Activity Representation in Crowd

Yunqian Ma[1] and Petr Cisar[2]

[1] Honeywell Labs, 1985 Douglas Drive, Golden Valley, MN 55422, USA
yunqian.ma@honeywell.com
[2] Honeywell Prague Laboratory, V Parku 2326/18, 14800 Prague, Czech Republic
petr.cisar@honeywell.com

Abstract. Video surveillance of large facilities, such as airports, rail stations and casinos, is developing rapidly. Cameras installed at such locations often overlook large crowds, which makes problems such as activity and scene understanding very challenging. Traditional activity recognition methods which rely on input from lower level processing units dealing with background subtraction and tracking, are unable to cope with frequent occlusions in such scenes. In this paper, we propose a novel activity representation and recognition method that bypasses these commonly used low level modules. We model each local spatio-temporal patch as a dynamic texture. Using a Martin distance metric to compare two patches based on their estimated dynamic texture parameters, we present a method to temporally stitch together local regions to form activity streamlines and represent each streamline by its constituent dynamic textures. This allows us to seamlessly perform activity recognition without explicitly detecting individuals in the scene. We demonstrate our method on multiple real data sets and show promising results.

1 Introduction

Activity representation and understanding in video sequences has been extensively studied over the last few years. For simple activity recognition, people use shape features to perform human movement analysis [1]. Our previous activity recognition works can identify people walk, run, skip, etc. [2][3]. For complexity activity recognition, people use hidden markov model, dynamic Bayesian networks method to inference [4][5]. However, most activity recognition methods show poor results when a large number of people are seen occluding each other in the scene. Surveillance videos in public areas such as airports and railway stations often contain crowded environments causing problems for most computer vision applications. Many of these problems stem from the inability of low-level vision algorithms (such as background subtraction, tracking, feature extraction etc.) to deal with crowded video data set. Thus commonly seen activities such as people moving in different directions, crowd formation and dispersal, etc. become very hard to detect and represent reliably.

In this paper, we present a motion texture activity representation to capture dynamic content that is not well represented by individual objects, or small numbers of interacting objects. The types of activities can include people moving

N. da Vitora Lobo et al. (Eds.): SSPR&SPR 2008, LNCS 5342, pp. 107–116, 2008.

in different direction, crowd formation and dispersal, etc. To represent it, we use dynamic textures for characterizing the underlying activities without requiring the inference of objects trajectories. Dynamic texture can be viewed as sequences of images of moving scenes that exhibit certain stationary properties in space and time [6]. [6] showed that linear dynamic textures can be characterized as the output of an auto-regressive moving average model. They obtain a closed form solution for learning the parameters of a linear dynamic texture. They demonstrated being able to extrapolate dynamic texture sequences as well as classify them based on the learnt model. Saisan et. al [7] further extended this work to model and recognize over 50 different dynamic textures. They evaluated three distance measures - the Geodesic distance metric, the Martin distance metric and Subspace angles.

Fujita et. al [9] extended the notion of a single dynamic texture spanning the whole image to multiple dynamic textures in different regions of an image. They partitioned the image into several regions, each region having its own dynamic model. They use a distance metrics for a non-linear space, so the features of dynamic textures are based on the impulse responses of state variables. These responses were shown to be efficient to compute and effective for matching purposes.

Chan et. al [8] proposed using mixtures of dynamic textures to model and segment video sequences. They used an expectation maximization algorithm to learn the parameters of the model and demonstrated their approach on natural scenes and highway traffic videos. Zhao et. al [10] extended the Local Binary Pattern operator commonly used in texture analysis to combine appearance as well as motion information. This operator was rotation invariant and robust to gray-scale variations making and was used to recognize dynamic textures. Our previous works in dynamic texture is for spatial segmentation [11] for video surveillance data.

In this paper, we first partition the whole image to multiple regions. Each region (patch) has its own dynamic models. Second, we link multiple spatio-temporal patches in time to form video streamlines. This consists of three components: temporal association of patches, stop criterion for the temporal association and streamline editing to filter out noisy links. We then aggregate multiple streamlines in a video sequence using the Karcher mean concept [12] to form an activity model. An extension of the Martin distance serves as a distance metric between two activities. This allows us to recognize activities of people and cars in different video scenarios. We demonstrate our approach on a wide variety of scenes such as an airport data set, a subway station data set, a retail store data set and a traffic intersection data set.

The rest of the paper is organized as follows. Section 2 gives a brief overview of dynamic texture modeling and presents the distance metrics that between two dynamic texture models. Section 3 discusses activity representation and recognition using dynamic texture. We present experimental results in Section 4. Section 5 is conclusion.

2 Dynamic Textures

In this section, we begin by providing a brief overview of the research in dynamic textures. The interested reader can refer the detail in [6]. For a sequence of images, we use $y(t) \in R^m, t = 1, ..., \tau$ to represents pixel intensity. Doretto et al. [6] view $y(t)$ as a noise observation on a linear dynamic texture. Therefore, they represent this linear dynamic texture by an auto-regressive, moving average process with unknown input distribution:

$$\begin{aligned} x(t+1) &= Ax(t) + v(t),\ v(t) \sim N(0, Q); x(0) = x_0 \\ y(t) &= Cx(t) + w(t), \qquad\quad w(t) \sim N(0, R) \end{aligned} \tag{1}$$

where $x(t) \in R^n$ is the hidden state, $A \in R^{n \times n}$ is the state transition matrix, representing the dynamics. $C \in R^{m \times n}$ is the output matrix, $v(t)$ is the driving input to the system, which is assumed to be Gaussian white noise, $w(t)$ is measurement noise.

A system identification method can be used to estimate the dynamic texture parameters. Let's form a matrix $Y_1^\tau = [y(1), ..., y(\tau)] \in R^{m \times \tau}$ for the sequence of τ frames. To make the estimation of the dynamic texture unique, [6] let $m \gg n$ and $rank(C) = n$, and use canonical model $C^T C = I_n$ (I_n is the $n \times n$ identity matrix) to give a closed form solution by singular value decomposition $Y_1^\tau = U \Sigma V^T$. The estimated dynamic texture parameters of matrix C and A are

$$\begin{aligned} \hat{A} &= \Sigma V^T \begin{bmatrix} 0 & 0 \\ I_{r-1} & 0 \end{bmatrix} V (V^T \begin{bmatrix} I_{r-1} & 0 \\ 0 & 0 \end{bmatrix} V)^{-1} \Sigma^{-1}, \\ \hat{C} &= U \end{aligned} \tag{2}$$

The discrimination between two dynamic models can be calculated through parameters A and C, since the dynamic texture is viewed as ARMA model. Several distance matrices between dynamic textures were proposed [7][9]. For example, the subspace angle between two dynamic textures parameters $[A_1, C_1]$ and $[A_2, C_2]$ can be computed. Using these subspace angles $(\theta_i,\ i = 1, ..., n)$, [7] used the Martin distance as the distance between the two dynamic texture parameters.

$$d = \ln \prod_{i=1}^{n} \frac{1}{\cos^2(\theta_i)} \tag{3}$$

3 Activity Representation

Our proposed activity representation framework is based on the analysis of motion flow in different parts of the video sequence. This allows us to recognize video sequence that correspond to different activities.

3.1 Motion Flow Forming

We begin by partitioning an image into overlapping rectangular regions. We group temporally neighboring regions to form a spatio temporal volume (which we call a *patch*) and consider each patch volume to be a dynamic texture. For

example, we can use five consecutive frames to form the patch volume. The dynamic texture parameters of the patch volume are estimated as in Section 2 and the Martin distance metric is used as a distance measure between two dynamic textures. Patches are then connected temporally to form streamlines, as shown in Figure 1. There are three components to streamline formation: temporal association of the patches, a stop criterion for the temporal association and streamline editing to remove noise.

The temporal association of patches is carried out as follows. Each patch's dynamic texture parameters are estimated independent of other patches in the neighborhood. The estimation of these parameters is carried out using a five frame sequence in the immediate future of the current frame. Temporal association is to connect a patch at time t to a patch at time $t+1$ based on the distance between the corresponding dynamic texture parameters. If the motion of objects in the video sequence is sufficiently small, one can assume that the patch at time $t+1$ that is connected to a patch $B_i(t)$ at time t lies in its immediate spatial neighborhood. Thus we compute the Martin distance between patch $B_i(t)$ and the nine patches in the immediate spatial neighborhood at time $t+1$. A patch $B_{\hat{j}}(t+1)$ is temporally associated with $B_i(t)$ if,

$$\hat{j} = \text{argmin } d(DT(B_i(t)), DT(B_j(t+1))) \tag{4}$$

This procedure is continued until a stopping criterion is satisfied, resulting in a streamline.

Next we present two stopping criteria for the streamline formation procedure. First, if all nine neighboring patches at time $t+1$ do not exhibit any motion, one can assume that the object has either stopped moving or has left the scene. Thus the corresponding streamline is ceased. Second, if objects overlap leading to the occlusion of the object in consideration, the dynamic texture parameters of all nine neighborhood patches will differ greatly with the dynamic texture parameters of the current patch $B_i(t)$. In particular, if $d(DT(B_i(t)), DT(B_j(t+1))) > (\mu + 3\delta)$, where μ is the mean and δ is the standard deviation of the Martin distances calculated during the streamline formation, the corresponding streamline is ceased.

Finally we perform streamline editing. We filter out three types of noise. Firstly, we filter out the streamlines whose starting patch is on the boundary. Such streamlines may join neighboring patches out of the boundaries of the image in the next few frames, and are thus removed. Secondly, streamlines with no significant change of coordinates are considered noisy streamlines and are removed. These streamlines correspond to isolated movement observed in the background. Thirdly, spurious streamlines characterized by lengths shorter than a predefined threshold are filtered out. The resulting set of streamlines is a robust set of streamlines that can be used for the purpose of activity recognition.

Figure 2 presents an example of streamlines formed in an occlude situation. The upper images in Figure 2 shows three image frames at three time instants to represent a image sequence. Two persons move towards, get occluded and then separate. The lower image in Figure 2 present the streamlines corresponding to

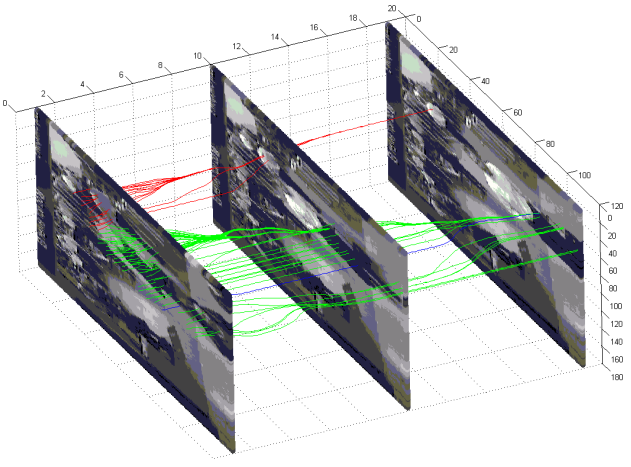

Fig. 1. Flow forming by connecting patches using the similarity between the dynamic texture parameters of the patches

the occluded person get divided at the time of the occlusion using the stopping criteria described above.

3.2 Activity Recognition

The proposed activity representation uses the motion flow formed in the above section. For example, a query sreamline S^q composes DT_t^q, where $DT, t = 1, ..., n$ means dynamic texture parameters, including A and C. That is, the estimated dynamic texture parameters of the patches forming a streamline can be used as features to describe the streamline. An extension to the Martin distance serves as a distance metric between any two streamlines. This metric is used to compare a query streamline S^q with a streamline S^p representing activities in the training data as follows,

$$d(S^q, S^p) = \frac{1}{n} \sum_{t=1}^{n} d(DT_t^q, DT_t^p) \tag{5}$$

We perform activity recognition by comparing the distance of a query activity (streamline) to all activities in the training database. Then we use a k-nearest neighbor classification method to select the best match.

There may be many streamlines labeled with the same activity category in the training data. Instead of performing an exhaustive search over every training data, which is time consuming, we present a method to aggregate multiple streamlines in each class of the training database to form one representative streamline. Thus the number of comparisons that must be made to classify a query streamline will equal the number of categories in the training dataset. Since the dynamic texture parameters are not defined in a Euclidean space, a simple mean calculation cannot be used to aggregate multiple streamlines. Instead, in

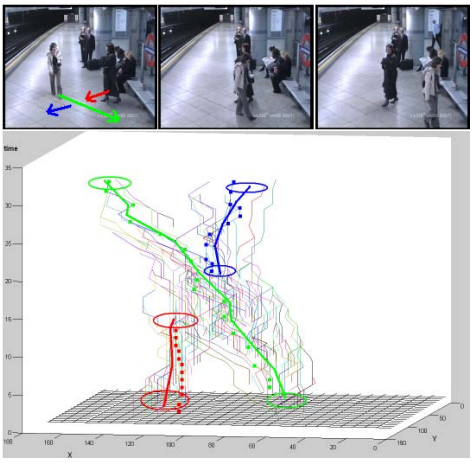

Fig. 2. upper images are an example of overlapping movement; lower image is the formed streamlines, the overlapped object is divided into two parts according to the stop criteria

this paper we use the Karcher mean concept to aggregate multiple streamlines into a representative set of dynamic texture parameters S^{rep} given by,

$$rep(t) = \underset{k}{\operatorname{argmin}} \sum_{j} d(DT(B_k(t)), DT(B_j(t))) \tag{6}$$

$$\text{for } j \neq k, t = 1, ..., n \tag{7}$$

where $d(\cdot)$ denotes the Martin distance given by Equation (3).

We perform activity recognition using the nearest neighbor classification by comparing the distance between the query streamline S^q with the representative streamline S^{rep} for each activity category in the training data.

4 Experimental Results

In this section, we present experimental results for the proposed activity representation and recognition method. All the experiments were conducted on a PC with an Intel Core2, 2.4 GHz processor and 2GB RAM.

Dataset: the experiments were conducted on video sequences from various real dataset. First, we have airport dataset collected at an airport. This dataset is also used in our previous paper [12]. Second, we downloaded the public dataset, i-Lid dataset[1] [13], which consists of video sequences collected from surveillance cameras overlooking a subway station. Third, we downloaded the UCF dataset from [14]. We used two sets of videos from this dataset: the retail escalator video sequences and the vehicle traffic video sequences.

[1] http://www.elec.qmul.ac.uk/staffinfo/andrea/avss2007_ss_challenge.html

For each dataset, we selected n video sequences with k different activities. Activities in every video sequence were manually labeled to form the ground truth. Half the sequences were used for training, and the other half for testing. The training video sequences were utilized to obtain representative streamlines for the different activities as given in Section 3.2. The performance of the system is given by the classification accuracy of all streamlines in the video sequences used for testing.

Table 1 gives the performance of the proposed method on the airport dataset. This consists of two major activities: people moving towards the camera (down) and away from the camera (up). The training data includes 12 *down-streamlines* and 18 *up-streamlines*. The test data includes 74 down-streamlines and 161 up-streamlines. Table 1 shows the classification accuracy for the two activities.

Table 1. Classification accuracy - Airport dataset

Activity	Down	Up
Down	97%	3%
Up	0%	100%

Table 2. Classification accuracy - Subway dataset

Activity	Down	Up	Left	Right
Down	97%	0%	3%	0%
Up	0%	76%	24%	0%
Left	15%	0%	85%	0%
Right	0%	0%	0%	100%

Fig. 3. Activity recognition shown for sample frames from the subway dataset

Fig. 4. 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.

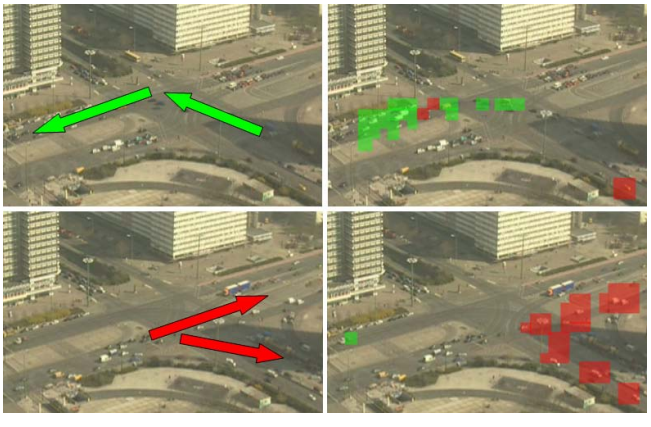

Fig. 5. Activity recognition shown for sample frames from the UCF traffic dataset. The red and green arrows denote the two major motion patterns. The corresponding color coded patches denote the outputted activity labels. The small resolution of objects in the scene causes some patches to be erroneously classified.

There are four major activities involved in the video sequences of the subway dataset- moving left, moving right, moving up, and moving down. Figure 3 shows a few frames from the subway dataset with the recognized activities. The training data included 17 down-streamlines, 18 up-streamlines, 19 left-streamlines and 13 right-streamlines. The test data included the following streamlines: 38 down, 153 up, 100 left and 44 right. Table 2 shows the classification accuracy matrix using the subway dataset.

The UCF data set is a challenging dataset with a larger number of object in the scene. Figure 4 shows a few sample frames of the activity recognition results on the retail escalator set of videos. Figure 5 presents the activity recognition results on the vehicle traffic set of videos from the dataset.

5 Conclusion

Activity representation and recognition in video surveillance is a very important area which has been received a lot attention from researchers. The high density environment is very common in video surveillance data, where the trajectory based activity representation which needs track individual person first work poorly in this environment. We present a novel activity recognition method using dynamic textures. We first partition image sequences into patches. Then we form motion flow by temporally connecting the patch in the current frame to the patch in the next frame. Activity representation is from the dynamic texture features extracted from the streamline. We use various real data set to test the proposed method. The experimental results show good performance for activity recognition. In the future, we are going to perform more complex activity representation and recognition under this direction.

References

1. Veeraraghavan, A., RoyChowdhury, A.K., Chellappa, R.: Matching Shape Sequences in Video with Applications in Human Movement Analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (December 2005)
2. Ma, Y., Damelin, S.B., Masoud, O., Papanikolopoulos, N.: Activity Recognition Via Classification Constrained Diffusion Map. In: International Symposium on Visual Computing 2006, pp. 1–8 (2006)
3. Ma, Y., Miller, B., Buddharaju, P., Bazakos, M.: Activity Awareness: From Predefined Events to New Pattern Discovery. In: IEEE International Conference on Vision Systems, New York, NY, USA, January 5-7 (2006)
4. Park, S., Aggarwal, J.K.: A Hierarchical Bayesian Network for Event Recognition of Human Actions and Interactions. Multimedia Systems; Special issue on Video Surveillance 10(2), 164–179 (2004)
5. Muncaster, J., Ma, Y.: Hierarchical Model-Based Activity Recognition With Automatic Low-Level State Discovery. Journal of Multimedia 2(5), 66–76 (2007)
6. Doretto, G., Chiuso, A., Wu, Y.N., Soatto, S.: Dynamic Textures. International Journal of Computer Vision 51(2), 91–109 (2003)
7. Saisan, P., Doretto, G., Wu, Y.N., Soatto, S.: Dynamic Texture Recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 58–63 (2001)
8. Chan, A., Vasconcelos, N.: Modeling, Clustering and Segmentating Video with Mixtures of Dynamic Textures. IEEE Trans. on Pattern Analysis and Machine Intelligence 30(5), 909–926 (2008)
9. Fujita, K., Nayar, S.K.: Recognition of Dynamic Textures using Impulse Response of State Variables. In: 3rd International Workshop on Texture analysis and synthesis (2003)
10. Zhao, G., Pietikainen, M.: Dynamic Texture Recognition Using Volume Local Binary Patterns. In: 9th European Conference on Computer Vision, Graz, Austria, May 7 - 13 (2006)
11. Ma, Y., Cisar, P.: Motion Analysis Using Dynamic Texture in Crowd Environment. Image Analysis - From Theory to Applications, Research Publishing, pp. 49–54 (2008)

12. Ma, Y., Miller, B., Cohen, I.: Video Sequence Querying Using Clustering of Objects' Appearance Models. In: Bebis, G., et al. (eds.) International Symposium on Visual Computing 2007, pp. 328–339 (2007)
13. i-lids dataset for avss (2007)
14. Ali, S., Shah, M.: A Lagrangian Particle Dynamic Approach for Crowd Flow Segmenta-tion and Stability Analysis. In: IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN 2007 (2007)