Simultaneous Tracking and Action Recognition using the PCA-HOG Descriptor

Wei-Lwun Lu, James J. Little
Department of Computer Science
University of British Columbia
Vancouver, BC, CANADA
{vailen, little}@cs.ubc.ca

Abstract

This paper presents a template-based algorithm to track and recognize athlete's actions in an integrated system using only visual information. Conventional template-based action recognition systems usually consider action recognition and tracking as two independent problems, and solve them separately. In contrast, our algorithm emphasizes that tracking and action recognition can be tightly coupled into a single framework, where tracking assists action recognition and vice versa. Moreover, this paper proposes to represent the athletes by the PCA-HOG descriptor, which can be computed by first transforming the athletes to the grids of Histograms of Oriented Gradient (HOG) descriptor and then project it to a linear subspace by Principal Component Analysis (PCA). The exploitation of the PCA-HOG descriptor not only helps the tracker to be robust under illumination, pose, and view-point changes, but also implicitly centers the figure in the tracking region, which makes action recognition possible. Empirical results in hockey and soccer sequences show the effectiveness of this algorithm.

1 Introduction

Vision-based tracking and action recognition systems have gained more and more attention in the past few years because of their potential applications on smart surveillence systems, advanced human-computer interfaces, and sport video analysis. In the past decade, there has been intensive research and giant strides in designing algorithms for tracking humans and recognizing their actions [6]. Briefly, the task of visual tracking is to predict and update the target's position, velocity, and size based on the video, while the task of visual action recognition is to recognize (classify) the person's action given the person's most recent appearance.

In this paper, we develop a system that integrates visual tracking and action recognition, where tracking assists action recognition and vice versa. The first novelty of this paper is to represent the target by the PCA-HOG descriptor. The HOG descriptor and its variants have been used in human detection [4] and object class recognition [15], and have been shown to be very distinctive and robust. This paper shows that the HOG descriptor can be also exploited in visual tracking and action recognition as well. Moreover, after projecting the HOG descriptor to its principal subspaces, the speed of the tracking and action recognition system can be increased without loss of accuracy. The second novelty of this paper is an iterative algorithm that solves the tracking and action recognition problems together using learned templates. Given the examples of athletes' appearances of different actions, we learn the templates and the transition between the templates offline. During the runtime, we use the templates and the transition matrix to classify the athlete's actions and compute the most probable template sequence consistent with the visual observation. Moreover, the last template of the most probable sequence can be used for the visual tracking system to search for the next positions and sizes of the atheletes, which determines the next visual observation of the athletes.

This paper is organized as follows: In Section 2, we review some related work in visual tracking and action recognition. In Section 3, we introduce the PCA-HOG representation. Section 4 details our tracking and action recognition algorithms. The experimental results in hockey and soccer sequences are shown in Section 5. Section 6 concludes this paper.

2 Previous Work

The task of vision-based action recognition can be described as follows: given a sequence of consecutive images containing a single person, we have to determine the *action* of the person. Therefore, the vision-based action recognition problem is indeed a classification problem of which the input is a set of images, and the output is a finite set of labels.

The input of a vision-based action recognition system is usually a set of *stabilized* images: figure-centric images containing the whole body of a single person (including the limbs). Fig. 3 provides some examples of the stabilized images. In order to obtain the stabilized images, vision-based action recognition systems usually run visual tracking and stabilization algorithms prior to the action recognition. Then, the systems will extract relevant features such as pixel intensities, edges, optical flow from the images. These features are fed into a classification algorithm to determine the action of the person. For examples, Yamato *et al.* [23] transform a sequence of stabilized images to mesh features, and use a Hidden Markov Model classifier to recognize the actions. Efros *et al.* [5] transform the images to novel motion descriptors computed by decomposing the optical flow of images into four channels. Then, a nearest-neighbor algorithm is performed to determine the person's actions. In order to tackle the stabilization problem, Wu [21] develop an algorithm to automatically extract figure-centric stabilized images from the tracking system. They also propose to use Decomposed Image Gradients (DIG), which can be computed by decomposing the image gradients into four channels, to classify the person's actions.

The task of visual tracking can be defined as follows: given the initial state (usually the position and size) of a person in the first frame of a video sequence, the tracking systems will continuously update the person's state given the successive frames. During the tracking algorithm, relevant features should be extracted from the images. Some systems use intensities or color information [1, 19], some use shape information [7, 9], and some use both [22]. The tracking problem can be solved either deterministically [1, 3, 10] or probabilistically [9, 22]. In order to improve the performance and robustness of the tracker, many systems also combine the tracking system with other systems such as object detection [18] and object recognition [11].

In order to simplify the tracking problem, many trackers use a fixed target appearance [3, 18, 19]. However, having a fixed target appearance is optimistic in the real world because the view point and the illumination conditions may change, and the person constantly change their poses. In order to tackle this problem, [8, 12] project the images of the person to a linear subspace and incrementally update the subspace based on the new images. These systems are efficient; however, they have difficulties recovering from drift because the linear subspace also accumulates the information obtained from the images containing only the background. Jepson *et al.* [10] propose the WSL tracker of which the appearance model is dominated by each either the stable (S), wandering (W), or lost (L) components. They use an online EM algorithm to update the parameters of the stable component, and therefore the tracker is robust under smooth appearance changes.

The tracking systems that most resemble ours are Giebel *et al.* [7] and Lee *et al.* [11]. Giebel *et al.* [7] learn the templates of the targets and the transition between the templates from examples. However, they do not divide the templates into different actions. During the tracking, they use particle filtering [19] to infer both the next template and the position and size of the target. Lee *et al.* [11] introduce a system that combines face tracking and recognition. They also learn templates of faces and the transition between the templates from examples, and partition the templates into different groups according to the *identity* of the face. During the runtime, they first recognize the identity of the face based on the history of the tracking results. Knowing the identity of the face, the target template used by the tracker can be more accurately estimated, and thus improve the robustness of the tracker.

3 The PCA-HOG Descriptor

In this paper we propose to use the PCA-HOG descriptor to represent the athletes. The PCA-HOG descriptor can be constructed by first transforming the tracking region to the grids of Histograms of Oriented Gradient (HOG) descriptor [4], and then using Principal Components Analysis (PCA) to project the HOG descriptor to a linear subspace.

The HOG representation is inspired by the SIFT descriptor proposed by Lowe [13]. It can be constructed by dividing the tracking regions into non-overlapping grids, and then computing the orientation histograms of the image gradient of each grid (Fig. 1). Since the HOG/SIFT representation has been shown to be very distinctive and robust under small affine transformation and illumination changes, it has gained widespread use among the object recognition and object detection community [4, 13, 15]. In our previous work [14], we have also studied using the HOG descriptor to the tracking and action recognition problems.

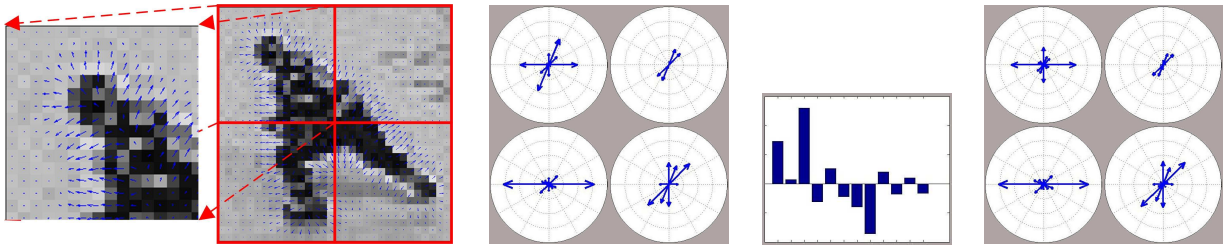
Let $\boldsymbol{I} \in \mathbb{R}^{m \times n}$ denote an image of width m and height n, and $\boldsymbol{I}(x, y)$ denote the pixel intensity in position (x, y), the PCA-HOG descriptor can be computed by the following procedures:

1. The image $\boldsymbol{I}$ is filtered by a symmetric low-pass Gaussian filter of size w_g with standard deviation σ_g. Then, we compute the image gradient along the x and y direction by a 1-D centered mask $[-1, 0, 1]$:

$$\begin{aligned} \boldsymbol{g}_x(x, y) &= \boldsymbol{I}(x+1, y) - \boldsymbol{I}(x-1, y) \quad \forall x, y \\ \boldsymbol{g}_y(x, y) &= \boldsymbol{I}(x, y+1) - \boldsymbol{I}(x, y-1) \quad \forall x, y \end{aligned} \tag{1}$$

   where $\boldsymbol{g}_x(x, y)$ and $\boldsymbol{g}_y(x, y)$ denotes the x and y components of the image gradient, respectively.

2. The magnitude $\boldsymbol{m}(x, y)$ and orientation $\boldsymbol{\theta}(x, y)$ of the



(a) (b) (c) (d)

Figure 1. Examples of the PCA-HOG descriptor: (a) The image gradient. (b) The HOG descriptor with a 2×2 grid and 8 orientation bins. (c) The PCA-HOG descriptor with $n_p = 12$. (d) The reconstructed HOG descriptor.

image gradient are computed by

$$\boldsymbol{m}(x,y) = \sqrt{\boldsymbol{g}_x(x,y)^2 + \boldsymbol{g}_y(x,y)^2} \quad (2)$$

$$\boldsymbol{\theta}(x,y) = tan^{-1}(\boldsymbol{g}_y(x,y)/\boldsymbol{g}_x(x,y)) \quad (3)$$

In order to make the PCA-HOG representation insensitive to the color of the athletes' uniform, we use the unsigned orientation of the image gradient suggested by Dalal *et al.* [4], i.e.,

$$\tilde{\boldsymbol{\theta}}(x,y) = \begin{cases} \boldsymbol{\theta}(x,y) + \pi & \text{if } \boldsymbol{\theta}(x,y) < 0 \\ \boldsymbol{\theta}(x,y) & \text{otherwise} \end{cases} \quad (4)$$

3. We partition the image into $s_w \times s_h$ non-overlapping grids. For each grid, we quantize the orientation $\tilde{\boldsymbol{\theta}}(x,y)$ for all pixels into s_b orientation bins weighted by its magnitude $\boldsymbol{m}(x,y)$.

4. We normalize each feature by the sums of all features. The resulting feature vector, $\boldsymbol{H} \in \mathbb{R}^{n_f}$, $n_f = s_w \times s_h \times s_b$, is the HOG descriptor of the image $\boldsymbol{I}$.

5. Let $\boldsymbol{\Gamma} \in \mathbb{R}^{n_p \times n_f}$ denote the first n_p principal components learned from the HOG descriptors of the training images. We project the HOG descriptor $\boldsymbol{H}$ to the linear subspace spanned by the principal components $\boldsymbol{\Gamma}$, i.e.,

$$\boldsymbol{Y} = \boldsymbol{\Gamma}^T(\boldsymbol{H} - \bar{\boldsymbol{H}}) \quad (5)$$

where $\bar{\boldsymbol{H}} \in \mathbb{R}^{n_f}$ is the mean HOG descriptor of all training images, and $\boldsymbol{Y} \in \mathbb{R}^{n_p}$ is the PCA-HOG descriptor of the image $\boldsymbol{I}$.

Fig. 1 (a) gives an example of the image gradient and Fig. 1 (b) and (c) are its corresponding HOG and PCA-HOG descriptor, respectively. Fig. 1 (d) shows the reconstructed HOG descriptor computed from the PCA-HOG descriptor. We can observe that there is no significant difference between the original HOG descriptor and the reconstruction.

Using the PCA-HOG representation has several advantages. Firstly, since the HOG/SIFT representation is based on *edges*, the rectangular tracking region can enclose the entire body of the athletes (including the limbs) without sacrificing the discrimination between the foreground and background. This is especially the case in tracking athletes in sports such as hockey and soccer because the background is usually homogeneous. Another attractive property of the HOG/SIFT representation is that it is insensitive to the changes of athletes' uniform because the use of the unsigned orientation of the image gradient. This enables the tracker to focus on the *shape* of the athletes but not the colors or textures of the uniform. Secondly, the HOG/SIFT representation improves the robustness of the tracker because it is robust to small misalignments and illumination changes [4, 16, 15]. Thirdly, the HOG/SIFT representation implicitly centers the figure in the tracking region because it preserves some spatial arrangement by dividing the tracking region into non-overlapping grids. In contrast to [5, 21], no further stabilization techniques need to be used to center the figure in the tracking region. This helps integrate tracking and action recognition into a single framework. Finally, using PCA with the HOG descriptor reduces the dimensionality of the feature vector and thus greatly increases the processing speed.

4 Tracking and Action Recognition

The probabilistic graphical model of our system (Fig. 2) is a hybrid Hidden Markov Model with two first-order Markov processes. The first Markov process, $\{E_t; t \in \mathbb{N}\}$, contains discrete random variable E_t denoting the template at time t. The second Markov process, $\{\boldsymbol{X}_t; t \in \mathbb{N}\}$, contains continuous random variable $\boldsymbol{X}_t$ denoting the position, velocity, and size of a single athlete at time t. The random variable $\{\boldsymbol{I}_t; t \in \mathbb{N}\}$ denote the frame of the video at time t, and the deterministic parameter α_t denote the action of the athlete at time t. The joint distribution of the entire system

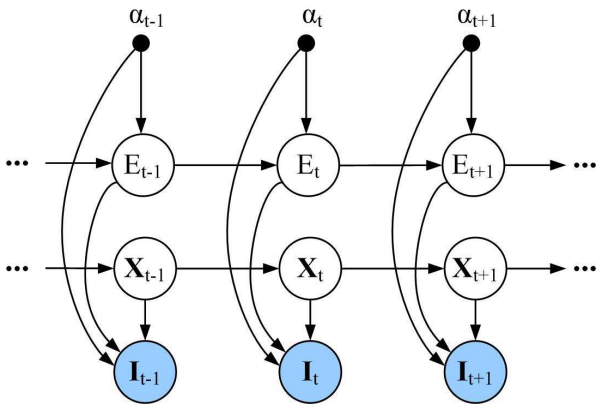


Figure 2. Probabilistic Graphical Model.

is given by:

$$p(\boldsymbol{X}, \boldsymbol{E}, \boldsymbol{I}|\boldsymbol{\alpha}) = p(\boldsymbol{X}_0)p(E_0)\prod_t p(\boldsymbol{I}_t|\boldsymbol{X}_t, E_t, \alpha_t)\cdot \prod_t p(E_t|E_{t-1}, \alpha_t)\prod_t p(\boldsymbol{X}_t|\boldsymbol{X}_{t-1}) \quad (6)$$

The transition distribution $p(E_t|E_{t-1}, \alpha_t)$ is defined as:

$$p(E_t = j \mid E_{t-1} = i, \alpha_t) \;=\; A_{ij}^{\alpha_t} \quad (7)$$

where A_{ij}^{α} is the transition distribution between templates i and j of action α_t.

The continuous random variable $\boldsymbol{X}_t$ is defined as $\boldsymbol{X}_t = \{x_t, y_t, v_t^x, v_t^y, w_t\}^T$, where (x_t, y_t) denotes the center of the athlete, w_t denotes the width of the rectangular tracking region (we currently fix the aspect ratio of the tracking region), and v_t^x and v_t^y denote the velocity of the athlete along the x and y direction, respectively. The transition distribution $p(\boldsymbol{X}_t|\boldsymbol{X}_{t-1})$ is a Linear Gaussian:

$$p(\boldsymbol{X}_t \mid \boldsymbol{X}_{t-1}) \;=\; \mathcal{N}(\boldsymbol{X}_t \mid \boldsymbol{B}\boldsymbol{X}_{t-1},\; \boldsymbol{\Sigma}_{\boldsymbol{X}}) \quad (8)$$

where $\boldsymbol{\Sigma}_{\boldsymbol{X}}$ is a 5×5 covariance matrix and the dynamic matrix $\boldsymbol{B}$ is defined as:

$$\boldsymbol{B} = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (9)$$

In order to track and recognize the athlete's actions simultaneous, we perform the following three procedures at time t:

1. **Tracking**: Under the assumption that the the appearance of the athlete changes smoothly, we use the template at time $t-1$ to update the current state of the tracker using particle filtering.
2. **Action Recognition**: To estimate the action α_t, we perform Maximum Likelihood Estimation (MLE) based on the previous T observations using the Hidden Markov Model classifier [20, 23].
3. **Template Updating**: Having the optimal action, we update the current template E_t based on the most probable sequence in the Hidden Markov Model [20, 23].

By repeatedly perform these three procedures at time t, the system can approximately update the athlete's position, velocity, size, and determine the athlete's action in practice though we do not prove the convergence of the system.

4.1 Tracking

The posterior distribution $p(\boldsymbol{X}_t|\boldsymbol{I}_{1:t}, E_{0:t}, \alpha_{1:t})$ can be computed by the following recursion:

$$p(\boldsymbol{X}_t|\boldsymbol{I}_{1:t}, E_{0:t}, \alpha_{1:t}) \;\propto\; p(\boldsymbol{I}_t|\boldsymbol{X}_t, E_t, \alpha_t)\int p(\boldsymbol{X}_t|\boldsymbol{X}_{t-1})\cdot p(\boldsymbol{X}_{t-1}|\boldsymbol{I}_{1:(t-1)}, E_{0:(t-1)}, \alpha_{1:(t-1)})\, d\boldsymbol{X}_{t-1} \quad (10)$$

Since computing the exact posterior distribution of Eq. (10) is intractable, we use *particle filtering* [9, 18, 19] to approximate Eq. (10). Assume that we have a set of N particles $\{\boldsymbol{X}_t^{(i)}\}_{i=1\ldots N}$. In each time step, we sample candidate particles from an proposal distribution $\tilde{\boldsymbol{X}}_t^{(i)} \sim q(\boldsymbol{X}_t|\boldsymbol{X}_{0:t-1}, \boldsymbol{I}_{1:t}, E_{0:t}, \alpha_{1:t})$ (In this paper, we set $q(\boldsymbol{X}_t|\boldsymbol{X}_{0:t-1}, \boldsymbol{I}_{1:t}, E_{0:t}, \alpha_{1:t}) = p(\boldsymbol{X}_t|\boldsymbol{X}_{t-1})$, yielding the bootstrap filter), and weight these particles according to the following importance ratio

$$\omega_t^{(i)} = \omega_{t-1}^{(i)} \frac{p(\boldsymbol{I}_t|\tilde{\boldsymbol{X}}_t^{(i)}, E_t, \alpha_t)p(\tilde{\boldsymbol{X}}_t^{(i)}|\boldsymbol{X}_{t-1}^{(i)})}{q(\tilde{\boldsymbol{X}}_t^{(i)}|\boldsymbol{X}_{0:t-1}^{(i)}, \boldsymbol{I}_{1:t}, E_{0:t}, \alpha_{1:t})} \quad (11)$$

We re-sample the particles using their importance weights to generate an unweighted approximation $p(\boldsymbol{X}_t|\boldsymbol{I}_{1:t}, E_{0:t}, \alpha_{1:t})$.

The problem of Eq. (11) is that E_t and α_t are unknown. Assuming that the appearance of the athletes changes smoothly, we approximate the current template and action by the previous ones, i.e., $\tilde{E}_t = E_{t-1}, \tilde{\alpha}_t = \alpha_{t-1}$. Following [17], we define the sensor distribution $p(\boldsymbol{I}_t|\boldsymbol{X}_t, \tilde{E}_t, \tilde{\alpha}_t)$ as:

$$p(\boldsymbol{I}_t|\boldsymbol{X}_t, \tilde{E}_t = i, \tilde{\alpha}_t) \;\propto\; exp(-\lambda\, d^2(\boldsymbol{H}_t, \boldsymbol{\Pi}_i^{\tilde{\alpha}_t})) \quad (12)$$

where $\boldsymbol{H}_t$ is the HOG descriptor of the image given the state $\boldsymbol{X}_t$, $\boldsymbol{\Pi}_i^{\tilde{\alpha}_t}$ is the PCA-HOG descriptor of the template i of the action $\tilde{\alpha}_t$, and λ is a constant. The similarity measure $d(\cdot,\cdot)$ is the weighted sums of distance-from-feature-space $d_1^2(\cdot,\cdot)$ and distance-in-feature-space $d_2^2(\cdot,\cdot)$:

$$d^2(\boldsymbol{H}_t, \boldsymbol{\Pi}_i^{\tilde{\alpha}_t}) = \epsilon \cdot d_1^2(\boldsymbol{H}_t, \boldsymbol{Y}_t) + d_2^2(\boldsymbol{Y}_t, \boldsymbol{\Pi}_i^{\tilde{\alpha}_t}) \quad (13)$$

where $\boldsymbol{Y}_t$ is the PCA-HOG descriptor of the image given the state $\boldsymbol{X}_t$, and ϵ is a constant. The distance-from-feature space $d_1^2(\cdot,\cdot)$ and distance-in-feature-space $d_2^2(\cdot,\cdot)$ are given by:

$$d_1^2(\boldsymbol{H}_t, \boldsymbol{Y}_t) = (\boldsymbol{H}_t - \tilde{\boldsymbol{H}}_t)^T(\boldsymbol{H}_t - \tilde{\boldsymbol{H}}_t) \quad (14)$$

$$d_2^2(\boldsymbol{Y}_t, \boldsymbol{\Pi}_i^{\tilde{\alpha}_t}) = (\boldsymbol{Y}_t - \boldsymbol{\Pi}_i^{\tilde{\alpha}_t})^T(\boldsymbol{Y}_t - \boldsymbol{\Pi}_i^{\tilde{\alpha}_t}) \quad (15)$$

where $\tilde{\boldsymbol{H}}_t$ is the reconstructed HOG descriptor defined as $\tilde{\boldsymbol{H}}_t = \boldsymbol{\Gamma}\boldsymbol{Y}_t + \bar{\boldsymbol{H}}$. Intuitively, the distance-from-feature-space (DFFS) measures the distance between the HOG descriptor and its projection on the linear subspace, which can be interpreted as "how likely the tracking region is a player". The distance-in-feature-space (DIFS) measures the distance between the projection of the HOG descriptor on the subspace and the template, which can be interpreted as "how likely the tracking region is performing a specific pose". The combination of the DFFS and DIFS provides a robust similarity measure between the tracking region and the templates.

4.2 Action Recognition

Knowing the position and size of the athlete, the PCA-HOG descriptor $\boldsymbol{Y}_t$ of the tracking region can be computed. Then, we can estimate the action of the athlete at time t by the Maximum Likelihood Estimation (MLE) based on the previous T observations. Let $s = t - T + 1$ denote the time of first observation we use to classify the athlete's action, the likelihood of the previous T observations can be defined as:

$$p(\boldsymbol{Y}_{s:t}|\alpha_t) = \sum_{E_t} p(\boldsymbol{Y}_{s:t}, E_t|\alpha_t) \quad (16)$$

$$p(\boldsymbol{Y}_{s:t}, E_t|\alpha_t) = p(\boldsymbol{Y}_t|E_t, \alpha_t) \cdot \sum_{E_{t-1}} p(\boldsymbol{Y}_{s:(t-1)}, E_{t-1}|\alpha_t)\, p(E_t|E_{t-1}, \alpha_t) \quad (17)$$

The sensor distribution $p(\boldsymbol{Y}_t|E_t, \alpha_t)$ is defined as a Gaussian distribution:

$$p(\boldsymbol{Y}_t \mid E_t = i, \alpha_t) = \mathcal{N}(\boldsymbol{Y}_t \mid \boldsymbol{\Pi}_i^{\alpha_t}, \boldsymbol{\Sigma}_i^{\alpha_t}) \quad (18)$$

where $\boldsymbol{\Pi}_i^{\alpha_t}$ and $\boldsymbol{\Sigma}_i^{\alpha_t}$ are the mean and covariance of the PCA-HOG descriptor of the template i in action α_t.

The optimal action of the athlete at time t can be computed by

$$\alpha_t^* = \underset{\alpha_t}{argmax}\ p(\boldsymbol{Y}_{s:t} \mid \alpha_t) \quad (19)$$

Note that Eq. (16), (17), and (19) form the Hidden Markov Model classifier [20, 23], and can be efficiently computed using the forward-backward algorithm [20]. The parameters of the Hidden Markov Model can be learned using the Baum-Welch (EM) algorithm [20].

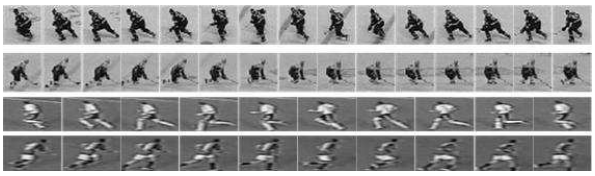


Figure 3. Examples of the training images.

4.3 Template Updating

Knowing the current action α_t^*, we estimate the most probable template sequence from time s to t given the observations represented by the PCA-HOG descriptors:

$$E_{s:t}^* = \underset{E_{s:t}}{argmax}\ p(E_{s:t} \mid \boldsymbol{Y}_{s:t}, \alpha_t^*) \quad (20)$$

We can use the Viterbi algorithm [20] to compute the most probable template sequence $E_{s:t}^*$.

To update the current template E_t, we simply set $E_t = E_t^*$. In other words, we use the last template of the most probable sequence as the template of time t.

5 Experimental Results

We tested our algorithm in soccer sequences [5] and hockey sequences [18]. For both soccer and hockey, we first collect examples of athletes performing a specific action. Fig. 3 shows some of the training images.

For the hockey sequences, we collect images of players performing 6 actions (skate left, skate right, skate in, skate out, skate left 45, and skate right 45), and transform them to the PCA-HOG descriptors. The PCA-HOG descriptors are computed with $w_g = 5$, $\sigma_g = 5$, $s_w = 5$, $s_h = 5$, $s_b = 8$, $n_f = 200$, and $n_p = 20$. In other words, each player is represented by 20-D PCA-HOG descriptor. Next, we learn the parameters of the Hidden Markov Models by the Baum-Welch (EM) algorithm. For each action, we assume that there are 10 possible templates, and we use the most recent 7 images ($T = 7$) to classify the player's action. Tracking will start with a manually initialized tracking region and with 60 particles in our experiments.

We compute the PCA-HOG descriptor with the same settings for the experiments on soccer sequences. The difference is that we classify the soccer player's actions into 8 categories (run left, run right, run left 45, run right 45, run in/out, and walk left, walk right, walk in/out) [1]. In addition, we assume that there are 10 possible templates for each action, and we use the most recent 10 images ($T = 10$) to classify the player's action.

We implement the entire system using Matlab, and the processing time is about 2 fps under a Linux machine with a

[1] The action categories are the same as Efros *et al.* [5]

2.6GHz CPU. Fig. 4 and Fig. 5 show some experimental results in the hockey sequences. We can observe that the system can track a single hockey player and recognize his actions effectively even though the player constantly changes his pose. Moreover, the tracker is also robust under significant illumination changes (flashes) because we rely on shape rather than color information. Fig. 5 also indicates that the tracker is robust under *partial* occlusions. A possible explanation is that we project the HOG descriptor to its principal subspace and thus the influences of other players in the tracking region can be alleviated. Unfortunately, when two players cross over, further techniques such as [2] are needed to tackle the occlusion problem.

The experimental results in soccer sequences are shown in Fig. 6 and 7. Fig. 6 displays the results of a player running across a line. This example shows that even the background contains strong edges, the tracker can still accurately track the player due to the robustness of the PCA-HOG descriptor. Fig. 7 presents the results of a player running with a ball and constantly changing actions. The action recognition results show that using shape (HOG/SIFT representation) alone also works for the soccer sequence even though the tracking region is small and of low resolution.

6 Conclusion

This paper presents a system that tightly couples tracking and action recognition into an integrated system. In addition, the PCA-HOG descriptor representation not only provides robust and distinctive input features, but also implicitly centers the figure in the tracking region. Therefore, no further stabilization techniques need to be used. Experimental results in hockey and soccer sequences show that this system can track a single athlete and recognize his/her actions effectively.

In the future, we plan to use the action as a random variable instead of a deterministic parameter, and introduce dependencies between the current action and the previous action, and between the action and the state of the tracker. The resulting Dynamical Bayesian Network (DBN) will be a hybrid hierarchical Hidden Markov Model (HHMM) containing three interacting Markov processes. Furthermore, we also plan to extend our work to a multi-target tracking system, where the actions of the athletes can be used as an addition cue when players occlude each other.

7 Acknowledgments

This work has been supported by grants from NSERC, the GEOIDE Network of Centres of Excellence and Honeywell Video Systems.

References

[1] M. Black and A. Jepson. EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation. *IJCV*, 26(1):63–84, 1998.

[2] Y. Cai, N. de Freitas, and J. Little. Robust visual tracking for multiple targets. In *ECCV, to appear*, 2006.

[3] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-Based Object Tracking. *PAMI*, 25(5):564–575, 2003.

[4] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In *CVPR*, pages 886–893, 2005.

[5] A. Efros, C. Breg, G. Mori, and J. Malik. Recognizing Action at a Distance. In *ICCV*, pages 726–733, 2003.

[6] D. Gavrila. The Visual Analysis of Human Movement: A Survey. *CVIU*, 73(1):82–98, 1999.

[7] J. Giebel, D. Gavrila, and C. Schnörr. A Bayesian Framework for Multi-cue 3D Object Tracking. In *ECCV*, pages 241–252, 2004.

[8] J. Ho, K. Lee, M. Yang, and D. Kriegman. Visual Tracking Using Learned Linear Subspaces. In *CVPR*, volume 1, pages 782–789, 2004.

[9] M. Isard and A. Blake. CONDENSATION–Conditional Density Propagation for Visual Tracking. *IJCV*, 29(1):5–28, 1998.

[10] A. Jepson, D. Fleet, and T. El-Maraghi. Robust online appearance models for visual tracking. *PAMI*, 25(10):1296–1311, 2003.

[11] K. Lee, J. Ho, M. Yang, and D. Kriegman. Visual tracking and recognition using probabilistic appearance manifolds. *CVIU*, 99:303–331, 2005.

[12] J. Lim, D. Ross, R. Lin, and M. Yang. Incremental Learning for Visual Tracking. In *NIPS*, 2004.

[13] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, 60(2):91–110, 2004.

[14] W. Lu and J. Little. Tracking and recognizing actions at a distance. In *CVBASE, to appear*, 2006.

[15] K. Mikolajczyk, B. Leibe, and B. Schiele. Local features for object class recognition. In *ICCV*, pages 1792–1799, 2005.

[16] K. Mikolajczyk and C. Schmid. A Performance Evaluation of Local Descriptors. *PAMI*, 27(10):1615–1630, 2005.

[17] B. Moghaddam and A. Pentland. Probabilistic visual learning for object recognition. *PAMI*, 19(7):696–710, 1997.

[18] K. Okuma, A. Taleghani, N. de Freitas, J. Little, and D. Lowe. A Boosted Particle Filter: Multitarget Detection and Tracking. In *ECCV*, pages 28–39, 2004.

[19] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-Based Probabilistic Tracking. In *ECCV*, pages 661–675, 2002.

[20] L. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proc. of the IEEE*, 77(2):257–286, 1989.

[21] X. Wu. Templated-based Action Recognition: Classifying Hockey Players' Movement. Master's thesis, The University of British Columbia, 2005.

[22] Y. Wu and T. Huang. Robust visual tracking by integrating multiple cues based on co-inference learning. *IJCV*, 58(1):55–71, 2004.

[23] J. Yamato, J. Ohya, and K. Ishii. Recognizing Human Action in Time-Sequential Images using Hidden Markov Model. In *CVPR*, pages 379–385, 1992.

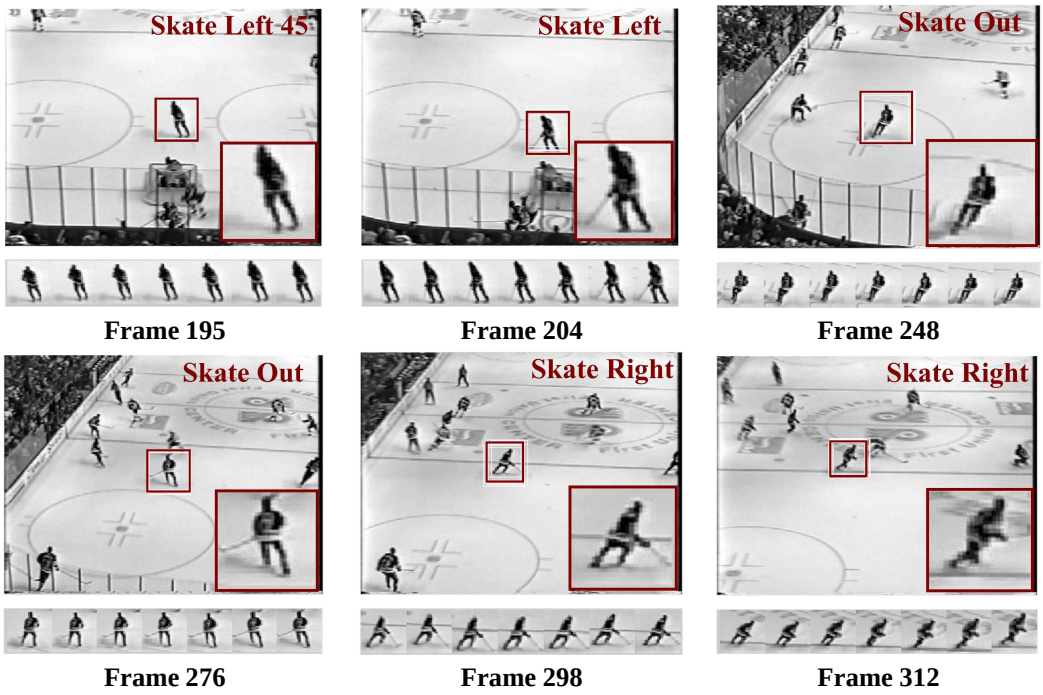


Figure 4. (Hockey sequence 1) Player changing actions: The upper parts of the images are the video frames and the lower parts are the most recent observations used to classify the player's actions.

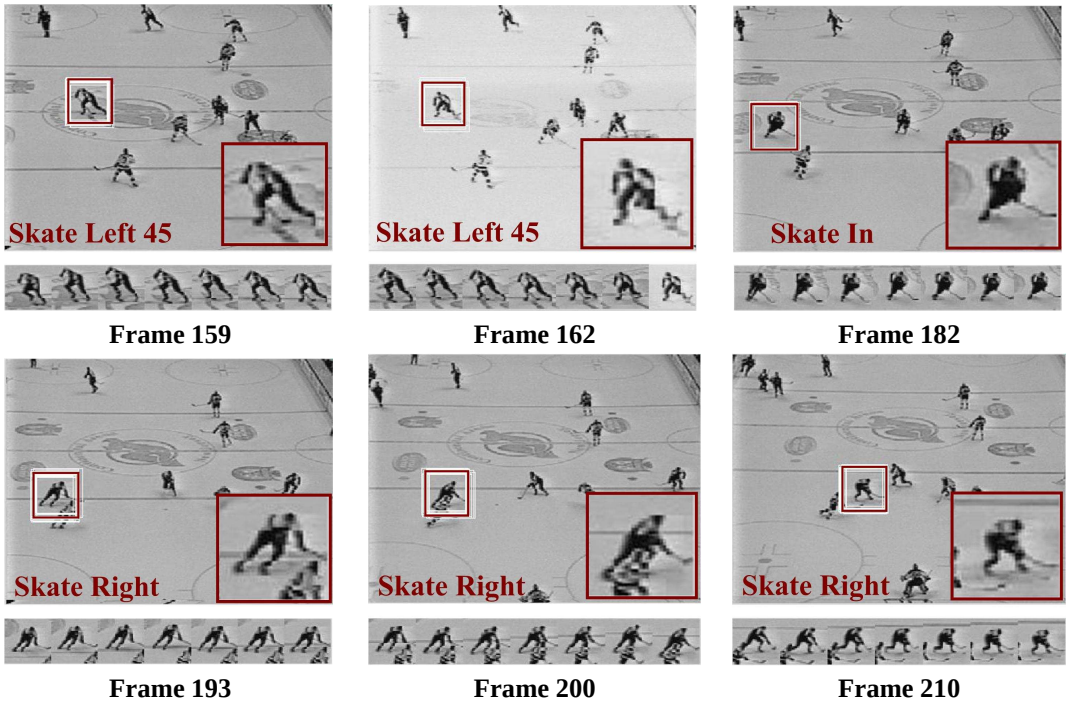


Figure 5. (Hockey sequence 2) Flash and partial occlusion: The upper parts of the images are the video frames and the lower parts are the most recent observations used to classify the player's actions.

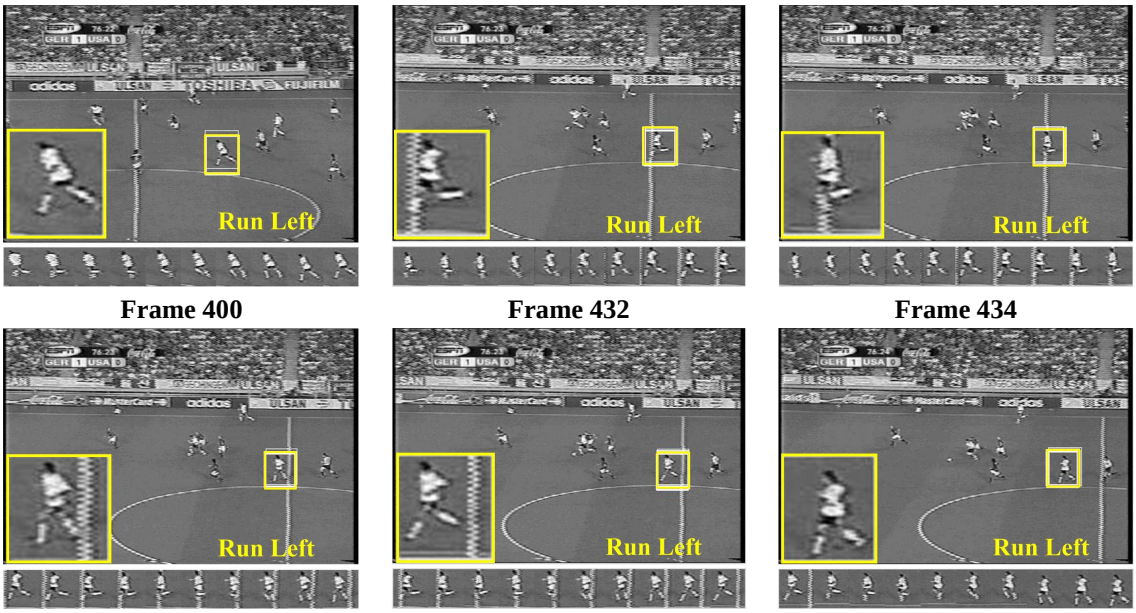


Figure 6. (Soccer sequence 1) Player running across a line: The upper parts of the images are the video frames and the lower parts are the most recent observations used to classify the player's actions.

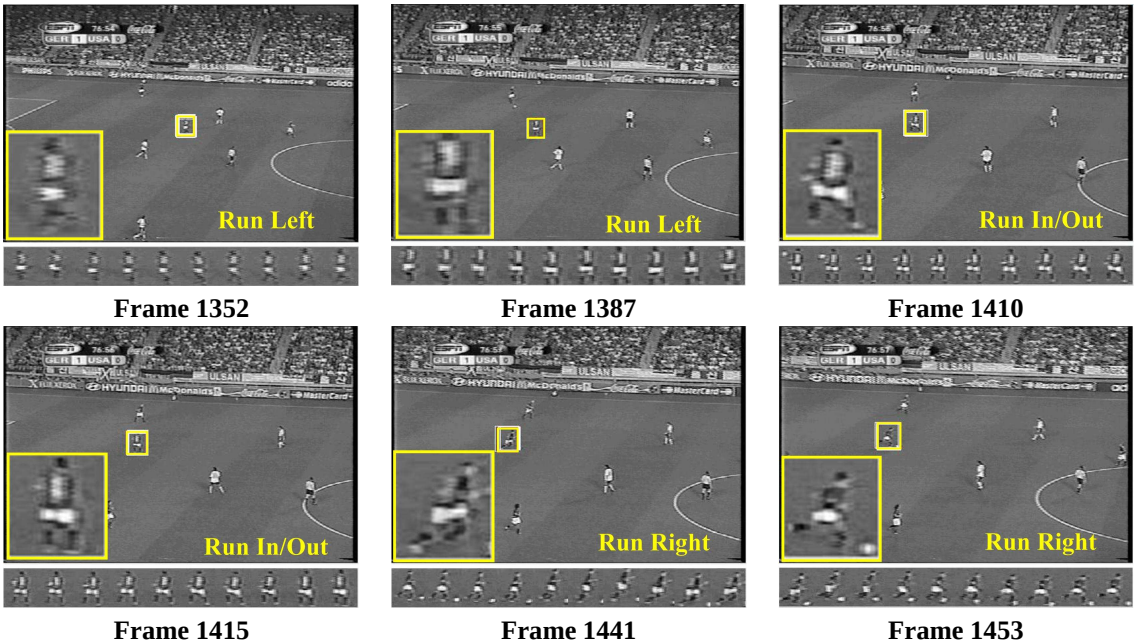


Figure 7. (Soccer sequence 2) Player running with a ball: The upper parts of the images are the video frames and the lower parts are the most recent observations used to classify the player's actions.