Online Motion Classification using Support Vector Machines

Dongwei Cao, Osama T Masoud, Daniel Boley, Nikolaos Papanikolopoulos Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455
Email: {dcao, masoud, boley, npapas}@cs.umn.edu

***Abstract*— We propose a motion recognition strategy that represents a videoclip as a set of filtered images, each of which encodes a short period of motion history. Given a set of videoclips whose motion types are known, a filtered image classifier is built using support vector machines. In offline classification, the label of a test videoclip is obtained by applying majority voting over its filtered images. In online classification, the most probable type of action at an instance is determined by applying the majority voting over the most recent filtered images, which are within a sliding window. The effectiveness of this strategy was demonstrated on real datasets where the videoclips were recorded using a fixed camera whose optical axis is perpendicular to the person's trajectory. In offline recognition, the proposed strategy outperforms a Principal Component Analysis based recognition algorithm. In online recognition, the proposed strategy can not only classify motions correctly and identify the transition between different types of motions, but also identify the existence of an unknown motion type. This latter capability and the efficiency of the proposed strategy make it possible to create a real-time motion recognition system that can not only make classifications in real-time, but also learn new types of actions and recognize them in the future.**

***Indexed Terms*— human motion recognition, recursive filtering, support vector machines.**

I. Introduction

The purpose of human motion recognition is to assign a specific label to a motion. Recognition can be offline or online based on the requirements of the specific application. In offline recognition, an entire videoclip is available and it is desirable to identify the type of motion recorded in the videoclip. In online recognition, the entire videoclip is typically not available, and we want to identify the most probable motion type at each instance. In this paper, we propose a strategy that is applicable to both offline and online recognition.

In general, there are two tightly related steps in building a motion recognition system, i.e., extracting motion features and training a classifier using these features. The majority of relevant work in motion recognition focuses on motion feature selection, including extracting features from 2-D tracking data ([1], [2], [3], [4], [5], [6], [7]) or 3-D tracking information ([8], [9]), or extracting motion information directly from images ([10], [11], [12], [13]). Given a set of features that is believed be able to characterize the motion of interest, most recognition algorithms are based on either template matching ([13], [12]) or state-space matching which usually uses Hidden Markov Model (HMM) [11]. Other recognition algorithms employ neural networks [2]. The performance of these recognition algorithms, especially the ones based on template matching, is highly dependent on the quality of the extracted motion features, which in general should reflect the nonlinear nature of human motions. A comprehensive review on human motion analysis can be found, for example, in [14].

In this paper, motion information is encoded through *recursive filtering* and *frame grouping*. A filtered image is constructed for every frame of a videoclip using the *recursive filtering*. It encodes the spatial layout of the scene in the current frame, the temporal relation between consecutive frames, and the speed of the motion within a short period of time. The recursive filtering was proposed by Osama and Papanikolopoulos [15], and it is conceptually similar to the Motion History Image (MHI) proposed in [12]. The idea of *frame grouping* is to classify every filtered image of a videoclip separately and to use the resulting labels to determine the motion type of the videoclip through majority voting. The filtered image classifier is built using support vector machines. The reason for choosing support vector machines is that, through the implicit mapping induced by a Mercer kernel [16], some nonlinear features are extracted implicitly and a nonlinear classifier can be constructed. In addition, support vector machines have been proved to be very effective on similar classification tasks, such as handwritten digits recognition [17].

The rest of this paper is organized as follows: after a brief introduction of support vector machines in Section II, Section III describes the proposed motion representation strategy. The experimental results are summarized in Section IV. Section V concludes the paper with future research topics.

II. Support Vector Machines

In a two-class classification problem, given a training dataset $\mathcal{D}^k$ of size n^k

$$\mathcal{D}^k = \left\{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in \mathcal{R}^N,\ y_i \in \{1, -1\}\right\} \tag{1}$$

where N is the dimension of $\mathbf{x}_i$, $i = 1, 2, \cdots, n^k$, and y_i is the label of $\mathbf{x}_i$, the support vector classifier $f^k(\mathbf{x})$ is defined as [16]

$$f^k(\mathbf{x}) = sign\left(d^k(\mathbf{x})\right) = \begin{cases} 1 : d^k(\mathbf{x}) \geq 0 \\ -1 : d^k(\mathbf{x}) < 0 \end{cases} . \tag{2}$$

The term $d^k(\cdot)$ is called the *functional margin* and it can be obtained through

$$d^k(\mathbf{x}) = \sum_{x_i \in \mathcal{D}_{SV}} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + \theta, \tag{3}$$

where $K(\cdot,\cdot)$ is a Mercer kernel [16]. $\mathcal{D}_{SV}$ is the set of *support vectors*, which are the training data having nonzero α's. The α's are obtained by

$$\text{Maximize : } W(\alpha) = \boldsymbol{\alpha}^T \mathbf{1} - \frac{1}{2}\boldsymbol{\alpha}^T \mathbf{H} \boldsymbol{\alpha} \tag{4a}$$

$$\text{Subject to : } 0 \leq \boldsymbol{\alpha} \leq C \text{ and } \boldsymbol{\alpha}^T \mathbf{y} = 0, \tag{4b}$$

where $\mathbf{1}$ is a vector of ones, C is a positive parameter that controls the trade-off between accuracy and smoothness of the classifier and needs to be specified *a priori*, and $\mathbf{H}$ is the Gram matrix with component $H_{ij} = y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$. Detailed explanations on how to solve problem (4) can be found, for example, in [16] and [18]. We have two remarks about support vector machines:

- **Remark 1:** In order to build a support vector classifier, we need to specify the kernel $K(\cdot,\cdot)$ and the penalizing coefficient C. By choosing an appropriate K, we implicitly map the vector $\mathbf{x}$ into some Hilbert space $\mathcal{H}$ using a (usually nonlinear) mapping Φ that satisfies
$$K(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle_{\mathcal{H}}\,, \tag{5}$$
where $\langle \cdot,\cdot \rangle_{\mathcal{H}}$ is the dot product in $\mathcal{H}$. With reference to (3), this means that the support vector classifier is a linear classifier in $\mathcal{H}$. It should be noted that no dimensionality reduction is used here. In fact, the dimensionality is increased since the dimension of $\mathcal{H}$ is usually much larger than N, and could be infinity for certain K [16].
- **Remark 2:** Assume a test datum $\mathbf{x}$ is classified to the label y with functional margin $d^k(\mathbf{x})$. For $yd^k(\mathbf{x}) \leq 1$, $\mathbf{x}$ would very *likely* become a support vector if it was included in the training dataset; on the other hand, for $yd^k(\mathbf{x}) > 1$, $\mathbf{x}$ would very *unlikely* become a support vector if it was included in the training dataset. Since it is the set of support vectors that determines the decision boundary, we can use $yd^k(\mathbf{x})$ as a measure of confidence for the classification, and a natural choice for the rejection threshold of this confidence is 1.

To discriminate more than two candidate motion types, we use a strategy called *one-versus-the-rest* [18]. Assuming that there are L candidate motion types, the idea of *one-versus-the-rest* is to train L support vector classifiers, and the k-th ($k = 1, \ldots, L$) support vector classifier $f^k(\mathbf{x})$ discriminates motions of type k from all the other types of motions. The label y of a test datum $\mathbf{x}$ is then obtained by

$$y = \underset{k \in \{1,2,\cdots,L\}}{\operatorname{argmax}}\; d^k(\mathbf{x}), \tag{6}$$

where $d^k(\mathbf{x})$ is the functional margin given by the k-th support vector classifier (3).

III. Motion Representation

The proposed motion representation strategy has two parts: *recursive filtering*, which encodes the temporal relationship among the frames of a videoclip and the speed of the motion within a short period of time, and *frame grouping*, which encodes the fact that it is the ensemble of all frames of a videoclip that encodes the videoclip's motion information, especially the type of the motion. The details of these two parts are described below.

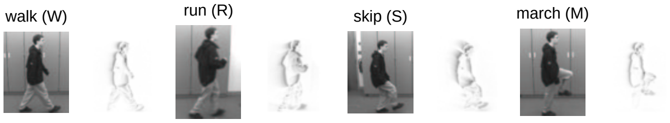

Fig. 1. Raw images and filtered images of four types of actions.

A. Recursive Filtering

Without losing generality, we take a videoclip $\mathcal{X}_i$ as an example and assume that there are n_i frames in $\mathcal{X}_i$. The idea of *recursive filtering* is to represent the motion by its "recency". Let I_t be the frame at time t, then the filtered image F_t at time t is defined as [15]

$$F_t = |I_t - M_t| \tag{7a}$$

$$M_t = (1-\beta)M_{t-1} + \beta I_{t-1} \tag{7b}$$

$$M_0 = I_0 = \text{Background}, \tag{7c}$$

where $t = 1, 2, \cdots, n_i$ and $|\cdot|$ denotes the absolute value. The coefficient β is a pre-specified constant. If $\beta = 0$, the filtered image F_t will be the foreground and, if $\beta = 1$, F_t will be equivalent to image differencing. In the current study, a $\beta = 0.5$ was used, which was suggested in [15]. Fig. 1 shows the representative frames for four types of actions and their filtered images when $\beta = 0.5$. It can be seen from Fig. 1 that the temporal relationship among consecutive frames is encoded in the filtered images and, from the "tail" of the filtered images, we can easily tell the direction of motion, the recent trajectories of the parts of a person such as legs, and even the relative speed of different parts of the person's body. The filtered image F_t given by (7) is further thresholded to remove the noise and down-sampled to a lower resolution having width w and height h.

B. Frame Grouping

A filtered image is treated as a real valued matrix with w rows and h columns and it can be represented by a column vector of length $w \times h$ by concatenating the columns of the matrix. By *frame grouping*, we mean that the videoclip $\mathcal{X}_i$ having n_i frames is represented by n_i points in $\mathcal{R}^{w\times h}$, all of which have the same label as $\mathcal{X}_i$. Thus, assuming that there are n training videoclips and L candidate motion types, the training dataset $\mathcal{D}$ can be written as

$$\mathcal{D} = \bigcup_{i=1}^{n} \bigcup_{j=1}^{n_i} \left\{ (\mathbf{x}_{ij}, y_{ij}) | \mathbf{x}_{ij} \in \mathcal{R}^{w\times h}, y_{ij} = y_i \right\}, \tag{8}$$

where n_i is the number of frames in videoclip $\mathcal{X}_i$, y_{ij} is the label of $\mathbf{x}_{ij}$, y_i is the label of $\mathcal{X}_i$, and $y_i \in \{1,\ 2,\ \cdots,\ L\}$. Then, a filtered image classifier having L support vector classifiers can be built using $\mathcal{D}$.

In offline recognition, a test videoclip $\mathcal{X}_0$ having n_0 frames is represented by a set $\mathcal{D}_0$ of n_0 filtered images, that is,

$$\mathcal{D}_0 = \left\{ \mathbf{x}_{0j} | \mathbf{x}_{0j} \in \mathcal{R}^{w\times h}, \quad j = 1, 2, \cdots, n_0 \right\}. \tag{9}$$

The label y_0 of $\mathcal{X}_0$ is obtained using the *majority voting*, i.e.,

$$y_0 = \underset{k \in \{1,2,\cdots,L\}}{\operatorname{argmax}}\; n_{0,k}, \tag{10}$$

where $n_{0,k}$ is the number of filtered images in $\mathcal{D}_0$ that are assigned the label k.

There are two desired properties for an online motion recognition system:

- It should be able to identify the most probable type of motions at each instance based only on the video presented so far and, when the type of motion changes, it should be able to detect such change in a timely manner.
- Any motion type that has not been used to build the recognition system is unknown to the system and, if an unknown type of motion is happening, the system should be able to detect it.

In order to determine the most probable type of motion at time t, the majority voting defined in (10) is taken over the set of the most recent filtered images, i.e.,

$$\{F_{t-b+1}, F_{t-b+2}, \cdots, F_{t-1}, F_t\}, \tag{11}$$

where b is the width of the sliding window. The width b of the sliding window controls the trade-off between the sensitivity to motion changes and the robustness to environmental noise. In order to detect motions of an unknown type, with reference to Remark 2 in Section II, we define a confidence CF_t for the classification at time t and a threshold T_{CF} for this confidence as follows

$$CF_t = \frac{1}{b}\left(\sum_{y_j=y} |d^{y_j}(\mathbf{x}_j)| - \sum_{y_j\neq y} |d^{y_j}(\mathbf{x}_j)|\right) \tag{12}$$

$$T_{CF} = 1, \tag{13}$$

where y is the label given by the majority voting, $d^{y_j}(\mathbf{x}_j)$ is the functional margin of the j-th filtered image within the sliding window given by the y_j-th support vector classifier (3) (i.e., the support vector classifier corresponding to the winning class in (6)), and $j = t-b+1, t-b+2, \cdots, t-1, t$ (c.f. equation (11)). This definition can be seen as an average functional margin penalized by the "misclassified" filtered images. The classification at time t will be rejected if $CF_t < T_{CF}$. If, in a video stream, many consecutive classifications are rejected, we can conclude that an unknown type of motion has happened, provided there is no significant noise in the video stream.

IV. Experimental Results

In this section, we will demonstrate the effectiveness of the proposed motion recognition strategy on real datasets. There are 8 types of actions, namely walk (W), run (R), skip (S), march (M), line walk (LW), hop (H), side walk (SW) and side skip (SS). The datasets are described in Table I. Dataset A has 232 videoclips that were obtained by letting 29 subjects perform each of the 8 actions once. This dataset was used for offline recognition. Datasets B_0 through B_3 were used for online recognition. Dataset B_0 is a subset of dataset A, and it consists of walking and running actions performed by subjects 2 through 29. Dataset B_1 has three videoclips that were obtained by letting subject 30 perform walk, run, and march once. Dataset B_2 has two artificial videoclips that were obtained by manually concatenating walking and running videoclips performed by subject 1, which are also in dataset A. The dataset B_3 has three videoclips that were obtained by letting subject 30 perform actions having transitions between walk, run, and march.

TABLE I
Description of datasets

Dataset	Subjects	Actions
A	$1 \sim 29$	All 8 possible types
B_0	$2 \sim 29$	Run and walk
B_1	30	March, walk, and run
B_2	1	*Artificial* combination of walk and run
B_3	30	*Real* combination of walk, run, and march

A. Offline Recognition

The filtered image classifier was built using

$$K(\mathbf{x}_{ij}, \mathbf{x}_{pq}) = \left(\mathbf{x}_{ij}^T \mathbf{x}_{pq}\right)^{15} \text{ and } C = 10. \tag{14}$$

The resulting motion recognition system was evaluated using a modified cross validation called "Leave One Person Out Cross Validation" (LOOCV-Person). In each fold of the LOOCV-Person, all videoclips performed by one subject were kept as test videoclips and the classifier was built using the other videoclips. The results of the 29-fold LOOCV-Person on dataset A are summarized in Table II, where the action error rate ER_{action} is defined as

$$ER_{\text{action}} = \frac{NV_{\text{misclassified}}}{NV_{\text{test}}}, \tag{15}$$

where $NV_{\text{misclassified}}$ is the number of misclassified test videoclips and NV_{test} is the number of test videoclips. In addition to the action error rate, Table II also shows the image error rate ER_{image}, which is defined as

$$ER_{\text{image}} = \frac{NF_{\text{misclassified}}}{NF_{\text{test}}}, \tag{16}$$

where $NF_{\text{misclassified}}$ is the number of misclassified filtered images in test videoclips and NF_{test} is the number of filtered images in test videoclips. It can be seen from Table II that the action error rate is smaller than the image error rate in almost all folds. In the ideal case, we can correctly classify an action using majority voting as long as more than half of its filtered images are classified correctly. This observation is especially important since it can make the system robust to local similarities between different types of actions (for example, walking and running have similar gestures when both feet touch the ground) and to the noise in the videoclips. As a comparison, the same 29-fold LOOCV-Person was performed using the strategy proposed in [15], which is a nearest neighbor method using PCA and Hausdorff distance, and the resulting average action error rate over 29 cross validation was 7.8%. For completeness, the confusion matrix for one cross validation is shown in Table III.

TABLE II

ERROR RATES OF THE MOTION RECOGNITION SYSTEM (29-FOLD LOOCV-PERSON)

Fold	1	2	3	4	5	6	7	8
ER_{image}	14.6	9.2	14.3	7.7	17.6	10.0	13.3	4.6
ER_{action}	0	0	12.5	0	0	0	0	0
Fold	9	10	11	12	13	14	15	16
ER_{image}	19.1	10.1	21.4	27.0	14.0	9.0	17.6	8.6
ER_{action}	12.5	0	0	12.5	0	0	0	0
Fold	17	18	19	20	21	22	23	24
ER_{image}	13.4	4.8	32.4	23.3	9.3	5.1	26.6	9.5
ER_{action}	0	0	25	25	0	0	0	0
Fold	25	26	27	28	29	**Average**		
ER_{image}	24.9	13.1	14.9	30.5	16.2	**15.2**		
ER_{action}	12.5	0	0	25	0	**4.3**		

TABLE III

CONFUSION MATRIX FOR ONE CROSS VALIDATION (1 OUT OF 8 ACTIONS WAS MISCLASSIFIED). AN EMPTY ENTRY MEANS ZERO.

		Predicted label							
		W	R	S	M	LW	H	SW	SS
True label	W			1					
	R		1						
	S			1					
	M				1				
	LW					1			
	H						1		
	SW							1	
	SS								1

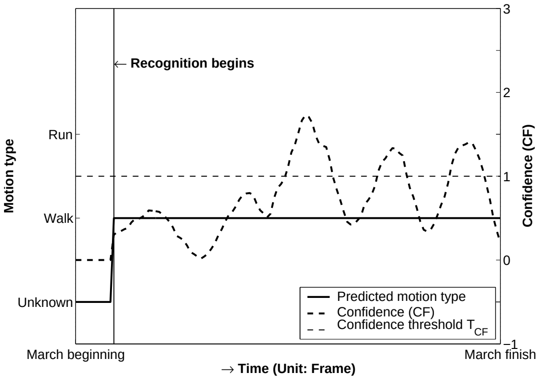

Fig. 2. Online classification for a videoclip of *pure marching*.

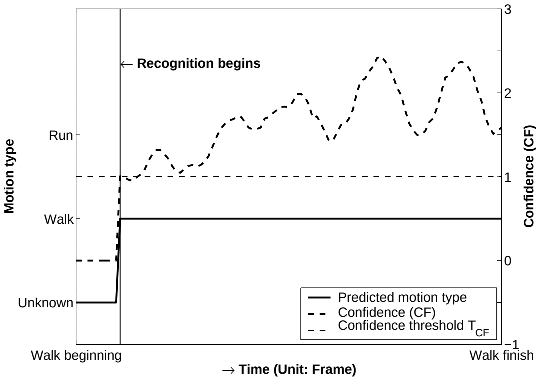

Fig. 3. Online classification for a videoclip of *pure walking*.

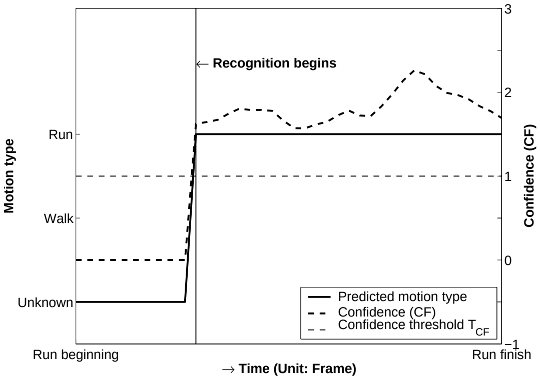

Fig. 4. Online classification for a videoclip of *pure running*.

B. Online Recognition

Assuming that the only known types of actions are walking and running, we built a motion recognition system using dataset B_0. This system can recognize walk and run, and will treat all the other types of actions as unknown. The unknown type of action examined in this experiment is marching. The filtered image classifier was built using

$$K(\mathbf{x}_{ij}, \mathbf{x}_{pq}) = \left(\mathbf{x}_{ij}^T \mathbf{x}_{pq}\right)^3 \text{ and } C = 0.01. \qquad (17)$$

The average error rates over the 28-fold LOOCV-Person on the dataset B_0 are

$$ER_{\text{image}} = 8.31\% \text{ and } ER_{\text{action}} = 3.57\%. \qquad (18)$$

Fig. 3 through Fig. 10 demonstrate the online classification results. Unless noted otherwise, we have in these figures: (i) the size of the sliding window $b = 12$ and the recognition begins at the 12-th frame; (ii) the labels on the horizontal axis indicate the true motion type; (iii) with reference to the left Y-axis, the **thick solid** line shows the classifications as a function of time; (iv) with reference to the right Y-axis, the **thick dashed** line shows the confidence for the classification as a function of time, and the *thin horizontal dashed* line shows the confidence threshold defined in (13).

First, we examine the dataset B_1 whose videoclips have only one type of action. With reference to Figs. 2, 3, and 4, we can see that walking and running were classified correctly with confidence $CF > 1$ at almost every instance, while marching was classified with $CF \leq 1$ at almost every instance. This means that an action of unknown type has been detected and, in this case, it is marching.

Second, we examine the dataset B_2 whose videoclips have an artificial transition between walking and running. In these videoclips, the time when the transition occurred is known exactly. With reference to Fig. 5, 6, and 7, we have two observations.

- Transitions between walking and running are identified in a timely manner. The delay of such detection can be reduced by using a smaller sliding window, however,

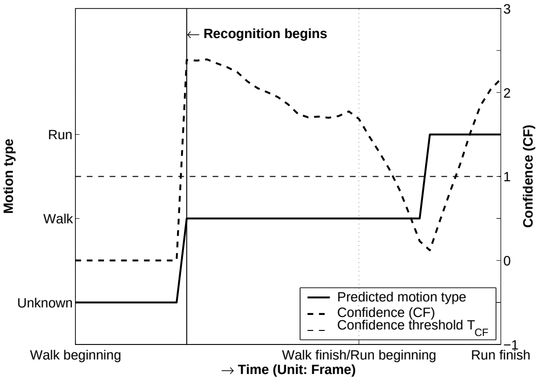

Fig. 5. Online classification for a videoclip having *an artificial transition from walking to running.*

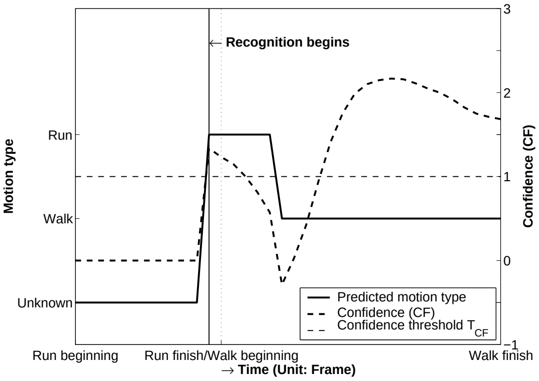

Fig. 6. Online classification for a videoclip having *an artificial transition from running to running.*

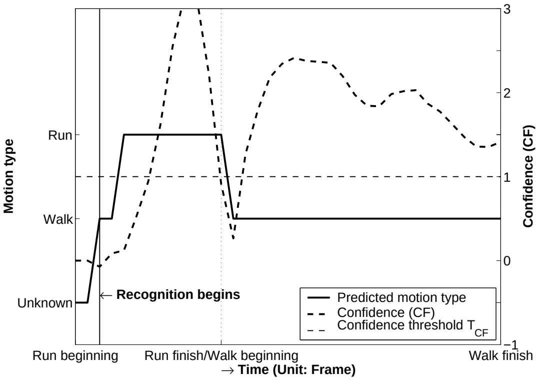

Fig. 7. Online classification for a videoclip having *an artificial transition from running to walking.* The videoclip used here is the same as that used in Fig. 6 and the size of the sliding window is 3, which means that the recognition began at the 3-rd frame.

this would make the classification less robust. Instead of using a sliding window of size 12, Fig. 7 used a sliding window of size 3. Compared to Fig. 6, the transition can be detected with less delay in Fig. 7 but at the price of increased misclassifications in the beginning.

- Based on the confidence measure defined in (12), we can see that almost all classifications are accepted, except the ones around the transition point, which is expected.

Third, we examine the dataset B_3 whose videoclips have a real transition between two types of actions. It should be pointed out that a real transition happens gradually and the exact time when the transition occurs is not available. Thus, a transition period was manually labeled. For example, in Fig. 8 that contains a transition from walking to marching, the time when the subject stops walking (“Walk finish”) and the time when it is obvious that the subject was marching (“March beginning”) are both labeled.

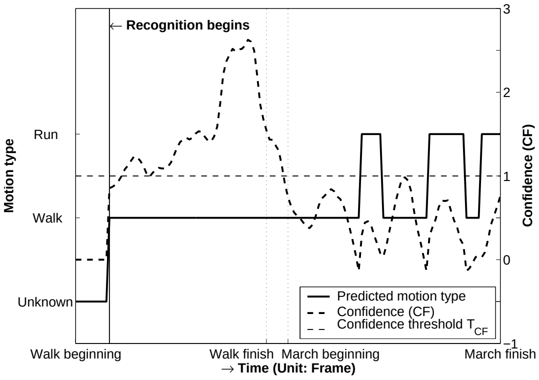

Fig. 8. Online classification for a videoclip having *a real transition from walking to marching.*

In Fig. 8, from “Walk beginning” to “Walk finish”, almost all the classifications are correct and the corresponding confidences are larger than 1. From “Walk finish” to “March beginning”, the confidence drops significantly. After “March beginning”, all classifications’ confidences are smaller than 1 and, based on this, we can tell the existence of an unknown action type.

In Fig. 9, from “March beginning” to “March finish”, the confidences of all classifications are smaller than 1, which indicates the existence of an unknown action type. From “March finish” to “Run beginning”, the confidence increases significantly. After “Run beginning”, all classifications are correct and their respective confidences are larger than 1.

In Fig. 10, from “Run beginning” to “Run finish”, all classifications are correct and the corresponding confidences are larger than 1. However, after “Walk beginning”, all classifications are wrong and, except at the end, the corresponding confidences CF are larger than 1. The explanation for this result is that it takes a long time for a subject who is running to slow down and begin to walk. In addition, as illustrated on dataset B_2, using a sliding window of size 12 has a large delay in detecting the transition between different action types. It is expected that, for a longer videoclip in which a subject slows down from running gradually and walk for several steps, the classifier would be able to recognize the walking with $CF > 1$. However, due to the limited size of the room where we recorded the video, the subject did not have enough room to perform such a long action.

V. Discussion and Future Work

A novel human motion recognition strategy is proposed in this paper, and it solves the problem of motion recognition through classifying filtered images using support vector

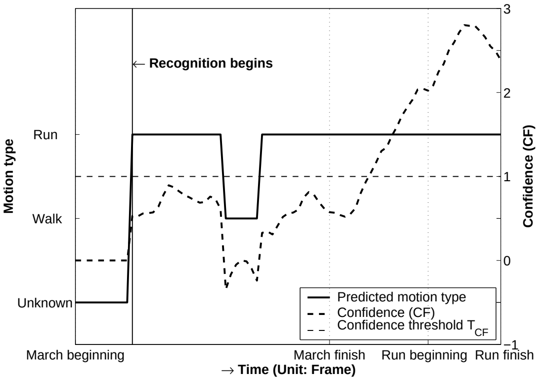

Fig. 9. Online classification for a videoclip having *a real transition from marching to running.*

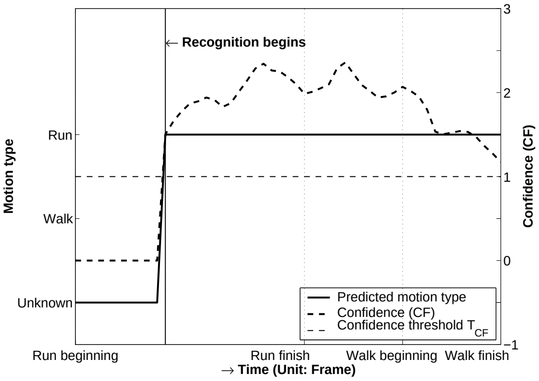

Fig. 10. Online classification for a videoclip having *a real transition from running to walking.*

machines. The effectiveness of this strategy is demonstrated using real datasets for both the offline recognition and the online recognition. The current research can be extended in the following directions.

First, it is assumed in the current study that the subject performing the motion is located in the center of the filtered image and the trajectory of the subject is perpendicular to the optical axis of the camera. However, these assumptions may not be true in practice, and we need a system that works well after removing these restrictions. This direction can be exploited, for example, by combining techniques like jittered support vectors [18] or virtual support vectors [18] with the proposed strategy.

The ability of real-time recognition is critical in many applications, thus the second direction is to implement a *real-time* recognition system. This goal is achievable because, according to our experiments, the average time for deciding the type of motion at an instance is 0.0285 seconds using a Pentium 4 2.8 GHz processor, which is shorter than the time for recording a frame since the frame rate is 30 frames per second.

ACKNOWLEDGMENTS

This work has been supported in part by NSF grants IIS-0219863 and IIS-0208621, the Minnesota Department of Transportation, and the ITS Institute at the University of Minnesota. The authors would like to thank anonymous reviewers for their valuable comments.

REFERENCES

[1] N. H. Goddard, "The perception of articulated motion: Recognizing moving light displays," Ph.D. dissertation, University of Rochester, 1992.
[2] Y. Guo, G. Xue, and S. Tsuji, "Understanding human motion patterns," in *Proceedings of the 12th International Conference on Pattern Recognition*, 1994, pp. 325–329.
[3] W. H. Dittrich, "Action categories and the perception of biological motion," *Perception*, vol. 22, pp. 15–22, 1993.
[4] Y. Yacoob and M. J. Black, "Parameterized modeling and recognition of activities," *Computer Vision and Image Understanding: CVIU*, vol. 73, no. 2, pp. 232–247, 1999.
[5] C. Bregler, "Learning and recognizing human dynamics in video sequences," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1997.
[6] K. Rangarajan, W. Allen, and M. Shah, "Matching motion trajectories using scale space," *Pattern Recognition*, vol. 26, no. 4, pp. 595–610, 1993.
[7] V. Pavlovic and J. Rehg, "Impact of dynamic model learning on classification of human motion," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2000, pp. 788–795.
[8] L. W. Campbell and A. F. Bobick, "Recognition of human body motion using phase space constraints," in *Proceedings of the International Conference on Computer Vision*, 1995, pp. 624–630.
[9] D. M. Gavrila and L. S. Davis, "3-D model-based tracking of humans in action: A multiview approach," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1996, pp. 73–80.
[10] N. Krahnstöver, M. Yeasin, and R. Sharma, "Towards a unified framework for tracking and analysis of human motion," in *Proceedings of the IEEE Workshop on Detection and Recognition of Events in Video*, 2001, pp. 47–54.
[11] J. Yamato, J. Ohya, and K. Ishii, "Recognizing human action in time-sequential images using a hidden markov model," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1992.
[12] J. W. Davis and A. F. Bobick, "The representation and recognition of human movement using temporal templates," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1997, pp. 928–934.
[13] R. Polana and R. C. Nelson, "Low level recognition of human motionn," in *Proceedings of the IEEE Workshop on Non-rigid Motion*, 1994, pp. 77–82.
[14] J. K. Aggarwal and Q. Cai, "Human motion analysis: A review," *Computer Vision and Image Understanding*, vol. 73, no. 3, pp. 428–440, March 1999.
[15] O. Masoud and N. Papanikolopoulos, "A method for human action recognition," *Image and Vision Computing*, vol. 21, pp. 729–743, 2003.
[16] V. Vapnik, *Statistical Learning Theory*. NY: Wiley, 1998.
[17] B. Schölkopf, P. Simard, A. Smola, and V. Vapnik, "Prior knowledge in support vector kernels," in *Advances in Neural Information Processing Systems*, M. I. Jordan, M. J. Kearns, and S. A. Solla, Eds., vol. 10, 1998, pp. 640–646.
[18] B. Schölkopf and A. J. Smola, *Learning with Kernels, Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2002.