A comparative study of two state-of-the-art sequence processing techniques for hand gesture recognition

Agnès Just[a,b], Sébastien Marcel[a,b]

[a]*IDIAP Research Institute, 1920 Martigny, Switzerland*
[b]*Ecole Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland*

Abstract

In this paper, we address the problem of the recognition of isolated, complex, dynamic hand gestures. The goal of this paper is to provide an empirical comparison of two state-of-the-art techniques for temporal event modeling combined with specific features on two different databases. The models proposed are the Hidden Markov Model (HMM) and Input/Output Hidden Markov Model (IOHMM), implemented within the framework of an open source machine learning library (`www.torch.ch`). There are very few hand gesture databases available to the research community; consequently, most of the algorithms and features proposed for hand gesture recognition are not evaluated on common data. We thus propose to use two publicly available databases for our comparison of hand gesture recognition techniques. The first database contains both one- and two-handed gestures, and the second only two-handed gestures.

Key words: Human-computer interaction, Hand gesture recognition, Hidden Markov Models, Input/output HMM

1. Introduction

Since the early days of computers, Human-Computer Interaction (HCI) has changed a great deal. From the use of punched cards, the interaction has evolved and even surpassed the typewriter paradigm, through the use of keyboards and mice. These changes made the computer easier to interact with. Furthermore, computing is not longer constrained to the use of desktop computers and laptops. Some novel application environments include virtual and augmented reality, interaction with large displays, enhanced visualization of large data collections to name a few. In such cases, the keyboard and mouse pair can show its limits. From this perspective, the use of hand gestures for HCI can help people to communicate with computer-based systems in a more intuitive way. For instance, the potential power of gestures has already been demonstrated in applications that use hand gesture inputs to control a computer while giving a presentation [2]. Other possible applications of gesture recognition techniques include computer-controlled games, teleconferencing, robotics or the manipulation of objects by CAD designers. Furthermore, using two-handed inputs is of potential benefit to the user. In his thesis, Sturman [24] emphasizes that it is necessary to use the skills of the user. In [23], experiments on selection-positioning tasks as well as navigation-selection tasks show that users adopt strategies involving performing two sub-tasks simultaneously. These experiments also show that two-handed methods outperform the standard one-handed methods, and thus could be of potential benefit for HCI [25].

To interpret gestures, computers need to perceive the outside world. A common technique is to instrument the user's hand with special gloves equipped with sensors, or with the new remote controls developed by Nintendo for the Wii console.[1] Another possibility is to apply vision-based technologies. The use of video cameras is more natural than any dedicated acquisition device, but also more challenging. When dealing

[1] http://wii.nintendo.com

with gestural HCI, it is necessary to distinguish two aspects of hand gestures: the *static* aspect and the *dynamic* aspect. The *static* aspect is characterized by a pose or configuration of the hand in an image. The *dynamic* aspect is defined either by the trajectory of the hand, or by a series of hand postures in a sequence of images. Furthermore, there are two sub-problems to address when dealing with dynamic hand gesture recognition: spotting and classification. On one hand, spotting aims at identifying the beginning and/or the end of a gesture given a continuous stream of data. Usually, this stream of data is a random sequence containing both known and unknown gestures. On the other hand, given an isolated gesture sequence extracted from a continuous sequence of gestures, classification outputs the class the gesture belongs to.

In this paper, we will focus on the classification of dynamic hand gestures. In Section 2, we present an overview of related work in the field of vision-based Hand Gesture Recognition (HGR). In Section 3, we introduce Hidden Markov Models (HMMs) and Input/Output Hidden Markov Model (IOHMM). We then describe in Section 4 the two approaches used to capture one- and two-handed gestures, as well as the two different databases presented in this paper. In Section 5, we present experiment results and we finally conclude.

2. Related work

Hand Gesture Recognition (HGR) is a **sequence processing** problem that can be tackled using various techniques.

Finite State Machine (FSM) is one of the first techniques applied to HGR. Davis and Shah [3] decomposed gestures into four distinct phases. Since the four phases occurred in fixed order, a FSM was used to guide the stream and recognize seven gestures. Hong et al. [4] proposed another approach based on FSM that used 2D positions of the centers of the user's head and hands as features. Their system recognized in real-time four one-handed gestures. Arguably the most important technique, widely used for HGR, is the HMM. This approach is inspired by the success of the application of HMMs both in speech recognition and in hand-written character recognition fields [5]. Starner and Pentland [6] used an eight element feature vector consisting of each hand's x- and y- position, angle of axis of least inertia, and eccentricity of bounding ellipse. They used networks of HMMs to recognize a sequence of gestures taken from American Sign Language. With a lexicon of 40 words, they obtained 92% of accuracy in the test phase. Chan Wah and Ranganath [8] proposed a vision-based system which was able to interpret user's gestures for the manipulation of windows and objects within a graphical user interface. For that purpose, they compared HMMs with recurrent neural networks once again demonstrating the efficacy of the HMMs. Another interesting approach is taken by Lee and Kim [2]. They introduce the concept of a threshold model applied to HMMs to handle non-gesture patterns. Marcel et al. [18] have proposed to use an extension of HMMs, namely the IOHMM, for hand gesture recognition. An IOHMM is based on a non-homogeneous Markov chain where emission and transition probabilities depend on the input. In contrast the HMM is based on homogeneous Markov chains since the dynamics of the system are determined only by the transition probabilities, which are time independent. Their system was able to recognize four types of gestures with 98% of accuracy.

In the field of sign language recognition, reference methods such as neural networks, Hidden Markov Models and their variants [9] have been used for dynamic sign recognition. Another main category of techniques is based on the linguistic approach. This approach makes the assumption that signs can be broken into smaller parts called phonemes, which have an analogy with spoken 'phonemes'. Vogler and Metaxas [10] uses parallel Hidden Markov Models to model these phonemes. Derpanis [11] proposes to use a two level decomposition. Movements are decomposed into their static (kinematic features) and dynamic (lexical level) components. Other variants and extensions based on this approach have also been proposed by Wong and Cipolla [12] and Bowden et al. [13]. The gestures studied in this paper can be defined as interactive/manipulative gestures (for example, pointing at an object) and thus do not belong to the sign language. As they do not convey any linguistic information, comparing techniques specifically designed for sign language recognition on these data are not possible. We thus propose to provide results on two methods proposed for the more general problem of HGR.

As the above brief overview demonstrates, several methods have been developed for hand gesture recognition leading to good performances on very different databases. Unfortunately, most of the databases used to evaluate these methods are not publicly available. Furthermore, no benchmark or common database, as well as experimental protocol exist, making comparison of these techniques impossible. In this paper, we propose to evaluate two models combined with specific features on the same databases. We will compare the results obtained with a well-known state-of-the-art model, namely the Hidden Markov Model with a more complex model recently applied to gesture recognition, the Input/Output Hidden Markov Model. Let x be an observation and C_k the class the gesture belongs to ($k \in [1, \dots, N]$ where N is the number of classes). When using standard HMMs to model several gesture classes, a separate HMM is built for each gesture class C_k, trained on the data of that particular class. This approach is generative [28] as class conditional densities $p(x, C_k)$ and priors $p(C_k)$ are separately modeled. For recognition, posterior probabilities $p(C_k|x)$ are evaluated using Bayes' theorem. Whereas each HMM tries to best model the observations for a given gesture class, the IOHMM learns to map input sequences, the observations, to output sequences, the gesture class, for all observations of all gesture classes, using a supervised discriminant learning. Compared to HMMs, IOHMM is a discriminative approach as it directly models posterior probabilities. Discriminative approaches have often been preferred over generative approaches as they focus on the core problem of recognition, namely modeling $p(C_k|x)$. IOHMM should thus be giving good recognition results. Furthermore, IOHMM has already been applied to HGR with good results, but only on a small database of two gesture classes. We thus propose here to extend the study of IOHMM to the recognition of one- and two-handed gestures. We also provide public access to the databases (containing raw data) used in these experiments and to the source code of the algorithms.

3. Hidden Markov Models and Input/Output Hidden Markov Models

3.1. Hidden Markov Models

3.1.1. Statistical model

A HMM [7] may be used to model sequences of data. It consists of a set of N states, so-called hidden states because they are non-observable, 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 emissions from these states. Let $x_{1\dots T} = \{x_1, \dots, x_t, \dots, x_T\}$ be an output sequence, where $T \in \mathbb{N}$ is the length of the observation sequence, and let $q_t \in \{1, \dots, N\}$ be the state at time t. The emission probability $p(x_t|q_t = i), \forall i \in \{1, \dots, N\}$ at time t depends only on the current state q_t. The transition probability between states $p(q_t = i|q_{t-1} = j), \forall (i, j) \in \{1, \dots, N\}^2$ depends only on the previous state.

When dealing with continuous data, the set of emission probabilities are represented by a mixture of Gaussians. If we observe, at time t, the observation $x_t = X$ in state $q_t = j$, then the emission probability to observe X in state j, modeled by a mixture of K Gaussians, is given by

$$p(x_t = X|q_t = j) = \sum_{k=1}^{K} \alpha_k \mathcal{N}(X; \mu_{jk}, \sigma_{jk})$$

where $\alpha_k > 0, \forall k \in [1, \dots, K]$ with $\sum_{k=1}^{K} \alpha_k = 1$, and

$$\mathcal{N}(X; \mu_{jk}, \sigma_{jk}) = \frac{1}{\sqrt{(2\pi)^d |\sigma_{jk}|}} \exp(-\frac{1}{2}(X - \mu_{jk})^T \sigma_{jk}^{-1} (X - \mu_{jk}))$$

it the probability density function of the kth Gaussian in state j, for data in d dimensions.

If a quantization process is applied to the data, codebooks can be used to model these probabilities.

To efficiently use HMMs, it is necessary to impose a topology on the state graph. This topology limits the number of free parameters and allows us to inject in the model some *a priori* knowledge on the nature of the data. Figure 1 represents the state graph of an HMM with a left-right topology, meaning that no transitions from right to left (i.e. backwards in time) are allowed.

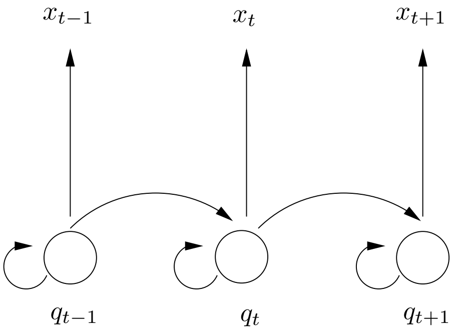


Figure 1: An HMM with a left-right topology showing dependencies between the observations x and the hidden states q

3.1.2. Training

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

$$L = p(\mathcal{X}|\Lambda) = \prod_{k=1}^{K} p(x_{1\dots T_k}|\Lambda)$$

where Λ represents the set of parameters of the model, K is the total number of observation sequences in $\mathcal{X}$ and $x_{1\dots T_k} \in \mathcal{X}$ is the kth observation sequence of length T_k. The *Expectation-Maximization* algorithm [14] is typically used to maximize this likelihood by adjusting the parameters of the model. More details can be found in [14, 7].

3.1.3. 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 $c \in [1 \dots C]$. The classification is performed using Bayes' rule, assuming equal prior probabilities for each gesture class. In the recognition phase, the gesture is classifies as belonging to class $\hat{c} = \arg\max_{c \in \{1,\dots,C\}} P(x_{1\dots T}|\Lambda_c)$.

3.2. Input/Output Hidden Markov Models

3.2.1. Statistical model

An IOHMM is an extension of the HMM described previously. First introduced by Bengio and Frasconi [15], IOHMMs are able to discriminate temporal sequences using a supervised discriminative training algorithm by mapping an input sequence to an output sequence. In our case, input sequences correspond to the observations and output sequences correspond to the class of the gesture.

Let $x_{1\dots T} = \{x_1, \dots, x_t, \dots, x_T\}$ be an input sequence, namely the observations (which corresponds to the output sequence of the HMM), and $y_{1\dots T} = \{y_1, \dots, y_t, \dots, y_T\}$ be an output sequence, where $y_t \in \{1, \dots C\}$ is the index of the gesture class at each time step. C is the number of gesture classes and $T \in \mathbb{N}$ is the length of input/output sequences. The architecture of an IOHMM also consists of a set of N states so let $q_t \in \{1, \dots, N\}$ 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 $p(y_t|q_t = i, x_t), \forall i \in \{1, \dots, N\}$ at time t depends on the current state q_t, but also depends on x_t. The transition probability between states $p(q_t = j|q_{t-1} = i, x_t), \forall (i,j) \in \{1, \dots N\}^2$ depends on the previous state and also on x_t (Figure 2).

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.

Similarly to HMMs, an IOHMM is able to model discrete or continuous data sequences. However, the main difference with HMMs is that emission and transition probabilities in an IOHMM also depend on the

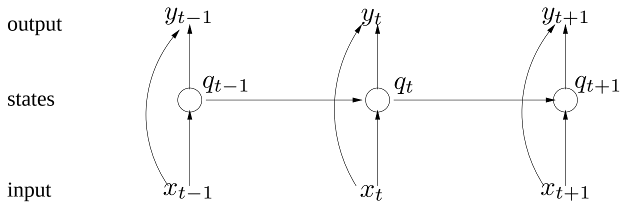


Figure 2: An IOHMM showing dependencies between the input x, output y and hidden states q of the model.

input sequence x_t. When the input data sequences x_t is continuous, modeling of conditional distributions in the IOHMM using Gaussian Mixture Models (GMMs) is difficult. To alleviate this limitation a more complex approach can be undertaken using, for example, a Multi-Layer Perceptron (MLP) [16], or the input data sequence can be discretized using, for example, Vector Quantization (VQ). The latter approach to discretize the input data x_t has been undertaken in this work (Section 3.3).

3.2.2. 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

$$L = p(\mathcal{Y}|\mathcal{X}, \Lambda) = \prod_{k=1}^{K} p(y_{1\ldots T_k}|x_{1\ldots T_k}, \Lambda)$$

where Λ represents the set of parameters of the model, K is the length of the input/output sequence, $x_{1\ldots T_k} \in \mathcal{X}$ is the kth input sequence and $y_{1\ldots T_k} \in \mathcal{Y}$ is the kth output sequence of length T_k. The *Expectation-Maximization* algorithm [14] is also used to maximize this likelihood and to adjust the parameters of the model. More details can be found in [17, 18].

3.2.3. Recognition

Let $x_{1\ldots T}$ be an input sequence to recognize. The test sequence is assigned to the class with the highest conditional probability on each frame such that

$$\hat{c} = \underset{c\in\{1,\ldots,C\}}{\text{argmax}}\ P(y_1 = c, \ldots, y_T = c|x_{1\ldots T}).$$

This recognition method is consistent with previous work on HGR using IOHMM [18].

3.3. Preprocessing the data

To use continuous HMMs, no preprocessing on the feature vectors is needed, but to efficiently use discrete IOHMM, a quantization step on the data are needed. The output sequence still encodes the gesture classes: $y_t = \{0, \ldots, 15\}, \forall t$ for the first database, and $y_t = \{0, \ldots, 6\}, \forall t$ for the second. To model more closely the class distribution of the data, we apply a K-means algorithm [21] class per class on the input features.

In case of the first database, for each gesture class, a K-means codebook with 75 clusters has been trained using the training set. The 16 resulting K-means codebooks are merged into a single codebook 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.

For the second database, the number of clusters has been selected on a class-by-class basis. The seven resulting K-means codebooks have also been merged into a single one codebook 565 clusters. In the same way as the previous database, sequences have been quantized into discrete values.

To study the effect of the quantization process, experiments were also conducted using discrete HMMs. In that case, data used correspond to the quantized data used with the IOHMM, and described above.

4. Databases

In this section, we describe the two databases used for experiments. In both cases, simple cameras have been used to record the data. These databases are available for download.[2] During the recording process, gesturers wear colored gloves of two different colors to facilitate the hand tracking and simplify hand recovery during and after occlusions, in the case of two-handed gestures. In practice, tracking can be carried out by various other means, such as particle filtering [26]. In this paper, the research focuses on the hand gesture recognition task, and does not attempt to deal with the tracking task.

4.1. InteractPlay database

4.1.1. Description

The database consists of 16 gestures (Table 1) that can be classified into four main categories:

- communication gestures: “Stop/Yes”, “No/Wipe”, “Raise hand” and “Hello/Wave”,
- direction gestures: “Left”, “Right”, “Up”, “Down”, “Front” and “Back”,
- pointing gestures: “Point left”, “Point front” and “Point right”,
- “fun” gestures: “Swim”, “Fly” and “Clap”

The first three main categories are interacting gestures. They are mainly designed for the purpose of collaborative working. Such environments require enhancing the workers’ capabilities. The last category corresponds to gestures that can be used for gaming. Vision-based gaming interfaces are already a reality, well illustrated by commercial products such as Eye Toy.[3]

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

Table 1: Description of the 16 gestures. A hand gesture may involve one hand (**M**ono-manual) or both hands (**B**i-manual). The gesture could be also a **R**epetitive movement such as *clap*.

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

[3] http://www.eyetoy.com

These gestures are carried out by 20 different people. The majority of gestures are one-handed and some are two-handed (*fly, swim* and *clap*). For each person and each gesture, five sessions and 10 shots per session have been recorded. Before recording, each gesturer was asked to place his/her feet in a predefined position (marks on the ground) to ensure rotational invariance of the data. The plane defined by the axes X and Z shown on the bottom of Figure 4 is parallel to the coronal plane of the person. This permits to ensure rotational invariance of the data. Furthermore, 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.

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. The extraction process is explained in section 4.1.2. To have unbiased data, gesturers were given a verbal description of the gestures followed by a short practical example. During recording, gestures were freely executed, without any correction on the way they were performed. Furthermore, to increase the naturalness of gestures, they were produced with the natural hand (left hand for left-handed and right hand for right-handed people). For the left-handed people, trajectories have been mirrored. Altogether, the database is composed of 1000 trajectories per gesture.

Figure 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). This sequence gives the maximum arm spread. Figure 4 also presents in a three dimensional space the coordinates of the center of each blob (head, torso and hands) for a “fly” gesture sequence.

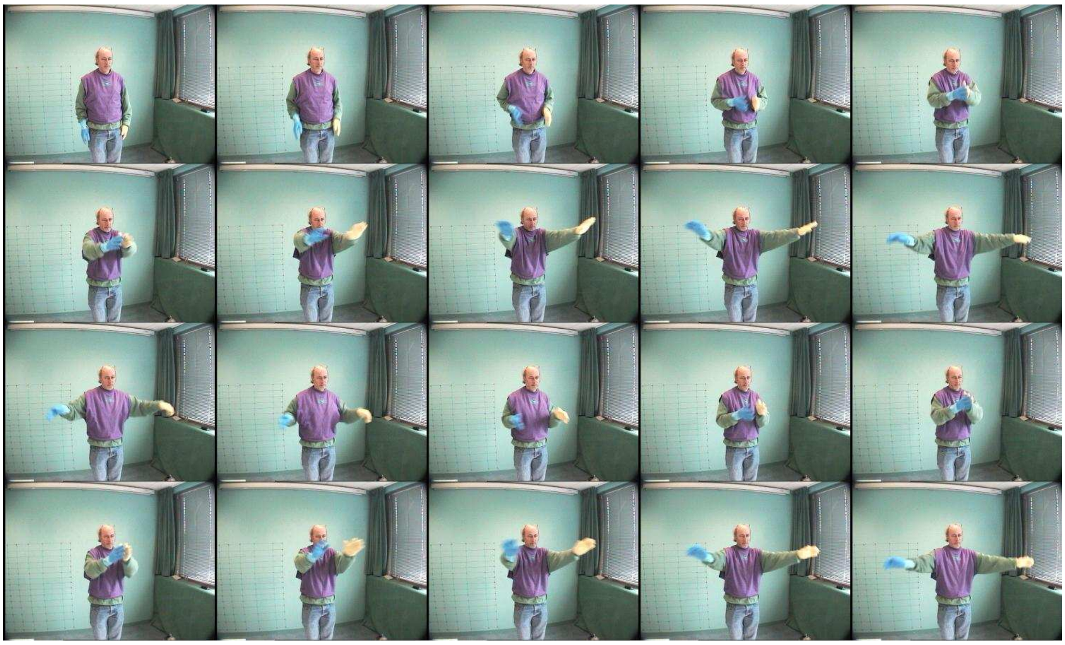


Figure 3: From top-left to bottom-right, a frame-by-frame decomposition of a “swim” gesture from the point of view of the right camera.

4.1.2. Image processing

Hand gestures are represented here by 3D trajectories of blobs. Blobs are obtained by tracking colored body parts in real-time using the EM algorithm. A detailed description of the 3D blob tracking can be found in [22]. Two cameras (Figure 5) are used to track head and hands in near real-time (12Hz).

Firstly, the preprocessing consists of background subtraction followed by specific color detection using a simple color lookup 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

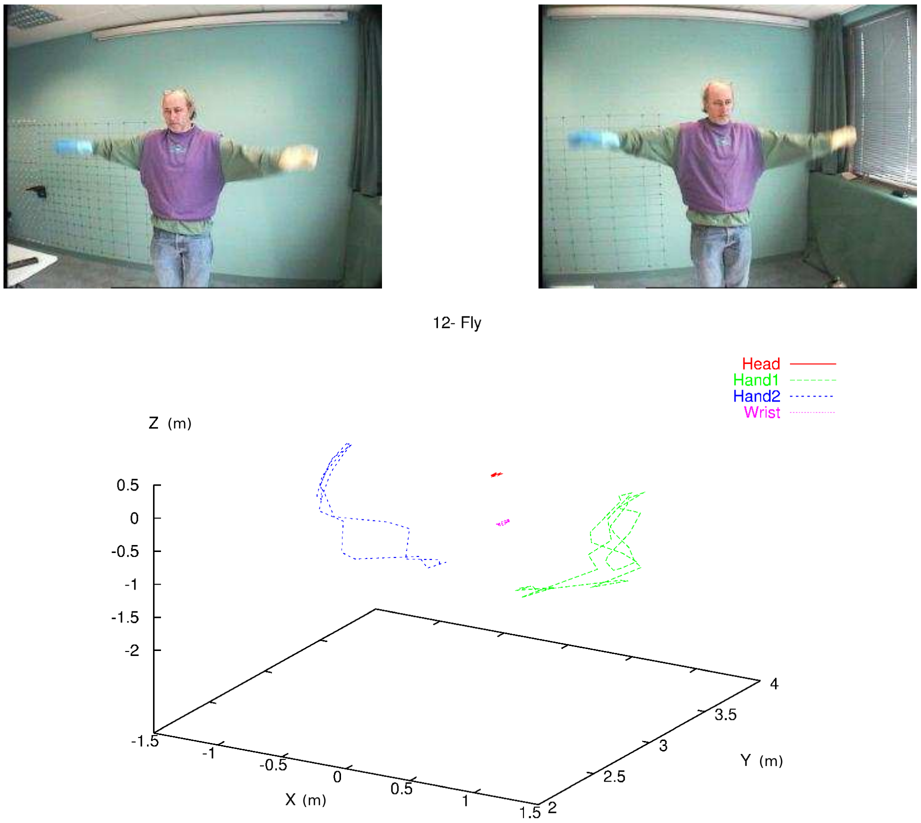


Figure 4: Top: Example of images of the "Vinci" sequence from the point of view of the left camera (on the left) and from the point of view of the right camera (on the right). Bottom: 3D coordinates of the center of each blob (head, torso, left hand and right hand) for a "fly" gesture. The z axis is the vertical axis of the person.

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 using the EM algorithm.

The use of gloves during the recording permits to ease the correspondence problem during the segmentation of the hands, which can be a major problem with two-handed gestures. In that case, hands are often occluding one another. To simplify the segmentation and tracking of the torso, the person performing the gesture also wears a sweat-shirt of specific color. Thus, tracking of the torso is transformed into a simple color blob tracking.

Finally, the database consists of 3D trajectories of the hands, head and torso in an unnormalized space. Additional recordings are the "Vinci" sequences. Because of the bulky size of the data (uncompressed video files), images have not been kept. Only the coordinates files of each region of interest is available.

4.1.3. Feature extraction

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 with vertex size given by the maximum arm spread obtained from the "Vinci" sequence. This cube is then normalized to reduce the vertex to

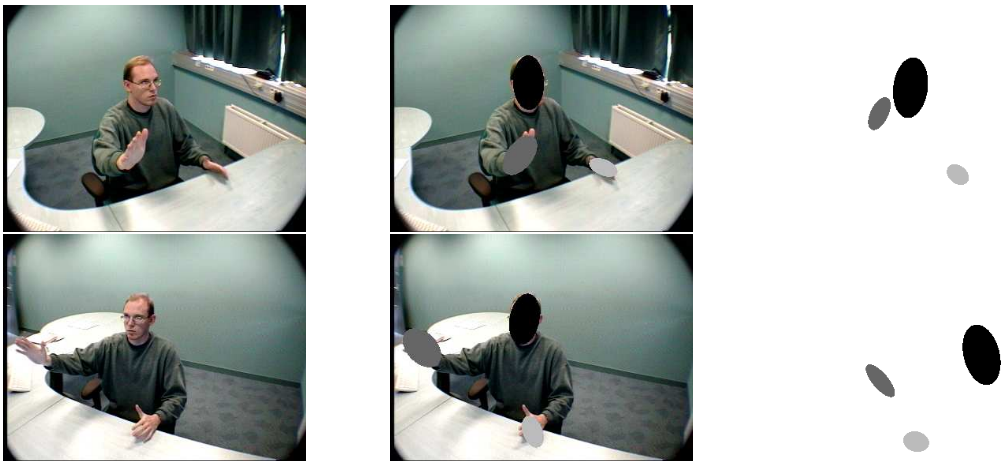


Figure 5: Left: left and right captured images. Center: left and right images with projected ellipsoids. Top right: ellipsoids projection on the frontal plane. Down right: ellipsoids projection on the sagittal plane (the cameras are on the left side).

1. The range of $x-, y-$ and z-coordinates thus varies between -0.5 and 0.5.
   One can notice that the 3D-coordinates of the head and torso are almost stationary (Figure 4). Thus, as a preprocessing step, we only keep the normalized 3D-trajectories of both hands. This leads to an input feature vector of size 6.
2. *Feature extraction:* We also compute the difference between the coordinates of each hand between two consecutive frames. These features have been multiplied by 100 to have values within the same order of magnitude as x, y and z. These features convey information about hand displacements during a sampling interval. Their resolution depends on the sampling frequency. These delta features have been used to improve the model. Recent papers in the field of speech recognition and speech synthesis [19, 20] have shown that including static and dynamic feature parameters in the HMM can improve the model. The generative model implied by the use of these features is different from the one obtained using no delta features, leading to better performances.

The final vector is then $[x_l, y_l, z_l, x_r, y_r, z_r, \Delta x_l, \Delta y_l, \Delta z_l, \Delta x_r, \Delta y_r, \Delta z_r] \in \mathbb{R}^{12}$.

It can be noticed that the normalization step is user-dependent. For each new user, a “Vinci” sequence will be recorded. This stage permits us to extract features that are somewhat user-independent for training and testing. Instead of training one model for each user, we only need to train a general model and apply it to the normalized data.

4.2. TwoHandManip database

4.2.1. Description

This second database is a set of seven two-handed manipulation gestures. The goal of these gestures is to manipulate objects on the screen the same way we interact with objects in real life. The gesturer holds an imaginary cube and rotates it along the three main axes and in the two directions or pushes it in front of her/him (Figure 6).

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 7 shows the set-up used to record the data. The acquisition is synchronized at 12 frames per second.

For each gesture, each camera records a different view. Each gesturer wears two colored gloves: one blue and the other yellow (Figure 8) to facilitate hand tracking.

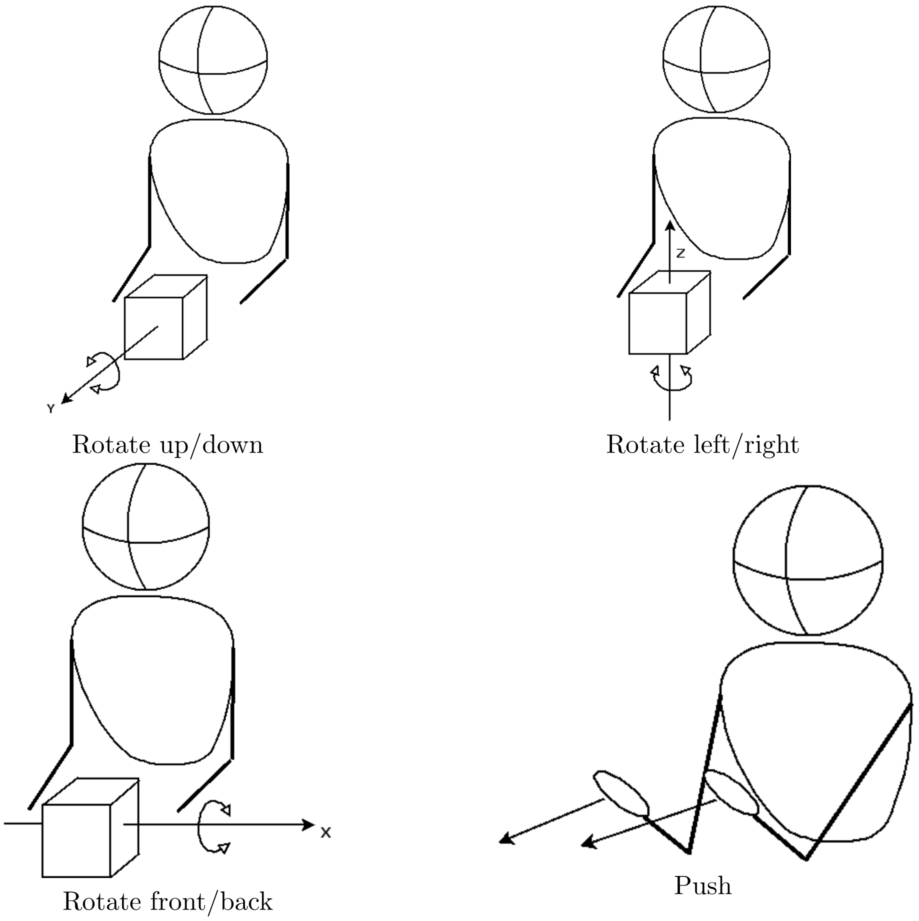


Figure 6: Description of the 6 rotation gestures and the “push” gesture contained in the TwoHandManip database.

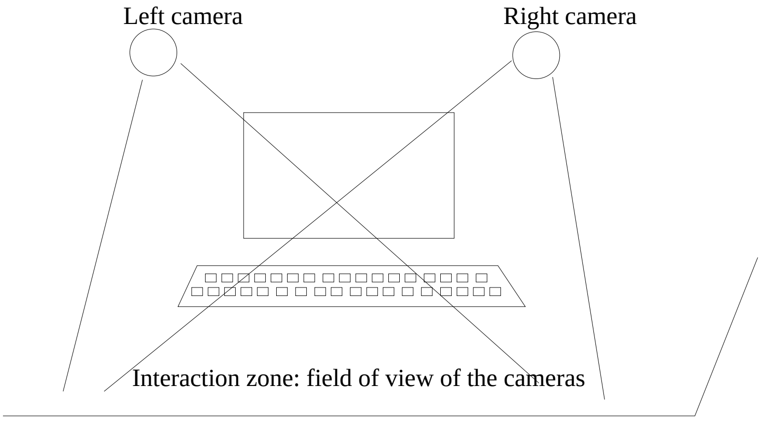


Figure 7: Camera set-up for recording the two-handed database

Seven people performed the gestures, with two sessions and five 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 s.

To avoid biased data, gesturers were only instructed verbally. No visual example was shown before or

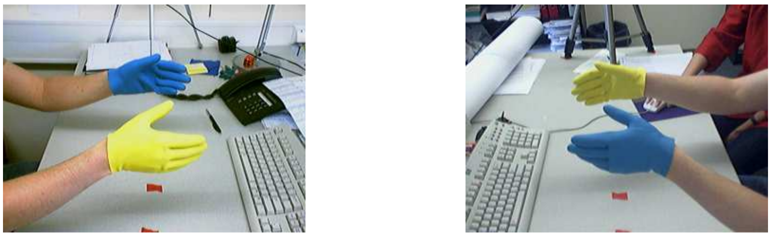


Figure 8: Point of view from the right and left cameras.

even during recording sessions. The gestures performed are thus very natural and personal to each gesturer. Even if the database is quite small, it covers a wide variability within each gesture class.

4.2.2. Feature extraction

Similarly to the processing performed on the InteractPlay database, hands are modeled as blobs and tracked using color (Figure 9). Whereas tracking was performed in 3D using images of both cameras (Section 4.1.2), here we perform tracking separately for each image. Each hand is thus approximated as a blob in 2D space for the right and left camera, respectively. The shape of each blob is also used as features.

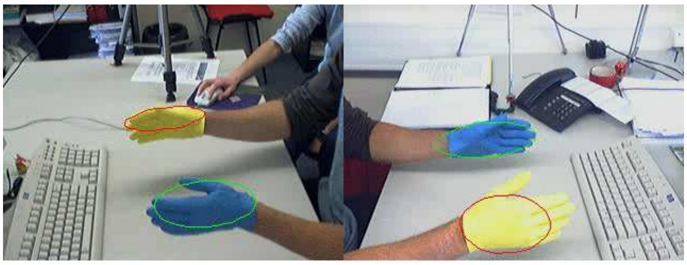


Figure 9: Representation of each hand blob for the left and right cameras

The mean (μ_x, μ_y) of each blob, namely the center of the boundary ellipse, is used as a feature. Let a be the half major axis and b the half minor axis of the ellipse. We can compute the eccentricity of the ellipse using the formula $e = \sqrt{1 - \frac{b^2}{a^2}}$ and the surface of the ellipse $s = \pi ab$. We also compute the angle α between the major axis of the ellipse and the horizon (Figure 10). These features are very similar to the ones used by Starner and Pentland [6]. The difference lies in the fact that we calculate also the surface of the ellipse for each hand in each camera.

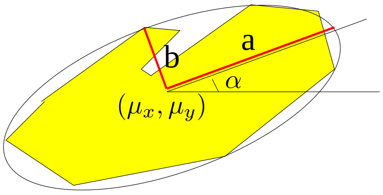


Figure 10: Features extracted

In some gestures, it happens that a hand is occluded by the other. In such a case, the features cannot be computed. Therefore, the features corresponding to the occluded hand in the previous image are used until new features can be computed.

The feature vector X corresponds then to the x- and y-coordinates of the center of each blob from the right and left images, 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. Experimental results

The open source machine learning library used for all experiments is Torch (`http://www.torch.ch`).

5.1. Parameter tuning

In our experiments, we have used both left/right and fully connected topologies for the HMMs and the IOHMM. To find the optimal hyper-parameters (number of states of the discrete IOHMM, number of states and number of Gaussians for the continuous 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. Table 2 gives the number of sequences for each subset and each database.

	Train	Valid	Test
InteractPlay	4000	4000	8000
TwoHandManip	142	140	210

Table 2: Statistics on the two databases

Models with different possibilities for the hyper-parameters (number of states for the IOHMM and HMMs, number of components in the GMM for the continuous HMMs) have been trained on T. The selection of the best hyper-parameters has been based on the results obtained with 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 3).

	InteractPlay Database		TwoHandManip Database	
	HMMs	IOHMM	HMMs	IOHMM
Topology	left-right	left-right	left-right	full connect
Number of clusters (quantization)	×	1200	×	565
Number of Gaussians per state	1	×	1	×
Number of states	13	3	12	17

Table 3: Hyper-parameters for the 16-gesture and manipulative databases respectively

Figure 11 shows the evolution of the error of the HMMs on the validation set V for the two databases over the space of evaluated hyper-parameters. We can note that the surface representing the error is smoother for the first database due to there being more data.

In the same way, Figure 12 shows the evolution of the error as a function of the number of states for the IOHMM. For both databases, we can notice that the error is not very stable.

Results obtained with the IOHMM are highly sensitive to the choice of the parameters. On the contrary, results obtained with the HMMs are not overly sensitive to the choice of parameters.

5.2. Results

5.2.1. InteractPlay database

Figure 13 provides comparative results on the test set Te between discrete IOHMM and baseline continuous HMMs on the first database.

Continuous HMMs and IOHMM achieve, respectively, 75% and 63% average recognition rate. Discrete HMMs achieve more than 73% average recognition rate, which is comparable with the results obtained with continuous HMMs. If we analyze more deeply the results obtained on the first database (Figure 14), we observe that two-handed gestures are very well classified. Few mistakes occur between "swim" and "clap" gestures.

Concerning one-handed gestures, a misclassification between the "stop", "no/wipe", "raise" and "hello/wave" gestures occurs. According to Table 1, the only differences between these four gestures are the hand level

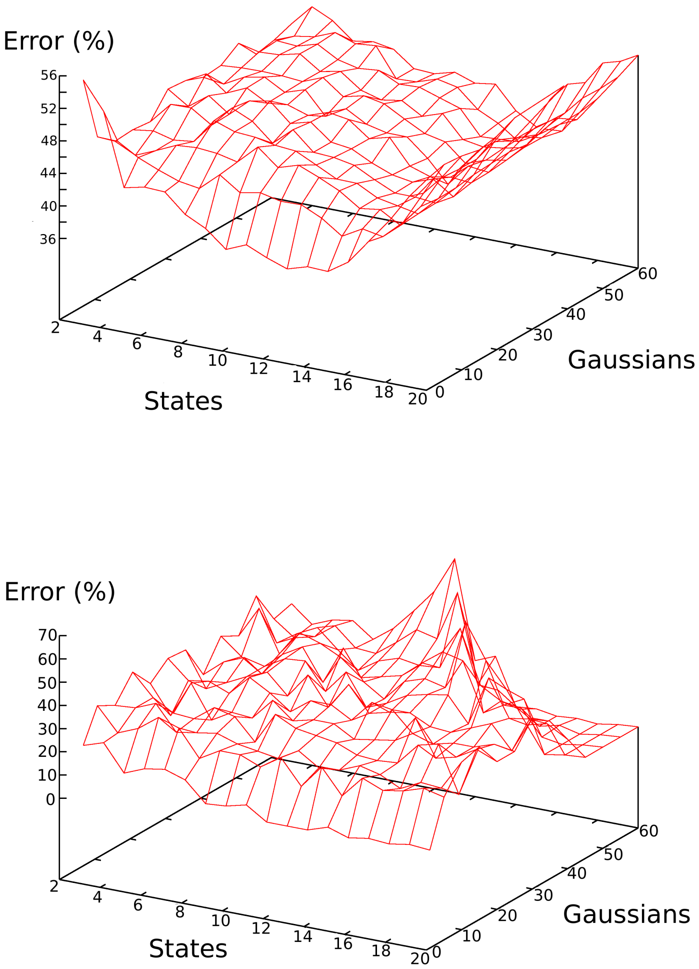


Figure 11: Error of the HMMs on the validation set as a function of the number of states (from 3 to 20) and number of Gaussians (from 1 to 60) for the InteractPlay database (top) and for the TwoHandManip database (bottom).

and the oscillatory movement of the hand from the left to the right. HMMs model quite well the oscillatory movement of these gestures (average recognition rate: 85%). Features extracted from these hand gesture 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 is less defined. The "no" and "hello" gestures are misclassified with each 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 is not sufficient 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 gestures from the others, but has difficulties to provide correct recognition within this category of gestures. These hand gestures are mostly misclassified by pairs, according to the

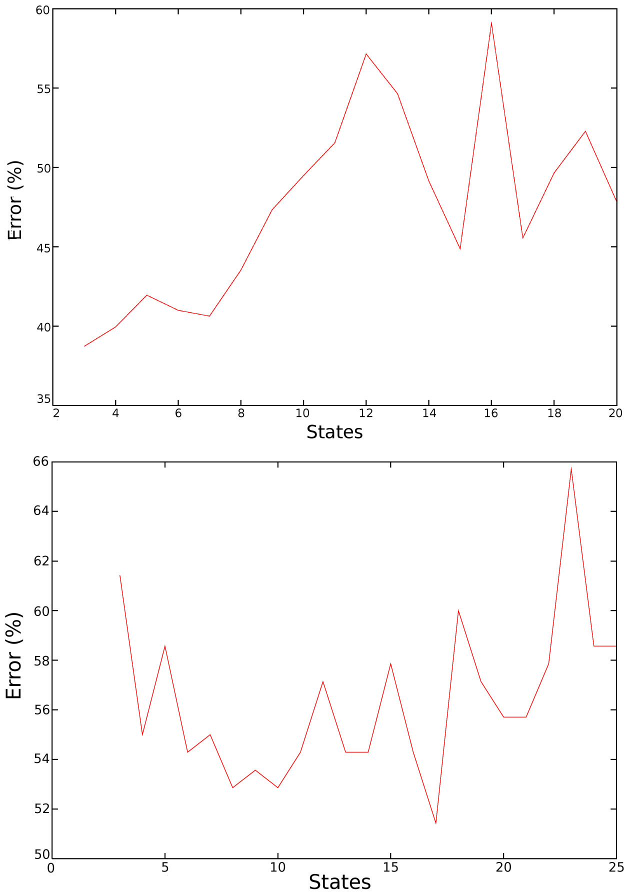


Figure 12: Error of the IOHMM on the validation set as a function of the number of states (from 3 to 20) for the InteractPlay database (left) and for the TwoHandManip database (right).

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. This can be explained by the dominance of right-handed people in the dataset, leading to a larger amplitude of these gestures, hence providing a better classification of these two gestures.

For pointing gestures, HMMs and IOHMM differentiate also quite accurately this category of hand gestures. 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 gestures with the positioning gestures. The location of the hand at the end of the pointing gesture is not precise enough to give a discriminant information to the IOHMM and to the HMMs. Furthermore, even if the IOHMM is sufficiently modeling oscillatory hand gestures, it has difficulties to differentiate non-oscillatory movements from oscillatory ones.

5.2.2. TwoHandManip database

Figure 15 shows results obtained with the two algorithms on the second database described in the previous section. For more details, the confusion matrix is also provided (Figure 16).

Continuous HMMs achieve 98% average recognition rate, while results with IOHMM are very poor, only achieving 43% average recognition rate. Discrete HMMs perform better than the IOHMM, achieving 66% average recognition rate, but less than continuous HMMs. Furthermore, even with few gesture examples, continuous HMMs perform very well on the test set.

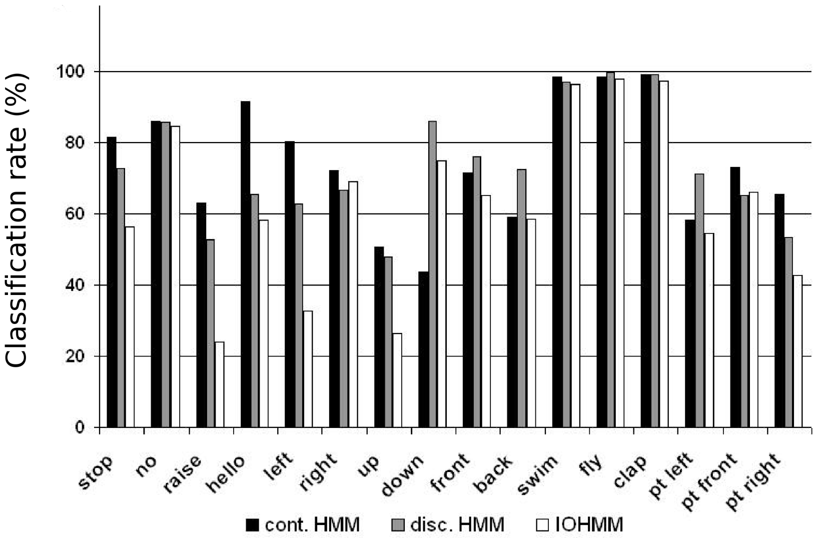


Figure 13: Classification rate (%) of the HMMs and IOHMM on the test set for the InteractPlay database

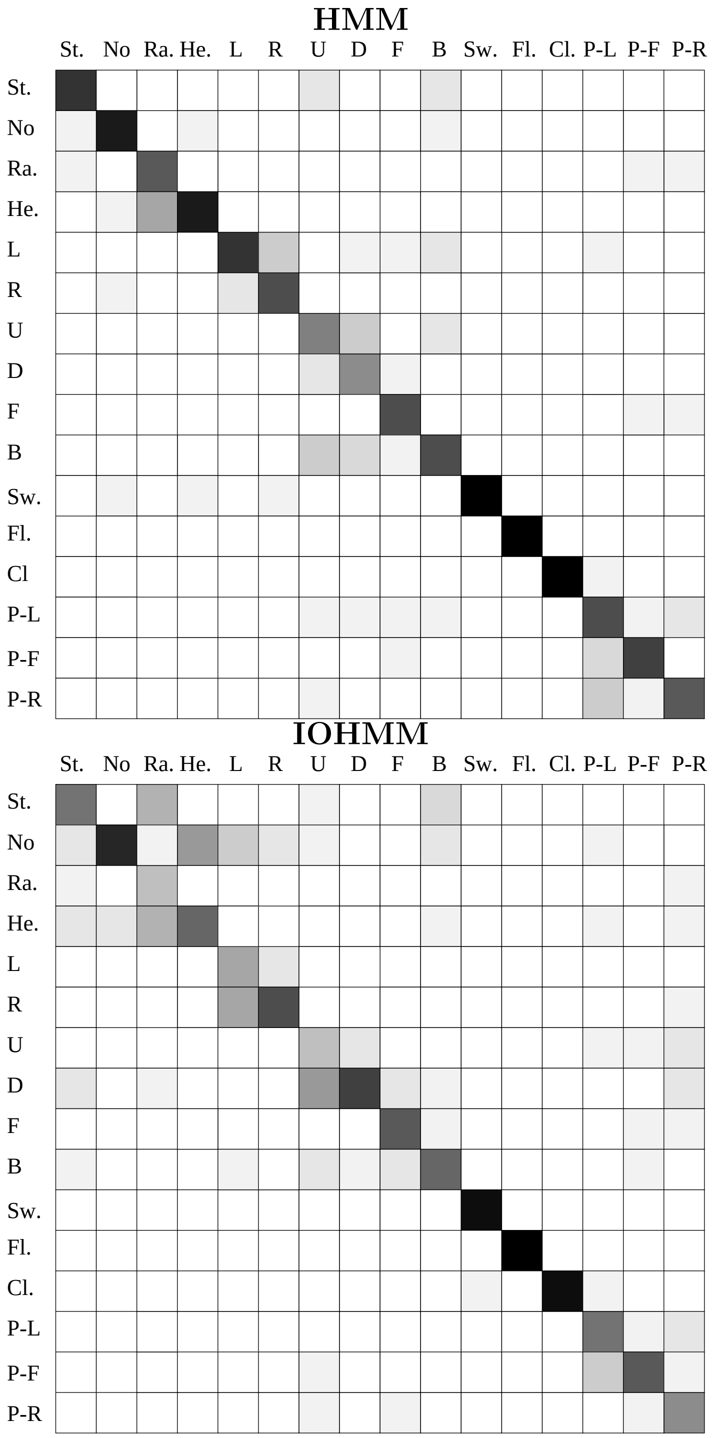
Concerning the IOHMM, only the “rotate-up”, “rotate-down”, “rotate-front” and “push” gestures are well classified. The “rotate-up” and “rotate-down” gestures are the only rotational gestures in which the movement of the hand is facing the cameras. And the “push” gesture is different from all the other gestures. For the “rotate-front” gesture, the angle gives very discriminative information. Figure 17 shows the evolution of the angle with time for this hand gesture.

These gesture classes are easy to discriminate with little training data. 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 continuous 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.

6. Summary and discussion

On the InteractPlay database, HMMs perform better than IOHMM. Results obtained with both HMMs and IOHMM show that 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.

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 (99% and 97% average recognition rate respectively). On the contrary, the average recognition rate for one-handed gestures using HMMs is equal to 64% and is equal to 55% when using IOHMM. This improvement in performance with two-handed gestures show that two-handed gestures help disambiguates the HGR process.



HMM

IOHMM

Figure 14: Confusion matrix for IOHMM and HMM on the test set for the 16-gesture database (rows: desired, columns: obtained). Black squares correspond to the well-classified gestures. Legend: Stop (St.), no (No), raise (Ra.), hello (He.), left (L), right (R), up (U), down (D), front (F), back (B), swim (Sw.), fly (Fl.), clap (Cl.), point left (P-L), point front (P-F), and point right (P-R).

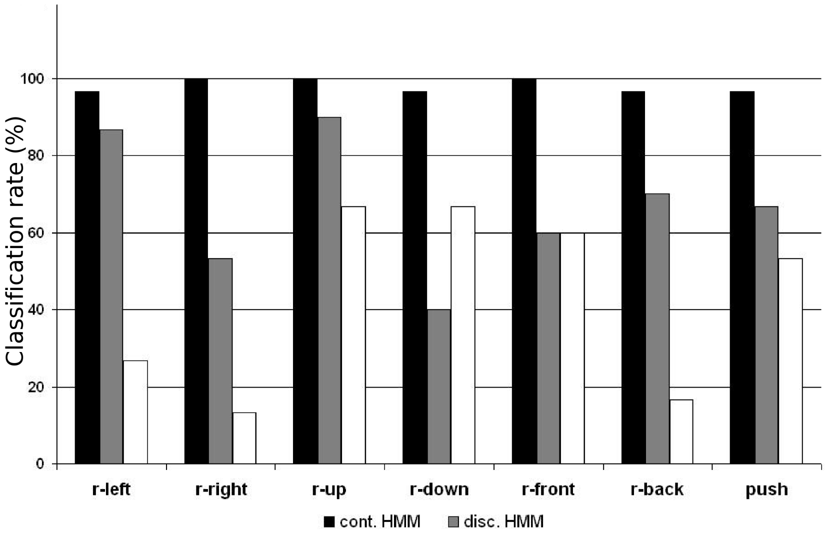


Figure 15: Classification rate (%) of the HMMs and IOHMM on the test set for the TwoHandManip database

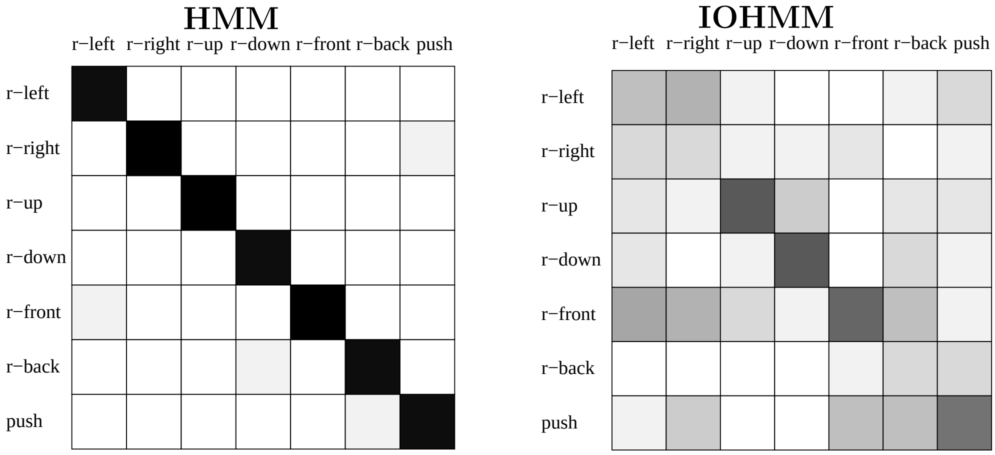


Figure 16: Confusion matrix for IOHMM and HMM on the test set for the manipulative gestures database (rows: desired, columns: obtained). Black squares correspond to the well-classified gestures.

The poor performances of the IOHMM cannot have arisen from the quantization process. Performance obtained with continuous HMMs and discrete HMMs, namely 75% for continuous HMMs versus 73% for discrete HMMs, show that there is no loss of discriminant information during the quantization process. This preprocessing step cannot explain the low results of the IOHMM on the InteractPlay database. A lack of training data could explain these performances. In particular, the estimation of the transition probability distributions, which are conditioned on the input sequences, can be difficult to achieve without a large quantity of training data [1].

Experiments performed on the two-handed gesture database showed, once again, that best performances were obtained with HMMs. Furthermore, they show that two-handed gestures can be easily recognized, even with few training data. On the contrary, IOHMM performs poorly on this purely two-handed gesture database, which had an even greater paucity of data. We observe that the recognition is not uniform with IOHMM. Some gestures are quite well recognized whereas some others are barely classified above the

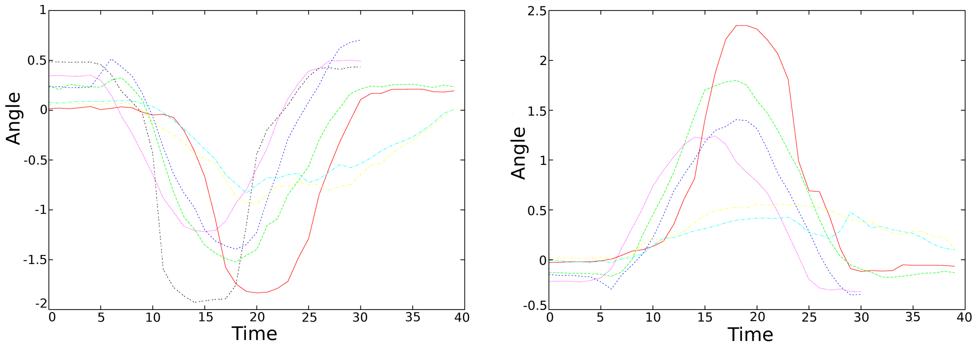


Figure 17: Evolution of the angle, in radians, with time, from the point of view of the left and right cameras, for the "rotate-front" gesture. The x-axis represents the time and the y-axis represents the angle. Each colored line is a different realization of the same gesture.

random rate. We note that 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. These observations suggest that, while overall the IOHMM performed poorly, some gestures are relatively easy to discriminate with relatively little training data. The quantization process can partially explain the poor performances of the IOHMM, as a degradation of performances is observed with discrete HMMs, but the main problem lies in the insufficiency of training data. When training and testing the IOHMM on the training data, the average recognition rate is higher than 75%. The IOHMM over trains on training data, which induces generalization problems when dealing with test data.

7. Conclusion

In this paper, we addressed the problem of the recognition of segmented dynamic hand gestures. Hand gestures were represented in the first database by the 3D trajectories of blobs obtained by tracking colored body parts and by 2D coordinates, angle, surface size and eccentricity of ellipses representing the two hands in the second database.

We provide recognition results obtained on these two open databases using two sequence processing techniques, namely IOHMMs and HMMs, implemented within the framework of an open source machine learning library. To summarize, our experiments have shown that the HMM is a good choice of model for performing hand gesture recognition. On the other hand, the IOHMM failed to meet our expectations on this task. Compared to previous experiments on hand gesture recognition using IOHMM [18], we see a dramatic decrease in performances. Compared to results obtained in [18], we have performed experiments on larger databases, ranging from 7 to 16 gesture classes versus only two gesture classes. Furthermore, we used discrete IOHMM instead of continuous IOHMM, but it has been shown that the quantization process is not the reason of the poor results. We cannot clearly state whether this is because it is a poorer model to the HMM or if these poor results were derived from a lack of training data. If the problem is derived from there being insufficient data to train the IOHMM then no simple solution exists, except to collect much more data. This brings us back to one of our initial motivations for performing this work, which was to promote the establishment of open resources for the evaluation of gesture recognition algorithms. We believe that while we have taken a positive step in this direction, more efforts still need to be undertaken in order to evaluate and compare more complex approaches.

In this paper, we have performed experiments using pre-segmented gesture sequences. All gestures performed in both databases start and end in the same rest position. One simple way to extend this work to the recognition of sequence of gestures is to ask the user to stay in this rest position during a few seconds [27]. Another possible alternative is to build a silence model to detect the rest position, such as what is done in speech recognition.

8. Acknowledgments

This research has been carried out in the framework of the GHOST project, funded by France Telecom R&D Pproject No. 021B276). This work was also co-funded by the Swiss National Science Foundation through the National center of Competence in Research (NCCR) on “Interactive Multimodal Information Management (IM2)”. The authors would like to thank J. Guerin and B. Rolland (from FTR&D Recherche et Développement) for recording and annotating the InteractPlay database. We also would like to thank O. Bernier for participating in the experiments conducted on this database.

References

[1] Y. Konig and H. Bourlard and N. Morgan, REMAP: Recursive Estimation and Maximization of A Posteriori Probabilities - Application to Transition-Based Connectionist Speech Recognition, in Advances in Neural Information Processing Systems 8, (1995) pp. 388-399

[2] H.K. Lee and J.H. Kim, An HMM-based threshold model approach for gesture recognition, in IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (1999) pp. 961-973

[3] J. Davis and M. Shah, Recognizing hand gestures, in Proceedings of European Conference on Computer Vision, 1994 pp. 331-340

[4] P. Hong and M. Turk and T.S. Huang, Gesture modeling and recognition using finite state machine, in Proceedings of the Fourth International Conference on Automatic Face and Gesture Recognition, 2000 pp. 410-415

[5] L.R. Rabiner, A tutorial on Hidden Markov Models and selected applications in speech recognition, in Proceedings of the IEEE, 1989 pp. 189-194

[6] T.E. Starner and A. Pentland, Visual recognition of american sign language using Hidden Markov Models, in International Workshop on Automatic Face and Gesture Recognition, 1995 pp.189-194

[7] L.R. Rabiner and B.H. Juang, An introduction to Hidden Markov Models, in IEEE ASSP Magazine 3, (1986) pp. 4-16

[8] N.C. Wah and S. Ranganath, Real-time gesture recognition system and application, in Image and Vision Computing 20, (2002) pp. 993-1007

[9] S.C.W. Ong and S. Ranganath, Automatic sign language analysis: A survey and the future beyond lexical meaning, in IEEE Transactions on Pattern Analysis and Machine Intelligence 27, (2005) pp. 873-891

[10] C. Vogler and D. Metaxas, A framework for recognizing the simultaneous aspects of american sign language, in Computer Vision and Image Understanding 81, (2001) pp. 358-384

[11] K.G. Derpanis and R.P. Wildes and J.K. Tsotsos, Vision-based gesture recognition within a linguistics-based framework, in Proceedings of the European Conference on Computer Vision, 2004 pp. 282-296

[12] S.F. Wong and R. Cipolla, Real-time interpretation of hand motions using a sparse bayesian classifier on motion gradient orientation images, in Proceedings of the British machine Vision Conference, 2005 pp. 379-388

[13] R. Bowden and D. Windridge and T. Kadir and A. Zisserman and M. Brady, A linguistic feature vector for the visual interpretation of sign language, in Proceedings of the Eight European Conference on Computer Vision, 2004 pp. 391-401

[14] A.P. Dempster and N.M. Laird and D.B. Rubin, Maximum-likelihood from incomplete data via the EM algorithm, in Journal of Royal Statistical Society B, 39, (1977) pp. 1-38

[15] Y. Bengio and P. Frasconi, An Input Output HMM architecture, in Advances in Neural Information Processing Systems, 1995 pp. 427-434

[16] D.E. Rumelhart and G.E. Hinton and R.J. Williams, Learning internal representations by back-propagating errors, in Nature 323, (1986) pp. 533-536

[17] Y. Bengio and P. Frasconi, Input/Output HMMs for sequence processing, in IEEE Transactions on Neural Networks 7, (1996) pp. 1231-1249

[18] S. Marcel and O. Bernier and J.E. Viallet and D. Collobert, Hand Gesture Recognition using Input/Output Hidden Markov Models, in Proceedings of the Conference on Automatic Face and Gesture Recognition, 2000 pp. 456-461

[19] J.S. Bridle, Towards Better Understanding of the Model implied by the Use of Dynamic Features in HMMs, in Proceedings of the International Conference on Spoken Language Processing, 2004 pp. 725-728

[20] C.K.I. Williams, How to pretend that correlated variables are independent by using difference observations, in Neural Computation 17, (2005) pp. 1-6

[21] J.A. Hartigan and M.A. Wong, A K-Means Clustering Algorithm, in Journal of Applied Statistics 28, (1979) pp. 100-108

[22] O. Bernier and D. Collobert, Head and hands 3D tracking in real time by the EM algorithm, in Proceedings of the IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, 2001 pp. 75-81

[23] W. Buxton and B. Myers, A study in two-handed input, in SIGGHI Bulletin 17, (1986) pp. 321-326

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

[25] A. Leganchuk and S. Zhai and W. Buxton, Manual and cognitive benefits of two-handed inputs: an experimental study, in ACM Transactions on Computer-Human Interaction 5, (1998) pp. 326-359

[26] C. Shan and Y. Wei and T. Tan and F. Ojardias, Real time hand tracking by combining particle filtering and mean shift, in Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004 pp. 664-674

[27] H. Yoon and J. Soh and Y.J. Bae and H.S. Yang, Hand gesture recognition using combined features of location, angle and velocity, in Pattern Recognition 34, (2001) pp. 1491-1501

[28] I. Ulusoy and C. Bishop, Comparison of generative and discriminative techniques for object detection and classification, in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005 pp. 173-195