Tracking HOG Descriptors for Gesture Recognition

Mohamed Kaâniche, François Brémond

To cite this version:

Mohamed Kaâniche, François Brémond. Tracking HOG Descriptors for Gesture Recognition. AVSS 2009 - Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance, Sep 2009, Genova, Italy. pp.978-0-7695-3718-4, 10.1109/AVSS.2009.26 . hal-00428697

HAL Id: hal-00428697
https://hal.archives-ouvertes.fr/hal-00428697

Submitted on 29 Oct 2009

Tracking HoG Descriptors for Gesture Recognition

Mohamed Bécha Kaâniche, IEEE Member and François Brémond
INRIA Sophia Antipolis - Mediterranean Research Center – PULSAR Project
2004 route des Lucioles B.P. 93, 06902 Sophia Antipolis Cedex, France
Email: {mbkaanic,fbremond}@sophia.inria.fr

***Abstract*—We introduce a new HoG (Histogram of Oriented Gradients) tracker for Gesture Recognition. Our main contribution is to build HoG trajectory descriptors (representing local motion) which are used for gesture recognition. First, we select for each individual in the scene a set of corner points to determine textured regions where to compute 2D HoG descriptors. Second, we track these 2D HoG descriptors in order to build temporal HoG descriptors. Lost descriptors are replaced by newly detected ones. Finally, we extract the local motion descriptors to learn offline a set of given gestures. Then, a new video can be classified according to the gesture occurring in the video. Results shows that the tracker performs well compared to KLT tracker [1]. The generated local motion descriptors are validated through gesture learning-classification using the KTH action database [2].**

***Keywords*-Gesture Recognition; Motion Descriptors; Tracking HoG; Kalman Filter;**

I. Introduction

Local Motion Detection remains a challenging issue in Machine Vision and especially for Gesture Recognition. Indeed, during the last two decades, many approaches [3], [4] have been proposed to extract motion from video sequences. However, motion detection algorithms are still brittle due to illumination variability, background noise and occlusions. Therefore, it is crucial to detect motion features that account and describe more faithfully the concerned gesture. Global motion descriptors or local motion descriptors can be selected depending on the type of video.

In order to recognize gesture, some approaches are based on tracking body parts using 3D or 2D models of body part posture [5]. Other ones are based on learning global or local motion descriptors [6], [7], [8]. Techniques from the first category assume a good segmentation of body parts and handle two interacting models: a spatial model for posture and a temporal model for gesture. Thus, the recognition is usually computationally expensive and is strongly dependent on body part segmentation and tracking. However, techniques from the second category use a unique motion model consisting of sparse spatio-temporal descriptors. Therefore, these techniques are less dependent on the segmentation quality and are computationally cheaper.

In this paper, we propose a gesture recognition method based on local motion learning. First, for a given individual in a scene, we track feature points over its whole body to extract the motion of the body parts. Hence, we expect that feature points are sufficiently distributed over the body to capture fine gesture. We have chosen corner points as feature points to improve the detection stage and HoG as descriptor to increase the reliability of the tracking stage. Thus, we track the HoG descriptors in order to extract the local motion of feature points. Then, we learn offline a set of given gestures by clustering a set of local motion descriptors. Finally, we classify gestures occurring in a new video. We demonstrate the effectiveness of our motion descriptors by recognizing the actions of KTH database [2].

After over-viewing previous work on motion detection for gesture recognition in section II, we describe, in section III, how the 2D HoG Descriptor are computed and, in section IV, how temporal HoG descriptors are built by tracking the 2D descriptors. Section V presents the local motion descriptor for gesture learning-classification. Section VI illustrates experiments and results and section VII concludes this paper by summarizing the contributions and exposing future work.

II. Previous Work

As stated in previous section, we distinguish two categories of techniques in order to recognize gestures: (1) techniques using tracked 3D or 2D models of body parts and (2) techniques using local or global motion learning without body part models. Hereafter, we focus on the second category.

Yilmaz and Shah [7] have proposed to encode an action by an “action sketch” extracted from a silhouette motion volume obtained by stacking a sequence of tracked 2D silhouettes. The “action sketch” is composed of a collection of differential geometric properties (e.g. peak surface, pit surface, ridge surface) of the silhouette motion volume. For recognizing an action, the authors use a learning approach based on a distance and epipolar geometrical transformations for viewpoint changes. Lu and Little [9] propose to recognize gestures via maximum likelihood estimation with hidden markov models and a global HoG descriptor computed over the whole body. The authors extend their method in [8] by reducing the global descriptor size with principal component analysis. Gorelick and Blank [10] extract space-time saliency, space-time orientations and weighted moments from the silhouette motion volume. Gesture classification is performed using nearest neighbors algorithm and euclidean distance. Recently, Calderara et al. [3] introduce action signatures. An action signature is a 1D sequence of angles, forming a trajectory, which are extracted from a 2D map of adjusted orientation of the gradients of the motion-history image. A similarity measure is used for clustering and classification. As these method are using global motion, they depend on the segmentation quality of the silhouette which influences the robustness of the classification. Furthermore, local motion, which can help to discriminate similar gestures, can easily be lost with a noisy video sequence or with repetitive self-occlusion.

Local motion based methods overcome these limits by considering sparse and local spatio-temporal descriptors more robust to brief occlusions and to noise. For instance, Scovanner et al. [11] propose a 3-D (2D + time) SIFT descriptor and applied it to action recognition using the bag of word paradigm. Schuldt et al. [2] propose to use Support Vector Machine classifier with local space-time interest points for gesture categorization. Luo et al. [6] introduce local motion histograms and use an Adaboost framework for learning action models. More recently, Liu and Shah [4] apply Support Vector Machine learning on correlogram and spatial temporal pyramid extracted from a set of video-word clusters of 3D interest points. To go beyond the state of the art, we propose to track local motion descriptors over sufficiently long period of time thanks to a robust HoG tracker. The generated descriptors are used for gesture learning-clustering using the bag of word paradigm. Thus, we combine the advantages of global and local gesture descriptors to improve the quality of recognition.

III. 2D HoG Descriptor

To detect efficiently human gestures, we need to detect and isolate individuals from each other in the input video sequences. Thus, a people classifier and tracker is used to identify people and to assign features to each of them independently. The description of these algorithms is beyond this paper since they do not belong to the paper contributions.

A. Computing Corner Points

Once people have been detected, we compute a set of feature points for each individual. Feature points enable us to localize points where descriptors have to be computed since they usually represents body parts where the movement can be discernible. We have chosen to detect texture-based feature points (corners) since highly textured regions helps to detect motion. Feature points are extracted for each detected person using Shi-Thomasi corner detector [1] or Features from Accelerated Segment Test (FAST) corner detector [12]. Then, we select the most significant corners by maximizing the corner strength and ensuring a minimum distance between them. The minimum distance between corners ensures that the computed descriptors will not overlap and so will improve the point distribution through the individual's body.

B. Computing HoG Descriptor

For each corner point, we define a neighborhood (a small square centering on the considered feature point) where a 2D descriptor is computed. This 2D descriptor is based on Histogram of Oriented Gradients (HoG) [13]. Each feature point is associated to a descriptor block composed of 3×3 cells; each of them has a pixel size of 5×5. The gradient magnitude g and the gradient orientation θ are computed for all the pixels in the block using respectively equation 1 and equation 2 from the image gradients computed by simple 1D-filters (g_x and g_y).

$$g(u,v) = \sqrt{g_x(u,v)^2 + g_y(u,v)^2} \tag{1}$$

$$\theta(u,v) = \arctan \frac{g_y(u,v)}{g_x(u,v)} \tag{2}$$

For each cell $c_{ij}$where $(i,j) \in \{1,2,3\}^2$ in the block, we compute a feature vector f_{ij} by quantizing the unsigned orientation into K orientation bins weighted by the gradient magnitude as defined by equation 3.

$$f_{ij} = [f_{ij}(\beta)]^T_{\beta \in [1..K]} \tag{3}$$

where $f_{ij}(\beta)$ is defined by equation 4.

$$f_{ij}(\beta) = \sum_{(u,v) \in c_{ij}} g(u,v)\ \delta[bin(u,v) - \beta] \tag{4}$$

The function $bin(u,v)$ returns the index of the orientation bin associated to the pixel (u,v) and the function $\delta[]$ is the Kronecker delta. Therefore, the 2D descriptor of the block is a vector concatenating the feature vectors of all its cells normalized by the coefficient ρ which is defined in equation 5.

$$\rho = \sum_{i=1}^{3}\sum_{j=1}^{3}\sum_{\beta=1}^{K} f_{ij}(\beta) \tag{5}$$

Each cell encapsulates a local and specific information about the 2D descriptor which increases its trackability.

IV. Temporal HoG Descriptor

Temporal HoG descriptors are built by tracking 2D descriptors. We have developed a new tracking algorithm based on a frame-to-frame HoG tracker and using an extended kalman filter. A newly computed descriptor initializes a new tracking process through the extended kalman filter. For a tracked 2D descriptor d_{t-1} in the frame f_{t-1}, we determine the descriptor d_t in the frame f_t which can be identified to d_{t-1} through the "Predict" and "Correct" stages of the kalman filter. When a descriptor is lost, it may be replaced by a new computed descriptor associated to a newly detected corner. Fig. 1 illustrates the tracking algorithm of 2D HoG Descriptors. In the following subsections, we describe respectively the kalman filtering, the HoG tracking algorithm and the construction of the temporal 2D descriptor.

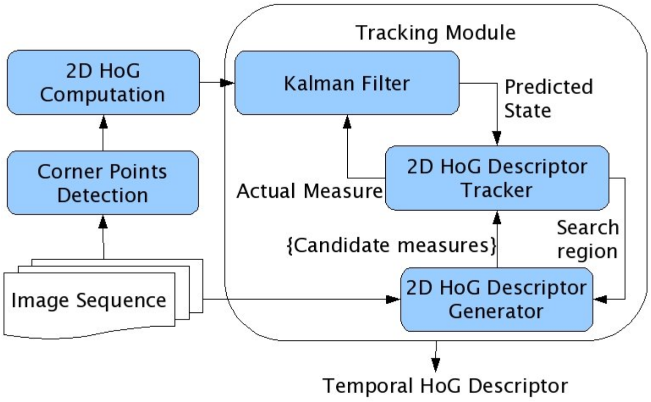


Figure 1. 2D HoG Descriptor tracking using extended kalman filter

A. Kalman Filtering and Descriptor Metrics

In order to track the descriptor position, we use a Kalman filter with a linear random-walk motion model for prediction. The basic idea of a kalman filtering based tracker is to recursively estimate the state vector given the last estimate and a new measurement which is made by the traditional tracker. The state vector $\mathcal{X}$ used by the kalman filter is defined by equation 6.

$$\mathcal{X} = [\, p_x \; p_y \; v_x \; v_y \; d \,]^T \tag{6}$$

where (p_x, p_y) is the position of the descriptor, (v_x, v_y) is its velocity and $\hat{d}$ is the descriptor value. A candidate descriptor (which has not been tracked yet) is described by a measurement vector $\mathcal{Z}$ as defined in equation 7.

$$\mathcal{Z} = [\, p_x \; p_y \; d \,]^T \tag{7}$$

When a corner point is newly detected and its correspondent descriptor is computed (first appearance of the individual or replacing a lost descriptor), the kalman filter is initialized with the state $\mathcal{X}^{(0)}$ with the speed of the individual's centroid and with the initial covariance matrix $P^{(0)}$ using large variances for the speed. At each tracking step, the kalman filter predicts a state $\hat{\mathcal{X}}_-^{(t)}$ from the previous actual state $\hat{\mathcal{X}}^{(t-1)}$ and the motion model. Using the predicted state and the apriori covariance matrix, the HoG tracking algorithm returns the actual current measure to the filter. Finally, the filter computes the actual current state by updating the predicted state with the innovation between the actual measure and the predicted measure. Hereafter, the static filtering of the descriptor is detailed. First of all, we define an error function $\mathcal{E}$ that measures the dissimilarity between two given descriptors $d^{(n)}$ and $d^{(m)}$ according to equation 8:

$$\mathcal{E}(d^{(n)}, d^{(m)}) = \sum_{i=1}^{9\times K} (d_i^{(n)} - d_i^{(m)})^2 \tag{8}$$

where $d_i^{(n)}$ and $d_i^{(m)}$ are respectively the i^{th} component of the descriptors $d^{(n)}$ and $d^{(m)}$. We want to calculate the estimate $\hat{d}$ such that the least square error between past measurements (i.e. $d^{(1)}, ..., d^{(t)}$) and the descriptor value d of the state vector is minimum. The solution for this minimization problem is given by this equation:

$$\hat{d} = \frac{1}{t}\sum_{i=1}^{t} d_i \tag{9}$$

Since we aim to have an estimate of the state of the descriptor at each step of the tracking, we use the recursive least square method by using the results of equation 10.

$$\underbrace{\hat{d}^{(t)}}_{\text{actual state}} = \underbrace{\hat{d}^{(t-1)}}_{\text{predicted state}} + \overbrace{\frac{1}{t}}^{\text{Gain}} \underbrace{\Big(\overbrace{d_t}^{\text{Actual measure}} - \overbrace{\hat{d}^{(t-1)}}^{\text{Predicted measure}} \Big)}_{\text{Innovation}} \tag{10}$$

Here the gain specifies how much do we pay attention to the difference between what we expected and what we actually get. Note that the gain decreases while the tracking advance which means that we become more and more confident in the descriptor estimation while the tracking progress. To decide whether the descriptor is correctly tracked, the following constraint should be verified:

$$\mathcal{E}(\hat{d}^{(t)}, \hat{d}^{(1)}) \leq \frac{9 \times K}{100} \tag{11}$$

In order to compute the actual measure, the HoG tracking algorithm needs two metrics: A distance between a state vector and a measurement vector and a confidence measure between these two vectors. Equation 12 defines a distance $\mathcal{D}$ between a candidate descriptor $\mathcal{Z}$ in the current frame and a tracked descriptor $\mathcal{X}$ through previous frames.

$$\mathcal{D}(\mathcal{Z}, \mathcal{X}) = \alpha \frac{\sqrt{\mathcal{E}(\mathcal{Z}_d, \mathcal{X}_d)}}{\sigma_d \sqrt{||\mathcal{Z}_d|| + ||\mathcal{X}_d|| + 1}} + \gamma \frac{||\mathcal{Z}_p - \mathcal{X}_p||}{\sigma_p} \tag{12}$$

where α and γ are empirically derived weight parameters, $\mathcal{Z}_d$, $\mathcal{X}_d$, $\mathcal{Z}_p$, $\mathcal{X}_p$ are respectively the estimated value and the descriptor position of $\mathcal{Z}$ and $\mathcal{X}$, and σ_d and σ_p are the covariance parameters extracted from the kalman filter's covariance matrix. The first term of the distance represents the scaled difference between the two descriptor values and the second term represents the difference between the candidate and predicted descriptor location. The confidence $\mathcal{C}$ in the candidate descriptor $\mathcal{Z}$ to correspond to the tracked descriptor $\mathcal{X}$ is defined by equation 13.

$$\mathcal{C}(\mathcal{Z}, \mathcal{X}) = \frac{1}{1 + \mathcal{D}(\mathcal{Z}, \mathcal{X})} \tag{13}$$

B. The HoG Descriptor Tracking Algorithm

The tracking algorithm consists in a downhill search around the predicted position by minimizing the quadratic error function $\mathcal{E}$. Thus, we determine search regions for next measurements using last states, predicted states and uncertainties and then we get new measurements in the search regions. Note that as a preliminary stage, all positions in the search area are used to compute the candidate descriptors. The confidence measure (as defined in equation 13) is used to select the best solution, if the downhill search gives several solutions.

For each descriptor, the tracking algorithm of 2D HoG descriptors is run as described hereafter. Where the $ellipse$ procedure returns the ellipse with foci ($\hat{\mathcal{X}}^{(t-1)}$, $\hat{\mathcal{X}}_{-}^{(t)}$) and eccentricity e defined by equation 14.

Algorithm 1 2D HoG descriptor tracking algorithm

```
Require: $\hat{\mathcal{X}}^{(t-1)}, \hat{\mathcal{X}}_{-}^{(t)}, P_t^{-}$ {Last estimated state, predicted state and error covariance matrix}
Ensure: $\mathcal{Z}^{(t)}$ {The actual measure}
$R \leftarrow ellipse(\hat{\mathcal{X}}^{(t-1)}, \hat{\mathcal{X}}_{-}^{(t)}, P_t^{-})$ {Compute the search region}
$\mathcal{S} \leftarrow \emptyset$ {Initialize the set of candidate measures}
for all $\mathcal{Z} \in R$ do
    if $\mathcal{E}(\mathcal{Z}, \hat{\mathcal{X}}_{-}^{(t)}) < \frac{9\,K}{100}$ then
        $\mathcal{S} \leftarrow \mathcal{S} \cup \{\mathcal{Z}\}$
    end if
end for
$maxConfidence \leftarrow 0$
for all $\mathcal{Z} \in \mathcal{S}$ do
    $confidence \leftarrow \frac{\mathcal{C}(\mathcal{Z}, \hat{\mathcal{X}}_{-}^{(t)}) + \mathcal{C}(\mathcal{Z}, \hat{\mathcal{X}}^{(t-1)})}{2}$
    if $confidence > maxConfidence$ then
        $maxConfidence \leftarrow confidence$
        $\mathcal{Z}^{(t)} \leftarrow \mathcal{Z}$
    end if
end for
```

$$e = \frac{c}{a} \tag{14}$$

where c is the focal distance and a is the semi-major axis of the ellipse which is computed as defined by formula 15.

$$a = \sqrt{c^2 + b^2} \tag{15}$$

where b is the semi-minor axis computed as defined by formula 16.

$$b = c + \sqrt{\sigma_x^2 + \sigma_y^2} \tag{16}$$

where σ_x and σ_y are the variance extracted from the covariance matrix P_t^{-} of the kalman filter. Note that $b \geq c$ which implies that $b^2 + c^2 > 2\,c^2$ and thus $a \geq \sqrt{2}\,c$ (using equation 15). This ensures that the search region has a minimum size according to the focal distance which is the half of the predicted motion of the descriptor.

C. The Temporal 2D Descriptor

The temporal 2D descriptor is the vector obtained by the concatenation of the final descriptor estimate $\hat{d}$ and the positions of the descriptor during the tracking process. The dimension of this vector is $9 \times K + 2 \times \ell$ where ℓ is the number of the 2D tracked positions. If we assume that $\mathcal{T}^d = [(x_1, y_1), ..., (x_\ell, y_\ell)]^T$ is the array of the descriptor d locations, then the temporal 2D descriptor is the heterogeneous vector $\mathcal{V} = [\hat{d}\ \mathcal{T}^d]^T$

V. Gesture Recognition

Hereafter, we present our learning-classification framework for gesture recognition based on the bag of word paradigm.

A. Local Motion Descriptor

Given the tracked position vector $\mathcal{T}^d$, we define the line trajectory vector $\mathcal{L}$ as:

$$\mathcal{L}^d = [(w_1, h_1), ..., (w_{\ell-1}, h_{\ell-1})]^T \tag{17}$$

where $w_i = x_{i+1} - x_i$ and $h_i = y_{i+1} - y_i$. The trajectory orientation vector $\Theta^d = [\theta_1, ..., \theta_{\ell-2}]^T$ is computed thanks to the formula defined by equation 18.

$$\forall i \in [1, \ell - 2];\ \theta_i = \arctan(h_{i+1}, w_{i+1}) - \arctan(h_i, w_i) \tag{18}$$

where the $\arctan$ function returns the orientation of the given line with respect to the x axis. Since $-2\pi \leq \theta_i \leq 2\pi$, we normalize the vector by dividing all its components by 2π. The resulting vector is noted $\tilde{\Theta}^d$. The local motion descriptor is defined as the concatenation of the descriptor estimation $\hat{d}$ which indicates the texture involved in the motion and the normalized trajectory orientation vector $\tilde{\Theta}^d$ which represents the motion. Its dimension is $9 \times K + \ell - 2$. To reduce the dimension of the local motion descriptor, we apply Principal Component Analysis (PCA) and project the θ_i on the three first principal axis $\hat{\theta}_1, \hat{\theta}_2, \hat{\theta}_3$. Thus we get a final local motion descriptor $\hat{\Theta}^d = [d\ \hat{\theta}_1\ \hat{\theta}_2\ \hat{\theta}_3]^T$. Compared to the global PCA-HOG descriptor proposed by [8] (one global HoG descriptor for each gesture/action), the proposed gesture/action descriptor consists in a set of local motion descriptors which accounts more faithfully for local motion. Instead of computing a global HoG volume from a person already tracked, we use local HoGs tracked independently. Our method contrasts from traditional local motion methods by using the tracking process of 2D descriptors instead of 2D descriptor time-volume.

B. Gesture Learning-Classification

In order to recognize gestures, we propose to learn and classify gesture based on the k-means clustering algorithm and the k-nearest neighbors classifier. For each video in a training dataset, we generate all correspondent local motion descriptors and annotate them with the correspondent gesture. Then, for each training video taken separately, the descriptors are clustered into k clusters using the k-means clustering algorithm as proposed by [14]. The k parameter is set up empirically. Each cluster is associated to its corresponding gesture, so similar clusters can be labeled with different gestures. Finally, with all generated clusters

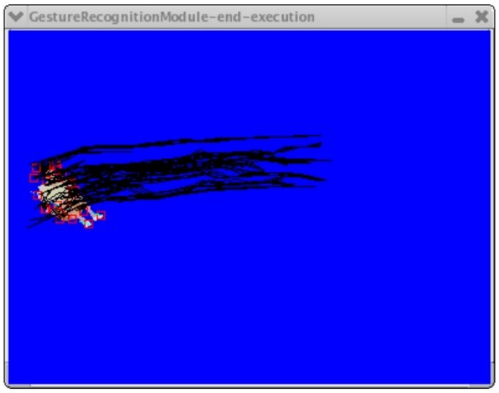


Figure 2. Synthetic dataset: Red rectangles represent the tracked descriptors and the lines represent their trajectories.

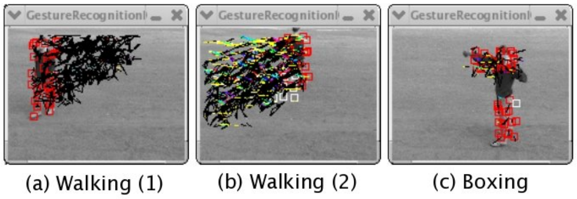
as a database, the k-nearest neighbors classifier is used to classify gestures occurring in the test dataset. A video is classified according to the amount of neighbors which have voted for a given gesture providing the likelihood of the recognition.

VI. Experiments and Results

We have tested our tracking algorithm on two datasets: a synthetic dataset and a real dataset. Then we have validated the local motion descriptors with KTH database [2] which consists of 598 videos of 6 actions: walking, jogging, running, boxing, hand-clapping and hand-waving. The database is splitted into three independent datasets: (1) a training dataset, (2) a validation dataset for tuning parameters and (3) a testing dataset for evaluation. We have chosen to use $K = 9$ orientation bins. So the size of a 2D HoG Descriptor is 81. We have found that the optimal values for the empirical weights α and γ of the distance $\mathcal{D}$ are respectively 5 and 2. The minimum distance between corner points (i.e. HoG descriptors) is set to 9. These parameters and the noise variances of the kalman filter have been fixed by testing the algorithm on the validation dataset of the KTH database. Hereafter, we detail the results for each experiment.

A. Tracking Results on Synthetic Sequence

In order to demonstrate the effectiveness of the HoG tracker, we have tested it on a synthetic dataset as illustrated by fig. 2. The generated sequence is composed of 247 frames. During the whole sequence our tracker has lost only 39 descriptors while the mean number of descriptors per frame is 35. The lost of descriptor occurs when there is a sudden and strong change in the motion direction. This is due to the linear motion model used by the kalman filter. An improvement to cope with this high temporal gradient is to use more sophisticated motion model (e.g. Brownian motion).



(a) Walking (1) (b) Walking (2) (c) Boxing

Figure 3. KTH dataset: Red rectangles represent the tracked descriptors, white ones represent the newly detected descriptors and the lines represent the trajectories of tracked descriptors.

Table I
Results of HoG tracking module with the validation dataset of the KTH database.

	Mean	Var	Min	Max
#Desc./frame	22.32	03.37	15.15	34.38
#Tracked/frame	20.70	03.57	15.00	27.88
#Lost/frame	01.62	01.50	00.15	06.50

B. Tracking Results on Real Sequence

To go forward in the validation of the developed tracker, we have tested it on the validation dataset of the KTH database. Fig. 3 illustrates the results on KTH database and table I resumes the obtained results. This table describes the mean, variance, minimum and maximum values of the number of descriptors (detected, tracked and lost) per frame. The proposed tracker outperforms the KLT feature point tracker [1] (there are only nine tracked feature points per frame in average) which is sensitive to noise and thus losts many more feature points.

C. Application to Gesture Recognition

We have trained our algorithm on the KTH training dataset and tested it on the corresponding test dataset. Results are illustrated by the confusion matrix II and are compared to the state of the art method in table III. We have obtained better or slightly better results than recent methods. We have also found out that FAST corners outperform Shi-Tomasi corner which is consistent with results in [12]. Note that even though Kim et al. [15] obtain slightly better results, their results are not comparable to ours since they use a different experimental protocol (Leave-one-out cross-validation). The gesture recognition results have been obtained mainly to demonstrate the effectiveness of the HoG tracker. These preliminary results are encouraging. Thus, the next step will consist in exploring more complex clustering techniques than k-means to be able to recognize gestures on more challenging videos (e.g. everyday life videos). We can see that very few descriptors are lost and most of them are correctly tracked throughout the video. Only some descriptors on legs and arms are regularly lost due to self occlusions.

Table II
CONFUSION MATRIX FOR THE CLASSIFICATION USING SHI-TOMASI (UPPER VALUES) AND FAST CORNER POINTS (LOWER VALUES).

	W.	J.	R.	B.	H.C.	H.W.
W.	0.95 0.97	0.03 0.03	0.02 0.00	0.00 0.00	0.00 0.00	0.00 0.00
J.	0.03 0.02	0.85 0.91	0.10 0.07	0.02 0.00	0.00 0.00	0.00 0.00
R.	0.05 0.03	0.07 0.05	0.88 0.92	0.00 0.00	0.00 0.00	0.00 0.00
B.	0.00 0.00	0.00 0.00	0.00 0.00	0.95 0.97	0.03 0.02	0.02 0.01
H.C.	0.00 0.00	0.00 0.00	0.00 0.00	0.05 0.03	0.88 0.92	0.07 0.05
H.W.	0.00 0.00	0.00 0.00	0.00 0.00	0.02 0.01	0.01 0.00	0.97 0.99

Table III
COMPARISON OF DIFFERENT RESULTS.

Method	Variant	Precision
Our method	Shi-Tomasi	91.33%
	FAST	94.67%
Liu and Shah [4]	SVM VWCs	91.31%
	VWC Correl.	94.16%
Luo et al. [6]		85.10%
Kim et al. [15]		95.33%

VII. CONCLUSIONS

We have presented a novel local motion descriptor for gesture recognition. First, we select corner points in order to compute and track HoG descriptors. Second, we learn local motion descriptors using k-means with PCA. Finally, we classify the gestures using the k-nearest neighbors algorithm. Our main contribution is to combine local motion and global motion techniques by tracking reliably HoG descriptors over long periods of time. Results on synthetic data and on KTH database shows the effectiveness of our approach. For future work, we are going forward in the validation process by testing the proposed algorithm on multi-view dataset (e.g. IXMAS) and real world dataset (e.g. TrecVid). Additionally, we plan to improve the tracking algorithm through the use of the unscented kalman filter instead of the extended kalman filter. Finally, we will improve the learning-classification process by proposing a new learning-classification framework.

REFERENCES

[1] J. Shi and C. Tomasi, “Good features to track,” in *International Conference on Computer Vision and Pattern Recognition*. Seattle, WA, USA: Springer, June 1994, pp. 593–600.

[2] C. Schuldt, I. Laptev, and B. Caputo, “Recognizing human actions: a local svm approach,” in *International Conference on Pattern Recognition*, vol. 3. Cambridge, UK: IEEE Computer Society Press, August 23-26 2004, pp. 32–36.

[3] S. Calderara, R. Cucchiara, and A. Prati, “Action signature: A novel holistic representation for action recognition,” in *International Conference on Advanced Video and Signal Based Surveillance*. Washington, DC, USA: IEEE Computer Society Press, 2008, pp. 121–128.

[4] J. Liu and M. Shah, “Learning human actions via information maximization,” in *International Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society Press, June 2008, pp. 1–8.

[5] C.-W. Chu and I. COHEN, “Posture and gesture recognition using 3d body shapes decomposition,” in *International Conference on Computer Vision and Pattern Recognition: Workshops*, vol. 3. Washington, DC, USA: IEEE Computer Society, 2005, p. 69.

[6] Q. Luo, X. Kong, G. Zeng, and J. Fan, “Human action detection via boosted local motion histograms,” *Machine Vision and Applications*, November 2008. [Online]. Available: http://dx.doi.org/10.1007/s00138-008-0168-5

[7] A. Yilmaz and M. Shah, “A differential geometric approach to representing the human actions,” *Computer Vision and Image Understanding*, vol. 109, no. 3, pp. 335–351, 2008.

[8] W.-L. Lu and J. J. Little, “Simultaneous tracking and action recognition using the pca-hog descriptor,” in *Computer and Robot Vision, 2006. The 3rd Canadian Conference on*, Quebec, Canada, June 2006, pp. 6–13.

[9] W. L. Lu and J. J. Little, “Tracking and recognizing actions at a distance,” in *Proceedings of the ECCV Workshop on Computer Vision Based Analysis in Sport Environments (CVBASE ’06)*, Graz, Austria, May 2006.

[10] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, “Actions as space-time shapes,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29, no. 12, pp. 2247–2253, 2007.

[11] P. Scovanner, S. Ali, and M. Shah, “A 3-dimensional sift descriptor and its application to action recognition,” in *ACM International Conference on Multimedia*. New York, NY, USA: ACM, 2007, pp. 357–360.

[12] E. Rosten and T. Drummond, “Machine learning for high-speed corner detection,” in *European Conference on Computer Vision*, vol. 1. Graz, Austria: Springer, May 7-13 2006, pp. 430–443.

[13] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *International Conference on Computer Vision and Pattern Recognition*, vol. 1. San Diego, CA, USA: IEEE Computer Society Press, June 20-25 2005, pp. 886–893.

[14] C. Ding and X. He, “K-means clustering via principal component analysis,” in *International Conference on Machine Learning*. New York, NY, USA: ACM Press, 2004, pp. 225–232.

[15] T.-K. Kim, S.-F. Wong, and R. Cipolla, “Tensor canonical correlation analysis for action classification,” in *International Conference on Computer Vision and Pattern Recognition*, June 2007, pp. 1–8.