IDIAP RESEARCH REPORT

Two-Handed Gesture Recognition

Agnès Just Sébastien Marcel

IDIAP–RR 05-24

May 2005

SUBMITTED FOR PUBLICATION

Dalle Molle Institute
for Perceptual Artificial
Intelligence • P.O.Box 592 •
Martigny • Valais • Switzerland

phone +41 − 27 − 721 77 11
fax +41 − 27 − 721 77 12
e-mail secretariat@idiap.ch
internet http://www.idiap.ch

[a] IDIAP Research Institute, CH-1920 Martigny, Switzerland

IDIAP Research Report 05-24

Two-Handed Gesture Recognition

Agnès Just Sébastien Marcel

May 2005

submitted for publication

Abstract. Nowadays, computer interaction is mostly done using dedicated devices. But gestures are an easy mean of expression between humans that could be used to communicate with computers in a more natural manner. Most of the current research on hand gesture recognition for Human-Computer Interaction deals with one-handed gestures. But two-handed gestures can provide more efficient and easy to interact with user interfaces. It is particularly the case with two-handed gestures we do in the physical world, such as gestures to manipulate objects. It would be very valuable to permit to the user to interact with virtual objects in the same way that he/she interacts with physical ones. This paper presents a two-handed gesture database to manipulate virtual objects on the screen (mostly rotations) and some recognition experiment using Hidden Markov Models (HMMs). The results obtained with this state-of-the-art algorithm are really encouraging. These gestures would improve the interaction performance between the user and virtual reality applications.

1 Introduction

Computers are about fifty years old but their fast evolution, from machines weighting tonnes to personal computers and laptops, has transformed them into a key element of our society. As they influence our day-to-day life, Human-Computer Interaction (HCI) has been a lively field of research. Firstly based on the use of punched cards, reserved to experts, the interaction has evolved to the direct manipulation of graphical objects, where visible objects on the screen, such as icons and windows, are directly manipulated with a pointing device. This evolution has permitted to the computer to be easier to interact with, and is not anymore reserved to experts in the field. But this interaction is still done using "stone-age" devices, such as keyboards, mice and graphic boards. Such interfaces are often not natural nor intuitive to use. 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.

The use of natural hand gestures for computer-human interaction can help people to communicate with computer in a more intuitive way. We constantly use our hands to interact with objects: move them, modify them, transform them. In the same unconscious way, we gesticulate while speaking, in order to communicate ideas ('stop', 'come closer', 'no', etc). Gestures are thus a mean of non-verbal communication, ranging from simple actions (pointing at objects for example) to the more complex ones (such as expressing feelings or communicating with others).

The potential power of gesture has already been demonstrated in applications that use the hand gesture input to control a computer while giving a presentation for instance [5]. Other possible applications of gesture recognition techniques include computer-controlled games, teleconferencing, robotics or the manipulation of objects by CAD designers. Nevertheless, in order to use hand gestures to interact with computers, it is necessary to provide the means by which they can be interpreted by the computers. Gestural HCI can be effectuated by using video cameras or special numeric gloves for example. But the use of video cameras is more natural than any dedicated acquisition device, and unfortunately much more challenging too, as hand gestures are highly variable from one person to another.

Hand gestures can be seen on two different angles: the *static* angle and the *dynamic* angle. The *static* aspect of gestures 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 the 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, classification outputs the class the gesture belongs to.

In this paper, we will present some results on the recognition of segmented hand gestures using a state-of-the-art sequence processing algorithm, namely Hidden Markov models (HMM). For that purpose, we propose to the scientific community a new database, publicly available. The novelty of this database is that all gestures are **two-handed**. In order to argument our research directions, we will first present an overview of related work and describe the database. We will then introduce Hidden Markov models, discuss the results obtained using HMMs and propose some future research that will need to be done.

2 Related Work

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 [8], Sato et al. proposed a system in which users could execute commands by changing their hand shape in order to handle a computer generated object in a virtual reality environment. Wah and Ranganath [11] proposed a prototype which permitted to the user to move and resize windows and objects, and open/close windows, by using some simple hand gestures. Even with common devices, such as the mouse or the graphic board, we only use one hand 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 seems to be of potential benefit for the users. Many experiments have been conducted in order to test the validity of this hypothesis. And the obtained results are very encouraging. Buxton and Myers [1] run two experiments to investigate two-handed input. The first experiment involved the performance of a compound selection/positioning task. The second experiment involved the performance of a compound navigation/selection task. The conclusions that can be drawn from the results on these two experiments are that using the two hands is a natural behavior for the users, and that using two-handed gestures increases significantly the speed of the interaction process. Leganchuk et al. [6] conducted some more experiments on the benefit of two-handed input. There experimental task was area sweeping, which consists in drawing a bounding box surrounding a set of objects in a graphics program. There conclusions support the fact that two-handed techniques outperform the conventional unimanual techniques. Bimanual techniques are faster, and for high-demanding tasks, the advantage of two-handed input over one-handed input becomes more pronounced. Very 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 gesture movements. They developed a two-handed drawing tool [2] with which the user can draw and manipulate objects interactively. The right hand is used to draw and to manipulate objects. The left hand is used to manipulate menus and to assist the right hand.

In his thesis, Sturman [10] emphasizes the fact that it is necessary to use the skills of the user. Thus, the use of the two hands is even more natural in case of object manipulations. Myron Krueger [4] contended that users should be able to manipulate graphical objects like objects in the physical world. In the VideoDesk, he showed that it is possible for a range of tasks, using both hands to stretch, position, rotates different graphical objects in an intuitive and natural manner. Sturman showed also that tasks can be made easier to learn and master by taking advantage of pre-acquired skills in hand gesture input, reducing training expense and time. Another interesting quality of two-handed gestures are that they are body-centered. Body-centered coordinate systems are more natural to work with and can improve performance for object manipulation tasks.

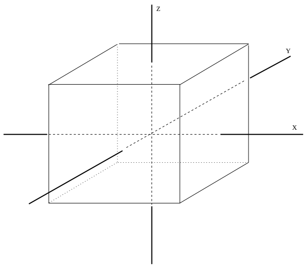
3 TwoHandManip Database

3.1 Description

It has been shown in section 2 that two-handed inputs are one of the most natural way to interact with the computer. It has also been demonstrated that the use of the two hands is even more natural for object manipulation. The database proposed here is a set of 7 two-handed manipulation gestures, 6 of them are rotation gestures. These gestures consist in the rotation of the two hands together along the 3 main axis, and the last one is the action of pushing an object, with the hands forward:

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

- 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



As the goal of the gestures is to manipulate virtual objects on a screen (rotate them in all the six direction and push them), they occur on a desk in front of a display. In order to record the gestures, two simple cameras (standard Web-cams with USB port) have been used to deal with occlusion problems. Figure 1 shows the set-up used to record the data. The two cameras are placed on each side of the hands of the user. The field of view is large enough to always record the two hands. The acquisition is synchronized at 12 images per second.

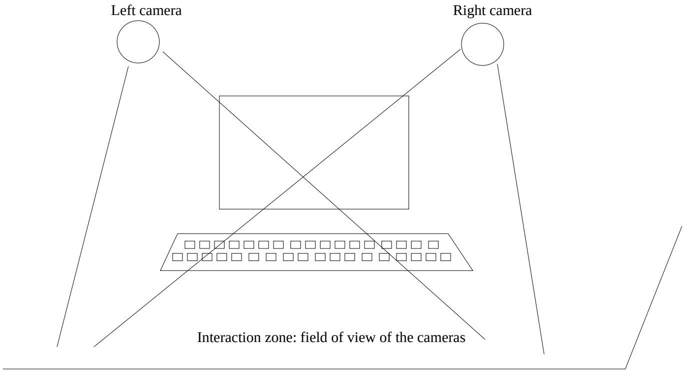


Figure 1: Camera set-up for recording the two-handed database.

For each gesture, two different views were recorded, one from each camera. Each gesturer wears two colored gloves: one blue and the other yellow (Figure 2). As the gesture are mostly rotation gestures, problems of occlusion occurred. Colored gloves are here to facilitate the hand tracking and avoid the occlusion problems. Indeed, in this paper we focus on the recognition and not on the hand segmentation from the image, nor the hand tracking. This could be achieved by several technics such as particle filtering.

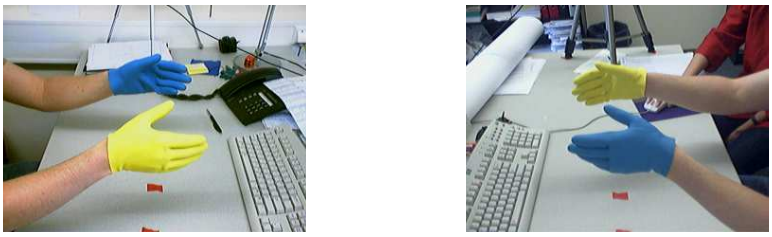


Figure 2: Point of view from the right and left cameras.

7 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. This database is available for download [2].

[2] http://www.idiap.ch/˜marcel/Databases/twohanded/main.html

3.2 Feature Extraction

As a first step, on each image of gesture sequences, a simple lookup table filter is applied (one for each glove color: blue and yellow). Color zones are then approximated to a Gaussian. Figure 3 shows the result of the colored pixel detection. The hands are bounded by an ellipse of center the mean of the Gaussian.

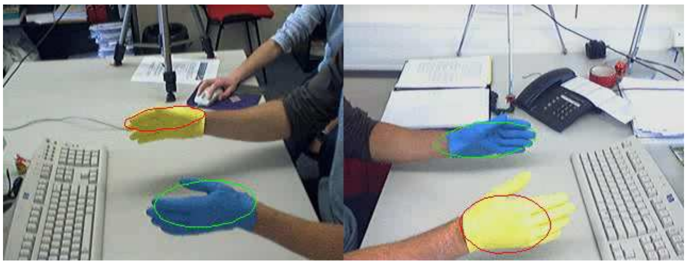


Figure 3: Representation of each hand blob for the left and right cameras

The mean (μ_x, μ_y) of each blob, namely the center of the boundary ellipse, is used as a feature. Let call a the half major axis and b the half minor axis of the ellipse. We can compute the excentricity of the ellipse using the formula $e = \sqrt{1 - \frac{b^2}{a^2}}$, and the surface of the ellipse, $s = \pi \times ab$. We also compute the angle α between the major axis of the ellipse and the horizon (Figure 7).

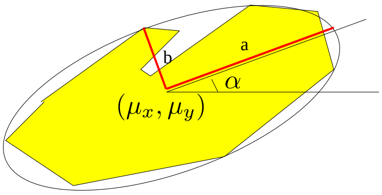


Figure 4: Features extracted

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 instead.

Let X be the feature vector corresponding 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 excentriciy of the two ellipses for the left and right camera images. Finally $X \in \mathbb{R}^{20}$.

4 Hidden Markov Models

A Hidden Markov Model (HMM) [7] 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 $P(y_t|q_t = i), \forall i =$

$1 \dots N$ at time t depends only on the current state q_t. The transition probability between states $P(q_t = i | q_{t-1} = j), \forall i, j = 1 \dots N$ 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 $i \in \{1, \dots, N\}$ is the initial distribution over all the states: $\pi_i = P(q_1 = i)$,
- the matrix $A = (a_{ij})$ with $i, j \in \{1, \dots, N\}$ which determine the transition probabilities from the state i to the state j:
  $a_{ij} = P(q_{t+1} = j | q_t = i), \forall (i, j) \in \{1, \dots, N\}^2$
- the set of parameters $B = (b_j(y_t))$ with $j \in \{1, \dots, N\}, t \in \{1, \dots, T\}$, which represents the observation probability of y_t in the state j:
  $b_j(y_t) = P(y_t | q_t = j), \forall j \in \{1, \dots, N\}$ and $\forall t \in \{1, \dots, T\}$

Then, the training of a HMM can be carried out using the *Expectation-Maximization* (EM) algorithm [3]. As we are in the case of continuous data, the set of observation probabilities are represented by a mixture of Gaussians.

In order 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 5 represent the state graph of a left-right topology for an HMM.

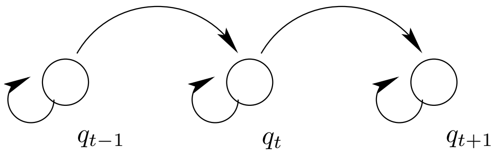


Figure 5: Left-right topology.

As we try to recognize several gesture classes, we have one HMM per class and we use a naive Bayes classifier to perform the classification, assuming equal prior probabilities for each gesture class. Let then C be the number of gesture classes. In the recognition phase, the class the gesture belongs to is $\text{argmax}_{c \in \{1, \dots, C\}} P(y_1^T | \Lambda_c)$.

5 Results and Conclusions

5.1 Experiment Results

In order to find the optimal hyper-parameters of the HMMs (number of states and number of Gaussians), the database has been split into three subsets: the training set T, the validation set V and the test set Te (cf. Table 1). The gestures of one particular person are only used once in a particular subset.

	T	V	Te
♯ people	2	2	3

Table 1: Number of person in each set

Different possibilities for the hyper-parameters have been tried on T. The selection of the best parameters has been effectuated on V. Finally, a model has been trained on both T and V and tested on Te. The best results were obtained with 1 Gaussian per state and 12 states.

Table 2 provides the recognition rate for each gesture. HMMs achieve 98% average recognition rate.

	rotate left	rotate right	rotate up	rotate down	rotate front	rotate back	push
HMM	96.67	100	100	96.67	100	96.67	96.67

Table 2: Classification rate (in %) on the test set

Even if the results are exceptionally good(cf. Figure 6), the conclusion has to be balanced. The database is quite small and we can expect that the results will slightly degrade if we increase the number of data.

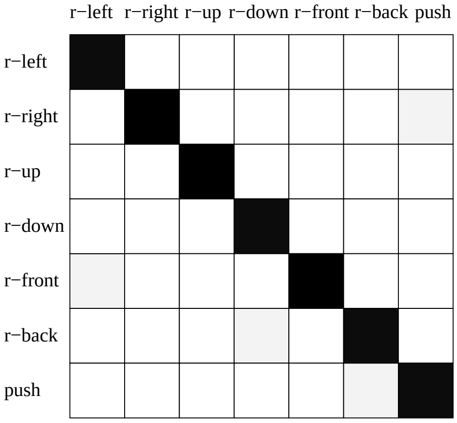


Figure 6: Confusion matrix for HMMs on the test set. Black squares correspond to the well-classified gestures

We can also conclude that the features extracted are suited to the HMMs. They permit to the HMMs to model the 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-front” and “rotate-back” gestures (cf. Figure 7).

The center of the ellipse helps also in the modeling of the gestures. Concerning the other features, such as the surface and excentricity, it is difficult to know if they add something to the recognition process as the gloves are very loose. In order to verify the usefulness of these features, an other experiment has been conducted. HMMs have been trained with only the following features: μ_x, μ_y, α, without e and s (see section 3.2). The best result has been obtained with 14 states and 45 Gaussians per state. Table 3 shows the classification rate on the seven gestures. HMMS achieve 72% average recognition rate.

	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

Table 3: Classification rate (in %) on the test set

Intuitively, the coordinates of the center of each blob as well as the angle are characteristic of the gestures. We can see that with only these three types of features, the recognition rate of the ’rotate-up’, ’rotate-down’ and ’rotate-front’ gestures decreases. In the light of this last experiment, we can conclude that some information is present in the excentricity and surface features thus helping the recognition process.

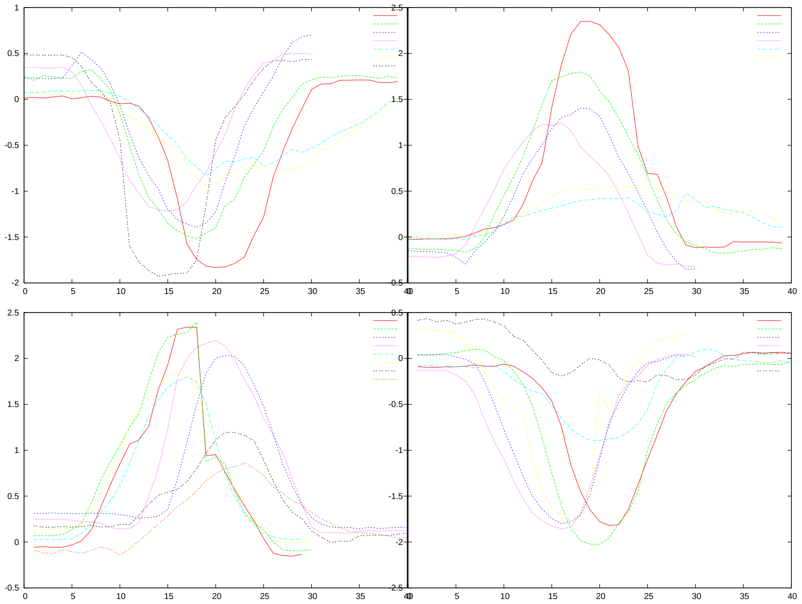


Figure 7: Evolution of the angle from the point of view of the left (left column) and right (right column) cameras, for the 'rotate-front' (first row) and 'rotate-back' (second row) gestures

5.2 Future Directions

This database is made of segmented gestures. But in order to manipulate virtual objects on the screen, it is necessary to be able to perform gestures in a continuous way. A next research step would be to try to recognize the same gestures, but sequentially. For that purpose, we need to recognize the beginning and the end of a gesture. Spotting is a crucial problem for real-time gesture recognition. It is also obvious that users would make errors while performing gestures, so it is also necessary to be able to reject unknown gestures.

Another interesting research direction would be to try to recognize these gestures without the need of the colored gloves. But this problem is even more complex than the previous one as it is related to occlusion problems. Some techniques such as particle filtering could be used for that purpose. It has already been applied successfully [9] to hand gesture tracking and recognition.

6 Acknowledgments

This research has been carried out in the framework of the Swiss National Science Foundation through the National Center of Competence in Research (NCCR) on “Interactive Multimodal Information Management (IM2)”. This work was also cofunded by France Telecom R&D through the GHOST project (project number 021B276).

References

[1] W. Buxton and B.A. Myers. A study in two-handed input. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 321–326, 1986.

[2] 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 (AVI 2002).*

[3] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum-likelihood from incomplete data via the EM algorithm. *Journal of Royal Statistical Society*, 39:1–38, 1977.

[4] M. Krueger. *Artificial Reality II.* Addison-Wesley, 1991.

[5] H. Lee and J. H. Kim. An HMM-Based Threshold Model Approach for Gesture Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(10):961–973, 1999.

[6] A. Leganchuk, S. Zhai, and W. Buxton. Manual and cognitive benefits of two-handed input: An experimental study. *ACM Transactions on Computer-Human Interaction*, 5(4):326–359, December 1998.

[7] L.R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In *Proceedings of the IEEE*, volume 77, february 1989.

[8] 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 the Virtual Reality 2001 Conference (VR'01).*

[9] C. Shan, Y. Wei, T. Tan, and F. Ojardias. Real time hand tracking by combining particle filtering and mean shift. *Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, 2004.

[10] D.J. Sturman. *Whole-Hand Input.* PhD thesis, Massachusetts Institute of Technology, 1992.

[11] N.C. Wah and S. Ranganath. Real-time gesture recognition system and application. *Image and Vision Computing*, 20:993–1007, 2002.