constraint on the applicability of optical flow: it can be used as detection method if the background color is different from that of the hand.

In our acquisition conditions, a hand may cross the whole acquired scene rather rapidly, hence, we adapt a pyramidal version [18] of Lucas-Kanade optical flow. This pyramidal version includes a multi-scale strategy, which permits to handle larger displacements, while keeping the reduced computational load of Lucas-Kanade sparse method [73].

If strong derivatives are observed in two orthogonal directions then we can hope that this point is more likely to be unique. For this reason, many trackable features are called corners. Intuitively, corners are the points that contain enough information to be picked out from one frame to the next. The most commonly used definition of a corner was provided by Harris [48]. This definition relies on the matrix of the second-order

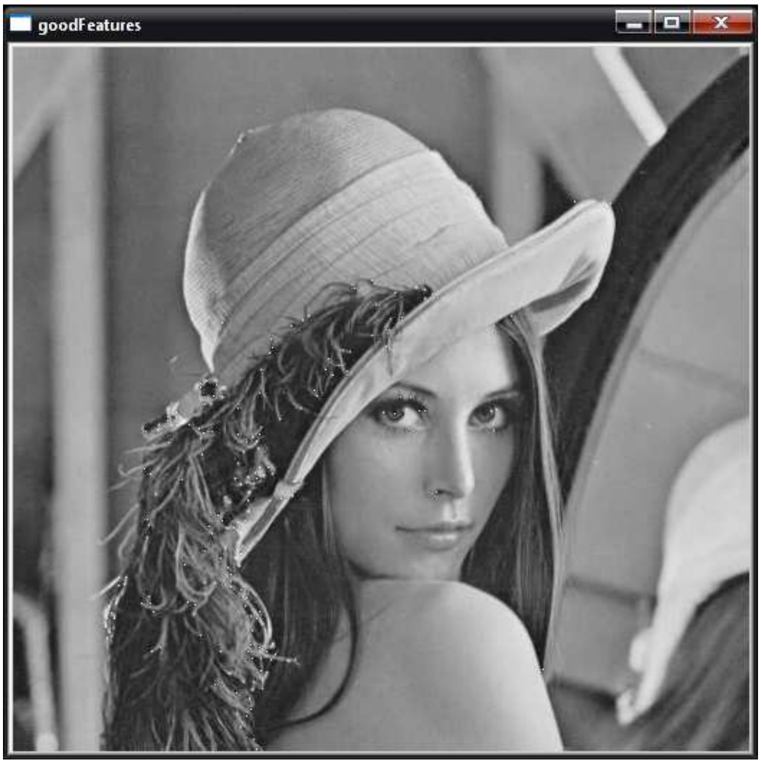


Figure 5.6 — Selection of good features without prior knowledge: detected corners

derivatives of the image intensities. Corners, according to Harris definition, are places in the image where the autocorrelation matrix of the second derivatives has two large eigenvalues. In essence this means that there are texture properties (or edges) going in at least two separate directions centered around such a point, just as real corners have at least two edges meeting in a point.

It was later found by Shi and Tomasi [97] that good corners were selected as long as the smaller of the two eigenvalues was greater than a minimum threshold. See for instance the corners that were obtained, in Fig. 5.6, from the 'Lena' picture (Fig. 5.5).

If we have a prior knowledge on the location of the expected corners, we can delimitate a search box to an area defined beforehand, called a mask, which can limit the region of good features to track. This is illustrated in Fig. 5.8, which was obtained with the mask presented in Fig. 5.7.

In the context of hand posture characterization, the region of interest can be selected through least square ellipse fitting. The implementation of this algorithm will be detailed further in the manuscript.

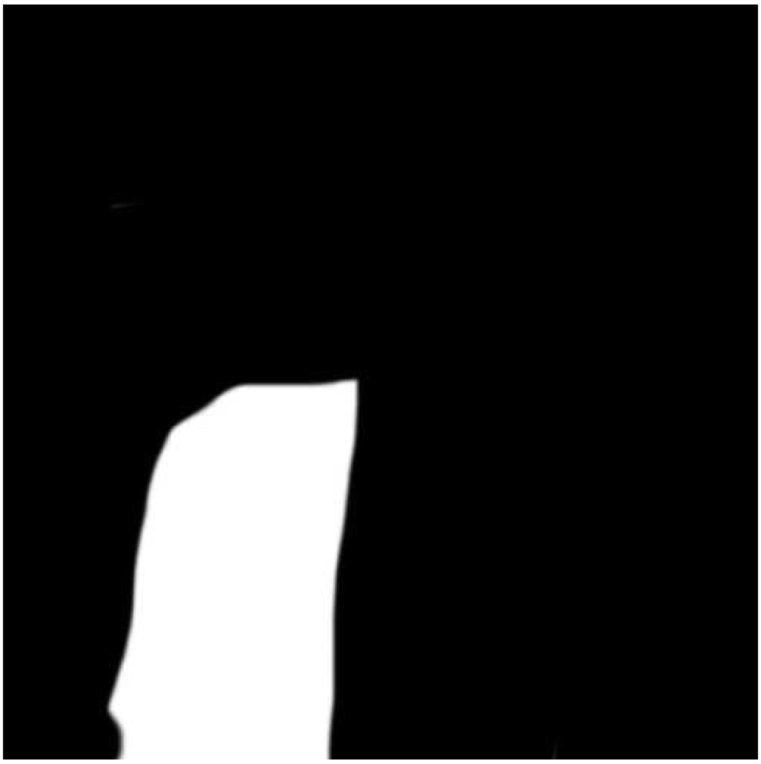


Figure 5.7 — Selection of good features with prior knowledge: mask selecting the region of interest

5.5 Conclusion of the chapter

In this chapter we present a method used to track movements in a video sequence or between two successive frames, and we try to adapt it to hand detection. Respecting the various constraints in this work, this adaptation exhibits huge advantages.
In section 5.1, we remind the main goal of optical flow, and the problematics that arise while applying this method. In section 5.2, we present optical flow in a historical context. We present the different optical flow techniques and their conditions of use, insisting on the version from Lucas-Kanade [73], which is the one that we have chosen for our hand detection application. In Section 5.3 we state the essential role of optical flow for tracking a moving object. We explain briefly how it can be adapted to the localization of the hand. Optical flow thereby characterizes dynamic gestures in a video sequence. In section 5.4, we discuss a novel way to use optical flow, as we adapt it to the detection of hand contours. Optical flow thereby characterizes static gestures, also called postures, in a series of frames extracted from a video sequence.

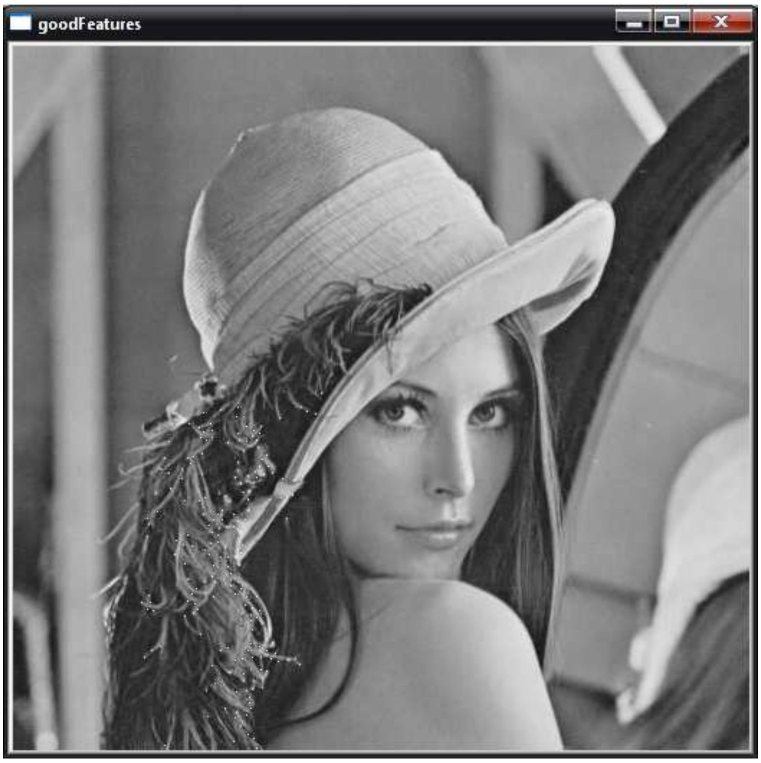


Figure 5.8 — Selection of good features with mask: resulting detected corners, appearing only in the region selected by the mask

CHAPTER 6

Overall algorithm, results and discussion

6.1 Introduction of the chapter

IN this chapter we propose the hole hand posture recognition method, which overcomes the main drawbacks of existing methods [19, 53, 115]: our method should be valid whatever the hand color; for this, we adapt optical flow, which is originally meant to detect moving objects, to improve hand detection. Also, we wish to improve recognition rate, especially for very similar postures, while keeping the computational load and the memory requirements as low as possible; for this we have proposed a novel approach for hand posture characterization in 4.

Our overall approach is based on the optical flow as a detector, and signature generation as characterization, combined with the reduction of matrix characteristic by PCA, but also to the reduction of dictionary of gestures with the geometric criterion (isometric rate).

To validate this approach a comparison with other existing approaches in the literature is needed, but the questions that arise, what are the different preprocessing used to improve our approach? how is it organized this algorithm? is that we have good recognition rate compared to other methods? and eventually the constraints imposed by the industrial context are resolved?

6.2 Preprocessing and proposed algorithm

We process images of size 320×240 with a 2-core processor @3.2 GHz, using Matlab®. This result section falls into two subsections: we first present the results of hand contour segmentation with optical flow; and secondly we present the results of hand posture recognition obtained with Bayesian distance from the images containing the hand contours.

6.2.1 Hand image acquisition setup

This setup contains a CMOS camera (see Fig. 6.1). It has the size of a webcam, and could further be integrated in an embedded system. The camera is placed over the desk surface, it axis is orthogonal to the desk surface. Wide angle optics (90°) are used so that the field of vision is wide enough. The acquisition format can be either CIF, or VGA. The video stream is transmitted to the computer by a USB connection in RGB format. The user can then interact with his computer, and follow the evolution of his experiment directly on the screen.

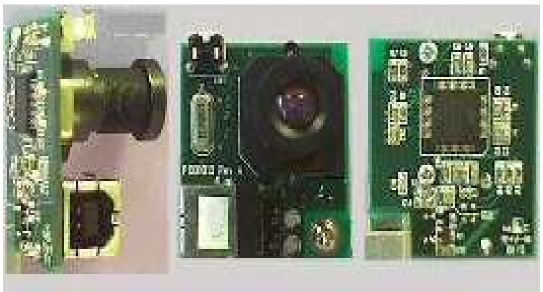


Figure 6.1 — Camera

6.2.2 Preprocessing and algorithm

As we will show in the result section, only the hand contour is retrieved by optical flow. Thus, this result is not used as final hand contour. It is however essential for the selection of a region of interest, which is the first preprocessing applied to the processed image: Let N_{OF} be the number of moving points of interest, retrieved by optical flow, from two frames: one obtained at time t, the other at time $t' > t$. The coordinates of these points are denoted by $\{(x_o, y_o),\ o = 1, \ldots, N_{OF}\}$.

The selection of a region of interest (ROI) is based on ellipse least-squares fitting [42]. Because of the sensitivity of least-squares fitting methods, and to ensure the robustness of the ROI selection, the moving points of interest which include an extreme (minimal or a maximal) coordinate value are removed. Let I^p denote the image containing the remaining moving points.

Firstly, a rather large ROI is extracted. Indeed the ellipse might not include the whole hand, so we choose as ROI a rectangle which is somehow larger than the rectangle which strictly includes the ellipse.

The second preprocessing is hand surface segmentation: firstly, we compute the center of mass of the pixels of interest; secondly, we deduce the hand pixel grey level distribution in each RGB band from the region next to the center of mass; thirdly, according to this distribution, we perform histogram threshold to each RGB band of the ROI. The combination of each threshold image provides a binary image. The binary image obtained at this point, denoted by I^{Th}, contains the hand surface filled with 1-valued pixels and noise, that is, 1-valued pixels randomly distributed in the image.

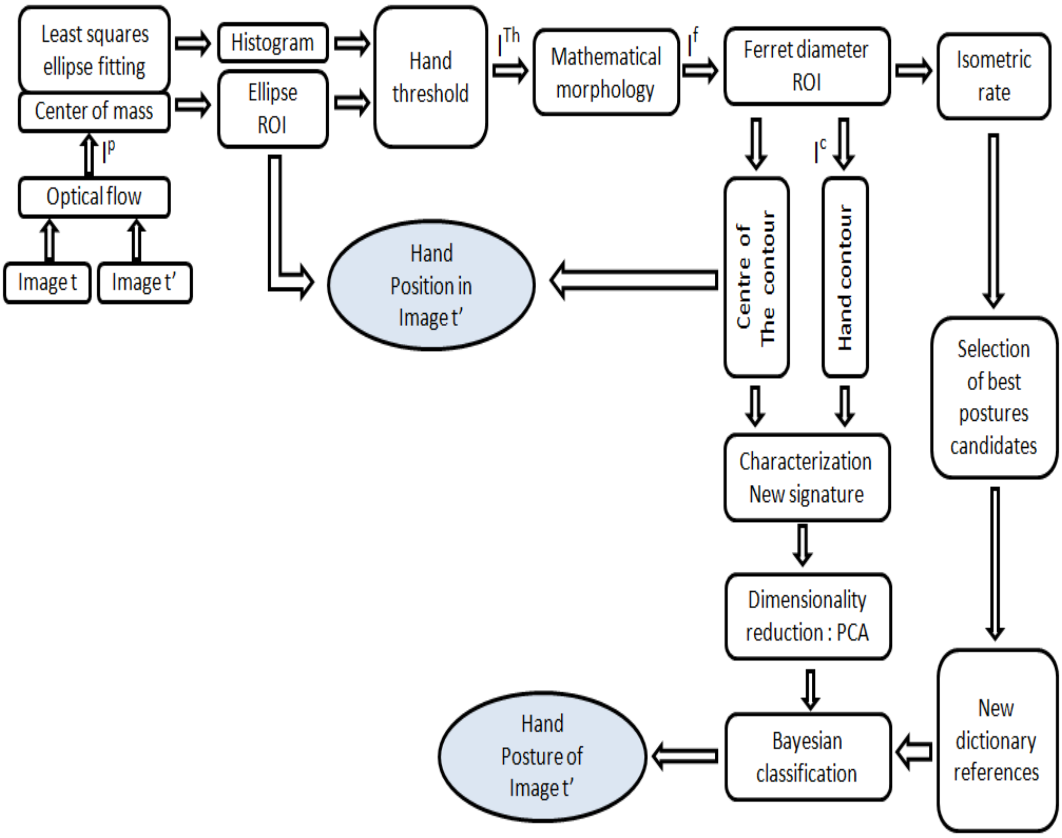


Figure 6.2 — Improved algorithm for hand gesture recognition

The third preprocessing consists in removing isolated pixels and filling out holes. First, we select the largest set of connexe pixels, assuming that this object is the hand. Then, we remove the pixels which are connected to the hand but unexpected with morphological filtering operations -erosions and dilations [115]. These mathematical morphology operations remove the possibly remaining unexpected pixels from the background. This third preprocessing turns the whole algorithm robust to variations in illumination and inclusion of unexpected objects in the background. We then select once again a region of interest: the smallest square subimage containing the whole hand. The number of rows or columns of this image is $\max(\mathrm{FD_h}, \mathrm{FD_v})$ where FD_h and FD_v are the horizontal and vertical Feret diameters of the hand. Extracting this ROI, independently of course from its location in the processed image, ensures the invariance to translation and scaling. We get an image I^f which is supposed to contain only a filled hand.

The fourth preprocessing consists in retrieving the hand contour, with a linear

'roberts' filter. This yields an image I^c where the hand contour consists in 1-valued pixels, over a background of 0-valued pixels. This image will be used to compute a contour signature.
The preprocessing operations presented in this subsection permit to focus on a region of interest and isolate the hand contour, but also to ensure invariance proprieties of the characterization method which is presented in section 4.2.

6.3 Results and Discussion

Adapting optical flow exhibits advantages but also requirements on the experimental conditions and specific preprocessings. The required experimental conditions for which the optical flow works properly are as follows: the hand whose posture must be recognized should be moving between two frames of the database, the background color must be different from the hand color, and the variations of luminosity should be as low as possible. This may be the case for instance if all images are subsequent frames of a video sequence where the user's hand is moving. However, optical flow may still yield poor results if the luminosity varies too much between frames.
A test permits to get rid of the images which are not in compliance with these requirements: it involves the ellipse which is supposed to fit the moving points of interest. The image is skipped by the program and not considered for posture recognition in the following cases: if one axis of the fitting ellipse is larger than the image size, or if the large axis is larger than 3 times the small axis. The consequence for the user of the hand posture recognition method is that he may wait a bit longer for the recognition result, until the luminosity does not vary too much, or until his hand, while exhibiting a novel posture, is moving fast enough for optical flow to consider it as a moving object.

6.3.1 Performance assessment on colored hands

The main advantages of the proposed method, which adapts optical flow [17] instead of the classically used YC_bC_r mapping, are as follows: it handles the case of colored hands, such as those wearing gloves of any color, or hands of coloured people. This is a great advantage respect to the existing method which are supposed to fail as soon as the hand surface cannot be distinguished from the background in the C_b component.
Figures 6.3 and 6.4 show the results obtained by optical flow on a white and a black hand. It consists in pixels which are about to move between the current and the next frame. These pixels of interest match part of the the hand contour pixels.

As shown in Figs. 6.3 and 6.4, the optical flow method provides a set of points, among the moving points of the scene. As a sparse version of optical flow was chosen, these points are mainly focused on the hand contour.

In Fig. 6.5 we exemplify the steps of the proposed method, on a hand posture of type '3'. Fig. 6.5 shows how the moving points provided by optical flow contribute to the image threshold: in Fig. 6.5(b) we show the moving points detected by optical

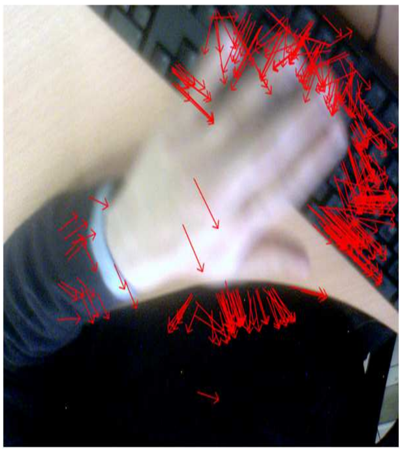


Figure 6.3 — Motion detection with optical flow: white hand.

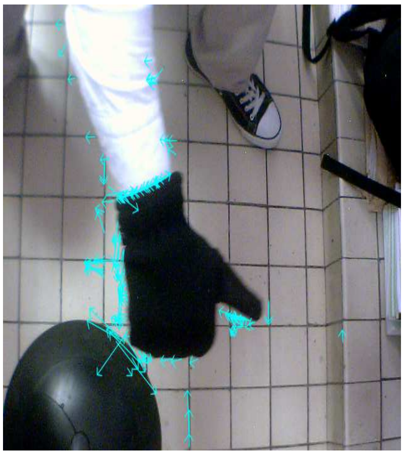


Figure 6.4 — Motion detection with optical flow: colored hand.

flow, their center of mass, and the fitting ellipse. The hand grey level distribution is computed around the center of mass, and its knowledge permits to apply a threshold and obtain the image I^{Th} of Fig. 6.5c).
In Fig. 6.6 we exemplify the method in the same way, with a hand wearing a black glove.

The results obtained on these two hands show the ability of the proposed method to handle white, but also colored hands. The preprocessings permit to remove the undesired pixels which are present in the threshold image I^{Th} (see Fig. 6.5c) and (Fig. 6.6c)).
To exemplify the proposed method for hand contour segmentation on more examples, including all postures for both white and colored hands, a website presents the image I containing the hand contour obtained from eleven cases -one for each posture type-, for a white and a colored hand [1].

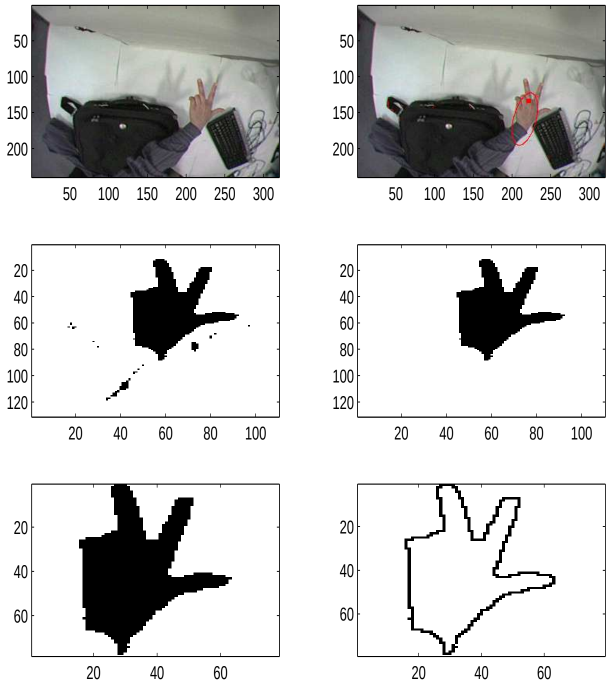


Figure 6.5 — White hand, Steps of the proposed method. (Read from left to right. First row: processed image; moving points, fitting ellipse, and center of mass. Second row: threshold image I^{Th} in the ROI defined from the fitting ellipse; result obtained after mathematical morphology operations. Third row: I^f -square ROI whose height is the maximum Ferret diameter of the hand; I^c, obtained from Roberts linear filtering, and containing the expected hand contour).

6.3.2 Statistical of posture recognition performance

In this subsection, we present a statistical study involving a database of hand posture images. We study the performance of hand posture recognition of the proposed method. We remind that it includes optical flow for hand contour detection. This turns the method adequate for colored hands, but we chose a database of white hands to enable the comparison with existing methods.
To generate the signature $\mathbf{Z}$ whose components are $z_{p,i}$, with $p = 1, \ldots, P$, and $i = 1, \ldots, Q$ (see Eq. (4.1)), a value $P = 24$ levels is large enough to get an exclusive signature for each posture and small enough to get a reasonable computational load. To ensure the invariance to scaling, the number Q of directions depends only on the maximum size of the enclosing box. To perform dimensionality reduction we chose $K = 12$, that is, the size of the posture dictionary +1. This value yield the best results, which was observed empirically.

We compare the proposed method with two comparative methods: The first method combines Gabor filter, PCA, and SVM (support vector machine) [54]. The second comparative method relies on Fourier descriptors [19, 34]. The third comparative method relies on the same process for signature generation [15, 16], but differs in the obtention of the binary image I which is used as an input for the computation of the contour signature: in [15, 16], this binary image is obtained mainly through a

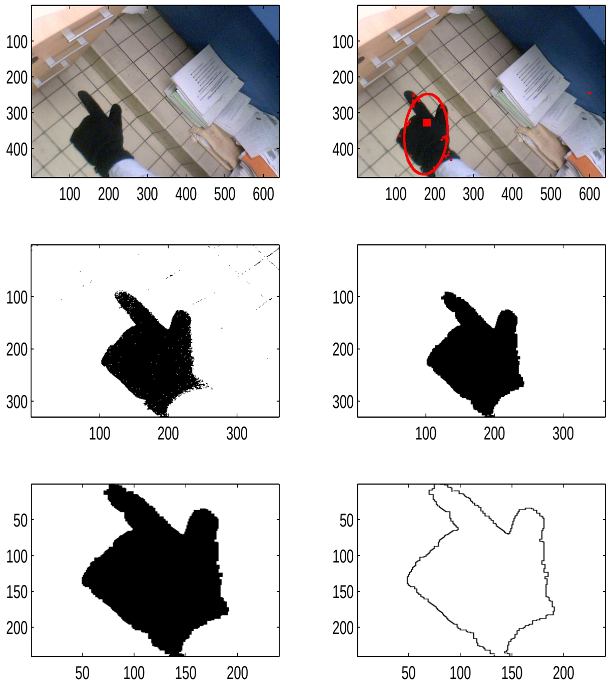


Figure 6.6 — Colored hand, steps of the proposed method. (Read from left to right. First row: processed image; moving points, fitting ellipse, and center of mass. Second row: threshold image I^{Th} in the ROI defined from the fitting ellipse; result obtained after mathematical morphology operations. Third row: I^f -square ROI whose height is the maximum Ferret diameter of the hand; I^c, obtained from Roberts linear filtering, and containing the expected hand contour).

YC_bC_r mapping and a threshold applied to the C_b component. In [16], PCA is already used to reduce the dimensionality of the data.

In Table 6.1, we present the results obtained with YC_bC_r mapping and Fourier coefficients as invariant characteristics. This table shows that Fourier descriptors encounter difficulties with postures 4 (60.8%), 8 (64.8%), and 10 (74.4%). This is due to the unability of Fourier coefficients to preserve details: contours are smoothed, and subtle differences such as the presence of one supplementary finger as occurs between posture 4 and posture 5, and between posture 8 and posture 9, are not detected when Fourier coefficients are used. On the contrary, our method based on the proposed signature generation technique offers a 1-pixel resolution, and does not encounter such problems.

In Table 6.2, we present the confusion matrix of the comparative method based on YC_bC_r mapping [16] and using the signature generation process presented in [17]. It shows that it exhibits good results, except that: posture 4 is recognized as 5 in 11.3 % of the cases, posture 8 is recognized as posture 9 in 25.6 % of the cases; posture 5 as 4 in 5.5 % of the cases.

The confusion matrix obtained with the proposed method [17] is presented in Table 6.3.

	‘1’	‘2’	‘3’	‘4’	‘5’	‘6’	‘7’	‘8’	‘9’	‘10’	‘11’
1	86.6	0	0	0	0	0	0	0	0	0	0
2	0	90.8	0.4	0.4	0.2	0.2	0.1	0	1.7	0.1	0.1
3	0	0.7	96.4	0.5	0.4	1	0.4	0	0.7	0.1	3.3
4	5.5	0	0	60.8	0	0.1	0.4	0	0	0	0
5	2.9	1.8	0.5	35.9	97.8	0.9	7.8	3.2	4.9	20.2	0.1
6	4.6	0.1	0	0.1	0.3	94.3	0.8	0	0.2	2	0
7	0.2	0.4	0.1	0.7	0.5	1.1	80.6	8.3	0.3	2.8	0
8	0	0.2	0	0.3	0.3	0.1	1.9	64.8	2.8	0.5	0
9	0	5.9	1.7	0.9	0.3	0.4	6	23.2	88.6	0.9	0.4
10	0	0.1	0.1	0.3	0	0.2	0.8	0.4	0.2	73.4	0
11	0.2	0.2	0.8	0.1	0.1	1.6	1.1	0.1	0.7	0	96.2

Table 6.1 — Confusion matrix (in %, precision 0.1). Obtained with: Fourier coefficients, and Bayesian distance [34]

	‘1’	‘2’	‘3’	‘4’	‘5’	‘6’	‘7’	‘8’	‘9’	‘10’	‘11’
’1’	**97.7**	0	0	0	0	0	0	0	0	0	0
’2’	0	**100**	0	0	0	0	0	0	0	0	0
’3’	0	0	**90.8**	0	0	0	2.3	4.7	2.4	0	0
’4’	0	0	0	**86.4**	5.5	0	0	0	0	0	0
’5’	2.3	0	2.3	11.3	**91.7**	0	0	0	0	0	0
’6’	0	0	0	0	0	**95.5**	0	0	0	0	0
’7’	0	0	0	0	0	0	**93.1**	2.3	0	0	0
’8’	0	0	0	0	0	0	2.3	**67.4**	2.4	0	0
’9’	0	0	4.5	2.3	2.8	0	2.3	25.6	**92.8**	2.1	0
’10’	0	0	0	0	0	4.5	0	0	2.4	**97.9**	0
’11’	0	0	2.4	0	0	0	0	0	0	0	**100**

Table 6.2 — Confusion matrix (in %, precision 0.1). Obtained with: proposed signature, PCA, and Bayesian distance.

This confusion matrix obtained with the method proposed in this thesis, shows that our method involving optical flow exhibits better recognition results for postures 1, 3, 4, 5, 6, 7, 8, and 9. The obtained results are better in particular for the similar posture couples $\{4, 5\}$ and $\{8, 9\}$, and even much better for posture 8, for which the rate of good recognition increases from 67.4% to 82.8%. In the case where the comparative method exhibits better recognition results, they were excellent (100%, 97.9%, and 100% for postures 2, 10, and 11 respectively), and they are still very good when the proposed method is applied (99.2%, 96.7%, and 99.3%).

In the following in Table 6.4, we consider the performance of the proposed and comparative methods in terms of speed, and overall recognition results.

The method combining Gabor filter, PCA, and SVM (support vector machine) [54] processes 6 frames per second as well (see Table 6.4a). Fourier descriptors programmed

	‘1’	**‘2’**	**‘3’**	**‘4’**	**‘5’**	**‘6’**	**‘7’**	**‘8’**	**‘9’**	**‘10’**	**‘11’**
’1’	**98.2**	0	0	1.0	0	0	0	0	0	0	0
’2’	0	**99.2**	0	0	0	0	0	0	0	0	0
’3’	0	0	**93.1**	0	0	0	0	0	0	0	0.7
’4’	0.6	0	0	**87.7**	7.6	0	0	0	0	0	0
’5’	0	0	0	9.3	**92.4**	0.8	0	0.8	0	0	0
’6’	0	0	0	0	0	**95.8**	0	0	0	2.5	0
’7’	0.6	0	0	0	0	0	**94.3**	3.1	0	0	0
’8’	0	0	0	1.0	0	0	2.5	**82.8**	5.6	0	0
’9’	0.6	0.8	4.9	1.0	0	0	3.2	13.3	**93.6**	0.8	0
’10’	0	0	2.0	0	0	3.4	0	0	0.8	**96.7**	0
’11’	0	0	0	0	0	0	0	0	0	0	**99.3**

Table 6.3 — Confusion matrix (in %, precision 0.1). Obtained with: optical flow, proposed signature, PCA, and Bayesian distancecite [17].

	’Classif. method’	’Speed’	’System’	’Soft’	’%’	’Database’
a)	PCA+SVM	4 frames/sec	3.4 GHz	C	93.7	11*120
b)	Fourier + Bayesian	20 frames/sec	2 GHz	C	84.6	11*1000
c)	PCA + Bayesian	6 frames/sec	3.1 GHz	Matlab	91.8	11*45
d)	OF + PCA + Bayesian	4 frames/sec	3.1 GHz	Matlab + C	94.1	11*110

Table 6.4 — Proposed and comparative methods, comparison of performances. a) Gabor filtered + PCA + SVM [54] ; b) Fourier descriptors (FD1) + Bayesian; c) YC_bC_r mapping, PCA and Bayesian distance [16]; d) proposed method involving optical flow (OF) [17].

in C++ [34] are faster, namely 20 frames per second (see Table 6.4b). The method involving YC_bC_r, PCA and Bayesian distance [16] (see Table 6.4c) mapping is faster (6 frames per second) but it exhibits a major drawback as all methods using YC_bC_r mapping: it does not handle colored hands. Also, the overall recognition rate is lower (91.8%). When we consider the method that we propose [17] (see Table 6.4d), we notice that the computational load dedicated to the recognition of the 1210 images of the database is 302 sec., that is, a mean rate of 4 frames per second. Our method exhibits the best overall recognition rate (94.1%) of all considered method. This good performance relies on the quality of the binary images I which are provided to the signature generation method: whereas the YC_bC_r mapping tended to blur the frontiers and reduce the contrast between hand surface and background on the C_r channel, optical flow permits to apply a threshold to the R,G, and B channels of the RGB color image, where the contrast between hand surface and background is elevated. Currently, the programmes dedicated to optical flow, that is, 15% of the programs, are written in C++. we can expect that transferring all our programmes from Matlab® to C++ would decrease the required computational time.

6.4 Conclusion of the chapter

The issue of hand posture recognition is considered in this chapter. This work is based on a signature generation which divides the image into rings and signal generation directions, thereby getting a matrix. To generate this signature, a binary image containing the contour of the considered hand must be available. To get this binary image from any input image, which is any frame of a video sequence, we adapt, for the first time, optical flow as a contour detection method: we avoid the classically used YC_bC_r mapping, which turns the proposed algorithm fit for colored hands.

Ellipse fitting of the moving points detected by optical flow permits to select a region of interest, thereby ensuring the invariance of the signature to scaling and translation. We assume the center of gravity of the moving points is located in the hand, which provides the grey level distribution for each RGB channel and permits to apply the adequate threshold which segments the hand surface. We then remove the unexpected pixels, which are either isolated or connected to the hand, by retaining the largest connexe region and applying mathematical morphology operations.

The proposed signature is a sparse matrix, hence our proposal to apply principal component analysis to reduce the data dimensionality. We also reduce the dimension of the test set through a first rejection test based on geometric criterion (isometric rate). Hand posture recognition is eventually performed by computing a Bayesian distance between test and pre-selected reference signatures. The visual results show that, despite a complex background, a hand contour is correctly retrieved.

Statistical results summarized as a confusion matrix show that the difficult cases of close postures yield a correct recognition result in more than 82% of the cases. Overall, the mean recognition rate reaches 94.1%, which is more than the rate obtained with the selected comparative methods, in similar testing conditions involving white hands. Our method offers a good compromise between recognition rate and computational load. Our hand posture recognition method has been combined with movement tracking. This could yield a complex but effective set of instructions, in the frame of a Human-machine interaction system.

Conclusion and perspectives

IN this thesis we are interested in achieving a gesture recognition system as part of the design of a touchless Human-machine interface. We studied the various components of such a system and we proposed solutions taking into account important applicative constraints, including the processing of a video stream in real-time. The addressed issues concern hand detection in a video stream, extraction of features representing the shape and position of the hand, recognition of postures from a previously determined vocabulary. This summary outlines the main results of this study and the contributions of our work to achieve a system of recognition. We then give some tracks to further our work.

To evaluate and compare the recognition results, we created a database consisting of 11 postures performed by different people. This database is representative of gestures that can be used in our application, and easy to perform by all users.

We first presented the different methods used for gesture recognition in the literature, and we discussed the constraints in computer vision in general and the industrial context of this thesis in particular. We then proposed a set of methods to achieve these goals.

The first step concerns the detection of the hand in a video stream with a robust method for hand movement and the presence of other objects of same color as the hand in the scene. We found that the segmentation of the hand is a sensitive phase of hand posture recognition. The obtained contour is sometimes too vague, especially because of brightness variations, which affect feature extraction and the recognition of postures (based on the contour). To solve this problem we used the technique of optical flow that we adapted to contour detection. It has allowed us to detect the hand especially for colored people, assuming that the hand moves in the scene in the video stream even if its slightly, but especially more than the other objects.
The extraction of moving points, combined with a least-squares fitting method, allows to determine the ROI of the hand. Then we compute the histogram on the ROI, and apply histogram threshold, with some preprocessings to provide a perfect segmentation of the hand.
For the first time to our knowledge, we handle, by adapting optical flow, the case of colored hands, either wearing gloves or of colored people. Also, we get a dynamic gesture recognition system, which combines hand tracking and hand posture charac-

terization.

The second phase of our work relates to the characterization of postures and feature extraction of the hand. We studied and compared several shape descriptors to calculate a feature vector representing the shape of the hand, taking into account the invariance to Euclidean transformations (translation, rotation and scaling). We notice that the hand contour is generally approximately circular and non star-shaped. Hence, we apply specific methods inspired from array processing. We propose, for the first time in this thesis, a review of all possible types of contours and the corresponding characterization methods inspired by array processing models and methods. Such methods have given, in the past, good results in the frame of possibly distorted linear and circular contours. We insist on the case of highly distorted star-shaped contours, and notice their shape is similar to a hand contour's one. This yielded us to propose a novel 2-D signature which involves the generation of signals. The main difference with respect to the previously existing methods which are inspired from array processing method is that this signature handles the case of non star-shaped contours. We detail how the signals are generated and we prove the different properties of invariance of this new characterization method.
In this step, reviewing all the variants of the methods of array processing transferred to image processing is an important contribution. However, the most novel aspect is our 2D-signature. This signature ensures essentially the invariance to rotation, but also the invariance to the axial asymmetry which allows us recognize both left and right hands, whatever the learning phase.

The proposed signature is a matrix with very large size, which turns very difficult the classification with a geometric classifier. To solve this problem, we have reduced the size of the matrix using the principal component analysis. This dimensionality reduction allowed us to classify the postures with a Bayesian distance criterion, which involves a matrix inversion that scales the components of reference and test vectors. This distance gave us the best results. Also to further improve our results and especially the computational load (0.04 sec/frame), we make a first selection of candidates among the vocabulary through a geometric criterion, the isometric rate.
In this step it can be estimated that the combination of signature generation method and the method of geometric criterion has yielded excellent effects.

The results obtained show that we have reached the best compromise between computational load (4 frames/sec) and recognition rate (94.1%) and we prove that the difficult cases of close postures yield a correct recognition result in more than 82 % of the cases. This compromise corresponds perfectly to the wishes of our interface utilization, in solving constraints as the presence of another object in the scene and variations in acquisition conditions. We can conclude that our process perfectly meets the requirement of our problem.

Among the various prospects of our work an extension and enlargement of vocabulary of postures to recognize are desired. The PCA has allowed us to reduce the dimension of our matrix signature, but we could also apply other methods of dimensionality reduction such as linear discriminant analysis (LDA). Other methods could be applied. For instance adaptive dimension reduction combines dimension reduction and unsupervised learning (clustering) together to improve the reduced data (subspace) adaptively. To continue this work and improve it, we can also attempt to solve the occlusion problem or solve the cases of the presence of multi-target (two hands). For this we could turn our detection method into a multi target one. We can also perform classification by the combination of different classifiers or by SVM (in cascade or multi-class SVM). An optimization of the algorithm, using a single programming language (C++ language), is always possible to accelerate the process and facilitate the industrialization of our algorithm. In the long term, cooperating with institutions and organizations which take care of deaf and dumb persons could help building an adequate vocabulary of postures and gestures which is suitable to define a dictionary of sign language.

Liste of publications:

N. Boughnim, J. Marot, C. Fossati et S. Bourennane, "Hand posture classification by means of a new contour signature" , *Lecture Notes in Computer Science 7517*, (pp. 384-394), 2012.

N. Boughnim, J. Marot, C. Fossati, S. Bourennane et F. Guerault, "Fast and improved hand classification using dimensionality reduction and test set reduction", *ICASSP'13*, (pp. 1971-1975), 2013.

N. Boughnim, J. Marot, C. Fossati, S. Bourennane et F. Guerault, "Hand posture recognition using jointly optical flow and dimensionality reduction", *EURASIP Journal on Advances in Signal Processing (under revision)*, 2013.

List of Figures

List of Tables

Bibliography

[1] *https://sites.google.com/site/boughnimmarothandposture/*.

[2] H. Aghajan, *Subspace techniques for image understanding and computer vision*, PhD Thesis, Stanford University, 1995.

[3] H. Aghajan et T. Kailath, *Slide: subspace-based line detection*, IEEE int. conf. ASSP, vol. 5, pp. 89–92, 1993.

[4] H. Aghajan et T. Kailath, *Slide: Subspace-based line detection*, IEEE Trans. on PAMI, (pp. 1057–1073), 1994.

[5] H. K. Aghajan et T. Kailath, *A subspace fitting approach to super resolution multi-line fitting and straight edge detection*, Proc. of IEEE ICASSP, vol. 3, pp. 121–124, 1992.

[6] H. K. Aghajan et T. Kailath, *Sensor array processing techniques for super resolution multi-line-fitting and straight edge detection*, IEEE Trans. on IP, vol. 2, pp. 454–465, 1993.

[7] V. Athitsos et S. Sclaroff, *Estimating 3d hand pose from a cluttered image*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (pp. 432–439), 2003.

[8] J. Aujol, G. Aubert et L. Blanc-Fraud, *Wavelet-based level set evolution for classification of textured images*, IEEE Trans. on IP, vol. 12, no. 12, pp. 1634–41, 2003.

[9] D. H. Ballard, *Generalizing the hough transform to detect arbitrary shapes*, Pat. Rec., vol. 13, no. 2, pp. 111–22, 1981.

[10] H. Bay, T. Tuytelaars et L. V. Gool, *Surf: Speeded up robust features*, ECCV 2006. LNCS, (pp. 404–417), 2006.

[11] S. S. Beauchemin et J. L. Barron, *The computation of optical flow*, ACM Computing Surveys 27, (pp. 433–466), 1995.

[12] M. J. Black et P. Anandan, *A framework for the robust estimation of optical flow*, Fourth International Conference on Computer Vision, (pp. 231–236), 1993.

[13] M. J. Black et P. Anandan, *The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields*, Computer Vision and Image Understanding 63, (pp. 75–104), 1996.

[14] A. F. Bobick et J. W. Davis, *The recognition of human movement using temporal templates*, on Pattern Analysis and Machine Intelligence, (pp. 257–267), 2001.

[15] N. Boughnim, J. Marot, C. Fossati et S. Bourennane, *Hand posture classification by means of a new contour signature*, Lecture Notes in Computer Science 7517, (pp. 384–394), 2012.

[16] N. Boughnim, J. Marot, C. Fossati, S. Bourennane et F. Guerault, *Fast and improved hand classification using dimensionality reduction and test set reduction*, ICASSP'13, (pp. 1971–1975), 2013.

[17] N. Boughnim, J. Marot, C. Fossati, S. Bourennane et F. Guerault, *Hand posture recognition using jointly optical flow and dimensionality reduction*, EURASIP Journal on Advances in Signal Processing (under revision), 2013.

[18] J. Bouguet, *Pyramidal implementation of the lucas kanade feature tracker: Description of the algorithm*, Technical report, OpenCV documents, Intel Corporation, Microprocessor Research Labs, vol. 1, 2000.

[19] S. Bourennane et C. Fossati, *Comparison of shape descriptors for hand posture recognition in video*, Signal, Image and Video Processing, vol. 6, pp. 147–157, 2012.

[20] S. Bourennane et J. Marot, *Line parameters estimation by array processing methods*, IEEE ICASSP, vol. 4, pp. 965–968, 2005.

[21] S. Bourennane et J. Marot, *Contour estimation by array processing methods*, Applied signal processing, 2006.

[22] S. Bourennane et J. Marot, *Estimation of straight line offsets by a high resolution method*, IEE proceedings - Vision, Image and Signal Processing, vol. 153, pp. 224–229, 2006.

[23] A. Braffort, *Reconnaissance et comprÃ©hension de gestes, application Ã la langue des signes*, Phd, Universite Paris-XI, 1996.

[24] T. Brox et J. Malik, *Large displacement optical flow: Descriptor matching in variational motion estimation*, IEEE trans. on PAMI, vol. 33, pp. 500 – 513, 2011.

[25] C. Cadoz, *Le geste canal de communication homme/machine - la communication instrumentale*, Techniques et Science Informatiques, vol. 13, pp. 31–61, 1994.

[26] J. Canny, *A computational approch to edge detection*, IEEE Trans. Pattern Anal. Machine Intell, vol. 8, pp. 679–714, 1986.

[27] A. Caplier, L. Bonnaud, S. Malassiotis et M. Strintzis, *Comparison of 2d and 3d analysis for automated cued speech gesture recognition*, SPECOM, 2004.

[28] M.-E. Celebi et Y.-A. Aslandogan, *A comparative study of three moment-based shape descriptors*, ITCC'05, vol. 1, pp. 788–793, 2005.

[29] T. Chan et L. Vese, *Active contours without edges*, IEEE Trans. on IP, vol. 10, no. 2, pp. 266–277, 2001.

[30] M.-A. Charmi, S. Derrode et F. Ghorbel, *Fourier-based geometric shape prior for snakes*, Pattern Recognition Letters, vol. 29, no. 7, pp. 897–904, 2008.

[31] F.-S. Chen, C.-M. Fu et C.-L. Huang, *Hand gesture recognition using a real-time tracking method and hidden markov models*, Image and Vision Computing, (pp. 745–758), 2003.

[32] C. Chong, P. Raveendran et R.Mukundan, *A comparative analysis of algorithms for fast computation of zernike moments*, Pattern Recognition, vol. 36(3), pp. 731–742, 2003.

[33] J.-F. Collumeau, H. Laurent, B. Emile et R. Leconge, *Hand posture recognition with multiview descriptors*, advanced concepts for intelligent vision systems (ACIVS), (pp. 455–466), 2012.

[34] S. Conseil, S. Bourennane et L. Martin, *Comparison of fourier descriptors and hu moments for hand posture recognition*, 15th European Signal Processing Conference (EUSIPCO'07), 2007.

[35] T. R. Crimmins, *A complete set of fourier descriptors for two dimensional shape*, IEEE trans. on Systems, Man, and Cybernetics, (pp. 848–855), 1982.

[36] T. Darrell et A. Pentland, *Space-time gestures*, on Computer Vision and Pattern Recognition, (pp. 335–340), 1993.

[37] R. O. Duda et P. E. Hart, *Use of the hough transformation to detect lines and curves in pictures*, Communications of the ACM, vol. 15, no. 1, pp. 11–15, 1972.

[38] Y. Fang, J. Cheng, J. Wang, K. Wang, J. Liu et H. Lu, *Hand posture recognition with cotraining*, International Conference on Pattern Recognition, 2008.

[39] Y. Fang, J. Cheng, K. Wang et H. Lu, *Hand gesture recognition using fast multi-scale analysis*, International Conference on Image and Graphics (ICIG), (pp. 694–698), 2007.

[40] A. Foulonneau, P. Charbonnier et F. Heitz, *Geometric shape priors for region-based active contours*, ICIP'03, vol. 3, pp. 413–416, 2003.

[41] Y. Freund et R. E. Schapire, *A decision-theoretic generalization of on-line learning and an application to boosting*, Journal of Computer and System Sciences, (pp. 119–139), 1997.

[42] W. Gander, G. Golub et R. Strebel, *Least-squares fitting of circles and ellipses*, BIT, vol. 34, pp. 558–578, 1994.

[43] F. Ghorbel, *Stability of invariant fourier descriptors and its inference in the shape classification*, IEEE International Conference on Pattern Recognition, (pp. 130–133), 1992.

[44] W. Gong, J. Gonzalez et F. Roca, *Human action recognition based on estimated weak poses*, EURASIP Journal on Advances in Signal Processing, 2012.

[45] L. Gu et K. Rose, *Perceptual harmonic cepstral coefficients for speech recognition in noisy environment*, Proceedings of the IEEE ICASSP, (pp. 189–192), 2001.

[46] B. Halder, H. Aghajan et T. Kailath, *Propagation diversity enhancement to the subspace-based line detection algorithm*, Nonlinear Image Processing VI, (pp. 320–328), 1995.

[47] J. Han, F. Qi et G. Shi, *Gradient sparsity for piecewise continuous optical flow estimation*, ICIP'11, (pp. 2341–2344,), 2011.

[48] C. Harris et M. Stephens, *A combined corner and edge detector*, Proceedings of the 4th Alvey Vision Conference, (pp. 147–151), 1988.

[49] H.Bay, A.Ess, T. Tuytelaars et L. V. Gool, *Surf: Speeded-up robust features*, ComputerVision and Image Understanding (CVIU), vol. 110(3), pp. 346–359, 2008.

[50] B. Horn et B. Schunck, *Determining optical flow*, Artificial Intelligence, vol. 17 (1-3), pp. 185–203, 1981.

[51] V. Hough et C. Paul, *Method and means for recognizing complex patterns*, numéro 3069654, 1962.

[52] M.-K. Hu, *Visual pattern recognition by moment invariants*, IEEE trans. on Information Theory, vol. 8, pp. 179–187, 1962.

[53] M.-K. Hu, *Visual pattern recognition by moment invariants*, IEEE trans. on Information Theory, vol. 8, pp. 179–187, 1962.

[54] D.-Y. Huang, W.-C. Hu et S.-H. Chang, *Gabor filter-based hand-pose angle estimation for hand gesture recognition under varying illumination*, Expert Systems with Applications, vol. 38, pp. 6031–6042, 2011.

[55] Y. Huang et X. H. Zhuang, *Motion-partitioned adaptive block matching for video compression*, International Conference on Image Processing, vol. 1, p. 554, 1995.

[56] S.-K. Hwang et W.-Y. Kim, *A novel approach to the fast computation of zernike moments*, Pattern Recognition, vol. 39(11), pp. 2065–2076, 2006.

[57] J. Illingworth et J. Kittler, *A survey of the hough transform*, Comput. Vis. Graph. IP, , no. 44, pp. 87–116, 1988.

[58] K. Imen, R. Fablet, J.-M. Boucher et J.-M. Augustin, *Region-based segmentation using texture statistics and level-set methods*, IEEE ICASSP'06, , no. 2, pp. 693–96, 2006.

[59] B. Ionescu, D. Coquin, P. Lambert et V. Buzuloiu, *Dynamic hand gesture recognition using the skeleton of the hand*, EURASIP Journal on Applied Signal Processing, vol. 2101-2109, 2005.

[60] H. Jiang, J. Marot, C. Fossati et S. Bourennane, *Circular contour retrieval in real-world conditions by higher order statistics and an alternating-least squares algorithm*, Eurasip Journal on advances in signal processing, 2011.

[61] H. Jiang, J. Marot, C. Fossati et S. Bourennane, *Strongly concave star-shaped contour characterization by algebra tools*, Elsevier Signal Processing, 2012.

[62] D. Jones, C. Pertunen et B. Stuckman, *Lipschitzian optimization without the lipschitz constant*, Journal of Optimization and Applications, vol. 79, pp. 157–181, 1993.

[63] M. J. Jones et J. Rehg, *Statistical color models with application to skin detection*, on Computer Vision and Pattern Recognition, 1999.

[64] M.-B. KaÁniche et F. BrÃ©mond, *Tracking hog descriptors for gesture recognition*, IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 2009.

[65] M. Kass, A. Witkin et D. Terzopoulos, *Snakes: Active contour model*, Int. J. Computer Vision, vol. 321-31, 1988.

[66] A. Khotanzad et Y. H. Hong, *Invariant image recognition by zernike moments*, IEEE trans. on Pattern Analysis and Machine Intelligence, vol. 12(5), pp. 489–497,, 1990.

[67] N. Kiryati et A. Bruckstein, *What's in a set of points? [straight line fitting]*, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 14, no. 4, pp. 496–500, 1992.

[68] W. W. Kong et S. Ranganath, *3d hand trajectory recognition for signing exact english*, on Automatic Face and Gesture Recognition, (pp. 535–540), 2004.

[69] S. Kumar et C. Singh, *A study of zernike moments and its use in devanagari handwritten character recognition*, In Proc. of the Int. Conf. on Cognition and Recognition, (pp. 514–520), 2005.

[70] J. C. Lagarias, J. A. Reeds, M. H. Wright et P. E. Wright, *Convergence properties of the nelder-mead simplex method in low dimensions*, SIAM Journal of Optimization, (pp. 112–147), 1998.

[71] A. Licsar et T. Sziranyi, *User-adaptive hand gesture recognition system with interactive training*, Image and Vision Computing, (pp. 1102–1114), 2005.

[72] D.-G. Lowe, *Object recognition from local scale-invariant features*, the International Conference on Computer Vision, 1999.

[73] B. Lucas et T. Kanade, *An iterative image registration technique with an application to stereo vision*, Procs. of the 7th International Joint Conference on Artificial Intelligence, 1981.

[74] S. Marcel, O. Bernier, J.-E. Viallet et D. Collobert, *Hand gesture recognition using input-output hidden markov models*, on Automatic Face and Gesture Recognition, (pp. 456–461), 2000.

[75] J. Marot et S. Bourennane, *Array processing and fast optimization algorithms for distorted circular contour retrieval*, EURASIP Journal on Advances in Signal Processing, 2007.

[76] J. Marot et S. Bourennane, *Subspace-based and direct algorithms for distorted circular contour estimation*, IEEE Trans.Image Process, vol. 2369-2378, 2007.

[77] J. Marot et S. Bourennane, *Subspace-based and direct algorithms for distorted circular contour estimation*, IEEE Transactions on IP, vol. 16(9), pp. 2369–2378, 2007.

[78] J. Martin, *Reconnaissance de gestes en vision par ordinateur*, Phd Institut National Polytechnique de Grenoble, 2000.

[79] J. Martin et J.-B. Durand, *Automatic handwriting gestures recognition using hidden markov models*, In Proc. of the IEEE Int. Conf. on Automatic Face and Gesture Recognition, 2000.

[80] S.-J. McKennaa, Y. Rajab et S. Gongb, *Tracking colour objects using adaptive mixture models*, Image and Vision Computing, vol. 17, pp. 225–231, 1999.

[81] M. Mozerov, I. Rius, X. Roca et J. Gonzalez, *Nonlinear synchronization for automatic learning of 3d pose variability in human motion sequences*, EURASIP Journal on Advances in Signal Processing, 2009.

[82] D. N., T. B. et S. C., *Human detection using oriented histograms of flow and appearance*, ECCV 2006. LNCS, vol. 3952, pp. 428–441, 2006.

[83] J. New, E. Hasanbelliu et M. Aguilar, *Facilitating user interaction with complex systems via hand gesture recognition*, Southeastern ACM Conference, 2003.

[84] C. W. Ng et S. Ranganath, *Real-time gesture recognition system and application*, Image and Vision Computing, vol. 993-1007, 2002.

[85] K. Oka, Y. Sato et H. Koike, *Real-time fingertip tracking and gesture recognition*, on Computer Graphics and Applications, (pp. 64–71), 2002.

[86] N. Otsu, *A threshold selection method from gray level histogram*, IEEE Trans. Syst. Man Cybern, vol. 9, pp. 62 – 66,, 1979.

[87] P. V. Otterloo, *A contour-oriented approach to shape analysis*, Prentice Hall International (UK), Hertfordshire, vol. ISBN 0-13-173840-2, 1991.

[88] E. Persoon et K. S. Fu, *Shape discrimination using fourier descriptors*, IEEE trans. on Pattern Analysis and Machine Intelligence, (pp. 388–397), 1986.

[89] S. L. Phung, A. Bouzerdoum et D. Chai, *Skin segmentation using color pixel classification : analysis and comparison*, on Pattern Analysis and Machine Intelligence, (pp. 148–154), 2005.

[90] S. Pillai et B. Kwon, *Forward/backward spatial smoothing techniques for coherent signal identification*, Proc. of IEEE trans. on ASSP, vol. 37, pp. 8–15, 1989.

[91] S. Poularakis, G. Tsagkatakis, P. Tsakalides et I. Katsavounidis, *Sparse representation for hand gesture recognition*, (pp. 3746–3750), 2013.

[92] R. Roy et T. Kailath, *Esprit: Estimation of signal parameters via rotational invariance techniques*, IEEE trans. on Information Theory, vol. 37, no. 7, pp. 984–995, 1989.

[93] Y. Sato, K. Oka, H. Koike et Y. Nakanishi, *Video-based tracking of user's motion for augmented desk interface*, on Automatic Face and Gesture Recognition, 2004.

[94] R. O. Schmidt, *Multiple emitter location and signal parameter estimation*, IEEE Trans. Antennas and Propag, (pp. 276–280), 1986.

[95] J. Segen et S. Kumar., *Fast and accurate 3d gesture recognition interface*, International Conference on Pattern Recognition, vol. 1, p. 86, 1998.

[96] J. Sheinvald et N. Kiryati, *On the magic of slide*, Machine Vision and Applications, vol. 9, pp. 251–261, 1997.

[97] J. Shi et C. Tomasi, *Good features to track*, 9th IEEE Conference on Computer Vision and Pattern Recognition, 1994.

[98] X. Shu et X.-J. Wu, *A novel contour descriptor for 2d shape matching and its application to image retrieval*, Image and Vision Computing, vol. 29, pp. 286–294, 2011.

[99] M. Soriano, B. Martinkauppi, S. Huovinen et M. Laaksonen, *Skin detection in video under changing illumination conditions*, Procs. of the 15th ICPR, vol. 1, pp. 839–842, 2000.

[100] T. Starner et A. Pentland, *Visual recognition of american sign language using hidden markov models*, on Automatic Face and Gesture Recognition, (pp. 189–194), 1995.

[101] T. Starner, J. Weaver et A. Pentland, *Real-time american sign language recognition using desk and wearable computer based video*, on Pattern Analysis and Machine Intelligence, (pp. 1371–1375), 1998.

[102] M. R. Teague, *image analysis via the general theory of moments*, J. Optical Soc. Am., vol. 70, pp. 920–930, 1980.

[103] J. Triesch et C. von der Malsburg, *Robust classification of hand postures against complex backgrounds*, In Proc. of the IEEE Int. Conf. on Automatic Face and Gesture Recognition, (pp. 170–175), 1996.

[104] D. Tufts et R. Kumaresan, *Estimation of frequencies of multiple sinusoids: making linear prediction perform like maximum likelihood*, Proc. IEEE, vol. 70, pp. 975–989, 1982.

[105] C. Vogler et D. Metaxas, *Asl recognition based on a coupling between hmms and 3d motion analysis*, on Computer Vision, (pp. 363–369), 1998.

[106] C. Vogler et D. Metaxas, *Parallel hidden markov models for american sign language recognition*, on Computer Vision, (pp. 116–122), 1999.

[107] C.-C. Wang et K.-C. Wang, *Hand posture recognition using adaboost with sift for human robot interaction*, International Conference on Advanced Robotics (ICAR), 2007.

[108] M. Wax et T. Kailath, *Detection of signals by information theoretic criteria*, IEEE Trans. Acoustics, Speech, and Signal Process, vol. 33, no. 2, pp. 387–392, 1985.

[109] A. D. Wilson et A. F. Bobick, *Parametric hidden markov models for gesture recognition*, on Pattern Analysis and Machine Intelligence, (pp. 884–900), 1999.

[110] X. Xianghua et M. Mirmehdi, *Rags: region-aided geometric snake*, IEEE Trans. on IP, vol. 13, no. 5, pp. 640–52, 2004.

[111] C. Xu et J. Prince, *Gradient vector flow: a new external force for snakes*, IEEE Comp. Soc. Conf. Comp. Vis., Pat.Rec., (pp. 66–71), 1997.

[112] T.-W. Yoo et I.-S. Oh, *A fast algorithm for tracking human faces based on chromatic histograms*, Pattern Recognition Letters, vol. 20(10), pp. 967–978, 1999.

[113] D. Zhang et G. Lu, *A comparative study on shape retrieval using fourier descriptors with different shape signatures*, on Intelligent Multimedia and Distance Education, 2001.

[114] H. Zhou, D. Lin et T. Huang., *Static hand gesture recognition based on local orientation histogram feature distribution model*, on Computer Vision and Pattern Recognition Workshop, vol. 10, p. 161, 2004.

[115] Y. Zhu, G. Xu et D. J. Kriegman, *A real-time approach to the spotting, representation, and recognition of hand gestures for human-computer interaction*, Computer Vision and Image Understanding, vol. 85, pp. 189–208, 2002.