Activity Identification Utilizing Data Mining Techniques

Jae Young Lee
Dept. of Math. and Computer Sciences
Colorado School of Mines
Golden, Colorado, USA
jaelee@mines.edu

William Hoff
Engineering Division
Colorado School of Mines
Golden, Colorado, USA
whoff@mines.edu

Abstract

We propose a novel method that, given an unknown moving object trajectory, determines which known activity type the trajectory would belong to. The proposed method utilizes various data mining techniques, including clustering, classification, and Markov model. We collect trajectories of moving objects of known activity types and build one Markov model for each activity type. Given an unknown trajectory, we compute the likelihood of this trajectory belonging to each activity type using the Markov model and the trajectory is determined to belong to the activity type that results in the highest likelihood. We use only location information of moving objects. We do not use any other information such as color, size, or shape of objects, or contextual information. We demonstrate the effectiveness of this method using trajectories of students playing two sports activities – Ultimate Frisbee and volleyball. We show that the accuracy of this method is as high as 94%.

1. Introduction

Surveillance and monitoring of a large number of moving objects (e.g., humans or vehicles) have been studied extensively in recent years. One important task in this research is to identify an unknown moving object trajectory (or trajectories) as that of one of known activity types. We call this problem *activity identification problem*.

We propose a novel method to solve this activity identification problem. There are two main issues that need to be addressed: (1) What kind of information can be and need to be collected to model activity types, and (2) what kind of model would be most effective to capture the behavioral patterns of moving objects. With the advent of sophisticated surveillance systems, we can collect many different types of information from target objects. We can collect their shapes, colors, textures, sizes, movements, and other contextual information such as temporal information and geographic information in the surrounding area. However, it is not always possible to collect all this information. Even when all this information is available, it is desirable to use only a part of available information if that much is sufficient to perform an intended task because it will decrease the computational complexity and storage requirements. In the proposed method, we use only location information of moving objects. We show that we can still achieve a very high accuracy without using any other information.

Assume that we have a collection of trajectories, each of which is a sequence of (x, y)'s, for different activity types. If we represent each trajectory with a fixed number of features (or attributes), then the activity identification problem becomes a typical classification problem. We can build a classifier model from these trajectories and use it to classify an unknown trajectory. One problem in this approach is due to the fact that the behavior of a moving object trajectory is not homogeneous over time; rather it may change over time (e.g., walk slowly, run fast, and walk slowly again). Representing a trajectory with a set of features may capture a global behavior of the trajectory but it will effectively hide the local behaviors. One viable approach is to divide each trajectory into segments in such a way that each segment may represent an atomic behavior type (e.g., walk slowly, run fast, or move in a zigzag fashion) and to represent a trajectory as a sequence of such segments. As for a model to capture such information effectively, we choose a discrete-time Markov chain because it naturally captures the time-varying aspects of moving objects.

The paper is organized as follows. Section 2 reviews previous work. Section 3 presents a high-level description of the proposed method. Section 4 discusses how we extract trajectories of moving object from video images. Section 5 describes a model building process, Section 6 shows how we validated our method, and Section 7 concludes the paper.

2. Related work

Activity recognition from video has received widespread attention in the past several years within the computer science and AI communities. Many approaches extract low level features from the imagery and use statistical methods to classify events. Zhang et al. [14] divide the video into equal duration segments and compute spatial and motion histograms for each segment. The dynamic relationships of these features are then captured in a co-occurrence matrix.

Instead of computing global features such as a histogram, most approaches detect localized events such as the appearance of an object, or they compute the trajectory of a moving object. For example, Gaborski et al. [1] look for motion events in each 8-by-8 pixel region in the video. Each region has a pool of clusters that represents unique events observed by the system in that region. Localized novel events can be recognized in each region.

In our approach, we take continuous time series data (i.e., the $x(t)$, $y(t)$ locations of tracked objects) and represent it as a sequence of discrete symbols. This allows the tools of symbolic pattern recognition to be used to recognize specific patterns, or to detect anomalous patterns. For example, the behavior of an electronic circuit can be represented as a time sequence of symbols, where the symbols correspond to discretized phase measurements [8]. Another example is the representation of behaviors of visitors to web sites [2].

One way to model a behavior that is represented by a sequence of discrete symbols, is to use a Markov model. Having achieved good success in time series data, Markov models have been extended to recognizing activities using machine vision [4,13,16].

In the work described above, the "symbols" (e.g., states of the Markov model) are manually defined by the programmer, although the parameters of the model (such as transition probabilities) are learned automatically from examples. In contrast, our method automatically discovers the symbols through a clustering algorithm.

Much previous work on vision-based activity recognition makes use of additional information derived from the imagery, such as the shape, size, or other contextual information (e.g., [3, 5, 9, 11]). However, our work only uses the raw track position data, ($x(t)$, $y(t)$), and no other information. This makes our work potentially applicable to situations where the sensor only returns the position of the object and no other information; e.g., a laser tracker [7].

A study in [10] is similar in that it also discusses classification of trajectories. However, this study is about classification of trajectories of moving cells or viruses and uses fixed length trajectories. On the contrary, our method allows varying trajectory lengths and classifiers are built on segments of trajectories.

3. Overall process

The main idea of our method is as follows. If a trajectory is discretized and represented as a sequence of symbols, each of which represents an atomic behavior type, the collective behavior of an activity type, or the *signature* of the activity type, can be modeled by a discrete-time Markov chain. Suppose there are n activity types of interest. We first collect trajectories of these activity types, segment and discretize them, and build n Markov models, one for each activity type. Given an unknown trajectory, then, we compute the likelihood of this trajectory belonging to each activity type using the corresponding Markov model. The trajectory is determined to belong to the activity type that results in the highest likelihood.

When an unknown trajectory is tested, it needs to be discretized in the same way as training trajectories were discretized. For this purpose, we build a classifier model from the segments of training trajectories and use the model to discretize unknown/test trajectories. Figure 1 shows the overall process of our activity identification method. The left column shows a model building process and the right column represents a validation/test process. We used the data mining software *weka* [12] to perform necessary clusterings and classifications.

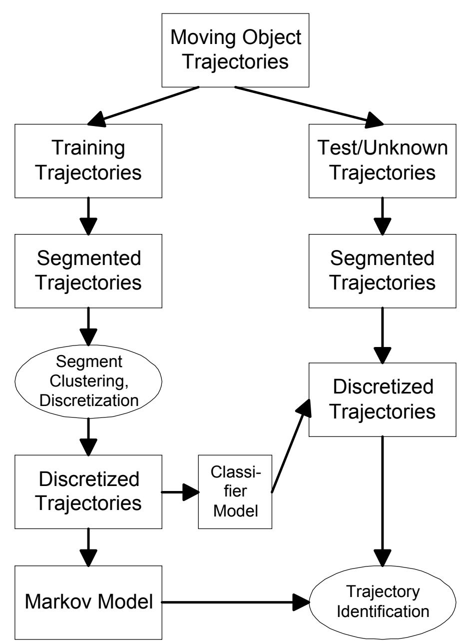

Figure 1. Overall activity identification process.

4. Trajectory extraction

We used a digital camcorder, and then transferred the video to a computer in the form of an AVI movie file. Progressive scan mode was used. We then converted the AVI movie into a sequence of still images. The images were 640x480x8 pixels (grayscale), and were taken at 30 frames per second.

An automatic tracker program was developed to automatically track all moving objects in the scene and write their coordinates to a trajectory file. First, a background, or "reference" image was obtained by averaging all the images. Because moving objects tend to pass quickly in front of the background, they tend to average out. The operation of the tracker was as follows:

1. Read in the next image of the sequence.
2. Take the difference between this image and the reference image. Threshold the difference image and find connected components.
3. For each component that is not already being tracked from previous images, start a new track. The track file information consists of the location of the object, its "bounding box" in the image, and an "appearance template" of the object within the bounding box. A "shape template" is also computed, which when thresholded is a binary image of the silhouette of the object (Figure 2).
4. For each tracked object, perform a cross correlation operation to find the most likely location of the template in the new image. Then update the template image of the object by computing a running average of its image.

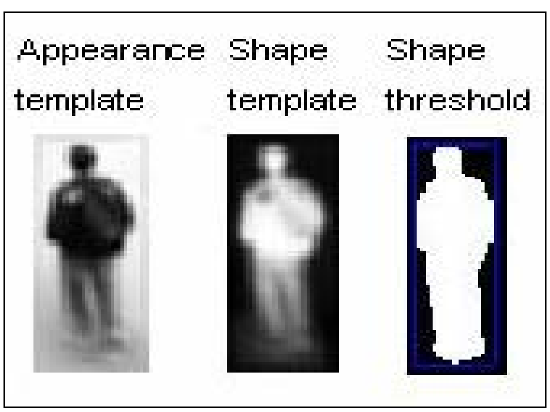

Figure 2. Object templates.

The actual world coordinates were also computed, using the assumption of a flat world. Very short trajectories (less than 3 seconds) were discarded, because they were considered to be due to noise. The (x, y) locations of the remaining trajectories were smoothed by running a low pass (Gaussian) filter, with σ = 0.4 seconds. Figure 3 shows video captures and the plot of resulting trajectories of Frisbee and volleyball games.

Although the automatic tracker worked well for most scenes, it gave incorrect results when objects occluded each other or moved together. Therefore, for the volleyball scenes, we manually extracted the trajectories using a mouse.

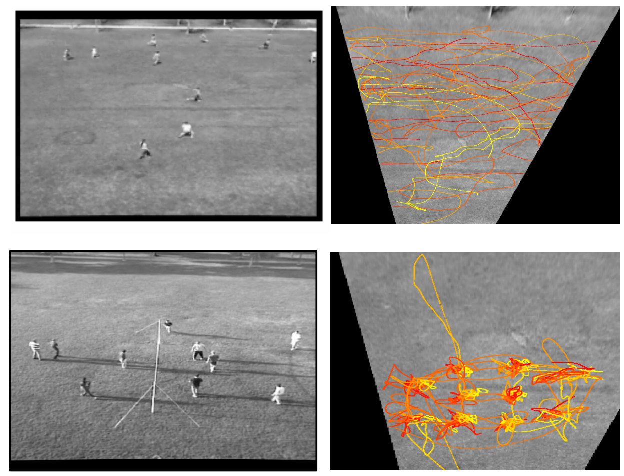

Figure 3. Video captures and trajectories of Frisbee and volleyball.

5. Model building

In this section, we discuss how we build Markov models from training trajectories. We also describe how we build a classifier model, which is used to discretize test/unknown trajectories.

5.1 Segmentation and discretization

The collected trajectories are divided into two sets – training data set and test data set. We first describe how training trajectories are discretized. Discretization of test trajectories will be discussed in Section 6. We divide each trajectory into segments of the same length in terms of the time duration of a segment. If the length of a segment is too long, it is possible that a segment may include more than one type of behavior (e.g., walking and running). On the other hand, if a segment is too short, the incremental behavioral change between two consecutive segments are too small and, thus, the resulting segmentation may become meaningless. We found the value of 3 seconds maximizes the overall activity identification accuracy. Since 30 samples were taken per second, there are 90 samples, or 90 (x, y) locations, in each segment.

Our goal at this stage is to automatically identify atomic behavior types in the training trajectories. Since there are no known behavioral labels on the segments, we need to apply a clustering algorithm. First, a behavior of a segment must be represented by a set of features. Let a segment $s = \langle p_1, p_2, \ldots, p_N \rangle$, where $N = 90$, $p_i = (t_i, x_i, y_i)$, and t_i is a timestamp. We considered, for each segment, the following features: average speed ($as = \sum_{i=1}^{N-1} v_i \,/\, (N-1)$), mean change of speed ($mCs = mean(cs_1 : cs_{N-2})$), variance of change of speed ($varCs = var(cs_1 : cs_{N-2})$), overall heading ($oh = h_{1,N}$), mean change of heading ($mCh = mean(ch_1 : ch_{N-2})$), variance of change of heading ($varCh = var(ch_1 : ch_{N-2})$), path length ($pl = \sum_{i=1}^{N-1} d_{i,i+1}$), and the ratio of the overall distance to path length ($odPl = d_{i,N} \,/\, pl$). Here, $d_{i,j}$ is the Euclidean distance between p_i and p_j, v_i is the forward speed of the moving object at time t_i (or $v_i = d_{i,i+1} \,/\, (t_{i+1} - t_i)$), $h_{i,j}$ is the heading of the vector formed by connecting p_i to p_j, h_i is short for $h_{i,i+1}$, $cs_i = v_{i+1} - v_i$, $ch_i = h_{i+1} - h_i$, $mean(cs_i : cs_j) = \sum_{k=i}^{k=j} cs_k \,/\, (j - i + 1)$, and $var(cs_i : cs_j) = \sum_{k=i}^{k=j} (cs_k - mean(cs_i : cs_j))^2 \,/\, (j - i)$. The $mean(ch_i : ch_j)$ and the $var(ch_i : ch_j)$ are defined in the same manner. Among these eight features, we found *as* and *odPl* maximize the activity identification accuracy. The *as* represents how fast an object moves and *odPl* represents how straight/wiggly the movement is.

After each segment is represented by (*as*, *odPl*), we apply a clustering algorithm to all segments and each segment is assigned a cluster identifier to which it belongs. Then, each trajectory is represented as a sequence of cluster identifiers, or distinct symbols.

5.2 Markov model

A Markov model for an activity type consists of a *prior probability vector*, also called *state probability vector*, and a state transition matrix. Let us assume that there are l different activity types $A_1, \ldots, A_l$, and m distinct symbols $sym_1 \ldots sym_m$ (or m clusters). We construct a *prior probability vector* V_k, $1 \le k \le l$, of size m and an $m \times m$ matrix M_k for each activity type A_k. Here, $V_k(i)$, $1 \le i \le m$, is the prior probability of symbol sym_i, and $M_k(i, j)$, $1 \le i, j \le m$, is the transition probability from symbol sym_i to symbol sym_j in the trajectories of activity type A_k. Both probabilities can be learned from the discretized training trajectories of A_k.

Given an unknown track $tr = \langle s_0, s_1, \ldots, s_n \rangle$, where $s_i \in \{ sym_1 \ldots sym_m \}$, the likelihood that tr belongs to the activity type A_j is computed as:

$$P[A_j \mid tr] = V_j(s_0) \left(\prod_{i=0}^{n-1} M_j(s_i, s_{i+1}) \right) \qquad \text{(Eq. 1)}$$

We will refer this probability as *membership likelihood* of the trajectory tr for the activity type A_j.

5.3 Classifier model

Once Markov models are built, it is necessary to validate the models with test trajectories. For each test trajectory, we compute the likelihood of belonging to each activity type using the corresponding Markov model. First, all test trajectories needed to be segmented and each segment is assigned an appropriate symbol. Since this symbol assignment should be done consistently with the symbol assignment to training trajectory segments, we also build a classifier model from training segments. Then, we use this classifier model to classify segments in the test trajectories and assign resulting symbols to them.

6. Validation

To validate our method, we used trajectories of two sports activities – Ultimate Frisbee and volleyball. We used a video camera to record the games from the rooftop of a building on our campus while students were playing the games on the ground. We processed the images as described in Section 4 and obtained 70 Frisbee trajectories and 72 volleyball trajectories. Each trajectory was divided into equal length segments with segment length of 3 seconds. The statistics on the trajectories and their lengths (in terms of number of symbols in each trajectory) of the trajectories are shown in Table 1.

Table 1. Statistics on trajectories.

	Frisbee	*Volleyball*
# trajectories	70	72
Total # segments (or symbols)	428	646
Mean length	6.11	8.97
Std. deviation of length	3.07	2.13

Among these trajectories about one fourth were randomly chosen and set aside as test trajectories. 53 Frisbee trajectories and 54 volleyball trajectories were used as training trajectories, which included total 798 training segments. After each segment was represented by (*as*, *odPl*), a *k*-means clustering was applied to all segments with $k = 5$. The value of k was chosen in such a way that it maximizes the overall accuracy of activity identification. Figure 4 shows the result of the clustering.

Using the result of this clustering, each segment was assigned an appropriate symbol and all training trajectories were discretized. We also built a classifier model using all 798 segments. Figure 5 shows a simplified decision tree built from these segments. To make the figure simple, we did not include all branches in the tree. Some branches which have a small number of segments were pruned.

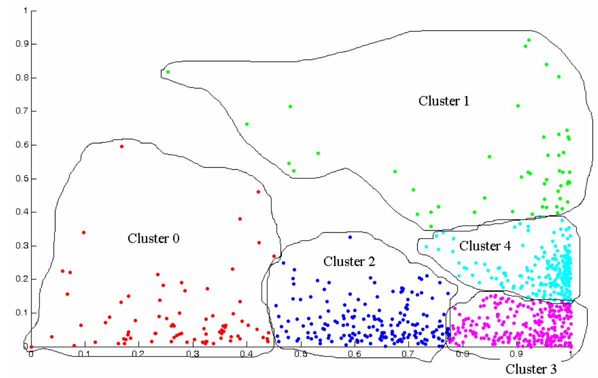

Figure 4. Segment clustering (x odPl, y as)

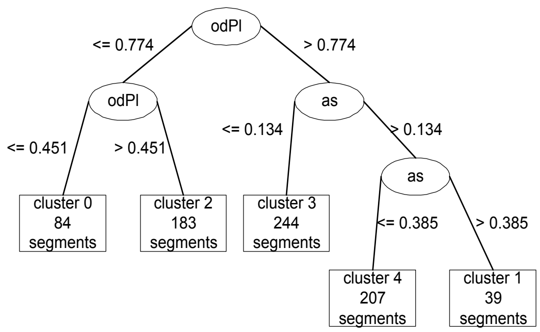

Figure 5. Decision tree.

The next step is to build Markov models. Let A_1 = Frisbee and A_2 = volleyball. The prior probability vectors and the state transition matrices are shown below.

V_1 = (0.0445, 0.1085, 0.1177, 0.1465, 0.5828)
V_2 = (0.1446, 0.0352, 0.3037, 0.4256, 0.0909)

M_1 (Frisbee)

	0	1	2	3	4
0	0.0000	0.2727	0.0909	0.2727	0.3637
1	0.0400	0.4000	0.1200	0.0400	0.4000
2	0.0312	0.0625	0.1562	0.1250	0.6251
3	0.0732	0.0244	0.0975	0.3171	0.4878
4	0.0526	0.0855	0.1316	0.1118	0.6185

M_2 (volleyball)

	0	1	2	3	4
0	0.2222	0.0318	0.3016	0.3174	0.1270
1	0.2000	0.4000	0.2667	0.0000	0.1333
2	0.1000	0.0000	0.3308	0.5000	0.0692
3	0.1778	0.0000	0.3111	0.4722	0.0389
4	0.0952	0.1667	0.231	0.2143	0.2857

Finally, we segmented test trajectories (17 Frisbee and 18 volleyball trajectories) and assigned appropriate symbols to all segments using the decision tree of Figure 5. Then, for each trajectory, we predicted which activity type the trajectory would belong to by computing the membership likelihood using Eq. 1 and compared this predicted membership with the known membership of the trajectory. Table 2 shows the results. All 17 Frisbee trajectories were correctly identified and only 2 out of 18 volleyball trajectories were incorrectly identified, and the overall accuracy was 94.2%.

Table 2. Test results.

Test Trajectories		*Classified as*		*Acc*
Activity	#traj	Frisbee	volleyball	(%)
Frisbee	17	17	0	100
Volleyball	18	2	16	88.8
Overall	**35**	**correct: 33**	**incorrect: 2**	**94.2**

We conducted another experiment to see whether prior probability vectors alone are sufficient to identify different activities. Given an unknown track $tr = <s_0, s_1, \ldots, s_n>$, the membership likelihood of tr for the activity type A_j is now computed by:

$$P[A_j \mid tr] = \prod_{i=0}^{n} V_j(s_i) \qquad \text{(Eq. 2)}$$

The result of this experiment is shown in Table 3.

Table 3. Test result with prior probability vectors.

Test trajectories		*Classified as*		*Acc*
Activity	# traj	Frisbee	volleyball	(%)
Frisbee	17	16	1	94.1
Volleyball	18	3	15	83.3
Overall	**35**	**correct: 31**	**incorrect: 4**	**88.6**

This experiment shows that using only prior probability vectors also results in a reasonably high accuracy of 88.6%. However, we can achieve higher accuracy if we also use state transition matrices. Note that there is no increase in computational complexity. An additional burden is an extra storage needed to store the transition matrices, which is negligible.

Since a decision tree divides a feature space along the lines parallel to the dimension axes, the decision tree of Figure 5 does not accurately match the result of the clustering shown in Figure 4. One possible solution to this problem is to use other types of classifiers to build a model. For example, a support vector machine (SVM) is known to split a feature space with arbitrary lines or nonlinear boundaries. So, we also conducted an experiment with different combinations of (clustering algorithm, classifier). In addition to the *k*-means algorithm, we considered EM and also two more classifiers – SVM and artificial neural network (ANN). Table 4 shows the overall activity identification accuracies of different combinations.

The result shows only the (*k*-means, SVM) combination gives us comparable accuracy as the (*k*-means, decision tree) combination. Since a decision tree classifier is more efficient (in terms or computational complexity), we chose it over SVM.

Table 4. Accuracies of different (cluster, classifier) combinations.

Clustering	*Classification*	*Accuracy*
k-means	Decision tree	94.2% (proposed method)
k-means	SVM	94.2%
k- means	ANN	88.5%
EM	Decision tree	82.8%
EM	SVM	85.7%
EM	ANN	85.7%

7. Conclusion

Identifying unknown moving object trajectories is an important but nontrivial task in many application areas. We proposed a novel method with which we can accurately identify different activity types of moving trajectories. The proposed method utilizes various data mining techniques, including segmentation, clustering, classification, and Markov model. We demonstrated the effectiveness of the method using two sports activity trajectories. We showed, through a validation process, that the accuracy of the model is as high as 94%.

In the future, we plan to test our method on trajectories of more different types of activities. We will also study how to incorporate interactions among trajectories of the same activity type into the model (or signature) of the activity type.

8. References

[1] Gaborski, R.S., Vaingankar, V.S., Chaoji, V.S., and Teredesai, A.M., “VENUS: A System for Novelty Detection in Video Streams with Learning,” *Proc. of Seventeenth International Florida Artificial Intelligence Research Society Conference*, Miami Beach, Florida, USA. AAAI Press 2004, pp. 232-237.

[2] Grady N., Flanery R. J., Donato, J., and Schryver, J., "Time Series Data Exploration," *Proc. of CODATA Euro-American Workshop*, June 1997.

[3] Hakeem, A. and Shah, M., “Multiple Agent Event Detection and Representation in Videos,” *Proc. Twentieth National Conference on Artificial Intelligence* and the *Seventeenth Innovative Applications of Artificial Intelligence Conference*, July 9-13, 2005, Pittsburgh, USA, pp. 89-94.

[4] Ivanov, Y.A., and Bobick, A. F., “Recognition of Visual Activities and Interactions by Stochastic Parsing,” *IEEE Trans. PAMI*, vol. 22, no. 8, 2000, pp. 852-871.

[5] Madabhushi, A. and Aggarwal, J. K., "A Bayesian Approach to Human Activity Recognition," *Proc. of Second IEEE Workshop on Visual Surveillance*, 1999, Fort Collins, Colorado, June 1999, pp. 25-30.

[6] Oliver, N.M., Rosario, B., and Pentland, A. P., "A Bayesian computer vision system for modeling human interactions," *IEEE Trans. PAMI*, vol. 22, no. 8, 2000, pp. 831-843.

[7] Panangadan, A., Mataric, M., and Sukhatme, G., “Detecting Anomalous Human Interactions using Laser Range-Finders,” *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vol. 3, September, 2004, pp. 2136-2141.

[8] Ray, A., “Symbolic dynamic analysis of complex systems for anomaly detection,” *Signal Processing*, vol. 84, 2004, pp. 1115-1130.

[9] Rota, N. and Thonnat, M., “Video Sequence interpretation for Visual Surveillance,” *Proc. of Third IEEE Workshop on Visual Surveillance (VS2000)*, pp. 59-67, 2000.

[10] Sbalzarini, I.F., Theriot, J., and Koumoutsakos, P., “Machine learning for Biological Trajectory Classification Applications,” *Proc. of Summer Program 2002*, Center for Turbulence Research, 2002, pp. 305-316.

[11] Stauffer, C. and Grimson, W.E.L., “learning Patterns of Activity Using Real-Time Tracking,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 8, August 2000, pp. 747-757.

[12] Witten, I.H. and Frank, E., “Data Mining: Practical Machine Learning Tools and Techniques,” Second Ed., Morgan Kaufmann, 2005.

[13] Yamato, J, Ohya, J., and Ishii, K., “Recognizing Human Action Time-Sequential Images Using Hidden Markov Model,” Proc. Conf. Computer Vision and Pattern Recognition, 1992, pp. 379-385.

[14] Zhong, H., Shi, J., and Visontai, M., “Detecting Unusual Activity in Video,” IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’04), Washington, DC, vol. 2, June 2004, pp. 819-826.