Searching in Space and Time: A system for forensic analysis of large video repositories

Anton van den Hengel Rhys Hill Henry Detmold
Anthony Dick

Australian Centre for Visual Technologies
School of Computer Science, University of Adelaide
{hengel,rhys,henry,ard}@cs.adelaide.edu.au

ABSTRACT

The use of surveillance cameras to monitor public buildings and urban areas is becoming increasingly widespread. Each camera delivers a continuous stream of video data, which, once archived, is a valuable source of information for forensic analysis. However, current video analysis tools are primarily based on searching backwards and forwards in time at a single location (i.e. camera), which does not account for events or people of interest that change location over time. In this paper we describe a practical system for tracking a target backwards and forwards in both space and time, effectively following a feature of interest as it moves within and between cameras in a surveillance network. This provides a video analysis tool that is target-centred rather than camera-centred, and thus allows rapid access to the footage that matters for forensic analysis.

Categories and Subject Descriptors

I.4 [**Image Processing and Computer Vision**]: Applications; C.2.4 [**Distributed Systems**]: Distributed applications

Keywords

Surveillance, Forensics, Distributed systems

1. INTRODUCTION

The number of installed surveillance cameras has been rising in recent years, driven by the reduction in the price of camera hardware, installation and infrastructure. Modern surveillance cameras are digital, and deliver a video stream via standard computer network infrastructure. Previously, surveillance cameras were typically black and white analogue cameras, whose output was recorded to tapes. Such cameras require special-purpose cabling and are restricted to transmitting video within a few hundred metres. Longer distance transmission can only practically be accomplished by physically retrieving tapes – a costly and time consuming process.

e-Forensics 2008, January 21-23, 2008, Adelaide, Australia.

Furthermore, automated analysis of analogue footage is impossible without extensive use of video digitising equipment.

The shift to network surveillance cameras has been a driving force behind the large numbers of cameras currently being installed, to the point where networks of thousands of cameras have become common. A surveillance camera must be monitored to be of any value, however, even in a forensic context. Human monitoring of even a single video stream has been shown to be an ineffective means of surveillance[3]. As the number of cameras grows this problem is compounded, important events are increasingly likely to be missed, and value of the surveillance system is reduced. One of the fundamental problems with human monitoring is that surveillance footage tends to be very repetitive and uninteresting. No human operator could be expected to watch hours of footage of an empty warehouse, and notice an event which may only last for a few seconds. Thus, automated analysis is critical, particularly for large networks of cameras.

The forensic analysis of surveillance video is typically concerned with two classes of event: those which have occurred in the very recent past (minutes or seconds ago); and those which could be considered historical, (occurred days or weeks ago). Video evidence for recent events may be used to provide information which can be used to pursue or monitor the parties involved and must be obtained as rapidly as possible. Evidence for historical events is generally less urgent, but must be as thorough and complete as possible. In both cases, there are three important requirements which a surveillance system must meet to provide maximum utility. The most critical is that the system be reliable, in terms of availability and resilience to both hardware and software failures or changes. The second is rapid calculation and extraction of data from the video streams, for further analysis by a human operator. The third is to ensure relevance of the data reported to the analyst, to allow them to focus on important sequences within the footage.

This paper describes a system which allows a moving target to be tracked forwards and backwards in time through a large network of surveillance cameras, without any prior knowledge of the layout of the cameras. This enables rapid and accurate forensic searches to be executed upon very large video archives, with minimal effort by the human analyst. Furthermore, a set of algorithms and designs for constructing a rapid, reliable surveillance system is presented,

along with some data from a current implementation on a large camera network.

2. SYSTEM ARCHITECTURE

The overall structure of the system is shown in Figure 1. The cloud represents a cluster of processing nodes, each responsible for executing tasks such as those described in Section 3. A large storage repository is necessary for archiving the footage. For ease of management, the cluster is controlled by a single computer, through which all contact with the system is arbitrated. The following sections describe in detail the operation of each of these components.

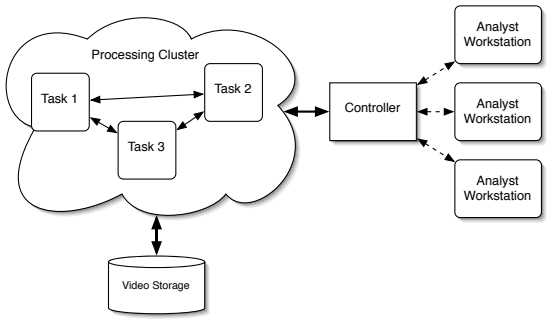

Figure 1: The processing cluster is structured as shown. An important requirement is that all components be independent in order to improve robustness.

3. PROCESSING PIPELINE

The processing pipeline for each task, shown in Figure 2, is composed of a number of different stages. Each of them is now described in turn.

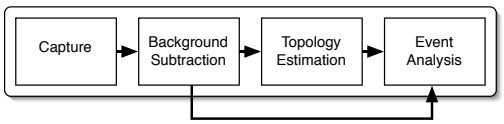

Figure 2: Each task within the processing cluster is structured as a processing pipeline. Elements later in the pipeline are able to draw results from those stage which have already executed.

3.1 Capture

The capturing stage is the simplest, yet the most important within the pipeline. Footage is acquired from the camera and saved out to the disk for later retrieval. As each frame is captured, it is passed on to the remainder of the pipeline, for automated analysis.

3.2 Background Subtraction

Background subtraction is a process which segments foreground objects from the background of the video. The system currently uses a Mixture of Gaussians (MoG) background model [6], known for its accuracy in difficult conditions. The calculation of the background model consumes the largest part of execution time in the system. However, this time is well spent, since the quality of the foreground segmentation is critical to the success of later stages. Once the MoG process has executed, a number of morphological operations are applied to improve the results, followed by a connected components algorithm to label individual objects within the scene. These objects are then passed to the exclusion algorithm, to allow calculation of the topology.

3.3 Topology Estimation

Tracking targets between cameras within the network is a challenging task. However, if a topological relationship between the cameras is known, then the tracking task is far simpler. An algorithm for extracting the topology of a surveillance camera network has recently been described in the literature. This method, called "exclusion"[8], relies on a single logical statement, namely if a point in one camera is occupied at time t, but a second point in another camera is not, then they cannot be viewing the same physical location. This simple fact provides enough constraints to greatly narrow the range of possible topology of a surveillance network.

Exclusion, while a simple idea, is very different to the approach taken by other work in the literature (See [8] for an overview). Most other approaches rely on matching occupied areas, usually via cues derived from the video data. This is difficult task, since the same object can look vastly different from two different perspectives. Other approaches require significant user interaction or a large degree of camera overlap, both of which are unrealistic in a large surveillance system. Exclusion is computationally efficient, does not require explicit target matching and scales up to networks with thousands of cameras[7], making it appropriate for this system.

The topology estimation stage is quite different to earlier stages, since it requires information from all the cameras at once. For this reason, the topology estimation stage in each camera processing pipeline simply compiles a list of occupied regions within the view of the camera, and sends it to the control node of the cluster. This node collates data from all the cameras and uses the exclusion algorithm to generate estimates of the topology.

Once the topology is known, tracking can be performed by following links between cameras from the target's current location. Each target is tracked within a separate task, allowing the tracking process to be load-balanced and scheduled along with the camera processing tasks. Tracking can be peformed both forwards and backwards in time – a critical feature when examining recent past behaviour of a target. This immediately transforms a standard surveillance system into a powerful forensic tool, one which can be utilitised *immediately* after an event has occurred. Figure 4 shows an example of the type of search that can be conducted using this mechanism and Figure 3 shows an example topology.

3.4 Event Analysis

A system which is intended to be largely automatic must have a way of alerting human operators to interesting events. Thus, it must be able to detect that such events have occurred. The scope of this problem is very large, since many different classes of events can be conceived. The processing already carried out by the pipeline can be utilised to monitor a range of different event types, reducing the com-

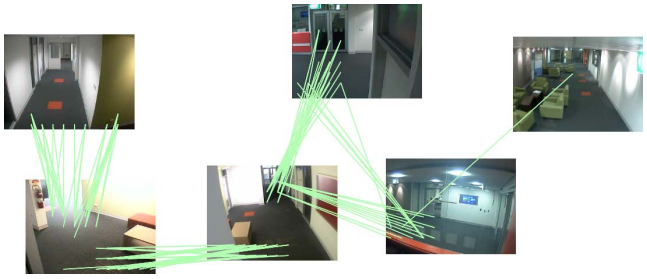

Figure 3: This figure shows an example topology from a real set of surveillance cameras. These results are typical of those recovered from a larger network. Each line represents a link between regions in two cameras.

putational cost of the event analysis process. For example, an analyst may wish to know when a particular area is occupied, a fact that can be obtained from the inputs to the topology estimation algorithm. Similarly, an analyst may be interested to know when the level of activity within a region deviates from its historical trend (for instance, a crowded room suddenly empties, or a large crowd appears in a normally quiet corridor). This can again be obtained via the topology estimation data. The authors are also currently conducting research into extracting additional event types from the MoG background model, to further leverage the existing analysis pipeline.

4. SYSTEM CHARACTERISTICS

4.1 Reliability

The reliability of a surveillance system is measured by the proportion of total time during which it is operational, where operational means capturing, processing and archiving surveillance footage. A surveillance system should be able to survive individual cameras going on and off line, as well as some degree of hardware failure within the processing system itself.

In a large surveillance system, there will often be cameras that go on or off line, due to maintenance, network issues or reallocation of resources. These changes must be compensated for as part of the processing system design. Accommodating changes to the processing infrastructure itself, however, can be more problematic. To counter this problem the processing hardware must be distributed across multiple redundant nodes. Distributing the processing across multiple computers allows the system to survive hardware failure by moving the tasks to be completed to the remaining resources. Facilitating the moving of tasks between processing hardware requires a specific form of system architecture. Once this architecture has been adopted, however, achieving the desired degree of robustness requires only the acquisition of more hardware.

One form of architecture facilitates hardware redundancy by running the program across multiple independent computers, or nodes. This architecture is typically associated with clusters of identical computers which are co-located. Co-location, however, reduces the robustness of the system by introducing single points of failure, and is not necessary. Distributing processing and storage resources provides greater robustness against catastrophic events, and attacks on the surveillance system in particular. The advantage of a (possibly physically distributed) cluster is that if one node fails another can take over without outside intervention. The fact that, for the most part, the processing of the video associated with each camera is independent of others makes surveillance particularly well suited to this form of distribution.

The system represents the processing of video from one camera as a task, which then forms the basis of distribution. One or more tasks execute on each node in the system, and each task may be migrated from one node to another. Internally, a task is subdivided into a number of stages arranged in a pipeline. The number and complexity of these stages depends on the complexity of the surveillance system. A basic surveillance system must be able to capture and display a number of video streams. A more sophisticated system would perform more processing on the data, to highlight important events or areas to the analyst. The task structure used in this system is shown in Figure 2. A number of additional stages could be added, such as a video pre-processing stage, or a watermarking stage, to ensure the integrity of the data is maintained.

Load balancing and fault recovery are achieved by allocating the tasks in a fair way across the nodes in the cluster. Fairness is currently measured by network bandwidth consumption. If a node in the cluster fails, the control node can re-allocate the its tasks to other nodes. If a camera goes on or off line, the central node can ensure that the number of tasks assigned to each node is roughly equal, ensuring best use of computational and network resources. The use of a single control node currently limits the scalability of the system, however this is not an inherent limitation[2].

4.2 Rapidity

Previously, two classes of surveillance events of interest were defined – those which occurred very recently, and those which are considered historic. In both cases, rapidity is an important quality of the surveillance system. For a recent event, rapidity means alerting an analyst as soon as possible that the event has occurred, to allow them to act upon it. Historic events, on the other hand, require that the analyst be provided with rapid access to salient sequences from the video archive. Single camera analysis can give good results for the first criteria, but progress on the second has been slow.

A network-based surveillance system can rapidly browse historical footage by appropriate selection of storage hardware. Careful organisation of video footage within the storage repository can provide improvements to browsing performance as well as ensuring that footage is easy to locate, assuming the user knows how to identify the exact footage they require. If both these points are addressed, historical browsing can be achieved by simply connecting to the storage repository, and examining the captured footage.

However, this naive approach can be significantly improved through the use of a video agent. A video agent takes requests from the user, and interrogates the storage system on

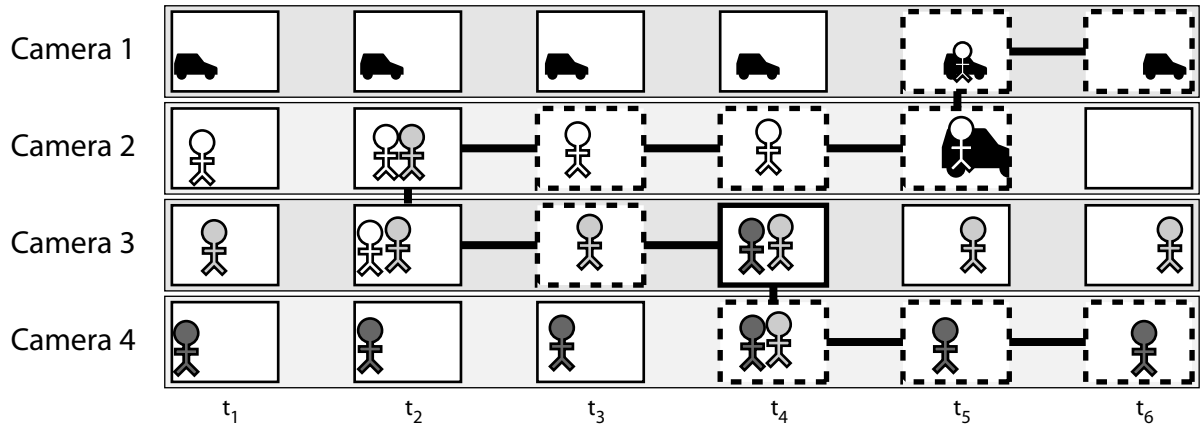

Figure 4: This figure illustrates the potential of a searching approach which can traverse forwards and backwards in time, and between topologically connected cameras. An event initially occurs in `Camera 3` at time t_4. An analyst can then search forwards and backwards in time to locate any other people or objects a particular target interacted with.

their behalf. Such an agent allows a number of intelligent optimisations, which save on network bandwidth between the storage system and the user and allows meta-data to sent along with the video. This meta-data could include event information, highlighting of important areas, target locations, etc. An agent can provide other practical benefits, such as user-authentication and access logging.

Agents are designed to act on the user's behalf, performing actions in their place, such as searching for particular events, tracking individuals or alerting other users of important information or significant events. The agent executes its actions on the processing cluster, rather than the user's workstation, which is particularly useful if the user is controlling the system from a poorly resourced access point, such as a PDA or mobile phone. Neither device has the bandwidth or computational power required to sift through large amounts of video data. Once the agent has completed the requested task, it can return results in a format which is most appropriate for the users access point. This could mean reducing the size of the video, or only returning textual descriptions if necessary.

4.3 Relevance

Existing surveillance systems often overwhelm users with data, much of which is unnecessary, uninteresting and simply distracts attention from important information. A critical function of a modern surveillance system is to alleviate this problem by filtering out data and presenting only that which is relevant. After the London bombing, police were faced with many thousands of hours of footage, within which critical sequences were buried. The police were able to draw upon their own officers to analyse the footage, and were able to locate the key sequences within a few days[1]. However, not all surveillance systems have commensurate levels of analysts available. For example in an airport, where thousands of cameras may be installed, the available staff levels are far fewer. If there was a need to rapidly search the footage from all the cameras in their system, additional staff would need to be obtained temporarily, or the search would simply have to take a long time.

The requirement for human filtering of surveillance video means that under the current model, the cost of filtering goes up with the number of installed cameras. In order to justify searching for an event within a video archive the value of the video to be recovered must be greater than the cost of the search. Thus, as the number of cameras in a network rises, the importance of the event required to justify searching the archive rises too. This means that large camera networks can be less effective than small ones, as the importance of the event required to justify searching the footage generated by a 10,000 camera network can be quite high. One solution to this problem is not to treat each video sequence as being independent, but rather to identify the relationships between the fields of view of the cameras using the exclusion algorithm and to search the video on the basis of these relationships.

Most footage captured by surveillance cameras is uninteresting, because it does not contain imagery which relates to the event in question. Often the redundant footage can be removed by simple means – looking for activity within the view of the camera is a simple but effective filter. Likewise, the time of the event can be used to narrow the range of footage to search. However, to capture the behaviour and location of a subject prior to the event, a large region of time may still need to be searched. A more sophisticated approach would allow a subject to be highlighted by the analyst and then tracked by the surveillance system as they move between cameras and forwards or backwards in time, allowing their path through a particular facility to be determined.

Such a technique also allows an analyst to observe the *interactions* the subject had with objects and people with the area being surveilled. If such information can be obtained rapidly, then witnesses or suspected perpetrators will still be close to the location of the original event, allowing them to be questioned by security or police personnel. Figure 4 shows an example of such a search.

5. CASE STUDY: CAMPUS NETWORK

The bulk of the system described in this paper has been implemented at the University of Adelaide. The University owns a surveillance camera networked comprised of approximately 120 network surveillance cameras, spread across its campus. Processing and analysis is performed on a cluster of 16 2GHz AMD dual-core Opteron servers, with a two 1.86GHz Intel Quad-Core Xeons server as the control node. The cluster nodes are linked to each other, and to the main University network, via gigabit ethernet. Each node in the cluster is capable of processing about 80 frames of video per second. Typically, this is allocated amongst 8 cameras, giving the cluster the ability to process 128 cameras at 10 frames per second. The head node is tasked with estimating the topology of the network. This calculation must be continuous, to allow the network to evolve over time. The head node will also be responsible for performing tracking and for executing video agents, though the current implementation does not yet provide this ability. Figure 5 shows an example of the system tracking a target, a few minutes after the target arrived in the building.

Figure 5: The system shows the cameras in which the target is believed to be present, along with a set of cameras where the camera may travel next. At any time, the user may override the system and request that tracking be restarted on any visible target.

The density of cameras across the campus is highly variable – they tend to be clustered around particular areas. Thus, the topology extracted from the network is often compartmentalised. Tracking between these compartments is difficult, but research is being conducted into appropriate mechanisms to either extend the topology to join the compartments together, or to provide an alternate tracking mechanism between compartments. Figure 3 shows an example of the topological links derived between a typical group of cameras.

Image processing techniques form the heart of this system, and they must be applied to a wide variety of different scenes. The cameras across the University campus are installed in a large number of different environments. Many are indoors, monitoring computer laboratories or lecture theatres, while others are directed at important outdoor thoroughfares through the campus. These scenarios each raise unique challenges. Computer laboratories are difficult to monitor because people tend to sit quite still for extended periods, causing the background model to believe they are a stationary object. When no users are present, the computers all display screen savers, causing continuous motion which must be filtered out for meaningful results to be obtained. Outdoor scenes are challenging due to environmental effects, such as waving trees and moving clouds. There are existing solutions to each of these problems in isolation, but tackling them together is challenging.

A key problem, which is currently being researched, is the removal of foreground detections caused by shadow. These can be particularly troublesome in areas with transitions between indoor and outdoor regions. The literature contains much work in this area[5, 4, 9], but these algorithms often do not work as well on real data, as on the data shown in the publications. The authors are currently trialling two approaches – one based on an intensity invariant colour model, the other on edge information – which offer consistent, if not perfect, results across a wide range of scenarios.

6. CONCLUSIONS

This paper has presented an framework for constructing a powerful surveillance system, which enables human analysts to execute real-time searches across very large repositories of video. This is achieved through a combination of distributed computing, image analysis and network topology estimation. The system allows for evolution of the surveillance system, through its task-based architecture and dynamic load-balancing mechanism. The topology of the network provides information which enables target tracking between linked cameras. When combined with the video storage repository, this allows searching between linked cameras as well as forwards and backwards in time – something which no other system offers for realistic surveillance networks.

To improve the utility of the system, research is being conducted into extracting additional information from the MoG background model along with methods for shadow detection and removal. Further work is being conducted to allow the cluster itself to be segmented, allowing even larger networks to be monitored.

With surveillance cameras becoming increasingly important and wide-spread, rapid, reliable and relevant access to footage is critical if these cameras are to live up to their potential.

7. REFERENCES

[1] Report of the official account of the bombings in london on 7th july 2005, May 2006.

[2] H. Detmold, A. Dick, K. Falkner, D. Munro, A. van den Hengel, and R. Morrison. Middleware for video surveillance networks. In *Proceedings of Middleware for Sensor networks (MidSens2006)*, Melbourne, Australia, November 2006.

[3] M. W. Green. The appropriate and effective use of security technologies in u.s. schools. Technical report, National Institute of Justice, September 1999.

[4] A. Leone and C. Distante. Shadow detection for moving objects based on texture analysis. *Pattern Recogn.*, 40(4):1222–1233, 2007.

[5] E. Salvador, A. Cavallaro, and T. Ebrahimi. Shadow

identification and classification using invariant color models. In *ICASSP '01: Proceedings of the Acoustics, Speech, and Signal Processing, 2001. on IEEE International Conference*, pages 1545–1548, Washington, DC, USA, 2001. IEEE Computer Society.

[6] C. Stauffer and W. E. L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–757, 2000.

[7] A. van den Hengel, A. Dick, H. Detmold, A. Cichowski, and R. Hill. Finding camera overlap in large surveillance networks. In *Proceedings of ACCV 2007*, 2007. To appear.

[8] A. van den Hengel, A. Dick, and R. Hill. Activity topology estimation for large networks of cameras. In *Proceedings of IEEE Conference on Advanced Video and Signal Based Surveillance*, Sydney, Australia, November 2006.

[9] Y. Wang, T. Tan, K. Loe, and J. Wu. A probabilistic approach for foreground and shadow segmentation in monocular image sequences. *Pattern Recognition*, 38(11):1937–1946, November 2005.