Image and Vision Computing 30 (2012) 860–874

Contents lists available at SciVerse ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Exploiting eye–hand coordination to detect grasping movements ☆

Miguel Carrasco [a,*], Xavier Clady [b]

[a] Escuela de Informática y Telecomunicaciones, Facultad de Ingeniería, Universidad Diego Portales, Vergara 432, Santiago, Chile
[b] Vision Institute, University Pierre and Marie Curie-UPMC, INSERM UMR S968, CNRS UMR7222, Paris, France

ARTICLE INFO

Article history:
Received 9 September 2011
Received in revised form 31 March 2012
Accepted 1 July 2012

Keywords:
Visual system
Grasping movements
Motion analysis
Hand posture
Hand gesture
Object recognition

ABSTRACT

Human beings are very skillful at reaching for and grasping objects under multiple conditions, even when faced with an object's wide variety of positions, locations, structures and orientations. This natural ability, controlled by the human brain, is called eye–hand coordination. To understand this behavior it is necessary to study both eye and hand movements simultaneously. This paper proposes a novel approach to detect grasping movements by means of computer vision techniques. This solution fuses two viewpoints, one viewpoint which is obtained from an eye-tracker capturing the user's perspective and a second viewpoint which is captured by a wearable camera attached to a user's wrist. Utilizing information from these two viewpoints it is possible to characterize multiple hand movements in conjunction with eye-gaze movements through a Hidden–Markov Model framework. This paper shows that combining these two sources makes it possible to detect hand gestures using only the objects contained in the scene even without markers on the surface of the objects. In addition, it is possible to detect which is the desired object before the user can actually grasp said object.

1. Introduction

Human movement analysis is an area of study that has been quickly expanding over the past few years. Progress in analyzing image sequences, the evolution of computer systems and the miniaturization of technology used to capture movement, have made motion analysis applications possible in areas such as athletic performance analysis, surveillance, man–machine interfaces, entertainment systems, video-games and robot-based rehabilitation therapy, among others [54,36,46,41,53,17].

On the human gesture level, the majority of research has been conducted around the analysis of gesture language, and in particular, sign language which is probably the most common. Among certain industrial sectors, sign language composed of limb and facial movements is also used to build human–computer interfaces. However, communicative gestures are only one small part of the wide range of gestures made by humans. Within the taxonomy of gestures, we will distinguish grasping movements. These gestures have not been studied extensively in the computer vision field, as they are movements that occur in human interaction with their environment. Therefore, it is necessary to include perception and knowledge of the environment within the parameters of the system. The analysis of human movement is already a complex issue. In fact, accurately collecting and analyzing human body movement are very complicated and difficult. Without even considering human body segmentation and motion estimation in image sequences, the recognition system must solve many inherent difficulties that come with gestures. These difficulties include variability of the signal in time (a gesture's direction and dynamics vary with each execution, even if made by the same person), variability of the signal in space (just as a gesture may vary during each execution, it also varies in each dimension of space) and temporal segmentation or macro segmentation (to recognize a specific sequence of gestures it is necessary to segment each sequence temporarily in order to effectively study every individual gesture). Each human gesture belongs to a particular context. Gesture execution links to its intrinsic nature (shape, size) and the object's scope with which people interact as well as its extrinsic characteristics (position and orientation relative to the object in relation to the operator). Therefore, determining a correct representation adapted to each type of gesture and a decision process by macro segmentation as well as recognition of that gesture, constitutes an important focus of study. To achieve this goal, a multimodal analysis is required to simultaneously analyze the reconstruction, monitoring, and recognition of the hand position, eye tracking and recognition of objects (estimate the position and search for real objects). This approach is particularly effective because knowledge of the object in hand during a gesture allows researchers to better infer gesture recognition of grasping and manipulation gestures [20,40,30] as well as reduce the inherent difficulties related to an elevated number of degrees of freedom (models of the hand and forearm have 26° of freedom) and self-occlusion [16].

☆ This paper has been recommended for acceptance by Daniel Gatica-Perez.

* Corresponding author. Tel.: +56 26768134.
E-mail addresses: miguel.carrasco@udp.cl (M. Carrasco), xavier.clady@upmc.fr (X. Clady).

0262-8856/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2012.07.001

In general, the main contribution in this area has been limited to the external analysis of human movement. We propose building a model that utilizes the same visual information the user has access to by capturing said visual field with a camera. The main research objective is the automatic translation of gestures created by grasping objects. To that end, it is necessary to identify each kind of movement made in addition to indicating the object a user wants to grasp. Why is it necessary to determine this type of movement? One example of its importance is for people suffering from neurodegenerative diseases, which cause motor problems and limit movement, who are greatly hindered in performing grasping tasks. In these cases motion control is altered, causing tremor, slowness, imprecision, etc. Even though visual functions may not be affected, the control system is unable to plan the motion in a normal manner.

This paper proposes recognizing user grasping movements by fusing the analysis of multiple devices attached to the human body. The investigation differs from classical methods utilized to recognize grasping gestures. Generally, most motion recognition methods capture user actions by tracking body parts from a position in front of the user. Instead, we propose to capture the scene by utilizing the user's gaze and grasping movements by exploiting the user perspective; thus, making it possible to infer the user's action. The system is composed of (1) an eye-tracker with an integrated camera that captures a scene similar to the user's field-of-view (FOV) and estimates the user's gaze position; and (2) a camera placed on the user's wrist that captures a scene similar to that of the eye-tracker camera (Fig. 1).

The rest of the paper is organized as follows: Section 2 discusses prior work on human gesture recognition; Section 3 explains the proposed method; Section 4 shows experimental results; and finally, Section 5 presents the paper's contributions and succinctly describes various ongoing and future works.

2. Related work

Gesture recognition can be defined as a problem in tracking body parts over space-time in order to interpret motion behavior as a particular gesture. Based on the Aggarwal and Cai [3] definition, gesture recognition requires the performance of three general tasks. First, to identify 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 specific human action by matching the motion descriptor captured in the tracking process against the recognition framework. The last step is considered a higher level task given that the recognition task requires the classification of varying feature data over time [25]. The problem of interpreting human gestures is defined as a learning process. In the training phase, some sequences are used to learn the user's behavior, labeling each sequence as a particular human gesture. Later, in the matching phase, unknown test sequences are compared against a model so as to be classified as a particular gesture. Most approaches designed to detect human gestures are based on template matching (e.g. [35,44]) or appearance-based models [48,1]. This section discusses the current principal approaches to detect human gestures using computer vision methods. First, a brief introduction to the main paradigms for motion detection; second, a discussion of the eye–hand coordination involved in grasping recognition.

2.1. Computer vision methods that analyze human motion

The study of human motion using computer vision methods can be divided into three main approaches: passive, wearable and pointer paradigms which are relative to user and camera position. (1) With the passive approach, the camera is located in a fixed position, normally in front of the user, meaning the camera's field-of-view (FOV) remains constant. There are two main scenarios depending on whether the subject is captured with just one stationary camera or with multiple cameras from multiple perspectives in correspondence (see recent surveys [52,36]). (2) The wearable approach uses external devices attached to the human body. The objective is to obtain a continuous representation of the user's environment. At present, wearable cameras are offering new ways to increase human–computer interactions, mainly by allowing the user to move freely and view any given scene without being constrained by fixed cameras (e.g. [10,12,24,30]). (3) The pointer approach is based on the idea *where I am looking is what I want*. Currently, the device most employed to obtain a user's gaze is the eye-tracker. The eye-tracker allows researchers to track eye movements by giving an estimated position of the user's gaze, in real-time, relative to an image frame, normally after an initial calibration. The system is composed of two head-mounted cameras: (i) a camera that shares a similar view as the user. This camera has almost the exact same field-of-view (FOV) as the user and therefore answers the first part, *what I am looking at*; and (ii) a camera that captures eye movements by means of corneal reflection; thus covering the position, *what I want*. In general, this technology is providing new opportunities to understand visual perception from a cognitive perspective and to explain the inherent mechanisms that control eye–hand coordination. However, so far there is no clear consensus or a unified theory that can explain this process effectively (see [11] for detailed discussions). Therefore, it is not possible to use a specific model that explains the underlying procedure of eye–hand coordination.

Most methods which detect human gestures have been designed to utilize both passive and wearable approaches. These methods have proven to be effective in representing the action that takes place in the scene [13,10,27,24]; unfortunately, they cannot detect grasping movements because they do not take into account the direction of a user's gaze. On the other hand, even though the pointer approach has been designed to predict the user's gaze, it cannot differentiate a grasping motion unless the user maintains a constant gaze toward an object for a prolonged period of time. For example, the system proposed by [41] exploits the human gaze to support cooperative work with robots.

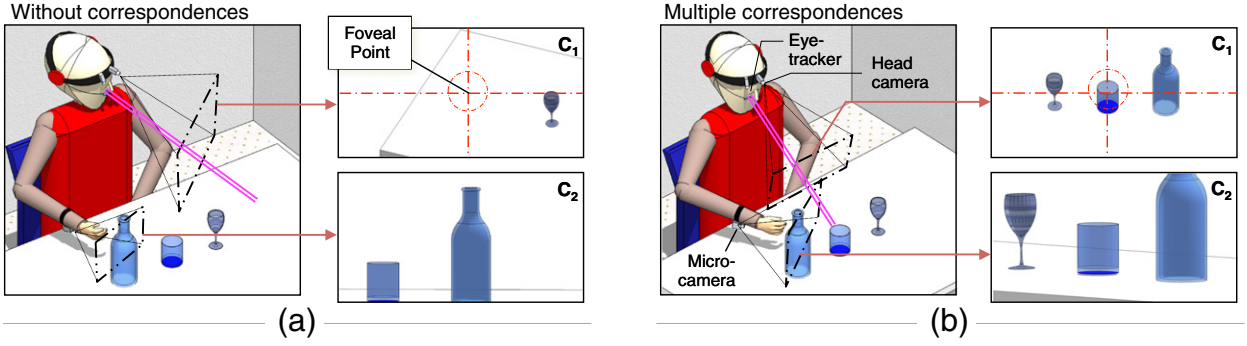

Fig. 1. User's posture when performing a grasping action using an eye-tracker and camera beneath the wrist.

Additional applications have also been designed to help people with motor difficulties, e.g. [34,37]. The main restriction of the aforementioned methods is that eye-trackers are not designed to analyze the trajectory of the hand toward an object because they have no vision of the hand. To overcome these drawbacks, this investigation develops a method that uses both wearable and pointer paradigms. The authors are unaware of any existing works that attempt to exploit eye–hand coordination by combining these two approaches with the final goal of predicting hand gestures.

2.2. Eye–hand coordination

Human beings posses a highly developed ability to grasp objects under many different conditions, taking into account variations in position, location, structure, motion and orientation. This natural ability controlled by the human brain is called *eye–hand coordination* [7]. Normally, a grasping movement is initiated before the hand actually reaches the desired object. This movement is regulated by the interaction of several sensorimotor systems such as the visual, vestibular and proprioceptive systems working in conjunction with the head, eye, hand and arm control systems [9,18].

The internal process that controls eye–hand coordination is complex and results from the multiple sensor-receptor, control and cognitive systems working in synergy [7,22]. This process is completely different from human motion as defined by Adams [2]. Human motion is strongly influenced by the cognitive process, and as a result, when movement becomes habitual, the cognitive process is used only to correct or perfect the movement. By contrast, eye–hand coordination requires a sensory signal mechanism that controls the eye and hand movements as a single unit. Such coordination demands three main brain tasks described as follows. Firstly, the brain must solve a geometric transformation between the internal world, encoded by the retinocentric frame of reference; and the external world, using a body-centered representation by proprioception [9,15]. Secondly, the brain must develop a plan to reach an object using body-centered coordinates by comparing gaze signals with hand coordinates and by estimating the hand motor difference in relation with the gaze coordinates [8]. Thirdly, the brain must control hand posture before reaching toward an object taking into account the size, shape, motion and orientation of the object [39].

For many years researchers have been studying this process trying to discover the underlying mechanism that controls eye–hand coordination. However, currently there is not a single theory that explains this process effectively and in fact, the process is not completely understood, e.g. [21,14]. The controversial question is how much information is utilized in the planning of a hand movement. More precisely, does human vision rely more on visual information or memory representations? In the nineties, many researchers supported the idea that only limited information is acquired across saccades [21,31,4]. Humans seem to maximize the coordination between eye and hand movements using visual information continuously instead of a memory representation to plan their movements [7]. The main reason being that memory is too old and uncertain even when nothing has changed; in contrast, visual information is constantly updated. However, recent studies have shown that people can use both systems simultaneously. Brouwer and Knill [7] stated that unconscious memory is used constantly to plan hand movements and focus on objects. They show that the brain can use both sources depending on their relative reliability. If visual information is more reliable than memory representations, the visual source seems to dominate. By contrast, when visual information is degraded, the brain increases the use of memory information to plan hand movements. That explains why people require more time to grasp an object in a situation of low contrast compared to high contrast conditions [7].

Another important issue concerning eye–hand coordination is related to gaze fixations. In general, there is a clear consensus that a gaze is directed at a specific target long before the hand reaches the object [7,9,22]. Likewise, fixations seem to be stable until the object has been grasped, when the hand arrives, fixations on the object are no longer required. As a consequence, the number of saccades around the object is fairly reduced, increasing the visual information in the retina [33]. This behavior indicates that fixations have three main features. First: task-dependent, different fixations are needed for performing different actions based on knowledge and target location [22]. Second: task-relevant, the sequence of fixations in relevant points allows the brain to estimate the geometric relationship of the world based on internal body coordinates [9,26]. Third: memory-dependent, fixations allow the brain to memorize different spatial positions of objects which can later be used when planning movements [7].

As stated above, the time needed to acquire a gaze fixation is directly related to the task context. It depends on the degree of complexity required to manipulate an object with the hand. Consequently, the period of time is variable and can fluctuate between 100 ms to 1500 ms, the general distribution is between 100 ms and 200 ms [22]. On the contrary, long fixations are the result of a prolonged action with continuous direction as stated by Land et al. [28]. This important finding is a key factor in understanding how the visual system works. Later, this understanding may provide insight into the grasping action process by increasing the probability of detecting the desired object.

3. Proposed method

The proposed system uses a temporal slide window (TSW) approach. Thus, by means of a pattern recognition scheme, it is possible to classify each motion pattern as a particular grasping movement. Nonetheless, there are two problems associated with this scheme. First, it is too complex to extract a good feature pattern of each action. Second, a grasping movement can be too quick and therefore a normal video camera (normally at 30 fps) is not able to locate correspondences between the first and last frames properly, causing difficulties when detecting a motion pattern. In order to avoid the mentioned disadvantages, we propose a new scheme that makes use of a combination of multiple frames in conjunction with an HMM process to predict grasping movements in addition to the desired object. The methodology for this system is composed of two main steps. Firstly, the proposed method only uses the visual information obtained from a camera beneath the user's wrist to recognize grasping movements. Secondly, the above information is combined with an object recognition methodology and an eye-movement analysis to differentiate fixations from grasping movements. Therefore, this method is capable of detecting when a user wants to grasp an object as well as recognizing the actual desired object. Experiments were conducted in two stages according to each phase of the methodology. A group of objects were placed on a fixed table in similar conditions to classical therapy protocol [51].

This section describes the proposed methodology to predict grasping actions by utilizing eye–hand coordination. As stated before, there are different approaches to detecting human gestures based on a combination of one, or multiple cameras, in addition to the camera's position relative to the user. The main idea consists of capturing two viewpoints of the same spatial domain, without external markers on objects, in order to infer grasping movements. To achieve this goal, it is necessary to detect a grasping action from the moment a movement has been initiated. According to Tamura et al. [49], when the speed of the hand movement is faster than 50 cm/s, the direction of the hand toward the target is stable and almost remains the same. For this reason, a grasping movement could be detected before the hand reaches the desired object.

Our analysis has been separated in two main steps. First, only the information provided by the camera beneath the user's wrist is used. The idea is to detect grasping actions using an HMM framework. Secondly, predicting gesture recognition requires a second HMM that combines grasping movements with the information provided by an eye-tracker. When a user wants to grasp an object, the gaze and hand trajectory remain almost stable [49,11]. Therefore, correspondences increase because there are more correlated areas over time.

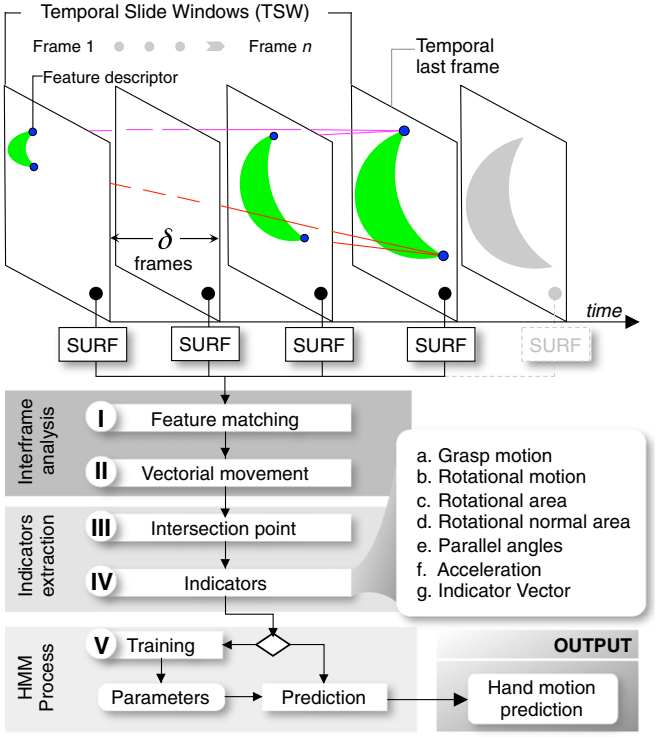

Fig. 2. Proposed hand gesture recognition model based on the analysis of temporal slide windows (TSW) interspaced by δ-frames.

This information is used later to differentiate fixations and grasping actions. Below is the proposed method's approach to detecting both hand posture recognition and grasping action recognition. A general configuration is presented in Fig. 2.

3.1. Hand motion recognition

The main problem when dealing with hand gesture recognition is developing a robust motion pattern. To achieve this goal, a point-to-point correspondence analysis is performed. Here a robust invariant descriptor called SURF is used [6]; mainly because of its robustness and speed against variations in scale and rotations. Experimentally, even though this method is efficient for objects in everyday life as well as objects used in this study, the number of corresponding points detected is low and their distribution is not uniform throughout the image. In addition, the distance between objects and the camera is limited and the movement happens very quickly [49]. In these conditions, classical methods for camera movement estimation are not well suited because they suppose a priori scene model or a large set of resilient points. The main objective is not to estimate the parameters of the movement, but to infer its direction and in particular, to estimate the probability that the current motion is a grasping movement. The proposed method is based on several cues related to observed motions and extracted from a robust motion pattern based on Temporal Slide Windows (TSW). These cues are provided to an HMM framework in order to recognize four normal hand gestures which are reach, retreat, translation and rotation movements. In natural grasping gestures (without obstacles), these movements are not completely mixed.

3.1.1. Movement representation using a TSW approach

Hand gesture recognition is performed using the appearance-base model. This configuration is better illustrated in Fig. 3. As observed in the sequence, it is possible to infer that the trajectory remains constant when the user initiates a movement toward a specific object. Accordingly, all objects in the scene start to disappear from the user's FOV until the hand has reached the required object. Conversely, if hand movements are stochastic, there is a reduced probability that the user is performing a grasping movement because the motion-descriptor does not show an approach pattern. The previous statement is the key point in our gesture recognition framework, however, the current problem is how to accurately build a robust motion pattern. To achieve this goal, a tracking analysis is performed to resolve the correspondence problem.

Most tracking algorithms founded on appearance-base models compute an object's 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 with example methods based on optical flow estimation [5]. Considering the current investigation, the hand motion is particularly fast when the user is performing a grasping action or conversely, it is too stochastic in other cases. For this reason, it is important to analyze the motion displacement between intermediate frames. Similar to the spatiotemporal methods described by Shechtman and Irani [44], the proposed method uses a temporal slide window (TSW) approach extracted along video sequences. As suggested by Shechtman and Irani [44], each human action induces a particular pattern despite differences in illumination, background, color or texture. The idea is to relate multiple corresponding points in order to estimate global motion features or indicators[1] on each TSW, which corresponds to the I)↦V) steps in Fig. 2.

3.1.1.1. Feature matching. The first step is to compute invariant interest-points using the SURF algorithm [6]. This task is performed for each δ-frames contained on a TSW, where $\delta\in\{1,\ldots,5\}$. Assuming that the features extracted by SURF are more resilient to long variations, it is possible to relate multiple corresponding points along time with more probability. For instance, let $\mathbf{p}_1^j=[x_1^j,y_1^j,1]^\top$ be the position of interest point *j-th* in time $t=1$ stored in homogenous coordinates. If this interest point is corresponding with point $\mathbf{p}_n^j$ in time $t=$n it must have a strong similarity between their features. Likewise, after δ-frames, point $\mathbf{p}_i^j$ is corresponding with $\mathbf{p}_n^j$ using the same similarity metric, where $i\in\{1,\ldots,n\}$.

Secondly, after extracting interest points for δ-frames, the system attempts to relate them. Specifically, it tries to find a vector that relates point *j-th* $\mathbf{p}_i^j\mapsto\mathbf{p}_n^j$, for all $i\in\{1,\ldots,n\}$. Here, the key idea is to relate multiple corresponding points with respect to the set of points extracted from the last frame. Even if some frames within this relation do not exist, it is not relevant as long as a minimum number of correspondences is established. As a result, the motion complexity caused by the inter-frame approach is reduced [43], additionally this also assures a single correspondence between multiple frames (Fig. 4).

The feature matching procedure to relate two points of interest is as follows. Firstly, it is necessary to calculate the distance of a feature vector $\mathbf{f}_i^j$ of point *j-th* in time $t=i$ against all feature vectors extracted in time $t=n$ as

$$\mathbf{F}_{i,n}^j(\omega)=\arccos\left(\frac{\mathbf{f}_i^j\cdot\mathbf{f}_n^\omega}{\|\mathbf{f}_i^j\|\|\mathbf{f}_n^\omega\|}\right)\ \text{for all}\ \omega\in\Omega, \tag{1}$$

where $\Omega=\{1,\ldots,s\}$ is the set of interest points detected in time $t=n$, and $\mathbf{F}_{i,n}^j$ is a vector containing the angle-value for each point j with regard to point ω. Although the cosine similarity is useful to find the most similar vector by seeking the lowest angle-value, in many cases this correspondence is incorrect because the corresponding point does not exist in the last frame. To avoid this error, it is necessary to employ a procedure to reinforce correct matching. Secondly, the two lowest values of vector $\mathbf{F}_{i,n}^j$ are extracted and defined as

$$d^{j,j'}=\mathbf{F}_{i,n}^j\left(j'\right)\ \text{and}\ d^{j,j''}=\mathbf{F}_{i,n}^j\left(j''\right), \tag{2}$$

[1] In order to reduce ambiguities between the motion features (or grasp features in Section 3.2) and the SURF features, indicators shall be called the motion features.

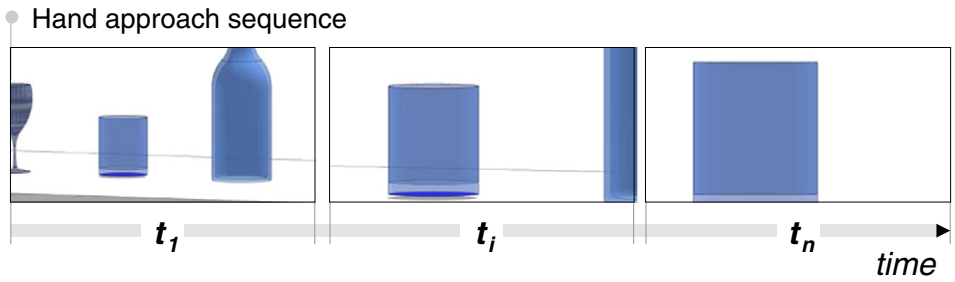

Fig. 3. Hand approach sequence using a camera beneath the user's wrist. In time t_1 multiple objects are detected, later, in time t_i and t_n the field-of-view (FOV) is almost filled due to the proximity between the hand and object.

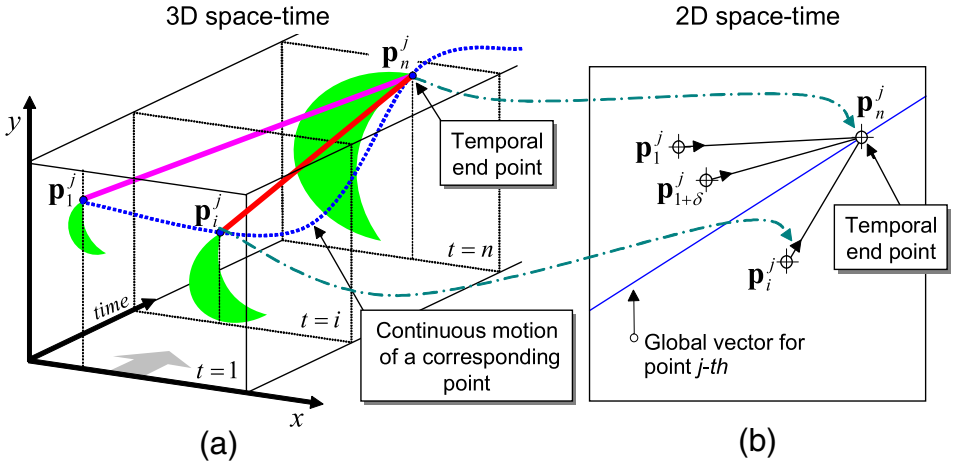

Fig. 4. Schematic view of point correspondence in time-space. (a) Corresponding points in 3D time-space volume. (b) Corresponding points in 2D coordinates.

where $d^{j,j'}$ is the first lowest angle-value, and $d^{j,j''}$ is the second lowest angle-value of $\mathbf{F}_{i,n}^j$, respectively. Therefore, the link between $\mathbf{p}_i^j \mapsto \mathbf{p}_n^{j'}$ is established if the constraint

$$\frac{d^{j,j'}}{d^{j,j''}} < r \qquad \text{where } r \in [0,1], \tag{3}$$

is fulfilled. Parameter r is the relative ratio between the best two feature candidates j' and j'' in order to reduce the number of mismatches and retain the maximum amount of correct matches.[2] In other words, this criterion assures that point *j-th* is matched with its nearest neighbor only if much closer than the second neighbor. Note that point j' refers to an unknown point in time $t=n$; nonetheless, in the case of a correct match $j'=j$, since it is the same point between time $t=\{1,\ldots,i,\ldots,n\}$. This matching criterion is known as the Nearest-Neighbor with Distance Ratio (NNDR) [29]. In general, the NNDR criterion reduces the number of corresponding points when there are noise-points and when a corresponding point does not exist. This last fault normally occurs when there is a rapid motion sequence, as often happens in the problem currently being discussed. According to Sidibe et al. [45], although the NNDR criterion does not have the best performance, it has been selected because it is less expensive computationally (Fig. 4).

3.1.1.2. Vectorial movement. Applying the same procedure on other images of the same TSW, it is possible to build a global vector map that converges on point $\mathbf{p}_n^j$. In order to establish a motion field along this time, several vectors of the same point are required. Namely, let $\mathbf{q}_{i,n}^j$ with $i \in \{1,\ldots,n-\delta\}$ be a homogenous vector that relates points $\mathbf{p}_i^j \mapsto \mathbf{p}_n^j$ defined as

$$\mathbf{q}_{i,n}^j = \mathbf{p}_i^j \times \mathbf{p}_n^j = \left[x_i^j, y_i^j, 1\right] \times \left[x_n^j, y_n^j, 1\right].$$

The $\mathbf{q}_{i,n}^{j,j'}$ vector is established in time $t=\{i,\ldots,n\}$ only for point *j-th*.[3] However, several vectors of the same point are required to establish a motion field along time. For this, it is necessary to define the general motion of multiple vectors that arrive at point $\mathbf{p}_n^j$ as

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

Matrix $\mathbf{Q}_{1\mapsto n}^j$ defines the motion field for point *j-th* for all frames until time $t=n$ (for each δ-frames). Nevertheless, this procedure does not ensure 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, parameter ρ is defined as the minimum number of rows in matrix $\mathbf{Q}_{1\mapsto n}^j$ where *inliers* $\geq \rho$ is fulfilled. Conversely, if this last constraint is not fulfilled, the motion field for that point is discarded. The next step is to derive only one vector, that represents the motion of point *j-th*, along time. For this reason, the angle of feature point *j-th* is mapped along all *inliers*-frames as

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

[2] According to Lowe [29] the r value used is fixed at 0.7.

[3] For simplicity, the notation $\mathbf{q}_{i,n}^{j,j'}$ was changed to $\mathbf{q}_{i,n}^j$, assuming a correct matching between *j-th* and $j'-th$ and in time $t=\{i,\ldots,n\}$.

where $\mathbf{F}^{j}_{1\mapsto n}$ is an $(1\times inlier)$ angle vector of the SURF feature vector extracted for each δ-frames for point *j-th*. In other words, each angle $\mathbf{F}^{j}_{i,n}$ weighs the relative significance between the features of points $\mathbf{p}^{j}_{i}\mapsto\mathbf{p}^{j}_{n}$. Thus, the smaller the angle between the two vectors, the stronger the relation of the same point. Conversely, when the angle-value is maximal, it can be considered noise. Based on such observation, this investigation proposes to represent each angle-value as a weight vector after a linear transformation. Therefore, vector $\mathbf{F}^{j}_{1\mapsto n}$ is transformed to vector $\tilde{\mathrm{F}}^{j}_{1\mapsto n}$, used for weighting each motion vector such that

$$\tilde{\mathrm{F}}^{j}_{1\mapsto n}=1-\frac{\alpha\mathbf{F}^{j}_{1\mapsto n}}{\max\left(\mathbf{F}^{j}_{1\mapsto n}\right)}. \tag{4}$$

Experimentally, α was fixed at 0.98 to use all vectors mapped in $\mathbf{F}^{j}_{1\mapsto n}$. Nonetheless, vector $\tilde{\mathrm{F}}^{j}_{1\mapsto n}$ is not correctly scaled. To determine a correct scale measure, $\mathbf{N}^{j}_{1\mapsto n}$ is computed as

$$\mathbf{N}^{j}_{1\mapsto n}=\frac{\tilde{\mathrm{F}}^{j}_{1\mapsto n}}{\sum_{i=1}^{inlier}\tilde{\mathrm{F}}^{j}_{1\mapsto n}(i)}, \tag{5}$$

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

$$\mathbf{v}^{j}_{1\mapsto n}=\mathbf{Q}^{j\top}_{1\mapsto n}\mathbf{N}^{j\top}_{1\mapsto n}, \tag{6}$$

where $\mathbf{v}^{j}_{1\mapsto n}$ is a (1×3) that maps all $\mathbf{Q}^{j}_{1\mapsto n}(k)$ vectors into a single one by giving more value to vectors with more similarity, based on the weight feature vector encoded in $\mathbf{N}^{j}_{1\mapsto n}$. More precisely, $\mathbf{v}^{j}_{1\mapsto n}$ is a directional vector of point *j-th*, as shown in Fig. 5a.

Additionally, we compute the normal directional vector with the aim of detecting rotational movements, as we shall see later. For this, let $\mathbf{q}_{\perp i,n}{}^{j}$ be the normal vector between points $\mathbf{p}^{j}_{i}\mapsto\mathbf{p}^{j}_{n}$ established between time $t=\{i,\ldots,n\}$ for point *j-th*, defined as

$$\mathbf{q}_{\perp i,n}{}^{j,j'}=\begin{bmatrix}x^{j}_{i}-x^{j}_{n}\\ y^{j}_{i}-y^{j}_{n}\\ x^{j'}_{n}\cdot\left(x^{j'}_{n}-x^{j}_{i}\right)+y^{j'}_{n}\cdot\left(y^{j'}_{n}-y^{j}_{i}\right)\end{bmatrix}. \tag{7}$$

Based on this, let $\mathbf{Q}_{\perp}{}^{j}_{1\mapsto n}$ be the matrix of the normal motion field for point *j-th*, in a manner similar to Eq. (6). Therefore the normal global vector is as follows

$$\mathbf{v}_{\perp}{}^{j}_{1\mapsto n}=\mathbf{Q}_{\perp}{}^{j\top}_{1\mapsto n}\mathbf{N}^{j\top}_{1\mapsto n}. \tag{8}$$

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 $\mathbf{Q}_{\perp}{}^{j}_{1\mapsto n}$ matrix is composed of an array of normal vectors.

3.1.1.3. Intersection point. For the sake of simplicity, the last procedure considered the motion of point *j-th*. Now the problem of estimating the intersection point of multiple corresponding points is discussed. Suppose there are determined multiple vectors $\mathbf{v}^{\Theta}_{1\mapsto n}$, where $\Theta=\{1,\ldots,j,\ldots,k\}$ is the set of interest points detected in time $t=\{1,\ldots,n\}$ and k is the last point in correspondence, as shown in Fig. 5a. For this, let $\mathbf{A}^{\Theta}_{1\mapsto n}$ be a $(k\times3)$ matrix that encodes all motion vectors as

$$\mathbf{A}^{\Theta}_{1\mapsto n}=\begin{bmatrix}\mathbf{v}^{1}_{1\mapsto n}\\ \vdots\\ \mathbf{v}^{j}_{1\mapsto n}\\ \vdots\\ \mathbf{v}^{k}_{1\mapsto n}\end{bmatrix}=\begin{bmatrix}a_1 & b_1 & c_1\\ \vdots & \vdots & \vdots\\ a_j & b_j & c_j\\ \vdots & \vdots & \vdots\\ a_k & b_k & c_k\end{bmatrix}. \tag{9}$$

The next step is to estimate the central point using the vectors contained in $\mathbf{A}^{\Theta}_{1\mapsto n}$. Experimentally, when a grasping movement has been initiated, multiple vectors intersect a common point called the *intersection point*. This situation is better illustrated in Fig. 5. To estimate the position of the unknown intersection point, a non-homogeneous system of linear equations is formulated. This is described as follows

$$\underbrace{\begin{bmatrix}a_1 & b_1 & c_1\\ \vdots & \vdots & \vdots\\ a_j & b_j & c_j\\ \vdots & \vdots & \vdots\\ a_k & b_k & c_k\\ 0 & 0 & 1\end{bmatrix}}_{\mathbf{H}}\begin{bmatrix}x\\ y\\ 1\end{bmatrix}=\underbrace{\begin{bmatrix}0\\ \vdots\\ 0\\ \vdots\\ 0\\ 1\end{bmatrix}}_{\mathbf{b}}. \tag{10}$$

Changing the notation in matrix terms, Eq. (10) can be expressed as

$$\mathbf{Hm}=\mathbf{b},$$

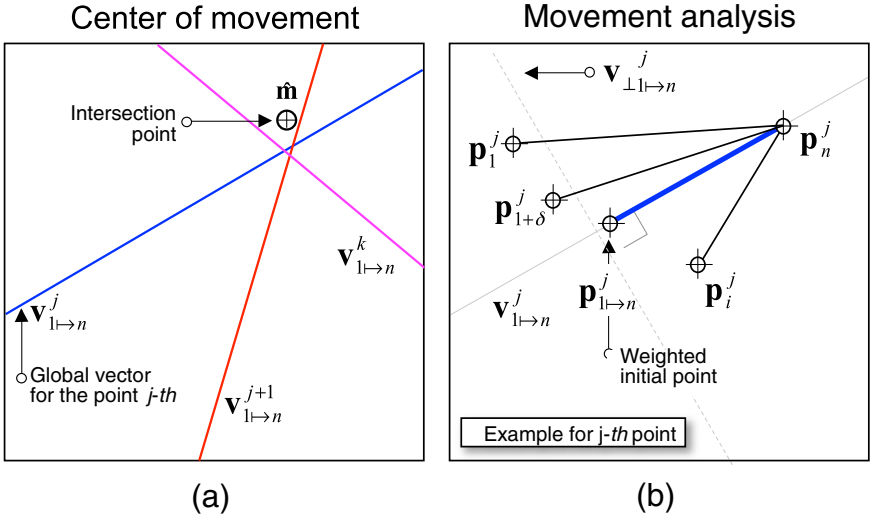

Fig. 5. (a) Multiple lines converge at one point when a reach-to-grasp movement is performed. (b) Once a central point is established, a movement analysis toward that point is performed.

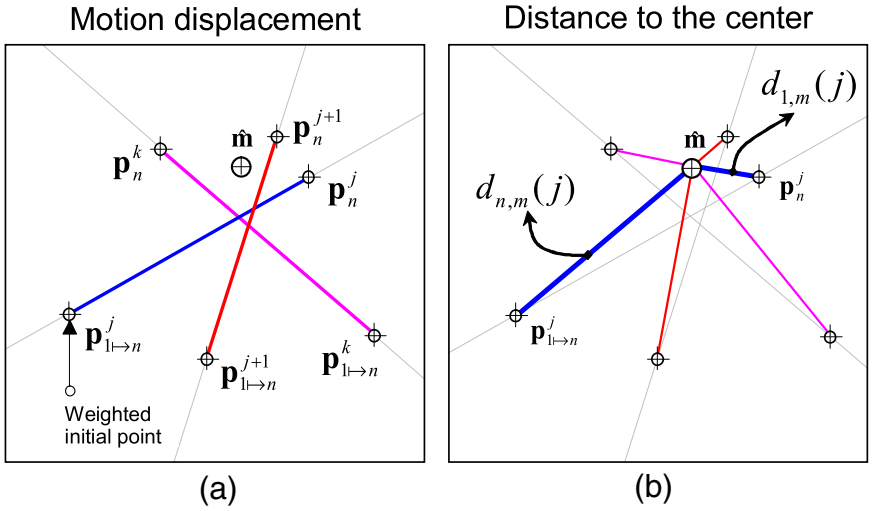

Fig. 6. Analysis of multiple points. (a) Motion of each initial and final trajectory points. (b) Distance to the intersection point.

where $\mathbf{H}$ is an overdetermined matrix of $\mathbf{A}^{\Theta}_{1\mapsto n}$ vectors; because $k \geq \rho$; $\mathbf{m} = [x, y, 1]^{\top}$ is the vector of unknown (x,y), and $\mathbf{b} = [0, \ldots, 1]^{\top}$ is the vector of the right hand side solution of the linear system. Since the intersection of vectors does not have a unique intersection point, here we aim to find a vector $\hat{\mathbf{m}}$ such that $\|\mathbf{Hm} - \mathbf{b}\|$ is minimum. A trivial solution to this problem is solved by means of the Least Square (LS)-solution, that is

$$\hat{\mathbf{m}} = \left(\mathbf{H}^{\top}\mathbf{H}\right)^{-1}\mathbf{H}^{\top}\mathbf{b}.$$

Nevertheless, if the $\mathbf{H}^{\top}\mathbf{H}$ product is ill-conditioned, the estimated LS amplifies the errors, giving an inaccurate position of the intersection point. Hence, we use an orthogonal solution because it is numerically more stable. In particular, we use the **QR** transformation [23]. The **QR** decomposition applied to the $\mathbf{H}$ matrix generates an orthogonal decomposition in terms of an orthogonal matrix $\mathbf{Q}$ and the upper triangular matrix $\mathbf{R}$ such as $\mathbf{H} = \mathbf{QR}$. Therefore, the solution for the nonhomogeneous system, using the **QR** transformation is

$$\hat{\mathbf{m}} = \mathbf{R}^{-1}\left(\mathbf{Q}^{\top}\mathbf{b}\right). \tag{11}$$

Finally, since $\hat{\mathbf{m}} = [\hat{m}_1, \hat{m}_2, \hat{m}_3]$ is in homogenous coordinates, the intersection point defined in the (x,y)-plane is $\hat{\mathbf{m}}_{x,y} = (\hat{m}_1/\hat{m}_3, \hat{m}_2/\hat{m}_3)$. Once the intersection point is established, we seek to compute the *normal intersection point* defined as the intersection of all normal vectors $v^{\Theta}_{\perp 1\mapsto n}$. Based on the above procedure, from Eqs. (10) to (11), first we define the $\mathbf{A}^{\Theta}_{\perp 1\mapsto n}$ matrix of all normal vectors contained in Θ as

$$\mathbf{A}^{\Theta}_{\perp 1\mapsto n} = \begin{bmatrix} \mathbf{v}^{1}_{\perp 1\mapsto n} \\ \vdots \\ \mathbf{v}^{j}_{\perp 1\mapsto n} \\ \vdots \\ \mathbf{v}^{k}_{\perp 1\mapsto n} \end{bmatrix}. \tag{12}$$

Then, changing the matrix terms notation, the problem of estimating the normal intersection point can be expressed as

$$\mathbf{H}'\mathbf{m}_{\perp} = \mathbf{b}', \tag{13}$$

where $\mathbf{m}_{\perp}$ is a non-homogenous vector that encodes the intersection point of normal vectors in correspondence. Using the **QR** transformation applied to the $\mathbf{H}'$, matrix such as $\mathbf{H}' = \mathbf{Q}'\mathbf{R}'$, the normal intersection point is defined as follows,

$$\hat{\mathbf{m}}_{\perp} = \mathbf{R}'^{-1}\left(\mathbf{Q}'^{\top}\mathbf{b}'\right). \tag{14}$$

3.1.1.4. Extracted indicators. Below is an explanation of the eight motion indicators proposed to predict different hand movements.

3.1.1.4.1. Grasp motion. The first two indicators proposed are related to grasping movements. In general, a grasping movement can be split up into two different events. *Reach*: when the hand is moving toward an object; and *retreat*: when the hand is moving backward away from an object. Whichever movement is performed, there will be an intersection point $\hat{\mathbf{m}}$ contained in the TSW. Here, a simple procedure is proposed to infer whether the hand is reaching toward an object or not. Firstly, let $\mathbf{p}^{j}_{1\mapsto n}$ be a $(inliers \times 3)$ matrix representing the 2D position in time $t = \{1,\ldots,n\}$ for each δ-frames; computed in the same way as matrix $\mathbf{Q}^{j}_{1\mapsto n}$

$$\mathbf{p}^{j}_{1\mapsto n} = \begin{bmatrix} \mathbf{p}^{j}_{1} \\ \vdots \\ \mathbf{p}^{j}_{i} \\ \vdots \\ \mathbf{p}^{j}_{n-\delta} \end{bmatrix}. \tag{15}$$

Then, it is necessary to re-map the motion field by taking into account the scale matrix $\mathbf{N}^{j\top}_{1\mapsto n}$. $\mathbf{p}^{j}_{1\mapsto n}$ is defined as a weighted mean position[4] of vector $\mathbf{v}^{j}_{1\mapsto n}$ (see Fig. 6a). Extending this procedure to all Θ-points, let $\mathbf{p}^{\Theta}_{1\mapsto n}$ be the motion of each point in the TSW in $[1,\ldots,n]$, and let $\mathbf{p}^{\Theta}_{n}$ be the final position of each point defined as

$$\mathbf{p}^{\Theta}_{1\mapsto n} = \begin{bmatrix} \mathbf{p}^{1}_{1\mapsto n} \\ \vdots \\ \mathbf{p}^{j}_{1\mapsto n} \\ \vdots \\ \mathbf{p}^{k}_{1\mapsto n} \end{bmatrix}, \quad \text{and} \quad \mathbf{p}^{\Theta}_{n} = \begin{bmatrix} \mathbf{p}^{1}_{n} \\ \vdots \\ \mathbf{p}^{j}_{n} \\ \vdots \\ \mathbf{p}^{k}_{n} \end{bmatrix}. \tag{16}$$

Since vector $\mathbf{p}^{\Theta}_{1\mapsto n}$ codes the initial weighted position, let $d_{1,m}$ be the Euclidean distance of each vector $\mathbf{p}^{\Theta}_{1\mapsto n}$ in relation with intersection point $\hat{\mathbf{m}}$, and let $d_{n,m}$ be the Euclidean distance of each final position $\mathbf{p}^{\Theta}_{n}$ in relation with the same intersection point $\hat{\mathbf{m}}$ as $d_{1,m}(j) = \|\mathbf{p}^{\Theta}_{1\mapsto n}(j) - \hat{\mathbf{m}}\|$ and $d_{n,m}(j) = \|\mathbf{p}^{\Theta}_{n}(j) - \hat{\mathbf{m}}\|$ (see Fig. 6b). Since the estimated position of the initial, final and intersection points can be

[4] Estimated as $\mathbf{p}^{j}_{1\mapsto n} = \mathbf{P}^{j\top}_{1\mapsto n}\mathbf{N}^{j\top}_{1\mapsto n}$.

known, the next step is to determine whether the movement is reaching or retreating. Based on these values, function $v(j)$ is defined as the number of nearest points to the intersection point as follows

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

The resultant function value can be used to define two parameters (g_1, g_2) which are mean $g_1 = \mu(v)$ and variance $g_2 = \sigma^2(v)$. Indeed, $g_1 \mapsto 1$ when movement is reaching and conversely, $g_1 \mapsto 0$ when movement is retreating. To confirm this prediction, variance (σ^2) should be low.

3.1.1.4.2. Rotational motion. The rotational motion indicator gives a temporal variation of each point in correspondence. The main idea is to capture rotational movements independent of turn direction, and thus, compute the angle velocity of each point. Firstly, suppose that the link between $\mathbf{p}_i^j \mapsto \mathbf{p}_n^j$ and $\mathbf{p}_\lambda^j \mapsto \mathbf{p}_n^j$ exists. Therefore, s_i^j and s_λ^j are two consecutive slopes of point *j-th* separated by λ-frames respectively defined as

$$s_i^j = \frac{y_i^j - y_n^j}{x_i^j - x_n^j}, \quad s_\lambda^j = \frac{y_\lambda^j - y_n^j}{x_\lambda^j - x_n^j}.$$

Since both points are signaling to the last point $\mathbf{p}_n^j$ in time $t = n$, by transitivity, this also implies that $\mathbf{p}_i^j \mapsto \mathbf{p}_\lambda^j$, where $t_\lambda > t_i$. Therefore, the angle between these consecutive slopes is

$$\theta_{i,\lambda}^j = arctan\left|\frac{s_i^j - s_\lambda^j}{1 + s_i^j s_\lambda^j}\right|.$$

Based on this result, the angular velocity ω is calculated between $\mathbf{p}_i^j$ and $\mathbf{p}_\lambda^j$ so as to compute the motion variation along time, defined as $\omega_{i,\lambda}^j = \frac{\triangle\theta_{i,\lambda}^j}{rianglet_{i,\lambda}}$, for all $i \in \{1, \ldots, inliers\}$, where $\triangle t_{i,\lambda}$ is the time difference between two consecutive frames (see Fig. 7a). Combining the above value with the Euclidean distance between points p_i^j and $\mathbf{p}_\lambda^j$, the third indicator is as follows

$$g_3 = \frac{\sum_{j=1}^{k}\sum_{i=1}^{inlier}\sigma^2\left(\omega_{i,\lambda}^j\right)}{\sum_{j=1}^{k}\sum_{i=1}^{inlier}\sigma^2\left(\left\|\mathbf{p}_i^j - \mathbf{p}_\lambda^j\right\|\right)}. \tag{17}$$

The above indicator is able to distinguish rotational and translational movements. In the first case $g_3 > 1$ and in the second case $g_3 \mapsto 0$.

3.1.1.4.3. Rotational area. The rotational area is formed by the triangle composed of intersection point $\hat{\mathbf{m}}$, the weighted mean position $\mathbf{p}_{1\mapsto n}^j$ and the final end position $\mathbf{p}_n^j$ for each *j*-point (see Fig. 7b). This indicator allows the system to estimate whether the motion is moving toward an object or not. The area variation of multiple points along the TSW is as follows

$$g_4 = \frac{1}{2k}\sum_{j=1}^{k} d_{1,m}(j) d_{n,m}(j) \sin\left(\phi_{1,n}^j\right) \tag{18}$$

where $\phi_{1\mapsto n}^j$ is the angle centered at $\hat{\mathbf{m}}$ and $d_{1,m}(j)$ and $d_{n,m}(j)$ are the adjacent segments.

3.1.1.4.4. Rotational normal area. When the movement is purely rotational, the proposed method suggests a similar indicator as in the above case; however, here the normal intersection point $\hat{\mathbf{m}}_\perp$ is utilized and defined as follows

$$g_5 = \frac{1}{2k}\sum_{j=1}^{k} d_{1,m_\perp}(j) d_{n,m_\perp}(j) \sin\left(\rho_{1,n}^j\right) \tag{19}$$

where $\rho_{1,n}^j$ is the angle centered at $\hat{\mathbf{m}}_\perp$ (see Fig. 9). The above value 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 $\hat{\mathbf{m}}_\perp$ that intersects all normal vectors $v_{\perp 1\mapsto n}^{\theta}$. Consequently, all points have the same spin angle and a similar variation. As a consequence of previous results, it is possible to obtain two angle variations. Combining angles $\rho_{1,n}^j$ and $\phi_{1,n}^j$ in the following indicator

$$g_6 = \frac{\sum_{j=1}^{k}\phi_{1,n}^j}{\sum_{j=1}^{k}\rho_{1,n}^j} \tag{20}$$

allows the system to obtain a variation of motion over time. For rotational movements g_6 tends to be constant. For translation movements, g_6 tends to be high and for reaching and retreating movements it increases or decreases respectively.

3.1.1.4.5. Parallel angles. Parallel angles give the relative variation between angles of each weighted mean position and its final end position. The key point of this indicator is to detect only translational movements, independently of angle direction and movement orientation. The seventh indicator is defined as follows

$$g_7 = \frac{\sum_{j=1}^{k}\left(\left\|p_{1\mapsto n}^j - p_n^j\right\|\right)}{k\sigma^2(\psi)} \tag{21}$$

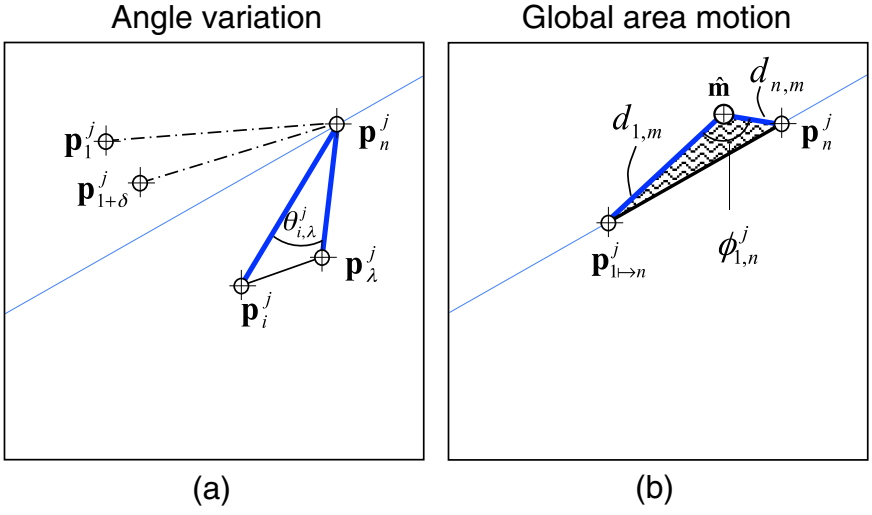

Fig. 7. (a) Temporal angle variation. (b) Global area motion between weighted mean position and last point contained in each time-window.

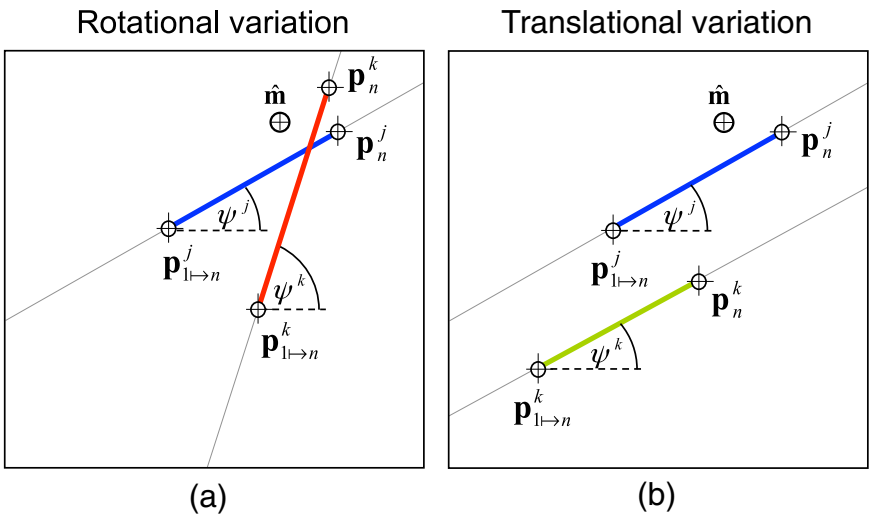

Fig. 8. (a) Different angles are found when motion is rotational. (b) Similar angles are found when the motion is purely translational.

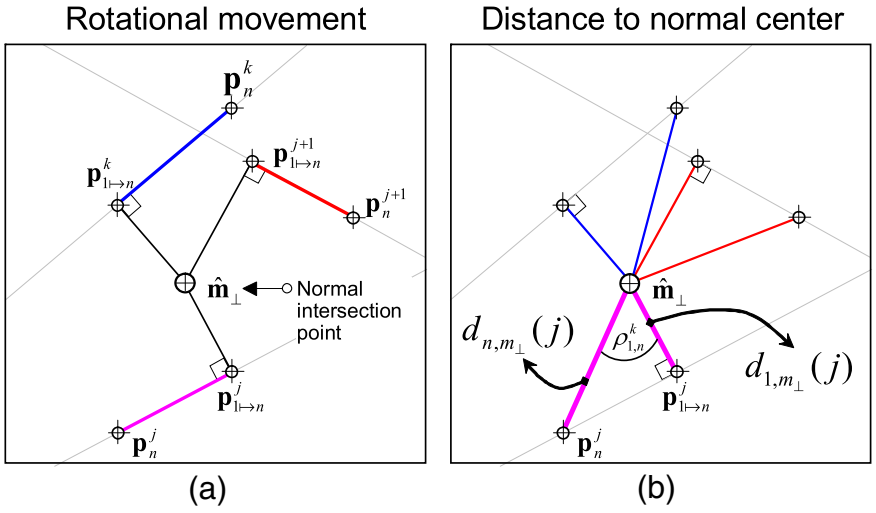

Fig. 9. Analysis of multiple points. (a) Motion of each initial and final trajectory points with respect to the intersection point. (b) Distance to the normal intersection point.

where ψ is the angle of absolute vector $\overrightarrow{p^j_{1\mapsto n}p^j_n}$. In general $g_7 \mapsto 0$ when the movement is rotational and $g_7 \mapsto \infty$ when the movement is purely translational (Fig. 8).

3.1.1.4.6. Acceleration. As mentioned earlier, human gestures are composed of continuous acceleration and deceleration phases. The proposed indicator is designed to detect these variations as follows

$$g_8 = \frac{\sigma^2(a_x)}{\sigma^2(a_x) + \sigma^2\left(a_y\right)} \tag{22}$$

where a^j_x and a^j_y are temporal accelerations with respect to point p^j_n by taking into account the temporal difference $t_{i,\lambda}$ for each i-frame contained in each TSW.

3.1.1.4.7. Indicator vector. In the previous steps eight indicators were proposed that encode different motion features for each TSW. This vector is used as an input for an HMM framework. For simplicity, the above analysis has considered a TSW in time $t=\{1,\ldots,n\}$. Thus, the first feature vector $\mathbf{o}_1$ is composed as follows

$$\mathbf{o}_1 \equiv \mathbf{o}_{1\mapsto n} = [g_1, g_2, g_3, g_4, g_5, g_6, g_7, g_8]^\top \tag{23}$$

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

$$\mathbf{O} = [\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_T] \tag{24}$$

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

3.1.1.5. Training HMM for recognition. Below, the principal component of an HMM based system used to recognize user intention is briefly described. 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, especially when it is necessary to deal with time-series that have spatial temporal variabilities, for example, intention and gesture recognition [32,52,36].

More specifically, HMM is composed of a number of N-states $\{S_1, S_2, \ldots, S_N\}$ connected by transitions, where each transition has an associated 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 $\pi = \{\pi_i\}$. Using a compact notation, an HMM is fully specified by the triplet $\lambda = (A, B, \pi)$ where

- $A = \{a_{ij}\}$ where $a_{ij} = Pr(q_{t+1} = S_j | q_t = S_i), 1 \le i,j \le N$ is the state transition probability distribution and q_t represents 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 the proposed method, observations are modeled with a Gaussian distribution $b_j(\mathbf{O}) = N\left(\mathbf{o}, \mu_j, gma_j\right)$ where $\mathbf{o}$ is the feature vector extracted in the last step.
- $\Pi \equiv \{\pi_1, \pi_2, \ldots, \pi_N\}$ where $\pi_i = p(q_1 = S_i), 1 \le i \le N$ is the initial state distribution.

Based on the above parameters, the problem we face is categorizing each class defined as a particular hand movement. First, it is necessary to create an HMM for each category using the well known Forward–Backward algorithm [38] in order to find the best parameters for each HMM. This is a generalized Expectation–Maximization (EM) algorithm by maximizing the probability of observation sequences given each HMM model for all training sequences. Once the HMM parameters are established, the goal is to recognize an observed symbol sequence as a particular hand gesture. Suppose that each λ_i where $i = 1, \ldots, C$, is a model parameter defined for i-class on C classes, where C is the number of movements detected by the system. Given a sequence of observations $\mathbf{O}$, it is possible to calculate $p(\mathbf{O}|\lambda_i)$ for each HMM λ_i and then choose the class with maximum probability as

$$class = arg \max_i (p(\mathbf{O}|\lambda_i)). \tag{25}$$

3.2. Grasp intention recognition

This section describes how the user's gaze, as well as hand gesture recognition, improves the detection of grasping actions. Below, the general process is described using an eye-tracker and camera placed beneath the user's wrist (Fig. 10).

3.2.1. Grasp features

This process has been developed using grasp features or new indicators combined with previous results in a new HMM framework.

3.2.1.1. Saccade detection. Various studies have shown that fixations are stable directly before the user initiates a grasping movement [22]. Conversely, saccade movements do not allow the gaze to remain in a stable position. Since eye-trackers provide the (x,y) position of the eye's gaze, it is possible to compute the velocity rate $v_x(i)$ and $v_y(i)$ of each TSW for all $i = 1, \ldots, n$. Based on the above information, the proposed method suggests the following feature to quantify the global velocity as

$$h_1 = \sigma(v_x) + \sigma\left(v_y\right).$$

Normally this feature has a low value when fixations are stable and a high value for saccade movements.

3.2.1.2. Features reduction. The main objective of this task is to find more resilient features over time in order to recognize the desired object in a video sequence. Here the proposed method is similar to the method put forth by [47] to build a visual vocabulary. The key idea is that few descriptors can be seen many times over the video-sequence. Accordingly, more resilient features are used later to classify an object against a new video sequence. In general, there are many ways to create a codebook [19]. Here a simple method to compute the codebook is implemented. First, random frames are extracted from a video sequence that captures the user's gaze. Second, each feature is

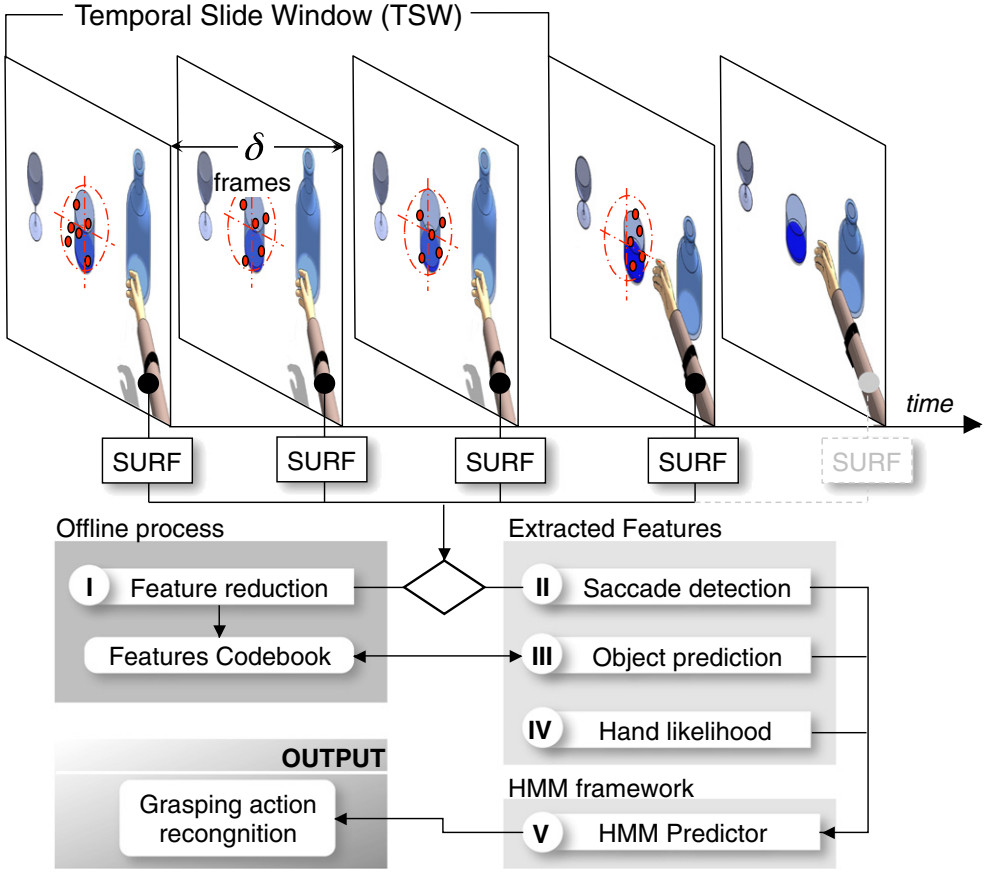

Fig. 10. Proposed recognition model for grasping motion.

Fig. 11. Rotational motion sequence for objects used in the experiments.

classified as part of an object. Third, all other feature space is explored using the Mahalanobis distance in order to create a codebook using a Vector Quantization (**VQ**) algorithm. Once the **VQ** features of each object are extracted, matrix $\mathbf{F_n}$ is defined as the codebook of *n*-objects of interest.

3.2.1.3. Object recognition. After creating a codebook for all objects contained in the scene, new features are extracted from another video sequence containing all previously analyzed objects. The key point is that some features have properties similar to a specific object contained in the codebook. To increase the probability of correctly classifying an object, several features contained in the same TSW have been extracted.

Here the cosine angle distance function is used to measure matching between an unknown feature vector and a known feature vector contained in the codebook. Considering a function named *class*, which provides the class of the nearest known feature in the codebook, the system computes $S_j = \sum_i (class(f_i, \mathbf{D}_n) == j)$, for all $i = \{1,\ldots,p\}$, where f_i is an unknown feature vector extracted from the camera's video sequence (this camera is attached in the user's head); p is the number of vectors contained in one TSW, j is the number associated with an object and $\mathbf{D_n}$ is the codebook containing an array of feature vectors. Then $h_2 = \max_j(\mathbf{S})$ denotes the recognized object.

3.2.1.4. Hand prediction. In the previous section, a set of features to detect hand intention based on an HMM system was examined. Normally the outcome of this process is defined by choosing the maximal class as $class = arg\max_i(p(\mathbf{O}|\Lambda_i))$. However, in some situations the maximal posterior probability could be incorrect when the probability ratio between multiple classes is low. For this reason the outcome probability of each class is used, given $h_{i+2} = p(\mathbf{O}|\Lambda_i)$, for $i = 1,\ldots,4$, where $p(\mathbf{O}|\Lambda_i)$ is the probability to have detected the i action in that TSW.

3.2.2. HMM for recognition

Six indicators have been defined in the steps above. These features have been designed to detect grasping movements using an HMM as shown below. The main reason for combining information about hand intention, eye position and object stability is when there is a time-delay, fixations are high, the object is always the same in the FOV and the hand motion is stable moving toward an object. In the same way as previously described, a new feature vector contained in a TSW is defined as $\mathbf{o}_1 \equiv \mathbf{o}_{1\ldots n} = [h_1, h_2, h_3, h_4, h_5, h_6]^\top$, where $\mathbf{o}_1$ is defined between time $t = 1,\ldots,n$ for the first temporal slide window. Finally the observed symbol sequence is defined as $\mathbf{O} = [\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_T]$.

4. Experimental results

This section presents the results of two experiments carried out with (i) a camera beneath a user's wrist and (ii) an eye-tracker[5] on the proposed framework.

In the first experiment, the results of an HMM framework are described in order to perform a motion prediction, without markers on objects, in five different objects (Fig. 11). In the second experiment, hand motion prediction is combined with the user's gaze position (captured by an eye-tracker) in order to predict grasping movements and detect the desired object.

4.1. Experiment 1

The goal of the first experiment is to evaluate the performance of the proposed eight features in correctly predicting each grasping movement. An example of the video sequence is shown in Fig. 12. At this stage, five video sequences at 30 fps digitized into 320×200 pixel with 256 gray-level images were employed and from this, 7131 TSWs (classified manually) of 21,544 frames using multiple objects were analyzed, as shown in Fig. 12. In the following experiments, we create multiple HMMs from a unique training object. Specifically, we used 1466 TSWs obtained from a bottle without markers on the surface. The above set was separated into ten blocks in order to evaluate the performance of each HMM. It this important to stress that this object was not used in the performance evaluation later. Thus, we expect to have real-life accuracy given that the objects used to evaluate the algorithm were not used to build each HMM.

In our experiments each TSW used a combination of non-consecutive four frames intercalated by three inter-frames (i.e. $\delta = 3$) (e.g. TSW_1 uses frames in time $t = \{1,4,7,11\}$, TSW_2 uses frames in time $t = \{4,7,11,14\}$

[5] An ASL Eye-Trac 6 was employed to capture user gaze.

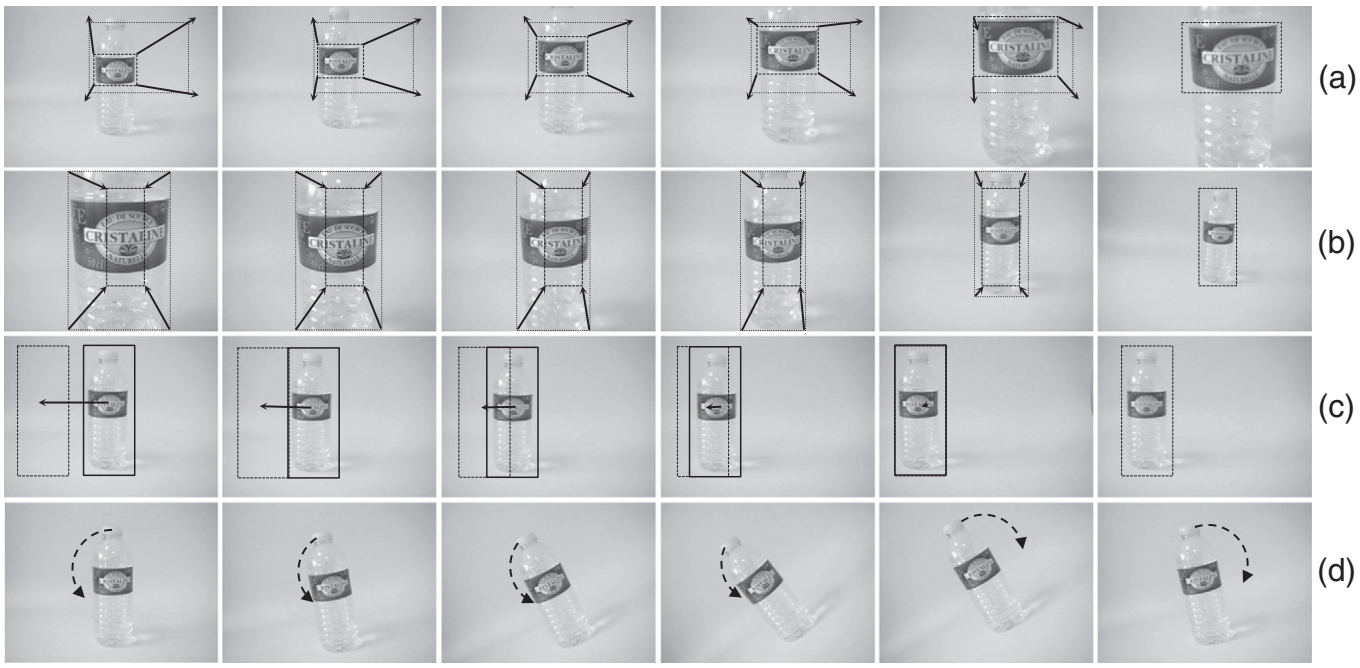

Fig. 12. Real image sequence with one object performing four actions (a) reach, (b) retreat, (c) linear and (d) rotary movements.

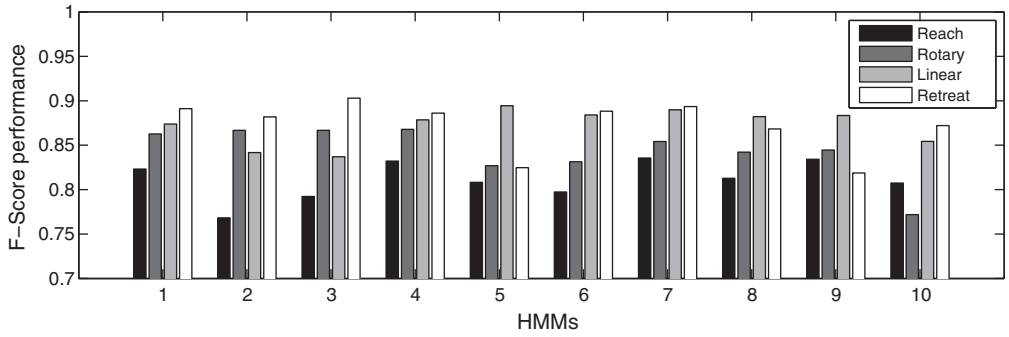

Fig. 13. Average performance of the F-Score using five objects.

and so on). We tested additional configurations but the best suited for our experiments was $\delta = 3$ mainly because it utilizes enough information to capture hand motion flows. To evaluate the performance, an action is considered correct if the motion contained on that TSW was predicted correctly. Additionally, the system must be independent of the objects contained in the scene, as was explained before. In general, the performance of an HMM varies according to the data used for testing. Keep in mind that our aim is to evaluate the performance of objects not used during the HMM training. For this reason each HMM was tested on five objects with users performing each particular action with one object at a time. These objects are a cup, bottle, mug,[6] box, and a stick of deodorant.

Fig. 13 shows the average performance of ten HMMs using five different objects (Fig. 11). It is possible to observe an average performance[7] of F-Score = 0.85. In relation to grasping movements, the reach action had a lower performance because it is normally classified incorrectly as a rotary movement. On the contrary, the retreating action had, on average, the best performance which was near 90%. Fig. 14a and b reveals that performance varies according to the object being analyzed. In relation to object performance, the bottle had the lowest performance because the SURF algorithm was unable to detect a large number of descriptors. Fewer descriptors do not allow for a robust TSW. On the other hand, the mug had the best performance given that a large number of descriptors were used to build robust features, as shown in Fig. 14b.

In these experiments the best HMM generated was used with the cross validation method. For this task the best performance of each action was selected using as criteria the best F-Score and the best True Positive (TP) rate. The results show that it is possible to increase the performance by 2% when using the best combination of HMMs with the F-Score and by 4% with the best combination of TP, as shown in Fig. 14b–c.

4.2. Experiment 2

The main objective of the second experiment is to evaluate factors influencing the performance of the algorithm. Through information garnered in past experiments, we use the best HMM algorithm as a base configuration, which in this case corresponds to "hmm best TP". At this point, we evaluate the algorithm's performance analyzing the five objects used in the experiment (Fig. 11).

Consistent with previous results, we note that performance varies according to which object is used (Fig. 15). Out of the set of objects used in the experiment, the bottle and deodorant generated the worst results. From these findings we conclude that the algorithm is not particularly robust when detecting movements where acceleration sharply

[6] Different from the training phase.
[7] F-Score = 2(precision ∗ recall) / (precision + recall).

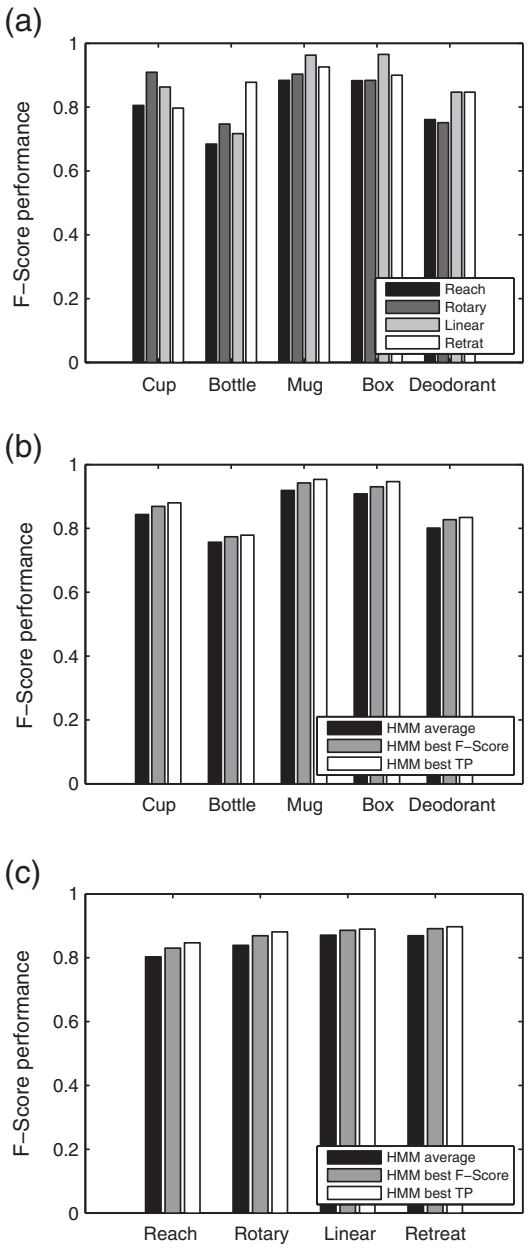

Fig. 14. (a) Average performance for each action using all HMMs. (b) Average performance for all actions on each object using three HMM parameters. (c) Average performance for all objects on each action using three HMM parameters.

increases (or decreases). This implies that acceleration directly affects correspondence, either by generating incomplete correspondence and/or incorrect correspondences through time (Fig. 16). On the contrary, as movement acceleration decreases, the number of matches over time increases, independent of the specific movement and object.

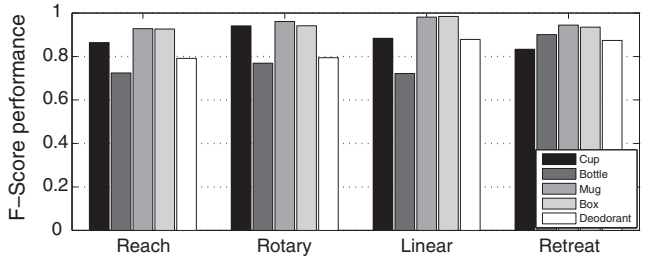

Fig. 15. HMM performance with the best HMM configuration.

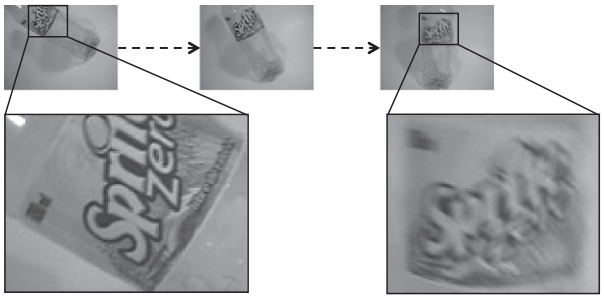

Fig. 16. Declining image quality implies a deteriorating descriptor set.

Thus, the algorithm obtains a higher performance. These results are consistent with objects that have the best performance (mug, box), given that they contain a greater number of matches due to increased surface points of interest.

When performing a hand movement (either a grasping, rotational or translational motion), accelerations and decelerations are naturally generated from the arm movement. The effect of this motion on a frame is evidenced by the linear degradation of the image (Fig. 16). Although the algorithm used to perform inter-matching is invariable to rotation, scaling and translation [6], it is not invariable to linear degradations. Therefore, the majority of the correspondences are incorrect which results in the degradation of not only descriptors extraction, but also the final classification algorithm. One way to avoid this effect may be through increasing the number of frames per second, or through applying a linear restoration filter; however, this would increase the computational cost. An increased amount of frames necessarily implies a lower level of degradation, which in turn would result in a greater number of correspondences in each temporal block.

4.3. Experiment 3

The goal of the third experiment is to evaluate the performance of the combination of grasping movements with the user gaze position, as described in Section 3.2. As stated below, when a user performs a grasping movement toward an object there is a delay in which he/she acquires the object in his FOV. In this period there are fixations around the object at one or multiple points. Only after that, he/she can move his hand toward the desired object. In our experiments we assume that an object is always the same in both views only when a grasping movement has been initiated. The main reason being is that it is always necessary to carry out fixations before grasping an object, as was described before. Combining the user's gaze and hand movements allows us to increase the probability of predicting a grasping movement. Generally the hand camera and the head camera are not pointing at the same object all the time. In fact, the grasping movement can be detected only for a small fraction of time. That is why this task is very complex. An example of this situation is illustrated in Fig. 17. As we can see, when a grasping movement has been initiated, both views share some part of the object. Although in some cases very little information is shared, it is not relevant as long as we can predict motion with the camera attached to the wrist and detect the desired object with the camera attached to the user's head. Additionally, the main reason for using TSWs is that they allow us to collect information about the object even if it is completely occluded for a small period.

To evaluate performance, a synchronized video using both the eye-tracker and the camera attached to the wrist was created. This stage was composed of 404 TSWs. Here four objects[8] were placed

[8] Specifically a mug, a key-ring, an ID card and a mint-box.

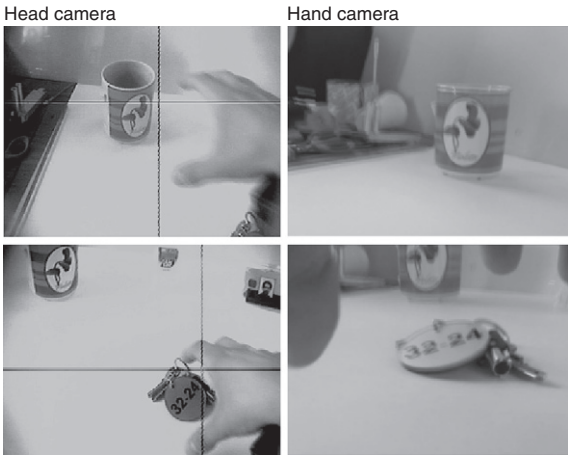

Fig. 17. Acquisition examples obtained from both cameras.

on a uniform table, separated approximately by 15 cm (without obstacles). Using the same configuration described in Fig. 1, a user performs grasping movements without actually grasping the object, he/she then performs the same action with other objects. Using this sequence, the HMM's ability to correctly predict each TSW as a fixation or grasping movement was evaluated. Although there were multiple objects on the table, an HMM trained with only one object (from the Experiment 1) was used. The main objective is for the system to correctly predict an action independent of the objects contained in the scene, validating our results in unknown environments.

Table 1 shows the performance obtained from this experiment as a confusion matrix; classified as True Positive Rate (TPR) and False Positive Rate (FPR). One can see that performance is high when detecting grasping movements, nonetheless, a high false positive rate is also present. The majority of those false positives occurred while observing abnormal user behavior: for example, the hand extends toward the object in a normal position but the user does not immediately initiate the gesture. This unpredicted hesitation is probably due to user stress. In future developments, this will be taken into account by integrating a static hand movement estimation.

Table 2 shows the object recognition performance. It is clear that the codebook constructed by the **VQ** method can be an efficient method to detect resilient features and thus build a robust object dictionary from each TSW. In addition, it is also clear that using a limited number of objects has been a key factor in achieving this high performance (Table 3).

Table 1
TSW analyzed over each hand motion video.

Object	Zoom in-out		Rotation motion		Translational motion	
	Frames	TSW	Frames	TSW	Frames	TSW
Cup	1876	622	1226	405	1051	347
Bottle II	1894	628	1211	401	726	240
Mug	2231	739	1221	404	1016	336
Box	2393	792	1209	400	942	312
Deodorant	2414	799	1200	397	934	309
$\sum$	10,808	3580	6067	2007	4669	1544
Bottle I [a]	2292	759	1208	400	928	307

[a] Training data.

Table 2
Grasp gesture performance.

Class	Classified as (TSW)		Performance	
	Fixation	Reach-to-grasp	TPR	FPR
Fixation	270	54	83.3%	1.9%
Reach-to-grasp	6	74	92.5%	16.7%

Table 3
Object recognition performance.

Class	Classified as				Performance	
	Mug	Keys	Box	Card	TPR	FPR
Mug	119	1	2	1	96.7%	0.7%
Keys	1	128	3	4	94.1%	2.2%
Box	0	2	74	1	96.1%	1.5%
Card	1	3	0	64	94.1%	1.8%
Mean					95.3%	1.6%

Therefore, in the future it would be interesting for the system to be tested with additional objects.

5. Conclusions

The main contribution of this work lies in creating a system by fusing the user's gaze and a hand motion estimation. These experiments show that the proposed method can predict user grasping movements as well as the targeted object in the scene by fusing two channels of information. The main contribution of this work lies in the choice to use human vision combined with active vision in the form of a micro-camera placed on the user's wrist. Indeed, the Temporal Slide Window (TSW) paradigm has proven to be an efficient way to recognize human gestures and objects. It can describe complex and diverse temporal visual descriptors compared with classical frame-to-frame analysis. In general, the system has obtained a gesture performance between 80% and 90%. Although the objects analyzed were limited in these experiments, the results are very promising due to the fact that a limited number of resilient features were used.

The proposed method could be utilized in human–computer interaction systems, e.g. [50,42], or as for example, in the project BRAHMA.[9] In this case, the control movements in people with neural degenerative disorders are altered causing motion tremor or slow movements. Despite the fact that the visual functions of those affected by this disorder have not been altered, the control system is unable to plan correct movements without any disruption. In this case our method could help by recognizing grasping gestures. As a future work, the redundancy between visual information provided from both points-of-view will be exploited.

References

[1] C. Achard, X. Qu, A. Mokhber, M. Milgram, Action recognition with semi-global characteristics and hidden Markov models, In: Conference on Advanced Concepts for Intelligent Vision Systems (ACIVS), 2007.
[2] J. Adams, Do cognitive factors in motor performance become nonfunctional with practice? J. Mot. Behav. 13 (1981) 262–273.
[3] J. Aggarwal, Q. Cai, Human motion analysis: a review, Comput. Vision Image Understanding 73 (1997) 90–102.
[4] D. Ballard, M. Hayhoe, J. Pelz, Memory representations in natural tasks, J. Cogn. Neurosci. 70 (1) (1995) 66–80.
[5] J. Barron, D. Fleet, S. Beauchimen, Performance of optical flow techniques, Int. J. Comput. Vision 120 (1) (1994) 43–77.
[6] H. Bay, T. Tuytelaars, L. Gool, Surf: speeded up robust features, In: Proceedings of the 9th European Conference on Computer Vision, May 2006.
[7] A.-M. Brouwer, D.C. Knill, The role of memory in visually guided reaching, J. Vis. 70 (5) (6 2007) 1–12.
[8] C.A. Buneo, M.R. Jarvis, A.P. Batista, R.A. Andersen, Direct visuomotor transformations for reaching, Nature 4160 (6881) (2002) 632–636.
[9] J. Crawford, W. Medendorp, J. Marotta, Spatial transformations for eye–hand coordination, J. Neurophysiol. 92 (2004) 10–19.
[10] T. de Campos, W. Mayol, D. Murray, Directing the attention of a wearable camera by pointing gestures, In: Proceedings of the 19th Brazilian Symposium on Computer Graphics and Image Processing, 2006. SIBGRAPI'06, Oct. 2006, pp. 179–186.
[11] M. Desmurget, D. Pelisson, Y. Rossetti, C. Prablanc, Neurosciences and biobehavioral review, Eye Hand: Plann. Goal-Directed Mov. 220 (6) (1998) 761–788.
[12] R. DeVaul, M. Sung, J. Gips, A. Pentland, Mithril 2003: applications and architecture, In: International Symposium on Wearable Computers (ISWC) IEEE, 2003, pp. 4–11.

[9] 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/.

[13] S. Dockstader, A. Tekalp, Multiple camera tracking of interacting and occluded human motion, Proc. IEEE 890 (10) (2001) 1441–1455.
[14] P. Donkelaar, J.-H. Lee, A. Drew, Transcranial magnetic stimulation disrupts eye–hand interactions in the posterior parietal cortex, J. Neurophysiol. 840 (3) (2000) 1677–1680.
[15] K.C. Engel, M. Flanders, J.F. Soechting, Oculocentric frames of reference for limb movement, Arch. Ital. Biol. 1400 (3) (Jul. 2002) 211–219.
[16] A. Erol, G. Bebis, M. Nicolescu, R. Boyle, T. Twombly, Vision-based hand pose estimation: a review, Comput. Vision Image Understanding 108 (2007) 52–73.
[17] D. Faria, H. Aliakbarpour, J. Dias, Grasping movements recognition in 3d space using a Bayesian approach, In: International Conference on Advanced Robotics, June 2009.
[18] J. Flanagan, S. Lederman, Neurobiology: feeling bumps and holes, Nature 412 (2001) 389–391.
[19] A. Gersho, R.M. Gray, Vector Quantization and Signal Compression, Kluwer Academic Press, 1992.
[20] H. Hamer, K. Schindler, E. Koller-Meier, L. Van Gool, Tracking a hand manipulating an object, In: IEEE International Conference on Computer Vision, 2009.
[21] M. Hayhoe, D. Bensinger, D. Ballard, Task constraints in visual working memory, Vision Res. 38 (1998) 125–137.
[22] M. Hayhoe, A. Shrivastava, R. Mruczek, J. Pelz, Visual memory and motor planning in a natural task, J. Vis. 3 (2003) 49–63.
[23] N. Higham, Accuracy and Stability of Numerical Algorithms, SIAM, 1996.
[24] S. Hodges, L. Williams, E. Berry, S. Izadi, J. Srinivasan, A. Butler, G. Smyth, N. Kapur, K. Wood, Sensecam: a retrospective memory aid, In: International Conference on Ubiquitous Computing (UBICOMP) International Conference on Ubiquitous Computing, International Conference on Ubiquitous Computing (UBICOMP), volume LNCS 4206, Springer-Verlag, 2006, pp. 177–193.
[25] A.K. Jain, R.P.W. Duin, J. Mao, Statistical pattern recognition: a review, IEEE Trans. Pattern Anal. Mach. Intell. 220 (1) (Jan. 2000) 4–37, http://dx.doi.org/10.1109/34.824819.
[26] R. Johansson, G. Westling, A. Bäckström, J. Flanagan, Eye–hand coordination in object manipulation, J. Neurosci. 210 (17) (2001) 6917–6932.
[27] O. Koch, S. Teller, Body-relative navigation guidance using uncalibrated cameras, In: IEEE ICCV, Kyoto, 2009.
[28] M. Land, N. Mennie, J. Rusted, The roles of vision and eye movements in the control of activities of daily living, Perception 28 (1999) 1311–1328.
[29] D.G. Lowe, Object recognition from local scale-invariant features, In: International Conference on Computer Vision, 1999, pp. 1150–1157, (Corfu, Greece).
[30] T. Maekawa, Y. Yanagisawa, K.I. Yasue Kishino, K. Kamei, Y. Sakurai, T. Okadome, Object-based activity recognition with heterogeneous sensors on wrist, In: Proc. of International Conference on Pervasive Computing (Pervasive 2010), 2010.
[31] G. McConkie, C. Currie, Visual stability across saccades while viewing complex pictures, J. Exp. Psychol. Hum. Percept. Perform. 22 (1996) 563–581.
[32] T. Moeslung, A. Hilton, V. Kruger, A survey of advances in vision-based human motion capture and analysis, Comput. Vision Image Understanding 104 (2006) 90–126.
[33] L.A. Mrotek, J. Soechting, Target interception: hand–eye coordination and strategies, J. Neurosci. 270 (27) (2007) 7297–7309.
[34] E. Perini, S. Soria, A. Prati, R. Cucchiara, Facemouse: a human–computer interface for tetraplegic people, In: ECCV Workshop on HCI 2006, volume LNCS 3979, 2006, pp. 99–108.
[35] R. Polana, 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, Nov. 11–12, 1994, pp. 77–82, http://dx.doi.org/10.1109/MNRAO.1994.346251.
[36] R. Poppe, A survey on vision-based human action recognition, Image Vision Comput. 280 (6) (2010) 976–990.
[37] H. Prendinger, A. Hyrskykari, M. Nakayama, H. Istance, N. Bee, Y. Takahasi, Attentive interfaces for users with disabilities: eye gaze for intention and uncertainty estimation, Univers. Access Inf. Soc. 80 (4) (2009) 339–354.
[38] L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proc. IEEE 770 (2) (Feb. 1989) 257–286, http://dx.doi.org/10.1109/5.18626.
[39] G. Rizzolatti, L. Fogassi, V. Gallese, Parietal cortex: from sight to action, Curr. Opin. Neurobiol. 7 (1997) 562–567.
[40] J. Romero, H. Kjellstrom, D. Kragic, Hands in action: real-time 3d reconstruction of hands in interaction with objects, In: IEEE International Conference on Robotics and Automation (ICRA), 2010.
[41] K. Sakita, K. Ogawara, S. Murakami, K. Kawamura, K. Ikeuchi, Flexible cooperation between human and robot by interpreting human intention from gaze information, In: IROS, 2004, pp. 846–851.
[42] D. Sasaki, T. Noritsugu, T. Masahiro, H. Yamanmoto, Wearable power assist device for hand grasping using pneumatic artificial rubber muscle, In: IEEE ROMAN, 2004, pp. 655–660.
[43] K. Shafique, M. Shah, A non-iterative greedy algorithm for multi-frame point correspondence, In: Proc. Ninth IEEE International Conference on Computer Vision, Oct. 13–16, 2003, pp. 110–115, http://dx.doi.org/10.1109/ICCV.2003.1238321.
[44] E. Shechtman, M. Irani, Space-time behavior based correlation, In: Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR 2005, vol. 1, June 20–25, 2005, pp. 405–412, http://dx.doi.org/10.1109/CVPR.2005.328.
[45] D. Sidibe, P. Montesinos, S. Janaqi, Fast and robust image matching using contextual information and relaxation, In: 2nd International Conference on Computer Vision Theory and Applications, VISAPP, Barcelona, Spain, March 2007.
[46] L. Sigal, M.-J. Black, State of art in image- and video-based human pose and motion estimation, Int. J. Comput. Vision 10 (87) (2010) 1–3.
[47] J. Sivic, A. Zisserman, Efficient visual search of videos cast as text retrieval, IEEE Trans. Pattern Anal. Mach. Intell. 310 (4) (April 2009) 591–606, http://dx.doi.org/10.1109/TPAMI.2008.111.
[48] T. Starner, A. Pentland, Real-time American sign language recognition from video using hidden Markov models, In: Proc. International Symposium on Computer Vision, Nov. 21–23, 1995, pp. 265–270, http://dx.doi.org/10.1109/ISCV.1995.477012.
[49] Y. Tamura, M. Sugi, J. Ota, T. Arai, Prediction of target object based on human hand movement for handing-over between human and self-moving trays, In: Proc. 15th IEEE International Symposium on Robot and Human Interactive Communication ROMAN 2006, Sept. 2006, pp. 189–194, http://dx.doi.org/10.1109/ROMAN.2006.314416.
[50] Y. Tamura, M. Sugi, J. Ota, T. Arai, Estimation of user's intention inherent in the movements of hand and eyes for the deskwork support system, In: Proc. IEEE/RSJ, IEEE, USA, Nov. 2007, pp. 3709–3714, http://dx.doi.org/10.1109/IROS.2007.4399618.
[51] G. Thielman, C. Dean, A. Gentile, Rehabilitation of reaching after stroke: task-related training versus progressive resistive exercise, Arch. Phys. Med. Rehabil. 85 (2004) 1613–1618.
[52] P. Turaga, R. Chelappa, V. Subrahmanian, O. Udrea, Machine recognition of human activities: a survey, Trans. Circuits Syst. Video Technol. 180 (11) (2008) 1473–1488.
[53] C. Yu, D. Ballard, A multimodal learning interface for grounding spoken language in sensory perceptions, In: International Conference on Multimodal Interfaces, 2004.
[54] H. Zhou, H. Hu, Human motion tracking for rehabilitation—a survey, Biomed. Signal Process. Control (3) (2008) 1–18.