Vol. 1, No. 1 (2013) 11-19

JOURNAL of COMPUTER SCIENCE
journals.sjp.ac.lk/jcs

Dynamic Object Path Detection in a Network of Surveillance Cameras

H.H. Weerasena, P. B. S. Bandara, J. R. B. Kulasekara, B. M. B. Dassanayake, U. A. A. Niroshika[1] and P. R. Wijenayake

Department of Information Technology, Sri Lanka Institute of Information Technology, Malabe, Sri Lanka

Corresponding Email: [1]aruni.n@sliit.lk

Abstract

Today, automated camera surveillance systems play a major role in securing public and private premises to ensure security and to reduce crime by detecting behavioral changes of moving objects. The important goal of such a surveillance system is to reduce human intervention while at the same time, provide accurate detection of moving objects. Many researchers have attempted to automate different aspects of camera surveillance such as tracking humans, traffic controlling, ground surveillance, etc. However, a system that overcomes overall difficulties that arise in the task of object detection and object tracking has not been developed because of high variance in the problem domain. The proposed system tracks the path of a locked object through a network of cameras. In contrast to traditional methods where the operators have to switch the screens manually to find the target objects, the proposed technique, once locked to an object; automatically tracks it through a camera network and generates the path on a map. We propose to use stereo cameras to enhance the detection and tracking of objects in 3D space.

Keywords: Electronic surveillance, object tracking, path detection

1. INTRODUCTION

In present day, automated camera surveillance systems play a major role in securing public/private premises to ensure security, reduce crimes and detect object behaviors [1]. The reduction of human intervention in the surveillance system and real-time object tracking are mainly requested from a surveillance system nowadays.

The task of object tracking over a camera network takes several routines to process. When an object moves out of the view of one camera, it is expected to appear in front of a different camera at the very next moment. When the object appears in the second camera, it should be identified correctly without any errors and it is a tedious and resource intensive task to be performed in a computer. In order to accomplish the task of tracking objects over multiple cameras, it splits the task into two subtasks. The first subtask, which is described in this paper, is the tracking of objects on one camera that will produce features and information about the appearance of an object. These extracted features can then be used to accomplish the second subtask; matching the object on a camera with images observed previously on another camera [2]. The accuracy depends on how many features are extracted and complexity of the method used to match features of interest. For a better result, a good combination of features and a specific matching model is required.

As explained above, as matching of appearance involves high cost in terms of computing resource, this research focuses on finding ways of improving the performance of feature based surveillance systems. Two main methodologies identified to achieve this objective were

increasing hardware capacity and the use of strategies and methods that reduces confounded matching. From these alternatives, we found out that the latter is more successful than the former. The scope narrows down to finding and evaluating possible strategies and methods for the reduction of unnecessary feature matching while maintaining the accuracy of object tracking.

The main objective of this research is to identify software based solutions to increase performance of feature based object tracking in a non-overlapping network of cameras. We could summarize the specific objectives of this research as follows:

1. Identify strategies or methods for the reduction of unnecessary feature matching and improving the accuracy of object tracking.
2. Evaluate identified strategies and methods for response time and accuracy.

2. SIGNIFICANCE OF THE RESEARCH

Today, surveillance systems are used everywhere and cost effective solutions are in great demand. Video processing surveillance systems need powerful hardware and it increases correspondently when number of resources increase. Introducing strategies or methods to increase the efficiency of software reduces unnecessary usage of hardware and enables us to handle the resources effectively. Response time, which is a critical requirement in a real-time system, greatly depends on the performance of the system. Using a hardware solution to increase the performance is costly, limited and is a waste of resources when optimizing is possible. This research focuses on identifying a software based solution that increases the system performance.

3. BACKGROUND

Teller and Antone [3] used a camera adjacency graph to calibrate hundreds of still omni-directional cameras in the MIT City project. However, this adjacency graph was obtained from a priori knowledge of the cameras' approximate location acquired by a GPS sensor instead of estimating the location from the images themselves.

Yunyoung et al. [4] proposed the topology of an arbitrary camera network by considering the spatio-temporal relationship between cameras which was used to support predictive tracking across the camera network. They have used the entry-exit and travel-transition time model for spatial relationship and temporal relationship, respectively.

Kumar et al. [5] described a 3D surveillance system using multiple cameras surrounding a particular scene where the application is concerned about identifying humans in the scene and identifying their postures. The cameras used are fully calibrated and assumed to remain fixed in their positions where the detection of objects and interpretation are performed completely in 3D space. Using depth information, a person can easily be separated from the background as the system is mainly concerned with the foreground detection and object identification. They have used multiple stereo camera pairs surrounding the scene to create a 3D reconstruction of the scene. The cameras were fully calibrated, meaning their intrinsic as well as extrinsic parameters are determined a priori. After performing correspondence matching between two views, the different stereo pairs generate 3D points in the scene. Subsequently, these points are clustered into background and foreground where the foreground clusters are compared with a 3D human model to detect the presence and posture of a human in the scene. Their proposed system is more similar to our system which can more effectively handle occlusion using the depth and distance. Hietbrink [6] had carried out a research regarding feature-based visual object tracking system to gather information about objects in order to identify them in a multiple-camera environment. He has created a tracking method using stationary cameras and an EM-based background subtraction system to which a distance is added to avoid the "growing together" of the background kernels. It was proved that background subtraction can be effectively used as object detection and feature extraction can be a better solution if implemented correctly in tracking objects.

As mentioned above, although many researchers have carried out for object tracking and re-identification across multiple non-overlapping cameras, this research is unique as it uses software based performance increasing strategies to address the problem of unnecessary feature matching instead of utilizing expensive hardware.

4. MATERIALS AND METHODS

The current methods in the research area are enhanced with slight modifications to improve the performance of the system. Since the scope is narrowed down to object detection and tracking, it mainly focuses on the above point.

A. Motion Detection

Background subtraction is a critical step for motion detection of a moving object and there are many algorithms for motion detection in a continuous video stream. When the camera is stationary, almost all the available literature are based on comparing the current video frame with the previous frame or with a feature referred to as background subtraction [7]. It can be mathematically expressed as

$$|frame_{\ i} - frame_{\ i+1}| > TH$$

where, $frame_i$ = current i^{th} frame and TH = a threshold.

Several requirements should be taken into account during gradual or rapid changes in illumination of objects, motion changes such as camera oscillations, changes in high-frequency background objects (highly crowded area), and changes in the background geometry. We have selected the Gaussian mixture model among the available background subtraction models.

B. Object Depth

In this research, stereo cameras were used to get 3D information of objects such as the depths. The distance between the lenses in a typical stereo camera (the intra-axial distance) is approximately the distance between a person's eyes (known as the intra-ocular distance) which is about 6.35cm [8].

A depth map method is used to identify the relative depth of objects where objects placed at different distances are identified by the intensity difference of objects in the depth map. Objects located at as short distance exhibit a higher intensity (higher grey level) and objects located at a far distance show a lower intensity (lower grey level) in the depth map. This concept is very useful when tracking similar color objects placed at different distances and used only when multiple objects are detected. Figures 1, 2 and 3 show objects placed at different distances and how the intensity varies based on this distance.

(a) (b) (c)

FIGURE 1. Object located at 5 feet distance from the camera. (a) left camera, (b) right camera and (c) depth map,

(a) (b) (c)

FIGURE 2. Object located at 10 feet distance from the camera. (a) left camera, (b) right camera and (c) depth map.

(a) (b) (c)

FIGURE 3. Object located at 15 feet distance from the camera. (a) left camera, (b) right camera and (c) depth map.

C. Feature Extraction and Matching

Features (or properties) are used to uniquely identify an object in the camera network. The color feature is used to track objects because it is the most widely accepted feature for object recognition systems in the research community for its robustness towards size and orientation changes. Possible color features are; color templates, histograms, moments, signatures (dominant colors), and portative color layouts. In a multi-camera setting, there is a high probability of occurrences of illumination, camera distortion, and object resolution differences. Therefore, the color feature should be able to reduce the inter-camera distortions as well as illumination changes.

The color feature of an object is represented as a two dimensional histogram. The color model is converted from RGB to HSV before calculating the histogram. The accuracy of the histogram calculation was improved by using the background mask generated through motion detection while features were extracted only from selected moving region.

In order to identify an object, the color histogram of the desired human object is stored as a template [9] which is then used to track the targeted object across the camera network. When an object is selected to be tracked from one camera, the system starts to detect objects, extract features and generate color histograms. Subsequently, they are matched with the template histogram where an object is re-identified by the comparison percentage. This appearance matching is very inefficient because the system has to match the template with histograms generated from every other camera. The next section shows how we can avoid such inefficiencies.

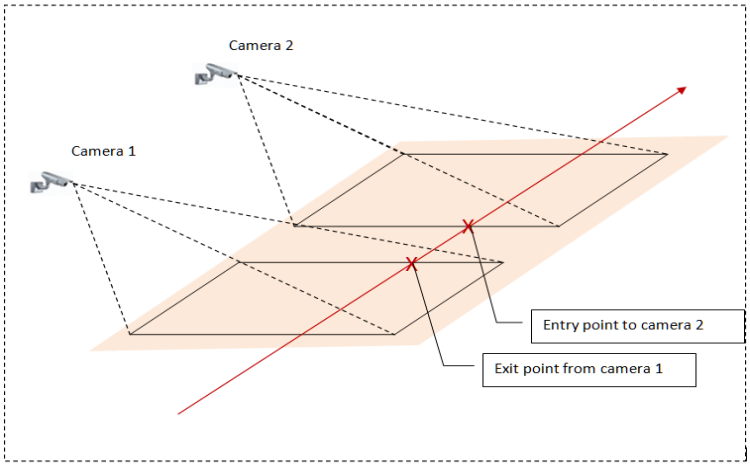


FIGURE 4. Entry and exit points of objects.

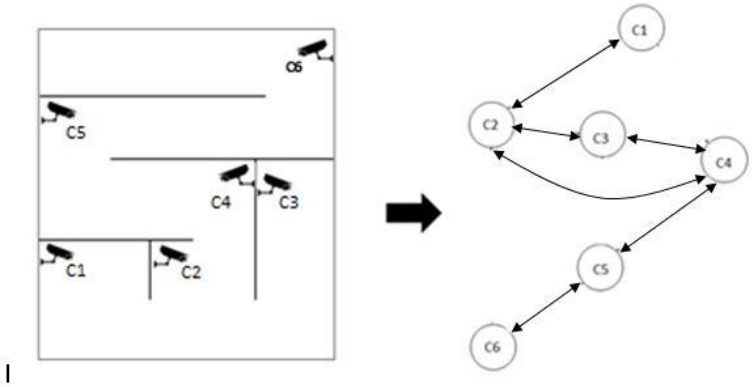


FIGURE 5. Topology of camera placement as a directed graph.

D. Camera Network Topology

Humans follow regular paths when walking because people tend to follow the same paths in most cases, i.e., roads, walkways, corridors etc. Fig. 4 shows such an example of a path of a common object laid across Camera 1 and Camera 2 which are adjacent. Each camera in the network is represented as a node in a directed graph where the edges represent the possible physical routes between the cameras as in Fig. 5.

Thus, if an object exits from camera 1 (C1), the system only needs to alert camera 2 (C2) reducing unnecessary costs in alerting all cameras connected to the system. Another way of reduce processing task is when tracking an object moving from one camera viewpoint to other viewpoint, to take its exit region and estimate possible entry regions to other cameras on the network using an adjacency matrix. Then, the system only has to match features with objects that will enter to camera viewpoints in the estimated regions above. Fig. 6 shows the possible scenario of the fact. Assume that the object to be tracked is Object 4 and it will exit Camera 1 via the highlighted region. After the object exits, the system will find the next possible camera, that is Camera 2, and possible entry regions. Now, the system has to match the features of objects that enter only through the calculated region. By using this method, computational resources can be reduced by matching features of interests with other objects such as Object 5, 6 and 7.

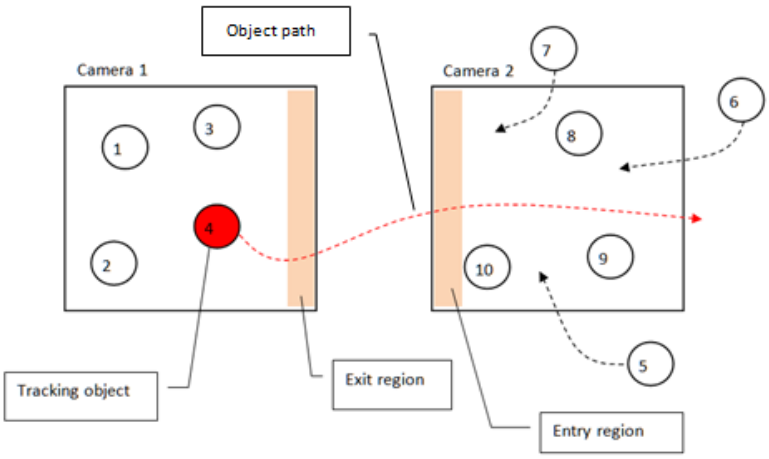


FIGURE 6. Exit and entry of objects placed within two specific regions covered by two cameras.

5. RESULTS AND DISCUSSION

To evaluate the performance of the system, experiments were performed with three stereo camera units (320 x 240 resolutions) and computer with Core 2 Duo Processor and 2GB of RAM as the hardware platform, and Microsoft SQL Server 2008 as the underlying database. In addition, the system was implemented using the C++ and OpenCV library [10] for image processing.

The first experiment for the camera topology was done with three web cameras in three rooms as shown in Fig. 7.

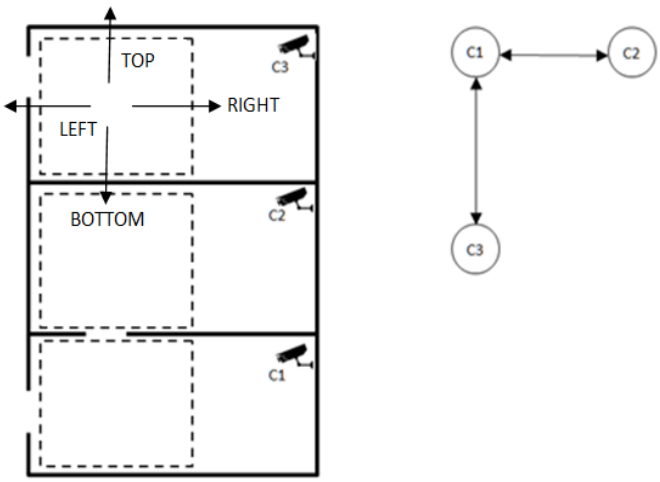
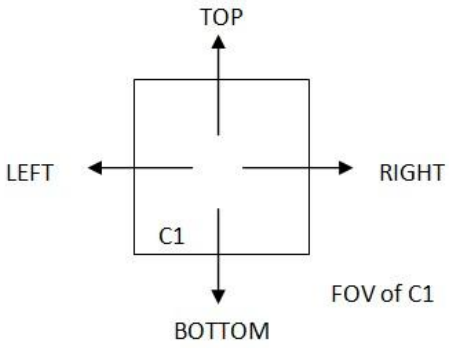


FIGURE 7. Camera topology for experiment 1.



	T	B	L	R
C1	C2	0	C3	0
C2	0	C1	0	0
C3	0	0	C1	0

FIGURE 8. Field of view (FOV) of camera 1 (C1) and its djacency matrix with other two cameras C2 and C3.

The adjacency matrix used to determine correspondences between cameras is shown in Fig. 8. If an object disappears from the field of view (FOV) of C1, the system has to process both C2 and C3. If an object disappears from C2, the system only processes C1 and if an object disappears from C3, the system only processes C1. Also, if an object exits from C1, the system is able to decide which camera to process (C2 or C3) by considering the exit region. The performance graphs in Fig. 9 shows the differences in frame rates based on the topology used. Without the topology (red line), the system exhibits only 3–5 frames per second (fps) and when the camera topology is used (blue line), it increases to up to 11–13fps.

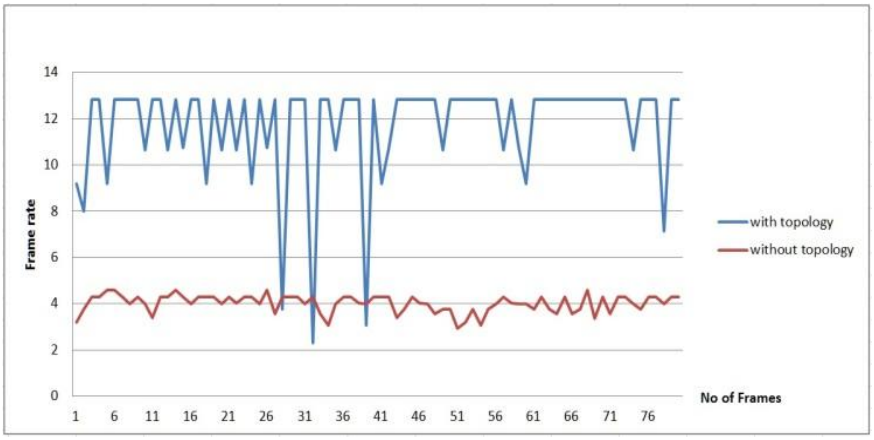
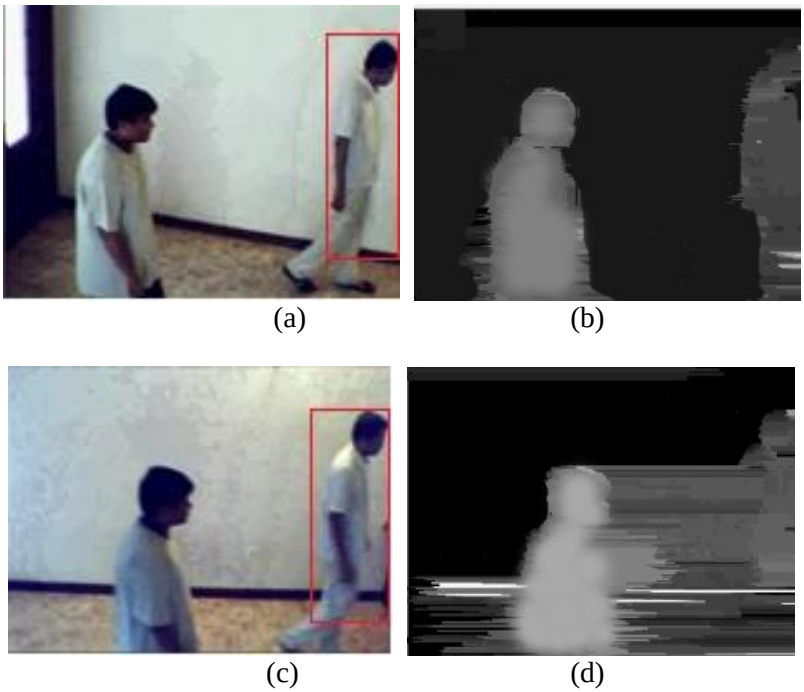


FIGURE 9. Performance graph based on camera topology.

The next experiment on the use of object distance to track similar color objects was done with two linear stereo camera units with two moving objects.



(a) (b)

(c) (d)

FIGURE 10. Process of distinguishing between similar color objects. (a) source and (b) depth map of camera 1 and (c) source and (d) depth map of camera 2.

Although objects are similar in color, the first object is tracked and the system distinguishes it by the intensity difference in the depth map as depicted in Fig. 10. It was observed that the minimum distance between two objects to be distinguished by the system accurately was 2 feet.

6. CONCLUTION

In this paper, a method to improve the performance of a feature based surveillance system is introduced. The expected performance can be defined as tracking performance (accuracy) or as system efficiency. In order to improve the accuracy in tracking an object, the relative object depth was used to distinguish a desired object from other objects without using only the color parameter. At the next step, a technique is proposed to determine which camera input should be processed when an object leaves from a particular camera by providing the topology to the system. The research can be extended to track multiple objects simultaneously and to identify possible strategies to increase the performance without compromising the accuracy.

7. REFERENCES

[1] Wikipedia, "Surveillance.", 3rd February 2009. [Online]. Available: http://en.wikipedia.org/wiki/Surveillance Accessed: 27th February 2011.

[2] W. Hu, T. Tan, L. Wang, and S. Maybank, "A Survey on Visual Surveillance of Object Motion and Behaviors", IEEE Transactions on systems, man, and cybernetics - Part C: Applications and Reviews, vol. 34, no. 3, 2004.

[3] S. Teller and M. Antone. "Scalable extrinsic calibration of omni-directional image networks", International Journal of Computer Vision, vol. 49, no. 2, pp. 143–174, 2002.

[4] Y. Nam, J. Ryu, Y. Choi, and W.D. Cho. "Learning Spatio-Temporal Topology of a Multi-Camera Network by Tracking Multiple People", Proceedings of World Academy of Science, Engineering and Technology, vol.24, pp. 178, 2007.

[5] A.K. Mishra, B. Ni, S. Winkler and A. Kassim. “3D Surveillance System Using Multiple Cameras”, Proc. SPIE 6451, Videometrics IX, 2007.
[6] N.P. Hietbrink. “Visitracker: Feature extraction by single camera tracking for camera to camera matching”, Thesis, University of Amsterdam, 2002.
[7] M.A. Marzouk, "Modified background subtraction algorith for motion detection in surveillance systems", Journal of American Arabic Academy for Sciences and Technology, vol. 1, no. 2, pp. 112-123, 2010.
[8] T.K.S. Cheung. “Stereo Computer Vision with Multiple Cameras”, Hong King University of Science and Technology, 2009.
[9] D.R. Corbett, “Multiple Object Tracking in Real-Time”, BEng Thesis, University of Queensland, 2000.
[10] Gary Bradski, Adrian Kaehler, Learning OpenCV, O’Reilly, pp 377.