A Model of Real-Time Indoor Surveillance System using Behavior Detection

M.W. Lin and J.R. Tapamo
School of Computer Science, University of KwaZulu-Natal
Desmond Clarence Building, Durban 4041, South Africa
{limn, tapamoj}@ukzn.ac.za

Abstract. In this paper, we present a real-time surveillance system that is suitable for the indoor environment. The system is designed to detect, track and recognize the behavior of humans, using a single static camera. Background subtraction is applied to extract moving objects; these objects are tracked using linear approximation. Shadow regions are detected and removed using linear dependence and spatial connectivity properties of the shadow regions. Pattern matching and TDL (Two Dimensional Logarithmic) search approach are used to solve the problem of the occlusion of objects and depth reasoning. Behaviors of moving objects are detected by examining the sequence of shapes extracted from the scene. Shapes of moving objects are interpreted as characters of an alphabet. Each character represents a class of similar blob shapes classified using K-Means clustering. The model is used to recognize behaviors in an office with promising results.

1. Introduction

To automate surveillance solutions, a visual surveillance system using computer vision algorithms to detect actions of interest in real-time and gathering of data for events reasoning, is the current trend that is driving the shift from traditional surveillance systems that require important human resources in numbers and competencies to provide a real-time response. In general, visual surveillance systems are categorized into indoor or outdoor applications due to different environments and requirements. These requirements affect the low level implementation. On one hand outdoor environments may include the problems of unstable background, caused by situations ranging from tree waving to weather conditions like rain or snow. On the other hand indoor environments may get multi-lighting sources with shadow effects

Please use the following format when citing this chapter:
Lin, Ming-Wei, Tapamo, Jules-Raymond, 2006, in IFIP International Federation for Information Processing, Volume 204, Artificial Intelligence Applications and Innovations, eds. Maglogiannis, I., Karpouzis, K., Bramer, M., (Boston: Springer), pp. 204–211

which also increase the difficulties of surveillance.
Using computer vision to gather data and provide real-time event detection has been studied for both indoor and outdoor applications. To track objects, Masoud *et al.* [1] has built a system that is able to track and count pedestrians in real-time; this system proceeds at 3 levels: raw image, blobs and pedestrians. Blobs are obtained from raw image and one pedestrian is represented by one or many blobs; spatio-temporal coordinates of the pedestrians are recorded and tracking is done by using extended Kaman filtering. The system built by Koller *et al.* [2] extracts contour and tracks vehicles on highways using cubic splines combined with Kalman filter. Affine motion of the moving objects is also estimated using Kalman filter.

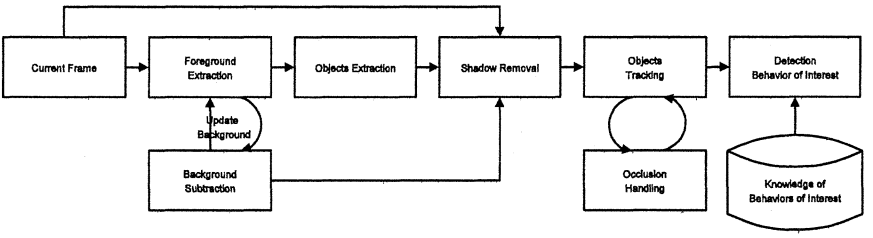


Fig. 1. Flowchart of the system.

To detect behaviors of interest, Cucchiara *et al.* [3] built an indoor system that is able to detect postures such as standing, crouching, sitting and lying by analyzing the vertical and horizontal projected histograms.
Haritaoglu *et al.* [4] and Wren *et al.* [5] designed systems that identified and tracked human using models of head, hands, feet and torso; gestures, postures and interactions between objects can be defined and detected.
The rest of the paper is organized as follows: in section 2 we describe the model of the system; some experiments are done and results are discussed in section 3; the conclusion and future work are presented in the last section.

2. System Model Overview

Foreground is extracted by taking the difference between the current frame and the background frame; background is updated using Gaussian model. Blobs are extracted by performing connected components finding. Candidate shadow regions are selected and used as a mask to filter out noising shadow from the moving objects. Each object is represented by one blob and tracked using linear approximation. TDL searching method combined with texture matching approach is used to solve the occlusion problem. Each behavior of interest is modelled as a sequence of characters; each character represents a set of shapes that has similar seven invariant moments [9]. A behavior is detected by the system if the sequence of the shapes extracted from the detected moving object is similar to a behavior of interest. The processes are detailed in following subsections. The flowchart of the system is shown in Fig. 1.

2.1 Background Subtraction

Background subtraction [6] is an approach to extract moving objects by taking the difference of the gray levels pixel by pixel of the current frame and the maintained background frame. A pixel is set as foreground if the difference of current frame and the background frame is larger than a threshold as shown in equation (1). In this system unimodal background model is used where each pixel in the background frame is modelled as a Gaussian distribution. For each pixel (x, y) of the background, two values are maintained: mean μ and variance σ. From frame to frame the background model is updated by Infinite Impulse Response filtering. Mean is the actual value of the pixel while the value of pixels in background frame is updated as in equation (2). The variance of the pixel is used to control the threshold in order to determine if a pixel from the current image belongs to the foreground or the background, it is updated as in equation (3).

$$| I_t(x,y) - B_t(x,y) | > \sigma_t(x,y) \quad (1)$$

where $I_t(x,y)$ and $B_t(x,y)$ represent the value of the pixel at (x,y) for the frame and maintained background at time t, is a real value used to adjust the system for better performance.

$$B_{t+1}(x,y) = (1-\alpha)\mu_t + \alpha| B_t(x,y) - I_t(x,y) | \quad (2)$$

$$\sigma_{t+1}(x,y) = (1-\alpha)\sigma_t + \alpha(B_t(x,y) - I_t(x,y))^2 \quad (3)$$

where is a constant used for the system to control the speed of the updated rate of the background.

2.2 Shadow Region Detection

In our system, we try to identify the behavior of the objects based on the stream of the shapes extracted from moving objects. Hence, it is important to filter out the regions of the shadow that may distort the shape of the moving objects. To detect the shadow we use an approach similar to the one proposed by Cucchiara *et al.* [8]. This method makes use of two properties of the shadows:

- *Linear dependence*- shadow regions are linear dependant on the covered background region.
- *Spatial connectivity*- shadow usually appears as a region.

For the first property, the shadow region is usually darker than the background region that is covered by it, and this can be expressed mathematically as in equation (4). For the second property of the shadow, morphology operation can be used to filter out the regions and pixels that are not large enough as a region of the shadow.

$$\frac{I_t^{(x,y)}}{B_t^{(x,y)}} = \lambda \text{, for all pixel (x,y) in the shadow region, where } \lambda \in \Re \tag{4}$$

Candidate shadow regions are used as a mask to filter out the shadow region. Shadow regions are removed by setting pixels as background if these belong both to foreground and shadow regions. Shadow candidate regions are extracted by applying algorithms as shown in equations (5), (6) and (7). Foreground with shadow removed is obtained using equation (8). Default mask size of the dilation and erosion in this system is set to 5 *by* 5. Experimental results show that shadow regions can be detected and removed. It is important to mention that shadow was not removed entirely. In most of the cases it shows sufficient performance. Some results are shown in *Fig.* 2.

$$SM1(x,y) = \begin{cases} 1, \ if \ \eta \leq \frac{I_t(x,y)}{B_t(x,y)} \leq \theta \\ 0, \ otherwise \end{cases} \tag{5}$$

$$SM2(x,y) = \begin{cases} 1, \ if \ SM1(x,y) \oplus Erosion(u,v) \\ 0, \ otherwise \end{cases} \tag{6}$$

$$SM3(x,y) = \begin{cases} 1, \ if \ SM2(x,y) \ominus Dilation(u,v) \\ 0, \ otherwise \end{cases} \tag{7}$$

$$newFG(x,y) = \begin{cases} fg(x,y), \text{ if } fg(x,y) \neq 0 \text{ AND } SM3(x,y) = 0 \\ 0, \ otherwise \end{cases} \tag{8}$$

where *newFG* is the new foreground with shadow removed, $fg(x,y)$ is the value of the pixel (x,y) of the initial foreground obtained from background subtraction.

2.3 Extraction of Moving Objects

After the foreground has been extracted using background subtraction, connected components operation is performed to merge all the foreground pixels that are connected into homogenous blobs. To avoid one object appearing in two or more separated blob components, morphology operations are performed after the connected components detection. Opening morphology operation with mask size 3 × 3 is used to remove the small noise and merge the closely situated yet separated components into one. Each blob is thereby represented as one object and after the shadow is removed, information about the blobs is then calculated. For each extracted blob, we maintain the following attributes: *Center*, *Area*, *Bounding Box*, *Density*, *Velocity*, and the *seven Invariant Moments*.

2.4 Objects Tracking

A linear approximation approach is used to track detected moving objects. For each frame, blobs from the previous frame are matched to the blobs extracted from the current frame. Blobs from the previous frame are first shifted to the estimated new location on the current frame; after the shifting of blobs to the estimated location, if the bounding box of the blob B^i_{t-1} from the previous frame t-1 overlaps the bounding box of the blob B^j_t from the current frame t, it is then a potential match for the blob B^i_{t-1} to the blob B^j_t. We refer to blob B^i_{t-1} as the parent of the blob B^j_t and blob B^j_t as the child of the blob B^i_{t-1}. For each blob from the previous frame, a new location is estimated by shifting the location of the blob by the velocity of the blob. The velocity of the blob is a linear approximation by taking the displacement of the blob between its current location and its location in the previous frame. As a result of using such parent-child relationship, five different following conditions could be encountered:

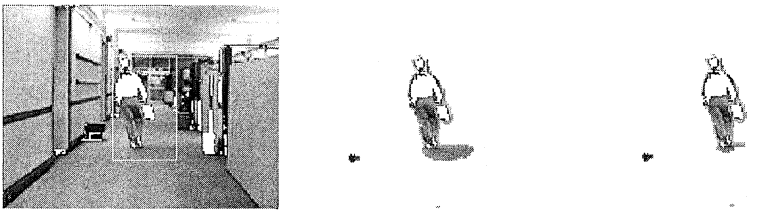
1) 1 parent blob matched to 1 child blob
2) 1 parent blob matched to 0 child blob
3) 0 parent blob matched to 1 child blob
4) 1 parent blob matched to more than 1 child blob
5) more than 1 parent blob matched to 1 child blob

For 1) the blob is traced successfully from the previous frame to the current frame, the same object still appears in the scene with a predicable movement; if a parent blob ends up with no child blob as in 2), a search is performed around the estimated location, if there are blobs around the search region, a parent-child relationship is assigned to these blobs, otherwise the object is assumed to have exited from the surveillance scene and tracking of this object is terminated; if a blob from the current frame has no parent blob as in 3), it is marked as a new object, new tracking is initiated for it; if a parent blob has more than 1 child blob, similarity function that makes use of the information of size, location and density of the blob is measured, the parent blob will retain the child relationship to the most similar child blob, parent-child relationship to all other child blobs with lower similarity is unset. If more than 1 parent blob assigned child relation to the same blob, object occlusion has occurred; details for solving the occlusion are described in the next section.

2.5 Occlusion Handling

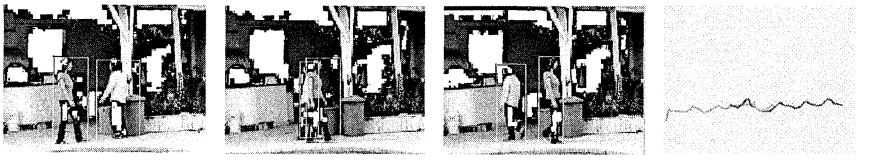
Occlusion occurs when two or more parent blobs are referenced to the same child blob. In this system, occlusions are solved by estimating new location for the parent blobs that give the maximum coverage rate to the occluded child blob and texture matching is used to find depth order of the parent blobs. To estimate the location of the parent blobs, the TDL (Two Dimensional Logarithmic) searching approach combined with the hit function is used. The best location for parent blobs is the location that maximizes the sum of the hit functions. The TDL searching approach is a suboptimal searching solution proposed by Jain *et al.* [7]; it gives efficient computation time with sufficient accuracy. Hit function is the function that returns the number of pixels that the parent blob covers to the occluded child blob after the

shifting of the parent blob with TDL searching displacement. The initial TDL search location is the position plus the velocity of the parent blob. Once the best location for each of the parent blobs is estimated, texture matching is then performed to solve the depth order of the occluded objects. The shape and texture of parent blobs from the previous frame shifted with the best estimated displacement found using TDL earlier on is used to match the texture of the occluded blob. The parent blob that gives the lowest cost will be the object that is closest to the camera. The MAD (Mean Absolute Difference) is used as the distance function for texture matching. Experiments conducted using pattern matching to solve short time occlusion give promising result, see Fig. 3.



a) b) c)

Fig. 2. Figure a) shows original hall monitor frame at index 70, b) and c) show detected and extracted moving objects from this system without and with shadow detection.



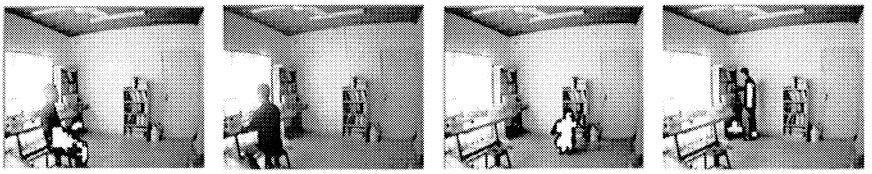
a) b) c) d)

Fig. 3. Figures a) b) c) show an example of the occlusion event, d) shows trajectories of the detected moving objects.

2.6 Characters and Behaviors Learning

Behaviors of interest are modelled as streams of characters and saved in a knowledge base. Each character represents a set of shapes that share 7 similar invariant moment features. Learning of the characters is by extracting 7 invariant moments from the shapes of the detected moving objects from a training data set. Based on 7 invariant moments, K-Means clustering is then performed to separate these shapes into K different classes; each class is then represented by a unique character. The centroid value for each class is then the features of the character. After the character learning, each character is considered as a unique pattern. Invariant moments are appropriate to model shapes of the moving objects because they conserve invariance by translation, rotation and scaling of objects. More detail about invariant moments can be found in [9, 10]. This set of characters is then used to represent any possible shape of the moving objects that the system can detect.

The learning of behaviors of interest is done by extracting each behavior of interest from a sequence of frames. From each frame the extracted shape of the blob is then assigned to a character by comparing 7 invariant moments of the blob to the characters. The nearest Euclidian distance is used to find the best matched character. Each behavior of interest is then represented as a stream of characters.



a) b) c) d)

Fig. 4. Figures show four different behaviors of interest the system aims to detect.

2.7 Detecting Behaviors of Interest

To detect behaviors of interest, the shape of the detected moving object is assigned to a character. Euclidian distance is used to find the nearest character amongst the character set and the shape of the object. For an object that is traced by the system, a stream of characters representing a sequence of the shape captured from the system will be maintained. If the end piece of substring from the traced object and the string of the behavior of interest are same, then the behavior of the object is assumed to be the behavior of interest.

3. Experiments

The experiment is done by monitoring a study room with a chair, a table and two bookshelves. There are four behaviors of interest for the system to monitor wherein a human is positioned as follows: seated facing away from the table, seated facing the table, touching bookshelf-1, touching bookshelf-2. Fig. 4 shows four different behaviors which the system aims to detect. A character set is obtained by first collecting 862 different shapes of the human posture in the room environment; these shapes are then clustered into 29 different classes, centroid values for each class are then represented as a unique character from A to Z and @,#,$.

The capture speed of the system is set to 5 frames per second, each behavior of interest is represented by 5 characters, by doing this, the system examines the behavior of moving objects by matching 5 key characters to the 5 shapes of the detected moving objects extracted in the last 1 second.

With just one training sequence for each behavior of interest, the system is able to detect 4 distinct behaviors of interest; each behavior of interest gives unique character streams: seated facing away from the table - AAAAA, seated facing the table - HHHHH, touching bookshelf-1 - NNNNN, touching bookshelf-2 - VVVVV. Through the 50 testing behaviors within about 3200 frame sequences: results show 36 behaviors recognized correctly in an appropriate time, the system missed detection of 11 behaviors and 3 behaviors are detected as wrong behavior of interest.

4. Future Work and Conclusions

We have designed and presented a model of real-time surveillance system. For our future works, we intend to explore further this modelling approach of behavior of moving object using character representation; character modelling is suitable for the movements that require regular posture and cadence as in aerobic exercise and Chinese martial art. This approach combined with Edit Distance has been used successfully to model and detect the movements of Chinese martial art. We also want to interpret the behavior of the detected human at higher levels by combining several short time behaviors.

Acknowledgments

The authors would like to thank the National Research Foundation of South Africa for their support of this work. The authors would like also thank to Cathy and Mei-Zhu for making available testing data.

References

1. O. Masoud and N.ikolaos P. Papanikolopoulos, "A Novel Method for Tracking and Counting Pedestrains in Real-Time Using a Single Camera," IEEE Trans. On Vehicular Tech. Vol. 50, No. 5 Sep 2001.
2. D. Koller, J. Weber and J. Malik, "Robust Multiple Car Tracking with Occlusion Reasoning," In Proceedings of Third European Conference on Computer Vision, Stockholm, Sweden, May 2-6, 1994, pp. 189-196, LNCS 800, Springer-Verlag, 1994.
3. R. Cucchiara, C. Grana, A. Prati and R. Vezzani, "Computer vision system for in-house video surveillance," In IEE Proceedings of Visual Image Signal Process. Vol. 152, No. 2, April 2005.
4. I. Haritaoglu, Larry S. Davis and D. Harwood. w4 who? when? where? what? a real time system for detecing and tracking people," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol.22 Aug 2000.
5. C. Wren, A. Azarbayejani, T. Darrell and A. Pentland, "Pfinder: Real-time Tracking of the human body," IEEE Trans. on Pattern Analysis and Machine Intell., 19(7):780-785, 1997.
6. S.Y. Chien, S.Y. Ma and L.G. Chen, "Efficient Moving Object Segmentation Algorithm Using Background Registration Technique," IEEE Trans. on Circuits and Systems for Video Tech. Vol. 12 No. 7 July 2002.
7. Jaswant R. Jain, Anil K. Jain, "Displacement measurement and its application in interframe image coding," IEEE Transactions on Communications, Volume COM-29, Number 12, p 1799 - 1808, December 1981
8. R. Cucchiara, C. Grana, M. Piccardi, A. Prati and S. Sirotti, "Detecting Moving Objects, Ghosts, and Shadows in Video Streams," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 25, pp. 1337-1342, 2003.
9. M-K. Hu, "Visual pattern recognition by moment invariants," IRE Trans. on Information Theory, IT-8:pp. 179-187, 1962.
10. A. Khotanzad and Y. H. Hongs, "Invariant image recognition by Zernike moments," IEEE Trans. on Pattern Analysis and Machine Intelligence, 12(5):pp. 489-497, 1990