Multi-camera Video Surveillance

Tim Ellis
Information Engineering Centre
School of Engineering
City University, London
t.j.ellis@city.ac.uk

Abstract

This paper describes the development of a multi-view video surveillance and the algorithms to detect and track objects (generally low densities of pedestrians, cyclists and motor vehicles) moving through an outdoor environment imaged by a network of video surveillance cameras. The system is designed to adapt to the widely varying illumination conditions present in such outdoor scenes, as well as coping with the spurious motion of non-objects (such as vegetation) and the interaction of objects within the scene. Where possible, the system takes advantage of multiple views of the same object to help resolve occlusion and increase the robustness of the tracking. Overlapping camera views are corresponded using a geometric analysis of camera viewpoints. The system aims to capture scene dependent information through learning, constructing models of the scene using observations extracted from the camera network. The system is currently undergoing a real-time implementation using live camera data, and results will be presented from this on-line system.

1. Introduction

Video surveillance systems typically monitor an environment with multiple video cameras, using the perceptual capabilities of a human operator to observe (detect and identify) objects moving within the viewfield of the cameras. Automating this process requires the development of image processing and computer vision algorithms to detect, locate and follow a target (generally either a pedestrian or a vehicle) as it moves through the environment. Previous work focussed on the identification of objects in single camera views [1,2,12]. In this paper the emphasis is on robustly tracking multiple targets through complex environments viewed by multiple cameras, where the video analysis must cope with a variety of disturbing conditions resulting from significant variations of illumination and a range of weather conditions.

In complex and cluttered environments with even moderate numbers of moving objects (e.g. 10-20) the problem of tracking individual objects is significantly complicated by occlusions, where an object may be partially occluded or totally disappear from camera view for both short or extended periods of time. Static occlusion results from objects moving behind (with respect to the camera) fixed elements in the scene (e.g. walls, bushes), whilst dynamic occlusion occurs as a result of moving objects in the scene occluding each other, where targets may merge or separate (e.g. a group of people walking together).

Multiple cameras are exploited in two different ways. Firstly, spatially adjacent cameras extend the coverage of the video surveillance across the spaces that are being viewed. The non-visible spaces between adjacent views can be treated as a form static of occlusion, and analysed in a manner similar to single view occlusion analysis. Secondly, overlapping views are used to provide selective observation of specific parts of the scene (for example, a doorway or entrance) and to deliver an element of redundancy into the camera network. The information generated from overlapping views can be used to minimise the ambiguities of occlusion, as well as improving the accuracy of the position estimate of the object [4].

A current trend is for the processing power to gravitate into the sensors, providing a self-contained intelligent video sensor capable of generating meaningful low-bandwidth symbolic output rather then pixels [7]. Installations of such intelligent sensors, linked by digital communication channels, provide a distributed system that can cooperatively function to construct an integrated and scaleable sensor network.

This paper considers some of the key computer vision tasks necessary to build an automated video surveillance system. Section 2 considers the architecture of the system. Section 3 briefly describes the algorithms developed to deliver robust tracking of moving objects, principally in a man-made outdoor environment. Section 4 concludes with a discussion of the operation of the system.

0-7803-7436-3/02/$17.00 ©2002 IEEE

2. Intelligent Camera Network

The systems architecture for the video surveillance system consists of a network of 'intelligent' cameras, comprising a sensor, video acquisition, image processor and network interface hardware. Moving targets are detected (independently) within the field of view of each camera and tracked using image features based on shape, motion and colour information [3] (see section 3).

In order to integrate the track data from multiple cameras, it is useful to consider the visibility of targets within the entire environment, and not just each camera view separately. Four region visibility criteria can be identified:

1. **visible FOV** (field-of-view) - this defines the regions that an individual camera will image. In cases where the camera view extends to the horizon, a practical limit on the view range is imposed by the finite spatial resolution of the camera
2. **camera FOV** - encompasses all the regions within the camera view, including occluded regions
3. **network FOV** - encompasses the visible FOV's of all the cameras in the network. Where a region is occluded in one camera's visible FOV, it may be observable within another FOV.
4. **virtual FOV** - covers the network FOV and all spaces in between the camera FOV's within which the target must exist. The "boundaries" of the system represent locations from which previously unseen targets can enter the network.

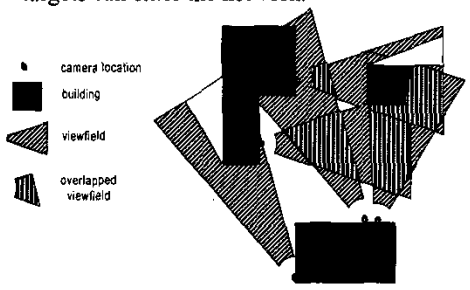

Figure 1. Depiction of five camera viewfields in a simplified environment. Non-visible regions are white.

Co-ordinating the tracking of objects over spatially adjacent and overlapping camera views could be addressed in two ways: a central monitor station is used to correspond and integrate the information from each camera; alternatively, camera stations operate autonomously, exchanging information directly with other camera stations, providing a handover mechanism when a target transits from one camera view to another. The first approach simplifies the management and communications required at each camera station, as all tracks are dispatched to a single source. However, such an approach does not scale to large camera networks (e.g. >50 cameras), as the central station becomes overloaded with increasing amounts of track data to manage and process. In the latter case, significant issues relate to developing a mechanism for controlling the handover and establishing how responsibilities are allocated for a given camera to track a particular target that is seen by several cameras simultaneously.

The system can display both full-frame video or just frame fragments (referred to as framelets) that correspond to regions of the image where motion has been detected and tracked. Video feed from the tracking algorithm, and 'live' feed of the video data from any camera can be sourced to a networked monitoring station. To reduce the bandwidth requirements, the video data is compressed using a MPEG-like video encoding algorithm, taking advantage of the motion detection and tracking algorithm to identify image segments containing the target. These fragments can then be individually encoded (using JPEG) and transmitted to the video monitor station where they are painted onto a background image of the scene (see figure 2).

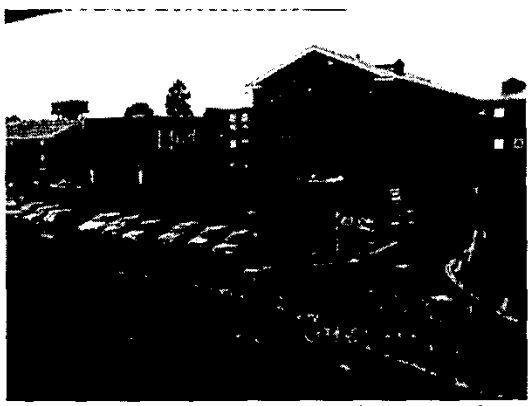

Figure 2. Set of framelets generated over 50 frames painted onto the background frame (PETS2001 sequence).

The image fragments are packed into a data packet which contains header information identifying the frame packet type, camera station, object identifier, image location coordinates (x,y), timestamp plus the JPEG encoded pixel data. At the monitor, the timestamp is used to determine the relevance of this data to the currently displayed frame. Old fragments may be discarded (or archived to disk). Current fragments are drawn into the display screen, at the coordinates indicated by the x,y location. Full-frame JPEG images (usually the background reference image) are occasionally sent by the camera to the monitor station, if significant changes in the background are detected at the camera.

In addition to the conventional display of video frames, the system can display track data projected into a 3D ground plane map of the environment (see figure 6).

To support robust tracking across the virtual FOV, the system maintains a record of each target entering the system and throughout its duration. When a target disappears from any camera FOV, motion prediction, colour identification, and learnt route patterns are used to re-establish tracking when the target reappears. Each target is maintained as a persistent object within the active database and spatial and temporal reasoning are used to detect these activities and ensure that entries are not retained for indefinite periods. Such situations (i.e. targets 'disappearing') are in themselves potential alarm triggers to the system. We use a probabalistic approach to learning route patterns, building activity maps that represent the expectations on target trajectories (see section 2.4).

3. Vision analysis

This section considers the image analysis algorithms that have been developed for video surveillance. The first sub-section examines the problem of detecting potential regions in the image where motion might be occurring. The second sub-section employs spatio-temporal reasoning to correspond observations over time, either for a single camera view, or across multiple views. Alternative methods of occlusion reasoning are used to minimise the likelihood of losing tracking, in the situation where the scene is viewed by only a single camera, or from multiple viewpoints. The last sub-section employs the detected trajectories to build spatial models of the typical routes followed by object motions.

2.1 Motion detection

Frame differencing provides a highly sensitive algorithm for detecting intensity changes between video frames when the camera is static. Differencing is applied between the current camera frame and a reference or background image. This reference image must be continuously adapted to ensure that non-motion related intensity changes do not trigger the object detection process. For example, fast illumination changes due to a flood of sunlight, self-shadows, the switching of artificial lighting or slow changes associated with the natural diurnal cycle. Other types of disturbing effects include motion due to wind-generated movement of vegetation.

To adapt to changes in the background, the probability of observing a given pixel value is modelled by a mixture of Gaussians [13], using both colour and monochrome pixel values [8,14]. The mixture model assumes multiple background values at each pixel, as might occur with wind moving leaves in a tree, periodically covering and un-covering part of the more distant background. A colour representation based on (normalised) chromaticity coordinates is employed, which also helps to suppress shadows in the images. This combination approach overcomes some of the problems associated with using colour in low-light conditions, where the monochromatic information can be more reliable and noise-tolerant.

The frame-differenced pixels are binarised using an adaptive threshold, computed from the variance of the foreground pixels. These detected pixels are clustered into foreground "blobs" using connected component analysis. Blobs considered originating from noise are rejected on the basis of a minimum area threshold and a lack of temporal consistency. Figure 3 shows an example of foreground detection during a minor illumination change. The spurious foreground regions detected at the top-left corner of the intensity-based result (figure 3b) are caused by an illumination change occurring at this frame, but are eliminated from the chromaticity-plus-intensity result (figure 3c).

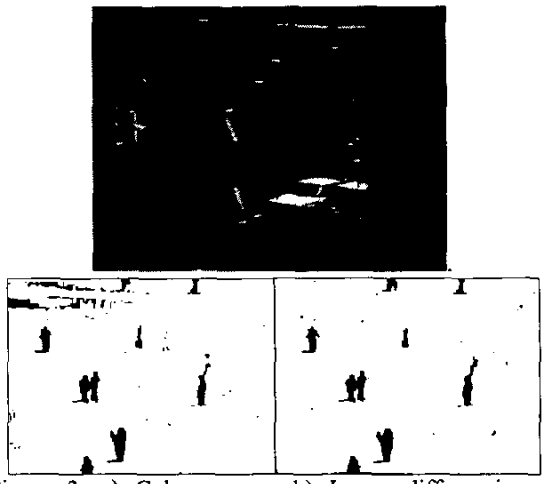

Figure 3. a) Colour mage. b) Image differencing of monochromatic channel only. c) Improved image differencing using combined chromaticity and monochromatic channels.

2.2 Single camera track detection

Following connectivity analysis each blob is represented by a measurement vector composed of the following features: centroid coordinates, height, width and mean chromaticity. The blob is represented in the output video data by a rectangular bounding box (see figure 4).

The tracking process attempts to match each blob with the 'best' candidate from the set of object trajectories constructed over previous frames. If this match is made, the object trajectory is updated with the blob's measurement vector. A Kalman filter is used to estimate

and predict the state of each object from one frame to the next. Blobs are matched to existing object tracks using a weighted comparison between optimum pairings of the blob measurement vector and an object's predicted state.

However, because of occlusions (either static or dynamic), or because the target has only just entered the camera FOV, it may not be possible to determine a match. Un-matched blobs and object trajectories may be ascribed to one of the following object categories:

1. NEW for objects entering the scene
2. TERMINATED for objects leaving the scene or being inactive for a long time
3. UPDATED for normally tracked objects
4. MERGED for colliding objects
5. SPLIT for new objects separating from existing objects
6. OCCLUDED for objects behind static occlusions
7. MISSING for objects lost in detection.

If the prediction indicates that the trajectories of two objects are likely to intersect, or that an object may intersect a static occlusion, then occlusion analysis is employed to infer the correct assignment. A Bayesian Network is used to reason probabilistically about the most likely labelling of the blob. This is shown in figure 4, through three frames of a dynamic occlusion. The person approaching from the centre right intersects the two people walking towards the camera. Partial detection information [15] and occlusion reasoning are used to maintain the original labelling through the occlusion.

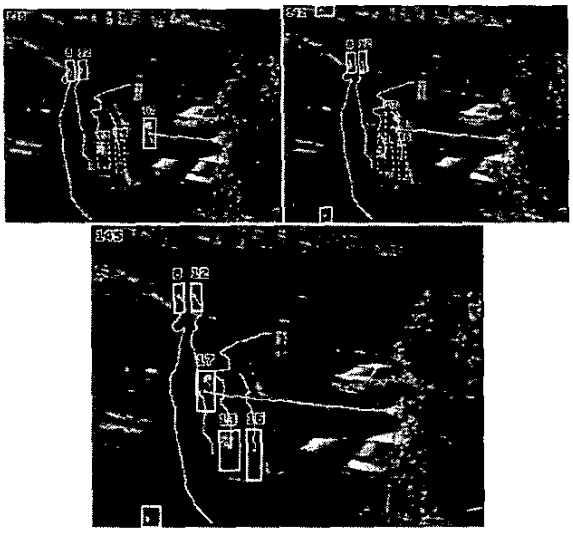

Figure 4. Object tracking through dynamic occlusion. The dashed bounding boxes indicate that only partial information is available. The white lines represent the current trajectories of all visible tracked objects.

2.3 Multi-camera track detection

Tracking objects that appear simultaneously in two or more camera viewfields can be used to minimise the effects of occlusion, on the basis that it will be unlikely that the object is occluded in both views at the same time. In order to be able to correspond information from the two views it is necessary to determine a mapping function that relates the location of a pixel in one view with the same pixel in another view. Such a mapping is called an homography, and can be determined by locating a minimum of four equivalent points in the two views. A constraint of the homography is that the points must lie in the same plane, but this is often satisfied for surveillance systems in man-made environments, where the tracked objects will typically share a common ground plane.

Figure 4 shows the set of trajectories corresponded between two camera views. For these two views, the homography is automatically learnt by an algorithm that identifies possible trajectory correspondences from a pair of cameras. In order to ensure that in-correct correspondences are reliably rejected, a robust search algorithm (least-median of squares, LMS) is employed.

Whilst the homography provides a means to determine point-to-point correspondences between two camera views by learning, it is only useful where camera viewfields overlap. However, it is also important to determine the geometric relationship between all the cameras in the network. This can be achieved by a process of camera calibration, which calculates the intrinsic (focal length, principal point and pixel aspect ratio) and extrinsic (camera location and pointing direction) parameters for each camera in a common world coordinate system. This allows objects from different views to be mapped into a common (ground plane) coordinate system [5,6,9]. By quantifying the accuracy of the calibration (maintained by the covariance of the matched correspondence points) the system can compute a location error for each tracked object. The camera calibration routine requires a minimum of five 3D location points and their corresponding points in image coordinates. The 3D points are usually generated by a manual survey the environment. Manual identification of the correspondence points was used for this work.

Figure 5. Trajectory points corresponded from two views overlapping using the LMS algorithm.

Trajectory data is generated from the algorithms in section 2.1 and 2.2. Correspondence between camera views is determined using the homography to match trajectory data in overlapping views. Following matching, objects are tracked using a 3D Kalman filter, which integrates the measurements from each view [5]. Figure 6 shows the occlusion analysis resulting from multi-view tracking, detecting the small group disappearing from view in one camera behind the white van, but clearly tracked in the second view. The trajectories are projected onto a ground plane map.

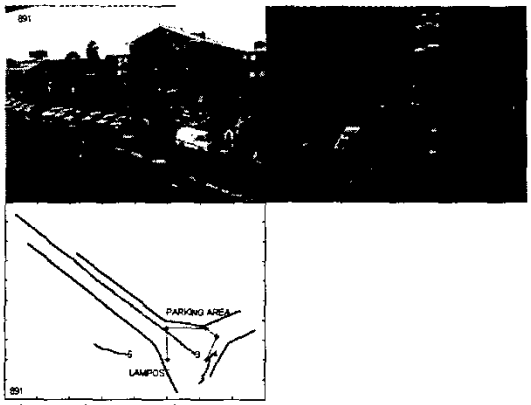

Figure 6. Occlusion reasoning based on analysis of two views. Ground plane map shows trajectories mapped onto a simple scene representation (PETS2001 data sequence [5]).

2.4 Route detection

The trajectory data can be analysed to detect commonly used routes in the scene. This analysis provides an efficient method to encode and annotate individual tracks to construct a log of movement patterns over long periods of time (e.g. days and weeks). Object trajectories can be assigned to one of only a limited number of detected pathways, resulting in significant compression for the logged data. Secondly, the information can be used to support the tracking process giving the system the capability to predict forward many frames, based on the current location and direction. Finally, accumulating tracks over a long time period establishes a pattern of typical movements that can support the recognition atypical or unusual activity.

The trajectories are grouped to construct a geometric model of the most popular routes followed by objects moving through the scene [10]. This model is augmented with a probabilistic model (a hidden Markov model) that is used to represent the usage of routes [11]. The combined geometric and probabilistic models can be used to identify locations in the scene that can be associated with particular types of activity: the popularity of a particular route; common entry and exit regions; regions where objects come to rest (e.g. at a road crossing). Multiple models are used to represent the changing patterns of activity over time. For instance, for the routes detected in figure 7, movement towards (and into) the University main entrance (centre of image) peaks between 8-10 am (see figure 8) and peaks between 4-6 pm for movement away from the entrance. Between 22 pm and 7 am, movement towards the University entrance might be determined to be atypical and could initiate an alarm.

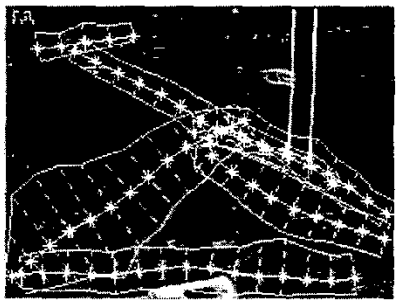

Figure 7. Detected paths extracted from 3500 trajectories observed over a 24-hour period.

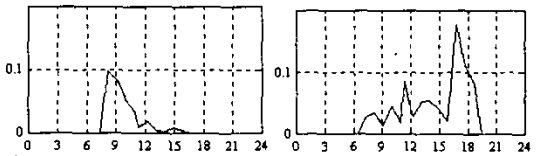

Figure 8. Probability of tracks terminating (left plot) and leaving (right plot) at the University entrance over a 24-hour (weekday) period.

4. Conclusions

This paper has presented a set of algorithms that have been used to detect, locate and track objects moving through an outdoor environment viewed by a number of video cameras. Analysis of the spatial and frequency distribution of trajectories is used to construct models of activity that can be employed to classify particular types of actions and behaviour. The system has been implemented in a real-time surveillance system using COTS (current off-the-shelf) technology and is currently undergoing performance evaluation.

Acknowledgements

This work was undertaken with support from the Engineering and Physical Science Research Council (EPSRC) under grant number GR/M58030. Thanks to Ming Xu, James Black and Dimitrios Makris.

References

1. Ellis T J, Rosin P L, Moukas P, Golton P, "A Knowledge-based Approach to Automatic Alarm Interpretation using Computer Vision, on Images Sequences", Int. Carnahan Conf. on Security Technology, Zurich, October, 1989.
2. Ellis T J, Rosin P L, Golton P, "Model-Based Vision for Automatic Alarm Interpretation", IEEE Aerospace and Electronic Systems Magazine, 6:3, March, 1991, pp 14-20.
3. Brock-Gunn S, Dowling G, Ellis T, "Tracking using colour information", 3rd ICCARV, Singapore, Nov. 1994, pp. pp. 686-690.
4. Clarke T A, Ellis T J, Robson S, "High accuracy 3D measurement using multiple camera views", In: IEE Colloquium Digest No. 1994/054, 1994.
5. Black J, Ellis T, "Multi Camera Image Tracking", PETS2001, Kauai, Hawaii, December 2001.
6. Black J, Ellis T, "Multi Camera Image Measurement and Correspondence", Measurement, 32, 2002, pp. 61-71.
7. Ellis T, "Co-operative computing for a distributed network of security surveillance cameras", IEE European Workshop. Distributed Imaging (Ref. No.1999/109). IEE, London, UK; 1999; 136, pp. 10/1-5.
8. Ellis T, Xu M, "Object detection and tracking in an open and dynamic world", PETS2001, Kauai, Hawaii, December 2001.
9. Lee L, Romano R, Stein G, "Monitoring Activities from Multiple Video Streams: Establishing a Common Coordinate Frame". IEEE Trans. on PAMI, Vol. 22, No. 8, August 2000, pp 117-123.
10. Makris D, Ellis T, "Finding Paths in Video Sequences", in Proc. BMVC2001, Manchester, Sept. 2001, pp 263-272.
11. Makris D, Ellis T, "Spatial and Probabilistic Modelling of Pedestrian Behaviour", British Machine Vision Conference 2002, Cardiff, UK, September 2002.
12. Rosin P L, Ellis T, "Detecting and Classifying Intruders in Image Sequences", in BMVC91, Proc. British Machine Vision Conference, Glasgow, UK, 24-26 September, 1991, pp. 293-300.
13. Stauffer C, Grimson W E L, "Adaptive background mixture models for real-time tracking", *Proc. CVPR'99*, 1999, pp. 246-252.
14. Xu M, Ellis T, "Illumination-invariant motion detection using colour mixture models", in Proc. BMVC2001, Manchester, Sept. 2001, pp. 163-172.
15. Xu M, Ellis T, "Partial observation vs. blind tracking through occlusion", British Machine Vision Conference, Cardiff, September 2002.