Active System for Hand Motion Recognition

Alejandro Viada, Matías Ortega and Miguel Carrasco
Escuela de Informática y Telecomunicaciones
Facultad de Ingeniería
Universidad Diego Portales, Chile
{alejandro.viada, matias.ortega, miguel.carrasco}@mail.udp.cl

***Abstract*—Taking something in a hand requires a complex coordination of sight and hand. However, many people with neurodegenerative diseases or other coordination problems are unable to correctly perform this action. This research paper presents a hand motion interpretation system, which uses a video flow captured from under the user's wrist, known as active perspective. To assess the algorithm, we have placed a variety of objects in a work area in front of the user. Through the video flow, our system can classify different hand movements with regard to the objects on the scene. For motion classification, we have proposed a set of descriptors, which are used by the classification algorithms kNN and HMM. Results show that our system is capable of detecting over 90% of approaching and lateral motions, regardless of what the objects on the scene are.**

I. Introduction

There is currently a growing interest in developing interfaces for arm and body motion recognition relating to object manipulations. Some of the best known systems are videogame controllers, like the Nintendo Wii sensor that enables game control through external body devices, or just by moving the body, as is the case of Microsoft's Kinect sensor. A less explored area is the application of visual systems to people whose daily lives are extremely limited by motion difficulties in arms, legs or both, or by a neurodegenerative disease. This is why new forms of man machine interaction are continuously being searched, which is not simple task, because recognizing a user's motion in respect of an object needs constant follow-up of the objects on the scene, apart from determining any geometric or photometric changes through temporal segmentation, as well as any changes in perspective with regard to previous moments [1]–[4].

This research proposes a system, which uses a below wrist camera to detect the objects in front of the user and determine the latter's movements in respect of these objects from changes in the input video flow. Most current methods involve cameras facing the user to capture people's gestures. These methods are called passive, and require segmenting hand or body motions (e.g. [2], [4]–[8]). In other systems, the user's eyes serve as pointer, through a device known as EyeTracker that detects the slightest cornea's reflection through very accurate sensors, thus yielding a specific scene position [9], [10].

Although passive methods have been very successful, their main limitation is precisely the fixed sensor position. This is why this research was designed to propose an active system for the recognition of hand motion towards an object. Active systems are based on sensors located on the body, and consequently, as the user moves, so do the sensors (e.g. [11]). However, it is only possible to visualize the motion flow towards or away from an object for each user action, which brings in the great complexity of inferring the hand motion from the optical sequence on the sole knowledge of a hand approach towards the object that the user wants to take.

Our solution to the stated problem was to develop a motion inference algorithm and apply it to a video containing user gestures in respect of various objects in front of him or her. Experimentation proved that one of the best camera positions to achieve this end was under the user's dominant arm, at the level of the wrist, as shown in the configuration captured in Fig. II-Aa. Even though we did not use a micro camera in our experiments, the system is not limited to function with any given device; hence, any video camera with the same or better features may be used. This representation exemplifies the work method of a person seating at a work station and his or her interaction with the various objects located in the work area. So as to simplify and make the prototype development more efficient, the experiments were conducted in an ideal and controlled environment comprised of a lit-up cubicle with white walls, blue bottom and background, designed to facilitate object itemization (Fig. II-Ab). The set of analysed gestures are forward, backward, rightward and leftward movements, and any combination of these motions.

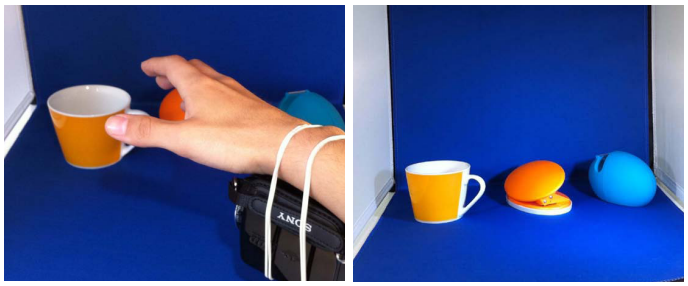
A brief description of the document sections is as follows: Section 2 provides a summarized state of the arts, with the algorithms used in the proposed solution. Section 3 describes the proposed model. Performance results from the conducted experiments are contained in Section 4. Finally, Section 5 presents the conclusions and possible improvements to our proposal.

II. Background Information

The main algorithms for active and passive hand motion determination are outlined in this section, followed by an analysis of the algorithms used in our research that will serve as basis to the next section.

A. Passive Algorithms

Passive methods are designed to operate from a perspective that is external to the user. This means that a camera captures, from a fixed position, the body motion towards an object or the interaction between them. The following is a brief review of some of these systems. Bobick et al. [1] use a



(a) (b)

Fig. 1. System environment. (a) Camera placement under wrist, (b) Example of development scenario

passive system for capturing a person's action and creating a motion history, without the need for any accessories, for further comparing the motion histograms with a knowledge base to interpret the action. Duca et al. [12] present a library that enables any user to handle tridimensional applications using one or both hands, by wearing finger colour labels. Since the project is focused on a domestic use, a low cost camera and easily available and affordable accessories are used. Freeman et al. [3] propose creating orientation histograms based on the user's hand position to yield feature vectors. This approach does not require any accessories, because it works directly with the hand, and it is aimed at a less complex analysis to speed up computation. Martin and Crowley [13] consider hand gestures in front of a projector to yield a virtual desktop. They do not deliver a final solution, but only the idea and methods that may be used in hand recognition, including the detection of skin, morphological transformations and differences between images. The project developed by Gupta [4] is inspired on hand signs, and is hence based on the American Sign Language (ASL). This approach analyses gestures of a static hand, and filtrates the acquired gesture for subsequent comparison with a knowledge base. The captured images are segmented and filtered by Otsu's algorithm, using morphological filtering, finally yielding a representation of the hand contour. Lastly, Kim [6] goes for hand motion and gesture recognition, on the basis of a skin colour obtained from facial recognition. Adjacent images are subtracted, because this system is passive.

B. Active Algorithms

Active systems are not limited by camera position, as they are designed for a continuous representation. Bajcsy [14] and Aloimonos et al. [15] delivered the paradigm for proposing new models and control strategies in active perception systems. The first active camera systems were designed to provide perception to autonomous robots. Active systems generally include cameras located on the human body, and new ways are presently being offered to increase man machine interaction, thus enabling users to move freely.

From a computer based perspective, this scheme gives the user access to a better representation of his or her environment [11], [16]–[18]. The nature of these algorithms makes them a more accurate option to achieve this goal, because they are directly based on the user's arm movements.

C. Methodological Algorithms

SURF: *Speeded Up Robust Features* [19], is a detector and descriptor of image keypoints, invariant to scaling or rotation. As it requires significantly less computing time, it is similar to existing detectors and can even surpass them in what regards repeatability and robustness [20]. Its applications are varied, including, for instance, looking for matches in two images, identifying objects and 3D remodelling [21]. Fast keypoint detection, distinctive description of detected points, expeditious descriptor matching and high repeatability scores are among its main features. Computing time being a critical factor in our proposal, point detection with a degree of invariance and robustness becomes essential, particularly in image sequences presenting substantial changes.

kNN: is a non-parametric and supervised classification algorithm that calculates the distance from a dataset to a larger database containing the classification model [22]. It is aimed at classifying by closeness to the k nearest neighbours. No learning concept is associated with this algorithm, since the class of an object is predicted for classification by finding the most similar objects. The most common metrics are used to measure neighbour proximity, such as Euclidian and Mahalanobis distances. Further details on their implementation can be found in [23]. The real time operation requirement gives relevance to implementing an algorithm capable of efficiently classifying data groups without the need for additional parameters, so as to obtain shorter times, and depending on data behaviour, more accurate classifications as well.

HMM: The Hidden Markov Model (HMM) algorithm [24] is a statistical model, where data to be modelled are assumed to have a Markovian behaviour with unknown parameters. This implies that there are hidden states that can only be observed through another set of stochastic processes. The model is aimed at determining these unknown parameters from observed data.Learning is a stage prior to classification and consists in a single run process yielding matrixes to be used for classifying [25], [26]. The main problem of Markov's hidden chain is that the probabilities needed to model the system are unknown. They are obtained throughout the learning cycle that considers a training dataset to find the information that marks the transition between states and enables modeling, as well as maximizing probabilities, a process completed through the use of the *Baum-Welch* algorithm. On the other hand, the optimal path to a state is determined by the *Forward-Backward* algorithm, which calculates the probability of a sequence resulting from the already known parameters of a HMM; i.e., the likelihood of the system being in a T state, given a set of observations. All possible states of a sequence must be determined to this end, based on the current combination of observed states. The learning stage makes a classifier available to the system for data behaviour interpretation prior to classification. Having more than one classifier enables their complementing one another for maximum accuracy.

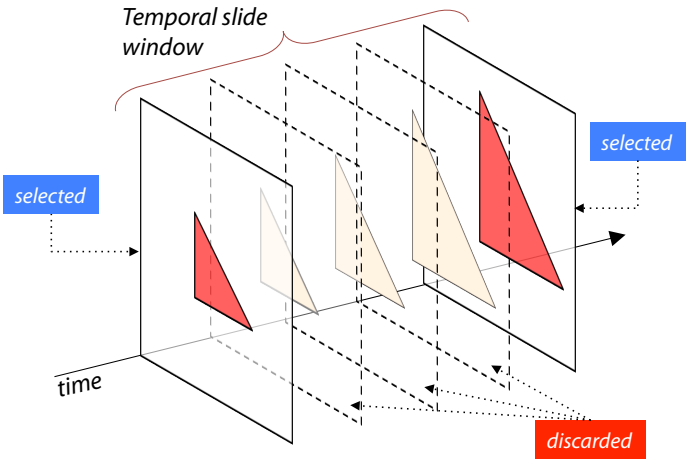


Fig. 2. Interleaved frame extraction model in a temporal frame block

III. Proposed Method

The algorithm's preliminary phase adjusts the video size defined with a 360×240 pixel resolution. Even if low, this resolution has two important advantages: Firstly, it enables SURF to create a sufficient number of descriptors, and secondly, it requires little computing time compared to higher resolutions. These steps significantly reduce video loading and processing times, and deliver results in real time. The proposed method is comprised of five phases: First comes the extraction of a given number of frames per second for processing by the SURF algorithm in search of keypoints [19]. The resulting more marked changes between periods of time facilitate motion recognition. Second, the RGB channels are merged into a single gray scale channel; hence, reducing the volume of video flow information. Thirdly, invariant keypoints are detected in each of the SURF selected frames, to then match, in the fourth phase, the various keypoints to the frames selected during the first phase. A set of descriptors is generated in the fifth phase to describe the motion yielded by the optical flow. Then, these descriptors are used by the classification algorithms that assess the information and classify the type of motion. The phases of the proposed algorithm are separately detailed below.

A. General Process Stages

I. Frame Extraction: A defined number of equidistant frames is extracted per second, so as to be able to identify the most marked motion changes on the scene in a temporal frame block. See an example in Fig. 2. The greatest advantage of this scheme resides in speeding up the algorithm and reducing descriptor noise during the fifth phase

II. Conversion to Gray-Scale: All three RGB information channels are merged into a single gray-scale channel during this phase. Although colour information may be useful, SURF uses a single channel for keypoint detection.

III. Descriptor Obtainment: Invariant keypoints extracted by SURF during this phase form a matrix with point descriptors and positions in each frame. Using these invariant keypoints gives an edge of independence from the objects on the scene. Fig. 3 shows an example of keypoints detected in an image.

IV. Keypoint Matching: During this phase, the keypoints that are found in one frame are matched to those in other selected frames (interleaved by λ-frames), eliminating those that show no similarity by means of the NNDR algorithm [27]. This process enables linking multiple points in different frames (Fig. 4).

V. Descriptor Generation: During this final phase, the proposed metrics are put to use to determine whether there was movement, and a matrix system is produced, where a keypoint map is created indicating point positions in each processed frame. Since the recognized motions are horizontal and in depth, a separate work model is defined for each. Because this aspect is a core matter in our research, a detailed description follows on how the set of descriptors is generated for interpretation. Horizontal motion is defined as the horizontal difference between several pairs of keypoints, as shown in Fig. 5, where $\overrightarrow{d_i}^t$ and $\overrightarrow{d_i}^{t+\lambda}$ represent the difference between pairs of corresponding keypoints in two matching frames. The motion direction is defined as rightward when the difference is positive, and leftward when negative. Motion in depth, whether forward or backwards, is determined by analyzing distances between keypoints in the same frame and comparing the difference with its correspondent in another frame. Any increase in the absolute difference between $\overrightarrow{d_i}^t$ and $\overrightarrow{d_i}^{t+\lambda}$ is interpreted as the user's hand approaching an object, with any decrease considered as moving away from the object. The diagrams in Fig. 5 represent these models.

B. Descriptor Generation

Next, we describe ten motion descriptors ($d_1 \mapsto d_{10}$) used to describe a gesture. We separate our analysis in regard to horizontal and depth motions. For this, let k the set of keypoints pairs in correspondences in two interleaved frames of size $n \times m$ pixels.

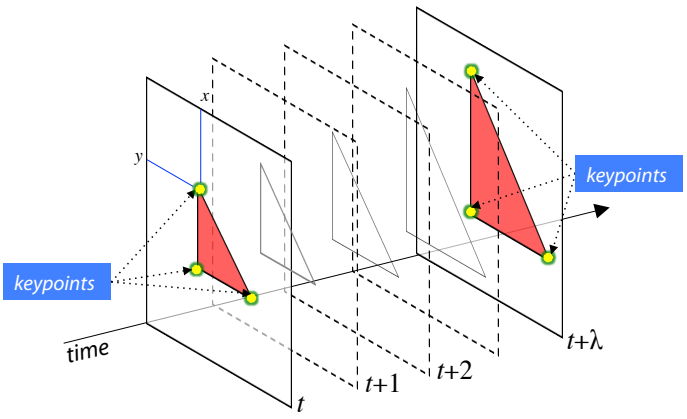


Fig. 3. An example of a keypoint detection for each selected frame

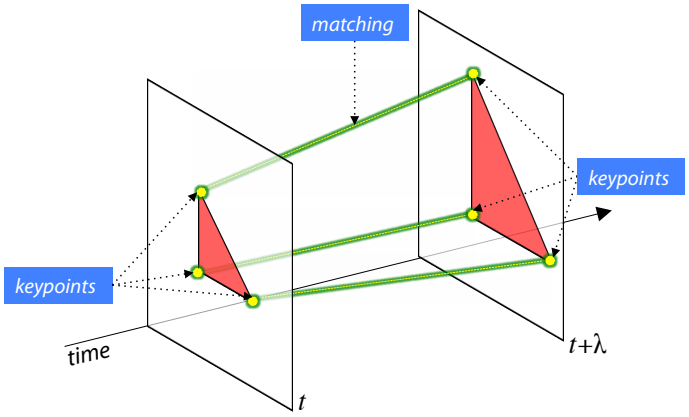


Fig. 4. Matching between two interleaved frames

I. Horizontal Motion: Horizontal motion descriptors are drawn from the information on keypoint pairs contained on the horizontal axis. For this, let $p_i^t(x,y)$ and $p_i^{t+\lambda}(x,y)$ a corresponding pair of keypoints contained in t and $t+\lambda$ times respectively, and let $l_i = ||p_i^t(x) - p_i^{t+\lambda}(x)||$ the euclidean horizontal distance between to interleaved frames of the i-corresponding pair (Fig.5a). The following proposed descriptors are,

- The arithmetic mean of Euclidean distance

$$d_1 = \frac{1}{k}\sum_{i=1}^{k} |l_i|$$

- The average value of difference measurements

$$d_2 = \frac{1}{k}\sum_{i=1}^{k} \overrightarrow{l_i}$$

Descriptor d_1 is used to determine the magnitude of a motion, while d_2 the difference provides the motion sense. Specifically, the difference in coordinates from t to $t+\lambda$ times is negative when moving to the left, and positive for rightward motions. From these descriptors, three classes of motion can be determined: (1) leftward, (2) rightward and (3) at rest.

II. Depth Motion: The following descriptors were generated to describe depth motion. For this, let A the set of keypoints pairs above the horizontal centre, and let B the set of keypoints pairs in correspondences below the horizontal centre, where the centre of the frame is located at $n/2$ pixels. The following proposed descriptors are,

- The average direction of the gradients formed by the pairs of points above and below the horizontal centre of the frame in a collapsed frame-time (Fig.5b).

$$d_3 = \frac{1}{|A|}\sum_{i\in A} u_i$$

$$d_4 = \frac{1}{|B|}\sum_{i\in B} v_i$$

where u_i and v_i are gradients of sets A and B respectively.

- The average keypoint distances to their mass centre point m^t and $m^{t+\lambda}$ (Fig.5c-d);

$$d_5 = \frac{1}{k}\sum_{i=1}^{k} {c_i}^t$$

$$d_6 = \frac{1}{k}\sum_{i=1}^{k} {c_i}^{t+\lambda}$$

where $c_i^t = ||p_i^t - m^t||$ and $c_i^{t+\lambda} = ||p_i^{t+\lambda} - m^{t+\lambda}||$.

- The centre point coordinate differences (Fig.5f-h),

$$d_7 = \left(m^{t+\lambda}(x) - m^t(x)\right)$$

$$d_8 = \left(m^{t+\lambda}(y) - m^t(y)\right)$$

- The arithmetic mean of the distance ratio between frames from the keypoints to the straight line drawn from the centre point in t time, using its horizontal coordinate (Fig.5h-i).

$$d_9 = \frac{\sum_{i\in A} |p_i^t(y) - \frac{n}{2}|}{\sum_{i\in A} |p_i^{t+\lambda}(y) - \frac{n}{2}|}$$

$$d_{10} = \frac{\sum_{i\in B} |p_i^t(y) - \frac{n}{2}|}{\sum_{i\in B} |p_i^{t+\lambda}(y) - \frac{n}{2}|}$$

Descriptors d_3 and d_4 represents the gradient ratio of matched points above and below the horizontal centre of the image, resulting from drawing an imaginary straight line connecting each point to its pair. When motion is an approach, keypoints tend to spread out, and they tend to compress when moving away (Fig. 5b). Descriptors d_5 and d_6 measures the average distance from a set of keypoints to its centre point per frame. This means that the keypoint cloud has a centre of mass, and each keypoint is a certain distance away from this centre (Fig. 5c-d). The distances are averaged out and compared to the next frame (Fig. 5e). A t time distance that is greater than that in $t+\lambda$ is considered an approaching motion, and the opposite, as moving away. These distances are illustrated in Fig. 5f. Descriptors d_7 and d_8 is the difference in centre point coordinates, as additional information on the displacement undergone by a point cloud from t to $t+\lambda$. Finally, descriptors d_9 and d_{10} are obtained through the following process:

1) a $t+\lambda$ time keypoint cloud is centered on the t mass centre of the keypoints, with the difference in coordinates provided by the previous descriptor;
2) a mean straight line is drawn by means of the y coordinate representing the t frame centre point $\frac{n}{2}$;
3) distances of the cloud of points to the mean straight line are determined;
4) distances over the mean straight line of the first cloud are divided by those of the second cloud, and the same applies to distances of points below the line, and
5) the average of both ratio vectors is found.

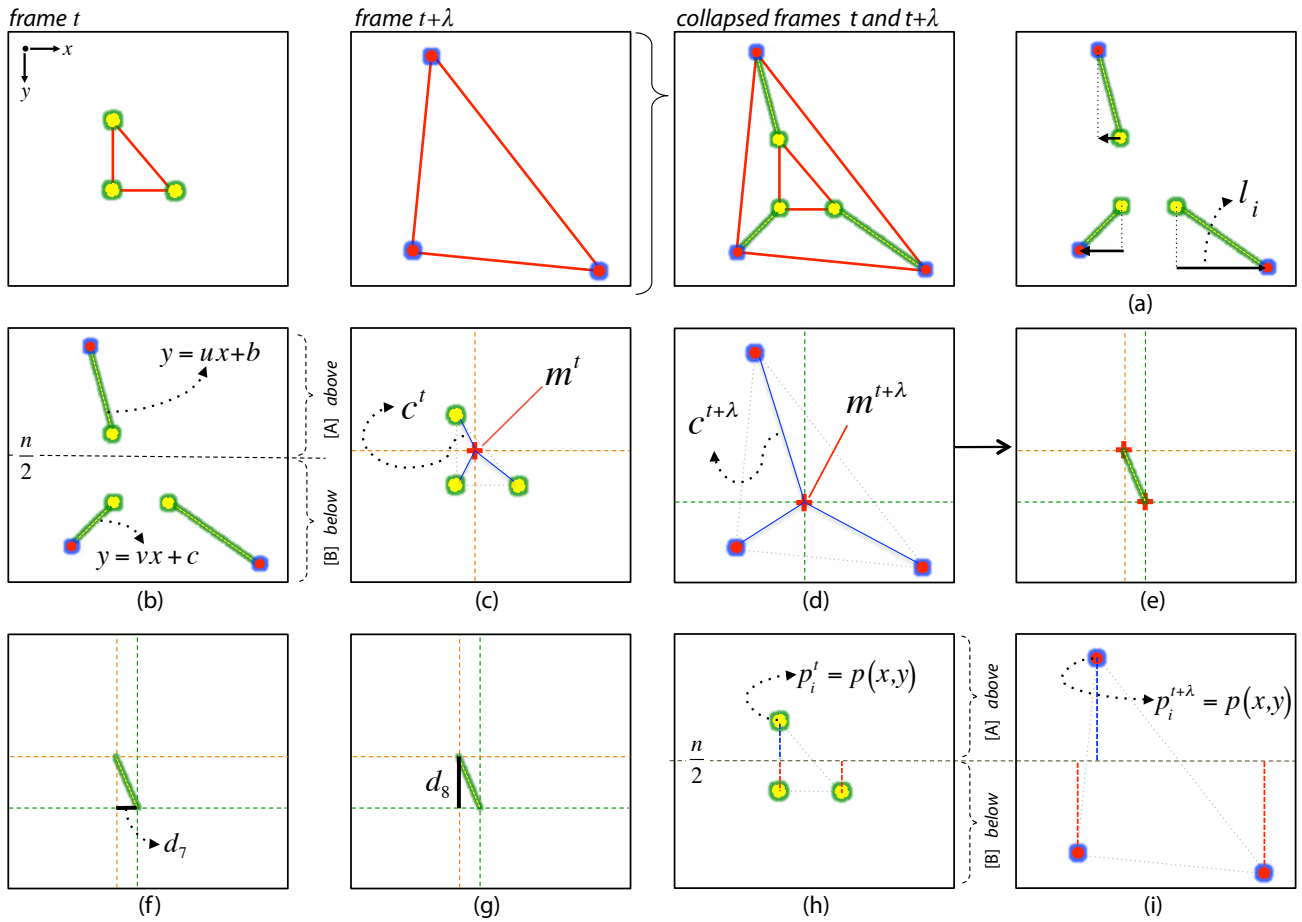


Fig. 5. Gradient behaviour analysis in multiple motions

The eight previously outlined descriptors ($d_3 \mapsto d_{10}$) allow for depth motion to be classified into the following categories: (1) forward, (2) backward, and (3) at rest.

C. Classification of Results

The preliminary classification yields a new descriptor that models the recognized motion, thus totalling three descriptors for horizontal and nine for depth motions. Using the previously mentioned classification algorithms requires calculating a large database, populated with all descriptors and several iterations of the various motions. Each proposed descriptor provides relevant information for interpreting changes throughout time, but using all of them does not contribute to motion classification, and may even duplicate information. Consequently, trials were needed to choose the best possible combination by feature selection.

D. Algorithm Training

The HMM algorithm learning process runs once and generates the probability matrixes needed for motion interpretation. Because there are two types of movements to be interpreted, learning must be run once for each. As previously stated, the descriptors generated differ according to the type of motion being analyzed, and the best combination of them is used in order to optimize results. Non-observable states of horizontal motion are: (1) leftward, (2) rightward and (3) at rest, and depth motion states are: (1) forward, (2) backward and (3) at rest. Unlike HMM, the kNN algorithm needs no training in classification, and hence, it must be run every time that a classification is required. Even though this may be counterproductive, its calculations are simple and do not affect the overall system performance. Both algorithms are used to interpret the descriptors generated for horizontal and depth motion, but only one must be chosen for interpreting both motions.

IV. Analysis of Results

A. Work Environment

The results shown in the following sections were always obtained with the same equipment, objects and work cubicle configuration. Selected objects are characteristically of a mainly uneven texture, so as to prevent that the algorithm detects insufficient keypoints for a good comparison, in reason of an even object surface.

B. SURF Thresholds

In practice, not many keypoints are necessary for motion recognition, normally requiring 70 to 100 in average for the algorithm to identify a gesture. Considering this, run times and the amount of keypoints kept upon filtering were analyzed, so as to shorten times. It must be noted that defining a very high threshold keeps less keypoints than the specified required range. In doing so, it must also be considered that the number of keypoints found is in direct proportion to the texture of the object in front of the camera, which may lead to finding an unacceptable number of keypoints. Run times are also proportionately shorter with lower threshold values, as less time is needed for keypoint detection and matching.

C. Run Times

The time shortening arrangement used enabled our system to process a sequence in real time. Final times also decrease as the SURF threshold is increased, and the kNN algorithm requires less processing times that HMM. This means that time is proportional to the number of frames extracted per second. Table I shows the system run times for processing one frame per second, including all five process phases.

TABLE I
FINAL SYSTEM RUN TIME

Threshold	Percentage retained	kNN	HMM
-	100%	0.1437	0.1451
100,000	79%	0.1153	0.1167
200,000	67%	0.0931	0,0945
300,000	45%	0.0778	0.0792
400,000	32%	0.0704	0.0718
500,000	23%	0.0642	0.0656

It must be considered that if more than eight frames are extracted per second without applying any threshold, processing times exceed the limit for real time.

D. System Performance

The number of frames to be extracted per video second must be defined. This number depends on the speed of the motion to be recognized, four being the recommended minimum for rather slow motion units, and eight for fast movements. Although the system is not restricted to this range of frames per second, these values actually yield the best performance results, both in respect to run and interpretation times, when classifying different types of motion.

The system performance measurement process begins with videos that are unknown to the system, so as to test its behaviour under unknown circumstances. Horizontal and depth motions were recorded, as well as a mixture of both and slight movements without any intentionality. Each recorded video lasted approximately 30 seconds. Since each second yields 30 frames, the test involved 900 frames in all. A more detailed explanation of the motions recorded is provided below.

- **Horizontal:** The prevailing arm moves to the left and to the right.
- **Depth:** Forward and backward motions are directed towards and away from the same object;
- **Robotic Motion:** All motion, whether horizontal or in depth, is in a straight line;
- **Motion Mix:** This is comprised of natural gestures, without any restrictions.

It must be taken into account that all motion is tridimensional, and therefore, always comprised of a horizontal coordinate, as well as a depth one, with one showing more change than the other. For instance, the horizontal coordinate of a leftward motion varies more than the depth coordinate, which should remain within the same range of values. The results presented in the following paragraphs were obtained without any threshold. A video was selected to show classification performance by category and to analyze the results obtained when extracting four, six and eight frames.

To assess the algorithm performance, an F-score was determined using recall and precision rates. The former is the number of correct results divided by the total number of cases classified in a given class, and the latter provides the total number of correct results for that class. Both rates are expressed in percentages and measure the algorithm performance. Their ratio is integrated as a single measurement:

$$F_{score} = 2 \times \frac{R \times P}{R + P}$$

where the ideal classification is attained when $F_{score} = 1$.

TABLE II
CLASSIFICATION WITH KNN

Category	Type of movement	Mov. classification	4 Frames	6 Frames	8 Frames
Horiz.	Horizontal	Leftwad	100%	99.10%	100%
		Rightward	97.78%	98.82%	96.43%
		Rest	99.55%	100%	99.09%
	Depth	Forward	92.68%	83.33%	90.00%
		Backward	80.00%	100%	100%
		Rest	98.64%	99.18%	99.70%
Depth.	Horizontal	Leftward	100%	100%	100%
		Rightward	22.22%	40%	0%
		Rest	96.30%	98.99%	99.75%
	Depth	Forward	76.96%	86.49%	90.48%
		Backward	30.77%	44.44%	72.00%
		Rest	98.14%	97.98%	99.11%
Robot	Horizontal	Leftward	100%	100%	100%
		Rightward	95.24%	93.02%	97.67%
		Rest	99.32%	99.36%	99.85%
	Depth	Forward	92.31%	97.44%	98.25%
		Backward	92.31%	96.77%	100%
		Rest	100%	99.34%	99.84%
Mix	Horizontal	Leftward	100%	100%	100%
		Rightward	100%	95.83%	98.04%
		Rest	100%	99.39%	99.79%
	Depth	Forward	95.83%	100%	100%
		Backward	92.31%	100%	100%
		Rest	98.55%	100%	100%

System performance results obtained from classifying four motion categories are presented in Tables II and III, with each motion having a horizontal and a depth coordinate. These results show that kNN performs better than the HMM algorithm,

TABLE III
CLASSIFICATION WITH HMM

Category	Type of movement	Mov. classification	$\lambda = 4$ frames	$\lambda = 6$ frames	$\lambda = 8$ frames
Horiz.	Horizontal	Leftward	55.17%	65%	80%
		Rightward	70%	69.88%	78.57%
		Rest	97.08%	95.91%	98.42%
	Depth	Forward	100%	100%	100%
		Backward	100%	100%	100%
		Rest	100%	100%	100%
Depth.	Horizontal	Leftward	100%	100%	100%
		Rightward	22,22%	40%	0%
		Rest	96.30%	98.99%	99.75%
	Depth	Forward	76,96%	86,49%	90,48%
		Backward	30.77%	44.44%	72.00%
		Rest	98.14%	97.98%	99.11%
Robot	Horizontal	Leftward	84.62%	91.89%	91.43%
		Rightward	64.52%	70.18%	73.68%
		Rest	93.57%	96.96%	98.15%
	Depth	Forward	73,47%	76,60%	88,89%
		Backward	32,00%	70,83%	88,70%
		Rest	98.64%	98.47%	99.35%
Mix	Horizontal	Leftward	90.32%	100%	100%
		Rightward	76,60%	82,14%	75.76%
		Rest	95.79%	96.86%	96.46%
	Depth	Forward	100%	98.55%	100%
		Backward	90.91%	66.67%	92.31%
		Rest	99.52%	99.07%	99.78%

when detecting motions comprised of depth and horizontal displacement. The results are derived from a measurement of classification by type of motion. Video processing results show an improvement when six or eight frames are extracted. The system could be strengthened by combining both classification algorithms, so as to increase the score rate. Finally, the system proved capable of detecting 98% of slight motions, which account for hand shaking or inactivity.

Classification results generally show that the kNN algorithm heightens system performance. However, it is important to consider that low performance outcomes in motion classification, below 70%, do not imply a significant error rate in the overall system's classification. These values are reviewed in more detail later in this paper. In sum, these results indicate that kNN yields better results, but the support provided by the HMM algorithm must not be discarded.

E. System Performance with different SURF Thresholds

The system performance curves obtained from defining different thresholds for SURF are presented below, together with an analysis of performance outcomes of extracting eight frames per video second ($\lambda = 8$) a number that leads to the best performance, according to the previously stated results. Additionally, performance is analyzed with a combination of kNN and HMM, given that the first showed better results.

Average performance percentages of motion classification {Leftward: LF, Rightward: RW, Backward: BW, Forward: FW, Horizontal Rest: HR and Depth Rest: DR} are shown in Fig.6, organized by type of motion and by applied threshold. Table IV shows the performance of each movement applied on different video motion and gesture.

TABLE IV
PERFORMANCE WITH $\lambda = 8$

		Video category			
	SURF	Horizontal	Depth	Robot	Mix
LW	100%	100%	100%	100%	100%
	79%	99%	100%	98%	97%
	67%	99%	100%	100%	100%
	45%	100%	100%	100%	100%
	32%	100%	100%	100%	98%
	23%	100%	100%	100%	100%
RW	100%	97%	100%	99%	97%
	79%	97%	25%	99%	92%
	67%	97%	67%	97%	89%
	45%	97%	70%	99%	92%
	32%	97%	45%	98%	87%
	23%	97%	54%	98%	88%
FW	100%	99%	86%	94%	93%
	79%	100%	93%	94%	98%
	67%	100%	93%	93%	98%
	45%	100%	94%	95%	97%
	32%	100%	92%	95%	97%
	23%	99%	95%	97%	97%
BW	100%	100%	68%	87%	82%
	79%	100%	76%	79%	89%
	67%	100%	82%	74%	89%
	45%	92%	73%	72%	84%
	32%	100%	72%	91%	84%
	23%	50%	58%	76%	45%
HR	100%	99%	100%	100%	100%
	79%	99%	100%	100%	100%
	67%	99%	100%	100%	99%
	45%	99%	100%	100%	99%
	32%	100%	100%	100%	99%
	23%	100%	100%	100%	99%
DR	100%	100%	99%	100%	99%
	79%	100%	99%	99%	100%
	67%	100%	99%	99%	100%
	45%	100%	99%	100%	100%
	32%	100%	100%	100%	100%
	23%	100%	100%	100%	100%

The analysis of leftward gestures yields classification results within similar levels for all categories, over 99%, while rightward motion detection shows better outcomes for Forward and Backward movements. With regard to depth, forward motion classification tends to improve as the number of keypoints found in an image decreases. The charts show an even and almost linear performance, always exceeding 98%, in respect of gestures at rest. This is basically due to the fact that there are no significant variations between frames that could affect descriptor generation and analysis.

F. SURF Threshold Impacts on Performance

To explain the cases where performance curves follow irregular patterns, we must review how octaves and thresholds work in the SURF algorithm. SURF uses octave filters, with each octave being a filter scaling level that uses a different size window to find keypoints with diverse contrast. According to Ehsan et al. [28], when no thresholds are defined for SURF, low contrast keypoints are predominantly found at the first octave levels, and higher octave levels contribute less keypoints with more contrast.

When no threshold is applied, the keypoints found in an image are ordered by octave, and thus, a larger amount of

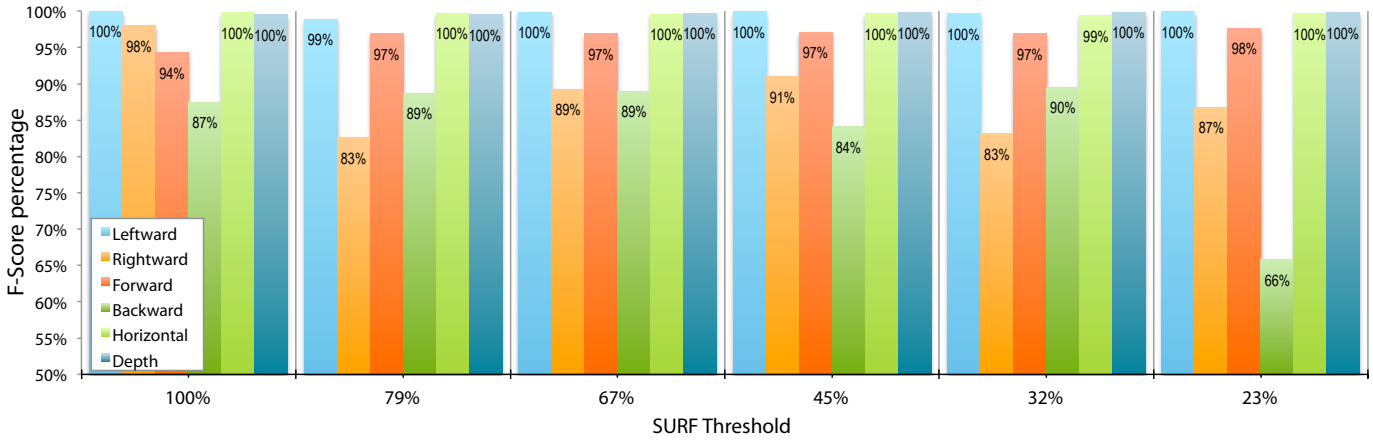


Fig. 6. Performance evaluation on multiple SURF Threshold with different type of motion

keypoints are found at the first octave and a lesser amount at each subsequent one [28]. It would be logical that the number of keypoints found by octave decreased proportionately to any threshold level applied. However, there are actual cases where higher octaves keep more points than the lower levels, or where the keypoints found concentrate at middle levels. This is due to the fact that threshold definition can discard keypoints that contribute information necessary for the system to generate robust matching descriptors. The reason behind this situation is that keypoints of importance for motion interpretation are eliminated at various octave levels. With less keypoints that do not sufficiently represent motion by themselves, the algorithm finds difficulty interpreting any such motion. As can be seen in Table IVBW, keypoint maintenance percentages show curves that are more stable at 79% and 67%.

G. Overall Performance of the Proposed Method

When different thresholds are applied with SURF, low contrast keypoints, which are found at the lower octave levels, are discarded. Normally, detected keypoints mostly concentrate at the lower octave levels, but the contrary can happen, or even that they converge at middle octave levels. When keypoints are discarded at the lower levels, frame information is lost, with the direct consequent impact on the system classification performance. Regardless of these unfavourable cases, system performance outcomes have proved encouraging, exceeding an 80% F-score for each type of motion. Average results are presented in more detail in Table V.

TABLE V
AVERAGE SYSTEM PERFORMANCE BY TYPE OF MOTION

	Performance
Leftward	95%
Rightward	85%
Forward	85%
Backward	80%
Rest	99%

From the results shown in Table V and considering unfavourable cases, it can be seen that the system has a high performance level in classifying any given instance.

V. CONCLUSIONS

This research has provided an active machine vision system for hand motion recognition, mainly when the gesture consists in grabbing or has a trajectory. We have shown that our solution, notably the proposed set of descriptors, yields efficient results in real time, through the use of a low resolution video and gray scale images. We want to emphasize the fact that our system works independently from the objects on the scene, since descriptor generation –on the basis of keypoint matching with SURF– enables usage of the algorithm in a variety of scenarios. The system could also be used to support a therapy or rehabilitation protocol, and even contribute to motion analysis in neurological studies.

In connection with the classification stage, two types of supervised classification algorithms, namely, kNN and HMM, were assessed, so as to get a performance benchmark. The former looks for the best distance from a given dataset to a space of neighbouring elements, while the latter requires a preliminary learning phase to be capable of finding the system parameters, specifically, transition probabilities. Even though both algorithms have different generation and classification processes, results showed that they both perform well in classifying the motions studied in this research.

The only factors that affect the matching process times are the number of keypoints found in an image and the size of the vector of descriptors created for each point. The process that interprets and classifies system generated descriptor data is the shortest of all. A direct analysis of system descriptors for output delivery by the pre classification module minimizes computing time for this process. The overall time of a frame processing by the system is generally reduced in proportion to the percentage of keypoints remaining when a SURF threshold is applied. It must be considered that the more frames extracted, the longer the processing time, which can even exceed the real time limit in some cases. In order to prevent

this from happening, the threshold value can be increased, always considering any system performance variations that may be entailed. Therefore, the system is capable of efficiently processing a video second when the number of frames defined for extraction is four to six per second, and even with eight frames in some cases.

There are several possibilities for run time improvement, of which the fastest to implement are:

- **Lowering the video resolution:** It is logical that lower resolution require shorter computing times. However, the SURF algorithm could find less keypoints, thus affecting the final system performance.
- **Changing the processing software:** MATLAB works with matrixes, but OpenCV is designed to work with images and videos, and could significantly reduce computing times.
- **Implementing a GPU Code:** More processors, namely graphic cards, would significantly shorten computing times, and allow for processing more frames per second and larger images.

For performance improvement in the horizontal category, new descriptors could be designed to provide support information, so as to increase the number of comparison points for a more accurate distinction of individual gestures. For instance, acceleration and difference in angles with regard to the perpendicular line of detected keypoints between a frame in t time and another in $t + \lambda$ could be considered. A way to increase robustness in the case of depth motion could be to take into account the eccentricity and size of the axes of a point cloud ellipse, as well as the rate of keypoints remaining in a frame per defined sector.

Finally, it must be noted that the delivered system and its tools are intended for improvement and implementation within a larger project. This research has yielded an efficient solution to the proposed problem, including several implementation options depending on the environment in which it is used, and may be supplemented by other systems to contribute to the science of active recognition.

Acknowledgment: This work has been supported by the National Commission of Science and Technology (CONICYT, Chile) under grant no. 11100098, and by the School of Information and Telecommunication Engineering, Faculty of Enginnering at Universidad Diego Portales.

REFERENCES

[1] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 3, pp. 257–267, March 2001.

[2] F. Duca, J. Fredriksson, and M. Fjeld, "Real-time 3d hand interaction: Single webcam low-cost approach."

[3] W. T. Freeman and M. Roth, "Orientation histograms for hand gesture recognition," in *In International Workshop on Automatic Face and Gesture Recognition*, 1994, pp. 296–301.

[4] L. Gupta, "Gesture-based interaction and communication: automated classification of hand gesture contours," *IEEE Transactions on Systems Man and Cybernetics Part C Applications and Reviews*, vol. SMC-9, no. 1, pp. 114–119, 2001.

[5] A. Bobick and J. Davis, "Real-time recognition of activity using temporal templates," in *Proceedings of the 3rd IEEE Workshop on Applications of Computer Vision (WACV '96)*, ser. WACV '96. Washington, DC, USA: IEEE Computer Society, 1996, pp. 39–42. [Online]. Available: http://dl.acm.org/citation.cfm?id=524178.836730

[6] K. K. Kim, K. C. Kwak, and S. Y. Ch, "Gesture analysis for human-robot interaction," in *The 8th International Conference on Advanced Communication Technology*, I. C. on Advanced Communication Technology, Ed., vol. 3, 2006, pp. 1824–1827.

[7] M. Bray, E. Koller-Meier, and L. Van Gool, "Smart particle filtering for high-dimensional tracking," *Comput. Vis. Image Underst.*, vol. 106, pp. 116–129, 2007.

[8] M. Bray, E. Koller-Meier, N. N. Schraudolph, and L. Van Gool, "Fast stochastic optimization for articulated structure tracking," *Image Vision Comput.*, vol. 25, pp. 352–364, March 2007.

[9] E. Perini, S. Soria, A. Prati, and R. Cucchiara, "Facemouse: A human-computer interface for tetraplegic people," in *ECCV Workshop on HCI 2006*, vol. LNCS 3979, 2006, pp. 99–108.

[10] H. Prendinger, A. Hyrskykari, M. Nakayama, H. Istance, N. Bee, and Y. Takahasi, "Attentive interfaces for users with disabilities: eye gaze for intention and uncertainty estimation," *Universal Accsess in the information society*, vol. 8, no. 4, pp. 339–354, 2009.

[11] T. de Campos, W. Mayol, and D. Murray, "Directing the attention of awearable 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.

[12] M. Bray, E. Koller-meier, P. Müller, L. V. Gool, and N. N. Schraudolph, "3d hand tracking by rapid stochastic gradient descent using a skinning model," in *In 1st European Conference on Visual Media Production (CVMP)*, 2004, pp. 59–68.

[13] J. Martin and J. L. Crowley, "An appearance-based approach to gesture-recognition," in *Proceedings of the 9th International Conference on Image Analysis and Processing-Volume II*. London, UK: Springer-Verlag, 1997, pp. 340–347.

[14] R. Bajcsy, "Active perception," *Proceedings of the IEEE*, vol. 76, no. 8, pp. 966–1005, Aug 1988.

[15] J. Aloimonos, "Purposive and qualitative active vision," in *10th International Conference on Pattern Recognition*, vol. 1, 1990, pp. 346–360.

[16] W. W. Mayol, B. Tordoff, and D. W. Murray, "Towards wearable active vision platforms," in *in IEEE Sys. Man and Cybernetics Conf*, 2000, pp. 1627–1632.

[17] A. J. Davison, W. W. Mayol, and D. W. Murray, "Real-time localisation and mapping with wearable active vision," in *Proceedings of the 2nd IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'03)*, 2003, pp. 18–27.

[18] T. Kurata, N. Sakata, M. Kourogi, H. Kuzuoka, and M. Billinghurst, "The advantages and limitations of a wearable active camera/laser in remote collaboration," in *Proceedings of the Computer Supported Cooperative Work, CSCW'04*, Chicago, IL, 2004.

[19] D. C. C. Tam, "Surf: Speeded up robust features," CRV Tutorial Day 2010, Ryerson University, 2010.

[20] H. Bay, B. Fasel, and L. Van Gool, "Interactive museum guide: Fast and robust recognition of museum objects," *Proceedings of the first international workshop on mobile vision*, 2006.

[21] A. N. Pina, "Clasificación y búsqueda de imágenes usando características visuales," Master's thesis, Facultad de Informática, Universidad de Murcia, June 2010.

[22] C. G. Cambronero and I. G. Moreno, "Algoritmos de aprendizaje: Knn y kmeans," Master's thesis, Universidad Carlos III de Madrid, 2006.

[23] A. Moujahid, I. Inza, and P. Larrañaga, "Clasificadores k-nn," Master's thesis, Universidad del País Vasco–Euskal Herriko Unibertsitatea, 2008.

[24] B. Resch, E. Rank, and C. Rank. Hidden markov models a tutorial for the course computational intelligence.

[25] A. Mesa, "Modelos markovianos para secuencias y aplicaciones a la predicción de genes," Master's thesis, Universidad de la República, November 2010.

[26] H. Koo, "Forward algorithm, baum-welch algorithm."

[27] D. G. Lowe, "Object recognition from local scaleinvariant features," in *International Conference on Computer Vision*, Corfu, Greece, 1999, pp. 1150–1157.

[28] S. Ehsan, N. Kanwal, E. Bostanci, A. F. Clark, and K. D. McDonald-Maier, "Analysis of interest point distribution in surf octaves," in *The 3rd International Conference on Machine Vision (ICMV)*, December 2010.