IDIAP RESEARCH REPORT

Two-Handed Gestures for Human-Computer Interaction

Agnès Just [a] [b]

IDIAP–RR 06-73

December 2006

[a] IDIAP Research Institute
[b] Ecole Polytechnique Fédérale de Lausanne

IDIAP Research Institute www.idiap.ch
Rue du Simplon 4 P.O. Box 592 1920 Martigny – Switzerland
Tel: +41 27 721 77 11 Fax: +41 27 721 77 12 Email: info@idiap.ch

Rapport de recherche de l'IDIAP 06-73

Two-Handed Gestures for Human-Computer Interaction

Agnès Just

December 2006

Abstract

The present thesis is concerned with the development and evaluation (in terms of accuracy and utility) of systems using hand postures and hand gestures for enhanced Human-Computer Interaction (HCI). In our case, these systems are based on vision techniques, thus only requiring cameras, and no other specific sensors or devices.

When dealing with hand movements, it is necessary to distinguish two aspects of these hand movements : the *static* aspect and the *dynamic* aspect. The static aspect is characterized by a pose or configuration of the hand in an image and is related to the Hand Posture Recognition (HPR) problem. The dynamic aspect is defined either by the trajectory of the hand, or by a series of hand postures in a sequence of images. This second aspect is related to the Hand Gesture Recognition (HGR) task. Given the recognized lack of common evaluation databases in the HGR field, a first contribution of this thesis was the collection and public distribution of two databases, containing both one- and two-handed gestures, which part of the results reported here will be based upon. On these databases, we compare two state-of-the-art models for the task of HGR. As a second contribution, we propose a HPR technique based on a new feature extraction. This method has the advantage of being faster than conventional methods while yielding good performances. In addition, we provide comparison results of this method with other state-of-the-art technique. Finally, the most important contribution of this thesis lies in the thorough study of the state-of-the-art not only in HGR and HPR but also more generally in the field of HCI.

The first chapter of the thesis provides an extended study of the state-of-the-art. The second chapter of this thesis contributes to HPR. We propose to apply for HPR a technique employed with success for face detection. This method is based on the Modified Census Transform (MCT) to extract relevant features in images. We evaluate this technique on an existing benchmark database and provide comparison results with other state-of-the-art approaches. The third chapter is related to HGR. In this chapter we describe the first recorded database, containing both one- and two-handed gestures in the 3D space. We propose to compare two models used with success in HGR, namely Hidden Markov Models (HMM) and Input-Output Hidden Markov Model (IOHMM). The fourth chapter is also focused on HGR but more precisely on two-handed gesture recognition. For that purpose, a second database has been recorded using two cameras. The goal of these gestures is to manipulate virtual objects on a screen. We propose to investigate on this second database the state-of-the-art sequence processing techniques we used in the previous chapter. We then discuss the results obtained using different features, and using images of one or two cameras.

In conclusion, we propose a method for HPR based on new feature extraction. For HGR, we provide two databases and comparison results of two major sequence processing techniques. Finally, we present a complete survey on recent state-of-the-art techniques for both HPR and HGR. We also present some possible applications of these techniques, applied to two-handed gesture interaction. We hope this research will open new directions in the field of hand posture and gesture recognition.

Keywords : Human-Computer Interaction, Computer Vision, Hand Posture Recognition, Modified Census Transform, Hand Gesture Recognition, Hidden Markov Model, Input-Output Hidden Markov Model

Résumé

Cette thèse a pour objet le développement et l'évaluation de systèmes utilisant les gestes et postures de la main pour l'interaction homme-machine améliorée.

Les postures de la main sont caractérisées par une pose ou conformation de la main dans une image. Les gestes de la main sont quant à eux caractérisés soit par la trajectoire de la main, soit par une série de postures de la main dans une séquence d'images. Etant donné le manque de bases de données pour l'évaluation des techniques de reconnaissance des gestes de la main, une première contribution de cette thèse est la collection et la distribution publique de deux bases de données, contenant à la fois des gestes mono- et bi-manuels. Une partie des résultats contenus dans cette thèse sont obtenus sur ces deux bases de données. Sur ces deux bases de données, nous comparons deux modèles de l'état-de-l'art dans le domaine de la reconnaissance des gestes de la main. Comme seconde contribution, nous proposons une technique pour la reconnaissance des postures de la main. Cette méthode est basée sur une nouvelle technique pour l'extraction des éléments caractéristiques dans l'image. Cette approche a l'avantage d'être plus rapide que les méthodes conventionnelles tout en fournissant de bonnes performances en reconnaissance. Nous fournissons aussi des résultats comparatifs entre la méthode que nous proposons et d'autres methodes de l'état-de-l'art. Finalement, la contribution la plus importante de cette thèse consiste en une étude complète des techniques récentes pour la reconnaissance des postures et des gestes de la main, mais aussi plus généralement dans le champ de l'interaction homme-machine.

Le premier chapitre de la thèse présente une étude de l'état-de-l'art. Le second chapitre contribue à la reconnaissance des postures de la main. Nous proposons d'appliquer à ce problème une technique employée avec succès pour la détection des visages. Cette méthode est basée sur la Modified Census Transform (MCT) pour extraire des informations pertinentes dans l'image. Nous évaluons cette technique sur une base de données de référence pour la reconnaissance des postures de la main. Nous fournissons des résultats comparatifs avec d'autres méthodes de l'état-de-l'art. Le troisième chapitre est relatif à la reconnaissance des gestes de la main. Dans ce chapitre, nous décrivons la première base de données enregistrée au cours de cette thèse. Cette base de données contient à la fois des gestes mono- et bi-manuels. Nous proposons de comparer deux modèles utilisés avec succès pour la reconnaissance des gestes de la main. Il s'agit des chaînes de Markov cachées et des chaînes de Markov cachées à entrée-sortie. Le quatrième chapitre se concentre sur la reconnaissance des gestes bi-manuels. Pour cette raison, une deuxième base de données a été enregistrée, utilisant deux caméras. Le but de ces gestes est de manipuler des objets virtuels affichés sur un écran. Nous proposons de comparer les mêmes techniques que précédemment, à savoir les chaînes de Markov cachées et les chaînes de Markov cachées à entrée-sortie, sur cette seconde base de données. Nous discutons ensuite les résultats obtenus en utilisant différentes caractéristiques extraites des images, ainsi qu'en utilisant les informations extraites d'une seule ou des deux caméras.

En conclusion, nous proposons une méthode de reconnaissance des postures de la main, méthode basée sur une nouvelle technique pour extraire de l'information dans les images. Dans le cadre de la reconnaissance des gestes de la main, nous proposons deux bases de données ainsi que des résultats comparatifs de deux techniques majeures pour la reconnaissance des gestes de la main sur ces deux bases de données. Finalement, nous présentons une étude complète des techniques de l'état-de-l'art dans les domaines de la reconnaissance des postures et des gestes de la main. Nous présentons aussi des applications possibles de ces techniques, appliquées à l'interaction bi-manuelle. Nous espérons que cette recherche ouvrira de nouvelles directions dans le champ de la reconnaissance des gestes et postures de la main.

Mots-clés : Interaction Homme-Machine, Vision par Ordinateur, Reconnaissance des Postures de la Main, Modified Census Transform, Reconnaissance des Gestes de la Main, Chaîne de Markov Cachée, Chaîne de Markov Cachée à Entrée-Sortie.

Table des matières

Table des figures

Liste des tableaux

Acknowledgments

These acknowledgments have been very difficult for me to write. But there are definitely some people who deserve my thanks. I would like first to thank Prof. Hervé Bourlard for welcoming me in his laboratory and Dr. Sébastien Marcel for his super-vision.
Yann, many thanks for the great time we had together during these four years. I don't think this time would have been as good without you. I also would like to thank Valentina for giving me good reasons to relax after work. And for sure, John, you have been very helpful, particularly when you were not here.

I also would like to thank Yves Kodratoff and Michèle Sebag for introducing me to the world of research and pushing me in that direction.

Deux autres personnes manquent à cette liste. Maman, papa, merci pour votre soutien indéfectible. Mais vous savez dejà tout ça. Il y a aussi une personne qui a une place spéciale dans mon coeur. Après avoir remercié mes parents, il est normal que ce soit le tour de mon deuxième père, mon petit père Barou. C'est lui qui m'a mis le doigt dans l'engrenage de la connaissance, c'est grâce à lui que j'ai pu dévorer mes premiers livres. C'est en quelques sortes grâce à lui que tout a commencé.

Obviously lots of people are missing from these acknowledgments, so I want to thank them all now.

Merci.

Chapitre 1

Introduction

1.1 Motivations

Since their first appearance, computers have become a key element of our society. Surfing the web, typing a letter, playing a video game or storing and retrieving data are just a few of the examples involving the use of computers. And due to the constant decrease in price of personal computers, they will even more influence our everyday life in the near future.

To efficiently use them, most computer applications require more and more interaction. For that reason, human-computer interaction (HCI) has been a lively field of research these last few years. Firstly based in the past on punched cards, reserved to experts, the interaction has evolved to the graphical interface paradigm. The interaction consists of the direct manipulation of graphic objects such as icons and windows using a pointing device. Even if the invention of keyboard and mouse is a great progress, there are still situations in which these devices can be seen as dinosaurs of HCI. This is particularly the case for the interaction with 3D objects. The 2 degrees of freedom (DOFs) of the mouse cannot properly emulate the 3 dimensions of space. Furthermore, such interfaces are often not intuitive to use.

As a result, much research is going on to develop interaction methods closer to those used in human-human interaction, e.g. by using speech and body language/gestures. For example, computer applications can be enhanced by providing the user the facility to combine gesture and vocal commands. Some obvious applications belong to the virtual reality, visualization and sign language recognition fields. Hand movements can also be used to command machines or appliances, a potential benefit to elderly or disabled people (such as motor disabled people, or with cerebral disease). Teleconferencing can also gain from gestural interfaces, enabling speakers to communicate in a more natural way. Moreover, the most intuitive and powerful applications make use of bi-manual movements, improving the benefit of hand gesture interaction.

Using hands as a device can help people communicate with computers in a more intuitive way. When we interact with other people, our hand movements play an important role and the information they convey is very rich in many ways. We use our hands for pointing at a person or at an object, conveying information about space, shape and temporal characteristics. We constantly use our hands to interact with objects : move them, modify them, transform them. In the same unconscious way, we gesticulate while speaking to communicate ideas ('stop', 'come closer', 'no', etc). Hand movements are thus a mean of non-verbal communication, ranging from simple actions (pointing at objects for example) to more complex ones (such as expressing feelings or communicating with others). In this sense, gestures are not only an ornament of spoken language, but are essential components of the language generation process itself.

The potential power of gestures has already been demonstrated in the system CHARADE [9]. In this application, hand gesture inputs were used to control a computer while giving a presentation. The user was able to interact with slides using a DataGlove, worn by the user and linked to the computer.

Other applications of gesture recognition techniques include computer-controlled games. For instance, rather than pressing buttons, players in a computer game can pantomime actions or gestures which the computer recognizes (Eye Toy[1]).

To interpret gestures, computers need to perceive the outside world. A common technique is to instrument the hand. It can be done with magnetic sensors (such as the polhemus device, used in the "Put-That-There" system [16]), using acoustic or inertial trackers. But all of these methods, even giving results with good range, suffer from main disadvantages (the interested reader would refer to [95]). Another possibility is to use a glove with numerous sensors which provide information about hand position, orientation and flex of the fingers. It is also possible to apply vision-based technologies. The set-up is much simpler as vision-based technologies only use a set of video cameras to record the gestures to analyze.

Glove-based and vision-based technologies are so far the best techniques to record hand movements, but they have both advantages and disadvantages :

- **Cost** : Even though glove-based technology has come down in price, the cost of robust and complex gesture recognition systems will still be high if a glove-based solution is used. The cost of a robust glove is still in the hundreds of dollars. On the other hand, the vision-based solution is relatively inexpensive, because of the decreasing price of cameras.
- **User Comfort** : Now wireless gloves exist[2], but even if the user is not connected anymore to the computer, she/he has to wear the device. On the other hand, the camera gives the user the complete freedom of motion and provides a cleaner way to interact and perform hand movements. Another important point is that the human hand varies in shape and size. This is a significant problem with gloves because one size is supposed to fit many hand sizes.
- **Computing power** : Depending on the algorithms used, both glove-based and vision-based solutions can require significant computing power. The only advantage of gloves over cameras is that data sent to the computer are already appropriate for recognition, thus skipping a major feature extraction preprocessing step. While in vision-based solutions the feature extraction step is still a difficult and unsolved task.
- **Calibration** : Calibration is an important point for both glove- and vision-based solutions, but it is more critical with gloves. Manufacturers assume that a user's hand conforms to a generic model. As a result, data gloves are not tailored to the specific geometry and features of an individual user's hand. Hence data from the gloves need to be filtered and altered in order to match a specific user. Therefore a calibration step is required for every user in the case of a glove-based system. But with camera, a general calibration procedure is enough for a wide variety of users.
- **Noise** : In glove-based systems, some type of noise is bound to be associated with data. Filtering algorithms are therefore necessary to reduce this kind of noise. On the contrary, with cameras, the noise is minimal. The only problem lies in the feature extraction process.

To overcome the limitations imposed by glove-based devices, a camera can be a powerful interface between the computer and the user. Such unencumbered interaction can make computers easier to use. Furthermore, the vision-based approach carries a tremendous advantage over techniques that require the use of mechanical transducers : it is unobtrusive. In this thesis we will focus on the vision-based approach for hand gestures applied to HCI. The following sections will present a brief taxonomy of gestures and problems inherent to vision-based methods applied to the recognition of hand movements.

1.2 Overview of Gestures

To efficiently use hand gestures for HCI, it is important to understand how humans communicate with their hands. This comprehension will help us use hand movements more efficiently as an interface to computer applications.

[1]http ://www.eyetoy.com
[2]http ://www.5dt.com/products/pdataglove_wirelesskit.html

Human gestures serve three functional roles [24] : semiotic, ergotic and epistemic. The *semiotic* function of gesture is to communicate meaningful information. The structure of a semiotic gesture is conventional and commonly results from shared cultural experience. Some examples are the good-bye gesture, the American Sign Language and all types of coded gestures. The *ergotic* function of gesture is associated with the notion of work. It corresponds to the ability of humans to manipulate the physical world, to create artefacts. The *epistemic* function of gesture allows humans to learn from the environment through tactile or haptic exploration.

Another approach to hand gestures consists of studying their linguistic content. Gestures can be classified into five categories [85] : gesticulation, language-like gestures, pantomime, emblems and sign language. *Gesticulation* is defined as the spontaneous movements of the hands and arms that accompany speech. *Language-like gestures* are a type of gesticulation that is integrated into a spoken utterance, replacing a particular spoken word or phrase. *Pantomimes* are those gestures that depict objects or actions, with or without accompanying speech. *Emblems* are familiar gestures such as "V for victory", "thumbs up", and assorted rude gestures (these are often culturally specific). And *sign language* is all the linguistic systems, such as American Sign Language, which are well defined.

As our goal is to recognize user's hand movements to communicate with computers, we will be mostly concerned with empty handed *semiotic* gestures. *Emblematic* gestures, although less spontaneous and natural, carry more clear semantic meaning. They can be used to create a commands-and-controls vocabulary for HCI. The vast majority of existing automatic gesture recognition systems are using pointing gestures, emblematic gestures (isolated signs) and sign language (with a limited vocabulary and syntax).
For more detailed taxonomies of gestures, the interested reader would refer to articles listed in the bibliography [85, 143, 124, 24].

Furthermore, it is necessary to distinguish two aspects of hand movements :

– the static aspect is, for instance, characterized by a pose or configuration of the hand in an image. The static aspect of hand movements is related to the Hand Posture Recognition (HPR) problem.
– the dynamic aspect is defined either by the trajectory of the hand, or by a sequence of hand postures in a sequence of images. The dynamic aspect of hand movements is related to the Hand Gesture Recognition (HGR) problem.

Hand gestures can be decomposed into three phases [124] : pre-stroke or preparation, stroke or nucleus (or peak) and post-stroke (or retraction). When hand gestures are produced continuously, each gesture is affected by the gesture that precedes it, and possibly by the gesture that follows it. And this co-articulation may be a problem for a gesture recognition system.

1.3 Scope of the Problem

Gestural interfaces based on vision technologies are the most natural way for the construction of advanced man-machine interfaces. But the use of images instead of dedicated acquisition devices is much more challenging. There are three main problems represented in Figure 1.1 : *segmentation* of the hand, *tracking* and *recognition* of the hand posture or gesture (feature extraction and classification).

– **Segmentation** : In most of the literature, hand segmentation has been performed either using a controlled (uncluttered) background, using a known background (i.e. background subtraction), using segmentation by motion, or using color segmentation (i.e. skin color filtering). Using controlled or known backgrounds can be problematic in dynamic environments where the background can change over time, and thus are non-realistic. Motion cues can be difficult to apply due to artefacts of motion caused by changing light and camera motions. Color segmentation is a fast and fairly robust approach to hand segmentation that works well under varying lighting conditions and against unknown backgrounds. It cannot be used if the background behind the hand is close to the color of the skin.

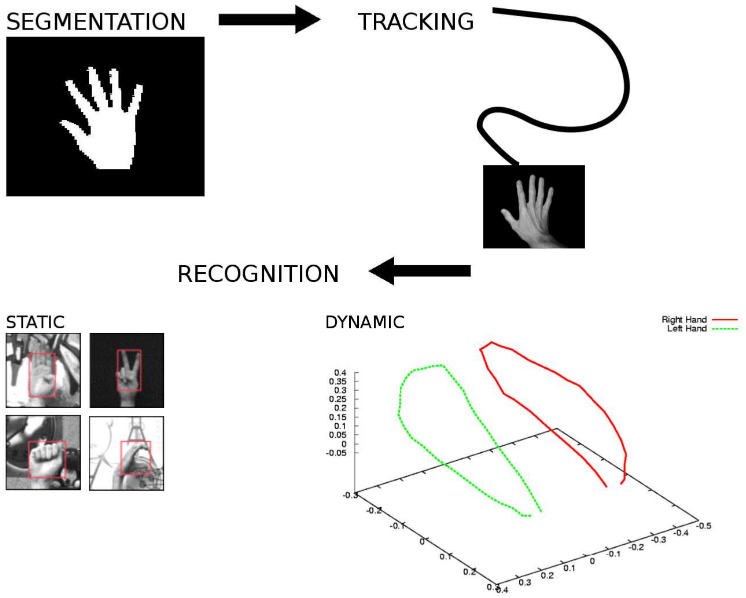


FIG. 1.1 – The three main problems related to gestural interfaces : Segmentation, Tracking and Recognition

– **Tracking** : Articulated objects (such as the hand) are more difficult to track than single rigid objects. The analysis process of the human hand is further complicated by the fact that the hand is a non-rigid articulated structure with 27 DOFs, and with changes in shape in various ways. The four fingers are adjacent to each other that leads to self-occlusions. To overcome these difficulties, tracking has been facilitated through the use of special markers (colored gloves or color marked gloves), and a combination of color and shape constraints. Tracking can be done using one single camera (mono) or using multiple cameras (stereo). To track the hand in 3D, a model of the hand is needed. This model is difficult to obtain due to the high number of DOFs of the human hand. Hand tracking needs to update well chosen parameters through consecutive images. And this problem is strongly tied with the hand segmentation in each image along the tracking process.

– **Recognition** : The major difficulties of hand posture and gesture recognition are feature extraction and the recognition itself. Because hand postures and gestures are highly variable from one person to another, and from one example to another within a single person, it is essential to capture their essence -their invariant properties- and use this information to represent them. This is the reason why feature description is an important problem for both HPR and HGR. The features must optimally distinguish the variety of hand gestures or postures from each other and make recognition of similar gestures or postures simpler. Another problem related to HGR is gesture segmentation (or gesture spotting). Hand gesture spotting consists of determining the start and end point of a particular gesture. Gesture spotting needs also to deal with the rejection of unknown gestures. The problem of gesture segmentation is met when we try to recognize a sequence of gestures consisting of known and unknown gestures.

In this thesis, we will focus on the recognition problem for both hand postures and gestures.

1.4 Contributions of the Thesis

The main contributions of the thesis are the following :

- **Overview of recent state-of-the-art techniques for gesture-based human-computer interaction** : no survey on recent state-of-the-art techniques for the development of vision-based gestural interface has been done lately. This overview presents in a concise way the different approaches. This study highlights progresses made in the development of vision-based gestural interfaces as well as challenges that still remain, particularly in the field of two-handed gesture interaction.
- **Databases and experimental protocols** : there is a lack of common evaluation databases in the field of HGR and HPR. We first defined two experiment protocols for an existing hand posture recognition database. Then, two hand gesture databases were collected and again experiment protocols were designed. All these resources are made publicly available to the research community for benchmark purposes.
- **Hand posture detection and recognition** : we propose a novel technique for the detection and recognition of hand postures in still images. The proposed approach is based on a new feature extraction technique based on the Modified Census Transform (MCT). Detection and recognition results are performed using a benchmark database in the field of hand posture recognition (HPR). Comparative results with a baseline approach are also provided.
- **Experimental evaluation of state-of-the-art techniques for HGR** : Hidden Markov Models (HMM) and Input-Output Hidden Markov Models (IOHMM) are state-of-the-art techniques in the field of HGR. However, no direct comparisons of these methods on common gesture databases using well defined protocols are available. We propose to evaluate these two models on two publicly available hand gesture databases using well defined protocols.
- **Focus on two-handed gesture recognition** : two-handed gestures improve computer interaction. However, little research has been done to recognize two-handed gestures. We propose to investigate the use of the above state-of-the-art HGR techniques on a new publicly available database of two-handed gestures. We particularly study the effect of using one or two cameras and one or two hands on recognition performance. The choice of features for better performance is also studied.

For all problems, we followed strict experiment protocols and we used publicly available databases to provide as much as possible unbiased results and fair comparisons.

1.5 Organization of the Thesis

This thesis is organized as follows. Chapter 2 provides an extended study of recent state-of-the-art approaches for the main problems of gestural interfaces, namely hand segmentation, tracking and recognition of hand postures and gestures. Results of the literature on two-handed HCI are also presented. Motivations behind the use of two-handed gestures for vision-based gestural interfaces as well as new challenges related to such interfaces are introduced.

Chapter 3 contributes to HPR. A method for the detection and recognition of hand postures in still images is presented. This method has already been applied to face detection with success. This technique is evaluated on an existing benchmark database of hand postures. Comparative results with baseline approaches are also provided.

Chapter 4 focuses on HGR. This chapter provides a description of the first database of hand gestures. This database contains both one- and two-handed gestures performed in the 3D space. We evaluate on this database two models used with success for HGR, namely Hidden Markov Models (HMM) and Input-Output Hidden Markov Models (IOHMM). We present and discuss the results obtained with these two models on this first database.

Chapter 5 is dedicated to two-handed gesture recognition. Given the lack of database in HGR and especially for two-handed gesture recognition, a second database has been recorded. This database contains only two-handed gestures, recorded using two cameras. The goal of these gestures is to manipulate virtual objects on a screen. We investigate the state-of-the-art sequence processing techniques presented in the previous chapter on this second database. We provide results using different features, and using features extracted from the images of one or two cameras, and using features of only one hand.

Chapitre 2

State-of-the-Art

The aim of this chapter is to introduce the reader to vision-based techniques used in the development of gestural interfaces. The chapter addresses the three specific problems introduced in Section 1.3 : Segmentation, Tracking and Recognition. This chapter also presents some applications and focuses specifically on two-handed gestures.

2.1 Segmentation of the Hand

Hand segmentation is the process which extracts the hand(s) from the rest of the image. This is a complex task due to the presence of complex background, changes in lighting conditions, and the shape of the hand itself. To reduce the problems encountered during the segmentation process, a variety of restrictions are usually enforced : restriction on *background*, restriction on *user*, and restriction in *imaging*. Restrictions on the background and in imaging are the most commonly used. A controlled background greatly simplifies the task. It can vary from a simple light background [129, 14] to a dark background [27, 65]. Most of the time, a uniform black background is used. Additional restriction on the user also simplifies the localization problem. For example, the user can wear long sleeves [48, 74]. In the case of restriction in imaging, cameras are focused on the hand [54, 94, 1, 6, 193, 14]. Nölker and Ritter have a radical position as they ask the user to put her/his hand in a box [134].

Another way to simplify this problem is to adorn the user's hand(s) with glove(s) [86, 2]. Well-chosen colors [97] can greatly help the segmentation task. Color-coded gloves and markers can also help in the segmentation and recognition of the fingers, fingertips and bending joints [156]. Although from a computational point of view these methods are easier to implement, they tend to reduce the naturalness of the interaction.

In less restrictive conditions, color analysis can be employed. This widely used technique [118, 46, 123, 169, 20, 4, 191] consists of modeling the color distribution of the human skin. This can be done using some simple histogram matching, such as in [191]. Skin color can also be modeled using mixtures of Gaussians [94]. These approaches assume that homogeneous objects in the image manifest themselves as clusters in some measurement space representing colors. Such spaces can be RGB (red, green and blue components) [23], normalized RGB [74], YUV space [136, 146], HSI (hue, saturation and intensity model) [196], HSV domain [25]. McAllister et al. propose to use background and foreground appearance models. These models are learned on-line leading to more robust segmentation results [119]. Some authors [89, 36, 70, 180] use infra-red light/cameras and even "black light" source with white paper fingertip markers [87] to facilitate hand segmentation.

Background subtraction [43, 81, 103, 106, 112, 70, 173] is also one of the main techniques used to segment the hand in images. Grzeszcuk et al. propose a disparity based background subtraction algorithm. Pixels which have a smaller depth compared to the mean background depth are segmented as being a hand [52]. This technique makes the assumption that the hand of the user is the only element in the view field of the camera. Background subtraction and color information can be used jointly [12, 29, 149]. The motion analysis of the scene can also help in the segmentation process [188]. The moving artefacts are supposed to be mostly produced by hand movements and can thus be used to segment the hand [93, 51, 178]. As for background subtraction, motion and color cues [195, 101, 186, 3, 79] can be associated for better results. Some other techniques take advantage of edge detection [13] which can also be coupled with color detection for more robust hand segmentation [168, 166]. All these techniques can be used in common for more robust segmentation systems, such as in [32] where motion, skin color and edge detection are used together.

2.2 Tracking

Human hand can be though of as an articulated object. This is valid since the deformations of the human hand skin do not convey any additional information needed to interpret gestures. In the literature, there exists two main approaches to solve the problem of hand gesture modeling : by using a 3D model of the hand, or only based on the appearance of the hand in images.
In the appearance-based approach, the hand states are estimated directly from images. The image feature space is thus mapped to the hand configuration space. In [25], a pattern-matching algorithm is used to detect and track the features. These features are the fingertips, four spaces between the fingers and wrist points. This approach assumes a fixed orientation of the hand, with the wrist and forearm situated on the right border of the image.
Most of the recent research on hand motion tracking is focused on 3D model-based approach. In that case, a reduced set of joint angle parameters together with segment lengths is used. The reduction is accomplished using several assumptions. Such assumptions introduce dependencies between different joints and also impose bounds on the moving range of joint angles. Such constraints are due to the anatomy of the hand. Furthermore, Huang et al. propose to collect data from a DataGlove to further reduce the configuration space [68]. 3D hand models are usually represented as a set of quadrics [162, 166], using a cardboard model [181] where fingers are represented as sets of planar patches, or through the use of more complex model such as in [19] where the hand model consists of polygonal skin with an underlying skeleton. Information extracted for tracking is usually based on edge detection [162, 68]. Edge detection can be combined with silhouette [107], color [166], or optical flow and shading information [110]. Zhou and Huang propose the likelihood edge feature, which is edge gradients on the likelihood ratio image [192]. Depth maps can also be used to have more precise data [19]. Tracking can thus be performed using particle filtering [19, 28], Kalman filtering [162], sequential Monte Carlo tracking [181] and graphical models [192, 166]. In most cases, images are focused on the hand and are recorded against dark background. To further simplify the task, the assumption is often made that the hand has very little global motion. And some methods even assume a fixed orientation of the hand [181, 25].

Another problem is the tracking of the hand as a whole in a sequence of images. Some simple information such as skin color and motion information can be used. Marcel proposes to utilize skin color blobs. The blob is defined as a region of interest in an image, and is extracted based on the location (x, y) and colorimetry (Y, U, V) of the skin color pixels. It permits to determine homogeneous areas. A skin color pixel belongs to the blob which has the same location and colorimetry component [118]. Yao and Zhu propose a Markov model to estimate and predict the skin color distribution. This skin color distribution is thus combined with a background color distribution and they are used to segment the hand in the next frame [185]. Skin color detection can be used in conjunction with motion heuristics [128]. This permits to define a search window based on the position of the hand in the previous frame and thus restrains the skin color detection to a definite zone in the image. If we make the assumption that hand gestures are non-stationary, motion information and skin color can be employed together to track the human hand in images [32]. Motion only can also provide enough information for hand tracking. In [149], tracking through a sequence of frames is done by computing a parametric motion model for consecutive images. In [178], optical flow techniques are used to capture the hand motion. Simple approach such as image differencing has been applied to the problem of hand

and fingertip tracking [104]. Image differencing segmentation and shape filtering have been proposed in [103] to track fingertips over a desktop. This method makes the assumption that the tracking will occur in a controlled environments.

An important technique used for hand tracking is active contour models also called Smart Snakes [60]. Tracking is achieved using the deformable active shape models. A contour which is roughly the shape of the hand is placed in the image, close to the object to track. The contour is attracted to nearby edges (due to changes in intensity) in the image and can be made to move towards these edges, deforming (with some constraints) to exactly fit the object (here, the hand). The process is iterative, with the contour moving in small steps. 3D active contour model has been proposed in the case of stereo vision [189]. The contour evolves in the 3D space due to a 3D potential function derived by projecting the contour of the hand onto the 2D stereo images. In [75], a switching linear model is combined with an active contour model using B-splines.

The main technique used for hand tracking is Kalman filter [87, 50, 71, 126]. Kalman filter has been proposed for bi-manual hand tracking [151]. By capturing the hands velocity and recognizing the hand synchronization, the system is able to handle occlusion problems. In [23] a spatio-temporal blob consistency graph is used in case of occlusions. Kalman filter can also be combined with other methods such as optical flow methods [144], color and motion [101]. Particle filters are another category of algorithms that has been widely used for hand tracking [173, 161, 29]. Variants of particle filters include hybridization with local search algorithms [139] and mean shift [153]. In [96], the hand is modeled as a mixture of Gaussians representing the palm, the fingers and fingertips respectively, and tracking is done using particle filter. CONDENSATION algorithm [109, 20, 194, 115] has also been used to track one or two hands in images.

Other tracking methods have been proposed such as mean shift algorithm [94] or flocks of features [91] which integrates optical flow and color probability distribution. Few methods have been proposed for the tracking of both hands. Barhate et al. propose a method based on EigenTracking that can handle simple occlusion situations [8]. In [120, 119], a 2D geometric model of the hands and forearms is presented. The hand is represented by a circle and the forearm by a segment. It can handle simple crossover gestures. Huang et al. track two-handed gestures using template-based method and dominant motion estimation, but hands should avoid occlusions with each other and with the head [69]. These last examples show that occlusion handling in the case of two-handed gesture tracking still remains an unsolved task.

2.3 Recognition

Gesture recognition can be decomposed into two main tasks : the recognition of gestures and the recognition of postures. Hand Posture Recognition (HPR) can be accomplished using template matching, geometric feature classification, neural networks, or other standard pattern recognition techniques to classify poses. Hand Gesture Recognition (HGR), however, requires consideration of temporal events. It is a sequence processing problem that can be accomplished by using Finite State Machines (FSM), Dynamic Time Warping (DTW), Hidden Markov Models (HMM) to cite a few techniques.

2.3.1 Hand Posture Recognition

Feature Extraction

The goal of feature extraction is to find the most discriminant information in the recorded images. HPR has to deal with the problem of cluttered/uncluttered background and changes in lighting conditions. For posture recognition, features such as fingertips, finger directions and hand's contours can be extracted. But such features are not always available due to self-occlusion and lighting conditions. Feature extraction is a complex problem, and often the whole image or transformed image is taken as input. Features are thus selected implicitly and automatically by the recognizer.

The position of the fingers and palm has been widely used for pose recognition [111, 174, 184, 87]. Other information on the hand can be computed such as roll, pitch and yaw angles [146], elongation, roughness, compactness and pixel value vector from down-sampled hand regions [73]. In [55, 54], 256 points are sampled on the contour of the hand region. These points are placed at constant interval and for each point the scale-normalized distance is computed. The scale-normalized distance corresponds to the normalized distance between the center of gravity and one of the points along the hand curve.

Another main category of features that can be used for posture recognition is statistical moments [100, 52]. Zernike moments [158], Hu moments [197] and more generally geometric moment invariants [187] have been proposed for HPR. Fourier analysis can also be applied for the extraction of meaningful features. Fourier descriptors can be used to represent the boundary of the hand [131, 106]. The phase information in the Fourier domain [168] can also be used for HPR.

The problem of robust pose classification becomes less difficult if the task of hand shape identification is separated from the determination of hand orientation. Features based on the curvature scale space [26, 27] are translation, scale and rotation invariant.

In [176, 175], features consist of 3×4 block intensities which are the ratio of white to black pixels in each block. Hand's silhouette can be used for posture retrieval [157]. In [6, 5], features are all local maxima in the direction of the image intensity gradient, combined with edge orientations histogram, fingertip position, central and Hu moments. Other features based on the Modified Census Transform (MCT) or Haar like features, firstly applied to face detection, have been proposed for pose detection [144, 92, 80].

Recognition

Once features have been computed, classification of hand postures can be performed. One simple way to classify hand postures is to count the number of fingers in the image [87, 130]. But posture recognition can also be seen as a matching process. In that case, clustering algorithm can be successfully used. In [26, 27, 193, 106] a nearest neighbor algorithm is performed. Fuzzy classification algorithms can also be applied [176, 175, 158], namely the fuzzy C-means algorithm. Self-organizing maps and variants can be used [135, 61].

Neural networks and extensions are another category that has been widely used for pose classification and detection [118]. Two-layer and three-layer perceptrons have been used in [73, 184]. Radial basis function networks are another class of neural networks than can be applied to HPR [100, 131].

The posture recognition problem can also be seen as a database retrieval problem. In that case, recognition consists of finding the best match among images representing possible poses. Hand models are often generated by computer graphics [157]. Hand's silhouettes are then extracted and matched using the Chamfer distance algorithm [6, 5]. Techniques used in information retrieval such as the Okapi-Chamfer matching algorithm have been applied to the problem of HPR [191].

Elastic graph matching [171] has been proposed for the classification of hand postures against cluttered backgrounds. Hand postures are represented by labeled graphs with an underlying two-dimensional topology. This approach can achieve scale-invariant and user-independent recognition and does not need hand segmentation. Another approach is taken in [37]. Hand configurations are classified in a space defined by a principal component analysis of the distribution of hand images. In [42], a two-layer classifier is applied. The first layer performs a crude classification using PCA whereas the second layer gives a more precise classification using MDA.

Some recent techniques applied with success to face detection have also been applied to the HPR problem. The Viola and Jones system using Haar like features [78] has been applied to HPR in [144, 92]. In [80], boosted classifiers based on the MCT are used for posture recognition. These features have been firstly used for face detection [49].

2.3.2 Hand Gesture Recognition

Feature Extraction

Recognition of temporal gestures not only needs spatial features, but also requires temporal information and so temporal features. Even if it is possible to recognize some gestures by 2D locations of hands, it is not general and view-depended. To achieve spatial invariant recognition, 3D features are necessary, but features for temporally invariant gesture recognition are hard to specify since they depend on the temporal representation of gestures. However, it can be done implicitly in some recognition approaches (such as FSM and HMMs).

Spatial and temporal information are some simple features to compute. Thus they have been widely used for HGR [149, 3, 197, 132, 86]. Trajectory of the hand center, orientation and velocity, variation in the shape of the hand region have been proposed [186] . Fuzzy trajectories of the hand gestures have also been proposed [79]. Montero and Sucar employ the position and velocity in polar and Cartesian coordinates as features [128].

Temporal template for gesture representation and recognition can also be used [15]. A temporal template consists of an image where the value at each point depends on the motion properties at the corresponding spatial location in an image sequence. This feature is based on motion-energy images and motion-history images that have been used in [152]. A new history-based representation, namely skin history images and composite history images has been proposed in [123, 122].

Fourier analysis is a domain used for feature extraction. Yang et al. perform Fast Fourier Transformation as preprocessing tool. The feature vector extracted from the signal is a set of 16-dimensional vectors obtained from the amplitude of FFT coefficients. They convert the feature vectors into finite symbols using a vector quantization technique [182]. Fourier descriptors have been applied to feature extraction [57, 32] .

Gestures have been represented in many other ways, using the appearance model [47] firstly proposed in [35]. The shape is represented through a point distribution model. A model network is created that links the corresponding single models. Gabor coefficients [66, 67], discreet cosine transform [129] have been proposed. In [150], the eigenspace of each individual gesture is built. Every input sequence is thus projected into its own subspace to get the manifold of that subspace. Recognition is performed using graph matching as each manifold can be represented as a directed graph.

When the trajectory of the gesture is the relevant feature, interpolating functions can be used. For character drawing recognition, set of uniform cubic B-splines can represent character trajectories [109]. Bezier curves [155] can be used to analyze and classify trajectories. Classification is thus performed using the angle and direction of the trajectory.

Recognition

Dynamic Time Warping (DTW) and variants [147, 3, 188, 109] have been proposed for HGR. Sequence processing has been first tried using finite state machines (FSM) and variants [50, 14]. But the most important technique is Hidden Markov Models. HMMs have been widely used for gesture recognition [128, 197, 142, 132, 86, 32, 129]. This approach is inspired by the success of the application of HMMs in the speech recognition and hand-written character recognition fields [140]. HMMs have been used by [182]. Ko et al. model gestures as a sequence of postures. Each posture is recognized using support vector machines and gesture recognition is done using HMMs [88]. Marcel et al. propose to use IOHMM which is an extension of HMMs [118].

Neural networks and variants have also applied to the problem of gesture recognition [57, 79, 127]. Time delay neural network [183, 66] have been proposed. Flòres et al. use a self-organizing neural network. Its topology determines postures and its adaptation dynamic through time determines gestures [48]. Other fields of machine learning have been explored : Bayesian classifiers [7], switching linear model [76]. Other approaches have been proposed for HGR. The two-handed gestural system proposed in [151] can handle occlusions. They made the hypothesis that hands are synchronized effortlessly. Thus the synchronized hand velocities and simultaneous hand pauses are exploited. Temporal analysis of velocity permits the construction of a mixture of Gaussians distributions to model the hand-pause and hand-pass behaviors. In [139], a fixed window is used for tracking. If the user closes the hand, the lower half of the window remains full of pixels and the upper half empty. In case the user rotates the hand of 90 degrees, the left half is full of skin pixels and the right half remains empty. In [56, 55], a gesture is decomposed into a sequence of poses. A shape transition network is proposed for posture mapping.

Gesture Segmentation

Gesture segmentation or spotting consists of determining the start and end point of a particular gesture. In addition, unknown gestures are rejected. The problem of gesture spotting comes from the co-articulation between two gestures. In order to simplify the problem, most techniques ask the user to stop, slow down or have a definite pose at the end and beginning of a gestures. Very few techniques let the user act naturally. Moreover, really few work has been done in the field of gesture segmentation. Most of the time, research is focused on the recognition of segmented gestures, avoiding the problem of gesture spotting.

In [39], each gesture was defined with an explicit start position, which makes the segmentation process easy. The gestures has a definite progression, with four distinct phases easily recognizable. Zhu et al propose a spotting technique relying on the three phases of a gesture : propagation, stroke and retraction [196, 195]. A gesture must follow a three-stage process : starts from rest, moves smoothly for a short period and slows down. Gestures must also not be longer than a certain number of frames. The user can also stand still for one or two seconds at the start and end of each gesture [186].

In [159, 160], network of HMMs are used to recognize a sequence of gestures taken from the American Sign Language. All that was necessary for training was a data stream and its transcription (the text matching the signs). The training process was automatically able to align the components of the transcription to the data. They used language modeling to segment the different words. The Viterbi decoding algorithm was both used with and without a strong grammar based on the know form of the sentences. An HMM-based threshold model has also been proposed [99]. The authors introduce a threshold model that calculates the likelihood threshold of an input pattern. A gesture is recognized only if the likelihood of the best gesture model is higher than of the threshold model. They propose a spotting gesture network where each gesture is represented by an HMM. Finding the start and end point is done using the Viterbi algorithm. In conjunction with the threshold model, it permits to reject unknown gestures.

In [147], continuous dynamic programming permits to perform gesture spotting. In the same way, Alon et al. propose a spotting and recognition system based on the continuous dynamic programming algorithm. A pruning method is proposed that allows to evaluate a small number of hypothesis and greatly speeds up the process [2].

If gestures are represented as an ordered sequence of postures [83], spotting can be performed in two steps : 1) candidate cut generation step : it detects candidate cuts, i.e. possible gesture ends, using contextual information (velocity, static gesture), 2) gesture spotting step : definite gestures are distinguished by tentative gestures using the recognition values of the tentative gestures in a sliding window (size empirically determined). In [116], Marcel used GIOHMM, which is an extension of the IOHMM model, in order to segment deictic from symbolic gestures. The system was able to recognize with low error four distinct gestures.

2.4 Applications

2.4.1 Virtual and Augmented Reality

Virtual and augmented reality are two fields in which gestures can be used to improve user's interaction. In the case of virtual reality, the user is surrounded by computer generated images she/he can interact with. Hand gestures can be used for the exploration of 3D scenes and objects [23]. As the mouse is not a very intuitive device for virtual reality, they also propose to use the right hand for pointing and the left hand for click and control actions. Hands are thus behaving as a normal point-and-click GUI interface. In [18] two-handed pointing gestures are applied to the visualization and interaction in 3D environments. But augmented reality has been lately the most important field of research for the use of hands to interact. The user is wearing glasses which enhance its view of the real environment, through the surimposition of virtual objects. Interacting with such environment can be done using *patterns*. A planar pattern is tracked in real-time and virtual objects are augmented on top of this pattern. This tool can also be employed as a mouse. In [113], the pattern is used to emulate point and click gestures. In [121], the combination of a finger and pointer location is viewed as a point gesture, and a selection gesture is the combination of multiple fingers and a pointer location on the pattern. Pointing gestures are also a class of gestures that could be used for interaction [62], such as the control of menu using fingertip. Another type of gesture that could be used for more natural interaction are symbolic gestures. In [184], hand postures are used to control an augmented reality map navigation system. When the hand postures are recognized, information concerning the geography, video clips or 3D scenes are surimposed on the map.

2.4.2 Wearable Computing

Another main field that has attracted lots of interest and which is related to augmented reality is the field of wearable computers. The user is equipped with glasses, a computer-backpack and is able to interact with the environment. In the system Snap&Tell [44, 84], the user can specify objects of interest by pointing at and encircling them with the fingertip. Then information concerning this objects are retrieved and surimposed on the real world. In [108, 109], a text input method based on the Graffiti2 alphabet is proposed. The user only needs her/his fingertip to write in the air the character. Both hands can also be used to take snapshots of selective areas [90]. The selected area is the rectangular area which is enclosed by both hands. But due to the fact that such systems are mostly used outdoor, they can have problems with hand segmentation. Gloves with special markers invariant to lighting can be used to overcome such limitation [156].

2.4.3 Hand Gesture-Based Graphical User Interface

For more classic interaction, hands can be used to draw or they can replace the mouse. Drawboard [96] is a drawing tool that uses hand motions to control visual drawings. The cursor on the screen is controlled by the position of the hand. In the same way, fingertip motion can be used to draw [1]. Hand postures are seen as commands by the computer. The SmartCanvas system [126] allows the user to perform freehand drawing on a desk or similar surface. Hand gestures can also be applied to character drawing [149]. Replacing the mouse is also one of the main applications of hand gesture interaction [178]. The mouse-click event can be modeled in many different ways. In [71], the contact between the thumb and index fingers corresponds to a mouse-click event. The position of the thumb can also be used as an action command [25]. Closing the hand and rotating the hand 90^o can model the left and right button click respectively [139]. Browsing the Net can be performed using fingertip motion [164]. But more generally, hand gestures are used to control windows [131]. In that case, other modalities can be added, such as speech recognition [179]. Two systems quite similar have been proposed in [190] and in [112] : the Visual Panel and the Visual Touchpad. They are both based on a quadrangle-shaped panel on which a fingertip can be used as an input device. In the first case, the system simulated a click event when the user holds her/his finger still in the same position for a short time. They apply the system to the control of a calculator and a virtual keyboard. In the case of the Visual Touchpad, the panel is more constrained as it is a black rectangle surrounded by a white border. They used this input system for picture manipulation, using both one- and two-handed gesture interaction. This system has been extended in [114] to the interaction with large display.

2.4.4 Augmented Workplaces

But gestures are also of great interest in the case of augmented workplaces, e.g. when projected informations are added to the environment. The EnhancedDesk [89, 33] is an augmented desk which integrates both paper and digital information. It can be used for computer-supported learning. Fingertip interaction is used for digital information generation. It can also be used for two-handed drawing where the right hand is used to draw and the left hand is used to manipulate menus. Hand gestures can be used in video-surveillance control rooms [70]. The user uses her/his finger to interact on a plane surface on which video streams are projected. The PlayAnywhere system [180] is a form-projected interactive table system. Fingertip is used to navigate through the interface and hands could be used to manipulate objects. In the case of interaction with information projected on a wall, the hand can emulate a virtual cursor [87]. In the same way, the VirtualBoard [36] is a window desktop projected on a wall. The position of the cursor is controlled by the hand moving over the screen. Selecting, dragging icons, opening applications are possible through tracking of the user's fingertip. These applications are mostly related to the interaction with large display which is mostly done using pointing gestures or through fingertip tracking. Other applications have been proposed such as a transport management interface [31]. A single hand is tracked and its location is used to point at certain region on the screen. Pointing gestures can also be used to determine location on a large display [34].

2.4.5 Interaction with Large Displays

Related to augmented workplaces, interaction with large display can also be used for data visualization and manipulation. Means of interaction are thus not only pointing gestures, but allow direct manipulation of data. Hand gestures can be used for a 3D virtual Fly-thru system where the user flies over a graphically generated terrain [148]. Or it is possible to browse a panoramic map [196, 195] using gesture commands. Hand gestures are of great interest for visualization of complex or/and large data sets [172]. Some fields are biomolecular systems [154], 3D bioinformatics data [155] or medical image visualization [130]. For data visualization on large displays, hand gestures are a very powerful mean of interaction.

2.4.6 Robotics

Other field of potential interest for hand gesture recognition is robotics. Hand gestures can be firstly used to communicate with robots [175], to share knowledge about the environment by pointing at objects. But it has mainly been applied to control robotic hands. Hand gestures can be used to control a robot gripper [98]. The gripper replicates the hand posture changes of the user. Similarly in [65, 72], a robotic hand is able to mimic human hand poses. In [72], the robotic hand is able to grasp a plastic glass. Hand gesture recognition can also be expanded to the interaction with electronic appliances. In [81] pointing gestures are used to interact with a robotic chandelier. In [21], hand postures permit to turn the television on and off, change channel. More generally, the Soft-Remocon-System [74] has been developed to facilitate the management of home appliances for disabled people.

2.4.7 Other Applications

Of course, more entertaining applications have also been proposed. Gestures have been used in [83] to interact with the game QuakeII, and the "Rock, scissors and paper" game has been proposed [45]. Hand gestures have also been used for sound and graphic generation and manipulation [63, 25, 127, 105]. It is also possible to play an imaginary piano using hand gestures [167].

But most of the applications described previously only make use of one hand. The main field of research which deal more often with bi-manual hand gestures is Sign Language Recognition [138], which is not directly related to HCI. Nevertheless, two-handed inputs can be a great addition to human-computer interfaces. Next section gives a summary of different studies that have been effectuated on the subject of two-handed interaction, showing the benefits expected from two-handed inputs.

2.5 Two-Handed Interaction

In our everyday life, our activities mostly involve the use of both hands. This is the case when we deal cards, when we play a musical instrument, even when we take notes. In the case of HCI, most interfaces only use one-handed gestures. In [146], the user executes commands by changing its hand shape to handle a computer generated object in a virtual reality environment. Wah and Ranganath propose a prototype which permits the user to move and resize windows and objects, open/close windows by using simple hand gestures [177]. Even with common devices, such as the mouse or the graphic board, only one hand is used to interact with the computer. The keyboard seems to be the only device that permits the use of the two hands in the same time.

But using two-handed inputs for computer interfaces can be of potential benefit. Many experiments have been conducted to test the validity of two-handed interaction for HCI. The obtained results are very encouraging. In [22], the authors ran two experiments to investigate two-handed inputs. The first experiment involves the performance of a compound selection/positioning task. The user is asked to position a graphical object with one hand and scale its size with the other hand. For that purpose, a graphic tablet and a slider box are used. This first experiment shows that performing parallel tasks is a natural behavior of the user for that particular task. Furthermore, they show that the efficiency correlates positively to the degree of parallelism involved in the task. The second experiment involves the performance of a compound navigation/selection task. The user is asked to select particular words in a document. Authors compare the one- versus two-handed techniques. The conclusions are again favorable to two-handed interaction as the two-handed method significantly outperforms the commonly used one-handed method. Furthermore, using the two-handed approach can reduce the gap between expert and novice users. The overall conclusion that Buxton and Myers draw from these two experiments is that performance can be improved by splitting tasks between the two hands. Furthermore, the performance improvement can even occur when the two hands interact sequentially and not only in parallel. Leganchuk et al. conducted some more experiments on the benefit of two-handed input. Their experimental task was area sweeping, which consists in drawing a bounding box surrounding a set of objects [102]. Their conclusions support the fact that two-handed techniques outperform the conventional one-handed technique. Bi-manual techniques are faster and for high-demanding tasks, the advantage of two-handed input over one-handed input becomes more obvious.

In his thesis, Sturman emphasizes the fact that it is necessary to use the skills of the user [165]. More recently, the Sato Laboratories[1] proposed an augmented desk interface system which provides man-machine interfaces based on direct manipulation of both real and projected objects with hands and fingers. Their system uses an infrared camera to track and recognize hand gestures. They developed a two-handed drawing tool [33] which permits the user to draw and manipulate objects interactively. The right hand was used to draw and manipulate objects, and the left hand was used to manipulate menus and to assist the right hand. Some interesting experiments were also conducted in [59]. The use of speech and gestures for graphic image manipulation was studied. Seven different operations involving rotation, transposition and scaling were defined. People were allowed to use whatever gesture/word to perform the task and the recognition process was carried out by a human. The conclusions relative to hand gestures are that any system which would restrict the user to a single hand would be inadequate for this manipulation task. Furthermore, restricting users' gestures to two dimensions would limit the expressiveness of hand gestures. In his study of situation calling for two-handed input, Bolt and Herranz propose graphical manipulations (object rotations), description of layout (description of the relative placement of items) and specification of actions [17]. The impact of this paper needs to be emphasized as the prototype system used two DataGloves, showing the effectiveness of *free* hand interaction. Two-handed interaction is possible and efficient even when not using dedicated devices.

[1]http ://www.hci.iis.u-tokyo.ac.jp/research/EnhancedDesk

An even more refined system has been proposed in [38]. The system, called the Responsive Workbench, is a virtual environment based on a tabletop display system. The user, wearing DataGloves, was able to interact directly with 3D virtual object using one or two-handed inputs. Users found this interaction system easy to use and natural. The system is very intuitive as no long training is necessary to reach a high degree of proficiency.

This leads to a second important advantage of two-handed interaction. Tasks can be made easier to learn and master by taking advantage of pre-acquired skills, reducing training expense and time [165]. Furthermore, two-handed gestures are body-centered. Body-centered coordinate systems tend to be more natural to work with and they can improve performance for objects manipulation tasks. In the case of body-relative interaction, the user takes advantage of proprioception (which is the sense of position and orientation of its own body [125]). Such techniques provide spatial reference in which to work, a more direct and precise sense of control and the user does not have to constantly monitor visually what he is doing. Using both hands can help the user gain a better sense of space and increase the degree of manipulation [64]. Furthermore, when using both hands, the user's attention is not anymore mainly focused on the manipulation itself. Her/his visual attention can then be directed to a secondary task such as watching the effects of her/his manipulation from a different viewpoint. One-handed techniques tend to rely more on visual feedback. Then it can prevent the user from splitting her/his attention between multiple tasks.

Nishino et al. showed that two-handed gestures tend to be more stable and reliable in case of similar gesture patterns [133]. Two-handed gestures can help developing more robust gestural interfaces. To design such two-handed systems, some guidelines are necessary. Work by Guiard is very helpful in that respect. Guiard proposed to model asymmetric two-handed gestures as a kinematic motor chain [53]. Let suppose that the user is right-handed, then her/his two-handed gestures follow these principles :

- The left hand sets spatial references for the motion of the right hand. For example, when we write, the left hand guides the paper sheet.
- The sequence of motion is first left and then right hand. The left hand first grabs the paper and then the right hand starts writing with the pen.
- The right hand is capable of a finer temporal and spatial resolution than the left hand. That is the reason why when learning to play an instrument, it is always more difficult to reach virtuosity with the left hand.

Kabbash et al. showed that techniques which are conform to Guiard's principles will impose a lower cognitive load on users [82]. It will lead to faster and more effortless interaction paradigms. But they also showed that techniques which assign independent subtasks to each hand can be worse than one-handed techniques. Such interaction would be very similar to the "tapping the head while rubbing the stomach" approach which is very difficult, even for trained people. Some guidelines for the development of two-handed interaction have also been proposed in [141]. Three types of interaction are defined :

- **Independent interaction** : in that case, the two hands could be seen as independent pointing devices. This type of interaction is only relevant if the two hands are working on the same task. One can think of the simple task of selecting and moving icons.
- **Parallel interaction** : this type of interaction is very natural. We are used to perform secondary tasks with the left hand, such as reaching for tools and passing it to the right hand. But systems using this type of interaction must not force the user to act sequentially. It would restrain the naturalness of such interaction. This type of interaction can be found in computer gaming, where hands are working together but independently.
- **Combined interaction** : such interaction can be found when the left hand provides support and/or strength to the right hand. The left hand holds the element still while the right hand effectuates an action on it, such as stretching. Combined interaction can also be found for critical operations, when commands are required in the same time.

Chatty mentioned in [30] that every interface will need to be both available in the form of one- and two-handed interaction. But today's interaction with the computer is mostly done using one hand. Review of the literature has shown the benefit that can be expected from two-handed gesture interaction, particularly for object manipulation. Furthermore, guidelines for the creation of two-handed interaction for enhanced HCI exist. Developing gestural interfaces using two-handed gestures is thus one of the future challenges of HGR.

2.6 Discussion

Several methods have been proposed to solve the three main problems of vision-based gestural interfaces, namely hand segmentation, tracking and recognition. Each of these methods makes assumptions. This section intends to show the strength and weaknesses of these methods as well as future challenges of vision-based gestural interfaces.

For hand segmentation, techniques using restrictions on the user (long sleeves, gloves, colored markers on the hand) or background (controlled background) tend to reduce the naturalness of interaction. Furthermore, techniques based on background subtraction and/or skin color detection ask for controlled environments (no dramatic changes in background and lighting conditions). Interaction using on-hand focused images is only possible with tabletop systems. Motion analysis assumes that the hand is the fastest moving object in the image. It is not a problem with images focused on the hand. Difficulties arise when the hand is not the main object in the image. Thus moving objects visible in the background can be segmented as a hand. Same problem arises with edge detection. Hand segmentation using edge detection is more complex for images against complex background. Robust hand segmentation thus necessitates to use in conjunction several techniques. Furthermore, it would also permit to relax restrictions on the environment and on the user. For tabletop systems or systems placed in controlled environment, hand segmentation gives accurate results. Hand segmentation remains a problem for systems in public places due to changing in background and lighting conditions.

The hand tracking problem is closely tied to the hand segmentation task. Thus systems designed to track finger motion are mostly trained and tested using images recorded against dark background to ease the segmentation process. Unfortunately, this reduces the applicability of such techniques to real life systems. Furthermore, such systems make the assumption that the hand has very little global motion. Some systems even ask for a fixed orientation of the hand. Both these hypothesizes reduce the naturalness of interaction. Finger motion tracking is a difficult problem and work still needs to be done to have robust and efficient tracking systems. Another problem is the tracking of whole hand(s) in a sequence of images. Techniques such as contour models are very sensitive to noise, particularly when a complex background is present. Thus it needs to be coupled with an efficient and robust segmentations algorithm. Kalman filter and Sequential Monte Carlo techniques give good results. But Kalman filter is very sensitive to abrupt changes in the position of the hand, and sequential Monte Carlo methods are not feasible for real time applications in cluttered environment due to the high number of samples required. Tracking methods under constrained environment give good result but are not applicable to real life applications. Another problem remains which is the tracking of the two hands in images. The main problem of bi-manual tracking is the presence of occlusions that can occur when one hand is hiding the other one. Simplifying hypotheses have been proposed, such as prohibiting any occlusion when interacting with both hands. But it reduces the possibilities of interaction as well as the naturalness. Bi-manual tracking has not been yet the subject of lots of research. It seems that only recently this problem has attracted attention of researchers, leading to novel approaches [151]. But two-handed gesture tracking still remains an unsolved problem and the task of bi-manual tracking is a future challenge for vision-based gestural interfaces.

In the field of HPR and HGR, several methods have been developed leading to good performances. However, hand postures are often recognized against simple background, reducing the applicability of such techniques to real life systems. Moreover, most of the HPR techniques are orientation dependent. Hence development of orientation independent methods is the next step towards a better and more robust recognition. Most of hand postures to recognize are taken from the Sign Language. No real study has been performed on the usability of such hand postures for HCI. Some research has been done to find the most recognizable and meaningful hand postures [163] but work still needs to be done to define a set of valuable hand postures for HCI. One problem that arises with both hand posture and gesture recognition is the lack of publicly available databases with well defined experimental protocol. Publicly available databases are necessary to evaluate existing state-of-the-art methods and future approaches. It will create a common base for researchers to develop more robust systems.

HGR and HPR fields have seen a decrease of interest from the research community. There are still applications where gesture interaction is an improvement over standard interaction means. Such domains are virtual and augmented reality, wearable computing, interaction with large displays as well as data manipulation and visualization. Furthermore, these fields can gain even more from two-handed gesture interaction (Section 2.5). Another challenge in the field of gestural HCI is thus the integration of two-handed gesture interaction.

Chapitre 3

Hand Posture Recognition

This chapter presents a novel approach to the problem of HPR, as well as comparative results with baseline methods. The technique used in this chapter was firstly applied with success to the problem of face detection in images [49]. We first present the problem of HPR and some state-of-the-art methods. We then describe the proposed approach as well as the benchmark database. Two different protocols for training and testing the classifiers are proposed. Results using these two protocols are provided as well as a comparison with standard approaches.

3.1 Problem Description and Standard Approaches

Human hands can be seen as articulated objects. Fingers are articulated around their joints and the palm has some flexibility. The high number of DOFs of the human hand makes it a very deformable object. Hand shapes and hand postures can thus vary greatly. Figure B.1 shows three different hand postures where fingers can be both flexed and extended in the same time.

Fig. 3.1 – Examples of three different hand postures.

In the case of HCI, some specific hand postures are chosen. These postures define a vocabulary of commands used to interact with the computer. To use these postures, we need systems able to detect and recognize them in images. Many different techniques have been proposed in the literature (Section 2.3.1). Neural networks are a category which has been widely used for this task. Marcel used them to detect hand postures against cluttered background [117]. Triesch and von der Malsburg proposed the elastic graph matching technique for HPR. Hand postures are represented by labeled graphs [171]. Some more recent techniques [144] use the Viola and Jones approach based on Haar like features and also initially proposed for face detection [78].

3.2 Proposed Approach

This approach is inspired by the work of Fröba and Ernst for the face detection task [49]. Their system is a four stage classifier. Each stage consists of a linear combination of elementary classifiers.

These elementary classifiers are based on a local non-parametric pixel operator called the Modified Census Transform (MCT). A boosting procedure is both used for feature selection and classifier training. Fröba and Ernst used a cascade of classifiers to perform face detection in images. The first stage of the cascade is able to reject 99% of background images. The detection of faces only occur on the last stage of the cascade. Furthermore, the number of elementary classifiers increases with the level in the cascade. As our goal is the detection and recognition of several hand postures, we will train one classifier (or model) for each hand posture. Our approach differs also in the number of layers of the classifiers. Each classifier associated to a particular hand posture is composed of one single layer with a high number of elementary classifiers.

3.2.1 Feature Space

The Modified Census Transform (MCT) [49], or the similar Improved Local Binary Pattern (ILBP) introduced at the same time by Jin et al. [77], is an extension of the Local Binary Pattern (LBP) proposed by Ojala et al. [137]. The LBP is a non-parametric local transform which summarizes the local spatial structure of a 3×3 neighborhood of pixels. This operator shows a very high discriminative power for texture classification. At a given pixel position (x_c, y_c) in an image, the LBP considers the eight surrounding pixels. The output of the LBP (LBP code) consists of an ordered set of binary comparisons of pixel intensities between the center pixel and its eight surrounding pixels (Figure 3.2).

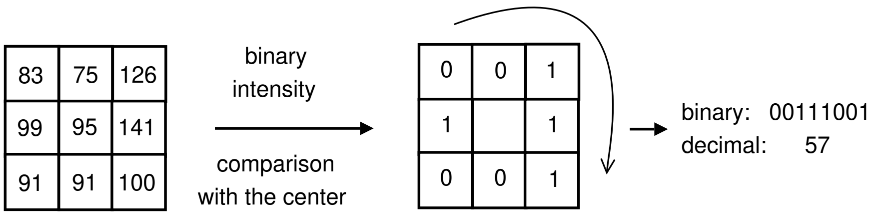


Fig. 3.2 – LBP operating on a 3×3 set of pixels, from left to right : original set of pixels, binary comparison of pixel intensities between the center and surrounding pixels, results of the transform and LBP code in the binary and decimal form.

The result of the LBP is a 8-bit string (LBP code) which is equivalent in its decimal form to :

$$LBP(x_c, y_c) = \sum_{n=0}^{7} s(i_n - i_c)2^n \tag{3.1}$$

where i_c corresponds to the grey value of the center pixel (x_c, y_c) and i_n is the grey value of n-th surrounding pixel for $n \in [0, \ldots, 7]$. The function s is defined as :

$$s(x) = \begin{cases} 1 & \text{if} \quad x \geq 0 \\ 0 & \text{if} \quad x < 0 \,. \end{cases} \tag{3.2}$$

By definition, the LBP operator is unaffected by any monotonic gray-scale transformation. Such transformation preserves the pixel intensity order in a local neighborhood (Figure 3.3).

Due to its texture discriminative property and its very low computational cost, the LBP has become very popular in the pattern recognition field. But Jin et al. noticed that LBP features miss the local structure under certain circumstances [77], and thus they introduced the ILBP or MCT operator (the differences between the ILBP and MCT operators reside in the order of the bit string and the comparison function). The MCT differs from the LBP in the sense that it consists of an ordered set of binary comparisons of pixel intensities between *all* the pixel intensities of the 3×3 neighborhood and the *mean intensity* of all the pixels of the neighborhood (Figure 3.4).

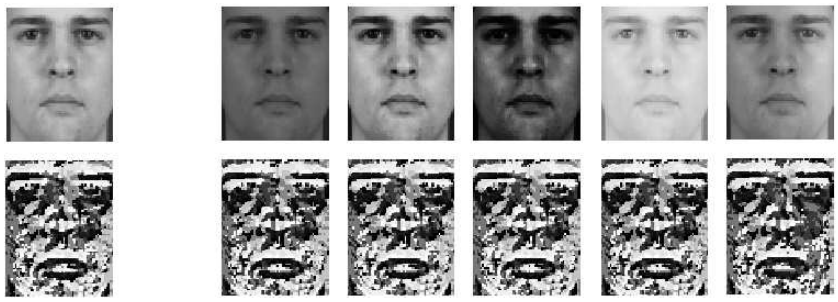


Fig. 3.3 – Original images of faces and their result after applying the LBP operator on the entire image : first line corresponds to the original image followed by images obtained by monotonic gray-scale transformations of the original image, the second line consists of the transformation of the face images after applying the LBP operator on the entire image. We can notice that the LBP transformed images are not perturbed by any monotonic gray-scale transformation.

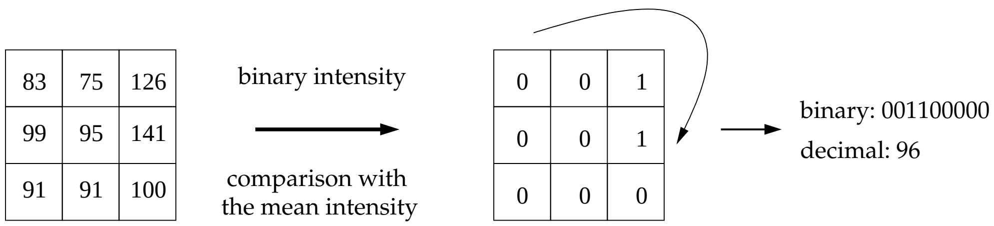


Fig. 3.4 – MCT operating on a 3×3 set of pixels, from left to right : original set of pixels, binary comparison of pixel intensities between all the pixels of the neighborhood and the mean intensity of the nine pixels of the neighborhood, results of the transform and MCT code in the binary and decimal form.

The decimal form of the resulting 9-bit string (MCT code) can then be expressed as follows :

$$MCT(x_c, y_c) = \sum_{n=0}^{8} s_{MCT}(i_n - i_m)2^n \tag{3.3}$$

where i_m corresponds to the mean of the pixel intensities of the entire neighborhood, including the center pixel, and i_n is the grey value of n-th pixel for $n \in [0, \ldots, 8]$. The function s_{MCT} is defined as :

$$s_{MCT}(x) = \begin{cases} 1 & \text{if} \quad x > 0 \\ 0 & \text{if} \quad x \leq 0\,. \end{cases} \tag{3.4}$$

By definition, the MCT keeps the properties of the LBP operator. Thus the MCT is unaffected by any monotonic gray-scale transformation and its computational cost is still very low.

3.2.2 Classifier

Let $P = \{P_i\}_{n=1}^{N}$ be the set of the N hand posture classes to recognize. To each posture class $P_i,\ i \in [1 \ldots N]$ is associated a classifier $H_i,\ i \in [1 \ldots N]$. Each classifier H_i is a linear combination of elementary (or weak) classifiers :

$$H_i(X) = \sum_{p \in W} h_p(x_p)$$

where X is the image of the hand posture, p is a pixel location in the image such as $p \in W$ (W is the set of pixel locations), h_p is the weak classifier associated to the pixel position p and x_p is the MCT code at the location p in the image.

Weak Classifiers

A weak classifier h_p consists of a look-up table of 511[1] bins, which is the total number of possible MCT codes. This weak classifier is associated to a specific pixel location p in the image. Each bin of the look-up table contains a real value which corresponds to the weight of the related MCT code. At a given location p in a test image, the output of the classifier $h_p(x_p)$ is the value of the bin x_p (x_p is the MCT code computed at location p). Figure 3.5 illustrates a classifier composed of five weak classifiers, as well as the look-up table of one of them.

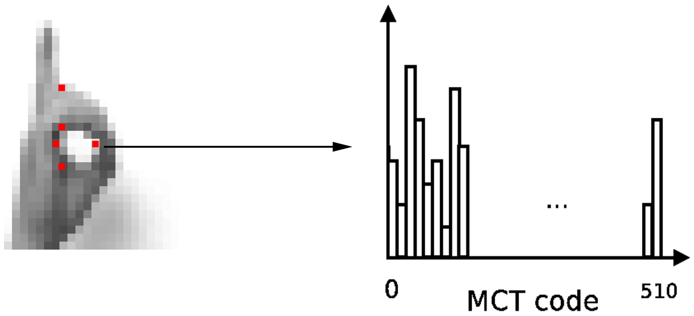


Fig. 3.5 – Example of a weak classifier, from left to right : image of a hand posture with position of five weak classifiers (in red) and look-up table associated to one of the weak classifier.

[1] $2^9 - 1 = 511$ because the MCT codes with pixels all equal to 0 or 1 convey the same information

AdaBoost Training

In the AdaBoost framework, the algorithm selects the weak classifier which minimizes the classification error rate on a weighted distribution of positive and negative samples of the class to recognize. Here, a weak classifier consists of a look-up table associated to a pixel location. Then, AdaBoost aims to select pixel locations and to build the associated look-up table. The training algorithm is detailed in [49] and is explained below.

To simplify notations, we suppose that we are dealing with a classifier H such that

$$H(X) = \sum_{p \in W} h_p(x_p) \tag{3.5}$$

The number N_H of weak classifiers is fixed, as well as the number N_{boost} of boosting iterations. N_H is then the size of the set of pixel locations W.

At each boosting iteration t, to select the best pixel classifier, two look-up tables $L_p^{posture}$ and $L_p^{nonposture}$ are allocated for each pixel location of W. Then, for each location p, the MCT operator is applied on a training set of hand posture samples. For each sample, the computed MCT code is used to identify the bin of $L_p^{posture}$ which is increased by an amount equal to the weight of the sample. The same is done with a training set of non-postures to populate the $L_p^{nonposture}$ table. The classification error ϵ at position p is given by

$$\epsilon_p = \sum_{j=0}^{510} min(L_p^{posture}[j], L_p^{nonposture}[j]) \tag{3.6}$$

At the beginning of the boosting process, each location in the image is a possible pixel location. A selected pixel location is thus chosen to minimize Eq. (3.6). When the number of selected pixel locations at iteration $t-1$ is equal to N_H, then the selected pixel location at iteration t is

$$p^* = \arg\min_{p \in W} \epsilon_p$$

where W is the set of already selected pixel locations ($card(W) = N_H$).

The look-up table L_{p^*} of the selected pixel classifier at iteration t is then computed for each bin j :

$$L_{p^*}[j] = \begin{cases} 1 & \text{if } L_{p^*}^{posture}[j] > L_{p^*}^{nonposture}[j] \\ 0 & \text{otherwise} \end{cases} \tag{3.7}$$

A pixel classifier thus consists of a look-up table of 0s and 1s. During the boosting learning, a discriminative pixel location may be selected several times. At the end of the boosting procedure, look-up tables associated to the same pixel location are merged into a single table. For each bin, a weighted sum is done on the bin values of each table. Weak classifiers $h_p(x_p)$ of Eq. 3.5 consists of these single weighted look-up tables. In the following of this chapter we will refer to the proposed approach as the MCT-Boost technique.

3.3 Experiment Set-up

3.3.1 Database

The database used in this thesis is the Jochen Triesch database[2] which is one of the rare benchmark database in the field of HPR. The database contains 128×128 gray-scale images of 10 hand signs (Figure 3.6) performed by 24 different gesturers against different backgrounds. The backgrounds are of three types : uniform light, uniform dark and complex (Figure 3.7). Among the 720 images, two were lost by the author of the database.

[2]http ://www.idiap.ch/resources/gestures/

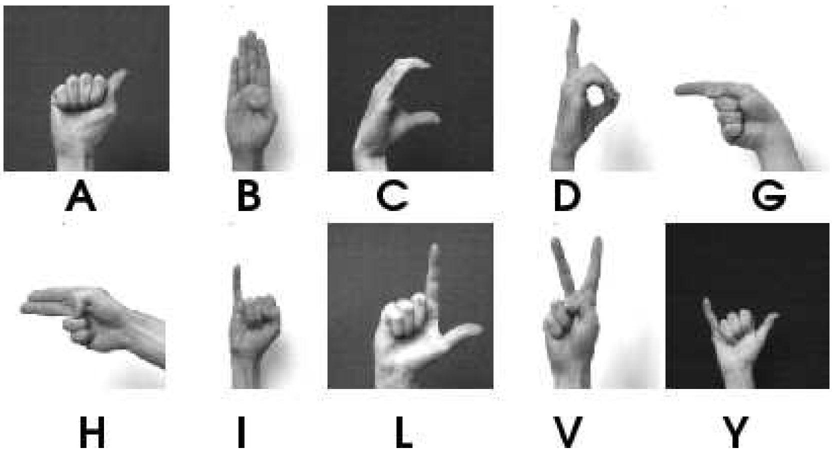


FIG. 3.6 – Examples of the 10 postures contained in the Jochen Triesch database, with their associated label.

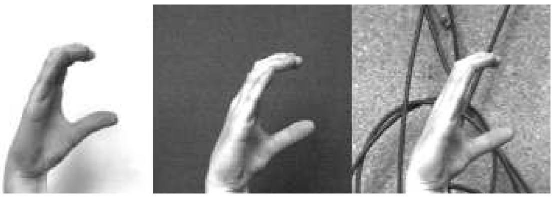


FIG. 3.7 – Examples of the three types of background for the C hand posture, from left to right : uniform light, uniform dark and complex.

3.3.2 Protocols

For experimental purposes, the database has been divided into three subsets : train set T, validation set V and test set Te. The decomposition into subsets has been done following two different protocols : Protocol 1 and Protocol 2. Images of four people were used for the training set. For Protocol 1, only images against uniform background have been used. For Protocol 2, both images against uniform and complex background were used for training. Images of four other people were used for the validation set. For Protocol 1, only images against uniform background were used in V whereas for Protocol 2, both images against uniform and complex background were used in V. Images of the sixteen remaining people were used for Te. For Protocol 1, as images against complex background of the eight people used for T and V were not used in the training and validation phases, they have been added to Te. Table 3.1 summarizes the decomposition into the three subsets for both Protocol 1 and Protocol 2.

	Protocol 1			Protocol 2		
	T	V	Te	T	V	Te
number of people	4	4	16	4	4	16
number of images	80	80	558	119	120	479
background type	U	U	U&C	U&C	U&C	U&C

TAB. 3.1 – Statistics of the training set T, validation set V and test set Te for both Protocol 1 and Protocol 2. Background type can be uniform (U) or complex (C).

For each hand posture and using Protocol 1, train and validation sets contain both 8 images and test set contains 56 images. Only postures H and V have 55 images in their test set due to missing images. For Protocol 2, train and validation set for each hand posture are composed of 12 images, and test set contains 48 images. Only posture H and V have respectively 11 images in the train set and 47 images in the test set, due to missing images.

3.4 Preprocessing

To train the classifier H_i associated with the hand posture P_i, $i \in [1 \ldots N]$, a train set containing images of the hand posture is needed, as well as a second dataset containing sample images of *non*-postures. The dataset of non-posture images consists of randomly collected images from the Internet as well as images of the $N-1$ remaining postures. We expect this addition to increase the robustness of the model. For images of hand postures, the train set and validation set are the ones defined in Section 3.3.2.

Before using the images to train the model, they have first been cropped and scaled to a common 30×30 size. Then a histogram equalization has been applied on the images. Some perturbations have been added to these images to increase the available number of images for the train and validation process. Three types of perturbations have been generated. The images have been shifted, scaled and rotated of a few pixels and degrees. For each original image, 30 perturbed images have been generated. At the end of the process, for the first protocol, we have $4 \times 2 \times (30+1) = 248$ [3] images per posture for the training and validation set. For the second protocol, we have $4 \times 3 \times (30+1) = 372$ [4] images per posture, except for the 'H' posture for which we only have 351 images (one image of the initial database was lost). Original image of the D hand posture followed by some perturbed images is shown on Figure 3.8.

[3] number of people × 2 uniform background (light and dark) × (number of perturbed images + original image)
[4] number of people × 3 types of background × (number of perturbed images + original image)

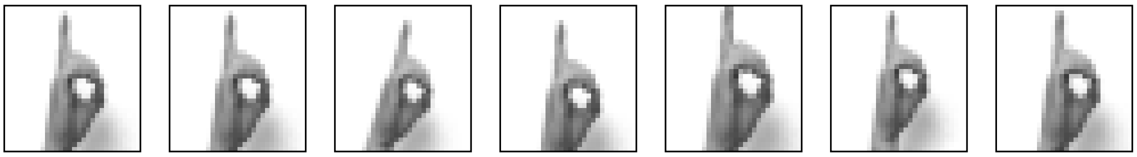


Fig. 3.8 – Images of the D posture contained in the training set, from left to right : original image and images after perturbations of the original image (scale, rotate and shift)

In the case of the test set, no perturbation has been added to the images. The test set contains the initial images cropped, scaled to the size of 30×30 and normalized using histogram equalization.

3.5 Results

This section is divided in two parts. The problem of detection of hand posture in images is first considered to validate the proposed approach. In the second part, the approach is applied to the HPR task. For experiments, we have used the Torch and TorchVision libraries [5].

3.5.1 Hand Posture Detection

State-of-the-Art Approach

So far only one hand posture detection approach [117] has been evaluated on the Jochen Triesch database. However, only hand postures 'A', 'B', 'C' and 'V' were used for detection. In this approach, the entire image of the hand is used as input of a Constrained Generative Model (CGM). Images are tested at different position and scale to frame the hand posture. The average detection rate for images against uniform background is 93.7% and 84.4% for images against complex background. Unfortunately, these results are not comparable to the one obtained with the MCT-Boost technique. First, the protocol used by Marcel is different from Protocol 1 and Protocol 2 used for the MCT-Boost approach. Furthermore, Marcel considers the detection on the entire image whereas the MCT-Boost technique is applied on already cropped images.

MCT-Boost Approach

Hand postures vary greatly due to the high number of DOFs of the human hand which is not the case for faces. It is then not obvious that the MCT-Boost approach can be extended to the HPR. To be sure that the MCT-Boost approach is still valid with hand postures, the problem of hand posture detection is considered. It will answer the following question : given an image of the posture P_i, is the classifier H_i able to differentiate this hand posture from background images ?

One model H_i has been trained for each hand posture class P_i, and for each protocol. Let H_i^1 be the classifier trained using Protocol 1 and H_i^2 the classifier trained using Protocol 2. Each classifier is a linear combination of 500 weak classifiers trained with 2500 iterations of boosting. To test these models, images from the test set corresponding to the hand posture class P_i are presented to both H_i^1 and H_i^2. Detection occurs when $H_i > \theta_i$, where θ_i is the decision threshold for hand posture P_i chosen on the validation set. This threshold is chosen by minimizing the classification error rate.

Detection results against both uniform and complex background are reported on Table 3.2. Each model H_i detects correctly the hand posture they have been trained to model. Some remarks can be drawn :

1. **Background** : For both background conditions, the detection rate is high. The average classification rate is equal to 99.2% for images against uniform background, and 89.8% for cluttered

[5]http ://www.torch.ch and http ://torch3vision.idiap.ch

conditions. Detection rate under uniform background provides better results than under complex background.

2. **Posture** : Against uniform background, all postures are well classified, whatever the protocol used. For images against complex background, detection rate decreases for some postures using Protocol 1. Postures 'B', 'C', 'V' and 'Y' are more difficult to detect against complex background when the corresponding classifier is trained using Protocol 1.
3. **Protocol** : The difference between the two protocols lies in the composition of the training and validation sets. In Protocol 1, the training and validation sets do not contain any images performed against complex background. In Protocol 2, images recorded in cluttered conditions are included. As stated above, for all hand postures and both protocols, the classification rates are very high for images against uniform background. In cluttered conditions, for Protocol 2, results are as good as in uncluttered conditions. On the other hand, for Protocol 1, the recognition rate decreases for most of the hand postures. The difference in the performances is particularly obvious with the 'C', 'V' and 'Y' postures. Using images against complex background in the training process permits the classifier to extract more relevant information.

	Uniform Background		Complex Background	
Hand Posture	Protocol 1	Protocol 2	Protocol 1	Protocol 2
A	100	100	91.67	100
B	93.75	100	75	100
C	96.88	100	66.67	93.75
D	100	100	87.5	100
G	100	100	87.5	100
H	100	100	100	100
I	100	100	95.83	93.75
L	100	100	100	100
V	96.77	100	54.17	100
Y	96.88	100	62.5	87.5
average	98.4	100	82.1	97.5

TAB. 3.2 – Detection rate (in %) on the test set obtained with the MCT-Boost method, for each hand posture and each protocol. Results on test images with a uniform background (in the 2^{nd} column) and test images against complex background (in the 3^{rd} column) are provided.

Baseline Approach

To compare the MCT-Boost approach with a baseline system, a simple multi-layer perceptron (MLP) using the Mean-Squared Error (MSE) criterion has been trained. The MLP has 900 inputs (30×30 hand posture image), and one hidden layer of 283 hidden units. The obtained MLP has then roughly the same number of parameters as the MCT-Boost classifier[6]. To train the MLP, two train sets are needed. The first train set $T_{posture}$ contains images of the hand posture to detect and is the same as the train set used for the MCT-Boost technique. The second train set $T_{nonposture}$ contains images

[6]MCT-Boost classifiers has $511 \times 500 = 255500$ parameters, and the MLP has $900 \times (283 + 1) = 255600$

of non-postures and thus is the concatenation of the train set of the $N-1$ remaining hand postures. For the i-th hand posture, with $i \in [1 \ldots N]$, $T_{nonposture} = \bigcup_{j\in[1\ldots N], j\neq i} T^{j}_{posture}$ where $T^{i}_{posture}$ is the train set for the i^{th} hand posture. Detection occurs when the score of the MLP is higher than a certain threshold. This threshold has been chosen using two validation sets, one containing images of the hand posture and the other one containing non-posture images. The first validation set is similar to the validation set used for the MCT-Boost approach. Similarly to the train set of non-postures, the validation set of non-postures is the concatenation of the validation sets of the $N-1$ remaining hand postures.

Table 3.3 presents detection results on the test set (with both images against uniform and complex background) for both the MCT-Boost and MLP approach, using Protocol 1 and Protocol 2. The average detection rate using the MCT-Boost is 90.25% using Protocol 1 and 98.75% using Protocol 2. Compared to the baseline results obtained using the MLP, the MCT-Boost method gives better results for detection. These results show the validity of the MCT-Boost approach.

	Protocol 1		Protocol 2	
Hand Posture	MLP	MCT-Boost	MLP	MCT-Boost
A	92.86	95.84	95.83	100
B	85.71	84.38	93.75	100
C	83.93	81.78	77.08	96.88
D	73.21	93.75	77.08	100
G	73.21	93.75	77.08	100
H	85.45	100	83.33	100
I	85.71	97.92	93.75	96.88
L	78.57	100	91.67	100
V	78.18	75.47	87.23	100
Y	89.29	79.69	85.42	93.75
average	82.61	**90.26**	86.22	**98.75**

Tab. 3.3 – Detection rate (in %) on the test set (with both images against uniform and complex background) for the baseline MLP and MCT-Boost method.

3.5.2 Hand Posture Recognition

State-of-the-Art Approach

In [170], the elastic graph matching technique is applied to HPR. Hand postures are represented as labeled graphs. To train the models, six images for each hand postures have been used. Furthermore, Triesch and von der Malsburg employed the entire image to recognize hand postures. They obtained 94.4% recognition rate in uniform light conditions, 86.2% in uniform dark conditions and 86.2% recognition rate for images against complex background. The protocol used in [170] is thus different from Protocol 1 and Protocol 2 defined in section 3.3.2. Thus results obtained by Triesch and von der Malsburg with elastic graph matching are not comparable to the one obtained with the MCT-Boost approach.

MCT-Boost Approach

In Section 3.5.1, the problem of hand posture detection was considered. In this section, the MCT-Boost approach will be applied to the HPR task. Given an image of an unknown posture, we would like to find its posture class label. For that purpose, a "one versus all" strategy has been chosen. For a given posture test image, all the models $H = \{H_i\}_{i=1}^{N}$ for each hand posture class are considered. The classifier giving the highest score is used to label the test image.

The recognition rate for each posture p_i, both for uniform and complex backgrounds, is reported in Table 3.4. Several remarks can be drawn from this table :

1. **Background** : In general, the recognition rate is higher for images against uniform than complex background. Some postures are not affected by the background type such as 'A' or 'B', while other postures are strongly affected, such as 'G', 'I', 'L', 'V' or 'Y' (Figures 3.9 and 3.10). The common characteristic of these postures is a closed fist with one or two pointing fingers. The detection of these thin finger regions is difficult using the MCT-Boost approach. Furthermore, the detection of these fingers is very sensitive to the background type. Fingers are "sunk" in the background and thus difficult to find out.

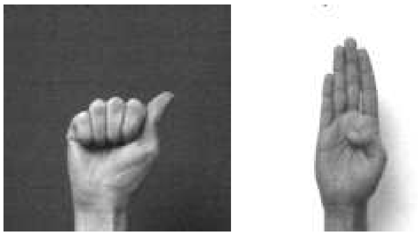


FIG. 3.9 – Examples of hand postures not affected by the background type, from left to right : A and B hand posture.

FIG. 3.10 – Examples of hand postures affected by the background type, from left to right : G, I, L, V and Y hand postures.

2. **Posture** : Some postures, for both background types, are easier to recognize. The 'A' posture for instance, achieves 100% recognition in both background conditions. On the other hand, the 'Y' posture achieves the lowest recognition rate in both conditions. The main difference between these postures lies in the presence of extended fingers. The 'A' posture is a fist whereas the 'B' posture is an extended hand with adjacent fingers. The variability within these hand posture classes is very low compared to the variability of the 'Y' hand posture class. This variability is a problem for the MCT-Boost approach.

3. **Protocol** : Integrating images against complex background in the train and validation sets helps the algorithm to model more accurately the hand postures. It is the case for the 'D', 'G', 'I', 'L', 'V' and 'Y' postures.

	Uniform Background		Complex Background	
	Protocol 1	Protocol 2	Protocol 1	Protocol 2
A	100	100	100	100
B	93.75	93.75	93.75	93.75
C	93.75	93.75	75	93.75
D	93.75	84.38	62.5	81.25
G	96.88	100	50	68.75
H	84.38	90.63	87.5	87.5
I	84.38	90.63	56.25	62.5
L	84.38	96.88	37.5	75
V	87.10	96.77	56.25	87.5
Y	81.25	81.25	25	62.5
average	89.97	**92.79**	64.38	**81.25**

Tab. 3.4 – Recognition rate (in %) on the test set obtained with the MCT-Boost method, for each hand posture and each protocol. Results on test images with uniform background (in the 2^{nd} column) and test images against complex background (in the $3^{rd} column$) are provided.

Baseline Approach

Table 3.5 presents recognition results for both the MCT-Boost approach and the MLP-based approach. Using the MLP, recognition occurs for the model giving the highest score. Performances obtained using the MCT-Boost approach and the MLP are comparable. In average, the MCT-Boost approach performs better than the MLP. Results obtained using Protocol 2 are better than the one obtained using Protocol 1 for both the MLP and the MCT-Boost approach. Adding images of hand postures against complex background improves the recognition, leading to more robust systems.

3.6 Discussion

The boosting approach based on MCT features, applied with success to face detection, can also be applied to the problem of hand posture detection. In that case, the MCT-Boost approach gives better results than the baseline MLP. Experiments using two different protocols have also shown that better results are obtained with Protocol 2 when dealing with images against complex background. It is particularly significant for the 'C', 'V' and 'Y' hand postures. These hand postures have in common a conformation with extended fingers in the diagonal direction. Compared to hand postures with extended fingers, the fact that fingers are not horizontally nor vertically directed leads to more variability in these hand postures. Adding images against complex background in the training phase helps classifiers to focus on invariant parts of the hand postures, such as the fingers.

The MCT-Boost approach has also been applied to the hand posture recognition task. Experiments show that the recognition rate is higher for images against uniform background for both protocols. The recognition rate decreases when dealing with images against complex background, particularly when training is performed using Protocol 1. The 'C', 'D', 'G', 'I', 'L', 'V' and 'Y' hand postures are very difficult to recognize against complex background due to the presence of extended fingers. The recognition rate can be improved for images against complex background by training classifiers

	Protocol 1		Protocol 2	
Hand Posture	MLP	MCT-Boost	MLP	MCT-Boost
A	82.14	100	91.67	100
B	85.71	93.75	89.58	93.75
C	60.71	84.38	79.17	93.75
D	66.07	78.13	70.83	82.82
G	69.64	73.44	77.08	84.38
H	73.21	85.94	81.25	89.07
I	75.0	70.32	87.50	76.57
L	67.86	60.94	83.33	85.94
V	69.64	71.68	75.0	92.14
Y	73.21	53.13	77.08	71.88
average	72.32	**77.18**	81.25	**87.02**

Tab. 3.5 – Recognition rate (in %) on the test set (with both images against uniform and complex background) for the baseline MLP and MCT-Boost method.

using Protocol 2. However, some hand postures are still difficult to recognize for both protocols. These postures are the 'G', 'I' and 'Y' postures. The 'H' and 'G' hand postures are very similar as the only difference is the number of extended fingers. For the 'G' posture only one finger is extended whereas for the 'H' hand posture two fingers are extended. The recognition rate for the 'H' hand postures is higher than 80% for both protocols. The extended fingers are thus more recognizable because they are wider. The 'I' hand posture consists of a fist with one extended finger. This posture can easily be mistaken with the 'A' hand posture. Finally, for the 'Y' hand posture, images of the test set present a high variability. Even if the fist conformation can be detected in images, the extended fingers are difficult to find as they do not have the same position in every test image. The 'L' and 'V' hand postures are also quite difficult to recognize because of their similarity with the 'A' hand posture. The 'A' and 'B' hand postures are the only two hand postures which recognition does not depend on the background type nor the protocol used. They have in common a very compact shape, based on a fist for the 'A' hand posture, and with close extended fingers for the 'B' hand posture. They are thus easier to recognize.

For the hand posture recognition task, the MCT-Boost approach shows some improvement over the baseline MLP. Furthermore, the MCT-Boost approach is potentially much faster than the baseline MLP. Indeed, in the context of a scanning approach, the MCT can be computed at any scale and location in constant time (using the integral image representation) while it is not possible with an MLP. In this study, only cropped images have been used. In each image, the posture is centered and all the images have the same size. In reality, images of hand postures are not already cropped, and hands can be of different size and at different locations in the image. Therefore, a scanning approach could be used to convolve the image (at different sizes and locations) with a hand posture classifier. This approach, widely used in face detection, produces a lot of false detections, typically in the background. These false detections are usually discarded using a skin color detector which segments skin color regions such as the hand. Given the various topics considered in this thesis (HPR, HGR, two-handed gestures), the scanning approach was not investigated. Focus was directed only on the design and the evaluation of the classifiers.

Chapitre 4

Hand Gesture Recognition

The present chapter is concerned with hand gesture recognition (HGR) while previous chapter was dealing with hand posture recognition (HPR). The gesture database used for the experiments is composed of both one- and two-handed gestures. These gestures consist of 3D trajectories of hands, head and torso. Because of a lack of comparisons on common hand gesture databases with strict experimental protocol, we evaluate two state-of-the-art sequence processing algorithms already applied to HGR, namely Hidden Markov Models (HMMs) and Input-Output Hidden Markov Models (IOHMM).

4.1 Problem Description and Standard Approaches

HGR is a sequence processing problem. Techniques such as Finite State Machines (FSM) [40, 50] or Dynamic Time Warping (DTW) [147, 2] have been widely used for the recognition of hand gestures. But the most important technique applied to the recognition of hand gestures is Hidden Markov Models (HMM) [140]. The Input-Output Hidden Markov Model (IOHMM) [11], which is an extension of the HMM, has also been proposed for HGR [118].

HGR suffers from the lack of available databases. This is an important problem to compare directly an algorithm A to an algorithm B. Moreover, few databases are provided with a strict train/test experimental protocol describing the usage of data. In this section, we thus propose to evaluate the performances obtained with both HMMs and IOHMM on the same database and using a strict experimental protocol. This new database is publicly available.

4.2 Sequence Processing Algorithms

4.2.1 Hidden Markov Models

Statistical Model

A Hidden Markov Model (HMM) [140] is a statistical machine learning algorithm which models sequences of data. It consists of a set of N states called hidden states because non-observable. It also contains transition probabilities between these states and emission probabilities from the states to model the observations. The data sequence is thus factorized over time by a series of hidden states and emission from these states. Let $y_1^T = \{y_1, \ldots, y_t, \ldots, y_T\}$ be an output sequence, where $T \in \mathbb{N}$ is the length of the observation sequence, and let $q_t \in \{1, \ldots, N\}$ be the state at time t. The emission probability $P(y_t|q_t = i), \forall i \in \{1, \ldots, N\}$ at time t depends only on the current state q_t. The transition probability between states $P(q_t = i|q_{t-1} = j), \forall (i, j) \in \{1, \ldots, N\}^2$ depends only on the previous state.

The model of a HMM is the set of all the following parameters $\Lambda = (\Pi, A, B)$:

– the parameter vector $\Pi = (\pi_i)$ with $i \in \{1, \ldots, N\}$ is the initial distribution over all the states :

$$\pi_i = P(q_0 = i),$$

– the matrix $A = (a_{ij})$ with $(i, j) \in \{1, \ldots, N\}^2$ which determines the transition probabilities from the state i to the state j :

$$a_{ij} = P(q_t = j | q_{t-1} = i), \forall (i, j) \in \{1, \ldots, N\}^2$$

– the set of parameters $B = (b_j(y_t))$ with $j \in \{1, \ldots, N\}, t \in \{1, \ldots, T\}$, which represents the observation probability of y_t in the state j :

$$b_j(y_t) = P(y_t | q_t = j), \forall j \in \{1, \ldots, N\} \; and \; \forall t \in \{1, \ldots, T\}.$$

As we are dealing with continuous data, the set of observation probabilities are represented by a mixture of Gaussians.

To efficiently use HMMs, it is necessary to impose a topology to the state graph. This topology limits the number of free parameters and allows to inject in the model some *a priori* knowledge on the nature of the data. Figure 4.1 represents the state graph of a left-right topology for a HMM.

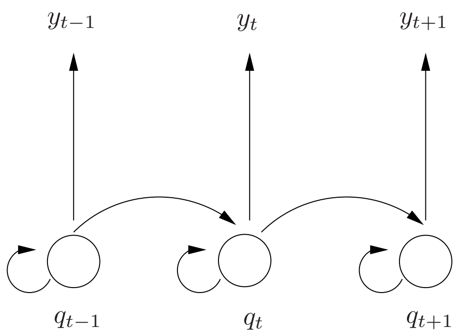


FIG. 4.1 – An HMM with a left-right topology showing dependencies between the observations y and the hidden states q

Training

Let $\mathcal{Y}$ be the set of observations to model. The HMM training is obtained by maximizing the likelihood

$$L = \prod_{k=1}^{K} P(y_1^{T_k} | \Lambda)$$

where Λ represents the set of parameters of the model, K is the total number of observation sequences in $\mathcal{Y}$ and $y_1^{T_k} \in \mathcal{Y}$ is the k-th observation sequence of length T_k. The *Expectation-Maximization* algorithm [41] is typically used to maximize this likelihood and to adjust the parameters of the model. More details can be found in [41, 140].

Recognition

The goal of HGR is to recognize gestures belonging to a set of predefined gesture classes. Let C be the number of gesture classes to recognize. Each gesture class will thus be modeled by a HMM with parameters Λ_c, with $c \in [1 \ldots C]$. A naive Bayes classifier is applied to perform the classification, assuming equal prior probabilities for each gesture class. In the recognition phase, the class the gesture belongs to is $\arg\max_{c \in \{1,\ldots,C\}} P(y_1^T | \Lambda_c)$.

4.2.2 Input-Output Hidden Markov Models

Statistical Model

An Input-Output Hidden Markov Model (IOHMM) is an extension of the HMM described previously. First introduced by Bengio and Frasconi [10], IOHMMs are able to discriminate temporal sequences using a supervised training algorithm. An IOHMM maps an input sequence to an output sequence. In our case, input sequences are the observations and output sequences correspond to the class of the gesture.

Let $x_1^T = \{x_1, \ldots, x_t, \ldots, x_T\}$ be an input sequence, and $y_1^T = \{y_1, \ldots, y_t, \ldots, y_T\}$ be an output sequence, where $T \in \mathbb{N}$ is the length of input/output sequences. The architecture of IOHMM also consists of a set of states so let $q_t \in \{1, \ldots, N\}$ be the state at time t. With each state are associated two conditional distributions : one for transition probabilities and one for emission probabilities. The emission probability $P(y_t|q_t = i, x_t), \forall i \in \{1, \ldots, N\}$ at time t depends on the current state q_t, but also depends on x_t. The transition probability between states

$$P(q_t = j|q_{t-1} = i, x_t), \forall (i, j) \in \{1, \ldots N\}^2$$

depends on the previous state and also on x_t (Figure 4.2).

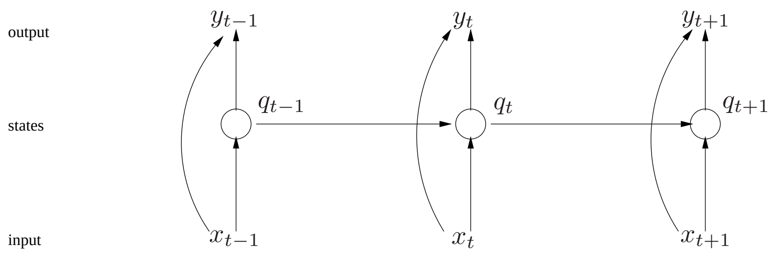


Fig. 4.2 – An IOHMM showing dependencies between the input x, output y and hidden states q of the model.

IOHMMs are time dependent since the emission and transition probabilities depend on x_t. Hence IOHMMs are based on non-homogeneous Markov chains contrary to HMMs. Consequently, the dynamic of the system is not fixed *a priori* such as in HMMs, but evolves in time and is function of the input sequence.

The data sequences to model can be of two types : discrete or continuous. In the discrete case, codebooks or multinomial distributions can be used to model the conditional distributions. In the continuous case, models such as MLP [145] can be used to represent the conditional distributions. Another solution to deal with continuous observations is to perform a quantization to discretize the data.

Training

Let $\mathcal{X}$ be the set of input sequences and $\mathcal{Y}$ the set of output sequences to model. The IOHMM training is obtained by maximizing the likelihood

$$L = \prod_{k=1}^{K} P(y_1^{T_k}|x_1^{T_k}, \Lambda)$$

where Λ represents the set of parameters of the model, K is the length of the input/output sequence, $x_1^{T_k} \in \mathcal{X}$ is the k-th input sequence and $y_1^{T_k} \in \mathcal{Y}$ is the k-th output sequence of length T_k. The *Expectation-Maximization* algorithm [41] is also used to maximize this likelihood and to adjust the parameters of the model. More details can be found in [11, 116].

Recognition

Let x_1^T be an input sequence to recognize. The test sequence is assigned to the class with the highest conditional probability on each frame such that

$$\text{gesture class} = \arg \max_{c \in \{1,\dots,C\}} P(y_1 = c, \dots, y_T = c | \mathbf{x_1^T}).$$

This recognition method is consistent with previous work on HGR using IOHMM [118].

4.3 Hand Gesture Recognition

4.3.1 Description of the Database

The database consists of 16 gestures (Table 4.1) carried out by 20 different people. Most of gestures are one-handed and some are two-handed (*fly*, *swim* and *clap*). For each person and each gesture, 5 sessions and 10 shots per session have been recorded. All the gestures start and end in the same rest position (the hands lying along the thighs). Temporal segmentation was manually accomplished after a recording session.

Name	Description	**R**	**1/2**
Stop/yes	Raised hand on the head level and facing palm		1
No/wipe	Idem with movements from right to left	R	1
Raise hand	Raised hand higher than the head		1
Hello/wave	Idem with movements from right to left	R	1
Left	Hand on the hip level, movements to the left	R	1
Right	Hand on the hip level, movements to the right	R	1
Up	Hand on the hip level, movements to the up	R	1
Down	Hand on the hip level, movements to the down	R	1
Front	Hand on the hip level, movements to the front	R	1
Back	Hand on the hip level, movements to the back	R	1
Swim	Swimming mimic gesture	R	2
Fly	Flying mimic gesture	R	2
Clap	On the torso level, clap the hands	R	2
Point left	On the torso level, point to the left		1
Point front	On the torso level, point to the front		1
Point right	On the torso level, point to the right		1

TAB. 4.1 – Description of the 16 gestures. A hand gesture could involve one hand (**1**-handed) or both hands (**2**-handed). The gesture could be also a **R**epetitive movement such as *clap*.

For each gesture, a trajectory for each region of interest has been generated. These trajectories correspond to 3D coordinates of the center of the head, of the two hands and of the torso. They are produced with the natural hand (left hand for left-handed and right hand for right-handed people). For left-handed people, trajectories have been mirrored. Finally, the database is composed of 1000 trajectories per gesture.

Figure 4.3 shows an example of the swim gesture sequence from the point of view of the right camera. Furthermore, for each person and each session, a 'Vinci' sequence has been recorded (Figure 4.4). This sequence gives the maximum arm spread. Figure 4.4 presents also in a three dimensional space[1] the coordinates of the center of each blob (head, torso and hands) for a 'swim' gesture sequence.

[1] the z axis is the vertical axis of the person.

Fig. 4.3 – From top-left to bottom-right, a frame-by-frame decomposition of a "swim" gesture from the point of view of the right camera.

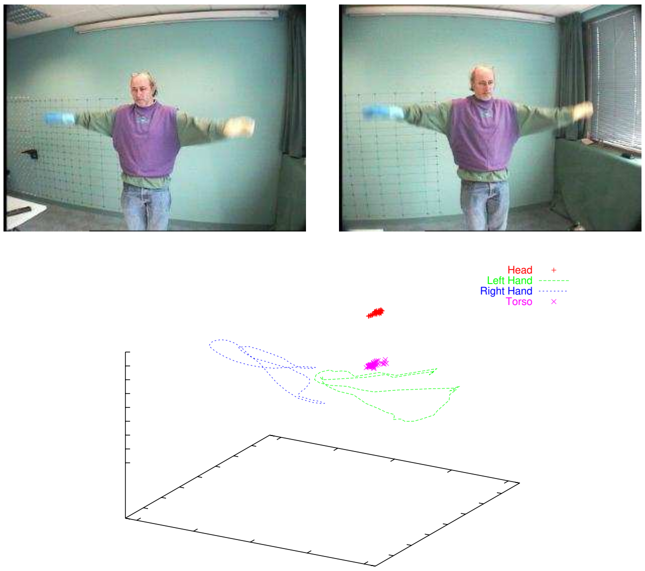


Fig. 4.4 – Top : example of images of the "Vinci" sequence from the point of view of the left camera (on the left) and from the point of view of the right camera (on the right). Bottom : 3D coordinates of the center of each blob (head, torso, left and right hand) for a "swim" gestures.

4.3.2 Feature Extraction

Hand gestures are represented by 3D trajectories of blobs. The blobs are obtained by tracking colored body parts in real-time using the EM algorithm. A detailed description of the 3D blob tracking algorithm can be found in [12]. Two cameras (Figure 4.5) are used to track head and hands in near real-time (12Hz).

Fig. 4.5 – Top : left and right captured images. Middle : left and right images with projected ellipsoids. Bottom left ellipsoids projection on the frontal plane. Bottom right : ellipsoids projection on the side plane (the cameras are on the left side)

First, the preprocessing step consists of background subtraction followed by specific colors detection, using a simple color look-up table. A statistical model is then applied, which is composed of four ellipsoids, one for each hand, one for the head and one for the torso. Each one of these ellipsoids is projected on both camera planes as ellipses. A Gaussian probability density function with the same center and size is linked to each ellipse. The parameters of the model (positions and orientations of the ellipsoids) are adapted to the pixels detected in the preprocessing stage. This adaptation takes into account the pixels detected by the two cameras, and is based on the maximum likelihood principle. The EM algorithm is thus used to attain the maximum of the likelihood.

The use of gloves during recording permits to avoid occlusion problems that occur with two-handed gestures. The person performing the gesture wears colored gloves and a sweat-shirt to help the segmentation process.

In order to extract features, we have done some simple preprocessing on the raw data as follows :

1. *Normalization :* As a first step, a normalization has been performed on all gesture trajectories. We suppose that each gesture occurs in a cube centered on the torso and of vertex size the maximum arm spread given by the 'Vinci' sequence. This cube is then normalized to reduce the vertex to 1. Finally, the range of x, y and z coordinates varies between -0.5 and 0.5.
   One can notice that the 3D-coordinates of the head and torso are almost stationary (Figure 4.4). Thus only the the normalized 3D-trajectories of both hands have been kept. This leads to an input feature vector of size 6.

2. *Feature Extraction :* Delta features have also been computed. For each hand, they correspond to the difference between the coordinates for two consecutive frames. They represent the speed of each hand in the three directions. These features have been multiplied by 100 to have values in the same order of magnitude than x, y and z. The final vector used is then $[x_l, y_l, z_l, x_r, y_r, z_r, \Delta x_l, \Delta y_l, \Delta z_l, \Delta x_r, \Delta y_r, \Delta z_r] \in \mathbb{R}^{12}$.

4.4 Sequence Processing Algorithms for HGR

In this section, baseline results obtained using HMMs and IOHMM are presented. In the case of the IOHMM, experiments have been conducted using discrete conditional distributions. Thus a quantization step has been performed on the data. This quantization is explained later in this section. The open source machine learning library used for all experiments is Torch (http ://www.torch.ch).

4.4.1 Preprocessing

To perform recognition with HMMs, no preprocessing on the feature vectors is needed. But to efficiently use discrete IOHMM, a quantization step on the data is needed. The output sequence still encodes the hand gesture classes : $y_t = \{0, \ldots, 15\}, \forall t$. To model more closely the class distribution of the data, a K-means algorithm [58] has been applied class per class on the input features.

For each gesture class, a K-means model with 75 clusters has been trained on the train and validation set. The 16 resulting K-means models have been merged into a single one with 1200 clusters. Finally, each frame of each sequence is quantized into one discrete value $\in \{1, \ldots, 1200\}$, which is the index of the nearest cluster.

4.4.2 Parameter Tuning

In experiments, both left/right and full connect topologies for HMMs and IOHMM have been used. To find the optimal hyper-parameters (number of states for the discrete IOHMM, number of states and number of Gaussians for the HMMs), the following protocol has been used. The database has been split into three subsets : the training set T, the validation set V and the test set Te. Different possibilities for the hyper-parameters have been tried on T. The selection of the best parameters has been done on V. Finally, a model has been trained on both T and V and tested on Te. The best results were obtained with the following hyper-parameters (Table 5.1)

	HMMs	IOHMM
Topology	left-right	left-right
Number of Gaussians	1	$\times$
Number of states	13	3

Tab. 4.2 – Hyper-parameters to test HMMs and IOHMM on the hand gesture database.

4.5 Results

Table 4.3 provides comparative results on the test set Te between discrete IOHMM and baseline continuous HMMs on the first hand gesture database.

HMMs and IOHMM achieve respectively 75% and 63% average recognition rate. A deeper analysis of the results obtained with both HMMs and IOHMM (Figure 4.6) shows that two-handed gestures are very well classified. Few mistakes happen between 'swim' and 'clap' gestures.

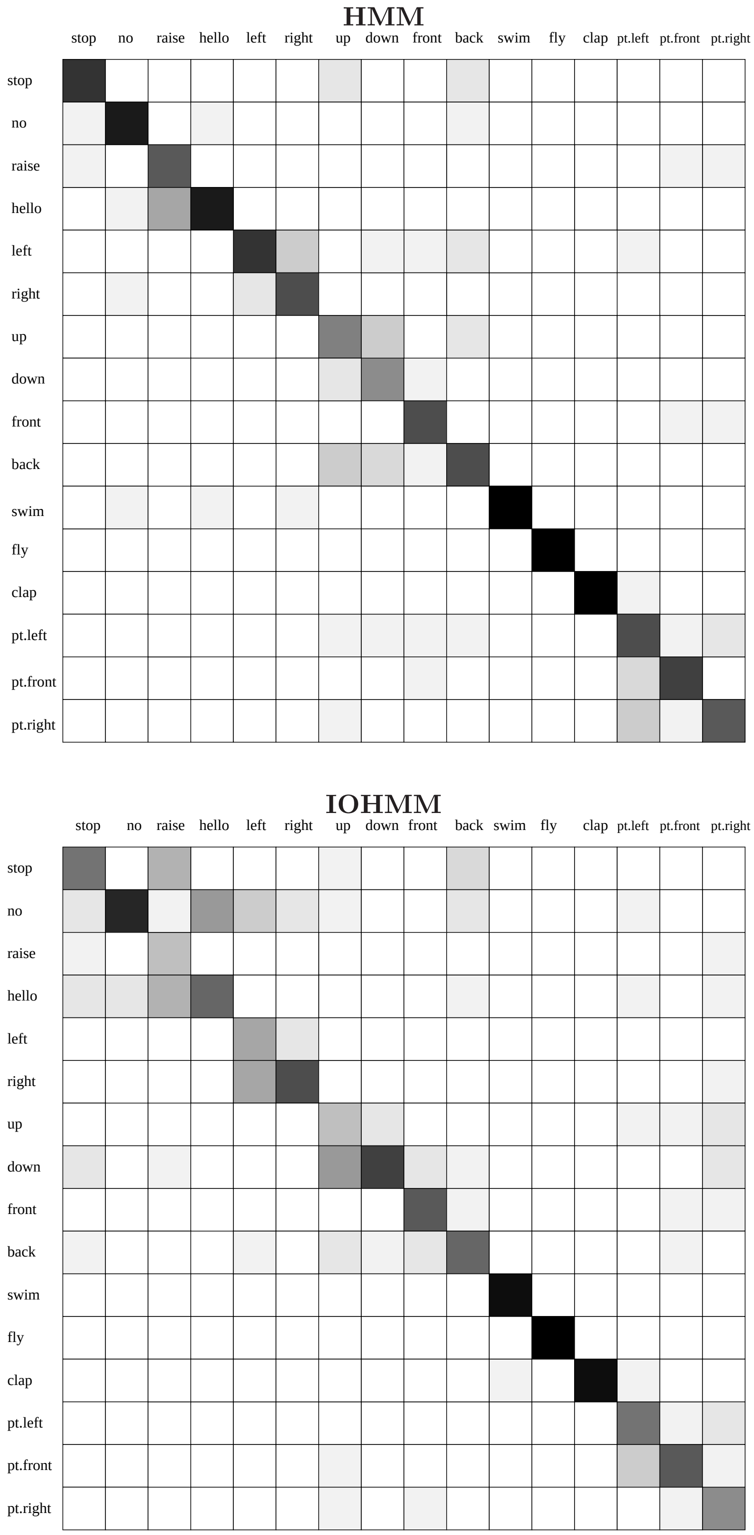


FIG. 4.6 – Confusion matrix for IOHMM and HMM on the test set (rows : desired, columns : obtained). Black squares correspond to the well-classified gestures.

Hand Gesture Class	stop	no	raise	hello	left	right	up	down	front	back	swim	fly	clap	pt.left	pt.front	pt.right
HMM	81.6	86.2	63	91.6	80.4	72	50.6	43.6	71.6	59.2	98.6	98.6	99.2	58.2	73	65.4
IOHMM	56.4	84.6	24	58.2	32.8	69.2	26.4	74.8	65.2	58.4	96.4	98	97.2	54.6	66.2	42.6

Tab. 4.3 – Recognition rate (%) on the test set between HMMs and IOHMM.

Concerning one-handed gestures, a misclassification between the 'stop', 'no/wipe', 'raise' and 'hello/wave' gestures occurs. According to Table 4.1, the only differences between these four gestures are the hand level and the oscillatory movement of the hand from the left to the right. HMMs have no problem to model the oscillatory movement of these gestures (average recognition rate : 85%). Features extracted from these hand gestures contain enough information for the HMMs to recognize them as there are two blocks 'stop-no' and 'raise-hello' around the diagonal. Only the 'raise' gesture is misclassified with the 'hello' gesture. For the IOHMM, a block around the diagonal is still visible for these gestures, but more blurry. The 'no' and 'hello' gestures are misclassified one with the other. This shows that the oscillatory characteristic of these hand gestures is still well modeled. The classification is more difficult for the 'stop' and 'raise' gestures showing that the hand level does not permit to discriminate between these two hand gestures.

Concerning the positioning gesture category ('left', 'right', 'up', 'down', 'front' and 'back' gestures), the block around the diagonal of the confusion matrices show that both HMMs and IOHMM differentiate quite accurately this category of hand gestures from the others. But it has difficulties to provide correct recognition within this category of gestures. But hand gestures are mostly misclassified by pairs, according to the direction of the waving movement : 'left/right', 'up/down' and 'front/back'. For the IOHMM, the 'right' and 'down' gestures are better classified than the 'left' and 'up' gestures. It can be explained by the fact that all people gesturing are right-handed, then the amplitude of these gestures is larger, hence providing a better classification of these two gestures.

For pointing gestures, HMMs and IOHMM differentiate also quite accurately this category of hand gestures from the others. The 'point left' gesture is misclassified by the HMMs with the 'point front' and 'point right' gestures and IOHMM misclassified this hand gesture only with the 'point front' gesture. Also, IOHMM misclassifies this category of hand gestures with the positioning gestures. The location of the hand at the end of the pointing gesture is not precise enough to give discriminant information to the IOHMM and to the HMMs. Furthermore, even if modeling well oscillatory hand gestures, IOHMM has difficulties to differentiate non-oscillatory movements from oscillatory ones.

4.6 Discussion

For that particular task, HMMs perform better than IOHMM. The performances of the IOHMM can be explained by a lack of training data. As the emission and transition probabilities are dependent on the input sequence, more data are needed to tune the hyper-parameters. The quantization process can also explain the performances of the IOHMM. When quantizing data, some information is lost thus degrading the recognition rate of the IOHMM. To validate this hypothesis, it would be necessary to run experiments using discrete HMMs on quantized data to see the evolution of performances.

The database contains numerous oscillatory hand gestures. Results obtained with both HMMs and IOHMM show that these oscillatory characteristics are quite well modeled. However, features extracted (3D trajectories and deltas) are not discriminant enough to recognize hand gestures within this category of oscillatory hand gestures, even if HMMs provide better results than IOHMM on this task. To model the hand gestures, one HMM is trained per hand gesture class whereas one IOHMM is trained to discriminate within the hand gesture classes which is a much more difficult task.

Two-handed gestures ('swim', 'clap' and 'fly') are very well recognized compared to one-handed gestures. Indeed, the average recognition rate on two-handed gestures is very high for both HMMs and IOHMM (98.8% and 97.2% average recognition rate respectively). On the contrary, the average recognition rate for one-handed gestures using HMMs is equal to 64.11% and is equal to 54.85% when using IOHMM. This improvement in performance with two-handed gestures show that two-handed gestures help disambiguate the HGR process.

Chapitre 5

Two-Handed Gesture Recognition

The present chapter is concerned with the recognition of two-handed gestures. Very few research has been done on two-handed gesture recognition. Furthermore, there is a lack a available two-handed gesture databases with rigorous experiment protocol. Therefore, to investigate the use of the two state-of-the-art sequence processing algorithms introduced in the previous chapter, a two-handed gesture database has been recorded. The goal of the two-handed gestures presented in this chapter is to manipulate virtual objects on a screen. We provide results obtained with different features, using images of one or two cameras, as well as features of one and two hands.

5.1 Scope of the Problem

In Section 1.3 we have described the three main problems related to the creation of gesture-based HCI. These difficulties still arise with two-handed gestures.

- **Segmentation of the two hands** : The problem now is to extract the two hands in images. Methods used for the segmentation of only one hand in images can still be applied for the extraction of both hands. The only supplementary difficulty lies in the fact that it is now necessary to distinguish the right hand from the left hand. The initial conditions can help disambiguate this situation.
- **Tracking** : Even if the tracking of one hand was not an easy task, tracking of both hands becomes even more difficult. The main problem of two-handed gestures lies in the high probability of one hand occluding the other one while gesturing. In that case, recovering hands after tracking is a very difficult problem. In order to ease the correspondence problem, users can wear markers or colored gloves. As in this thesis we are only focusing on the recognition task, we have chosen this last solution to ease the tracking and hand recovery after occlusion.
- **Recognition** : All the models available for one-handed gesture recognition can also be applied to the recognition of two-handed gesture. It is the case for the HMM and IOHMM used in the previous chapter.

5.2 Two-Handed Gesture Database

For this second database, two simple cameras (standard Web-cams with USB port) have been used to capture the gestures. As the goal of these gestures is to manipulate virtual objects on a screen, they occur on a desk in front of a display. Figure 5.1 shows the set-up used to record the data. The acquisition is synchronized at 12 images per second.

The database consists of 7 different two-handed gestures. These gestures are manipulative ones, mostly rotation gestures in each direction. For each gesture, two different views were recorded, one

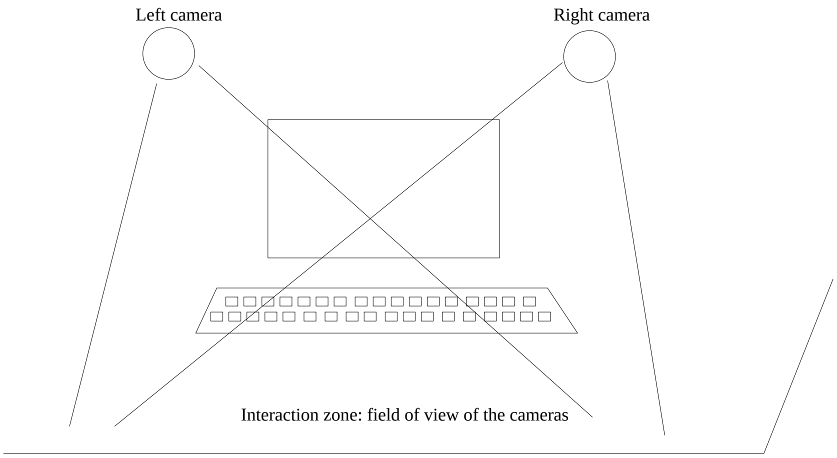


Fig. 5.1 – Camera set-up for recording the two-handed database.

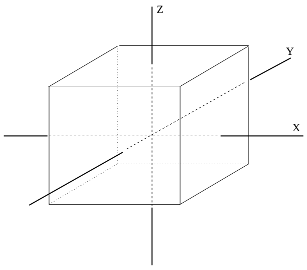
from each camera. Each gesturer wears two colored gloves : one blue and the other yellow (Figure 5.2) in order to facilitate hand tracking.

Fig. 5.2 – Point of view from the right and left cameras.

Seven people performed the gestures, with 2 sessions and 5 video sequences per session and per gesture. Thus, a total number of 10 video sequences per person and per gesture were recorded. The average duration of sequences is not longer than 2 or 3 seconds.

Our gestures are the following :

– Rotate front / Rotate back (along the x-axis)
– Rotate up / Rotate down (along the y-axis)
– Rotate left / Rotate right (along the z-axis)
– Push (move the hands forward)



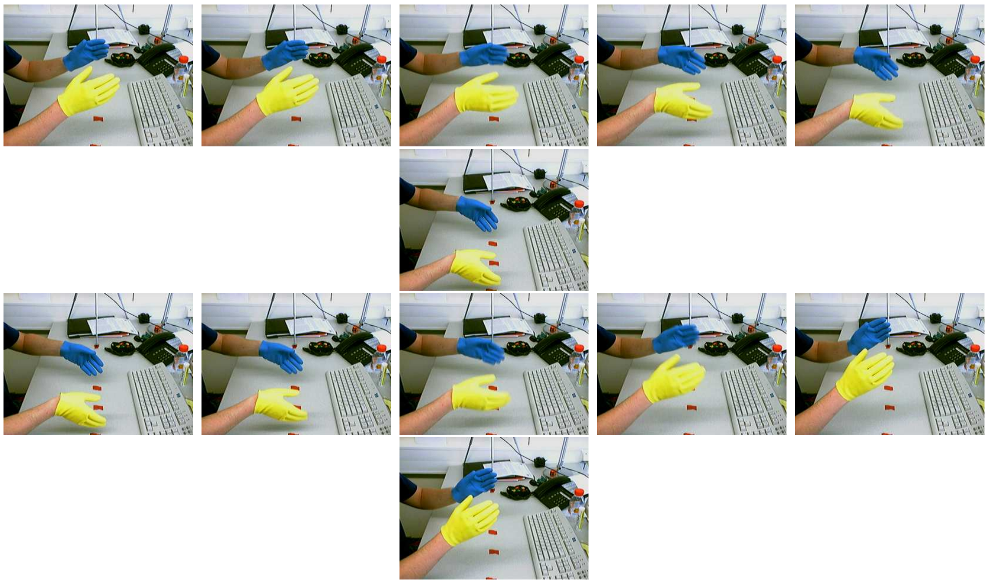


FIG. 5.3 – Rotate-front gesture sequence, top : preparation phase, bottom : retraction phase

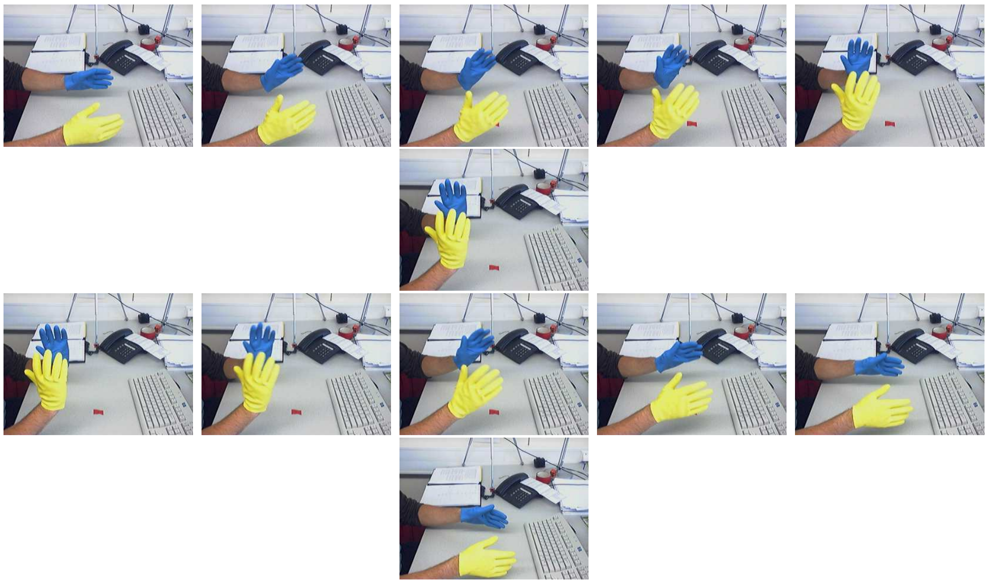


FIG. 5.4 – Rotate-back gesture sequence, top : preparation phase, bottom : retraction phase

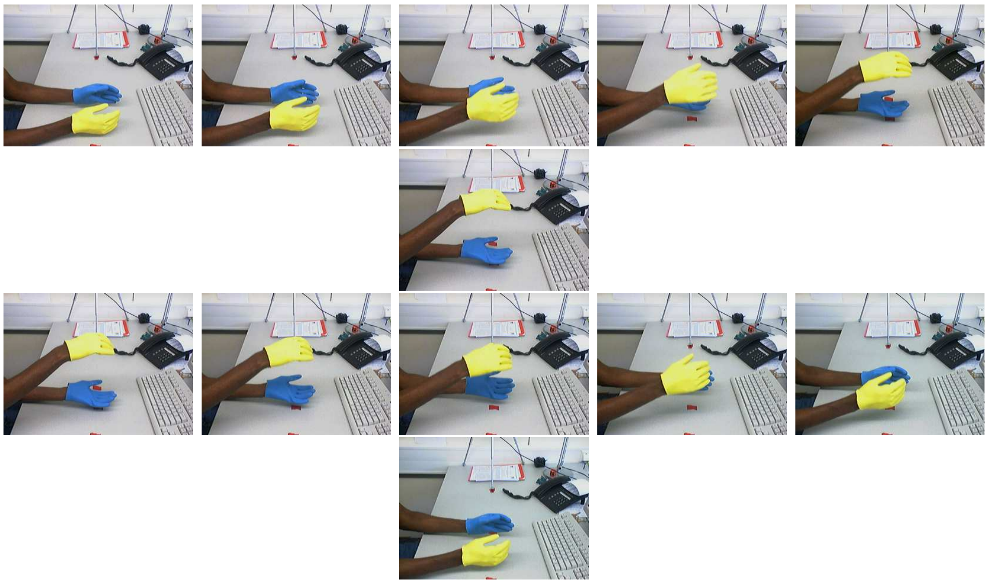


Fig. 5.5 – Rotate-up gesture sequence, top : preparation phase, bottom : retraction phase

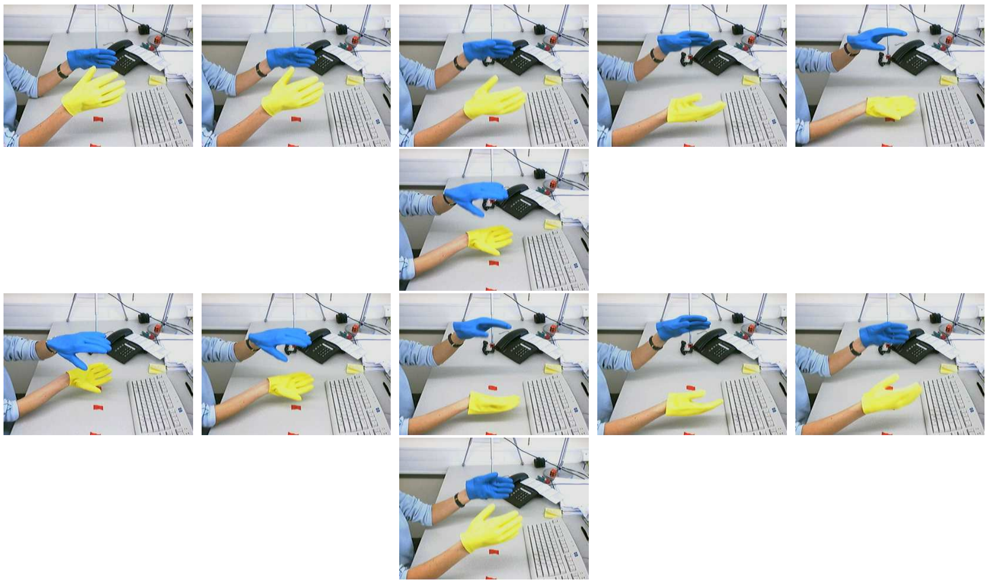


Fig. 5.6 – Rotate-down gesture sequence, top : preparation phase, bottom : retraction phase

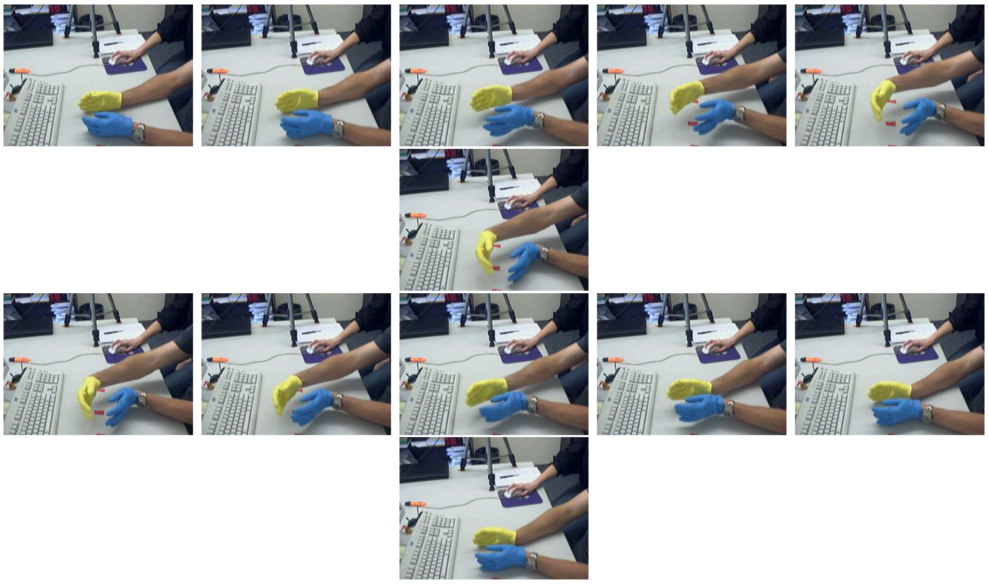


FIG. 5.7 – Rotate-left gesture sequence, top : preparation phase, bottom : retraction phase

FIG. 5.8 – Rotate-right gesture sequence, top : preparation phase, bottom : retraction phase

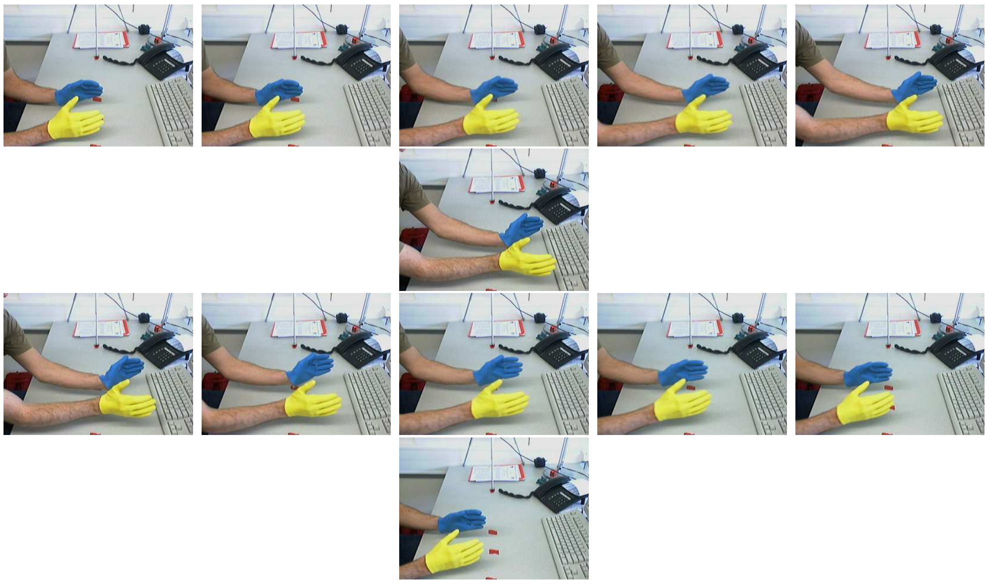


FIG. 5.9 – Push gesture sequence, top : preparation phase, bottom : retraction phase

5.3 Sequence Processing Algorithms for Two-Handed Gesture Recognition

This section describes the features extracted from image sequences and required preprocessing.

5.3.1 Feature Extraction

As a first step, on each image of gesture sequences, a simple lookup table filter (Section 2.2) is applied (one for each glove color : blue and yellow). The pixel of the two colors of interest are filtered in the image. Using the detected pixels in the images, each hand is approximated to a single blob [118] which is represented as a Gaussian distribution. The position of the detected pixels permit to compute the mean and variance of this Gaussian distribution. Each hand is thus approximated to an ellipse (Figure 5.10).

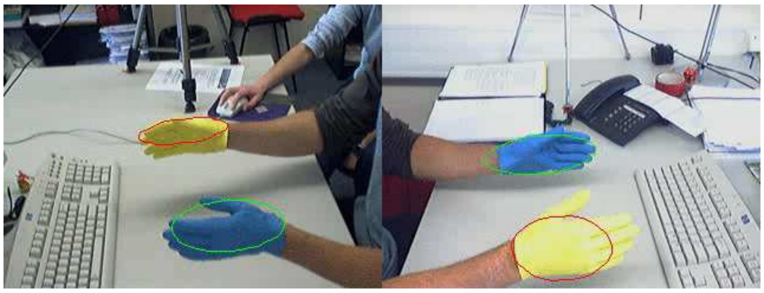


FIG. 5.10 – Representation of the hand blobs for the left and right cameras. The approximated ellipse are represented in red for the right hand and in green for the left hand.

The mean (μ_x, μ_y) of each blob, called 'center' of the bounding ellipse around the user's hand, is used as a feature. Let call a the half major axis and b the half minor axis of the ellipse. The eccentricity is computed using the formula $e = \sqrt{1 - \frac{b^2}{a^2}}$, and the surface of the ellipse is $s = \pi \times ab$. The angle α

between the major axis of the ellipse and the horizon can also be computed. Figure 5.11 shows these features on a stylized human hand.

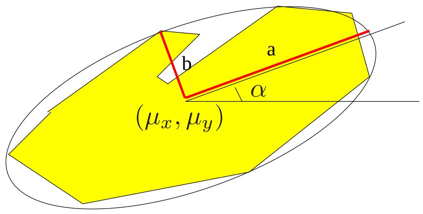


FIG. 5.11 – Features for two-handed gesture recognition : μ is the center of the Gaussian distribution represented by the surrounding ellipse. The major axis a and minor axis b are represented in red. The angle α represents the orientation of the hand.

For some gestures, it happens that a hand occludes the other one. In such cases, features cannot be computed for the occluded hand. Therefore, features of the first previous frame where the hand was not occluded are kept until the end of the occlusion. New features are then computed and updated.

The global feature vector X corresponds to the x and y coordinates of the center of each blob for images from the right and left camera, the angle between the horizon and the main axis of the ellipses in both images, the surface size, and the eccentricity of the two ellipses for the left and right camera images. Thus $X \in \mathbb{R}^{20}$.

5.3.2 Preprocessing

To perform recognition with the HMMs, no preprocessing on the feature vectors is needed. But to efficiently use discrete IOHMM, a quantization step on the data is needed. The output sequence still encodes the gesture classes : $y_t = \{0, \ldots, 6\}, \forall t$. To model more closely the class distribution of the data, a K-means algorithm [58] is applied class per class on the input features.

For each gesture class, the number of clusters has been selected class per class. The 7 resulting K-means models have been merged into a single model of 565 clusters. As for experiments with the previous hand gesture database, hand gestures sequences have been quantized into discrete values.

5.4 Experiments

Results obtained with HMMs and IOHMM on the first hand gesture database show that two-handed gestures are better recognized than one-handed gestures (Section 4.6). In this section, these algorithms are evaluated on the second hand gesture database containing only two-handed gestures. Some more experiments were performed using HMMs. To highlight the discriminative power of features, we performed experiments using various features. To show the benefit of using two cameras in term of robustness of the recognition system, experiments using one camera were also conducted. For each two-handed gesture, hands move in a symmetric way. To show that features extracted from image sequences of both hands are leading to better results, experiments using one hand have also been performed.

5.4.1 Experiments with HMMs and IOHMM

Parameter Tuning

In the experiments, both left/right and full connect topologies for both HMMs and IOHMM have been used. To find the optimal hyper-parameters (number of states of the discrete IOHMM, number of states and number of Gaussians for the HMMs), the following protocol has been used. The two databases have been split into three subsets : the training set T, the validation set V and the test set Te.

Different possibilities for the hyper-parameters have been tried on T. The selection of the best parameters has been done on V. Finally, a model has been trained on both T and V and tested on Te. The best results were obtained with the following hyper-parameters (Table 5.1)

	HMMs	IOHMM
Topology	left-right	full connect
Number of Gaussians	1	$\times$
Number of states	12	17

Tab. 5.1 – Hyper-parameters to test HMMs and IOHMM on the two-handed gesture database.

Results

Table 5.2 shows results obtained with the same two algorithms on the second database described in the previous chapter. For more details, the confusion matrix is also provided (Figure 5.12).

two-handed gesture	r-left	r-right	r-up	r-down	r-front	r-back	push
HMM	96.67	100	100	96.67	100	96.67	96.67
IOHMM	26.67	13.33	66.67	66.67	60	16.67	53.33

Tab. 5.2 – Recognition rate (%) on the test set of two-handed gestures, using HMMs and IOHMM.

HMMs achieve 98% average recognition rate. Results with IOHMM are very poor as the IOHMM only achieve 43% average recognition rate. Even with few gesture examples, HMMs perform very well on the test set. As the database of two-handed gestures is very small, we can expect the results to slightly degrade if we increase the number of test data.

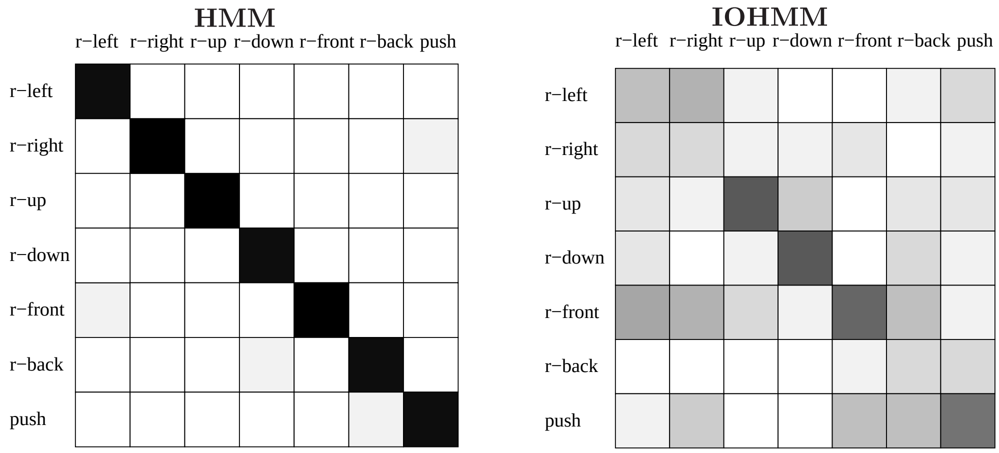


Fig. 5.12 – Confusion matrix for IOHMM and HMM on the test set for the two-handed gesture database (rows : desired, columns : obtained). Black squares correspond to the well-classified gestures.

Concerning the IOHMM, only the 'rotate-up', 'rotate-down' and 'push' gestures are fairly classified. The 'rotate-up' and 'rotate-down' gestures are the only rotational gestures in which the movement of the hand is facing the cameras. And the 'push' gesture is different from all the other gestures. These results show that the features themselves are the core of the misclassification problem in the case of this second database. During the recording of this two-handed gesture database, no preliminary advice has been given to the gesturers. They performed the gestures in the most natural way. The two-handed gesture sequences contained in this second database vary greatly from one gesturer to another. The position of the hands in the image also varies from one gesture to another and from one person to another. The two-handed gestures recorded are thus more difficult to recognize. Only HMMs are not affected by this variability and change in absolute position of the hands in images. Features extracted are well adapted to the HMMs as they permit to model accurately the different gestures, but they are not discriminant enough to have good recognition with the IOHMM.

Recognition results show that features extracted are suited to HMMs. HMMs are able to model data with great accuracy. For instance, the variation of the angle between the main axis of the ellipse and horizon is characteristic of the 'rotate-right' and 'rotate-left' gestures, as well as the 'rotate-front' gestures. Figure 5.13 and Figure 5.14 show the evolution of the angle α with time for all the hand gestures. Some specific pattern are visible in these graphics.

5.4.2 Experiments using various Features

Surface and eccentricity features are not very accurate because gloves are very loose. To evaluate the usefulness of these features, another experiment has been conducted with HMMs only (as HMMs have shown to perform better on this database). HMMs have been trained with only the following features : μ_x, μ_y, α, and without the eccentricity e nor the surface s features (Section 5.3.1). The best result has been obtained with 14 states and 45 Gaussians per state. Table 5.3 shows the recognition rate on the seven two-handed gestures. HMMs achieve 72% average recognition rate as compared to 98% with all the features.

	rotate left	rotate right	rotate up	rotate down	rotate front	rotate back	push
HMM	100	90	40	33.33	56.67	93.33	93.37

Tab. 5.3 – Recognition rate (in %) on the test set using HMMs. Only the mean μ of the Gaussian distribution approximating the user's hand, as well as the orientation angle have been kept.

The coordinates of the center of each blob as well as the angle are characteristic of the hand gestures. With only these three types of features, the recognition rate of the 'rotate-up', 'rotate-down' and 'rotate-front' gestures decreases. This experiment shows that even if no visible pattern can be extracted from the evolution of the eccentricity and surface features, some information is still present which helps the recognition process.

5.4.3 Experiments using one Camera

When recording the two-handed gesture database, we made the hypothesis that gestures would be easier to recognize when using features extracted from two cameras, even without using stereoscopic vision. To verify this hypothesis, some experiments were performed using HMMs and features of both hands extracted from only one camera. Table 5.4 and table 5.5 show the confusion matrix of the recognition rate using HMMs for the right and left camera respectively.

For the right camera, the best results have been obtained with a left-right topology, 17 states and 10 Gaussians per state. Concerning the results with the left camera, best results were reached with a full-connect topology, 9 states and 60 Gaussians per states. The average recognition rate using information extracted from the right camera is equal to 83.33%. Using the information from the left camera, the recognition rate increases up to 97.61%.

With features extracted using both cameras, best results were obtained with a left-right topology, 12 states and 1 Gaussian per state. The confusion matrix is provided in table 5.6.

In images taken from the right and left camera, difference in lighting conditions is noticeable. It could be due to a different set-up in the calibration of the web-cameras, or due to the office neon-light more obvious from the point of view of the right camera. As color segmentation is applied to extract

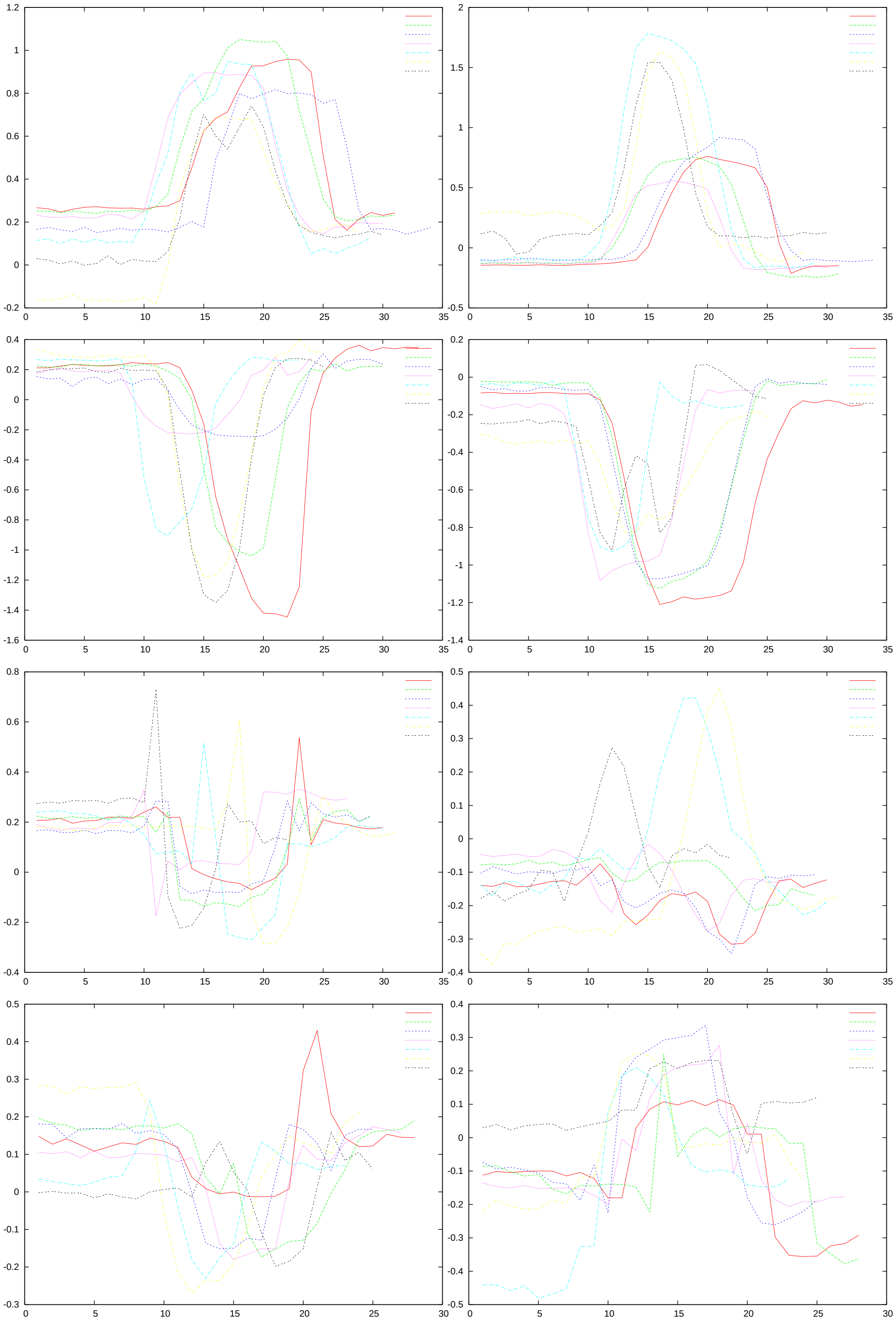


FIG. 5.13 – Evolution of the angle (in degrees) with time, from the point of view of the left (left column) and right (right column) cameras, respectively for the 'rotate-left' (first row), 'rotate-right' (second row), 'rotate-up' (third row), 'rotate-down' (last row) gestures. The x-axis represents the time (frame number) and the y-axis represents the angle. Each colored line is a different realization of the same gesture.

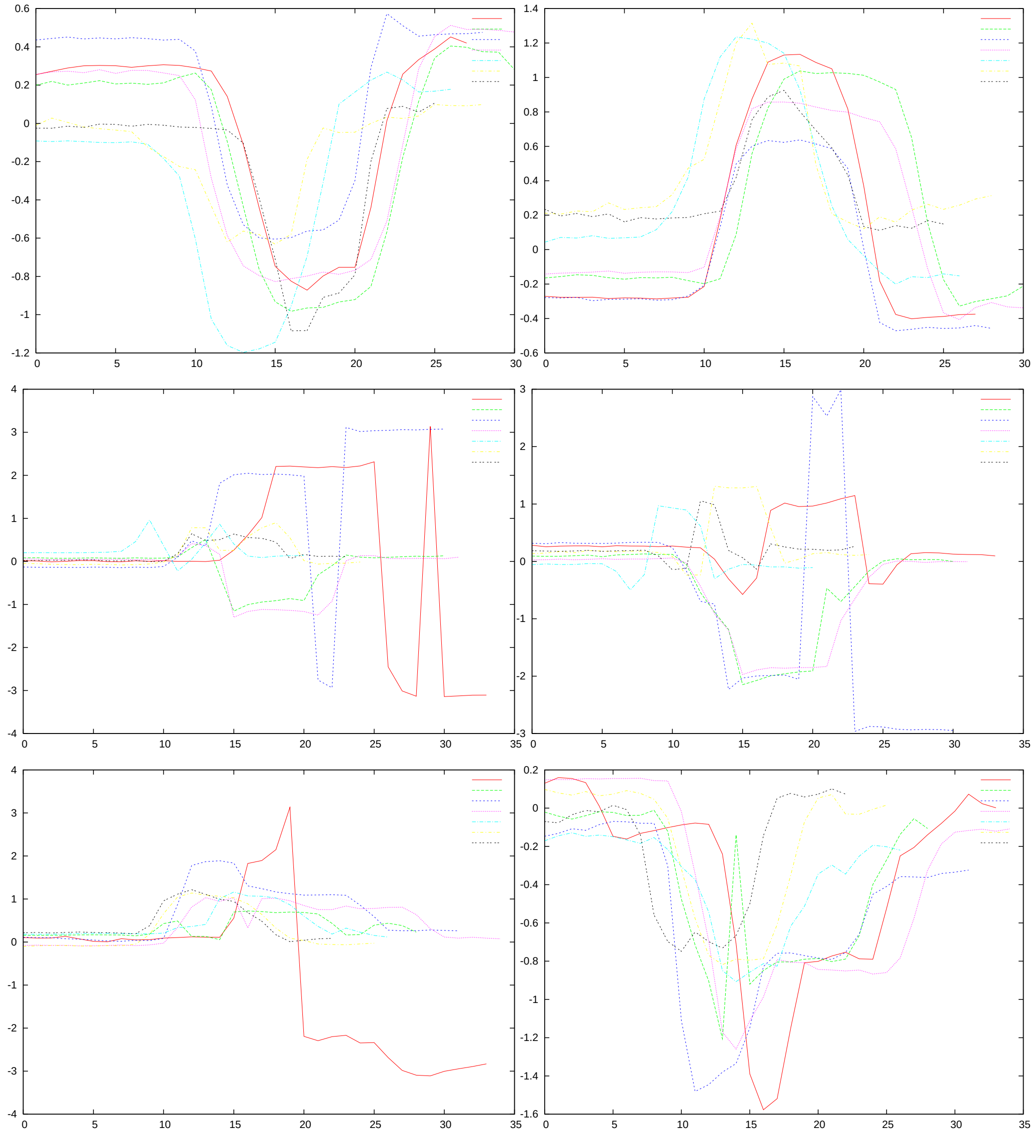


FIG. 5.14 – Evolution of the angle (in degrees) with time, from the point of view of the left (left column) and right (right column) cameras, respectively for the 'rotate-front' (first row), 'rotate-back' (second row) and 'push' (last row) gestures. The x-axis represents the time (frame number) and the y-axis represents the angle. Each colored line is a different realization of the same gesture.

	r-left	r-right	r-up	r-down	r-front	r-back	push
r-left	**86.67**	0	0	0	0	0	0
r-right	0	**53.33**	0	0	0	0	0
r-up	0	0	**100**	0	0	0	0
r-down	0	0	0	**66.67**	0	0	0
r-front	0	26.67	0	0	**76.67**	0	0
r-back	3.33	6.67	0	23.33	23.33	**100**	0
push	10	13.33	0	10	0	0	**100**

TAB. 5.4 – Confusion matrix for HMMs on the test set of two-handed gestures and for the right camera.

	r-left	r-right	r-up	r-down	r-front	r-back	push
r-left	**93.33**	0	0	0	3.33	0	0
r-right	0	**100**	0	0	0	3.33	0
r-up	0	0	**96.67**	0	0	0	0
r-down	0	0	3.33	**100**	0	0	0
r-front	6.67	0	0	0	**96.67**	0	0
r-back	0	0	0	0	0	**96.67**	0
push	0	0	0	0	0	0	**100**

TAB. 5.5 – Confusion matrix for HMMs on the test set of two-handed gestures and for the left camera.

	r-left	r-right	r-up	r-down	r-front	r-back	push
r-left	**96.67**	0	0	0	0	0	0
r-right	0	**100**	0	0	0	0	0
r-up	0	0	**100**	0	0	0	0
r-down	0	0	0	**96.67**	0	0	0
r-front	3.33	0	0	0	**100**	0	0
r-back	0	0	0	3.33	0	**96.67**	0
push	0	0	0	0	0	3.33	**96.67**

TAB. 5.6 – Confusion matrix using features of both hands extracted from the right and left cameras.

the hands, these changes in lighting conditions influence the quality of the hand segmentation process. This explains the degrading in performance while using only features extracted from the right camera. When there is no trouble in the hand segmentation process and thus the hand tracking, performances are very high. This is shown by results obtained using images from the left camera. Unfortunately real-life conditions are not optimal and degraded hand segmentation and tracking is highly probable in real-life conditions. When using features extracted from both cameras, the recognition rate is comparable to the one obtained when no major problem occur during hand tracking. Therefore, using images extracted from both cameras leads to a more robust recognition system.

A high number of parameters are trained when using only one camera. Best results are obtained with 17 states, 10 Gaussians per state and a left-right topology for the right camera and 9 states, 60 Gaussians per state and a full-connect topology for the left camera. Using both cameras, HMMs with a left-right topology, 12 states and 1 Gaussian per state is enough. Gestures are more difficult to model when using features provided by only one camera. In conclusion, features extracted from two cameras lead to simpler and more robust models, even if the number of features doubles when using two cameras.

5.4.4 Experiments using one Hand

As the two-handed gestures presented here are perfectly symmetrical, one can argue that using features from both hands is useless. In this experiment we intend to show the contrary. For that purpose, experiments using features of one hand extracted using images from both cameras have been performed. For these experiments, only HMMs have been used. For the right hand, best results are obtained with a left-right topology, 15 states and 1 Gaussian per state. For the left hand, best results are obtained with a full-connect topology, 19 states and 45 Gaussians per state. The average recognition is of 90% for the right hand and 91.43% for the left hand. Confusion matrices using these different features are shown in Table 5.7 and Table 5.8.

Experiments have shown that it is better to recognize two-handed gestures using both hand with both cameras (98% average recognition rate) than using only one hand also with both cameras (respectively 90% average recognition rate for right hand only and 91.4% for left hand only).

	r-left	r-right	r-up	r-down	r-front	r-back	push
r-left	**100**	0	0	0	0	0	0
r-right	0	**73.33**	0	0	0	0	6.67
r-up	0	0	**83.33**	0	0	0	0
r-down	0	0	0	**86.67**	0	0	0
r-front	0	0	0	13.33	**93.33**	0	0
r-back	0	23.33	16.67	0	3.33	**100**	0
push	0	3.33	0	0	3.33	0	**93.33**

TAB. 5.7 – Confusion matrix using features of the right hand extracted using images from both cameras.

	r-left	r-right	r-up	r-down	r-front	r-back	push
r-left	**100**	0	0	0	0	0	0
r-right	0	**100**	10	3.33	23.33	3.33	0
r-up	0	0	**83.33**	0	0	0	0
r-down	0	0	6.67	**96.67**	0	0	0
r-front	0	0	0	0	**76.67**	0	3.33
r-back	0	0	0	0	0	**86.67**	3.33
push	0	0	0	0	0	10	**93.34**

TAB. 5.8 – Confusion matrix using features of the left hand extracted using images from both cameras.

5.5 Discussion

Experiments performed on the two-handed gesture database showed that best performances were obtained with HMMs. IOHMM performs poorly on this purely two-handed gesture database. Furthermore the recognition is not uniform with IOHMM. Some gestures are quite well recognized whereas some others are barely properly classified. The push gesture is very different from all the other gestures which makes it easier to recognize. It is also the case for the 'rotate-up' and 'rotate-down' gestures. They are the only gestures where hands are moving in the opposite direction. Concerning the 'rotate-front' gesture, the angle of the ellipsoid is very discriminative which explains the higher recognition rate. The quantization process could explain the poor results obtained with IOHMM. To validate this hypothesis it would be necessary to run experiments on quantized data using discrete HMMs to see if performances degrade. Furthermore HMMs and IOHMM are not modeling information in the same way. A discriminative approach is taken with the IOHMM whereas for HMMs, it consists of a generative approach. The problem of discriminating between classes is more complex and thus requires more data. A lack of training data due to the small size of the database could explain the poor performances of the IOHMM.

Concerning the proposed features, they contain enough information to model accurately data using HMMs. To evaluate the discriminative property of the features, experiments were performed using various features. Results using only μ and α provide 72% average recognition rate. Compared to the average recognition rate obtained with the complete set of features (98%), performances clearly degrade. This shows that the mean μ of the Gaussian distribution approximating the user's hand, as well as the orientation angle α are very efficient. Furthermore, results also show that the remaining features provide even better results.

Experiments using features extracted from one or two cameras have also been performed. When using the left camera only, an average recognition rate of 97.61% is obtained. The recognition rate decreases down to 83.33% when image sequences of the right camera are used. Recording conditions from the right and left camera are different. This degrading in performance shows that recognition process is perturbed when hand segmentation and tracking are not optimal. When both cameras are used, the average recognition rate is equal to 98%. The system is thus more robust when images from both cameras are used. Multiple sources of information (here the two cameras) provide a system able to cope more easily with unperfect hand segmentation and/or tracking.

It can be argued that in these gestures the user's hands move in a very symetric way and thus that two hands are not useful. To confirm or deny this assuption, experiments using both cameras and only one hand have then been performed. Performances obtained with only one hand (90% average recognition rate for the right hand and 91.43% for the left hand) are not as good as the one obtained when using both hands (98% average recognition rate). It shows that using features from two hands ease the recognition process.

Chapitre 6

Conclusion

The three main problems related to vision-based gestural interfaces are hand segmentation, tracking and recognition. In this thesis, we only considered the recognition task for both hand postures and gestures. We proposed a novel technique for hand posture detection and recognition. Results were compared on a benchmark database with a baseline method. Due to a lack of available databases to evaluate hand gesture recognition techniques, two databases were recorded during the thesis. These databases are publicly available and experimental protocols are provided. Two state-of-the-art sequence processing algorithms in the field of HGR were evaluated on the first hand gesture database. We also investigated the use of these approaches on the second database containing only two-handed gestures.

6.1 Hand Posture Recognition

In Chapter 3, we proposed a boosting approach based on MCT features, firstly applied to the face detection task [49] to detect and recognize several hand postures. Experiments were performed on a hand posture benchmark database. Images contained in this database were recorded against three types of background : uniform light, uniform dark and complex background. We designed two experimental protocols. To train classifiers using Protocol 1, only images against uniform background are used whereas with Protocol 2 both images against uniform and complex background are used. We compared the proposed approach to a baseline MLP technique. Results show that the MCT-Boost method improves recognition over the baseline MLP approach. Furthermore, results obtained with the two different protocols demonstrate that adding images against complex background improves recognition performances.

6.2 Hand Gesture Recognition

We motivated the use of two-handed interaction in Section 2.5. Little research has been carried out to recognize this type of hand gestures. Furthermore, there is a lack of common evaluation databases in the field of Hand Gesture Recognition (HGR). In this thesis, we provided two hand gesture databases and designed strict experimental protocols. These databases are publicly available to the research community. The first database comprises both one- and two-handed gestures whereas the second database contains only two-handed gestures. On the first database, we evaluated two sequence processing algorithms, namely Hidden Markov Models (HMM) and Input-Output Hidden Markov Model (IOHMM). Best results were obtained with HMMs. However, both HMMs and IOHMM gave excellent results for the recognition of two-handed gestures. Two-handed gestures thus help disambiguate the recognition process.

We then proposed to investigate the above techniques for the recognition of two-handed gestures. The second database has been recorded using two cameras. Best results were obtained with HMMs. With this model, further experiments were conducted and various features were used to show their discriminative power. Moreover, experiments using information extracted from only one camera were conducted. Results show that using information of two cameras is better than using information of one camera only, and thus that adding multiple sources of information leads to a more robust recognition system. Experiments using only features extracted from one hand were also performed. The results show a degrading in the performances compared to the recognition rate obtained with features extracted from both hands. Even if two-handed gestures are symmetric, features extracted from two hands ease the recognition process.

Recently, the research community has shown less interest for HPR and HGR applied to HCI. Yet there are some application domains where gestural interaction is an improvement over standard interaction techniques. These domains are virtual and augmented reality, wearable computing, interaction with large display as well as data manipulation and visualization. Furthermore, such applications can profit from two-handed interaction. We thus hope this research will open new directions in the field of HPR and HGR.

6.3 Future Directions

The proposed MCT-Boost approach for hand posture recognition has been applied to already cropped images. A scanning approach could be investigated to overcome this limitation. This method consists of scanning the image at different sizes and locations using a hand posture classifier. As this technique produces a lot of false detections, it could be used in conjunction with a skin color detector to discard these false detections.

For the task of hand gesture recognition, we are only dealing in this thesis with already segmented hand gestures. A next step would be the recognition of a sequence of hand gestures. This task, known as gesture spotting, consists of the recognition of hand gestures performed during the gesture sequence as well as finding the start and end position of these hand gestures. Gesture spotting can also be extended to the recognition of known gestures and rejection of unknown gestures.

To efficiently use gestural HCI, it is necessary to be able to deal with bare-hand interaction. This task is related to hand tracking as well as two-handed gesture tracking. This is a very difficult problem particularly when dealing with two-handed gestures. In that case, both hands are in the view field of the camera(s). Hands are thus very often occluding each other. Few research has already been done to deal with this problem. Some work still needs to be done to efficiently track both hands without the help of colored gloves.

Annexe A

Acronyms

CGM	Constrained Generative Model
DOF	Degree Of Freedom
DTW	Dynamic Time Warping
EM	Expectation-Maximization
FSM	Finite State Machine
HCI	Human-Computer Interaction
HGR	Hand Gesture Recognition
HMM	Hidden Markov Model
HPR	Hand Posture Recognition
IOHMM	Input-Output Hidden Markov Model
MCT	Modified Census Transform
MLP	Multi-Layer Perceptron
MSE	Mean-Square Error

Annexe B

Experimental Protocols for the Hand Posture Database

B.1 Description

The Jochen Triesch Hand Posture database contains pgm images of 10 hand postures ('A', 'B', 'C', 'D', 'G', 'H', 'I', 'L', 'V' and 'Y') against 3 background types.

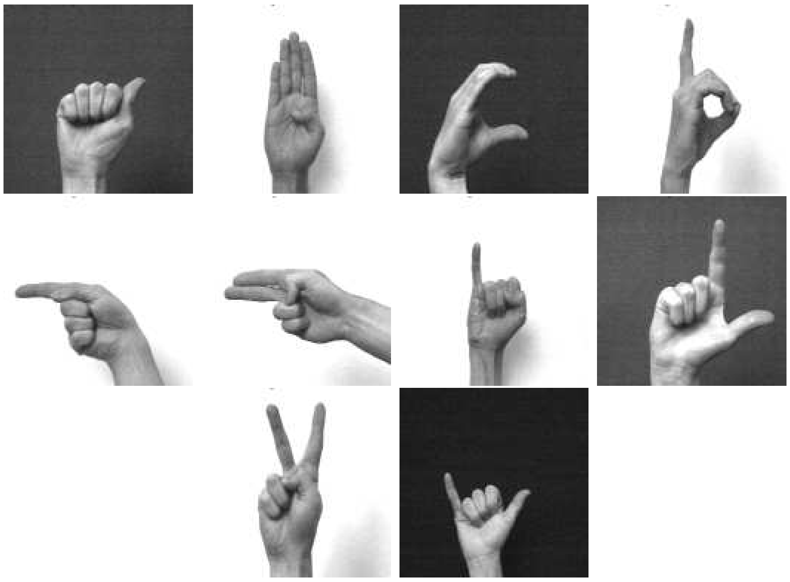


Fig. B.1 – Examples of the ten different hand postures.

The database contains 128×128 gray-scale images. The hand signs were performed by 24 different gesturers. Among the 720 images, the `haloosh3` and `mbeckev2` files were lost by the creator of the database. The database can be downloaded from `http ://www.idiap.ch/resources.php` under the section *Hand Posture and Gesture Datasets*.

B.2 Naming Convention

The naming convention for image files is

`<name><posture><background>.ppm`

where name corresponds to the person performing the hand posture, posture is the hand posture label and background is the background type. The list of names has been split into three subsets A, B and C (Table B.1). Possible entries for the postures are : `a, b, c, d, g, h, i, l` or `v`, and for the background fields : `1, 2` or `3`.

Subset	Name
A	bfritz
	haloos
	mpoetz
	mschue
B	hneven
	kbraue
	szadel
	tmaure
C	ckaise
	ermael
	gbanav
	gpeter
	jtries
	jwiegh
	mbecke
	mkefal
	mrinne
	nkrueg
	orehse
	pleuch
	sagins
	tkersc
	unasch
	uschwa

TAB. B.1 – Naming convention, from left to right : subset A, B and C with entries to the field name.

B.3 Protocols

For experiments, the database has been split into three subsets : the train set T, validation set V and test set Te. Table B.2 presents the files contained in each subset for Protocol 1 and Protocol 2. The list of names for each set A, B and C is shown on Table B.1.

	Protocol 1			Protocol 2		
Background	1	2	3	1	2	3
A	T	T	Te	T	T	T
B	V	V	Te	V	V	V
C	Te	Te	Te	Te	Te	Te

TAB. B.2 – List of files used in the train, validation and test set. A, B and C refer to the three different sets containing person names. 1,2 and 3 refer to the background type.

Annexe C

Experimental Protocol for the Hand Gesture Database

C.1 Description

The Hand Gesture database consists of 16 gestures carried out by 20 different people. For each person and each gesture, 5 sessions and 10 shots per session have been recorded. The database can be downloaded from `http ://www.idiap.ch/resources.php` under the section *InteractPlay Dataset*.

C.2 Naming Conventions

The naming convention for text files is

`<gesture>_<person>_<session>_<shot>.txt`

where gesture $\in \{01, \ldots, 16\}$, person $\in \{00, 02, \ldots, 11, 13, \ldots, 15, 17, \ldots, 22\}$, session $\in \{1, \ldots, 5\}$ and shot $\in \{0, \ldots, 9\}$.

C.3 Experimental Protocol

The database has been split into three subsets : the train set T, validation set V and test set Te. Images of people 18-19-20-21-22 have been kept in T, 11-13-14-15-17 for V and 00-02-03-04-05-06-07-08-09-10 in Te for all sessions and all shots.

C.4 Acknowledgments

This database has been recorded by France Telecom R&D. Some special thanks go to J. Guerin and B. Rolland (from FTR&D Recherche et Développement) for recording and annotating the gesture database. Some thanks also go to O. Bernier for participating in the experiments conducted on this database.

Annexe D

Experimental Protocol for the Two-Handed Gesture Database

D.1 Description

The Two-Handed Gesture database consists of seven two-handed gestures carried out by seven different people. For each person and each gesture, two sessions and five shots per session have been recorded, except for one person for whom only one session of ten shots has been recorded. The database can be downloaded from `http ://www.idiap.ch/resources.php` under the section *Two-Handed Datasets*.

D.2 Naming Conventions

The naming convention for video files is

`<gesture>_<person>_<session>_<cam>_<shot>.avi`

where gesture $\in \{push, rotate\_front, rotate\_back, rotate\_up, rotate\_down, rotate\_left, rotate\_right\}$, person $\in \{florent, iain, olivier, seb, agnes, sileye, yann\}$, session $\in \{record00, record01\}$, cam $\in \{cam1, cam2\}$ and shot $\in \{session00, session01, session02, session03, session04\}$.

D.3 Experimental Protocol

The database has been split into three subsets : the train set T, validation set V and test set Te. Images of *florent* and *iain* have been kept in T, *olivier* and *seb* for V and *agnes, sileye* and *yann* in Te for all sessions and all shots.

Bibliographie

[1] K. Abe, H. Saito, and S. Ozawa. Virtual 3-d interface system via hand motion recognition from two cameras. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 2002.

[2] J. Alon, V. Athitsos, and S. Sclaroff. Accurate and efficient gesture spotting via pruning and subgesture reasoning. In *Proceedings of the IEEE Workshop on Human Computer Interaction*, 2005.

[3] J. Alon, V. Athitsos, Q. Yuan, and S. Sclaroff. Simulatneous localization and recognition of dynamic hand gestures. In *Proceedings of the IEEE Motion Workshop*, 2005.

[4] Y. Ariki, T. Takiguchi, and A. Sako. Recognition of hands-free speech and hand pointing action for conversational TV. In *Proceedings of the 13th Annual ACM International Conference on Multimedia*, 2005.

[5] V. Athitsos and S. Sclaroff. An appearance-based framework for 3d hand shape classification and camera viewpoint estimation. In *Proceedings of the International Conference on Face and Gesture Recognition*, 2002.

[6] V. Athitsos and S. Sclaroff. Estimating 3d hand pose from a cluttered image. In *Proceedings of the IEEE Conference on Computer Vision and pattern Recognition*, 2003.

[7] H.H. Avilès-Arriaga, L.E. Sucar, and C.E. Mendoza. Visual recognition of gestures using dynamic naive bayesian classifiers. In *Proceedings of the XXXIV Congreso de Investigación y Extensión del sistema Tecnológico de Monterrey*, 2004.

[8] K.A. Barhate, K.S. Patwardhan, S.D. Roy, S. Chaudhuri, and S. Chaudhury. Robust shape based two hand tracker. In *Proceedings of the International Conference on Image Processing*, 2004.

[9] T. Baudel and M. Beaudoin-Lafon. CHARADE : Remote control of objects using free-hand gestures. *Communications of the ACM*, 1993.

[10] Y. Bengio and P. Frasconi. An input output hmm architecture. *Advances in Neural Information Processing Systems*, 1995.

[11] Y. Bengio and P. Frasconi. Input/output hmms for sequence processing. *IEEE Transactions on Neural Networks*, 1996.

[12] O. Bernier and D. Collobert. Head and hand 3d tracking in real time by the em algorithm. In *Proceedings of the IEEE International Conference on Computer Vision - Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, 2001.

[13] M.K. Bhuyan, D. Ghosh, and P.K. Bora. Finite state representation of hand gesture using key video object plane. In *Proceedings of the IEEE Region 10 Conference*, 2004.

[14] M.K. Bhuyan, D. Ghosh, and P.K. Bora. Threshold finite state machine for vision based gesture recognition. In *Proceedings of the INDICON Annual IEEE Conference*, 2005.

[15] A. Bobick and J. Davis. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2001.

[16] R.A. Bolt. "put-that-there" : Voice and gesture at the graphics interface. *Computer Graphics*, 1980.

[17] R.A. Bolt and E. Herranz. Two-handed gesture in multi-modal natural dialog. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, 1992.

[18] Y. Boussemart, F. Rioux, F. Rudzicz, M. Wozniewski, and J.R. Cooperstock. A framework for 3d visualisation and manipulation in an immersive space using an untethered bimanual gesture interface. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, 2004.

[19] M. Bray, E. Koller-Meier, and L. Van Gool. Smart particle filtering for 3d hand tracking. In *Proceedings of the 6th IEEE International Conference on Automatic Face and Gesture Recognition*, 2004.

[20] L. Brethes, P. Menezes, F. Lerasle, and J. Hayet. Face tracking and hand gesture recognition for human-robot interaction. In *Proceedings of the IEEE International Conference on Robotivs and Automation*, 2004.

[21] L. Bretzner, I. Laptev, and T. Lindeberg. Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering. In *Proceedings of the Conference on Automatic Face and Gesture Recognition*, 2002.

[22] W. Buxton and B. Myers. A study in two-handed input. In *Proceedings of the Conference on Human Factors in Computing Systems*, 1986.

[23] M.C. Cabral, C.H. Morimoto, and M.K. Zuffo. On the usability of gesture interfaces in virtual reality environments. In *Proceedings of the Latin American Conference on Human-Computer Interaction*, 2005.

[24] C. Cadoz. *Les réalités virtuelles*. Flammarion, 1994.

[25] A. Cassidy, D. Hook, and A. Baliga. Hand tracking using spatial gesture modeling and visual feedback for a virtual DJ system. In *Proceedings of the 4th IEEE International Conference on Multimodal Interfaces*, 2002.

[26] C.C. Chang. Adaptive multiple sets of css features for hand posture recognition. *Neurocomputing*, 2006.

[27] C.C. Chang and C.Y. Liu. Modified curvature scale space feature alignment approach for hand posture recognition. In *Proceedings of the International Conference on Image Processing*, 2003.

[28] W.Y. Chang, C.S. Chen, and Y.P. Hung. Appearance-guided particle filtering for articulated hand tracking. In *Proceedings of the IEEE Conference on Computer Vision and PatternRecognition*, 2005.

[29] T. Chateau, A. Vacavant, and J.M. Lavest. Skin detection and tracking by monocular vision. In *Proceedings of the International Symposium on Signals, Circuits and Systems*, 2005.

[30] S. Chatty. Issues and experience in designing two-handed interaction. In *Proceedings of the Conference on Human Factors in Computing Systems*, 1994.

[31] F. Chen, E. Choi, J. Epps, S. Lichman, N. Ruiz, Y. SHi, R. Taib, and M. Wu. A study of manual gesture-based selection for the PEMMI multimodal transport management interface. In *Proceedings of the 7th International Conference on Multimodal Interfaces*, 2005.

[32] F.S. Chen, C.M. Fu, and C.L. Huang. Hand gesture recognition using a real-time tracking method and hidden markov models. *Image and Vision Computing*, 2003.

[33] X. Chen, H. Koike, Y. Nakanishi, K. Oka, and Y. Sato. Two-handed drawing on augmented desk system. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, 2002.

[34] K. Cheng and M. Takatsuka. Real-time monocular tracking of view frustum for large screen human-computer interaction. In *Proceedings of the 28th Australasian Conference on Computer Science*, 2005.

[35] T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active appearance models. *Lecture Notes in Computer Science*, 1998.

[36] C. Costanzo, G. Iannizzotto, and F. La Rosa. Virtualboard : Real-time visual gesture recognition for natural human-computer interaction. In *Proceeding of the International Symposium on Parallel and Distributed Processing*, 2003.

[37] J.L. Crowley and J. Martin. Visual processes for tracking and recognition of hand gestures. In *Proceedings of the International Workshop on Perceptual User Interfaces*, 1997.

[38] L.D. Cutler, B. Fröhlich, and P. Hanrahan. Two-handed direct manipulation on the responsive workbench. In *Proceedings of the Symposium on Interactive 3D Graphics*, 1997.

[39] J. Davis and M. Shah. Recognizing hand gestures. In *Proceedings of the European Conference on Computer Vision*, 1994.

[40] J. Davis and M. Shah. Towards 3-d gesture recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 1999.

[41] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum-likelihood from incomplete data via the em algorithm. *Journal of Royal Statistical Society*, 1977.

[42] J. Deng and H.T. Tsui. A pca/mda scheme for hand posture recognition. In *Proceeding of the fifth IEEE Int. Conference of Automatic Face and Gesture Recognition*, 2002.

[43] J.M.S. Dias, P. Nande, N. Barata, and A. Correia. O.G.R.E. - open gesture recognition engine. In *Proceedings of the 17th Brazilian Symposium on Computer Graphics and Image Processing*, 2004.

[44] S.M. Dominguez, T. Keaton, and A.H. Sayed. Robust finger tracking for wearable computer interfacing. In *Proceedings of the Workshop on Perceptive User Interfaces*, 2001.

[45] W. Du and H. Li. Vision based gesture recognition system with single camera. In *Proceedings of the 5th International Conference on Signal Processing*, 2000.

[46] M. Ehreumann, T. Lutticke, and R. Dillmann. Dynamic gestures as an input device for directing a mobile platform. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2001.

[47] H. Fillbrandt, S. Akyol, and K.F. Kraiss. Extraction of 3d hand shape and posture from image sequences for sign language recognition. In *Proceedings of the IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, 2003.

[48] F. Flóres, J.M. García, J. García, and A. Hernández. Hand gesture recognition following the dynamics of a topology-preserving network. In *Proceeding of the fifth IEEE Int. Conference of Automatic Face and Gesture Recognition*, 2002.

[49] B. Fröba and A. Ernst. Face detection with the modified census transform. In *Proceedings of the Automatic Face and Gesture Recognition Conference*, 2004.

[50] C. Graetzel, T.W. Fong, S. Grange, and C. Baur. A non-contact mouse for surgeon-computer interaction. *Technology and Health Care*, 2004.

[51] A. Gruenstein. Two methods of gesture recognition. http ://www.mit.edu/˜alexgru/vision/review.pdf, 2002.

[52] R. Grzeszcuk, G. Bradski, M.H. Chu, and J.Y. Bouguet. Stereo based gesture recognition invariant to 3d pose and lighting. In *Proceedings of the IEEE Confernece on Computer Vision and Pattern Recognition*, 2000.

[53] Y. Guiard. Asymmetric division of labor in human skilled bimanual action : the kinematic chain as a model. *Journal of Motor Behavior*, 1987.

[54] Y. Hamada, N. Shimada, and Y. Shirai booktitle. Hand shape estimation using image transition network. In *Proceedings of the Workshop on Human Motion*, 2000.

[55] Y. Hamada, N. Shimada, and Y. Shirai. Hand shape estimation using sequence of multi-ocular images based on transition metwork. In *Proceedings of the International Conference on Vision Interface*, 2002.

[56] Y. Hamada, N. Shimada, and Y. Shirai. Hand shape estimation under complex backgrounds for sign language recognition. In *Proceedings of the 6th International Conference on Automatic Face and Gesture Recognition*, 2004.

[57] P.R.G. Harding and T. Ellis. Recognizing hand gesture using fourier descriptors. In *Proceedings of the 17th International Conference on Pattern Recognition*, 2004.

[58] J.A. Hartigan and M.A. Wong. A k-means clustering algorithm. *Journal of Applied Statistics*, 1979.

[59] A.G. Hauptmann. Speech and gestures for graphic image manipulation. In *Proceedings of ACM Conference on Computer-Human Interaction*, 1989.

[60] A.J. Heap and F. Samaria. Real-time hand tracking and gesture recognition using smart snakes. In *Proceedings of Interface to Real and Virtual Worlds*, 1995.

[61] G. Heidemann, H. Bekel, I. Bax, and A. Saalbach. Hand gesture recognition : self-organising maps as a graphical user interface for the partitioning of large training data sets. In *Proceedings of the International Conference on Pattern Recognition*, 2004.

[62] G. Heidermann, I. Bax, and H. Bekel. Multimodal interaction in an augmented reality scenario. In *Proceedings of the 6th International Conference on Multimodal Interfaces*, 2004.

[63] T. Hermann, C. Nölker, and H. Ritter. Hand postures for sonification control. In *Proceedings of the International Gesture Workshop on Gesture and Sign Languages in Human-Computer Interaction*, 2001.

[64] K. Hinckley, R. Pausch, and D. Proffitt. Attention and visual feedback : the bimanual frame of reference. In *Proceedings of the Symposium on Interactive 3D Graphics*, 1997.

[65] K. Hoshino and T. Tanimoto. Realtime estimation of human hand posture for robot hand control. In *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation*, 2005.

[66] A.J. Howell and H. Buxton. Active vision techniques for visually mediated interaction. *Image and Vision Computing*, 2002.

[67] J. Howell and H. Buxton. Learning gestures for visually mediated interaction. In *Proceedings of the British Machine Vision Conference*, 1998.

[68] T.S. Huang, Y. Wu, and J. Lin. 3d model-based visual hand tracking. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, 2002.

[69] Y. Huang, T.S. Huang, and H. Niemann. Two-handed gesture tracking incorporating template warping with static segmentation. In *Proceeding of the fifth IEEE Int. Conference of Automatic Face and Gesture Recognition*, 2002.

[70] G. Iannizzotto, C. Costanzo, F. La Rosa, and P. Lanzafame. A multimodal perceptual user interface for video-surveillance environments. In *Proceedings of the 7th International Conference on Multimodal Interface*, 2005.

[71] G. Iannizzotto, M. Villari, and L. Vita. Hand tracking for human-computer interaction with graylevel visualglove : turning back to the simple way. In *Proceedings of the Workshop on Perceptive User Interfaces (PUI 2001)*, 2001.

[72] I. Infantino, A. Chella, H. Dindo, and I. Macaluso. A cognitive architecture for robotiv hand posture learning. *IEEE Transactions on Systems, Man and Cybernetics*, 2005.

[73] H. Jang, J.H. Do, and Z.Z. Bien. Two-staged hand-posture recognition method for softremocon system. In *Proceedings of the International Conference on Systems, Man and Cybernetics*, 2005.

[74] H. Jang, J.H. Do, J. Jung, K.H. Park, and Z.Z. Bien. View-invariant hand posture recognition for soft-remocon-system. In *Proceedings of the International Conference on Intelligent Robots and Systems*, 2004.

[75] M.H. Jeong, Y. Kuno, N. Shimada, and Y. Shirai. Recognition of shape-changing hand gestures based on switching linear model. In *Proceedings of the 11th International Conference on Image Analysis and Processing (ICIAP'01)*, 2001.

[76] M.H. Jeong, Y. Kuno, N. Shimada, and Y. Shirai. Two-hand gesture recognition using couples switching linear model. In *Proceedings of the 16th International Conference on Pattern Recognition*, 2002.

[77] H. Jin, Q. Liu, H. Lu, and X. Tong. Face detection using improved LBP under bayesian framework. In *Proceedings of the Third International Confernece on Image and Graphics (ICIG)*, 2004.

[78] M. Jones and P. Viola. Face recognition using boosted local features. In *Proceedings of the International Conference on Computer Vision*, 2003.

[79] C.F. Juang, K.C. Ku, and S.K. Chen. Temporal hand gesture recognition by fuzzified TSK-type recurrent fuzzy network. In *Proceedings of the IEEE International Joint Conference on Neural Networks*, 2005.

[80] A. Just, Y Rodriguez, and S. Marcel. Hand posture classification and recognition using the modified census transform. In *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, 2006.

[81] J. Juster and D. Roy. Elvis : situated speech and gesture understanding for a robotic chandelier. In *Proceedings of the 6th International Conference on Multimodal Interfaces*, 2004.

[82] P. Kabbash, W. Buxton, and A. Sellen. Two-handed input in a compound task. In *Proceedings of the Conference on Human Factors in Computing Systems*, 1994.

[83] H. Kang, C.W. Lee, and K. Jung. Recognition-based gesture spotting in video games. *Pattern Recognition Letters*, 2004.

[84] T. Keaton, M. Dominguez, and H. Sayed. Browsing the environment with the SNAP&TELL wearable computer system. *Personal and Ubiquitous Computing*, 2005.

[85] A. Kendon. *Crosscultural perspectives in nonverbal communication*, chapter How gestures can become like words. Toronto, Canada, 1988.

[86] C. Keskin, A. Erkan, and L. Akarun. Real time hand tracking and 3d gesture revognition for interactive interfaces using hmm. In *Proceedings of the Joint Conference ICANN-ICONIP*, 2003.

[87] H. Kim and D.W. Fellner. Interaction with hand gesture for a back-projection wall. In *Proceedings of the CComputer Graphics International*, 2004.

[88] T. Ko, D. Demirdjian, and T. Darrell. Untethered gesture acquisition and recognition for a multimodal conversational system. In *Proceedings of the 5th International Conference on Multimodal Interfaces*, 2003.

[89] H. Koike, Y. Sato, and Y. Kobayashi. Integrating paper and digital information on enhanceddesk : a method for realtime finger tracking on an augmented desk system. *ACM Transactions on Computer-Human Interaction*, 2001.

[90] M. Kölsch. *Vision-based hand gesture interfaces for wearable computing and virtual environments.* PhD thesis, University of California, 2004.

[91] M. Kölsch and M. Turk. Fast 2d hand tracking with flocks of features and multi-cue integration. In *Proceedings of the Conference on Computer Vision and Pattern Recognition Workshop*, 2004.

[92] M. Kölsch and M. Turk. Robust hand detection. In *Proceedings of the 6th International Conference on Automatic Face and Gesture Recognition*, 2004.

[93] N. Krahnstoever, M. Yeasin, and R. Sharma. Automatic acquisiton and initialization of kinematic models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

[94] T. Kurata, T. Okuma, M. Kourogi, and K. Sakaue. The handmouse : Gmm hand-color classification and mean shift tracking. In *Proceedings of the IEEE Internation Conference on Computer Vision - Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, 2001.

[95] J.J. La Viola. A survey of hand posture and gesture recognition techniques and technology. Technical report, Department of Computer Science, Brown University, 1999.

[96] I. Laptev and T. Lindeberg. Tracking of multi-state hand models using particle filtering and a hierarchy of multi-scale image features. Technical report, Department of Numerical Analysis and Computer Science, KTH, 2000.

[97] F. Lathuilière. Visual tracking of hand posture with occlusion handling. In *Proceedings of the International Conference on Pattern Recognition*, 2000.

[98] F. Lathuiliere and J.Y. Herve. Visual hand posture tracking in a gripper guiding application. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2000.

[99] H.K. Lee and J.H. Kim. An hmm-based threshold model approach for gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1999.

[100] L.K. Lee, S. Kim, Y.K. Choi, and M.H. Lee. Recognition of hand gesture to human-computer interaction. In *Proceedings of the 26th Annual Conference of the IEEE Industrial Electronics Society*, 2000.

[101] S.H. Lee, S.C. Ahn, and H.G. Kim. MAWPUC algorithm for tracking face and hands in complex background. In *Proceedings of the 2nd International Symposium on Image and Signal Processing and Analysis*, 2001.

[102] A. Leganchuk, S. Zhai, and W. Buxton. Manual an cognitive benefits of two-handed inputs : an experimental study. *ACM Transactions on Computer-Human Interaction*, 1998.

[103] J. Letessier and F. Bérard. Visual tracking of bare fingers for interactive surfaces. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*, 2004.

[104] J.T. Letessier. Suivi de doigts nus pour surfaces interactives en vision par ordinateur. Master's thesis, Institut National Polytechnique de Grenoble, 2003.

[105] G. Levin and Z. Lieberman. Sounds from shapes : audiovisual performance with hand silhouette contours in the manual input sessions. In *Proceedings of the Conference on New Interfaces for Musical Expression*, 2005.

[106] A. Licsar and T. Sziranyi. Dynamic training of hand gesture recognition system. In *Proceedings of the 17th International Conference on Pattern Recognition*, 2004.

[107] J.Y. Lin, Y. Wu, and T.S. Huang. 3d model-based hand tracking using stochastic direct search method. In *Proceedings of the 6th IEEE International Conference on Automatic Face and Gesture Recognition*, 2004.

[108] Y. Liu and Y. Jia. A robust hand tracking for gesture-based interaction of wearable computers. In *Proceedings of the 8th International Symposium on Wearable Computers*, 2004.

[109] Y. Liu, X. Liu, and Y. Jia. Hand-gesture based text input for wearable computers. In *Proceedings of the IEEE International Conference on Computer Vision Systems*, 2006.

[110] S. Lu, G. Huang, D. Samaras, and D. Metaxas. Model-based integration of visual cues for hand tracking. In *Proceedings of the Workshop on Motion and Video Computing*, 2002.

[111] J. MacLean, R. Herpers, C. Pantofaru, L. Wood, K. Derpanis, D. Topalovic, and J. Tsotsos. Fast hand gesture recognition for real-time teleconferencing applications. In *Proceedings of the IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems (RATFG-RTS'01)*, 2001.

[112] S. Malik and J. Laszlo. Visual touchpad : a two-handed gestural input device. In *Proceedings of the 6th International Conference on Multimodal Interfaces*, 2004.

[113] S. Malik, C. McDonald, and G. Roth. Hand tracking for interactive pattern-based augmented reality. In *Proceedings of the International Symposium on Mixed and Augmented Reality*, 2002.

[114] S. Malik, A. Ranjan, and R. Balakrishnan. Interacting with large displays from a distance with vision-tracked multi-finger gestural input. In *Proceedings of the 18th annual ACM Symposium on User Interface Software and Technology*, 2005.

[115] J.P. Mammen, S. Chaudhuri, and T. Agrawal. Simultaneous tracking of both hands by estimation of erroneous observations. In *Proceedings of the British Machine Vision Conference*, 2001.

[116] S. Marcel. *Une approche générative neuro-markovienne du traitement de séquences d'images : Application à la reconnaissance statique et dynamique des gestes de la main*. PhD thesis, Université de Rennes I, 2000.

[117] S. Marcel. Hand posture recognition in a body-face centered space. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI'99)*, 99.

[118] S. Marcel, O. Bernier, J.E. Viallet, and D. Collobert. Hand gesture recognition using inpu/output hidden markov models. In *Proceedings of the Conference on Automatic Face and Gesture Recognition*, 2000.

[119] G. McAllister, S.J. McKenna, and I.W. Ricketts. Tracking a driver's hands using computer vision. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, 2000.

[120] G. McAllister, S.J. McKenna, and I.W. Ricketts. Hand tracking for behaviour understanding. *Image and Vision Computing*, 2002.

[121] C. McDonald and G. Roth. Replacing a mouse with hand gesture in a plane-based augmented reality system. In *Proceedings of the 2nd IASTED International Conference on Computer Graphics and Imaging*, 2003.

[122] S. McKenna and K. Morrison. A comparison of skin history and trajectory-based representation schemes for the recognition of user-specific gestures. *Pattern Recognition*, 2004.

[123] S.J. McKenna and K. Morrison. A comparison of skin history and trajectory-based representation schemes for the recognition of user-specified gestures. *Pattern Recognition*, 2003.

[124] D. McNeill. *Hand and mind : what gestures reveal about thought*. University of Chicago Press, Chicago, 1992.

[125] M.R. Mine, F.P. Brooks, and C.H. Sequin. Moving objects in space : exploiting proprioception in virtual environments. 1997.

[126] Z. Mo, J.P. Lewis, and U. Neumann. Smartcanvas : a gesture-driven intelligent drawing desk system. In *Proceedings of the 10th International Conference on Intelligent User Interfaces*, 2005.

[127] P. Modler, T. Myatt, and M. Saup. An experimental set of hand gestures for expressive control of musical parameters in realtime. In *Proceedings of the Conference on New Interfaces for Musical Expression*, 2003.

[128] J.A.V. Montero and L.E.S. Sucar. Feature selection for visual gesture recognition using hidden markov models. In *Proceedings of the 9th Mexican International Conference on Computer Science*, 2004.

[129] S. Müller, S. Eickeler, and G. Rigoll. Crane gesture recognition using pseudo 3-d hidden markov models. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, 2000.

[130] J.R. New, E. Hasanbelliu, and M. Aguilar. Facilitating user interaction with complex systems via hand gesture recognition. In *Proceedings of the Southeastern ACM Conference*, 2003.

[131] C.W. Ng and S. Ranganath. Real-time gesture recognition system and application. *Image Vision and Computing*, 2002.

[132] K. Nickel and R. Stiefelhagen. Pointing gesture recognition based on 3d-tracking of face, hands and head orientation. In *Proceedings of the 5th International Conference on Multimodal Interfaces*, 2003.

[133] H. Nishino, K. Utsumiya, S. Kuraoka, and K. Yoshioka. Interactive two-handed gesture interface in 3d virtual environments. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, 1997.

[134] C. Nölker and H. Ritter. Parametrized SOMs for hand posture reconstruction. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN'00)*, 2000.

[135] C. Nölker and H. Ritter. Visual recognition of continuous hand postures. *IEEE Transcations on Neural Networks*, 2002.

[136] R. O'Hagan and A. Zelinsky. Visual gesture interfaces for virtual environments. In *Proceedings of the 1st Australasian User Interface Conference*, 2000.

[137] T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition*, 1996.

[138] S.C.W. Ong and S. Ranganath. Automatic sign language analysis : A survery and the future beyond lexical meaning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005.

[139] J.J. Pantrigo, A.S. Montemayor, and A. Sanchez. Local search particle filter applied to human-computer interaction. In *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis*, 2005.

[140] L.R. Rabiner and B.H. Juang. An introduction to hidden markov models. *IEEE ASSP Magazine*, 1986.

[141] R. Raisamo. *Multimodal human-computer interaction : a constructive and empirical study*. PhD thesis, Faculty of Economics and Administration of the University of Tampere, 1999.

[142] A. Ramamoorthy, N. Vaswani, S. Chaudhury, and S. Banerjee. Recognition of dynamic hand gestures. *Pattern Recognition*, 2003.

[143] B. Rime and L. Schiaratura. *Fundamentals of nonverbal behavior*, chapter Gesture and speech. Press Syndicate of the University of Cambridge, New-York, 1991.

[144] P. Robertson, R. Laddaga, and M. Van Kleek. Virtual mouse vision based interface. In *Proceedings of the 9th International Conference on Intelligent User Interface*, 2004.

[145] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning internal representations by back-propagating errors. *Nature*, 1986.

[146] Y. Sato, M. Saito, and H. Koike. Real-time input of 3d pose and gestures of a user's hand and its applications for hci. In *Proceedings of IEEE Virtual Reality Conference (VR'01)*, 2001.

[147] S. Sclaroff, M. Betke, G. Kollios, J. Alon, V. Athitsos, R. Li, J. MAgee, and T.P. Tian. Tracking, analysis, and recognition of human gestures in video. In *Proceedings of the 8th International Conference on Document Analysis and Recognition*, 2005.

[148] J. Segen and S. Kumar. Look ma, no mouse! *Communications of the ACM*, 2000.

[149] A. Sepheri, Y. Yacoob, and L.S. Davis. Parametric hand tracking for recognition of virtual drawings. In *Proceedings of the IEEE International Conference on Computer Vision Systems*, 2006.

[150] A. Shamaie and A. Sutherland. Graph-based matching of occluded hand gestures. In *Proceedings of the Applied Imagery Pattern Recognition Workshop*, 2001.

[151] A. Shamaie and A. Sutherland. Hand tracking in bimanual movements. *Image and Vision Computing*, 2005.

[152] C. Shan, Y. Wei, X. Qiu, and T. Tan. Gesture recognition using temporal template based trajectories. In *Proceedings of the 17th International Conference on Pattern Recognition*, 2004.

[153] C. Shan, Y. Wei, T. Tan, and F. Ojardias. Real time hand tracking by combining particle filtering and mean shift. In *Proceedings of the 6th IEEE International Conference on Automatic Face and Gesture Recognition*, 2004.

[154] R. Sharma, M. Zeller, V.I. Pavlovic, T.S. Huang, Z. Lo, S. Chu, Y. Zhao, J.C. Phillips, and K. Schulten. Speech/gesture interface to a visual-computing environment. *IEEE Computer Graphics and Applications*, 2000.

[155] M.C. Shin, L.V. Tsap, and D.B. Goldgof. Gesture recognition using bezier curves for visualization navigation from registered 3d data. *Pattern Recognition*, 2004.

[156] R. Smith, W. Piekarski, and G. Wigley. Hand tracking for low powered mobile ar user interfaces. In *Proceedings of the 6th Australasian Conference on User Interface*, 2005.

[157] G. Somers and R.N. Whyte. Hand posture matching for irish sign language interpretation. In *Proceedings of the 1st International Symposium on Information and Communication Technologies*, 2003.

[158] Y. Sribooruang, P. Kumhom, and K. Chamnongthai. Hand posture classification using wavelet moment invariant. In *Proceedings of the IEEE Symposium on Virtual Environments, Human-Computer Interfaces and Measurement Systems*, 2004.

[159] T. Starner. Visual recognition of american sign language using hidden markov models. Master's thesis, Massachusetts Institute of Technology, 1995.

[160] T. Starner and A. Pentland. Real-time americam sign language recognition from video using hidden markov models. Technical report, M.I.T. Media Laboratory Perceptual Computing Section, 1996.

[161] N. Stefanov, A. Galata, and R. Hubbold. Real-time hand tracking with variable-length markov models of behaviour. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[162] B. Stenger, P.R.S. Mendoça, and R. Cipolla. Model-based 3d tracking of an articulated hand. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

[163] H.I. Stern, J.P. Wachs, and Y. Edan. Optimal hand gesture vocabulary design using psycho-physiological and technical factors. In *Proc. of the IEEE International Conference on Automatic Face and Gesture Detection*, 2006.

[164] D. Stotts, J.McC. Smith, and K. Gyllstrom. Facespace : endo- and exo-spatial hypermedia in the transparent video facetop. In *Proceedings of the 15th ACM Conference on Hypertext and Hypermedia*, 2004.

[165] D.J. Sturman. *Whole-hand input*. PhD thesis, Massachusetts Institute of Technology, 1992.

[166] E.B. Sudderth, M.I. Mandel, W.T. Freeman, and A.S. Willsky. Visual hand tracking using nonparametric belief propagation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition Workshop*, 2004.

[167] L. Tarabella and G. Bertini. Giving expression to multimedia performance. In *Proceedings of the ACM Workshops on Multimedia*, 2000.

[168] T.C. Terrillon, T. Pilpre, Y. Niwa, and T. Yamamoto. Robust face detection and hand posture recognition in color images for human-machine interaction. In *Proceedings of the 16th International Conference on Pattern Recognition*, 2002.

[169] C. Tomasi, S. Petrov, and A. Sastry. 3d tracking = classification + interpolation. In *Proceedings of the International Conference on Computer Vision*, 2003.

[170] J. Triesch and C. von der Malsburg. Robust classification of hand postures against complex backgrounds. In *Proceedings of the 2nd International Conference on Automatic Face and Gesture Recognition*, 1996.

[171] J. Triesch and C. von der Malsburg. A system for person-independent hand posture recognition against complex backgrounds. *IEEE Transactions on PAttern Analysis and Machine Intelligence (PAMI)*, 2001.

[172] L.V. Tsap. Gesture-tracking in real time with dynamic regional range computation. *Real-Time Imaging*, 2002.

[173] A. Vacavant and T. Chateau. Realtime head and hands tracking by monocular vision. In *Proceedings of the IEEE International Conference on Image Processing*, 2005.

[174] C. von Hardenberg and F. Bérard. Bare-hand human-computer interaction. In *Proceedings of the Conference on Perceptual User Interfaces*, 2001.

[175] J. Wachs, U. Kartoun, H. Stern, and Y. Edan. Real-time hand gesture telerobotic system using fuzzy c-means clustering. In *Proceedings of the 5th Biannual World Automation Congress*, 2002.

[176] J. Wachs, H. Stern, and Y. Edan. Parameter search for an image processing fuzzy C-means hand gesture recognition system. In *Proceedings of the International Conference on Image Processing*, 2003.

[177] N.C. Wah and S. Ranganath. Real-time gesture recognition system and application. *Image and Vision Computing*, 2002.

[178] A. Wilson and E. Cutrell. Flowmouse : a computer vision-based pointing and gesture input device. In *Proceedings of Interact*, 2005.

[179] A. Wilson and N. Oliver. Gwindows : robust stereo vision fo rgesture-based control of windows. In *Proceedings of the International Conference on Multimodal Interfaces*, 2003.

[180] A.D. Wilson. Playanywhere : a compact interactive tabletop projection-vision system. In *Proceedings of the 18th Annucal ACM Symposium on User Interface Software and Technology*, 2005.

[181] Y. Wu, J.Y. Lin, and T.S. Huang. Capturing natural hand articulation. In *Proceedings of the 8th IEEE International Conference on Computer Vision*, 2001.

[182] J. Yang, Y. Xu, and C.S. Chen. Gesture interface : modeling and learning. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 1994.

[183] M.H. Yang, N. Ahuja, and M. Tabb. Extraction of 2d motion trajectories and its application to hand gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002.

[184] Y. Yao, M. Zhu, Y. Jiang, and G. Lu. A bare hand controlled AR map navigation system. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 2004.

[185] Y. Yao and M.L. Zhu. Hand tracking in time-varying illumination. In *Proceedings of the International Confernece on Machine Learning and Cybernetics*, 2004.

[186] H. Yoon, J. Soh, Y.J. Bae, and H.S. Yang. Hand gesture recognition using combined features of location, angle and velocity. *Pattern Recognition*, 2001.

[187] Y.H. Yoon and K.H. Jo. Hand shape recognition using moment invariant for the korean sign language recognition. In *Proceedings of the 7th Korea-Russia International Symposium on Science and Technology*, 2003.

[188] N. Yoshiike and Y. Takefuji. Object segmentation using maximum neural networks for the gesture recognition system. *Neurocomputing*, 2003.

[189] R. Zaritsky, N. Peterfreund, and N. Shimkin. Velocity-guided tracking of deformable contours in three dimensional space. *International Journal of Computer Vision*, 2003.

[190] Z. Zhang, Y. Wu, Y. Shan, and S. Shafer. Visual panel : virtual mouse, keyboard and 3d controller with an ordinary piece of paper. In *Proceedings of the Workshop on Perceptive User Interfaces*, 2001.

[191] H. Zhou and T. Huang. Okapi-chamfer matching for articulated object recognition. In *Proceedings of the 10th IEEE International Conference on Computer Vision*, 2005.

[192] H. Zhou and T.S. Huang. Tracking articulated hand motion with eigen dynamics analysis. In *Proceedings of the IEEE International Conference on Computer Vision*, 2003.

[193] H. Zhou, D.J. Lin, and T.S. Huang. Static hand gesture recognition based on local orientation histogram feature distribution model. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2004.

[194] Y. Zhu. Hand detection and tracking in an active vision system. Master's thesis, York University, 2003.

[195] Y. Zhu, H. Ren, G. Xu, and X. Lin. Toward real-time human-computer interaction with continuous dynamic hand gestures. In *Proceedings of the 4th IEEE International Conference on Automatic Face and Gesture Recognition*, 2000.

[196] Y. Zhu and G. Xu. A real-time approach to the spotting, representation, and recognition of hand gestures for human-computer interaction. *Computer Vision and Image Understanding*, 2002.

[197] M. Zobl, M. Geiger, B. Schuller, M. Lang, and G. Rigoll. A real-time system for hand gesture controlled operation of in-car devices. In *Proceedings of the International Conference on Multimedia and Expo*, 2003.