Human–computer interaction based on visual hand-gesture recognition using volumetric spatiograms of local binary patterns

Ana I. Maqueda , Carlos R. del-Blanco, Fernando Jaureguizar, Narciso García

ABSTRACT

A more natural, intuitive, user-friendly, and less intrusive Human–Computer interface for controlling an application by executing hand gestures is presented. For this purpose, a robust vision-based hand-gesture recognition system has been developed, and a new database has been created to test it. The system is divided into three stages: detection, tracking, and recognition. The detection stage searches in every frame of a video sequence potential hand poses using a binary Support Vector Machine classifier and Local Binary Patterns as feature vectors. These detections are employed as input of a tracker to generate a spatio-temporal trajectory of hand poses. Finally, the recognition stage segments a spatio-temporal volume of data using the obtained trajectories, and compute a video descriptor called Volumetric Spatiograms of Local Binary Patterns (VS-LBP), which is delivered to a bank of SVM classifiers to perform the gesture recognition. The VS-LBP is a novel video descriptor that constitutes one of the most important contributions of the paper, which is able to provide much richer spatio-temporal information than other existing approaches in the state of the art with a manageable computational cost. Excellent results have been obtained outperforming other approaches of the state of the art.

1. Introduction

Recently, hand-gesture recognition systems based on vision have undergone an increasingly popularity due to their wide range of potential applications in the field of Human–Computer Interaction (HCI), such as multimedia application control [1], video-games [2], and medical systems [3]. These interfaces are considered more natural, intuitive, friendly, and less intrusive for the user than traditional HCI devices (mouse, keyboard, remote control, etc.). Although the use of keyboard and mouse can be still very useful for some applications, there are situations/applications where hand-based interfaces can be a key advantage. On the other hand, the fact that most of consumer devices are supplied with color cameras has also motivated the growth of HCI systems based on hand-gesture recognition and the design of color-based approaches.

In spite of the great body of works in hand gesture recognition, there are still some challenges affecting its performance. Recognizing a hand, and characterizing its shape and motion in images or videos is a complex task. The hand dynamics is highly complex because of its articulable nature with more than 25 degrees of freedom. This fact makes to model the different poses and motions very difficult. In addition, the appearance of a hand can change dramatically because of illumination changes, scaling, blurring, orientations, and occlusions. On the other hand, intraclass and interclass variance of the gestures are very high. The same action performed by the same individual several times is slightly different, and this problem gets worse if the same action is performed by two different individuals. Finally, since gestures typically appear within a continuous stream of motion, a temporal segmentation for determining when they start and end is necessary.

Two of the most popular approaches for hand-gesture recognition are based on machine learning approaches and template matching, using both color-based and depth-based imagery. They are used together with feature descriptors, in order to perform the recognition task. The works in [4] and [5] describe the hand pose by its contour shape, and then they perform the gesture classification by using template matching through a shape distance metric called Finger-Earth Mover's Distance (FEMD). In [6] the depth map of a hand pose is transformed to a point cloud, which is characterized by the Ensemble of Shape Function (ESF) descriptor, and then it is classified by multilayered random forest (MLRF). The work in [7] presents an interactive finger-spelling graphical user interface based on American Sign

Language (ASL). The hand shape features are based on Gabor filters of the intensity and depth images, and the classification task is carried out by multi-class random forest. Finally, the work in [8] presents a framework based on a 2D volumetric shape descriptor that is delivered by a SVM for hand posture classification using depth imagery.

A common issue of all the previous works is the use of descriptors, which must be able to represent the image region in a reliable way independently of the scene conditions. For this purpose, they should be invariant to translations, rotations, scale changes, and dramatic illumination changes. On the other hand, it is desirable that they have a reduced dimension in order to achieve a high computational efficiency. Therefore, it is necessary to find a good trade-off between recognition accuracy and computational efficiency.

On the other hand, hand gestures are intrinsically dynamic, i.e. they vary with the time dimension. For this reason, a hand gesture descriptor should take this information into account. For this purpose, some video descriptors have been developed [9], however most of them are focused on human action recognition [10,11]. Using these techniques for hand gesture recognition is not totally suitable because the durations of gestures are much shorter than human activities, which negatively impacts their effectiveness. Furthermore, the extraction of these features is generally slow with a reduced amount of global spatial information, and they do not offer a scalable solution for efficient matching when the database is large. Other descriptors include motion trajectories [12], spatio-temporal gradients [13], and global histograms of optical flow [14]. However, the comparison of existing methods is often limited given the different range of used experimental settings.

Therefore, a novel and highly discriminative video descriptor, which is called Volumetric Spatiograms of Local Binary Patterns (VS-LBP), has been designed for hand-gesture recognition in color imagery. It is not only computationally efficient and robust to dramatic illumination changes, but also provides much richer spatio-temporal information (at local and global levels) than other descriptors. This video descriptor has been integrated into a hand-gesture recognition system to provide a more natural and intuitive Human–Computer Interaction (HCI) interface. The proposed recognition framework has been tested to simulate a mouse-like pointing device to interact with a computer as an example of application. For this purpose, a new database has been created, which contains specific hand gestures to control the different mouse functionalities. Excellent results have been obtained regarding other approaches based on depth only or both color and depth.

The rest of the paper is structured as follows. Section 2 explains the designed video descriptor VS-LBP. Section 3 describes the proposed hand-gesture recognition system, explaining in detail each one of its stages: hand pose detection, hand pose tracking, and hand gesture recognition. Section 4 describes the proposed database and presents the experimental results obtained for the VS-LBP descriptor and the global system. Finally, Section 5 summarizes the conclusions of this work.

2. Volumetric spatiograms of local binary patterns (VS-LBP)

In the last years, the LBP descriptor [15] has been successful in several applications, such as texture recognition [16], face detection/recognition [17], and facial expression recognition [18], due to its powerful characteristics. However, the fact that the LBP descriptor does not consider any global spatial information is a disadvantage in the case of hand gestures. Textures can be seen as a set of patterns that recur several times. As for faces, they are formed by a uniform surface (forehead skin, cheeks, and chin), and by four patterns that do not greatly change their relative positions (two eyes, a nose, and a mouth). In all these cases, the global spatial information is not very determining since the appearance only can change in a limited and controlled way. However, a hand is a deformable object with more than 25 degrees of freedom, and its appearance can change largely. The hand patterns do not spread out uniformly as in textures, nor they are located in specific areas of the image as in faces. For hand poses, knowing what part of the image the patterns come from is as important as knowing the type of patterns, and the number of times that they appear.

Regarding the LBP extensions to include temporal information, there are two main ones: VLBP and LBP-TOP descriptors [19]. They essentially present the same problem as the LBP descriptor presents for describing static hand poses: they do not consider global spatial information. This information is vital since now, the hand appearance not only changes dramatically in the spatial domain, but also changes in the temporal domain. In addition, a hand gesture performed by an individual can differ significantly from the same hand gesture performed by other different individual. Therefore, the lack of localized patterns (global spatial information) turns into a more complex problem than before. On the other hand, the feature extraction process in VLBP and LBP-TOP produces already high dimensional feature vectors, and therefore adding spatial information to these descriptors to obtain more discriminative features, would be prohibited in terms of computational cost, and memory requirements.

The VS-LBP descriptor is a major extension of the LBP descriptor [15] to achieve reliable and compact representations from video sequences containing hand gestures. It includes global spatial information to be more discriminative by identifying from what part of the image the local binary patterns come, and temporal information to deal with dynamic hand gestures. The algorithm to compute the VS-LBP can be divided into the three following steps: Multi-scale LBP computation, S-LBP computation and Temporal sampling, which are described in the following sub-sections.

2.1. Multi-scale LBP computation

The first step consists of computing the Multi-scale LBP descriptor [15] from every image region. This descriptor is a variation of the LBP operator that was originally designed for texture description [20]. The LBP operator thresholds a 3 × 3 neighborhood by the intensity value of the center pixel in order to extract local spatial structures from an image region. The thresholded values are concatenated in an 8-bit binary number, and converted to decimal for a more compact representation. Finally, they are used to generate a histogram of $2^8 = 256$ labels. Fig. 1 summarizes the computation of the LBP. The Multi-scale LBP descriptor extends the capabilities of the LBP to deal with patterns at different scales by using neighborhoods of different sizes. The new neighborhood pattern is defined as a set of sampling points evenly spaced on a circle centered at the pixel to be labeled, as shown in Fig. 2.

The notation for defining this operator is $\mathrm{LBP}_{\mathrm{P,R}}$, where P means number of sampling points on a circle of radius R. The mathematical expression to obtain a label from $\mathrm{LBP}_{\mathrm{P,R}}$ is:

$$LBP_{\mathrm{P,R}} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p, \tag{1}$$

where g_c corresponds to the gray value of the center pixel of the local neighborhood, $g_p(p = 0, \ldots, P-1)$ corresponds to the gray values of the P equally spaced sampling points on the circular neighborhood, and $s(x)$ is the sign function defined as:

$$\mathrm{s(x)} = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0. \end{cases} \tag{2}$$

Bilinear interpolation is used whenever a sampling point does not fall in the center of a pixel.

As a result, an image of local binary patterns is obtained, as shown in Fig. 3. Then, the Histogram of Local Binary Patterns (H-LBP) is computed.

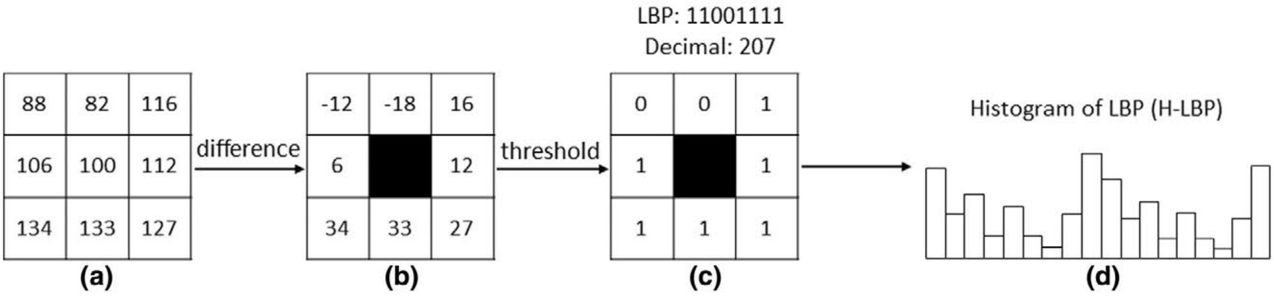

Fig. 1. Local binary pattern (LBP) from a pixel neighborhood. (a) 3 × 3 gray scale neighborhood. (b) Differences between the neighbor pixels and the center one. (c) Thresholded neighborhood differences. (d) Histogram of LBP (H-LBP) from the whole image.

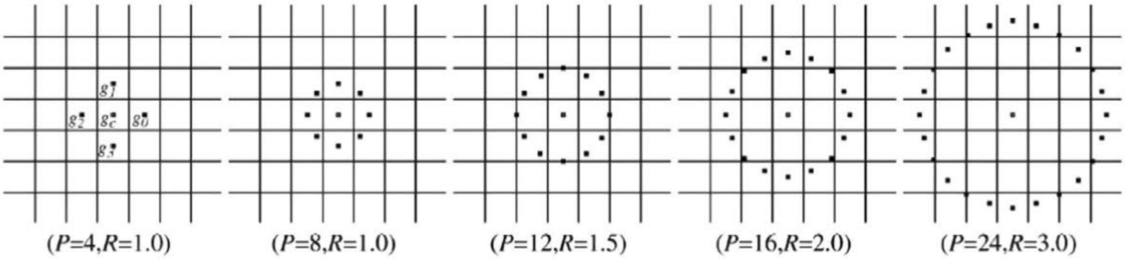

Fig. 2. Circularly symmetric neighbor sets for different P and R (extracted from [15]).

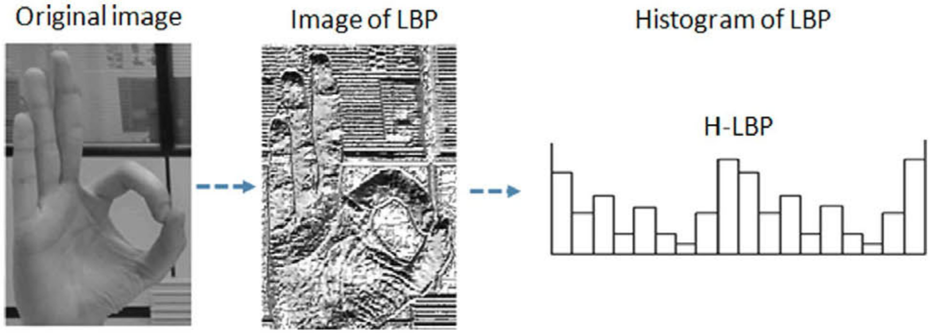

Fig. 3. Step 1: H-LBP computation.

2.2. S-LBP computation

The second step consists of extracting spatial information from the image of LBPs, as shown Fig. 4. First, the coordinates of all the LBP patterns that have contributed to a specific bin in the H-LBP histogram (representing a specific LBP type) are computed. From the algorithmic viewpoint, this computation is not necessary as it is previously performed during the multi-scale LBP computation. Second, a uniform sub-sampling of the image region coordinates is carried out, obtaining a total of $M \times N$ sampled coordinates, defining M as the number of rows, and N as the number of columns. The set of coordinates of each LBP bin contributes to one histogram of $M \times N$ sampled coordinates, which are called $S_0, S_1,..., S_{M\times N-1}$ in Fig. 4, using a bilinear interpolation. This way, a histogram of spatial coordinates is generated per each LBP bin of the computed H-LBP (spatial histograms). As a result, we obtain 2^P spatial histograms whose length is $M \times N$, where P was the number of neighbors in the $LBP_{P,R}$. The H-LBP itself and the set of spatial histograms are all concatenated to form a super-descriptor called Spatiogram of Local Binary Patterns (S-LBP), whose dimension is $2^P + [2^P \times (M \times N)]$.

The S-LBP descriptor is highly discriminative since it contains both local (the H-LBP) and global spatial information (histograms of spatial coordinates of all the LBP patterns). The uniform sub-sampling of the image coordinates allows to shorten the histograms length and keep the computational cost manageable, establishing a trade-off between the computational cost and the discrimination ability. On the other hand, the bilinear interpolation approach increases the robustness against slight image translations, and the grid effect.

2.3. Temporal sampling

The last step consists of adding temporal information to the S-LBP framework by carrying out a randomly and quasi-equally temporal sampling scheme in the video sequence. Close images in time hardly change their appearance, containing redundant information to identify the action that is being performed. This strategy also allows to deal with variations in the execution speed of the hand gestures by considering several sampling steps.

The randomly and quasi-equally spaced sampling is carried out as follows. An additive random shift is applied to those images corresponding to an equally spaced sampling in the temporal dimension defined by Δ_e, as shown in Fig. 5.

The random shifting is performed following a discrete uniform distribution over the considered maximum interval Δ_{max}. Once all the sampled images have been obtained, the S-LBP descriptors from those selected images are concatenated to form Volumetric Spatiograms of Local Binary Patterns.

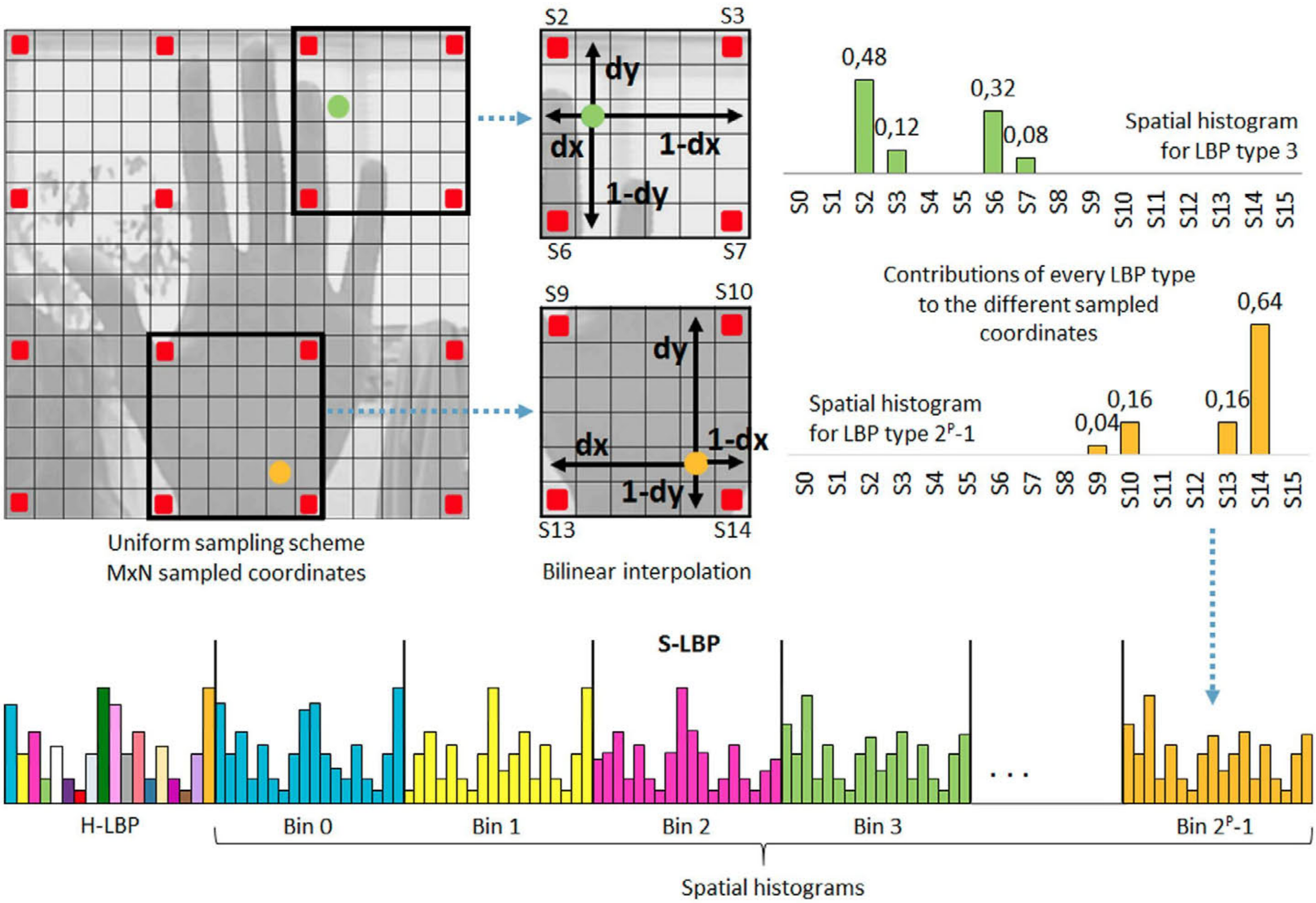

Fig. 4. Step 2: S-LBP computation. Red dots correspond to the $M \times N$ sampled coordinates, and the colored dots are pixel examples. The LBP computed from the green pixel and the LBP computed from the orange pixel contribute to the bins 3 and $2^P - 1$ of the H-LBP histogram respectively. The coordinates of each LBP type contribute to the $M \times N$ sampled coordinates using a bilinear interpolation. A histogram of spatial coordinates is generated per each LBP bin of the H-LBP (spatial histograms). LBPs of the same type contribute to the same spatial histogram. Finally, the resulting spatial histograms are concatenated along with the H-LBP computed in the first step, generating the S-LBP descriptor. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

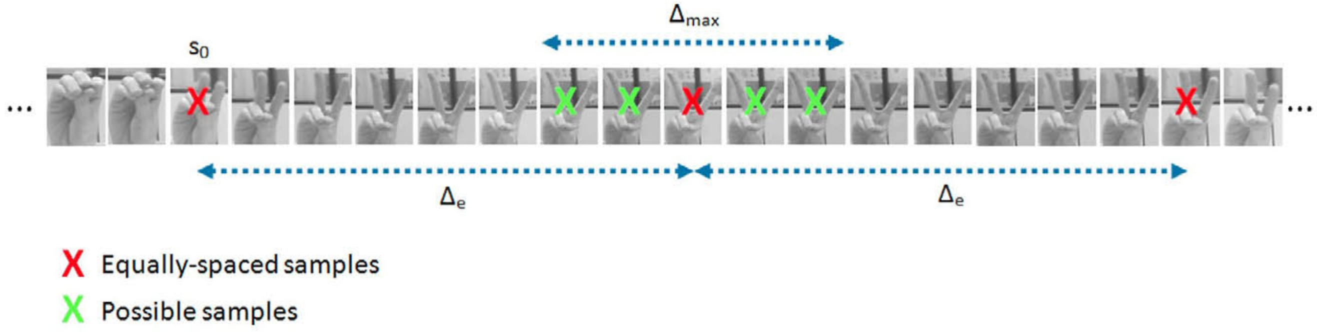

Fig. 5. Image sequence sampling and its parameters.

Therefore, VS-LBP descriptor includes spatio-temporal information in an efficient and compact way. On the one hand, local and global spatial information is added by means of S-LBP, which increases the discriminative power. On the other hand, the temporal sampling strategy allows to consider a smaller number of frames for computation and also reduce the computational cost. Moreover, this is a versatile approach since it can be used together with any image descriptor to compute the spatial features. But, we have to keep in mind that the overall length of the final video descriptor depends on the length of the image descriptor and the number of sampled images.

3. System description

The proposed HCI interface has been tested to simulate an example of application, in particular, a mouse device to interact with a computer by means of hand gestures. In this sense, a robust hand-gesture recognition system has been implemented, and a new database has been created, which contains hand gestures based on mouse functionalities. Nevertheless, the presented vision-based recognition system can be integrated into other multimedia devices provided with a color camera, such as smartphones and televisions, and extended to other applications by increasing

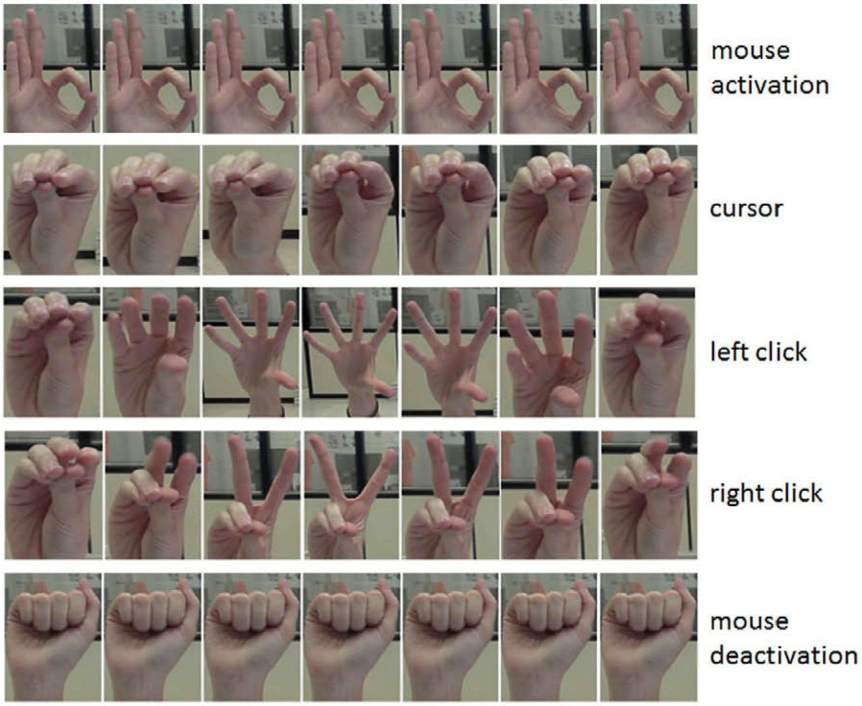

Fig. 6. Proposed hand gestures (visual examples).

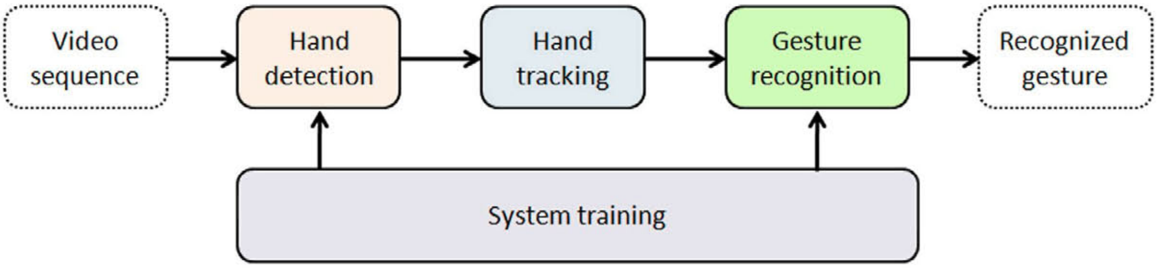

Fig. 7. Block diagram of the proposed hand-gesture recognition system.

the database to consider new hand gestures, and therefore, new functionalities.

The proposed database contains a set of hand gestures that represent the main functions of a mouse device, such as *cursor*, *left click* and *right click*; and two additional functions from the viewpoint of the application, such as *mouse activation* and *mouse deactivation*. As a result, five different hand gestures are considered to simulate a mouse device and interact with a computer, which can be seen in Fig. 6. Notice that the hand gestures proposed to activate and deactivate the mouse application are static, since they keep the same appearance, orientation and position along the time. However, they are considered as dynamic taking into account that they have to persist a determined period of time to be recognized.

On one hand, the proposed hand-gesture recognition system is based on machine learning techniques and feature extraction methods. In particular, SVM classifiers have been used because they work very well with high-dimensional data, and are capable of delivering high performance in terms of classification accuracy. On the other hand, they allow us to deal with non-linear boundaries by means of different kernels, which makes it more adaptable.

The system is composed of three stages: detection of hand poses, tracking of detected hand poses (Fig. 9), and recognition of dynamic hand gestures. The detection phase employs the LBP descriptor [15] and a binary SVM classifier to detect specific hand poses in every frame. The tracking phase uses those detections as input for a multiple object tracker that estimates potential trajectories of hand poses along time. These trajectories contain an ordered set of hand poses that performs different dynamic hand gestures. Finally, the recognition phase analyzes those trajectories by computing high efficient spatio-temporal features called VS-LBP, which are delivered to a bank of SVM classifiers for gesture recognition. A block diagram of the system can be seen in Fig. 7.

3.1. Hand pose detection

The aim of the detection stage is to detect those hand poses that can be part of the considered dynamic hand gestures. These specific hand poses (positive class) must be recognized among other irrelevant hand poses and background (negative class).

To detect hand poses in different spatial locations and scales, every frame is scanned by a spatial sliding window at multiple scales. Fig. 8 shows this strategy. The multi-scale analysis is carried out by generating a multiresolution pyramid, which contains different scales of the frame that is being processed. Then, a fixed rectangular window is slid along each scale so that the multiple windows are overlapped. The overlapping magnitude between consecutive windows is determined by the spatial step of the sliding window. The choice of this parameter is a trade-off between the computational cost and the accuracy of the detection. This strategy has been chosen instead of

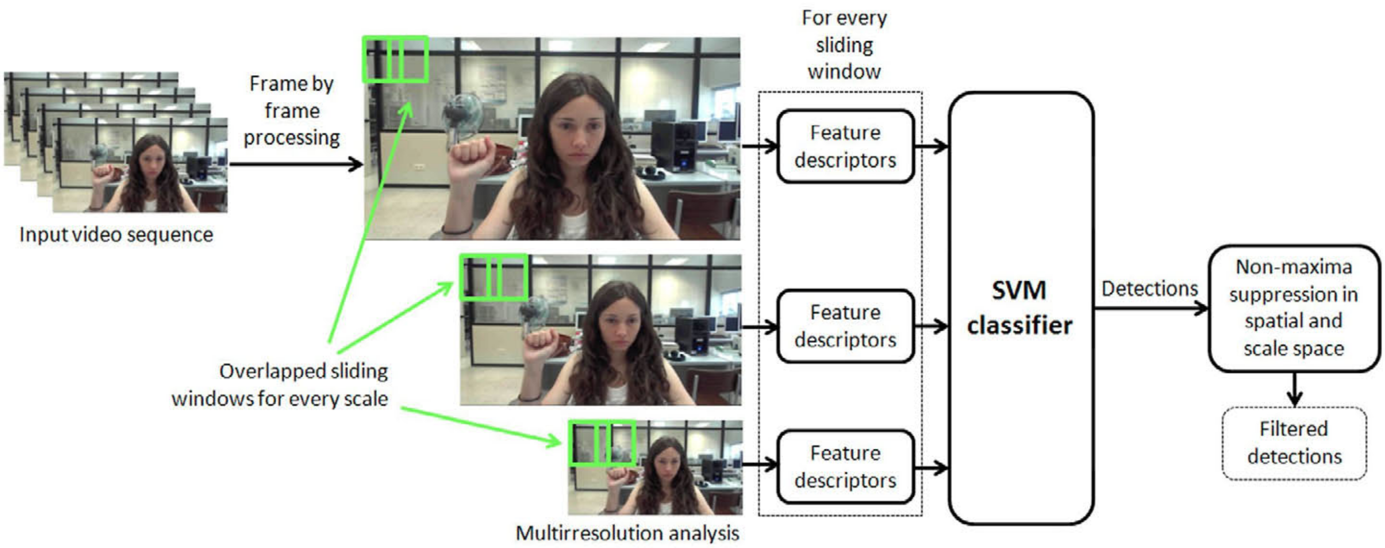

Fig. 8. Hierarchical sliding window strategy in the detection phase.

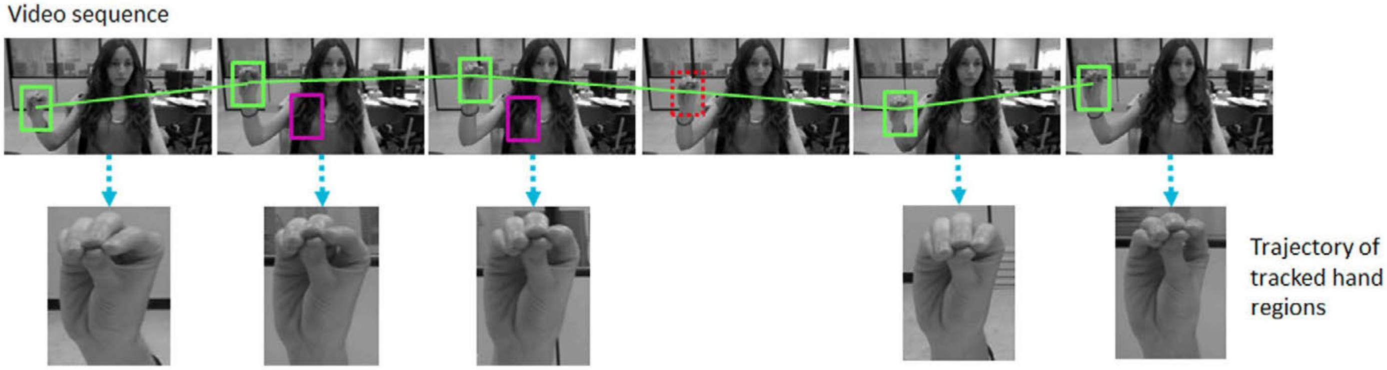

Fig. 9. Tracking of detected hand poses. Green image regions correspond to hand poses correctly detected from which a trajectory of tracked hand regions is extracted. The tracker is robust against missing detections (red image regions). On the other hand, typical false detections (purple image regions) do not usually generate a trajectory because of its short length. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

sliding several windows with different sizes applied to an only scale, because the computational cost is lower.

This way, the LBP image descriptor [15] is applied to every spatial window for feature extraction. It computes a feature vector that represents the image region, being robust against dramatic illumination changes, and very computationally efficient. Then, every feature vector feeds a classifier, which determines if that image region is a positive or a negative sample.

Since this stage has to deal with two classes, hand poses (positive class) and background (negative class), a binary SVM classifier is used for detection. In order to achieve a higher performance, a Hellinger kernel, more commonly known as Bhattacharyya coefficient [21], has been used. It allows to learn non-linear decision boundaries by projecting the features in a higher dimensional space, where linear boundaries can be computed to separate the different classes. It can be mathematically represented as:

$$k(f, f') = \sum_i \sqrt{f(i) f'(i)}, \tag{3}$$

where f and f' are normalized feature vectors.

At this point, potential hand poses are detected, however they need to be filtered because of windows overlapping. Since several overlapping windows belonging to one specific scale could contain a significant fraction of a relevant hand pose, all of them could be labeled as positive samples. In order to select the one that better represents the hand pose, a non-maxima suppression technique is applied [22] in every scale. However, the problem persists among different scales of the multiresolution pyramid. For this reason, another non-maxima suppression technique based on an overlapping criterion is applied to normalized windows from different scales: if the overlapping of two normalized windows (i.e. normalized to a common scale) is bigger than a specific threshold, the window that presents the highest classification score is labeled as a positive sample, and the other one as a negative sample.

Finally, a set of filtered detections is obtained in every frame, which is used as input of the tracking phase.

3.2. Hand pose tracking

The goal of the tracking phase is to estimate temporal hand trajectories from the detected hand poses. When the first frame is processed in the detection phase, the obtained detections are used as input of the tracker, which will create as many trajectories as the number of detected hand poses in the frame. This way, every time a frame is processed, new detected hand poses are associated to their corresponding trajectories according to the location of detections in previous frames.

Since there can be missing detections due to occlusions, strong changes in the hand appearance, and also false detections generated by background structures, the estimation of the hand locations can be inaccurate. The hand-detection identities can be interchanged, as well, due to erroneous associations between detections and

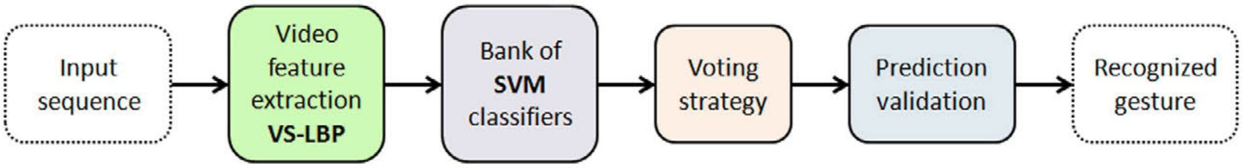

Fig. 10. Block diagram of the hand-gesture recognition phase.

trajectories. To deal with these problems, a multiple object tracker which is robust to erroneous, distorted, and missing detections [23] has been used. This tracker is based on a constant velocity model for predicting the object locations, and a soft probabilistic data association that recursively estimates the best correspondence between measurements/detections and existing objects/trajectories.

As a result, one or several trajectories are generated depending on the number of detected hand poses in every frame. Every trajectory can be seen as a cropped video sequence, which only contains hand poses. This volume of ordered hand poses is used as input of the recognition phase to be analyzed.

3.3. *Gesture recognition*

The goal of the recognition stage is to temporally segment the trajectories that contain an ordered set of hand poses, and recognize the dynamic hand gestures that are executed in every temporal segment. In this case, five different dynamic hand gestures (five classes) must be distinguished, therefore a set of five SVM classifiers is used for recognition, where everyone is trained to recognize a specific dynamic hand gesture. The same kernel as in the detection phase is used to learn non-linear decision boundaries (see Eq. (3)). The VS-LBP descriptor is used for feature extraction from every temporal segment. It is robust against dramatic illumination changes, and variations in the execution of the dynamic hand gestures. In addition, it contains spatio-temporal information in an efficient and compact way that makes it highly discriminative regarding other video descriptors. Fig. 10 shows a block diagram of this phase.

In order to determine where a specific dynamic hand gesture starts and ends, that is, in which temporal segments is performed, a sliding temporal window scans every trajectory by overlapping consecutive temporal segments. The overlapping between consecutive temporal windows or segments is a trade-off between the computational cost and the accuracy of the recognition. On the other hand, the size of the temporal window has to be fixed according to the average number of frames of the different dynamic hand gestures.

While processing frame by frame, once a trajectory contains the same number of cropped hand poses as the size of the temporal window, the sliding of the temporal window starts. For every temporal window, several VS-LBP feature vectors are computed. Every feature vector is slightly different because of the random temporal sampling scheme (see Section 2), which allows to adapt the system to different slight variations in the hand gesture execution, increasing the recognition accuracy.

This set of feature vectors, all associated to the same temporal window, are individually classified as belonging to a specific hand gesture class. Ideally, all of them should belong to the same class, but in practice there can be different recognized gestures due to the intra-class and inter-class variability. For this reason, a voting scheme is used to label the dynamic hand gesture as the most recognized class, as shown Fig. 11. This process is repeated for each trajectory that has at least the same size as the temporal window, discarding erroneously short trajectories.

The final step is a temporal validation of the predicted dynamic hand gesture, as shown in Fig. 12. The condition that the same prediction should be consistent over a determined number of consecutive windows is imposed. The reason is that if the step size is enough small, the windows will differ in a few frames, and should contain the same dynamic hand gesture. This strategy solves potential errors due to gestures transitions and erroneous estimated trajectories.

3.4. *System training*

Both detection and recognition phases have to be trained to estimate the optimal parameters for a binary SVM and for five SVM classifiers, respectively. The proposed database contains 30 video sequences for training (see Section 4.1), which are used to train both detection and recognition stages, but in different ways.

In the training stage of the detection phase, image regions containing hand poses that are part of the five dynamic hand gestures are used as positive samples. They are independently extracted from every frame of the video sequences, without any consideration if they belong to a specific gesture or another since all of them are just hand poses that we are interested in detecting. On the other hand, image regions containing background and other irrelevant hand poses are used as negative samples to train the classifier. Spatial windows that tightly surround the object are used to train the classifier, this way a small spatial step for sliding must be considered.

In the recognition stage, training samples from every gesture are the own video sequences, where every video sequence contains several repetitions of a hand gesture. Every training video sequence is spatially and temporally segmented, so that, training samples from every gesture are cropped video sequences that perform one repetition of the hand gesture, whose frames are image regions containing only the hand poses. The way of training the SVM classifiers is following a *one-vs-all* strategy, where each classifier is trained by considering as positive samples all the sequences from the class that is being trained, and as negative samples sequences belonging to the rest of classes.

In order to generate a larger number of training samples, the VS-LBP descriptor can be applied to every training sequence several times, taking advantage of its random temporal sampling scheme (see Section 2). Every computation of the video descriptor produces different feature vectors that should be strongly correlated, that is, they should theoretically belong to the same cluster in the feature space.

4. Experimental results

4.1. *Database*

A new visual database has been created to validate the hand-gesture recognition system and imitate a mouse device, which is called hand-gesture database (Set 2) and is publicly available on http://www.gti.ssr.upm.es/data/HandGesture_database.html. To that end, five dynamic hand gestures are proposed to carry out different mouse functions: *cursor*, *left click*, *right click*, *mouse activation*, and *mouse deactivation*, which can be seen in Fig. 6.

The database contains 30 video sequences for training, which are performed by 6 different individuals. Each individual executes five video sequences in which a different dynamic hand gesture is performed several times. These 30 video sequences are used to train both

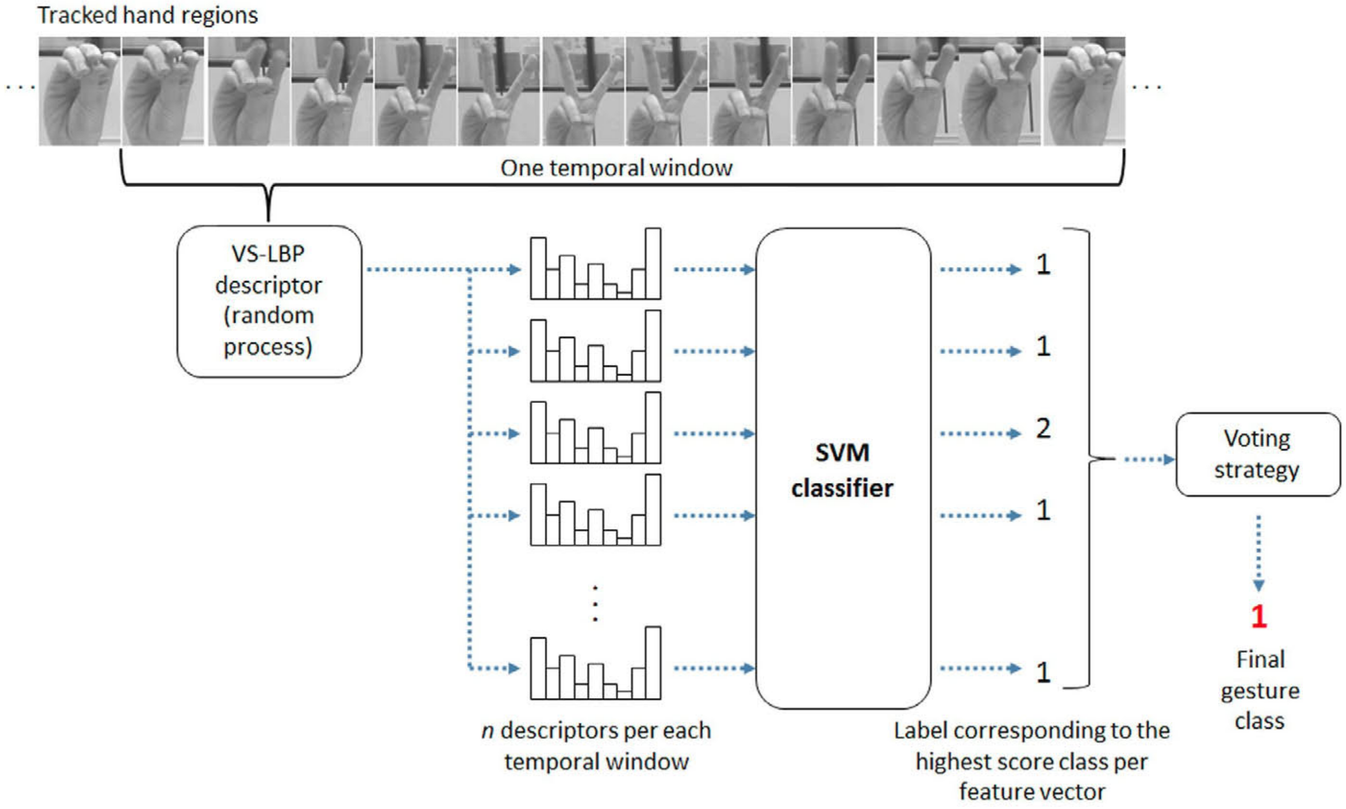
Tracked hand regions
One temporal window
VS-LBP descriptor (random process)
SVM classifier
1
1
2
1
1
Voting strategy
1
Final gesture class
n descriptors per each temporal window
Label corresponding to the highest score class per feature vector

Fig. 11. Feature extraction and gesture recognition for a given temporal window.

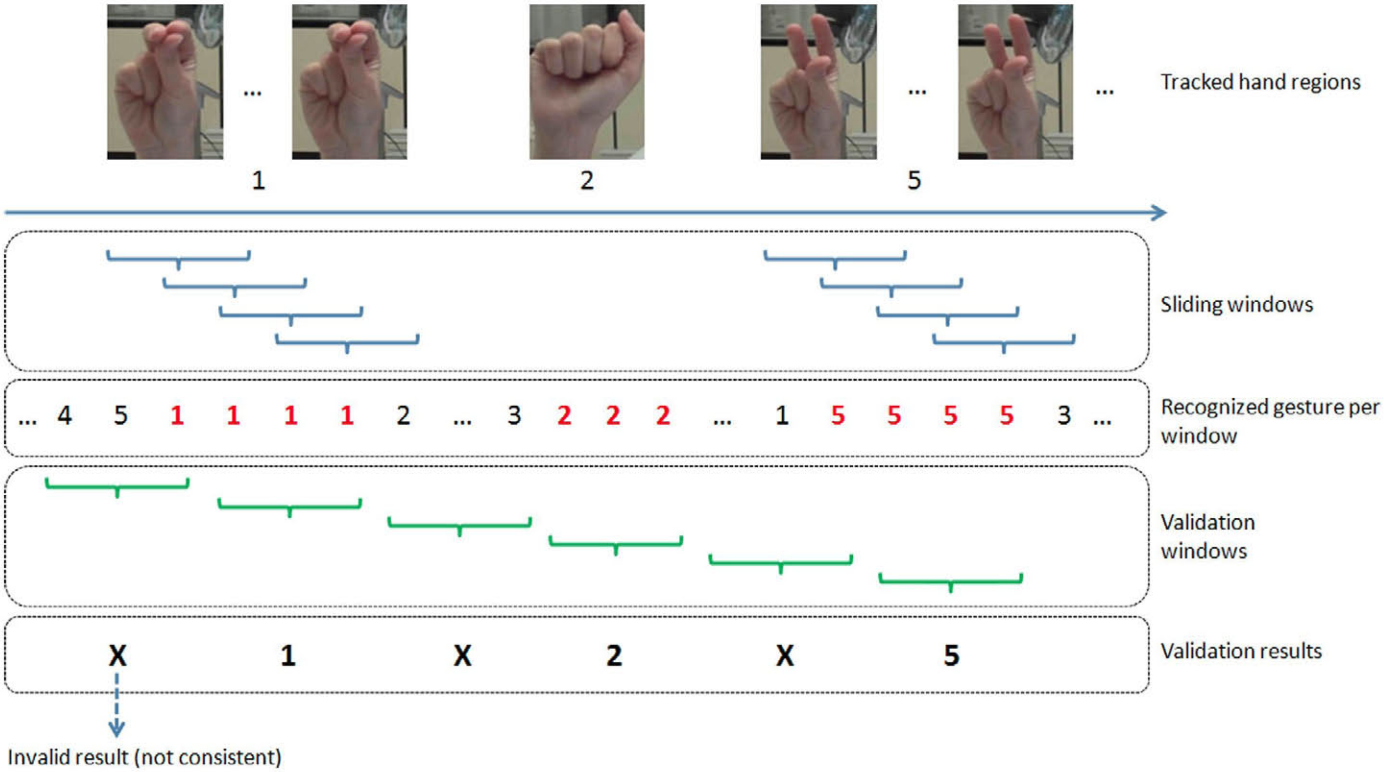

Fig. 12. Sliding window approach to validate a prediction.

the detection and recognition phases. The former takes image regions containing hand poses from the considered dynamic hand gestures as positive samples, and other irrelevant hand poses and background as negative samples. The latter takes cropped video sequences containing the performance of one dynamic hand gesture only once. The spatial cutting to extract relevant hand poses and the temporal segmentation to extract isolated dynamic hand gestures are carried out manually.

Moreover, the database includes six long video sequences for test. Every test video sequence contains the activity of a different individual as he was using the application, performing different dynamic hand gestures. These test video sequences are used to validate the whole hand gesture recognition system.

All the video sequences are recorded in a realistic scene, which means a standard environment with a non-uniform background, and other moving objects. The typical scene structure is composed of an individual interacting with the computer approximately 0.7m away from it. The sensor is located in the top of the screen.

4.2. Video descriptor evaluation

The VS-LBP descriptor has been compared to two state-of-the-art video descriptors, Volume Local Binary Patterns (VLBP) and Local Binary Patterns from Three Orthogonal Planes (LBP-TOP) [19].

The idea behind VLBP is the same as the $\text{LBP}_{P,R}$ operator, except that it is extended to the previous and posterior neighboring frames, as well. Given a pixel belonging to a specific frame, its local volume neighborhood is formed by its P spatial sampling points in the same frame, the center pixel in some previous frame, and its P sampling points, and the center pixel in some posterior frame and its P sampling points. Thus, VLBP descriptor uses three parallel planes. This neighborhood is thresholded considering the center pixel from the middle frame, resulting a binary number that represents a volume local binary pattern.

In order to solve the dimensionality problems of the VLBP descriptor, the LBP-TOP descriptor proposes to concatenate LBP histograms from three orthogonal planes XY, XT, and YT, which intersect in the center pixel. The XY plane represents appearance information, while the XT plane gives a visual impression of one row changing in time, and the YT plane describes the motion of one column in temporal space.

The VLBP descriptor depends on three parameters, which are the temporal interval L, the number of neighbors P, and the radius of the neighborhood R ($\text{VLBP}_{L,P,R}$). The original implementation of the VLBP descriptor is restricted to a value of $P = 4$, because of the computational cost. In addition, other works [19] have proven that the best results are achieved for the $\text{VLBP}_{4,1,1}$, therefore R is set to 1. Regarding the temporal interval, a range of values that goes from 1 to 4 is tested to limit the computational complexity.

The $\text{LBP-TOP}_{P_{XY},P_{XT},P_{YT},R_X,R_Y,R_T}$ descriptor presents similar restrictions regarding the number of neighbors in the three planes XY, XT and YT, although the number of neighbors in each plane can be increased up to $P_{XY} = P_{XT} = P_{YT} = 8$, since it is less computationally demanding. In addition, other works [19] have proven that the best results are achieved for the $\text{LBP-TOP}_{8,\,8,\,8,\,1,\,1,\,1}$. Therefore, eight neighbors are set for every plane, and $R_X = R_Y = 1$, which are the best values for extracting the local spatial structures. Regarding the radius in the temporal axis, a range of values that goes from 1 to 4 is tested.

Table 1
Average accuracy for different parameter configurations of the VLBP descriptor.

Descriptor Parameters			SVM parameter	Metrics
P	R	L	C	*accuracy*
4	**1**	**1**	**0.31**	**0.773**
4	1	2	30.61	0.336
4	1	3	0.01	0.555
4	1	4	0.01	0.382

Table 2
Average accuracy for different parameter configurations of the LBP-TOP descriptor.

Descriptor parameters						SVM parameter	Metrics
P_{XY}	P_{XT}	P_{YT}	R_X	R_Y	R_T	C	*accuracy*
8	**8**	**8**	**1**	**1**	**1**	**75.01**	**0.555**
8	**8**	**8**	**1**	**1**	**2**	**41.11**	**0.555**
8	8	8	1	1	3	80.41	0.382
8	8	8	1	1	4	0.01	0.382

As for the VS-LBP descriptor, the parameters are those corresponding to the spatial feature extraction S-LBP (intraframe encoding), and the number of samples *num_samples* for the temporal sampling procedure (interframe encoding). On the one hand, the S-LBP descriptor depends on four parameters: the number of neighbors P, the radius of the neighborhood R, and the number of samples per rows and columns, M and N respectively ($\text{S-LBP}_{P,R,M,N}$). Since previous works have proven that the best results for the $LBP_{P,\,R}$ descriptor are achieved by considering $R = 1$ and $P = 8$ [15], and the multi-scale scheme in our descriptor designs is carried out by a multiresolution Gaussian pyramid, other values have not been tested for these two parameters. For the M and N parameters a range of values that goes from 4 to 10 samples is tested. On the other hand, regarding the *num_samples* parameter, a maximum of *num_samples* = 5 is fixed due to its demanding memory requirements.

The regularization parameter for the SVM classifier, which is called C, is also tested. Large values of C can cause overfitting, while smaller values of C can cause a SVM model that is not able to separate the classes. A very wide range of values which goes from 0.01 to 100 taking steps of 0.3 has been tested.

The employed metric to make this comparison is the *Average accuracy* metric, defined as follows:

$$\textit{Average accuracy} = \frac{\text{Total number of correct gestures}}{\text{Total number of gestures}}. \tag{4}$$

Tables 1 , 2, and 3 show the best accuracy results for every parameter configuration of the three video descriptors. The LBP-TOP descriptor was designed to solve the dimensionality problem of the VLBP descriptor. However, this design is based on decreasing the length of the feature vector by reducing the information. For this reason, the

Table 3
Average accuracy for different parameter configurations of the VS-LBP descriptor.

Descriptor parameters		SVM parameter	Metrics
Intra-frame parameters	Inter-frame parameters: *num_samples*	C (best estimated parameter)	*accuracy*
$S - LBP_{8,1,4,10}$	4	2.11	0.984
$S - LBP_{8,1,4,10}$	**5**	**0.61**	**1**
$S - LBP_{8,1,8,8}$	4	77.71	0.985
$S - LBP_{8,1,8,8}$	5	89.41	0.995

Table 4
Latency for different video descriptors when computing video sequences with different characteristics.

		Video sequence 1	Video sequence 2
Characteristics	**Image size**	115 × 160	229 × 183
	Length	11 frames	18 frames
Latency	**VLBP**	2.799 s	11.654 s
	LBP-TOP	3.354 s	14.131 s
	VS-LBP	0.082 s	0.098 s

results for the LBP-TOP descriptor are worse than for the VLBP descriptor. The dimensionality problem is solved, but the extracted features are much less discriminative, causing bad results. In comparison with the VLBP and LBP-TOP descriptors, the VS-LBP descriptor is much more discriminative. In particular, it reaches a recognition accuracy of approximately 20% higher than the VLBP descriptor, and 40% higher than the LBP-TOP descriptor.

The reason why the VS-LBP descriptor is superior to the VLBP and LBP-TOP is the specific combination of spatial information and temporal information. In particular, the discriminative power of VS-LBP is obtained by means of the addition of global spatial information. As this information is more discriminative than the one used by VLBP and LBP-TOP, they need to use more temporal information to be as discriminative as possible, as it can be seen in Tables 1, 2, and 3. Whereas VS-LBP achieves a perfect accuracy considering only five frames (inter-frame parameter), VLBP and LBP-TOP need to analyze every frame ($L = 1$ and $R_T = 1$, respectively) to achieve high accuracy.

Regarding the computational efficiency, Table 4 shows the latency of the three descriptors when computing two video sequences with different characteristics. This latency measures how long every descriptor takes to compute the final feature vector once it has all the data to process it. The implementation has been run on Matlab on a computer configured with Intel i7-4510U processor and 3.1 GHz. It can be seen that VS-LBP is more computationally efficient than VLBP and LBP-TOP. Whereas VLBP and LBP-TOP need previous and posterior frames to compute both spatial and temporal information and wait for the final frame to generate the histogram, VS-LBP can compute spatial information frame by frame independently, and choose those according to the temporal sampling scheme to generate the final feature vector.

4.3. *Global system evaluation*

The results for the global system, integrating the detection, tracking and recognition phases are presented in this section. The global hand-gesture recognition system is first evaluated on the proposed Hand-gesture Database, but also tested on other databases related to hand gestures: NTU Dataset [4], American-Sign-Language (ASL) Finger-spelling Dataset [7], and other ASL Dataset captured by Intel Creative Camera [6].

In addition, a comparison with other state-of-the-art recognition frameworks ([4–7], and [8]) has been performed using the previous databases. These works were described in the Introduction section. Unlike the proposed recognition framework, which only uses color imagery, these works use depth imagery or both color and depth imagery. In particular the proposed video descriptor only uses intensity from color information.

4.3.1. *Evaluation on the proposed hand-gesture database*

The proposed hand-gesture database (Set 2) has been presented in Section 4.1.

Regarding the size of the temporal window to carry out the recognition task, it is fixed. The best value for this parameter has to take into account that every individual can execute the dynamic hand gestures with different speed, and every type of gesture has a specific length. Since there are only two real dynamic hand gestures: the “left click” and “right click”, the size of the sliding window can be estimated as the average value of the length of these two gestures. To that end, different sizes for the sliding window have been tested on six video sequences, where different individuals executes several dynamic hand gestures. To avoid outliers, that is, those cases in which individuals execute a hand gesture with a very different speed from the other ones, the median value for the best size of the temporal window has been chosen. In this case, this value is 24 frames.

In terms of computational cost, the latency of recognizing one gesture has been measured as:

$$t_{recog} = t_{VS-LBP} + t_{SVM-test} = 35.485\text{ ms} + 2.474\text{ ms} = 37.959\text{ ms}, \qquad (5)$$

where t_{VS-LBP} is the latency of the video descriptor computation, defined as the elapsed time since all the data is available (the temporal instant corresponding to the last frame of a set of frames that contains a potential gesture) until the descriptor is computed; and $t_{SVM-test}$ is the time taken to classify one temporal window (which includes the evaluation of the five classifiers).

In terms of average accuracy, the evaluation of the whole system is carried out by testing the six test sequences in the database. Since every test video sequence contains the continuous execution of several dynamic hand gestures, there are temporal windows that contain the end of a gesture and the beginning of the following one (transitions). As a rejection class has not been trained for the recognition phase, these transition windows have to be incorrectly labeled as one of the five considered classes. However, thanks to the validation prediction strategy this problem is satisfactorily solved. In addition, the fact that the size of the temporal window is fixed can degrade the recognition accuracy for those individuals whose execution speed is not appropriate for the size of the estimated temporal window. As expected, Table 5 shows that the recognition scores are in general worse than those obtained with segmented video sequences (see Tables 1, 2, 3). Despite of this fact, the recognition accuracy exceeds a rate of 0.9 for all the sequences when using the VS-LBP descriptor, outperforming the other approaches.

The high recognition accuracy obtained by using the proposed recognition framework is due to several mechanisms to be robust against typical detection/recognition problems. First, the hand detector is properly trained considering a large number of positive and negative samples. Second, the tracking algorithm is able to deal with noisy, false and missing detections still remaining from the detection stage. It estimates the best association between current detections and previous detections to generate robust hand trajectories. Finally, the random aspect of the video descriptor is used for training a large

Table 5
Comparison of the proposed framework using different methods for feature extraction on the hand-gesture database.

Method	Seq_1	Seq_2	Seq_3	Seq_4	Seq_5	Seq_6	Mean accuracy
$\text{VLBP}_{4,1,1}$ [19]	0.720	0.765	0.711	0.695	0.689	0.770	0.725
$\text{LBP-TOP}_{8,8,8,1,1,1}$ [19]	0.507	0.521	0.535	0.487	0.485	0.510	0.507
$\text{VS-LBP}_{8,1,4,10,5}$	**0.949**	**0.961**	**0.935**	**0.923**	**0.897**	**0.959**	**0.937**

Table 6
Comparison of the proposed framework with other state-of-the-art approaches on the NTU database.

Methods based on template matching	Accuracy
Thresholding decomposition+FEMD [4]	0.906
Near-convex decomposition+FEMD [4]	0.939
Shape context with bending cost [5]	0.791
Shape context without bending cost [5]	0.832
Skeleton matching [5]	0.786
Thresholding decomposition+FEMD [5]	0.932
Near-convex decomposition+FEMD [5]	0.939
Methods based on classifiers	**Accuracy**
Hand dominant line + SVM (h-h) [8]	0.971
Hand dominant line + SVM (l-o-o) [8]	0.911
VS-LBP + SVM (h-h)	**0.973**
VS-LBP + SVM (l-o-o)	**0.959**

number of sequence samples in the recognition stage, which simulates different execution speeds, behaviors, and noisy segmentations; and for testing several variations of the same hand gesture, allowing to select the one with the highest recognition score.

4.3.2. Evaluation on NTU dataset

NTU Dataset [4] provides both intensity and depth images. It is collected from 10 subjects, and it contains 10 poses. Each subject performs each pose 10 times, so that it has in total 1000 cases. For each pose, the subject changes the orientation, scale and articulation. Since this database contains single-image static gesture samples, the size of the temporal window is set to one frame. In this extreme case, the descriptor only considers spatial information, which corresponds to the called S-LBP descriptor.

Table 6 shows the comparisons between the proposed framework and other state-of-the-art approaches on the NTU Dataset, where average accuracy has been compute for different experiments: using half of the data for training and the other half for testing (h-h), and following a leave-one-out (l-o-o) strategy.

It can be observed that the proposed recognition system (*VS-LBP + SVM*) outperforms the rest ones obtaining the best accuracy in both training settings, h-h and l-o-o. Notice that the proposed recognition framework use color imagery (in this case, intensity information) unlike the others that use depth imagery. Depth imagery can offer more advantages from a recognition point of view, since it provides more structural information. Because of this fact, most of the current works in hand-gesture recognition are based on depth imagery. Nonetheless, most of the current electronic devices lack a depth sensor, but they are equipped with a color sensor. Consequently, the proposed color-based recognition system offers a substantial advantage in the sense that it can be apply to a wide range of existing electronic devices. The second best one is the *Hand dominant line + SVM* framework. In addition, as can be observed, methods based on classifiers obtain a better accuracy than those ones based on template matching techniques.

Table 7
Comparison of the proposed framework with other approaches in the state of the art on the ASL finger-spelling dataset.

Method	h-h	l-o-o
Hand dominant line + SVM [8]	0.962	0.583
3D model and hierarchical skeleton + SCF [24]	**0.978**	**0.843**
ESF descriptor + RF [6]	0.850	0.509
ESF descriptor + MLRF [6]	0.870	0.570
GR + RF (depth) [7]	0.690	0.490
GR + RF (color) [7]	0.730	–
GR + RF (depth+color) [7]	0.750	–
VS-LBP + SVM	**0.975**	**0.837**

Table 8
Comparison of the proposed framework with other approaches in the state of the art on the ASL database captured by intel creative gesture camera.

Method	h-h	l-o-o	For one subject
ESF descriptor + MLRF [6]	–	–	**0.847**
VS-LBP + SVM	**0.993**	**0.820**	0.843

4.3.3. Evaluation on ASL finger-spelling dataset

The ASL Finger-spelling Dataset [7] consists of 60k depth and color spatially segmented video sequences, corresponding to 24 of the 26 ASL letters performed by 5 subjects. Since this database already contains the spatio-temporal volume of hands segmented, only the recognition stage is applied in this case.

Table 7 compares different hand-gesture recognition frameworks with the proposed one in this paper, where average accuracy has been compute for the h-h and l-o-o experiments. Analyzing the results of the Table 7, the two best methods with a very similar score are the *3D model and hierarchical skeleton + SCF* and the one proposed in this paper, closely followed by *Hand dominant line + SVM*. Again, notice that *3D model and hierarchical skeleton + SCF* and *Hand dominant line + SVM* use depth imagery unlike the proposed one that only use color imagery. The responsible for this achievement is the novel and careful design of the novel video feature descriptor VS-LBP.

4.3.4. Evaluation on ASL dataset

The other ASL Dataset [6] contains video sequences where 3 subjects perform 24 of the 26 signs from the ASL sign language. For each letter, around 250 frames have been collected. Intensity and depth images have both been captured using Intel Creative Camera. To test this database, the whole recognition system has been applied to first extract the spatio-temporal volume of hands, and then to recognize the performed hand gestures.

Table 8 shows the comparison between the proposed framework and the *ESF descriptor + MLRF* method [6], where average accuracy has been compute for the h-h and l-o-o experiments. In addition, the best result obtained from the different settings of the l-o-o experiment (for one subject) is given.

According to the results in Table 8, both methods (*ESF descriptor + MLRF* and *VS-LBP + SVM*) obtain a similar accuracy score for the setting of one subject. Notice that the proposed recognition framework use color imagery instead depth imagery as the other one. Also, satisfactory results have been obtained for two additional experiments (h-h and l-o-o) using the proposed framework (these settings are not provided by the other method).

In conclusion, the proposed system is not only one of the best, but also this is achieved by using color imagery, instead of depth imagery, which can be an advantage in the current panorama where most of the multimedia devices only have a color camera.

5. Conclusions

A more natural, intuitive, user-friendly, and less intrusive Human–Computer interface for controlling an application by executing hand gestures has been developed. In particular, a mouse-like pointing device has been evaluated as an example of HCI application, where different mouse functions are triggered depending on the recognized hand gesture.

For this purpose, a robust hand-gesture recognition system has been designed and implemented. The system has been divided into three stages: detection, tracking, and recognition. The detection stage processes a video sequence frame by frame, and uses a binary SVM classifier together with an image descriptor in order to detect potential hand poses. These detections are employed as input of a multiple object tracker to generate a spatio-temporal trajectory of hand regions. Finally, the recognition stage segments the video sequence using the trajectory, then computes a video descriptor from the segmented video sequence, and lastly delivers the video descriptor to a set of classifiers to carry out the recognition task. The key contribution of the system has been the design of a novel and highly discriminative video descriptor for the recognition stage, which is called Volumetric Spatiograms of Local Binary Patterns. It has proven to be more discriminative and computationally efficient than other methods in the state of the art.

The final obtained recognition accuracy of the global system is quite high, allowing to use the developed hand-gesture recognition system for real HCI applications.

References

[1] S.-H. Lee, M.-K. Sohn, D.-J. Kim, B. Kim, H. Kim, Smart tv interaction system using face and hand gesture recognition, in: IEEE International Conference on Consumer Electronics, 2013, pp. 173–174.

[2] M. Pourazad, A. Bashashati, P. Nasiopoulos, A random forests-based approach for estimating depth of human body gestures using a single video camera, in: IEEE International Conference on Consumer Electronics, 2011, pp. 649–650.

[3] L. Gallo, A. Placitelli, M. Ciampi, Controller-free exploration of medical image data: experiencing the Kinect, in: International Symposium on Computer-Based Medical Systems, 2011, pp. 1–6.

[4] Z. Ren, J. Yuan, Z. Zhang, Robust hand gesture recognition based on finger-earth mover's distance with a commodity depth camera, in: Proceedings of the International Conference on Multimedia, ACM, 2011, pp. 1093–1096.

[5] Z. Ren, J. Yuan, J. Meng, Z. Zhang, Robust part-based hand gesture recognition using kinect sensor, IEEE Trans. Multimed. 15 (5) (2013) 1110–1120.

[6] A. Kuznetsova, L. Leal-Taixe, B. Rosenhahn, Real-time sign language recognition using a consumer depth camera, in: IEEE International Conference on Computer Vision Workshops, 2013, pp. 83–90.

[7] N. Pugeault, R. Bowden, Spelling it out: real–time ASL fingerspelling recognition, in: IEEE International Conference on Computer Vision Workshops, 2011, pp. 1114–1119.

[8] Y. Wang, R. Yang, Real-time hand posture recognition based on hand dominant line using kinect, in: IEEE International Conference on Multimedia and Expo Workshops, 2013, pp. 1–4.

[9] I. Laptev, T. Lindeberg, Space-time interest points, in: IEEE International Conference on Computer Vision, 1, 2003, pp. 432–439.

[10] H. Wang, M.M. Ullah, A. Klaser, I. Laptev, C. Schmid, Evaluation of local spatio-temporal features for action recognition, in: British Machine Vision Conference, BMVA Press, 2009, pp. 124.1–124.11.

[11] S. Umakanthan, S. Denman, S. Sridharan, C. Fookes, T. Wark, Spatio temporal feature evaluation for action recognition, in: International Conference on Digital Image Computing Techniques and Applications, 2012, pp. 1–8.

[12] M.-H. Yang, N. Ahuja, M. Tabb, Extraction of 2D motion trajectories and its application to hand gesture recognition, IEEE Trans. Pattern Anal. Mach. Intell. 24 (8) (2002) 1061–1074.

[13] W.T. Freeman, M. Roth, Orientation histograms for hand gesture recognition, in: International Workshop on Automatic Face and Gesture Recognition, 12, 1995, pp. 296–301.

[14] R. Chaudhry, A. Ravichandran, G. Hager, R. Vidal, Histograms of oriented optical flow and Binet-Cauchy kernels on nonlinear dynamical systems for the recognition of human actions, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 1932–1939.

[15] T. Ojala, M. Pietikainen, T. Maenpaa, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Trans. Pattern Anal. Machine Intell. 24 (7) (2002) 971–987.

[16] J. Huang, J. Zhao, W. Gao, C. Long, L. Xiong, Z. Yuan, S. Han, Local binary pattern based texture analysis for visual fire recognition, in: International Congress on Image and Signal Processing, 4, 2010, pp. 1887–1891.

[17] K. Meena, A. Suruliandi, Local binary patterns and its variants for face recognition, in: International Conference on Recent Trends in Information Technology, 2011, pp. 782–786.

[18] W. Liu, S-J. Li, Y-J. Wang, Automatic facial expression recognition based on local binary patterns of local areas, in: International Conference on Information Engineering, 1, 2009, pp. 197–200.

[19] G. Zhao, M. Pietikainen, Dynamic texture recognition using local binary patterns with an application to facial expressions, IEEE Trans. Pattern Anal. Machine Intell. 29 (6) (2007) 915–928.

[20] T. Ojala, M. Pietikäinen, D. Harwood, A comparative study of texture measures with classification based on featured distributions, Pattern Recognit. 29 (1) (1996) 51–59.

[21] E. Choi, C. Lee, Feature extraction based on the Bhattacharyya distance, in: International Geoscience and Remote Sensing Symposium, 5, 2000, pp. 2146–2148.

[22] A. Mahmood, S. Khan, Early terminating algorithms for adaboost based detectors, in: IEEE International Conference on Image Processing, 2009, pp. 1209–1212.

[23] C. del Blanco, F. Jaureguizar, N. Garcia, An efficient multiple object detection and tracking framework for automatic counting and video surveillance applications, IEEE Trans. Consum. Electron. 58 (3) (2012) 857–862.

[24] C. Keskin, F. Kirac, Y. Kara, L. Akarun, Randomized decision forests for static and dynamic hand shape classification, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2012, pp. 31–36.