Event Detection Using Local Binary Pattern Based Dynamic Textures

Yunqian Ma
Honeywell International Inc.
1985 Douglas Drive North
Golden Valley, MN 55442, USA
yunqian.ma@honeywell.com

Petr Cisar
Honeywell Prague Labs
V Parku 2326/18, 14800,
Prague, Czech Republic
petr.cisar@honeywell.com

Abstract

Detecting suspicious events from video surveillance cameras has been an important task recently. Many trajectory based descriptors were developed, such as to detect people running or moving in opposite direction. However, these trajectory based descriptors are not working well in the crowd environments like airports, rail stations, because those descriptors assume perfect motion/object segmentation. In this paper, we present an event detection method using dynamic texture descriptor. The dynamic texture descriptor is an extension of the local binary patterns. The image sequences are divided into regions. A flow is formed based on the similarity of the dynamic texture descriptors on the regions. We used real dataset for experiments. The results are promising.

1. Introduction

The event recognition in video surveillance is an important research topic. Normally the object (such as a person or a car) is detected, and tracked as a single object. Then a simple activity recognition, including people walking, running, skipping, can be performed [1][2]. However, the assumption that objects can be separated and tracked often fails in the crowd environment. For example, surveillance cameras in airport can capture a lot of persons walking together. Thus the commonly seen activities, such as people moving in different directions, crowd formation and dispersal, become hard to be detected and be represented reliably.

There is much interest to the event detection in crowd environment. Ali and Shah [3] developed an activity representation by modeling the crowded scene as a fluid flow to bypass object detection and tracking. They used the Lagrangian Particle Dynamics to segment high density crowd flows and detect flow instabilities. Andrade et al. [4] proposed a technique to automatically detect abnormal events in crowds. They characterized crowd behavior by observing the crowd optical flow and used unsupervised feature extraction to encode normal crowd behavior. Marana et al. [5] presented a technique for crowd density estimation based on Minkowski fractal dimension. Reisman et al. [6] presented a real time system that was able to detect crowds at distances of up to 70m. The system used slices in the spatiotemporal domain to detect inward motion as well as intersections between multiple moving objects. Ma and Cisar used dynamic texture for people segmentation in the crowd environment [7]. For a more comprehensive survey on activity recognition in crowded environments, we refer the reader to [8].

In this paper, we present an event detection method using the dynamic texture descriptor. The dynamic textures are the textures with motion. Specifically, we use the Local Binary Patterns from Three Orthogonal Planes (LBP-TOP) descriptor. We first partition the image frame into multiple regions. Each region and the consecutive several frames' regions form a volume. The LBP-TOP descriptor is calculated on the volume. Next, we form a flow by connecting several volume regions together along the time axis. Then the descriptor for the event is represented by the LBP-TOP descriptors along the flow. For the classification on the event, the distance between the test sequence's descriptor and the model sequence's descriptor is calculated using the log-likelihood statistics. This allows us to recognize events of people and cars in different video scenarios. This paper is organized as follows. Section 2 gives an overview of dynamic texture descriptor. Section 3 presents the proposed methods using the dynamic texture for the event recognition. We present experimental results in Section 4. Section 5 is discussion.

2. Dynamic texture descriptor

2.1. Previous work

In this section, we briefly describe the previous work for the Local Binary Pattern (LBP), its extension to the LBP-TOP descriptor [9][10][11][13][14][15].

The LBP was introduced as a texture descriptor [9][10]. An LBP descriptor for a local neighborhood on a center pixel is calculated with the eight neighbors using the grey level of the center pixel as a threshold. The LBP descriptor captures the spatial structure of local image texture, and has been successfully used in the application of texture classification [10].

978-1-4244-3993-5/09/$25.00 ©2009 IEEE

Zhao and Pietikäinen [11] extended the LBP descriptor to Volume LBP to capture the dynamic texture on the image sequences. However, the number of patterns will become very large when the number of neighboring pixels increases. Therefore the LBP-TOP descriptor was developed [13] for a fast calculation of the LBP features for dynamic texture. The LBP-TOP descriptor is a concatenation LBP on three orthogonal planes, XY, XT and YT.

Zhao and Pietikäinen [14] applied the LBP-TOP descriptor for the application of facial expression recognition. They first divided the face image into several regions. The LBP-TOP descriptor is calculated for each region. They viewed the facial expression recognition problem as a classification problem. The distance between a sample histogram and the model histogram is calculated using the log-likelihood statistics. Kellokumpu, Zhao and Pietikäinen [15] applied the LBP-TOP descriptor for action recognition including people bending, jumping when a person is well seperated. They only used the XT plane and the YT plane to calculate the LBP-TOP descriptor.

2.2. LBP-TOP descriptor

Before we present the LBP-TOP descriptor, we first present the LBP descriptor. The original LBP descriptor on a center pixel is calculated with the eight neighbors using the grey level of the center pixel as a threshold. Ojala et al. [10] defined a texture T in a local neighborhood of a gray image as a joint distribution of the gray level of the center pixel and those of its neighborhood pixels

$$T = t(g_c, g_o, ..., g_{P-1}) \quad (1)$$

where g_c is the gray level of the center pixel and P is the number of neighborhoods. Then we subtract the gray level of the center pixels and factorize the joint distribution.

$$T \approx t(g_c)t(g_o - g_c, ..., g_{P-1} - g_c) \quad (2)$$

Normally $t(g_c)$ is omitted, since it represents the whole luminance of an image. Moreover, in order to have the invariance with the scales of the gray level, only the signs of the difference are considered, so the LBP code is represented as follows:

$$\mathrm{LBP}(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c)2^p, \quad (3)$$

$$s(x) = \begin{cases} 1, x \geq 0 \\ 0, x < 0 \end{cases} \quad (4)$$

The reason to call the LBP code is because a binomial weight 2^p is assigned to each indication function between the gray level of the center pixel with the gray level of each neighborhood pixel.

The texture in one image can be extended to dynamic texture in one image sequence. Consider a volume neighborhood centered at pixel (x_c, y_c) at time t_c, the volume LBP is defined as the joint distribution of the gray levels of $3P+3$ image pixels on the current frame t_c, the previous neighboring frame $t_c - L$, and the next neighboring frame $t_c + L$ as follows:

$$V\mathrm{LBP}(x_c, y_c, t_c) = \sum_{p=0}^{3P+1} s(g_p - g_c)2^p \quad (5)$$

where g_c represents the gray level of the center (x_c, y_c) at time t_c, and g_p represents the gray level of the neighborhood pixels within the spatiotemporal volume.

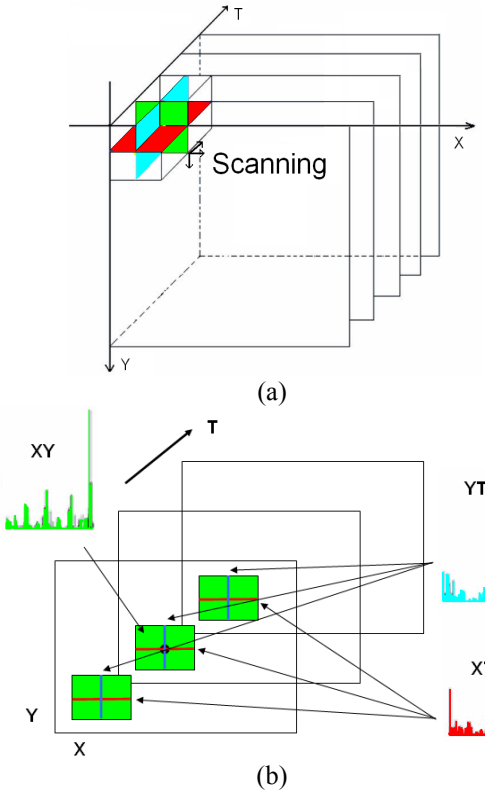

Figure 1: Illustration of the LBP-TOP descriptor (a) Calculation of three LBP codes for each point of dynamic texture through scanning, (b) histograms of LBP codes of three orthogonal planes.

When the number of neighborhood increases, the number of patterns for the VLBP increases by 2^{3P+2}, so the number will be very large, next we present the LBP-TOP descriptor [13]. The LBP-TOP descriptor is a descriptor concatenating the LBP on the three orthogonal planes: XY plane, XT plane, and the YT plane. In Figure

1(a), the XY plane represents the image frame; the XT plane represents the dynamics in the horizontal direction, and the YT plane represents the dynamics in the vertical direction. In Figure 1(b), the LBP code is calculated on the XY, XT, and the YT planes. The XY-LBP code represents the spatial information, and the XT-LBP code and the YT-LBP code represent the spatial temporal information. The number of bins is only $3 \cdot 2^p$ rather than 2^{3P+2} for VLBP. In LBP-TOP, all the pixels are calculated as the center pixel, and then the statistics is represented as a histogram on each plane. The three histograms are concatenated into one histogram as LBP-TOP descriptor.

$$H_{i,j} = \sum_{x,y,t} I\{LBP(x,y,t) = i\} \qquad (6)$$

$$I(A) = \begin{cases} 1 & A\_is\_true \\ 0 & o.w. \end{cases}$$

where $LBP(x,y,t)$ represents the LBP code of center pixel (x, y, t) in the jth plane (j = 0: XY plane, j = 1:XT plane, and j = 2: YT plane).

The radius in the time axis does not need to be the same as the radius in the space axis. We denote the radius in axe X is R_X, the radius in axe Y is R_Y, and the radius in axe T is R_T. Also, the number neighborhood points in XY, XT and YT plane corresponds to P_{XY}, P_{XT}, and P_{YT}. Then the notation of the LBP-TOP descriptor can be represented as $LBP-TOP_{P_{XY},P_{XT},P_{YT},R_X,R_Y,R_T}$.

The distance between two LBP-TOP descriptors can be measured using the log-likelihood statistic as follows:

$$d(H^1,H^2) = -\sum_{b=1}^{B} H_b^1 \log H_b^2 \qquad (7)$$

where B is the number of bins, H_b^1 represents the sample's probability at bin b for Histogram 1, and H_b^2 represents the sample's probability at bin b for Histogram 2. Other distance between the two histograms, such as histogram intersection distance can be used.

3. Proposed method

In this section, we present a region based method using the LBP-TOP descriptor. We partition the image frame into regions. Each region in the current frame and its next several frames form a spatio-temporal volume, as shown in Figure 2. Each spatio-temporal region is a three dimension of width X, height Y, and number of time frames T. The spatio-temporal regions can be overlapping or non-overlapping. If two spatio-temporal regions overlap, we denote Xo, Yo as the distance between centers of the regions in X axe and Y axe respectively, and To as the difference between central frames in T axe.

Figure 2: Image regions from consecutive frames.

The LBP-TOP descriptor of the region is calculated using the method (6) in Section 2. The event may cover many frames, which is longer than the number of frames forming the volume region, so the regions are connected temporally to form a flow.

There are three components to form a flow using the regions: temporal association of the patches, a stop criterion for the temporal association, and flow editing to remove noise.

Temporal association is to connect a region at time t to a region at time t+1. We assume that the patch at time t + 1 that is connected to a patch $B_i(t)$ at time t lies in its spatial neighborhood. A patch $B_j(t+1)$ is associated with $B_i(t)$-

$$\hat{j} = \arg\min d(\mathrm{H}(B_i(\mathrm{t})), \mathrm{H}(B_j(\mathrm{t}+1))) \qquad (8)$$

where the $d(.)$ is the distance between the two LBP-TOP descriptors (7). This procedure is continued until a stopping criterion is satisfied.

We considered two stopping criteria. First, if all nine neighboring patches at time t + 1 do not exhibit any motion, we assume that the object either stops moving or has left the scene. The corresponding flow forming is ceased. Second, if objects overlap leading to the occlusion of the object in consideration, the LBP-TOP descriptors of all nine neighborhood patches will differ greatly with the LBP-TOP descriptors of the current patch B_i (t). In particular, if the distance $>\mu+3\delta$, where μ is the mean and δ is the standard deviation of the distances calculated during the flow formation, the corresponding flow is ceased.

Figure 3 shows an example of flows formed in an occlude situation using the second stop criteria. The upper three image frames in Figure 3 represents an image sequence. The first image frame shows 'two persons move towards' in the left of the image, the second image frame shows 'the two persons get occluded', and the third image frame shows 'the two persons then separate'. The lower

image in Figure 3 presents the flows corresponding to the occluded person get divided at the time of the occlusion using the stopping criteria described above. So there are two red flows correspond to the person got occluded.

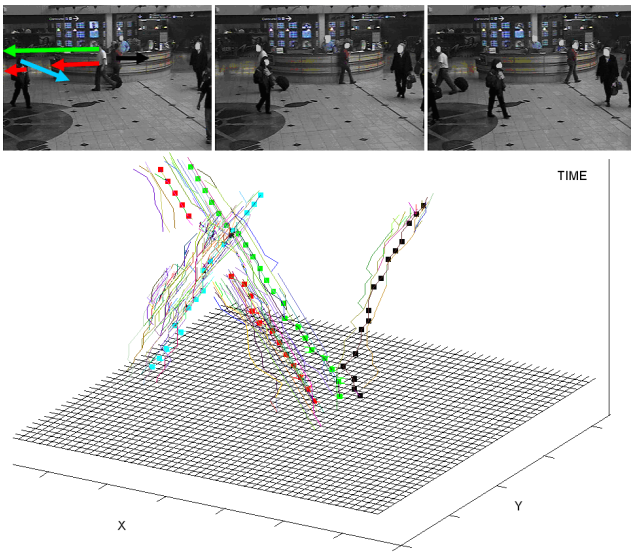

Figure 3: Upper images are an example of overlapping movement; lower image is the formed streamlines, the overlapped object is divided into two parts according to the stop criteria

After that, we perform flow editing. We filter out three types of noise. First, we filter out the flows with the starting region on the boundary of the image. Those flows may join neighboring regions out of the boundaries of the image in the next few frames. Second, flows with no significant change of coordinates are considered noisy flows and are removed. These flows correspond to isolated movement observed in the background. Third, spurious flows with their lengths shorter than a predefined threshold are filtered out.

The resulting flows are a robust set of flows that can be used for the event detection. Each flow is represented by the LBP-TOP descriptors along the flow.

The distance between the two flows is defined as a distance for classification of the flows' event. Suppose a test flow Sq and a model flow Sp in the database, we can calculate

$$d(S_q, S_p) = \frac{1}{n}\sum_{t=1}^{n} d(H(B_q(t)), H(B_p(t))) \quad (9)$$

where n is number of the LBP-TOP descriptors in each flow. The LBP-TOP descriptors for each volume region along the flow describe the dynamic information of the flow.

4. Experimental results

In this section, we present the experimental results for the proposed event recognition method. We conducted the experiments on three sets of data. The first one is the pedestrian dataset used in [16]. The second one is the retail escalator data and the vehicle traffic data from the UCF dataset [3]. The third data is a subway dataset from iLid dataset [17].

The pedestrian dataset [16] was used for the first experiment. The dataset contains video of high traffic of pedestrians from a stationary camera. The resolution of the video sequence is 238*157 at 10 frames per second. The pedestrians in the dataset are moving in two directions: moving right-up and moving left-down. We extracted the part of the data where high density of the crowd is presented with many occlusions of the pedestrian. We use 22 flows each with 7 frames for training of the LBP-TOP descriptor for two activities. For each type of event, 11 flows containing the activity were selected. The model was calculated and stored in the database. We used 400 image frames for testing the proposed method.

The video sequence was divided into spatio-temporal overlapping regions with the size X = 8, Y = 9 pixels and T = 7 frames of overlapping Xo = 4, Yo = 4, and To = 1. The parameter setting for the LBP-TOP descriptor is LBP-$TOP_{10,10,10,5,5,3}$. In the testing phase, each region was classified into one of the trained classes using the distance defined by (9). Figure 4 shows the event recognition results. The red patches show the people moving right-up direction, while the green patches show the people moving left-down direction using our method. The results showed that the method we presented is suitable for the pedestrian dataset.

In the second experiment, we used the UCF dataset [3]. We use two sets of data from this dataset: the retail escalator data and the vehicle traffic data. Figure 5 shows two frames of the event recognition results on the retail escalator set of videos. Figure 6 presents the event recognition results on the vehicle traffic set of videos from the dataset. Each region is colored by the color that corresponds to the color of the event class.

In the third experiment, we used the i-Lid dataset. The i-Lid dataset consists of video data of subway platform where people enter or leave the train. Different levels of people density are presented in the video data, and people are overlap often. The original resolution of the video data is 720*576 with 25 frames per second. The video data was down sampled the video data to 188*144 with 12.5 frame per second for faster processing.

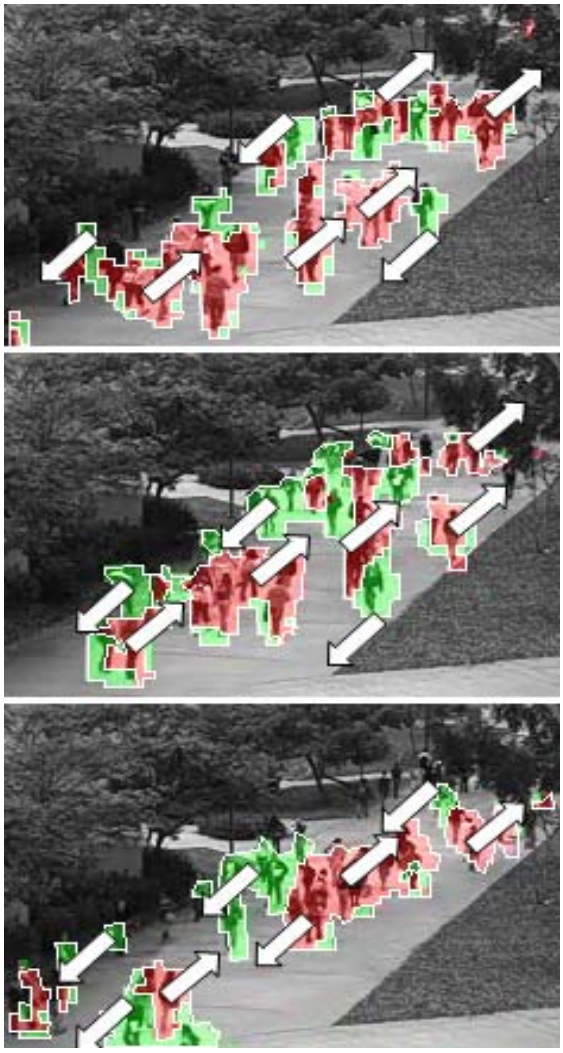

Figure 4: Sample results of activity recognition into two classes using the LBP description, red = patches with right-up motion, green = patches with left-bottom motion.

We selected *n* video sequences for each different activity (class). Each sequence was divided into spatio-temporal regions, and the flows were formed. The flows were formed for each testing sequence and each flow's event was automatically classified by proposed approach. After that we compared with the ground truth to calculate the confusion matrix as shown in the Table I.

There are four events in the dataset: people moving up, people moving down, people moving left, and people moving right. Figure 7 shows examples of the event in the subway dataset for the corresponding event. For each class we had more than 16 flows for training. For testing we have 4 sequences and 83 flows on the average for each class. In the table I we present the confusion matrix for the four types of events. These results were obtained with following settings: spatio-temporal volumes with the size X = 15, Y = 13 pixels and T = 5 frames with overlapping Xo = 6, Yo = 6 and To = 1 and descriptor LBP-$TOP_{9,9,9,5,5,3}$. The optimal width and height of spatio-temporal volume should correspond to the ¼ of the objects in the scene and the overlapping should correspond to the ½ of the size of the volume. The classification accuracy is very good for these four activities. The events are not simple because it contains many overlaps of the objects.

TABLE I
CLASSIFICATION ACCURACY - SUBWAY DATASET

Activity	Down	Up	Left	Right
Down	96%	0%	4%	0%
Up	0%	96%	0%	4%
Left	30%	0%	70%	0%
Right	0%	0%	0%	100%

5. Discussion

Event detection in video surveillance is important. The trajectory based descriptor, such as spatial coordinate, velocity, and shape works if the individual object can be segmented and tracked. However, the high density environment is common in video surveillance data, where the trajectory based activity descriptor work poorly in this environment.

In this paper, we present an event detection using the dynamic texture descriptor. The dynamic texture captures the stationary properties in time [18]. We use the LBP-TOP descriptors. We first partition image sequences into regions. Then we form motion flow by temporally connecting the region in the current frame to the patch in the next frame. Event representation is from the LBP-TOP descriptors extracted from the flow. We use various real data sets to test the proposed method. The experimental results show good performance for the event recognition.

References

[1] Y. Ma, S. B. Damelin, O. Masoud, N. Papanikolopoulos, "Activity Recognition Via Classification Constrained Diffusion Map". In: International Symposium on Visual Computing, pp. 1–8, 2006

[2] Y. Ma, B. Miller, P. Buddharaju, M. Bazakos, "Activity Awareness: From Predefined Events to New Pattern Discovery," In: IEEE International Conference on Vision Systems, New York, NY, USA, January 5-7, 2006.

[3] S. Ali, M. Shah, "A Lagrangian Particle Dynamic Approach for Crowd Flow Segmentation and Stability

Analysis," in IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN 2007.
[4] E. Andrade, S. Blunsden, and R. Fisher, "Detection of emergency events in crowded scenes". In IEEE International Symposium on Imaging for Crime Detection and Prevention, 2006.
[5] A. Marana, L. Costa, R. Lotufo and S. Velastin, "Estimating Crowd Density with Minkowski Fractal Dimension", IEEE International Conference on Acoustics, Speech, and Signal Processing, 1999.
[6] P. Reisman, O. Mano, S. Avidan, A. Shashua, M. Ltd, and I. Jerusalem, "Crowd detection in video sequences", IEEE Intelligent Vehicles Symposium, 2004.
[7] Y. Ma and P. Cisar, "Motion Analysis Using Dynamic Texture in Crowd Environment", Image Analysis - From Theory to Applications, Research Publishing, 2008, pp. 49–54.
[8] B. Zhan, N. Monekosso, P. Remagnino, S. Velastin, and L. Xu, "Crowd analysis: a survey", In Machine Vision and Applications, 2008, 19, pp. 345-357.
[9] T. Ojala, M. Pietikäinen and D. Harwood, "A comparative study of texture measures with classification based on featured distributions", Pattern Recognition 29, pp. 51-59, 1996.
[10] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns", IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 24, no. 7, pp. 971–987, 2002.
[11] G. Zhao and M. Pietikäinen, "Dynamic Texture Recognition Using Volume Local Binary Patterns", Proc. European Conference on Computer Vision, 2006 Workshop on Dynamical Vision, Graz, Austria.
[12] http://www.cwi.nl/projects/dyntex/
[13] G. Zhao and M. Pietikäinen, "Local Binary Pattern Descriptors for Dynamic Texture Recognition," in International Conference on Pattern Recognition, 2006, pp. 211-214.
[14] G. Zhao and M. Pietikäinen, "Dynamic Texture recognition Using Local Binary Patterns With an Application to Facial Expressions," in IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 29, no.6, pp. 915 – 928, 2007
[15] V. Kellokumpu, G. Zhao, M. Pietikäinen, "Human Activity Recognition Using a Dynamic Texture Based Method", in British Machine Vision Conference (BMVC), 2008
[16] A. Chan and N. Vasconcelos, "Modeling, Clustering, and Segmenting Video with Mixtures of Dynamic Textures," IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 30(5), pp. 909-926, May 2008.
[17] i-lids dataset for avss, 2007.
[18] G. Doretto, A. Chiuso, S. Soatto, and Y. Wu, "Dynamic textures", International Journal of Computer Vision, Vol. 51, no.2, pp 91-109, 2003.

Figure 5: Activity recognition shown for sample frames from the UCF retail escalator dataset. The red and green arrows denote the two major motion patterns observed in the video sequences. The corresponding color coded patches denote the labels outputted by our system.

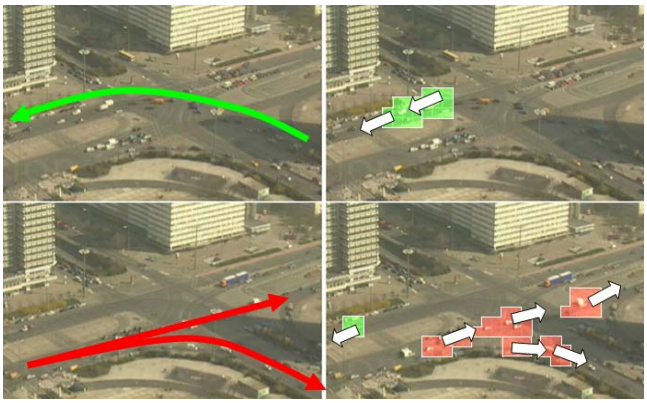

Figure 6: Activity recognition shown for sample frames from the UCF traffic dataset. The red and green arrows denote the two major motion patterns. The corresponding color coded patches denote the outputted activity labels. The small resolution of objects in the scene causes some patches to be erroneously classified.

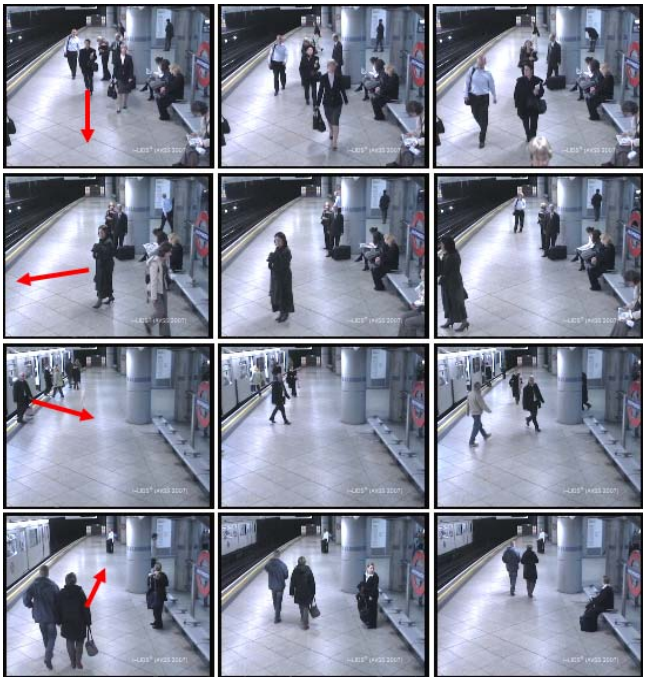

Figure 7: Examples of four events in the i-Lid dataset: person moving down, person moving left, person moving right.