


# Principal motion components for one-shot gesture recognition

Hugo Jair Escalante<sup>1,2</sup>  · Isabelle Guyon<sup>2</sup> · Vassilis Athitsos<sup>3</sup> · Pat Jangyodsuk<sup>3</sup> · Jun Wan<sup>4</sup>

Received: 18 November 2014 / Accepted: 17 April 2015 / Published online: 7 May 2015  
© Springer-Verlag London 2015

**Abstract** This paper introduces *principal motion components* (PMC), a new method for one-shot gesture recognition. In the considered scenario a single training video is available for each gesture to be recognized, which limits the application of traditional techniques (e.g., HMMs). In PMC, a 2D map of motion energy is obtained per each pair of consecutive frames in a video. Motion maps associated to a video are processed to obtain a PCA model, which is used for recognition under a reconstruction-error approach. The main benefits of the proposed approach are its simplicity, easiness of implementation, competitive performance and efficiency. We report experimental results in one-shot gesture recognition using the ChaLearn Gesture Dataset; a benchmark comprising more than 50,000 gestures, recorded as both RGB and depth video with a Kinect<sup>TM</sup> camera. Results obtained with PMC are competitive with alternative methods proposed for the same data set.

**Keywords** Motion energy maps · PCA · One-shot learning · Gesture recognition · ChaLearn gesture challenge

---

✉ Hugo Jair Escalante  
hugojair@inaoep.mx

<sup>1</sup> Instituto Nacional de Astrfísica, Óptica y Electrónica, Luis Enrique Erro No. 1, Tonantzintla, Puebla 72840, Mexico

<sup>2</sup> ChaLearn, 955 Creston Road, Berkeley, CA 94708, USA

<sup>3</sup> Computer Science and Engineering Department, University of Texas at Arlington, 701 S. Nedderman Drive, Arlington, TX 76019, USA

<sup>4</sup> National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China

## 1 Introduction

Gestures are a form of non-verbal communication, which is highly intuitive and very effective. Because of its relevance, automated gesture recognition is a research topic with a growing popularity in computer science, see, e.g., [2, 33]. Traditional approaches for the automated recognition of gestures learn a model (e.g., a hidden Markov model, HMM [37]) from a set of sample videos including the gestures of interests; where, commonly, the variation of spatial positions from body parts across time are used as inputs for the models. In general, the more examples we have for building a model, the better its performance is in new data [22]. However, in many domains gathering examples of gestures is a time-consuming and expensive process. Hence, methods that can learn from few examples are needed. On the other hand, it is also desirable that gesture recognition methods do not rely on specialized sensors to estimate body-part positions: it may not be straightforward to get access to such devices; or on the output of techniques for associated problems like hand tracking or pose estimation [11]: these techniques may introduce noise into the data acquisition process. Undoubtedly, methods that can be trained from very few examples and using unspecialized equipment would make the applicability of gesture recognition more widespread: e.g., anyone with access to a webcam would be able to build gesture recognizers.

In this paper, we approach the problem of gesture recognition by using a single example of each gesture to be recognized. This task, called one-shot gesture recognition, was proposed in the context of the ChaLearn gesture challenge [17, 18]. The targets for this type of methods are user adaptive applications that require the recognition of gestures from arbitrary and user-defined vocabularies;

domains where gestures can change with time and models need to be modified periodically; and scenarios where gathering data is too expensive or users are not willing to spend time collecting large amounts of data.

For each gesture to be recognized, the only information we have for building a model is a single video recorded with a Kinect<sup>TM</sup> camera, where both RGB and depth videos are available. Despite the fact that Kinect<sup>TM</sup> can record additional data (e.g., skeleton information), it was disregarded in the ChaLearn gesture challenge. This favored the development of new methods not relying on a first step of skeleton extraction, which is often not robust to occlusions, and requires spatial and temporal resolutions not available in many application settings. The problem is restricted to single-user gesture recognition, there is little variation in the background and the user is placed right in front the sensor. Notwithstanding, the problem is very challenging as single example is available for each gesture, thus, traditional methods (e.g., HMMs) cannot be applied directly. Also, there is a wide diversity of domains of gestures, ranging from highly dynamic (e.g., “aircraft-landing” signals) to static (e.g., “Chinese letters”) and some body parts may be occluded [16]. Additionally, the sampling rate is low (of the order of 12fps). Clearly, standard gesture recognition methods are not directly applicable, and even though the problem has been simplified, it remains a difficult task.

We propose a simple and efficient method, yet very effective, for one-shot gesture recognition called principal motion components (PMC). The main goal of PMC was to act as a strong baseline for the ChaLearn gesture challenge [16–18] and it has inspired several of the top ranking entries, see, e.g., [44]. The proposed method is based on a motion map representation that is obtained by processing the sequence of frames in a video. Motion maps are used in combination with principal component analysis (PCA) under a reconstruction-error classification approach. The proposed method was evaluated in a large database with 54,000 gestures used in ChaLearn gesture challenge [16–18]. We compare the performance of PMC to a wide variety of techniques. Experimental results show that the proposed method is competitive with alternative methods. In particular, we found that the proposed method resulted very effective for recognizing highly dynamic gestures, although it is less effective when static gestures are analyzed. The proposed method can be improved in several ways and it can be used in combination with other approaches, see, e.g., [8, 44].

The main contributions of this work are threefold:

- A new representation for motion in video. This representation can be seen as a *bag-of-frames* formulation, where each video is characterized by the

(orderless) set of motion maps it contains. The representation can be used with other methods for gesture recognition and for other tasks.

- A new one-shot gesture recognition approach based on PCA. Our method is capable of building a predictive PCA model from a single video without using any temporal information.
- The evaluation of the proposed method in a large-scale heterogeneous database and a comparison of it with a variety of alternative techniques.

The rest of this paper is organized as follows. The next section reviews work closely related to our proposed method. Section 3 introduces the principal motion components method. Section 4 describes the experimental settings adopted in this work and Sect. 5 reports experimental results. Finally, Sect. 6 presents the conclusions derived from this paper and outlines future work directions.

## 2 Related work

This section reviews related work on two key components of the proposed approach: motion-based representations and PCA-based recognition.

### 2.1 Motion-based representations

When it is not possible to track body parts across a sequence of images, motion-based representations have been used for gesture recognition. Different approaches have been proposed, mainly based on template matching [1, 2]. The seminal work of Bobick and Davis used motion history images (MHIs) to represent videos [6], where MHIs are obtained by accumulatively adding (thresholded) binary-difference images, this type of templates reveals information about the history of motion in a video (i.e., how movement happened). Statistical moments obtained from the MHIs were used for recognition. Davis [10] extended the MHI representation to generate histograms of motion orientation. The MHI is obtained for each video and the resulting template is divided into spatial regions. Gradients from motion values are obtained on each region separately, a histogram is generated per each region using as bins a set of predefined orientations over the gradients. Per-region histograms are concatenated to obtain a 1D representation for each video. A similarity-based approach was used for recognition in that work.

In [36], sequence of images are represented by a spatiotemporal template. As preprocessing, the object of interest is isolated from the rest of the scene. Then, the sequence of cropped frames is processed to obtain optical flow fields. Flow frames are divided into a spatial grid and

motion magnitudes are added in each cell. In [3], the authors obtained motion vectors (correspondences between blocks of pixels in adjacent frames) for successive frames and generated a 2D motion histogram, in which the occurrence of motion vectors is quantized. In [45] authors proposed a representation, called Pixel Change Ratio Map (PCRM), based on motion histograms that account for the occurrence of specific values of motion in the video sequence. That is, the bins correspond to different (normalized) motion values. This approach is very similar to that in [10]. However, under PCRM, the average of motion energy in cells of the grid is used instead of the orientation of gradients. The representation proved to be very effective for video retrieval, clustering and classification. In [38], authors proposed a method for key frame extraction from video shots. The core of the method is a representation based on motion histograms. Optical flow fields are obtained for each frame, a subset of different combinations of magnitude and direction of motion values is used as the bins of the motion histogram. Motion histograms, one per frame, are then processed to extract representative frames of the sequence.

Other approaches define motion histograms in terms of symbols derived from optical flow analysis [35]; build classification models using motion histograms over voxels as features [27]; and generate histograms of gradient orientations for static gesture recognition [14].

In most of the above-described approaches, a single template based on motion histograms is obtained to represent a whole sequence of frames. In our proposed representation, a motion map, accounting for the spatial distribution of motion across successive frames, is obtained per each difference image. This can be thought of as a relatively low-resolution 2D map, each location accounting for the amount of motion at a given position, at a given time. However, we discard the time ordering of the various maps and time is only taken into account by the fact that the maps are based on consecutive frame differences. Thus, by analogy to bag-of-words representations in text recognition that ignore word ordering in text, we can talk of a “bag-of-frame” type of representation, which is neither a template nor a time-ordered sequence of features. In this way, we have a set of observations (motion maps) associated with a single gesture, which can be used for the induction of classifiers. To the best of our knowledge, none of the above-described methods has been evaluated in one-shot gesture recognition [17, 18].

In the context of one-shot learning gesture recognition, template-based methods have been popular. A simple average template approach was the first baseline proposed by the organizers of the gesture recognition challenge, and it remained a difficult baseline to beat during the first weeks of the competition [18]. In [28], the authors proposed a

template matching approach for one-shot learning gesture recognition, where three ways of generating templates were proposed (2D standard deviation, Fourier transform and MHIs). For recognition, the authors used the correlation coefficient to compare templates and testing videos. In [43], an extended MHI that incorporates gait energy information and inverse recording was proposed, although the method obtained very good performance, it is difficult to assess the contribution of the sole recognition approach as several preprocessing steps were performed beforehand (the authors mention that preprocessing improves the performance of their method by about 9 %). Other methods have been proposed in the context of the gesture recognition challenge, including probabilistic graphical models [31], methods based on novel descriptors [41], and techniques from manifold learning [26], these and other methods are summarized by Guyon et al. [17, 18]. In Sect. 5, we compare the performance of our proposal to these methods.

## 2.2 PCA for gesture recognition

The second component of our proposal is a PCA-based method for gesture recognition. PCA has been widely used in many computer vision tasks, including gesture recognition [2, 34, 36], and very efficient implementations are available (see, e.g., [25]). In most of the times, PCA has been used to reduce the dimensionality of the representation or to eliminate noisy and redundant information, see, e.g., [4]; in fact, this is a common preprocessing step when facing any machine learning task [19].

Some authors have used PCA for recognition [15, 30, 32, 40]. The most used approach consists of estimating the reconstruction error obtained after projecting the data into a PCA model as a measure of the likelihood that an instance belongs to a class. This recognition method was first reported in the seminal work of Turk and Pentland for face recognition [40]. A similar approach was adopted in [32] to classify hand postures to be used for gesture recognition by a high-level approach. The authors used a large data set of images with diverse hand postures and used the PCA reconstruction approach to classify hand postures. This approach has proved to be very effective in other domains as well (e.g., spam filtering, [15], and pedestrian detection, [30]). The reconstruction approach based on PCA has been also used for one-class classification and outlier detection [20, 39].

The motivation behind using a reconstruction-error approach for one-shot recognition stems from the fact that we do not know what are the underlying motion dimensions associated to a particular gesture, and we would like PCA to automatically determine what are those dimensions and to use such information for recognition. One should note

that previous work has used the PCA-reconstruction approach considering a data set of labeled instances, where many instances are available per each class. In our proposal, we have multiple observations taken from a single instance associated to a class (the bag-of-frames for a gesture). Another way of thinking of our model is that it acts as a single-state hidden Markov model for each gesture, the PCA model representing an i.i.d. generating process. To the best of our knowledge, PCA has not been used similarly for recognition, not even for other tasks than gesture recognition.

### 3 Principal motion components

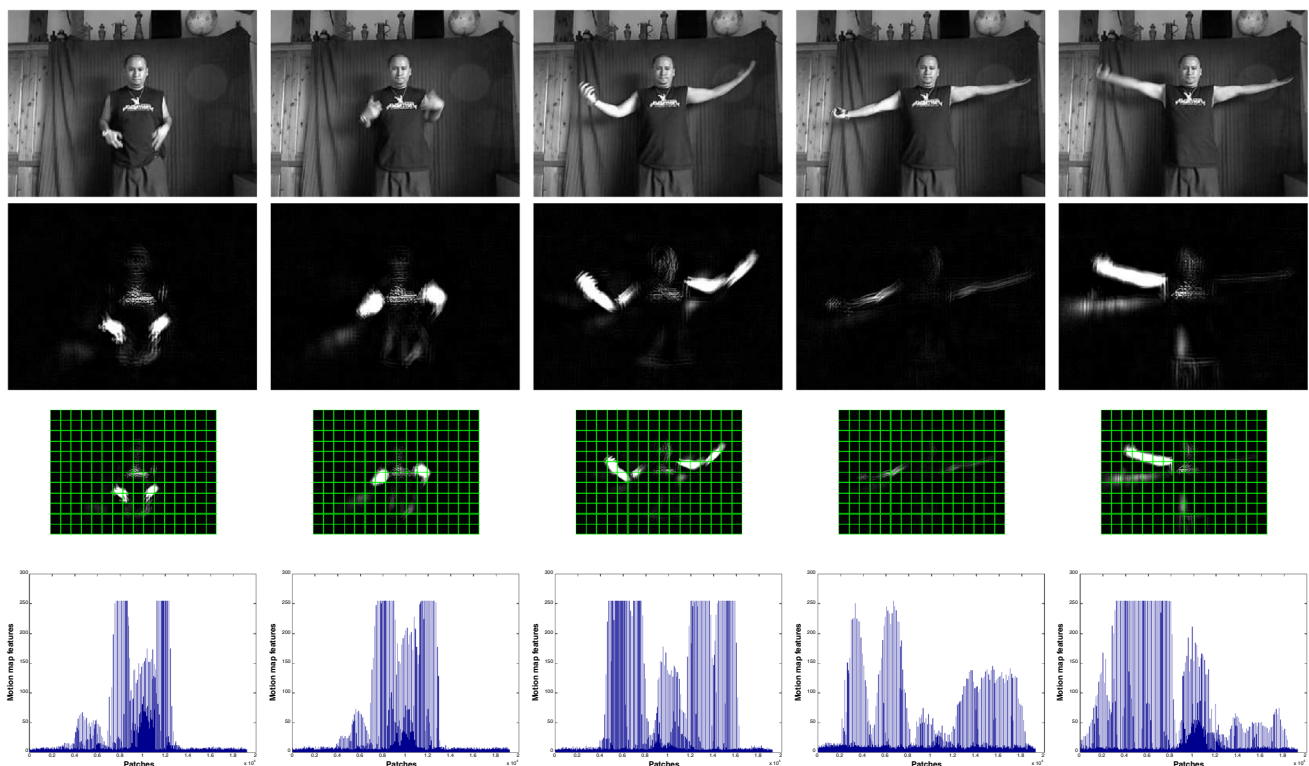
The proposed principal motion components (PMC) approach involves two main steps: (1) obtaining motion maps from videos and (2) obtaining PCA models to be used for recognition, these steps are described in detail in this section.

#### 3.1 Representation: motion maps (bag-of-frames)

Let  $\mathcal{V}$  be a video composed of  $N$  frames,  $\mathcal{V} = \{I_1, \dots, I_N\}$ , where  $I_i \in \mathbb{R}^{w \times h}$  is the  $i$ th frame,  $w$  and  $h$  being the width and height of the image, respectively. We represent a video

by a set of motion energy maps,  $H_1, \dots, H_{N-1}, H_j \in \mathbb{R}^{N_b}$ , one per each frame. Each map accounts for the movement taking place in consecutive frames on fixed spatial locations of the frames.

For obtaining motion maps, we first generate motion energy images by subtracting consecutive frames in the video:  $D_i = I_{i+1} - I_i$ ,  $i = \{2, \dots, N-1\}$  (we set  $D_1 = 0$  to have the same number of difference images as frames in the video). Next, a grid of equally spaced patches is defined over the difference images. The size of the patches is the same for all the images. We denote with  $N_b$  the number of patches in the grid. We estimate for each difference image  $D_i$ , the average motion energy in each of the patches of the grid; this is done by averaging motion values for pixels within each patch. That is, we obtain a 2D motion map for each difference image, where each element of the map accounts for the average motion energy in the image in the corresponding 2D location. The 2D maps are transformed into a 1D vector  $H_i \in \mathbb{R}^{N_b}$ . Hence, each video  $\mathcal{V}_i$  is associated to a matrix  $\mathbf{H}_i$  of dimensions  $N-1 \times N_b$ , with one row per frame and one column per patch. We call  $\mathbf{H}_i$  the bag-of-frames representation for the video, under the motion maps characterization. Figure 1 shows motion maps for a subset of frames in a video. In the figure motion maps are shown in temporal order, although, in the proposed approach, order of motion maps is not taken into account.



**Fig. 1** Extraction of motion maps for a video. From *top* to *bottom*: frames from the video; difference images; grid over images; 1D motion maps. Each patch from the grid corresponds to a value in the plot, this value is the average of motion in the corresponding patch



For the implementation, we adopted a more efficient approach to generate motion maps. Each motion energy image  $D_i, i = \{1, \dots, N - 1\}$  is downsized (e.g., via *cubic interpolation*) up to a specified scale  $\gamma$ . Motion maps are obtained by concatenating the rows from the downsized images.

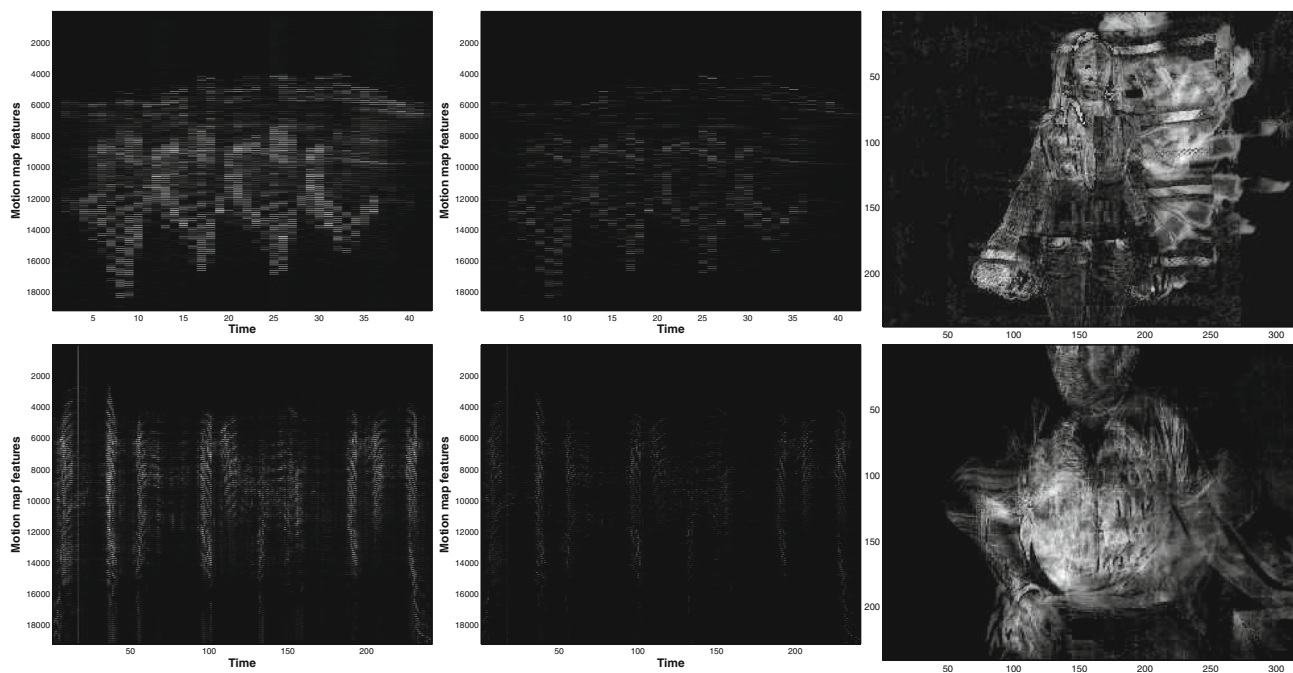
One should note that as the proposed representation captures motion in fixed spatial locations, translation variations may have a negative impact into the motion maps representation. The extreme case is when considering a large number of patches (e.g., when having one bin per pixel), resulting in a fine-grained map for which translation variance is a critical issue. To overcome this problem, we expand motion information in each difference image  $D_i$  as follows:  $D_i = D_i + D_i^l + D_i^r + D_i^u + D_i^d$ . Where  $D_i^l, D_i^r, D_i^u,$  and  $D_i^d$  are difference images  $D_i$  translated by a gap of  $\tau$ -pixels to the left, right, up, and down directions, respectively. Basically, we are growing the region of motion to make the representation less dependant on the position of the user with respect to the camera.

Figure 2 shows motion maps extracted from videos depicting different gestures and performed by different persons; row 1 shows a very dynamic gesture, whereas row 2 shows a static one. We can see that motion information is effectively captured by the proposed representation, as expected, the more dynamic the gesture (as depicted in the

accompanying MHIs), the higher the values of the motion map. It is interesting that even the representation for the static gesture shows high motion energy values, which can be due to unintentional movement from the user that is not related to the gesture. The PCA model is expected to capture the main dimensions of motion and to limit the contribution of such noisy movements. From Fig. 2, we can also see that the motion expansion emphasizes motion energy in neighboring patches (compare the leftmost and center images), which makes the representation slightly more robust against variance in translation.

### 3.2 Recognition: PCA-based reconstruction

For recognition, we consider a reconstruction-error approach based on PCA. Consider a training video representing a single gesture. We first compute a bag-of-frames representation  $H_1, \dots, H_{N-1}$ , (alternatively denoted by matrix  $\mathbf{H}_i$ ), as explained in the previous section. Here,  $n = 1, \dots, N - 1$  does NOT represent a time index and the frames representing motion (converted in feature vectors) can be arbitrarily re-ordered. The modeling approach then consists of treating the  $H_n$  feature vectors as training examples of a PCA model, globally representing the frames of that gesture. The principal components can be thought of as “principal motions”. Given now a new video also in a



**Fig. 2** Motion maps for selected gestures taken from two different vocabularies. We show motion maps with (left) and without motion expansion (center), together with the corresponding MHI (right). The

$x$ -axis of the images show the motion maps for the different frames. The domains of the depicted gestures are: “Canada-aviation ground circulation” (top), and “Gang hand-signals” (bottom)

bag-of-frames representation, its similarity to the training video can be assessed by the average reconstruction error of the frames of the video under the PCA model.

Let  $\mathbb{V} = \{\mathcal{V}_\infty, \dots, \mathcal{V}_K\}$  be the set of videos corresponding a gesture vocabulary (e.g., “diving signals”), where each video corresponds to a different gesture (e.g., “out of air” gesture). We apply PCA to each of the bag-of-frames representations  $\mathbf{H}_1, \dots, \mathbf{H}_K$  associated to the different training videos in  $\mathbb{V}$ . We center each matrix  $\mathbf{H}_i$ :  $\mathbf{H}_i = \mathbf{H}_i - \mathbf{H}_\mu^i$  where  $\mathbf{H}_\mu^i$  is a matrix with each row being the average of  $\mathbf{H}_i$ , and apply singular value decomposition:  $\mathbf{H}_i = \mathbf{USV}$ , we store the top  $c$  singular values  $\mathbf{S}_c$  from  $\mathbf{S}$  together with the corresponding eigenvectors  $\mathbf{V}_c$  (i.e., the principal components), where  $\mathbf{V}_c$  is the matrix formed by the first  $c$ -columns of  $\mathbf{V}$ . Hence for each gesture in the vocabulary, we obtain a PCA model represented by the pair  $(\mathbf{S}_c, \mathbf{V}_c)_{\{1, \dots, K\}}$ .

Figure 3 shows the principal motion components for a particular gesture vocabulary; the figure illustrates the benefits of the proposed approach. We can appreciate that the principal motion components indeed capture the intrinsic dimensions of motion of each gesture. By comparison, informative motion is not as clearly captured by competing motion-based representations, e.g., MHI (column 4) and the sequence of motion maps (column 5). For this particular vocabulary, the principal motion components can be easily associated by visual inspection with the image that visually describes the gesture (column 6).

A test video  $\mathcal{V}_T$ , depicting a single gesture<sup>1</sup> that needs to be classified, is processed similarly as training videos, thus it is represented by a matrix of motion maps  $\mathbf{H}_T$ . Matrix  $\mathbf{H}_T$  is projected into each of the  $K$ -spaces induced by the training PCA models  $(\mathbf{S}_c, \mathbf{V}_c)_{\{1, \dots, K\}}$ , where the projection of  $\mathbf{H}_T$  under the  $i$ th PCA model is obtained as follows [21]:

$$\hat{\mathbf{H}}_T^i = (\mathbf{H}_T - \mathbf{H}_\mu^i) \mathbf{V}_{c,i} \mathbf{S}_{c,i}^{-\frac{1}{2}} \quad (1)$$

where subscript  $i$  in  $\mathbf{S}_{c,i}$  and  $\mathbf{V}_{c,i}$  indicates the index of the associated PCA model. Next projections are reconstructed back, the reconstruction of  $\mathbf{H}_T$  under the  $i$ th-PCA model is given by:

$$\mathbf{R}_i = \hat{\mathbf{H}}_T^i (\mathbf{S}_i^{-\frac{1}{2}} \mathbf{V}_{c,i}^T) + \mathbf{H}_\mu^i \quad (2)$$

where superscript  $T$  indicates the transpose of a matrix.

We can measure the reconstruction error for each  $\mathbf{R}_i$  as follows:

$$\epsilon(i) = \frac{1}{Q} \sum_{q=1}^Q \sqrt{\sum_{m=1}^M (\mathbf{R}_{i,qm} - \mathbf{H}_{T,qm})^2} \quad (3)$$

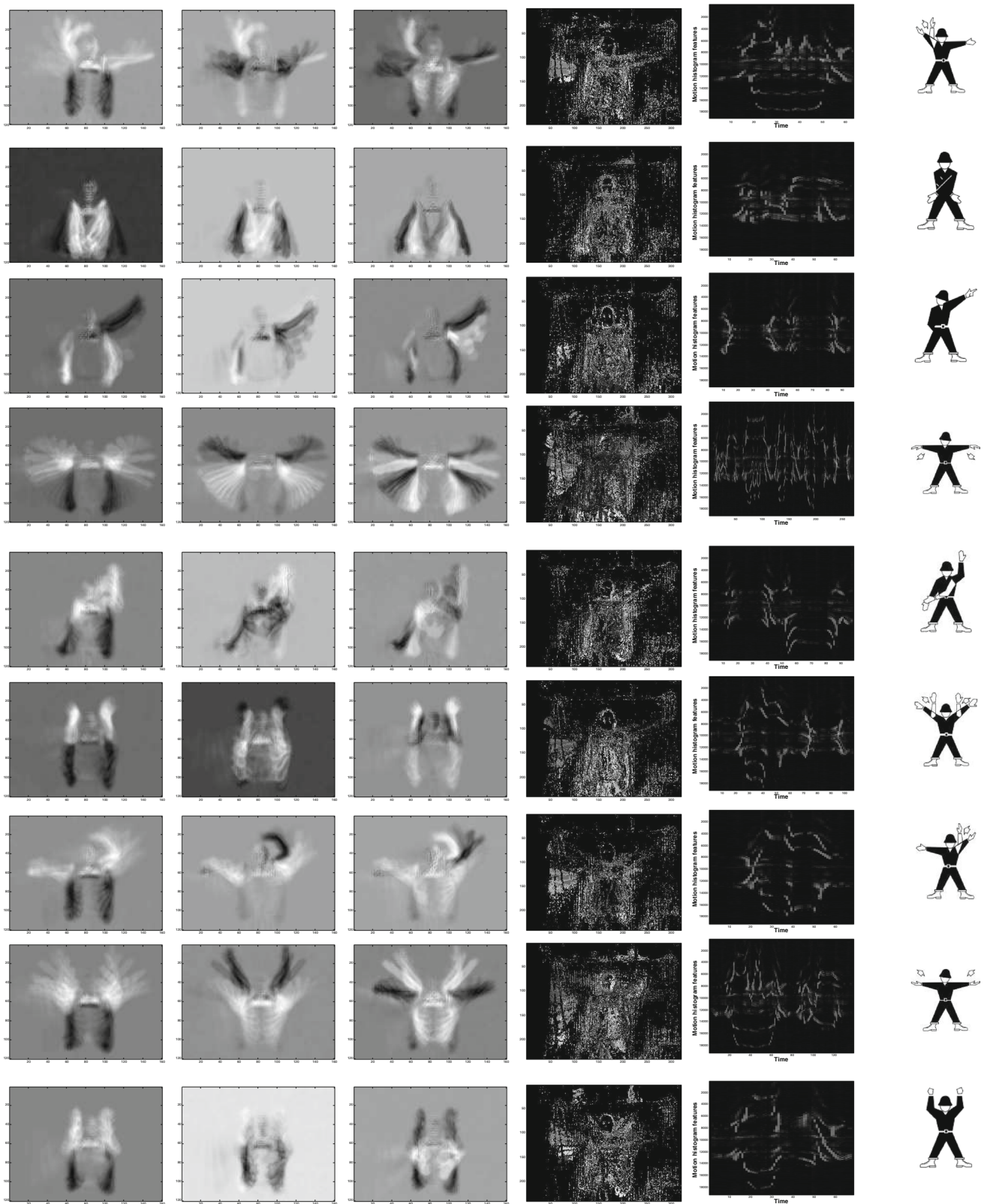
where  $q$  and  $m$  are the number of rows and columns of  $\mathbf{H}_T$ , respectively, and with  $i = 1, \dots, K$ . Finally, we assign  $\mathcal{V}_T$  the gesture corresponding to the PCA model that obtained the lowest reconstruction error, that is:  $\arg \min_i \epsilon(i)$ .

Similar reconstruction-error approaches have been adopted for one-class classification [39], where instances of the target class are used to generate the PCA model and a threshold on the reconstruction error is used for classification. Reconstruction error has been also used for spam filtering [15], face recognition [40] and pedestrian detection [30], see Sect. 2. One should note that in previous work a set of labeled instances has been used to generate the PCA model of each class, whereas under the proposed approach the elements of a single instance (the amount of motion in the frame differences under the bag-of-frames representation) are used. Besides the granularity, the main difference stems in that, in previous work, one can assume each instance is representative of the category, while in our setting the set of motion maps associated to a gesture are not necessarily representative of the gesture (e.g., similar motion maps may be shared by different gestures).

Figure 4 shows the difference image obtained by subtracting original from reconstructed motion maps for a particular vocabulary (“helicopter”). Specifically, image  $i, j$  in the array of images depicts the difference between: the average of motion maps for image  $i$ , minus the average of motion maps for image  $i$  reconstructed with PCA model  $j$  (e.g., images in the diagonal show the difference image obtained by subtracting original representations from the reconstruction with the *correct* model). Only differences exceeding the value of  $1 \times 10^{-10}$  are shown in the images. As expected, gestures reconstructed with the correct PCA model obtain lower differences than the threshold, while the reconstruction of gestures using other models results in large differences across the whole 2D space.

The main motivation for our recognition technique is the fact that principal components minimize the reconstruction error when projecting the data into the components’ space; it can be shown that this is equivalent to finding the directions that maximize the variance of the data, which is the most known derivation of PCA, see, e.g., [5, 21]. Since the PCA model for a gesture is the one that minimizes the average reconstruction error for motion maps belonging to the corresponding video, this model should be the one (among the PCA models for other gestures) that better reconstructs new motion maps belonging to the same gesture. Clearly, this is not a discriminant classifier, since the PCA model for a gesture is generated independently of the models for other gestures, hence no inter-gesture

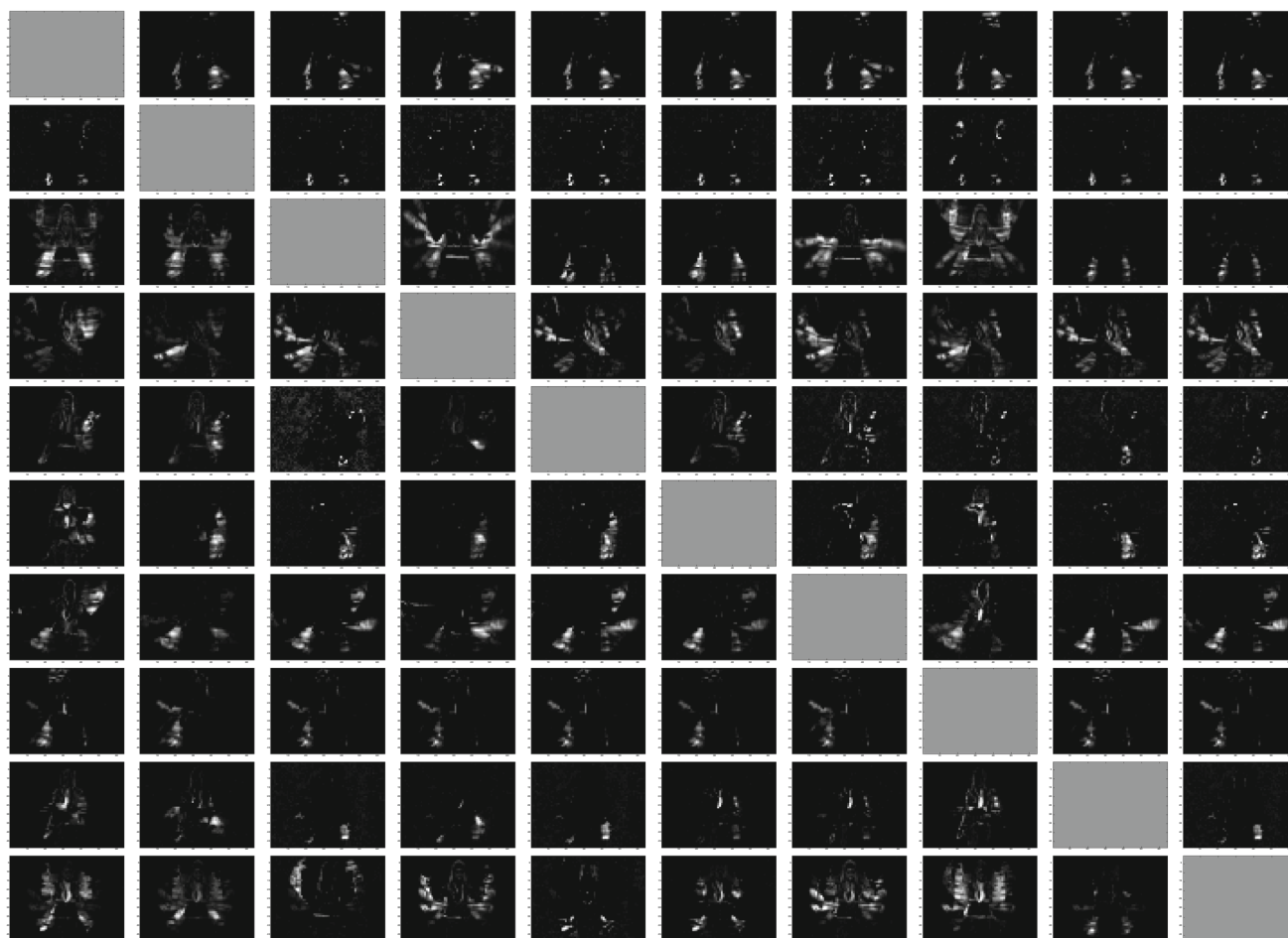
<sup>1</sup> We assume each video to be processed depicts a single gesture. Gesture segmentation is an open problem by itself that we do not approach in this paper, although we evaluate the performance of our method using gestures manually and automatically segmented with a basic technique.



**Fig. 3** Principal motion components for the gesture vocabulary: “Helicopter signals”. Each row is associated with a different gesture, the first three columns of each row display top 3 principal motion

components of the gesture; *columns 4–6* show the MHI, motion maps and a visual description of the corresponding gesture, respectively





**Fig. 4** Differences of the cumulative sum of reconstructed gestures and the original data

information is captured by the PCA approach. Nevertheless, our experimental study from Sect. 5 reveals that even with this limitation the proposed approach performs better than supervised methods that use the bag-of-frames representation.

#### 4 Experimental settings

We evaluate the performance of the principal motion components approach in the ChaLearn Gesture Dataset (CGD) [16]. CGD comprises 54,000 different gestures divided into 540 batches of 100 gestures each, gestures were recorded in RGB and depth video using a Kinect<sup>TM</sup> camera. The data set was divided into development (480 batches), validation (20 batches) and additional batches for evaluation (40 batches, referred to as final batches). Each batch is associated to a different gesture vocabulary, and it contains exactly one video from each gesture in the vocabulary for training and several videos containing sequences of gestures taken from the same vocabulary for testing. Each batch contains 100 gestures, the

number of training videos/gestures ranges from 8 to 12, depending on the vocabulary. There are 47 videos for testing in each batch containing sequences from 1 to 5 gestures each; hence, a gesture segmentation method has to be applied before recognition. The number of test gestures in each batch ranges from 88 to 92. About 20 different users contributed for the generation of gestures and there are about 30 different gesture vocabularies. See [16] for a comprehensive description of the CGD. It is important to mention that gesture vocabularies are quite diverse and come from many domains, e.g., see those mentioned in Table 1.

The CGD was developed in the context of ChaLearn gesture challenge<sup>2</sup>, an academic competition that focused in the development of gesture recognition systems under the one-shot-learning scenario [17, 18]. During the challenge, participants had access to the labels of all the development batches (1–480), although most participants used only twenty batches (1–20) when developing their systems. This can be due to the fact that for those batches

<sup>2</sup> <http://gesture.chalearn.org/>.



**Table 1** A few same vocabularies from the different batches

Referee wrestling signals	Motorcycle signals	Diving signals
Surgeon signals	Taxi South Africa	Gang hand signals
Tractor operation signals	Chinese numbers	Mudra signals

additional information was provided by the organizers (e.g., manual segmentation of test videos, hand tracking information, body-part estimates, etc.). Validation data were used by the organizers to provide immediate (online) feedback on the performance of participants’ methods. Final batches were used to evaluate the performance of the different methods. See [17] for more details on the ChaLearn gesture challenge.

The evaluation measure used in the challenge was the Levenshtein’s distance (normalized by the length of the truth labeling), which accounts for the number of edits that must be performed for taking a sequence of predictions into the ground truth labeling for a gesture. In the next section, we report experimental results on the CGD benchmark to evaluate the effectiveness of the principal motion components approach.

## 5 Experimental results

In this section, we report results from experiments that aim at evaluating different aspects of the proposed approach. First, we evaluate the performance of our method in the whole CGD collection. Next, we evaluate the method under different parameter settings. Then, we compare the proposed approach to a number of related techniques we implemented. Finally, we compare the performance of the principal motion components technique to other methods developed in the context of ChaLearn’s gesture challenge.

As explained previously, videos must be segmented to isolate gestures prior to recognition. We report results of experiments using both: manually segmented (batches 01–20 for development and validation only) and automatically segmented (all the batches) videos. For automatic segmentation, we used a simple method based on dynamic time warping, which is also based on the motion maps representation (a time-ordered version at a very coarse resolution). This method was provided by the organizers of the ChaLearn gesture challenge; it is publicly available from the challenge website.

### 5.1 Performance over the whole collection

In a first experiment, we applied the principal motion components approach to the whole GRC database of 54,000 gestures using both RGB and depth video. Results in terms of the Levenshtein score are shown in Table 2. For

**Table 2** Average (and standard deviation) of performance obtained by the proposed approach on the development (48,000 gestures), validation (2000 gestures) and final batches (4000 gestures)

Data set/type	RGB	DEPTH
Devel01–480	0.4079 (0.2387)	0.4103 (0.2068)
Valid01–20	0.3178 (0.2030)	0.3189 (0.1891)
Final01–20	0.2747 (0.1842)	0.2641 (0.1971)
Final21–40	0.2124 (0.1404)	0.2263 (0.1362)

this experiment, all the videos were automatically segmented. The translation gap was set to  $\tau = 5$  pixels, the scale for image downsizing was fixed to  $\gamma = 0.1$ , while the number of principal components was set to  $c = 10$ ; our choices were based on the results obtained in a preliminary study, see Sect. 5.2.

The performance of our method in the 480 development batches was worst than that obtained in the final and valid batches. This can be due to the difference in number of batches and the diversity of their vocabularies. In development batches, results using depth video are slightly worse than those obtained with RGB video, nevertheless, the difference in performance is not statistically significant according to a two-sample *t* test (*p* value = 0.8713). The corresponding differences for the validation (*p* value = 0.9879) and final (*p* value = 0.8607) batches were not statistically significant neither. Thus, we can conclude that the proposed method performs similarly, regardless of the type of information used: either RGB or depth video. This is advantageous as we do not need a Kinect sensor to achieve acceptable recognition performance with our method; one should note, however, that the standard deviation of performance is lower for depth video (in the 480 batches and for validation batches), hence, when available it would be preferable to use it.

The proposed approach took an average of 41.23 s to entirely process a batch<sup>3</sup> (i.e., training the PCA models from the training videos and labeling all the test videos, the time includes feature extraction and gesture segmentation). This means that a test video is processed in approximately 1 s, which makes evident the efficiency of our proposed method and can be used in real-time applications. One should note that we are using a standard PCA implementation, more efficient implementations of PCA could also be adopted if necessary (see, e.g., [25]).

The performance of our method in validation and final batches followed the same behavior as in development batches, although it is better. In fact, the performance of our method on validation and final batches is competitive with methods proposed by participants of the GRC. For

<sup>3</sup> Experiments were performed in a workstation with Intel® Core™i7-2600 CPU at 3.4 GHz and 8GB in RAM.

**Table 3** Performance of our method on batches 1–20 for the development and validation data sets using manual and automatic segmentation

Segmentation	Manual		Automatic	
	RGB	DEPTH	RGB	DEPTH
Devel01–20	0.2944	0.2741	0.3022	0.3016
Valid01–20	0.3151	0.3134	0.3178	0.3189

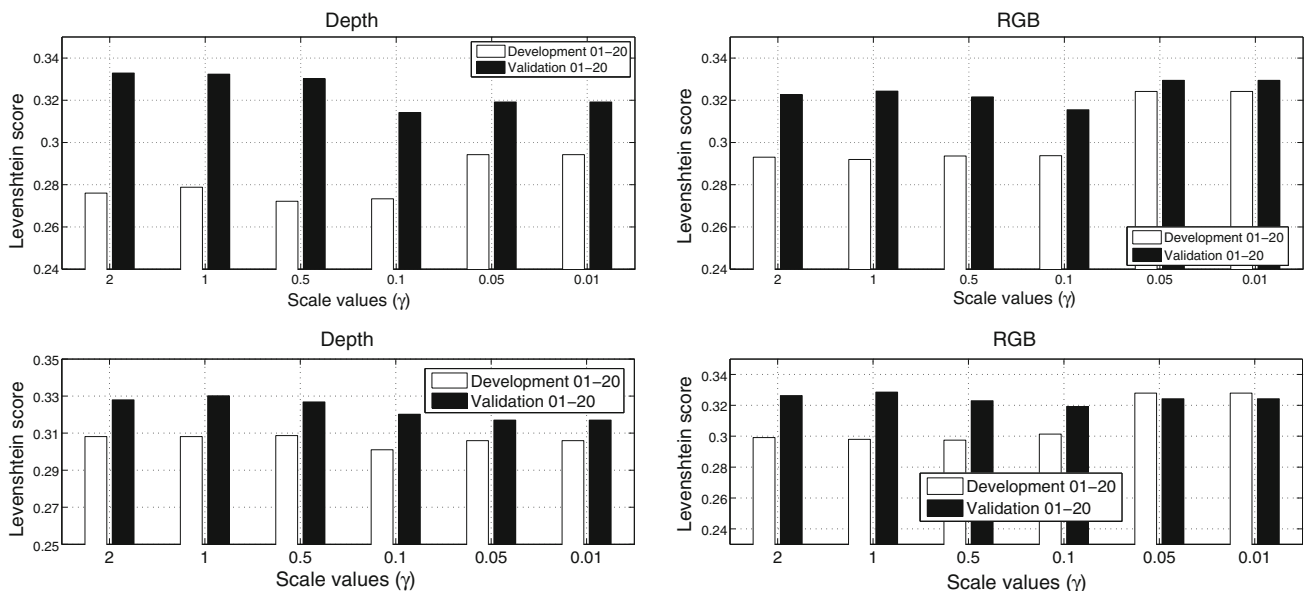
example, the results on the final batches 1–20 from Table 2 would be ranked 9th for the first round of the challenge, whereas for the final batches see 21–40 would be ranked 7th, see Sect. 5.3. One should note, however, that this method was not designed to handle all cases (e.g., static gestures). Competitive methods also used some handshape features to recognize static gestures, and that is beyond the scope of this paper.

We now evaluate the impact of gesture segmentation in the performance of the principal motion components technique. Table 3 compares the performance obtained in batches 1–20 for development and validation data when using manually segmented gestures and the automatic segmentation approach. As expected, using manual segmentation improves the performance of our approach, nevertheless the achieved improvements are modest. In fact, statistical tests did not reveal that the differences were statistically significant for both modalities (RGB and depth video) and batches (development and validation). Therefore, we can conclude that we can apply the principal motion approach using automated methods for gesture segmentation and still obtain competitive performance.

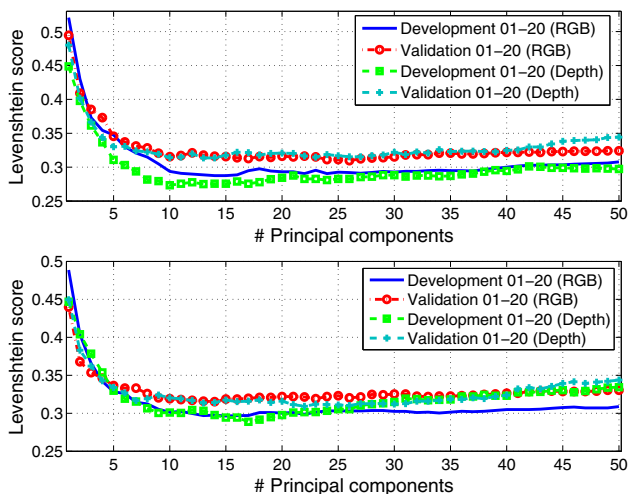
## 5.2 Performance under different parameter settings

Recall the only parameters of the proposed formulation are  $\gamma$  (the scale for downsizing the image, see Sect. 3.1), which is related to the size of the patches to generate motion maps, and  $c$ , the number of principal components used to generate PCA models, see Sect. 2.2. In a third experiment, we aimed to determine to what extent varying the values of such parameters affects the performance of the proposed approach. We proceeded by fixing the value of a parameter and then we evaluated the performance of our approach when varying the second parameter.

We start by analyzing the results in terms of the scale parameter ( $\gamma$ ). For this experiment, we fixed the number of principal components to  $c = 10$ . Results of this experiment are shown in Fig. 5. It can be seen that for both modalities there is not too much variation in the performance of the method for the different values we consider. This is due in part to the region growing preprocessing described in Sect. 3.1. The best results were obtained when  $\gamma = \{0.5, 0.1\}$ . Lower values of  $\gamma$  are preferred because the dimensionality of the motion maps is reduced and the proposed approach can be applied faster. Besides, the smaller the value of  $\gamma$ , the larger the size of the patches for the motion maps and the more robust is the approach to variations in the position of the user with respect to the camera. For instance, for  $\gamma = 0.1$ , the dimensionality of the motion maps is 192, the corresponding size of the patches is  $\approx 15 \times 27$ . Nevertheless, it can be seen from Fig. 5 that for smaller values than  $\gamma = 0.1$  the performance of principal motion components is worse.



**Fig. 5** Average performance of principal motion components for different scale values. For the *top plots* manual segmentation was used, while for the *bottom ones* automatic segmentation was performed



**Fig. 6** Average and standard deviation of the performance obtained by the proposed approach for different number of principal components. The *top plot* shows results when manual segmentation was used, whereas the *bottom graph* shows results obtained with automatic segmentation

For analyzing the influence of the number of components on the proposed technique, we fixed the value of the scale to  $\gamma = 0.1$  and varied the number of principal

components when building PCA models, experimental results are shown in Fig. 6. It can be seen from these plots that, in general, the performance of principal motion components is poor when using few components,  $c \in \{1, \dots, 5\}$ , for all the combinations of batches/modalities. The best performance for all the batches/modalities was obtained when using a number of components  $c \in \{10, \dots, 15\}$ ; the performance is somewhat stable for  $c \in \{10, \dots, 25\}$  and then it decreases considerably. This result may suggest that the best value for  $c$  is related to the number of gestures in the vocabularies ( $\{8, \dots, 12\}$ ). Actually, the average vocabulary lengths for development and validation batches are 9.7 and 9.5, respectively. Nevertheless, we did not find significant correlation between the best value for  $c$  and the size of the vocabulary ( $\rho = -0.0529$ ).

We also evaluated the correlation between the best value of  $c$  and the average and standard deviation of the length of training gestures, the minimum and maximum duration, the entropy on the duration of training gestures among other statistics. However, we did not find a statistically significant correlation value either. Thus, other aspects that have to do with the difficulty of vocabularies may have an

**Table 4** Sorted results for batches 1–20 of the development and validation data sets along with some characteristics of each batch

Devel01–20						Valid01–20					
B	M	T	c	LS	V	B	M	T	c	LS	V
d05	S	D	4	1.09	Gestuno-Disaster	v02	D	D	5	1.10	Helicopter
d08	S	D	19	2.25	Gestuno-Topography	v16	S	S	6	3.26	Referee-Volleyball2
d01	D	D	17	4.44	Canada-Aviation	v05	D	D	7	3.37	Tractor-Operation
d13	S	S	26	6.82	Crane-Hand	v17	S	S	8	8.70	Body-Language-Dom.
d04	D	S	12	8.89	Diving2	v10	S	D	8	11.96	Pantomime-Objects
d09	D	S	3	12.09	Referee-Volleyball1	v11	S	D	5	14.44	McNeill-Gesticulation2
d14	S	D	10	16.85	Diving	v04	S	S	10	16.30	Swat-Hand2
d07	D	S	12	19.57	Referee-Volleyball1	v13	S	D	7	19.78	Gestuno-Small-Animals
d17	S	S	10	19.78	Gang-Hand-Signals2	v06	D	S	32	21.11	Dance-Aerobics
d16	S	D	44	21.74	Gestuno-Landscape	v20	D	D	5	24.44	Canada-Aviation2
d20	S	S	7	22.99	Diving1	v12	S	D	22	31.11	Gestuno-Colors
d02	D	S	12	24.18	Referee-Wrestling1	v07	S	S	23	33.33	Referee-Wrestling2
d12	S	S	4	26.67	Italian-Gestures	v19	S	S	39	34.44	Taxi-SouthAfrica
d15	S	S	8	29.35	Swat-Hand1	v01	S	D	24	36.36	Motorcycle
d11	S	S	8	30.43	Music-Notes	v15	S	D	7	37.08	Italian-Gestures
d06	S	D	10	32.22	Diving3	v03	D	D	10	46.67	Diving2
d18	S	S	22	34.44	Taxi-SouthAfrica	v18	S	S	28	53.26	Music-Notes
d19	S	S	34	47.25	Mudra2	v08	D	D	11	54.35	Action-Objects
d10	S	S	11	48.35	Surgeon	v14	S	S	11	56.67	Mudra1
d03	S	S	8	60.87	Gang-Hand1	v09	S	S	10	67.42	Chinese-Numbers

Column **B** shows the id of the batch (either development,  $d$ , or validation,  $v$ ) and its number. Column **M** indicates whether the body of the user moves significantly ( $D$ ) or not ( $S$ ) when performing the gesture. Column **T** specifies the type of the gesture, which can be either static ( $S$ ) or dynamic ( $D$ ). Column **c** indicates the number of principal components used for the corresponding batch. Column **LS** shows the obtained Levenshtein score and column **V**, indicates the name of the vocabulary, see [16]

impact into the optimal value for  $c$ . In this regard, Table 4 shows information of the performance on each batch when using the optimal number of principal components for each of the development and validation batches (manual segmentation and RGB video were used).

Along with the performance obtained in each batch, the optimal value of  $c$  and some characteristics about the dynamism of gestures in batches are shown. Interestingly, a few principal components are enough to obtain outstanding performance for some batches [e.g., “Referee-Volleyball” (3), “Gestuno-disaster” (4), and “Helicopter” (5)], while a large value for  $c$  is used for some batches and yet the performance is poor (e.g., “Taxi-SouthAfrica” (39), and “Mudra2” (34)). It seems that easier vocabularies (too much motion, movement across the whole image, small inter-class similarity) require of less components than difficult ones (little motion, motion happening in small regions of the image, large inter-class similarity). Although is not easy to define what an easy/difficult vocabulary is.

Other interesting findings can be drawn from the results of this experiment. First, it can be seen that the principal motion components approach is very effective for some gestures. For example, performance similar to that of humans was obtained for “Helicopter”, “Gestuno-disaster”, “Gestuno-topography”, “Tractor-Operation” and “Canada-Aviation” vocabularies. These are highly dynamic gestures where motion happens in different regions of the image, thus the proposed approach can effectively capture the differences among gestures in the same vocabulary. In general, acceptable performance was obtained with the proposed approach when either the gesture is dynamic or the body of the user moves significantly when performing the gesture. The worst results were obtained when facing static gestures and users remained static when performed the gesture. This is a somewhat expected result as our approach attempts to exploit motion information.

Table 5 shows the average performance one would obtain when selecting the optimal value for  $c$  in each batch. The (hypothetical) relative improvements over the results reported in Table 3 range from 7.2 to 20 %. Hence, it is worth pursuing research on methods for selecting the number of principal components for each particular batch or gesture. Although one should note that the raw differences in performance are small: an improvement of 20.1 %

(RGB/Devel/MANUAL) corresponds to a raw difference of  $\approx 0.06$  in Levenshtein score. Development batches have a larger room for improvement than validation ones, the result is consistent with previous ones.

Summarizing, the principal motion components approach is rather robust to parameter selection. The scale parameter set to  $\gamma = 0.1$  achieved the best results for most of the configurations we evaluated. Although, other values obtained competitive performance as well. Selecting the number of principal components remains a difficult challenge, yet acceptable performance can be obtained by fixing  $c = 10$ . Finally, we showed evidence suggesting that the principal motion components method is particularly well suited to vocabularies involving a lot of motion, and when motion happens in different locations of the image.

### 5.3 Comparison with alternative methods

We now compare the performance of the principal motion approach to that obtained with alternative methods to solve the same one-shot learning problem. First, we compare the performance of principal motion components to that of other techniques that are based on similar ideas/features. Next, we compare the performance of the proposed technique to that obtained with other methods that were proposed during the ChaLearn gesture challenge [17, 18].

For the first comparison, we implemented the methods described in Table 6. The goal of this comparison is assessing whether using different features to represent the video, under the bag-of-frames formulation, could improve the performance of the one based on motion maps. We extracted the following (state-of-the-art) features widely used in computer vision: histograms of oriented gradients (HOG) [9]; histograms of oriented optical flow (HOF) [7]; space–time interest points with 3D HOG and HOF features [42]; and motion history images [6]. 2D HOG and HOF features were extracted from the frames themselves (HOG-I, HOF-I) and from difference images (HOG-M, HOF-M). For STIP-based features, we tried HOG-only, HOF-only and HOG+HOG 3D representations [42]. The variants of HOG, HOF and STIP-based features were represented under the bag-of-frames representation. Additionally, two variants of motion history images were implemented: the standard approach (MHI) [6] and another

**Table 5** Optimum performance that can be obtained with principal motion components when selecting the optimal value for  $c$  in each batch

Segmentation Data set/type	Manual		Automatic	
	RGB	DEPTH	RGB	DEPTH
Devel01–20	0.2351 (20.1 %)	0.2351 (14.2 %)	0.2749 (9.1 %)	0.2635 (12.6 %)
Valid01–20	0.2876 (8.7 %)	0.2876 (8.23 %)	0.2949 (7.2 %)	0.2832 (11.1 %)

It is shown between parentheses the relative improvement over the corresponding results from Table 3 (i.e., when using  $c = 10$  for all batches)



**Table 6** Description of the alternative methods we implemented for one-shot gesture recognition

ID	Representation	Recog.
PMC	Motion maps	PCR
HOG-I	HOG features from frames	PCR
HOG-M	HOG features from difference of frames	PCR
HOF-I	HOF features from frames	PCR
HOF-M	HOF features from difference of frames	PCR
STIP-F	STIP-HOF features	PCR
STIP-H	STIP-HOG features	PCR
STIP-HF	STIP-HOG + HOF features	PCR
PMC-SVM	Motion maps	SVM
HOG-SVM	HOG features from difference frames	SVM
HOF-SVM	HOG features from difference frames	SVM
STIP-BOW	STIP-HOG + HOF bag-of-features	KNN
MHI	Motion history image	TM
SMHI	Static-motion history image	TM

version that accounted for non-motion (SMHI). The latter variant aimed to be helpful for highly static gestures.

The different bag-of-frames representations were used for gesture recognition under the proposed PCA-based reconstruction-error technique. Also, we evaluated the recognition performance of supervised approaches using the same representations. For these methods, each vector of features (either motion maps, HOG, HOF, or 3D-HOG/HOF) is treated as an instance of a classification problem, where the class of the instance is the gesture from which the corresponding vector was extracted. In preliminary experimentation, we tried several classification methods including (linear discriminant analysis, neural networks, random forest, etc.), we report results for the best methods we found. For motion and static-motion history images, we used a template matching approach for recognition (correlation). Experimental results obtained with the considered variants and with the principal motion components approach are shown in Fig. 7.

From Fig. 7, it can be seen that principal motion components obtains the best performance for all but one of the configurations. HOG-M obtained the best results when using automatic segmentation and RGB video, the relative improvement was of 1.8 %. This result indicates the suitability of the reconstruction approach for one-shot gesture recognition under the bag-of-frames representation, which is not tied to a particular type of features. In fact, when using automatic segmentation the three methods: HOG-M, HOG-I and PMC obtained very similar results.

When manual segmentation was used, our approach outperformed the other methods by a considerable margin. The improvement over the nearest technique in performance (HOG-M) was of 36.12 and 45.9 % for RGB and

depth video, respectively. The widely used STIP features were not very useful for gesture recognition under neither the bag-of-frames nor the bag-of-visual-words formulations. This can be due to the fact that a single video is not enough to capture discriminative features. Actually, none of the supervised approaches to one-shot-gesture recognition performed decently. This is not surprising as we are using as labeled samples to features that may have high overlap with several gestures. It is interesting that the static history images outperformed the standard MHI technique [6].

Finally, we also compare the performance of principal motion components to that obtained by other authors that have used the ChaLearn Gesture Dataset [16]. We considered methods that have been already described in a scientific publication for this comparison. The performance of the considered methods as well as a brief description for each of them can be seen in Table 7.

It can be observed from Table 7 that the performance of the proposed approach is competitive with that obtained by the different methods. The best performance reported so far in a scientific publication is that reported by Wu et al. [44]. It is interesting that such method uses principal motion components as a preliminary step in their multi-layer architecture. Roughly, our method is used to determine if a gesture is dynamic or static. Dynamic gestures are treated with a method based on particle filtering and a tailored dynamic time warping; static ones are processed with a novel method that incorporates contextual information.

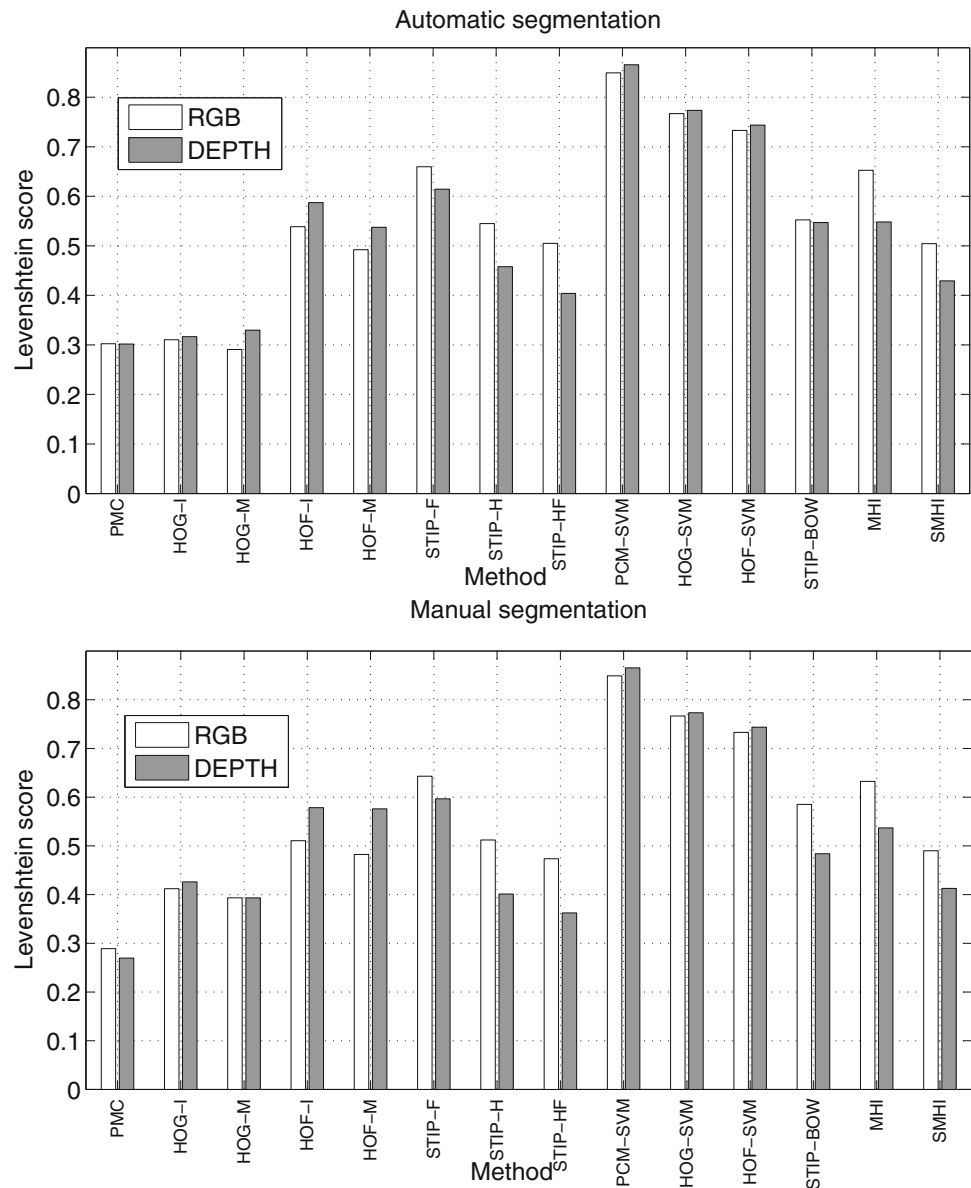
The performance of our automatic approach is close to that obtained by Malgireddy et al. [31] and Liu [26]. The former authors implemented a graphical model inspired in hidden Markov models that have been used for keyword spotting, both modalities (RGB and depth video) are used by the model. On the other hand, [26] represents videos with using a method based on higher-order singular value decomposition, recognition is done via least-squares regression for manifolds. Both approaches obtained outstanding performance in state-of-the-art data sets for human activity recognition and standard gesture recognition, besides they achieved acceptable results in data from ChaLearn Gesture challenge. The principal motion components approach obtained comparable performance to that techniques, hence, it is worth exploring the performance of our method on other closely related tasks.

Regarding the ChaLearn Gesture Challenge, the latest version of principal motion components would be ranked 9th and 7th in stages one<sup>4</sup> and two<sup>5</sup>, respectively. Principal motion components was proposed as a baseline method, whose simplicity and easy of implementation motivated

<sup>4</sup> <http://www.kaggle.com/c/GestureChallenge/leaderboard>.

<sup>5</sup> <http://www.kaggle.com/c/GestureChallenge2/leaderboard>.

**Fig. 7** Levenshtein score for the methods of Table 6 in the Development01–20 data set



participants to develop better methods. In this aspect, we accomplished our goal and exceed it by motivating other researchers to build better methods on top of our proposal.

## 6 Conclusions

We introduced a novel gesture recognition approach for the one-shot learning setting called principal motion components. The proposed approach represents the frames of a video by means of maps that account the amount of motion happening in spatial regions of the video. The bag of motion maps is used with a PCA-based recognition approach in which recognition error is used as a measure of gesture affinity.

We report experimental results in a large data set with 54,000 gestures, and two video modalities. Experimental results show that the proposed approach is very competitive, despite being simple and very efficient. The proposed method can work with RGB or depth video and obtain comparable performance. Likewise, the performance of the method does not degrade significantly when using manual or automatic gesture segmentation. We compare the performance of our approach to alternative methods we implemented ourselves and those reported by other researchers. Our approach compared favorably with some techniques and obtained close performance to others. We analyze the performance of our approach under different parameter settings and show characteristics of gestures that can be effectively recognized with it. This study

**Table 7** Description of the published methods for one-shot gesture recognition we consider for comparison

ID	Description	LS	References
Devel01–20			
MLS-Wu	Multi-layer Template+DTW	0.1950	[44]
GM-MM	Graphical model	0.2400	[31]
TM-Wu	Template matching	0.2600	[43]
PMC-M	PMC / Manual segmentation	0.2696	–
MF-LIU	Manifold learning	0.2873	[26]
PMC-A	PMC / Automatic segmentation	0.2890	–
TM-Mahbub	Template matching	0.3746	[28]
TM-2-Mahbub	Template matching	0.3125	[29]
Valid01–20			
GM-MM	Graphical model	0.2332	[31]
TM-Wu	Template matching	0.2968	[43]
PMC-A	PMC / Automatic segmentation	0.3178	–

revealed that the proposed approach is well suited for highly dynamic gestures.

There are several future work directions we would like to explore. First, we would like to study the suitability of the principal motion components approach for related tasks [12, 13, 23, 24], including gesture segmentation/spotting, key frame extraction and motion-based retrieval. Also, we are interested in developing alternative recognition methods that use the bag-of-frames representation. Other interesting areas for research include developing a hierarchical principal motion components formulation, and extending the proposed representation to spatiotemporal features.

## References

- Aggarwal JK, Cai Q (1999) Human motion analysis: a review. *Comput Vis Image Underst* 73(3):428–440
- Aggarwal JK, Ryoo MS (2011) Human activity analysis: a review. *ACM Comput Surveys* 43(3):Atr. 16
- Agrawal T, Chaudhuri S (2003) Gesture recognition using motion histogram. In: *Proceedings of the Indian National Conference of Communications*, pp 438–442
- Bekios-Calfa J, Buenaposada JM, Baumela L (2011) Class-conditional probabilistic principal component analysis: application to gender recognition. *Computacion y Sistemas* 14(4):383–391
- Bishop CM (2006) *Pattern Recognit Mach Learn*. Springer, New York
- Bobick AF, Davis JW (2001) The recognition of human movement using temporal templates. *IEEE Trans Pattern Anal Mach Intell* 23(3):257–267
- Chaudhry R, Ravichandran A, Hager G, Vidal R (2009) Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In: *Proceedings of the IEEE conference on pattern recognition and computer vision*. IEEE, pp 1651–1657
- Cheema MS, Eweiwia A, Bauckhage C (2014) Human activity recognition by separating style and content. *Pattern Recognit Lett* 50(1):130–138
- Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: *Proceedings of the IEEE conference on pattern recognition and computer vision*, pp 886–893
- Davis JW (2001) Hierarchical motion history images for recognizing human motion. In: *Proceedings of the IEEE workshop on detection and recognition of events in video*, pp 39–46
- Eichner M, Marín-Jiménez MJ, Ferrari V, Zisserman A (2012) 2d articulated human pose estimation and retrieval in (almost) unconstrained still images. *Int J Comput Vision* 99(2):190–214
- Escalera S, Baró X, Gonzalez J, Bautista MA, Madadi M, Reyes M, Ponce V, Escalante HJ, Shotton J, Guyon I (2015) ChaLearn looking at people challenge 2014: dataset and results. In: *Proceedings of ECCV 2014 workshops*, LNCS, vol 8925. Springer, pp 459–473
- Escalera S, Gonzalez J, Baró X, Reyes M, Guyon I, Athistos V, Escalante HJ, Sigal L, Argyros A, Sminchisescu C, Sclaroff S (2013) ChaLearn multi-modal gesture recognition 2013: grand challenge and workshop summary. In: *ICMI '13 Proceedings of the 15th ACM on international conference on multimodal interaction*. ACM, pp 365–368
- Freeman WT, Roth M (1995) Orientation histograms for hand gesture recognition. In: *Proceedings of the IEEE international workshop on automatic face and gesture recognition*
- Gomez JC, Moens MF (2012) PCA document reconstruction for email classification. *Comput Stat Data Anal* 56:741–751
- Guyon I, Athistos V, Jangyodsuk P, Escalante HJ (2014) The chlearn gesture dataset (cgd 2011). *Mach Vis Appl* 25(8):1929–1951
- Guyon I, Athistos V, Jangyodsuk P, Escalante HJ, Hamner B (2013) Results and analysis of the ChaLearn gesture challenge 2012. In: *WDIA: advances in depth image analysis and applications*, LNCS, vol 7854. Springer, pp 186–204
- Guyon I, Athistos V, Jangyodsuk P, Hammer B, Escalante HJ (2012) ChaLearn gesture challenge: design and first results. In: *Proceedings of the conference on computer vision and pattern recognition workshops*, Rhode Island. IEEE, pp 1–6
- Guyon I, Gunn S, Nikravesh M, Zadeh L (eds) *Feature extraction, foundations and applications*, *Studies in fuzziness and soft computing*, vol 207. Springer, Berlin
- Hoffmann H (2007) Kernel pca for novelty detection. *Pattern Recognit* 40(3):863–874
- Jolliffe IT (2002) *Principal component analysis*, 2nd edn. Springer, New York

22. Lee HK, Kim JH (1999) An hmm-based threshold model approach for gesture recognition. *IEEE Trans Pattern Anal Mach Intell* 21(10):961–973
23. Liu W, Tao D, Wang Y, Lu K (2015) Multiview hessian regularized logistic regression. *Signal Process* 110:101–107
24. Liu W, Li Y, Lin X, Tao D, Wang Y (2014) Hessian-regularized co-training for social activity recognition. *PLoS ONE* 9(9):1–10
25. Liu W, Zhang H, Tao D, Wang Y, Lu K (2015) Large-scale paralleled sparse principal component analysis. *Multimedia Tools Appl*
26. Liu YM (2012) Human gesture recognition on product manifolds. *J Mach Learn Res* 13(2012):3297–3321
27. Luo A, Kong X, Zeng G, Fan J (2010) Human action detection via boosted local motion histograms. *Mach Vis Appl* 21(3):377–389
28. Mahbub U, Imtiaz H, Roy T, Rahman S, Rahman-Ahad MA (2013) A template matching approach to one-shot-learning gesture recognition. *Pattern Recognit Lett* 34(15):1780–1788
29. Mahbub U, Roy T, Rahman S, Imtiaz H, Serikawa S, Rahman-Ahad MA (2013) A template matching approach to one-shot-learning gesture recognition. In: *Proceedings of the 1st international conference on industrial application engineering*, pp 186–193
30. Malagon-Borja L, Fuentes O (2009) Object detection using image reconstruction with pca. *Image Vis Comput* 27(1–2):2–9
31. Malgireddy MR, Nwogu I, Govindaraju V (2013) Language-motivated approaches to action recognition. *J Mach Learn Res* 14:2189–2212
32. Martin J, Crowley JL (1997) An appearance-based approach to gesture recognition. In: *Proceedings of the 9th international conference on image analysis and processing, LNCS, vol 1311*. Springer, pp 340–347
33. Mitra S (2007) Gesture recognition: a survey. *IEEE Trans Syst Man Cybern Part C Appl Rev* 37(3):311–324
34. Munoz-Salinas R, Medina-Carnicer R, Madrid-Cuevas FJ, Carmona Potayo A (2008) Histograms of optical flow for efficient representation of body motion. *Pattern Recognit Lett* 29:319–329
35. Pers J, Sulic V, Kristan M, Perse M, Polanec K, Kovaviv S (2010) Histograms of optical flow for efficient representation of body motion. *Pattern Recognit Lett* 31:1369–1379
36. Polana R, Nelson R (1994) Low level recognition of human motion. In: *Proceedings of the IEEE workshop on motion of non-rigid and articulated objects*, pp 77–82
37. Rabiner L (1989) A tutorial on hidden Markov models and selected applications in speech recognition. *Proc IEEE* 77(2):257–286
38. Shao L, Ji L (2009) Motion histogram analysis based key frame extraction for human/activity representation. In: *Proceedings of the Canadian conference on computer and robot vision*, pp 88, 92. IEEE
39. Tax D (2001) One-class classification. PhD thesis, Delft University of Technology
40. Turk M, Pentland A (1991) Eigenfaces for recognition. *J Cogn Neurosci* 3(1):71–86
41. Wan J, Ruan Q, Deng S, Li W (2013) One-shot learning gesture recognition from RGB-D data using bag of features. *J Mach Learn Res* 14:2549–2582
42. Wang H, Ullah MM, Klaser A, Laptev I, Schmid C (2009) Evaluation of local spatio-temporal features for action recognition. In: *Proceedings of the British machine vision conference*, pp 1–11
43. Wu D, Zhu F, Shao L (2012) One-shot learning gesture recognition from rgb-d images. In: *Proceedings of the conference on computer vision and pattern recognition workshops, Rhode Island*, pp 7–12. IEEE
44. Wu S, Pan W, Jiang F, Gao Y, Zhao D (2012) A mutiple-layered gesture recognition system for one-shot learning. In: *ICPR 2012 gesture recognition workshop*
45. Yi H, Rajan D, Chia LT (2005) A new motion histogram to index motion content in video segments. *Pattern Recognit Lett* 26:1221–1231