Prediction of user's intention based on the hand motion recognition

Miguel Carrasco
Computer Engineering Department
Pontificia Universidad Católica de Chile,
Av. Vicuña Mackenna 4860 (143), Santiago de Chile
mlcarras@puc.cl

Xavier Clady
Institut des Systèmes Intelligents et de Robotique
Université Pierre et Marie Curie - Paris VI
4 Place Jussieu, 75005 Paris. France.
xavier.clady@upmc.fr

***Abstract*—Most methods proposed in the literature for predicting movements involved in a reach-to-grasp action by human being are designed using passive methods, i.e, by using a camera in front of the user. A novel approach to understand the user's intentions with computer vision methods is proposed in this paper. Our solution performs a hand motion estimation using a wearable system. To realize such system, we employ a camera beneath the wrist to capture motion from the user's perspective. To predict hand intentions, we characterize a hand-movement trough a Hidden-Markov Model framework. As main result, our system can predict the movement before the hand can reach an object with a performance near to 90% in average.**

***Index Terms*—Gesture recognition, motion planning, image motion analysis, image motion detection.**

I. Introduction

The human beings posses a highly developed ability to grasp objects under many different conditions, taking into account variations in position, location, structure and orientation. This natural ability controlled by the human brain is called the eye-hand coordination. Normally, a grasp movement is initiated time before the hand can reach an object, and it is regulated by the interaction of several sensorimotor systems such as the visual system, vestibular system and proprioception working in conjunction with head, eye, hand and arm control systems [1]. A central part of this activity occurs in different cortical and subcortical brain regions, taking special importance the use of our underlying cognitive process, like the attention and memory. According to Flanagan and Lederman [2], when we grasp an object, the information perceived by our sensory signals is the result of our preconceived ideas of the shape of the object, and the interpretation of perceived, that is, our brain uses memory representations and visual information simultaneously to grasp objects. We infer that this dynamic activity is an exploratory searching, attempting to find the best solution for planning movements to the target and controlling the user's hand. Not as the use of active sensors with the purpose of capturing data. Researchers in many fields have been studying this process for many years, trying to understand the brain mechanism that controls this coordination. However, up to now there is not an unique theory which explains this effectively, and furthermore, it is not completely understood [3, 4]. In spite of this, we observe that this field has spread to other subjects, specially into the Human-Computer Interaction (HCI). Within the HCI domain, there is a special interest on designing computer interfaces by taking advantage of the human interaction, and one of them is particularly the human vision.

This paper describes a novel approach to recognize the user's intention based on the visual information captured from the user itself. Our work stands in contrast with classical methods to recognize hand gestures. Generally, most motion recognition methods captures the user's movements by tracking body parts. Instead, we develop a system that captures the scene using the reach-to-grasp movement; thus, implicitly we infer the user's intention. The system is composed by one visual acquisition system: a microcamera beneath the wrist. Today, with the advent of fast microcameras, it is possible to use a camera without annoying the user's interaction. A general overview of the system is presented in Fig.1

In our problem, the user's intention is the active conscious action with the goal to reach-to-grasp an object with a hand. Generally, we perform grasp actions very fast and precisely; almost unconsciously because our brain resolves this complex coordination in a small period. The grasp action has a particular period when the user initiates its movement toward an object. Based on such observation, our systems exploits this feature by allowing us to detect the user's intention. This work is very challenging because many postures for reaching a target can be presented in the same user, increasing the complexity to characterize an unique representation of the grasp movement. Likewise, due to the high variability involved in each movement, several assumptions are need it in order to predict the movement at an early stage. This is the main point addressed in this paper.

The central question here is, why is relevant to know the user's intention? The prediction of the user's intention can be used as a key factor within the HCI domain, as in the work described below: Interactive robots have been used efficiently in the rehabilitation domain; nonetheless, the co-manipulation domain is still under-exploited mainly by a lack of research in this area. Recently an important effort has been made for designing an active orthosis for aiding people

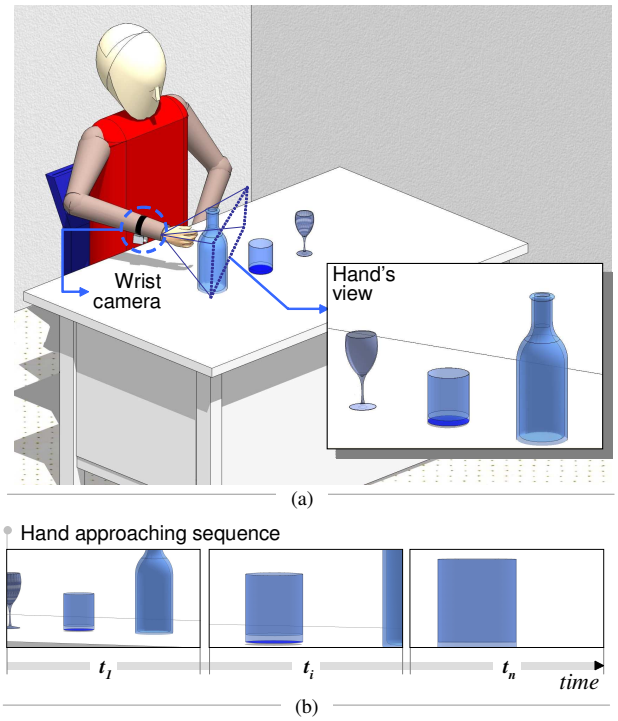


Fig. 1. Schematic view of the propose framework with a wearable camera. (a) Camera beneath the user's wrist share the same field-of-view (FOV) only when the user starts the movement to reach-to-grasp. (b) Hand approaching sequence using the camera beneath the user's wrist. In time t_1 multiple objects are detected, later, in time t_i and t_n the field-of-view (FOV) is almost filled due to the proximity between the hand and an object.

with arm disabilities by the BRAHMA[1] project. For realizing such system, it is critical to know the user's intention, in this way, the active orthosis can operate to control the user's arm. The rest of the paper is organized as follows: Section II discusses the prior work on the human gesture recognition; Section III explains our proposed method; Section IV shows the experimental results; and finally, Section V presents our contributions and our future work.

II. Related Work

The gesture recognition can be defined as the problem to follow body parts over the space-time in order to interpret the motion behavior as particular gesture. Based on the Aggarwal and Cai [5] definition, the gesture recognition requires to perform three general tasks. First, to identify some human body structure or low-level features such as points, blobs, 2D contours or 3D-volumes; second, to track human movements using low-level features by matching between consecutive frames or using the motion itself; and third, to recognize the human activity by matching the motion descriptor captured in the tracking process against the recognition framework. The last step is considered a higher level task due to, the recognition task requires the classification of varying feature data over time [6]. The problem of interpreting the human gestures is defined as a learning process. In the training phase, some sequences are used to learn the user's behavior, labeling as a particular human gesture. Later, in the matching phase unknowns test sequences are compared against a model so as to be classified as particular gesture. Most approaches designed to detect human gestures are based on template matching or appearance-base models:

A. Template matching

This approach characterizes the human motion as means of recognition instead of using specific parts of the body. Thus, the action is represented by one robust vector. Polana and Nelson [7] were pioneered in applying this approach. They compute the motion fields between successive frames by dividing each frame into the spatial grid, forming a high dimensional feature vector. This feature vector was conformed by optical flow magnitudes and periodicity flow frames. The recognition task was performed using a nearest centroid algorithm by comparing a feature vector against a reference motion template. Bobick and Davis [8] designed a similar approach, but they extract features using a motion-energy image (MEI) and motion-history image (MHI). First, they construct the motion images as a binary representation of temporal difference between successive frames. These motion images are accumulated in time forming the MEI. The MHI is forming by using a temporal decay of each pixel intensity at that position. Thereby, brighter pixels represents the more recently motion. In order to get an invariant representation of the action, they extract invariant features of a set of MHIs and MEIs. These features are used later to recognize an input action by calculating the Mahalanobis distance between the moment description and known actions. In a similar way, but considering the action template as a space-time 3D volume, Shechtman and Irani [9] extended the idea of 2D correlation into a 3D space-time volume by defining the action as a spatio-temporal geometric structure. For this, they compute the correlation of a small video by seeking peaks in the behavioral correlation surface.

B. Appearance-base models

This approach considers the human motion as a set of local features, where an action is described as a sequence of images. One common technic employed in this approach is the Hidden Markov Models (HMM). HMM is a probabilistic technique to recognize patterns in temporal time series, usually employed in speech recognition. Starting with the work of Yamato et al. [10], currently, HMMs has been adopted as a tool for recognizing the human motion. Yamato et al. developed the first human recognition method to recognize six tennis strokes using low-level features as an input to a HMM learning process. Although, low-level features do not provide rich descriptions of the motion, Yamato et al. shows that these features are enough to identify human movements. In the same line, Starner and Pentland [11] proposed a system

[1]The BRAHMA project is currently being carried out by five French laboratories with the aim to develop advanced robot technology for assistance to human upper limb motion. More information available in http://brahma.robot.jussieu.fr/

to interprets the American Sign Language (ASL) using an HMM. They used low-level features such as shape, orientation and trajectory as input to an HMM without describing the hand shape. Recently, a novel approach by Achard et al. [12] proposes to use semi-global features by estimating micro-movements from 3D spatio-temporal volumes. They determined invariant 14 moments so as to be used as an input of an HMM framework. In general, the aim of recognizing human intentions is to predict the inherent intentions on people without explicit instructions. In other words, we aim to know when an user initiates a grasp movement toward an unknown object. In following section we address this problem with further details.

III. Proposed Method

Suppose that a user is performing a grasp movement toward an object, one can infer that the trajectory remains steady. Accordingly, all objects in the scene start to disappear from the user's FOV until the hand has reached the object required (Fig.1b). Conversely, if the hand movement is too stochastic, the probability that a user is performing a reach-to-grasp movement is reduced because the motion-descriptor does not have a pattern of approaching. This last statement is the key point of the hand recognition. The problem now is how to build a robust pattern of motion. To achieve this goal, a tracking analysis is performed only for resolving the corresponding problem. Here we use a robust invariant descriptor called SURF [13]; mainly by its robustness and speed against variations in scale and rotations. A general overview of the hand motion recognition is presented in Fig.2.

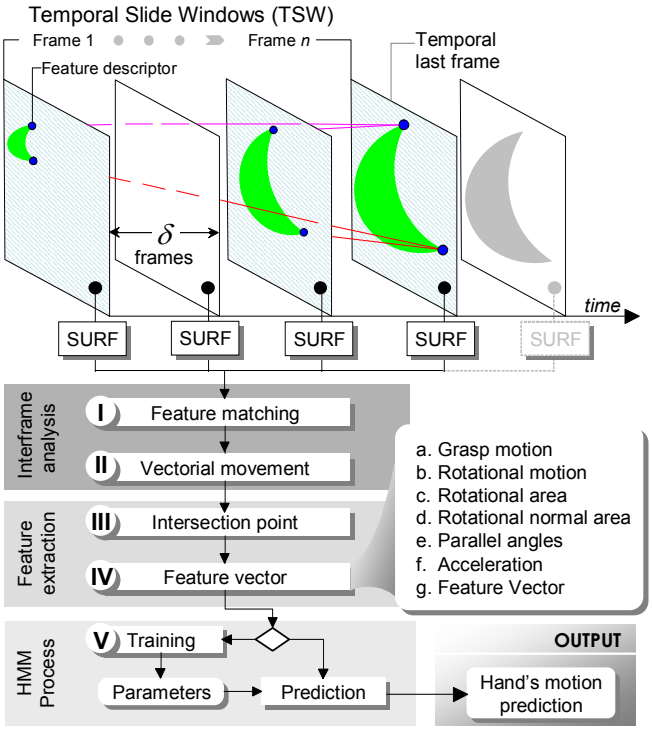


Fig. 2. Proposed hand intention recognition model based on the analysis of temporal slide windows (TSW) interspaced by δ-frames

I. Feature matching

Most tracking algorithms based on appearance-base models compute the object trajectory by using a displacement difference between multiple frames. Those methods are well suited when the object motion is smooth and without abrupt changes, as for example methods based on the estimation of optical flow [14]. Contrariwise, in our problem the hand motion is particulary fast when a user is performing a reach-to-grasp action, or too stochastic in other cases. For this reason, we propose to analyze the motion displacement between intermediates frames. In a similar way of the spatio-temporal methods described in [9], our method use Temporal Slide Windows (TSW) extracted along the video sequence.

Based on this idea, we propose to build a motion model of multiple corresponding points in relation with each last temporal corresponding frame. Unlike the current classical frame-to-frame correspondences, our method is able to estimate a global motion from each TSW. Firstly we compute invariant interest-points by means of the SURF algorithm. This task is performed for each δ-frames contained on a TSW. This is schematically outlined in Fig.3. For instance, let $\mathbf{p}_1^j = [x_1^j, y_1^j, 1]^\top$ the position of the j-th interest point in time $t = 1$ stored in homogenous coordinates. If this interest point is corresponding with a point $\mathbf{p}_n^j$ in time $t = n$ it must have a strong similarity between their features. Likewise, after δ-frames, a point $\mathbf{p}_i^j$ is corresponding with $\mathbf{p}_n^j$ using the same similarity metric, where $i \in \{1, \ldots, n-\delta\}$. Although points $\mathbf{p}_1^j$ and $\mathbf{p}_{1+\delta}^j$ are corresponding to each other, here we are not interesting in motion between some small displacement. Conversely, we seek to compute the global motion. In the following analysis we consider the first TSW contained on time $t = 1$ and $t = n$. Secondly, once extracted interest points for some δ-frames, we try to find a vector that relates the j-th point $\mathbf{p}_i^j \mapsto \mathbf{p}_n^j$, for all $i \in \{1, \ldots, n-\delta\}$. Here, the key point is to relate multiple corresponding points with respect to the set of points extracted in the last frame. If for some frames this relation does not exist, it is not relevant while a minimum number of correspondences are established. As a result, we reduce the motion complexity caused by the inter-frame approach, and also we assure a single correspondence between multiple frames. To resolve this task, we use the Nearest-Neighbor with Distance Ratio criterion (NNDR) [15]. In general, the NNDR criterion reduces the number of corresponding points when there are noise-points, and when it does not exist a corresponding point. This last fault normally occurs when there is a fast motion sequence, as in our problem.

II. Vectorial movement

Once established a set of corresponding points contained in the TWS, we proceed to determine the motion vector for that point. For instance, let $\mathbf{q}_{i,n}^j$ be an homogenous vector that crosses the points $\mathbf{p}_i^j \mapsto \mathbf{p}_n^{j'}$ defined as $\mathbf{q}_{i,n}^{j,j'} = \mathbf{p}_i^j \times \mathbf{p}_n^{j'} = [x_i^j, y_i^j, 1] \times [x_n^{j'}, y_n^{j'}, 1]$. The vector $\mathbf{q}_{i,n}^{j,j'}$ is established between

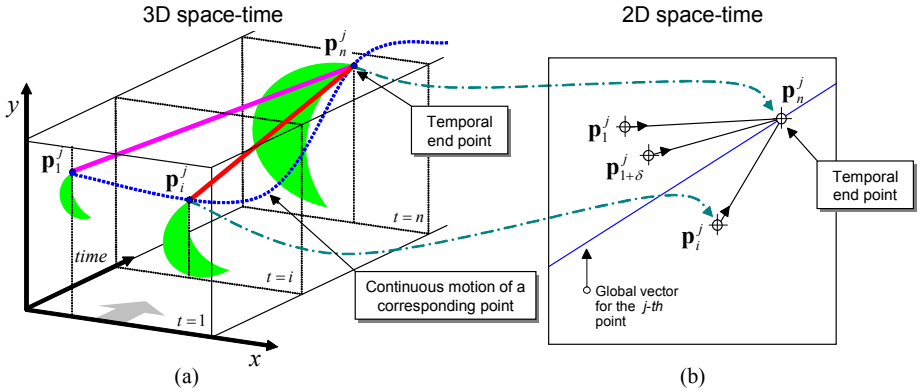


Fig. 3. Schematic view of the point correspondence in the time-space. (a) Corresponding points in 3D time-space volume; (b) Corresponding points in 2D coordinates.

time $t = i$ and $t = n$ only for the j-th point[2]; however, several vectors of the same point are required to establish a motion field along this time. For this, we define the general motion of multiple vectors that arrive at the point $\mathbf{p}_n^j$ as

$$\mathbf{Q}_{1\mapsto n}^j = \left[\mathbf{q}_1^j, \ldots, \mathbf{q}_i^j, \ldots, \mathbf{q}_{n-\delta}^j\right]^\top$$

The matrix $\mathbf{Q}_{1\mapsto n}^j$ defines the motion field of the j-th point for all frames until time $t = n$, for each δ-frames. Nevertheless, this procedure does not assures that in every δ-frames there is a correspondence, because of high geometric and photometric distortions, or partial occlusions that could be present in some frames. To assure that the motion field is correct, we define a parameter ρ as the minimum number of rows in the matrix $\mathbf{Q}_{1\mapsto n}^j$ where $inliers \geq \rho$ is fulfilled. Conversely, if this last constraint is not fulfilled, we discard the motion field for that point.

The next step is to derive only one vector that represents the motion of the j-th point along the time. For this, we map the angle of the j-th feature point along all $inliers$-frames as

$$\mathbf{F}_{1\mapsto n}^j = \left[\mathbf{F}_{1,n}^j, \ldots, \mathbf{F}_{i,n}^j, \ldots, \mathbf{F}_{n-\delta,n}^j\right],$$

where $\mathbf{F}_{1\mapsto n}^j$ is a $(1\times inlier)$ vector of feature angles extracted in different δ-frames for the j-th point. In other words, each angle $\mathbf{F}_{i,n}^j$ weights the relative significance between features of points $\mathbf{p}_i^j \mapsto \mathbf{p}_n^{j'}$. Thus, while smaller is the angle between two vectors, stronger is the relation of the same point. Conversely, when the angle is maximal, it could be considered as noise. Based on such observation, we propose to represent each angle-value as a weight vector after a linear transformation. Hence, the vector $\mathbf{F}_{1\mapsto n}^j$ is transformed to a vector $\tilde{\mathbf{F}}_{1\mapsto n}^j$, used for weighting each motion vector such that

$$\tilde{\mathbf{F}}_{1\mapsto n}^j = 1 - \frac{\alpha \mathbf{F}_{1\mapsto n}^j}{\max\left(\mathbf{F}_{1\mapsto n}^j\right)}.$$

[2]For simplicity, we will change the notation $\mathbf{q}_{i,n}^{j,j'}$ as $\mathbf{q}_i^j$, assuming a correct matching between the j-th and j'-th and between time $t = i$ and $t = n$

The vector $\tilde{\mathbf{F}}_{1\mapsto n}^j$ is a scale-value that gives more relevance to smaller values. That is, the maximal value is zero, and the smaller value is maximal when $\alpha = 1$. Experimentally α was fixed at 0.98 to use all vectors mapped in $\mathbf{F}_{1\mapsto n}^j$. Nonetheless, the vector $\tilde{\mathbf{F}}_{1\mapsto n}^j$ is not correctly scaled. To determine a correct scale measure, we compute $\mathbf{N}_{1\mapsto n}^j$ as

$$\mathbf{N}_{1\mapsto n}^j = \frac{\tilde{\mathbf{F}}_{1\mapsto n}^j}{\sum_{i=1}^{inlier} \tilde{\mathbf{F}}_{1\mapsto n}^j(i)},$$

where $\sum_{i=1}^{inlier} \mathbf{N}_{1\mapsto n}^j(i) = 1$. The resultant vector $\mathbf{N}_{1\mapsto n}^j$ gives a correct measure of each angle value by taking into account the relative significance between the angles contained in $\mathbf{F}_{1\mapsto n}^j$. Finally, we compute the global vector of the j-th point as the vector

$$\mathbf{v}_{1\mapsto n}^j = \mathbf{Q}_{1\mapsto n}^{j\top} \mathbf{N}_{1\mapsto n}^{j\top}$$

where $\mathbf{v}_{1\mapsto n}^j$ is a (1×3) vector that maps all $\mathbf{Q}_{1\mapsto n}^j(k)$ vectors into a single one by giving more value to vectors with more similarity, based on the weight feature vector encoded in $\mathbf{N}_{1\mapsto n}^j$. More precisely, $\mathbf{v}_{1\mapsto n}^j$ is a directional vector of the j-th point, as shown in Fig.4a. Additionally, we compute the normal directional vector in order to detect rotational movements. For this, let $\mathbf{q}_{\perp i,n}^{\ j}$ the normal vector between points $\mathbf{p}_i^j \mapsto \mathbf{p}_n^{j'}$ established between time $t = i$ and $t = n$, defined as

$$\mathbf{q}_{\perp i,n}^{\ j,j'} = \mathbf{q}_{\perp i}^{\ j} = \begin{bmatrix} x_i^j - x_n^{j'} \\ y_i^j - y_n^{j'} \\ x_n^{j'} \cdot (x_n^{j'} - x_i^j) + y_n^{j'} \cdot (y_n^{j'} - y_i^j) \end{bmatrix}^\top.$$

Based on this, let $\mathbf{Q}_{\perp 1\mapsto n}^{\ j}$ be the matrix of the normal motion field for the j-th point, defined as

$$\mathbf{Q}_{\perp 1\mapsto n}^{\ j} = \left[\mathbf{q}_{\perp 1}^{\ j}, \ldots, \mathbf{q}_{\perp i}^{\ j}, \ldots, \mathbf{q}_{\perp n-\delta}^{\ j}\right]^\top$$

Therefore the normal global vector is as follows

$$\mathbf{v}_{\perp 1\mapsto n}^{\ j} = \mathbf{Q}_{\perp 1\mapsto n}^{\ j\top} \mathbf{N}_{1\mapsto n}^{j\top}.$$

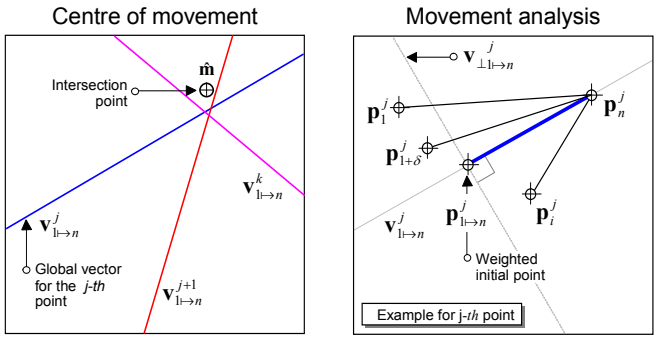


Fig. 4. (a) Multiple lines converge in one point when is performing a reach-to-grasp movement, (b) Once established a central point, a movement analysis toward that point is performed

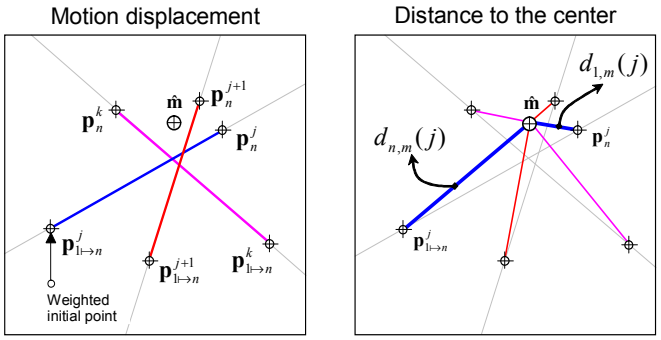


Fig. 5. Analysis of multiple points. (a) Motion of each initial and final trajectory point, (b) Distance to the intersection point

Note that $\mathbf{v}_{\perp}{}^{j}_{1\mapsto n}$ was computed in the same way as $\mathbf{v}^{j}_{1\mapsto n}$, however in this case the matrix $\mathbf{Q}_{\perp}{}^{j}_{1\mapsto n}$ is composed by an array of normal vectors.

III. Intersection point

For the sake of simplicity, the last procedure has considered the motion of the j-th point. We now turn to the problem of estimating the intersection point of multiple points in correspondences. Suppose we have determined multiple $\mathbf{v}^{\Theta}_{1\mapsto n}$ vectors, where $\Theta = \{1, \ldots, j, \ldots, k\}$ is the set of interest points detected between time $t = 1$ and $t = n$ and k is the last point in correspondence, as is shown in Fig.4a. For this, let $\mathbf{A}^{\Theta}_{1\mapsto n}$ be a $(k \times 3)$ matrix that encodes all motion vectors as

$$\mathbf{A}^{\Theta}_{1\mapsto n} = \left[\mathbf{v}^{1}_{1\mapsto n}, \ldots, \mathbf{v}^{j}_{1\mapsto n}, \ldots, \mathbf{v}^{k}_{1\mapsto n}\right]^{\top}.$$

The next step is to estimate a central point using vectors contained in $\mathbf{A}^{\Theta}_{1\mapsto n}$. Experimentally, when a reach-to-grasp movement has been initiated, multiple vectors cross over in one common point, called *intersection point*. To estimate the position of the unknown intersection point, we formulate a nonhomogeneous system of linear equations, described as follows

$$\underbrace{\begin{bmatrix} \mathbf{A}^{\Theta}_{1\mapsto n} \\ 0 \quad 0 \quad 1 \end{bmatrix}}_{\mathbf{H}} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{0_{k\times 1}} \\ 1 \end{bmatrix}}_{\mathbf{b}}. \tag{1}$$

Changing the notation in matrix terms, (1) can be expressed as $\mathbf{Hm} = \mathbf{b}$, where $\mathbf{H}$ is the over determined matrix coefficients of $\mathbf{A}^{\Theta}_{1\mapsto n}$ vectors; because $k \geq \rho$; and $\mathbf{m} = [x, y, 1]^{\top}$ is the vector of unknowns (x, y). To resolve this problem we use the $\mathbf{QR}$ transformation because it is numerically more stable [16]. Therefore, the solution for the nonhomogeneous system, using the $\mathbf{QR}$ transformation is

$$\hat{\mathbf{m}} = \mathbf{R}^{-1}\left(\mathbf{Q}^{\top}\mathbf{b}\right). \tag{2}$$

Then, we seek to compute the *normal intersection point* defined as the intersection of all normal vectors $v_{\perp}{}^{\Theta}_{1\mapsto n}$. Based on the above procedure, firstly we define the matrix $\mathbf{A}_{\perp}{}^{\Theta}_{1\mapsto n}$ of all normal vectors contained in Θ as

$$\mathbf{A}_{\perp}{}^{\Theta}_{1\mapsto n} = \left[\mathbf{v}_{\perp}{}^{1}_{1\mapsto n}, \ldots, \mathbf{v}_{\perp}{}^{j}_{1\mapsto n}, \ldots, \mathbf{v}_{\perp}{}^{k}_{1\mapsto n}\right]^{\top}.$$

Finally, the problem of estimating the normal intersection point can be expressed as $\mathbf{H}'\mathbf{m}_{\perp} = \mathbf{b}'$, where $\mathbf{m}_{\perp}$ is a non-homogenous vector that encodes the intersection point of normal vectors. Again, using the $\mathbf{QR}$ transformation applied to the matrix $\mathbf{H}'$, such as $\mathbf{H}' = \mathbf{Q}'\mathbf{R}'$, the normal intersection point is

$$\hat{\mathbf{m}}_{\perp} = \mathbf{R}'^{-1}\left(\mathbf{Q}'^{\top}\mathbf{b}'\right). \tag{3}$$

IV. Featured extracted

Below is an explanation of eight features proposed to predict different hand motions. Namely, approaching, distancing, rotational and translational invariant movements. Recall that in this stage we are not interested in detecting the object by itself nor detecting grasp movements.

a. *Grasp motion*: The first two features proposed are related with the grasp action. In general, the grasp motion can be split up in two different events. *Zoom-in*: when the hand is going to reach an object; and *Zoom-out*: when the hand is moving away of an object. Here, we propose a simple procedure to infer whether a hand is reaching an object or not based on the intersection point estimated in (2), and the motion transition along the TSW.

Firstly, let $\mathbf{P}^{j}_{1\mapsto n}$ be a $(inliers \times 3)$ matrix representing the 2D position in time $[1, \ldots, n]$ for each δ-frames; computed in the same way of matrix $\mathbf{Q}^{j}_{1\mapsto n}$.

$$\mathbf{P}^{j}_{1\mapsto n} = \left[\mathbf{p}^{j}_{1}, \ldots, \mathbf{p}^{j}_{i}, \ldots, \mathbf{p}^{j}_{n}\right]^{\top}.$$

Namely, the matrix $\mathbf{P}^{j}_{1\mapsto n}$ codes the motion of the j-th point until the last $(n - \delta)$ frame. Then, we re-map motion points taking into account variations in its features matching as

$$\mathbf{p}^{j}_{1\mapsto n} = \mathbf{P}^{j\top}_{1\mapsto n}\mathbf{N}^{j\top}_{1\mapsto n}, \tag{4}$$

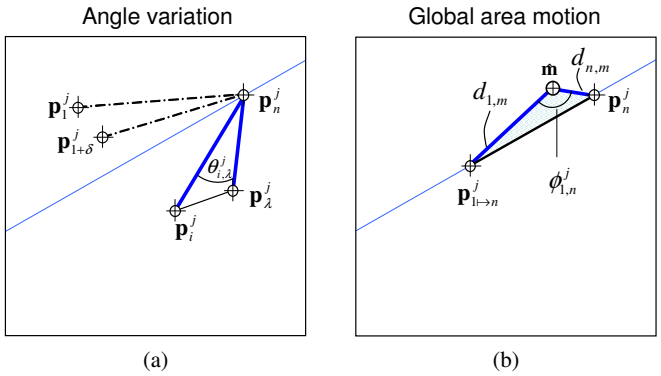


Fig. 6. (a) Temporal angle variation, (b) Global area motion between weighted mean position and last point contained in each time-window.

where $\mathbf{p}^j_{1\mapsto n}$ is a weighted mean position within vector $\mathbf{v}^j_{1\mapsto n}$ as is illustrated in Fig.4b. Extending this procedure for all Θ-points, let $\mathbf{p}^{\Theta}_{1\mapsto n}$ be the motion of each points in the TSW $[1,\ldots,n]$, and let $\mathbf{p}^{\Theta}_n$ be the final position of each point, defined as,

$$\mathbf{p}^{\Theta}_{1\mapsto n} = \left[\mathbf{p}^1_{1\mapsto n},\ldots,\mathbf{p}^j_{1\mapsto n},\ldots,\mathbf{p}^k_{1\mapsto n}\right]^\top$$

and

$$\mathbf{p}^{\Theta}_n = \left[\mathbf{p}^1_n,\ldots,\mathbf{p}^j_n,\ldots,\mathbf{p}^k_n\right]^\top$$

Since vector $\mathbf{p}^{\Theta}_{1\mapsto n}$ codes the initial weighted position, let d_1 be the Euclidean distance of each vector $\mathbf{p}^{\Theta}_{1\mapsto n}$ in relation with the intersection point $\hat{\mathbf{m}}$, and let d_n be the Euclidean distance of each final position $\mathbf{p}^{\Theta}_n$ in relation with same intersection point $\hat{\mathbf{m}}$ as,

$$d_{1,m}(j) = \left\|\mathbf{p}^{\Theta}_{1\mapsto n}(j) - \hat{\mathbf{m}}\right\|, \quad d_{n,m}(j) = \left\|\mathbf{p}^{\Theta}_n(j) - \hat{\mathbf{m}}\right\| \quad (5)$$

The Euclidean distance $d_{1,m}$ and $d_{n,m}$ represent the temporal movement around the intersection point $\hat{\mathbf{m}}$, as shown in Fig.5b. Since we know the estimated position of the initial, final, and intersection point, the next step is to determine whether motion is toward the center or not. Based on these values, we define the function $v(j)$, as the number of nearest points to the intersection point, as follows,

$$v(j) = \begin{cases} 1 & \text{if } d_{n,m}(j) \geq d_{1,m}(j) \\ 0 & \text{otherwise.} \end{cases}$$

Finally, from $v(j)$, we extract the mean f_1 and the second central moment f_2, as follows

$$f_1 = \mu(v) \quad (6)$$
$$f_2 = \sigma^2(v), \quad (7)$$

where $\mu(\cdot)$ is the mean and $\sigma^2(\cdot)$ is the variance. The above features indicate that the movement is toward an object if $f_1 \mapsto 1$; and conversely, the movement is against an object if $f_1 \mapsto 0$. To confirm this prediction, the variance σ^2 should be low in any case.

b. *Rotational motion*: The rotational motion feature gives a temporal variation of each point in correspondence. The main idea is to capture rotational movements independently of its turn direction, and thus, to compute the angle velocity of each point.

Firstly, suppose that the link between $\mathbf{p}^j_i \mapsto \mathbf{p}^j_n$ and $\mathbf{p}^j_\lambda \mapsto \mathbf{p}^j_n$ exists. Therefore, s^j_i and s^j_λ are two consecutive slopes of the j-th point separated by λ-frames respectively, defined as,

$$s^j_i = \frac{y^j_i - y^j_n}{x^j_i - x^j_n}, \qquad s^j_\lambda = \frac{y^j_\lambda - y^j_n}{x^j_\lambda - x^j_n}.$$

Since both points are pointing to the last point p^j_n in time $t = n$, as depicted in Fig. 6a; therefore, by transitivity, also implies that $\mathbf{p}^j_i \mapsto \mathbf{p}^j_\lambda$, where $t_\lambda > t_i$. Thereby, the angle between these consecutive slopes is

$$\theta^j_{i,\lambda} = \arctan\left|\frac{s^j_i - s^j_\lambda}{1 + s^j_i s^j_\lambda}\right|,$$

Based on this result, we calculate the angular velocity ω between p^j_i and p^j_λ so as to compute the motion variation along the time, defined as

$$\omega^j_{i,\lambda} = \frac{\triangle\theta^j_{i,\lambda}}{\triangle t_{i,\lambda}},$$

for all $i = 1,\ldots,inliers$; where $\triangle t_{i,\lambda}$ is the time difference between two consecutive frames. Clearly, the angular velocity ω is a useful feature to estimate the motion rate of each point along the TSW. Combining the above value with the Euclidean distance between points $\mathbf{p}^j_i$ and $\mathbf{p}^j_\lambda$ we propose an invariant feature that distinguish rotational and translational movements as follows

$$f_3 = \frac{\sum_{j=1}^{k}\sum_{i=1}^{inlier}\sigma^2(\omega^j_{i,\lambda})}{\sum_{j=1}^{k}\sum_{i=1}^{inlier}\sigma^2(\|\mathbf{p}^j_i - \mathbf{p}^j_\lambda\|)}, \quad (8)$$

The feature f_3 tends to zero when the movement is translational. This happens when the hand is moving constantly in the same direction; independently of its angle direction. Thus, the variance of the Euclidean distance is high and the variance of the angular velocity is low. Conversely, when motion is rotational, f_3 tends to be greater than one. Accordingly, the variance of the Euclidean distance tends to be low as well as the variance of the angular velocity, since every point describes the same angular rotation.

c. *Rotational area*: In the same line as the above feature, we propose to compute the area covered by central intersection point (2), the weighted mean position (4) and the final end position of each point as a measure to compute variations of the area along the time, as shown in Fig.6b. More formally, let $\phi^j_{1\mapsto n}$ be the angle between $\mathbf{p}^j_{1\mapsto n}$ and p^j_n on point $\hat{\mathbf{m}}$. The log-area variation of multiple points along the TSW is as follows

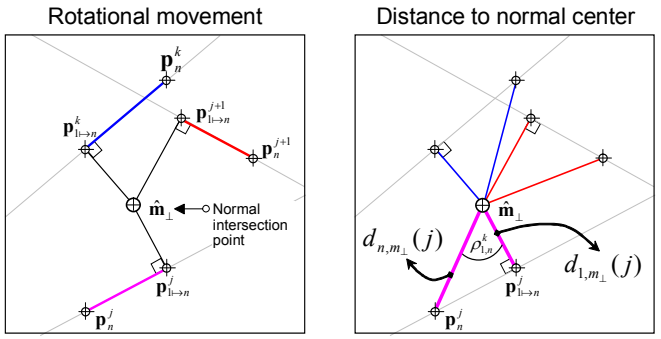


Fig. 7. Analysis of multiple points. (a) Motion of each initial and final trajectory point respect to the intersection point, (b) Distance to the normal intersection point

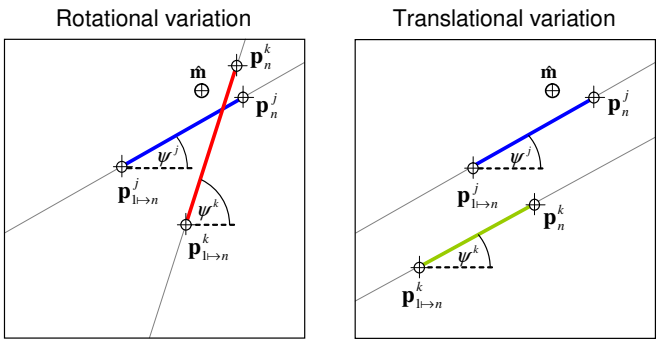


Fig. 8. (a) Different angles are found when the motion is rotational (b) Similar angles are found when the motion is purely translational

$$f_4 = \log\left(\frac{1}{2k}\sum_{j=1}^{k} d_{1,m}(j)d_{n,m}(j)\sin(\phi^j_{1,n})\right) \tag{9}$$

where $d_{1,m}(j)$ and $d_{n,m}(j)$ are the Euclidean distance with respect to the point $\hat{\mathbf{m}}$ of the j-th point, estimated previously in (5) contained in Θ. The above feature computes the relative area between the camera and the hand. In general, its variations along the time is a useful way to estimate if motion is toward an object or not. Herein, we compute log-area so as to reduce its scale variation.

c. ***Rotational Normal variation*****:** When movement is purely rotational, the intersection point $\hat{\mathbf{m}}$ does not represent the real center of motion. In a similar way to the last feature, firstly we propose to compute the Euclidean distance to the normal intersection point $\hat{\mathbf{m}}_\perp$, defined as,

$$d_{1,m_\perp}(j) = \|\mathbf{p}^{\Theta}_{1\mapsto n}(j) - \hat{\mathbf{m}}_\perp\|, d_{n,m_\perp}(j) = \|\mathbf{p}^{\Theta}_{n}(j) - \hat{\mathbf{m}}_\perp\| \tag{10}$$

where $d_{1,m_\perp}(j)$ and $d_{1,m_\perp}(j)$ are the temporal distance of the j-point around the normal intersection point $\hat{\mathbf{m}}_\perp$. Secondly, we compute the angle $\rho^j_{1,n}$ between $\mathbf{p}^j_{1\mapsto n}$ and $\mathbf{p}^j_n$ in regard to the point $\hat{\mathbf{m}}_\perp$, as shown in Fig.7. Using the above values, we compute the log-area of the rotational normal movement as,

$$f_5 = \log\left(\frac{1}{2k}\sum_{j=1}^{k} d_{1,m_\perp}(j)d_{n,m_\perp}(j)\sin(\rho^j_{1,n})\right) \tag{11}$$

Normally this variation is high when motion is not rotational because the intersection of normal vectors does not exist. However, when motion starts to be rotational there is a point $\hat{\mathbf{m}}$ that intersects all normal vectors $v^{\Theta}_{\perp 1\mapsto n}$. Consequently, all points have the same spin angle and a similar variation. Experimentally, this rotational detector can differentiate lineal movements from rotational movements.

A consequence of the above result is that we have obtained two angle variations. Namely the angle variation $\rho^j_{1,n}$ for the j-point with regard to the normal intersection point $\hat{\mathbf{m}}_\perp$, and the angle variation $\phi^j_{1,n}$ with regard to the intersection point $\hat{\mathbf{m}}$. Combining both angles in one feature allows us to get a variation of motion over time, defined as follows

$$f_6 = \frac{\sum_{j=1}^{k}\phi^j_{1,n}}{\sum_{j=1}^{k}\rho^j_{1,n}} \tag{12}$$

For example, for rotational movements, f_6 remains constant with a low value along the time. In case of lineal movements, f_6 tends to be high, and finally, for zoom in or zoom out movements, f_6 varies according to each movement growing or declining respectively.

d. ***Parallel angles*****:** The parallel angles gives the relative variation between the angles of each weighted mean position and its final end position, as is illustrated in Fig.8. The key point of this feature is to detect only translational movements, independent of its angle direction and orientation of the movement. Firstly, we compute the angle variation ψ^j of each j- points as,

$$\psi^j = \arctan\left|\frac{y^j_{1\mapsto n} - y^j_n}{x^j_{1\mapsto n} - x^j_n}\right|$$

where $\mathbf{p}^j_{1\mapsto n} = [x^j_{1\mapsto n}, y^j_{1\mapsto n}]$ is the weighted mean position described previously, and $\mathbf{p}^j_n = [x^j_n, y^j_n]$ is the final end position. Combining the above angle and the Euclidean distance, we compute the parallel variation as follows

$$f_7 = \log\left(\frac{\sum_{j=1}^{k}(\|\mathbf{p}^j_{1\mapsto n} - \mathbf{p}^j_n\|)}{k\sigma^2(\psi)}\right) \tag{13}$$

In contrast to the feature f_3, the above feature tends to zero when motion is rotational and is only high when motion is purely translational, because the angle variation is very low and the distance of each weighted point remains constant. In all other cases, the angle variation is high, and the Euclidean distance varies according to the type of movement.

e. ***Acceleration*****:** The last feature computes the acceleration of each corresponding point encoded in the vector $\mathbf{P}^{\Theta}_{1\mapsto n}$.

Unlike the time $\triangle t_{i,\lambda}$ relates the time different between two consecutive frames; here we compute the time difference in regard to the last temporal frame $\mathbf{p}_n^{\Theta}$ for each point contained in the set Θ; in other words, we compute $\triangle t_{i,n}$ so as to normalize the velocity to a single unit of time. For instance, let v_x^j and v_y^j be the temporal velocity respect to point $\mathbf{p}_n^j$ taking into account the temporal difference $t_{i,\lambda}$ as follows

$$v_x^j(i) = \frac{x_n^j - x_i^j}{\triangle t_{i,n}} \qquad v_y^j(i) = \frac{y_n^j - y_i^j}{\triangle t_{i,n}} \tag{14}$$

Based on the above results, the acceleration of the j-th point in time $t = i$ is defined by

$$a_x(i) = \frac{\sum_{j=1}^{k} v_x^j(i) - \sum_{j=1}^{k} v_x^j(i-\lambda)}{k\triangle t_{i,\lambda}} \tag{15}$$

$$a_y(i) = \frac{\sum_{j=1}^{k} v_y^j(i) - \sum_{j=1}^{k} v_y^j(i-\lambda)}{k\triangle t_{i,\lambda}} \tag{16}$$

In the above case, we compute the time $\triangle t_{i,\lambda}$ because we seek the relative acceleration between consecutive frames. Based on these results, we propose the following feature to quantify the global acceleration as

$$f_8 = \frac{\sigma^2(a_x)}{\sigma^2(a_x) + \sigma^2(a_y)}\,, \tag{17}$$

where a_x and a_y are a two vector containing the relative acceleration from each slide window.

f. *Feature vector*: In the previous steps we have proposed eight features descriptors that encode different aspects of point motion. Namely rotational acceleration, linear acceleration, angle variation, area variation and motion direction. These features are later used as an input for the HMM system as shown in the following section.

For simplicity, the above analysis has considered a TSW in time $[t = 1, \ldots, t = n]$. Thus, the first feature vector $\mathbf{o}_1$ is composed as follows,

$$\mathbf{o}_1 \equiv \mathbf{o}_{1\mapsto n} = [f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8]^\top\,, \tag{18}$$

nevertheless, to infer the user's intention it is necessary to get multiple TSWs. Recall that each TSW is composed by a sequence of δ frames, as shown in Fig.2. Therefore, a sequence is represented by a sequence of slide windows, each one composed by eight features.

$$\mathbf{O} = [\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_T]\,, \tag{19}$$

where T is the total frame number of the video sequence and $\mathbf{O}$ is the observed symbol sequence.

V. Training HMM for recognition

Below, we briefly describe the principal component of an HMM based system used for recognize the user's intentions. For a comprehensive review we refer to Rabiner [17]. HMM is a type of stochastic signal model composed by a Markov Chain whose states cannot be observed directly, but can be observed through the sequence of observations. Currently, HMMs have been employed in a wide range of applications. Specially in those where it is necessary to deal with time-series with spatial temporal variabilities, as for example, intention and gesture recognition [10–12, 18].

More formally, HMM is composed by a number of N-states $\{S_1, S_2, \ldots, S_N\}$ connected by transitions, where each transition has associated a probability, defined by matrix A; an emission distribution probability, or the probability of emitting an observation given a state, defined by matrix B; and an initial state distribution $\pi = \{\pi_i\}$. That is, using a compact notation a HMM is fully specified by the triplet $\lambda = (A, B, \pi)$ where:

- $A = \{a_{ij}\}$ where $a_{ij} = Pr(q_{t+1} = S_j | q_t = S_i), 1 \leq i, j \leq N$ is the state transition probability distribution, and q_t represent the state at time t.
- $B = \{b_1(\mathbf{o}), b_2(\mathbf{o}), \ldots, b_N(\mathbf{o})\}$ correspond to the observation probability for each state. In our problem, observations are modeled with a Gaussian distribution $b_j(\mathbf{O}) = N(\mathbf{o}, \mu_j, \sigma_j)$ where $\mathbf{o}$ is the feature vector extracted in the last step.
- $\Pi \equiv \{\pi_1, \pi_2, \ldots, \pi_N\}$ where $\pi_i = p(q_1 = S_i), 1 \leq i \leq N$ is the initial state distribution.

Based on the above parameters, the problem is to classify each class defined as a particular user's intention. Firstly, we create a HMM for each category using the well known Forward-Backward algorithm [17] in order to find the best parameters for each HMM. This is a generalized Expectation-Maximization (EM) algorithm by maximizing the probability of observation sequence given each HMM model for all training sequences.

VI. Prediction HMM for recognition

Once established the HMM parameters, our goal is to recognize an observed symbol sequence as a particular class or user's intention. Suppose that each λ_i where $i = 1, \ldots, C$, is a model parameter defined for i-class on C classes. Given a sequence of observations $\mathbf{O}$, we calculate $p(\mathbf{O}|\lambda_i)$ for each HMM λ_i and we choose the class with the maximum probability as:

$$class = arg\max_i(p(\mathbf{O}|\lambda_i)). \tag{20}$$

IV. Experimental results

In our experiments we have defined four movements independently of each object involved in the sequence such as: zoom in, zoom out, lineal and rotary movements. Since each movement is valid in a sequence of frames, our goal is to detect if each movement has been correctly predicted as the real movement performed by an user. In this experiment we use a HMM system for predicting user's intentions. In order to build a HMM we performed an action several times using one object in the scene. The goal is that providing more testing sequences, for each class, we can increase the probability to classify correctly an unknown sequence. In our experiments we employed video sequences at 30 fps

digitalized into 320x200 pixel with 256 gray-level images. An example of the video sequence is shown in Fig.9.

To evaluate the performance, we consider that an action is correct if motion contained in each TSW has been predicted correctly. Additionally, the system must be independent of objects contained in the scene. In general, the performance of a HMM varies according to the data used for testing. Therefore, in our experiments we used the cross-validation method with $k = 10$. Here we are not interested on evaluating the performance of the testing data used for training the HMM. Our goal is to evaluate the performance in videos with other objects. For this reason we have tested each HMM on five different objects performing each particular action with one object at once. Namely a cup, bottle, mug, box, deodorant. As we will show later, a HMM can be useful to predict the movement in sequences even with multiple objects.

Our solution uses TSWs with the aim of analyzing the temporal motion contained in this period. Using this idea we can reduce the number of frames analyzed about 67%, because each TSW is interspaced by δ-frames with $\delta = 3$. As shown in Table I, 7131 TSWs were analyzed from 21544 frames of five video sequences. In order to evaluate its performance of the HMM trained, we have classified manually each of 7131 TWSs. With respect to the data for training, we have classified 1466 TSWs from a mug without markers on the surface.

The performance of each HMM using different training sets shows that the Zoom-out movement has, in average the best performance near to 90%, as shown Fig.10. Also, the lower performance has been detected in the Zoom-in movement, because it is normally incorrectly classified as a rotary movement. In the same line, this performance can vary according to the object analyzed. For example, in the case of the bottle used for testing, the performance was lower because the SURF algorithm was unable to detect a large number of descriptors. Therefore, fewer descriptors can not be able to build a robust TSW. On the other hand, the mug analyzed had the best performance because a large number of descriptors were detected (Fig.11b).

In our experiments we also used the best HMM generated with the cross validation method. For this task, we have selected the best performance of each action taking as a criterion the best F-Score and $\mathbf{TPR}$ performance. The results shows that we can increase the performance by 2% with the best F-Score and over 4% with the best $\mathbf{TPR}$, as shown Fig.11b-c.

V. Conclusions

The main contribution of this work lies into perform a motion prediction using only a hand motion estimation. This results can be applied on the project BRAHMA as a method to predict the user's intention. Specially on people with neural degenerative disorders. In these, the control movements are altered causing motion termor, slow movements, etc. Despite the visual functions on these people have not been altered, the control system can not be able to plan a correct movement without any disruption. In our experiments we have shown that it is possible to detect hand intentions using only the objects contained in the scene, and without special markers on the object surface. However, the performance of this task varies according to the points of interest detected by the SURF method. Even if the object has been occluded, the system is able to detect it because our approach uses a combination of frames called Temporal Slide Windows (TSW). This approach allows us to increment the temporal features of the same object on time, therefore the paradigm of frame-by-frame comparison has been replaced by TSW-by-frame approach. Our future work is to improve the performance of the grasp intention by improving the matching step. This task can be conducted by searching a matching between both views.

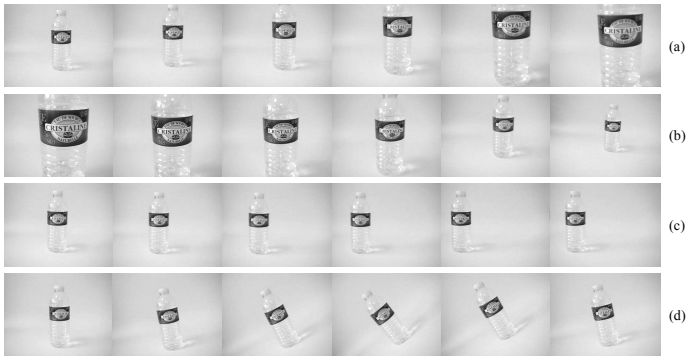


Fig. 9. Real image sequence with one object performing four actions(a) zoom in, (b) zoom out, (c) lineal and (d) rotary movements

TABLE I
TSW ANALYZED OVER EACH HAND MOTION VIDEO

Object	Zoom in-out		Rotation motion		Lineal motion	
	Frames	TSW	Frames	TSW	Frames	TSW
Cup	1876	622	1226	405	1051	347
Bottle	1894	628	1211	401	726	240
Mug II	2231	739	1221	404	1016	336
Box	2393	792	1209	400	942	312
Deodorant	2414	799	1200	397	934	309
$\sum$	10808	3580	6067	2007	4669	1544
Mug I (†)	2292	759	1208	400	928	307

†: training data

Acknowledgements

We gratefully acknowledge partial funding by CONICYT – Colegio Doctoral Franco-Chileno, grant no. 21050185.

References

[1] J. Crawford, W. Medendorp, and J. Marotta, "Spatial transformations for eyehand coordination," *Journal of Neurophysiology*, vol. 92, pp. 10–19, 2004.

[2] J. Flanagan and S. Lederman, "Neurobiology: Feeling bumps and holes," *Nature*, vol. 412, pp. 389–391, 2001.

[3] M. Hayhoe, D. Bensinger, and D. Ballard, "Task constraints in visual working memory," *Vision Research*, vol. 38, pp. 125–137, 1998.

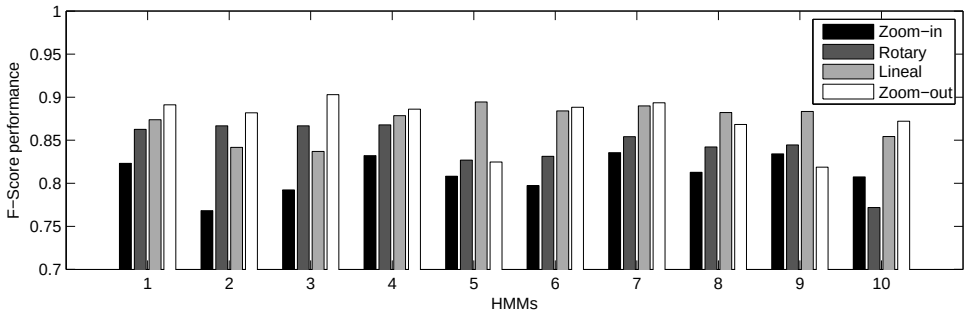
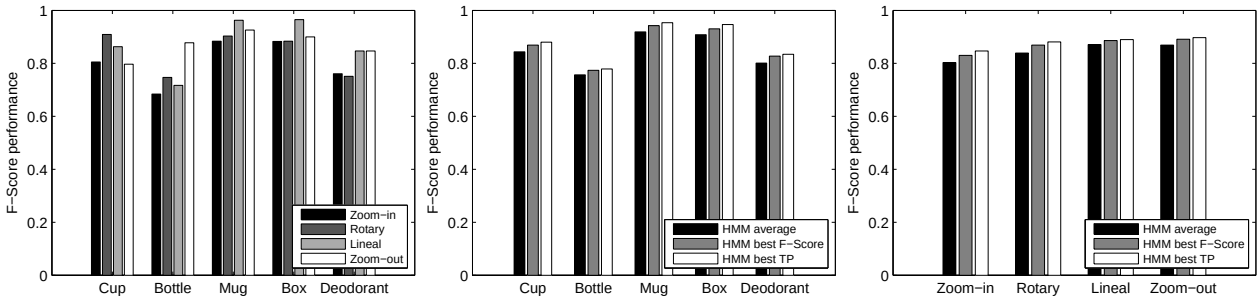


Fig. 10. Average performance of the F-Score over ten HMMs using five objects



(a) (b) (c)

Fig. 11. (a) Average performance for each action using all HMMs; (b) Average performance for all actions on each object using three HMM parameters; (c) Average performance for all objects on each action using three HMM parameters

[4] A.-M. Brouwer and D. C. Knill, "The role of memory in visually guided reaching," *Journal of Vision*, vol. 7, no. 5, pp. 1–12, 6 2007.

[5] J. Aggarwal and Q. Cai, "Human motion analysis: A review," *Computer Vision and Image Understanding*, vol. 73, pp. 90–102, 1997.

[6] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: a review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 4–37, Jan. 2000.

[7] R. Polana and R. Nelson, "Low level recognition of human motion (or how to get your man without finding his body parts)," in *Proc. IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, 11–12 Nov. 1994, pp. 77–82.

[8] A. Bobick and J. Davis, "Real-time recognition of activity using temporal templates," in *Proc. 3rd IEEE Workshop on Applications of Computer Vision WACV '96*, 2–4 Dec. 1996, pp. 39–42.

[9] E. Shechtman and M. Irani, "Space-time behavior based correlation," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR 2005*, vol. 1, 20–25 June 2005, pp. 405–412.

[10] J. Yamato, J. Ohya, and K. Ishii, "Recognizing human action in time-sequential images using hidden markov model," in *Proc. CVPR '92. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 15–18 June 1992, pp. 379–385.

[11] T. Starner and A. Pentland, "Real-time american sign language recognition from video using hidden markov models," in *Proc. International Symposium on Computer Vision*, 21–23 Nov. 1995, pp. 265–270.

[12] C. Achard, X. Qu, A. Mokhber, and M. Milgram, "Action recognition with semi-global characteristics and hidden markov models," in *Conference on Advanced Concepts for Intelligent Vision Systems (ACIVS)*, 2007.

[13] H. Bay, T. Tuytelaars, and L. Gool, "Surf: Speeded up robust features," in *Proceedings of the 9th European Conference on Computer Vision*, May 2006.

[14] J. Barron, D. Fleet, and S. Beauchimen, "Performance of optical flow techniques," *International Journal of Computer Vision*, vol. 12, no. 1, pp. 43–77, 1994.

[15] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[16] N. Higham, *Accuracy and Stability of Numerical Algorithms*. SIAM, 1996.

[17] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[18] Y. He and A. Kundu, "Shape classification using hidden markov model," in *Proc. International Conference on Acoustics, Speech, and Signal Processing ICASSP-91*, 14–17 April 1991, pp. 2373–2376.