A Hierarchical Database for Visual Surveillance Applications

James Black, Tim Ellis, Dimitrios Makris
Digital Imaging Research Centre, Kingston University, United Kingdom
{J.Black, T.Ellis, D.Makris}@kingston.ac.uk

Abstract

This paper presents a framework for event detection and video content analysis for visual surveillance applications. The system is able to coordinate the tracking of objects between multiple camera views, which may be overlapping or non-overlapping. The key novelty of our approach is that we can automatically learn a semantic scene model for a surveillance region, and have defined data models to support the storage of different layers of abstraction of tracking data into a surveillance database. The surveillance database provides a mechanism to generate video content summaries of objects detected by the system across the entire surveillance region in terms of the semantic scene model. In addition, the surveillance database supports spatio-temporal queries, which can be applied for event detection and notification applications.

1. Introduction

Wide area surveillance and monitoring using an intelligent network of cameras is a challenging task. Each camera must be capable of robustly detecting and tracking moving objects of interest, even with the presence of significant illumination changes that typically occur in outdoor environments. A process must also be defined to coordinate the tracking of objects between multiple views, so that a unique identity is assigned to objects visible in overlapping views, and an object's identity is preserved between non-overlapping camera views.

In this paper we primarily focus on how tracking data generated by a network of intelligent cameras can be utilized to support video content analysis for visual surveillance applications. We address several issues associated with data management, which include: how can object track data be stored in real-time in a surveillance database? How can we construct different data models to capture multiple levels of abstraction of the low level tracking data, in order to represent the semantic regions in the surveillance scene? How can each of these data models support high-level video annotation and event detection for visual surveillance applications?

One application of a continuous twenty-four hour surveillance system is that of event detection and recall. The general approach to solving this problem is to employ probabilistic frameworks in order to handle the uncertainty of the data that is used to determine if a particular event has occurred. A combination of both Bayesian classification and Hidden Markov Models (HMMs) were used in the VIGILANT project for object and behavioural classification [1]. The Bayesian classifier was used for identification of object types, based on the object velocity and bounding box aspect ratio. A HMM was used to perform behavioral analysis to classify object entry and exit events.

One problem associated with standard HMMs is that in order to model temporally extended events it is necessary to increase the number of states in the model. This increases the complexity and the time required to train the model. This problem has been addressed by modeling temporally extended activities and object interactions using a probabilistic syntactic approach between multiple agents [2].

The 'Spot' prototype is an information access system that can answer interesting questions about video surveillance footage [3]. The system supports various activity queries by integrating a motion tracking algorithm and a natural language system. The generalized framework supports: event recognition, querying using a natural language, event summarization, and event monitoring. In [4] a collection of distributed databases were used for networked incident management of highway traffic. A semantic event/activity database was used to recognize various types of vehicle traffic events.

In the next section we describe the hierarchical database model we have employed to support the storage of the various types of data that are generated by the intelligent network of surveillance cameras. In section three we discuss the numerous applications for the surveillance database and demonstrate how it can support spatial-temporal queries that can be used to provide event notification and recall applications.

0-7803-8603-5/04/$20.00

2. Database Model

Multi camera surveillance systems can accumulate vast quantities of data when running continuously over extended periods of time. In this paper we address the problem of how this data can be efficiently stored and annotated using a hierarchy of abstract data layers to support online queries and event recall.

2.1 Data Abstraction and Representation

The surveillance database is structured using four layers of abstraction: image framelet layer, object motion layer, semantic description layer, and meta data layer. This four-layer hierarchy supports the requirements for real-time capture and storage of detected moving objects at the lowest level, to the online query and activity analysis at the highest level. Computer vision algorithms are employed to automatically acquire the information at each level of abstraction.

2.1.1 Image Framelet Layer

The image framelet layer is the lowest level of representation of the raw pixels identified as a moving object by each camera in the surveillance network. Each camera view is fixed and background subtraction is employed to detect moving objects of interest [5]. The raw image pixels identified as foreground objects are transmitted via a TCP/IP socket connection to the surveillance database for storage This MPEG-4 like coding strategy enables considerable savings in disk space, and allows efficient management of the video data. Typically, twenty-four hours of video data from six cameras can be condensed into only a few gigabytes of data. This compares to an uncompressed volume of approximately 4 terabytes for one day of video data in the current format we are using, representing a compression ratio of more than 1000:1. The physical database is implemented using PostgreSQL running on a Linux server. PostgreSQL provides support for storing each detected object in the database. This provides an efficient mechanism for real-time storage of each object detected by the surveillance system.

In figure 1 an example is shown of some objects stored in the image framelet layer. The images show the motion history of two objects as they move through the field of view of the camera. Information stored in the image framelet layer can be used to reconstruct the video sequence by plotting the framelets onto a background image. We have developed a software suite that uses this strategy for video playback and review.

Figure 1 Example of objects stored in the image framelet layer.

2.1.2 Object Motion Layer

The object motion layer is the second level in the hierarchy of abstraction. Each intelligent camera in the surveillance network employs a robust 2D tracking algorithm to record an object's movement within the field of view of each camera [6]. Features are extracted from each object including: bounding box, normalized color components, object centroid, and the object pixel velocity. Information is integrated between cameras in the surveillance network by employing a 3D multi view object tracker [7] which tracks objects between partially overlapping, and non-overlapping camera views separated by a short spatial distance. Objects in overlapping views are matched using the ground plane constraint. A first order 3D Kalman filter is used to track the location and dynamic properties of each moving object. When an object moves between a pair of non-overlapping views we treat this condition as a medium term static occlusion, and use the prediction of the 3D Kalman filter to preserve the object identity when if it reappears in the field of view of the adjacent camera.

In figure 2 results from both the 2D tracking and multi-view object tracker are illustrated. The six images represent the viewpoints of each camera in the surveillance network. Cameras 1 and 2, 3 and 4, and 5 and 6 have partially overlapping fields of view. It can be observed that the multi-view tracker has assigned the same identity to each object for the two overlapping fields of view. Figure 3 shows the field of view of each camera plotted onto a common ground plane generated from a landmark-based camera calibration. 3D motion trajectories are also plotted on this map in order to allow the object activity to be visualized over of the entire surveillance region.

2.1.3 Semantic Description Layer

The semantic scene models define regions of activity in each camera view. The information in this layer is populated by post track analysis of trajectories stored in the object motion layer [8]. In Figure 4 the entry zones, exit zones, and routes identified for one of the camera views are shown. The entry zones are represented by black ellipses, while the exit zones are represented by white ellipses. Each

route is represented by a sequence of nodes, where the black points represent the main axis of the route, and the white points define the envelope of the route. Route one and two represent lanes of vehicle traffic in the scene. It can be observed that the entry and exit zones are consistent with driving on the left hand side of the road in the UK. The third route represents flows of pedestrian traffic along the pavement.

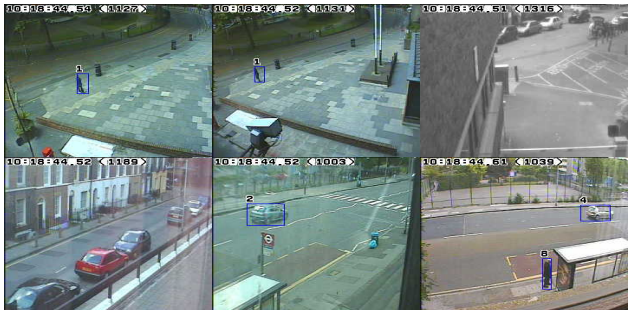


Figure 2. Camera network on University campus showing 6 cameras distributed around the building.

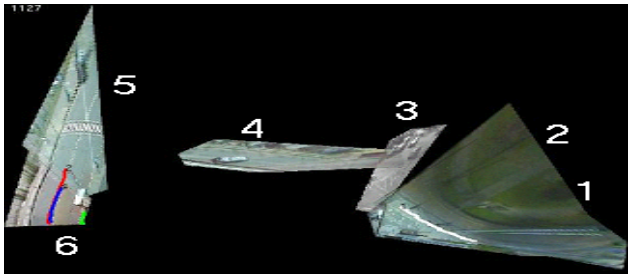


Figure 3. Re-projection of the camera views from figure 2 onto a common ground plane, showing tracked objects trajectories plotted into the views (white, red, blue and green trails).

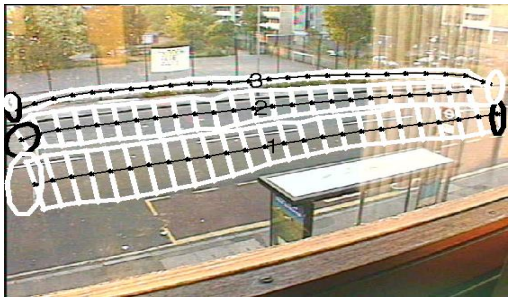
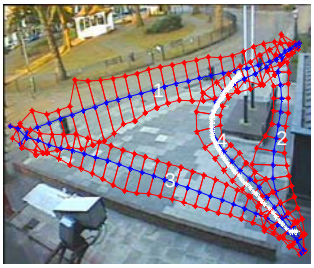


Figure 4. Popular paths learnt from trajectory data.

In figure 5 it is illustrated how the database is used to perform online route classification. Four routes are shown that are stored in the semantic description layer of the database in figure 5(a). In this instance the object trajectory is assigned to route 4, since this is the route with the largest number of intersecting nodes. The corresponding SQL query used to classify routes is shown in figure 5(b). Each node along the route is modeled as a polygon primitive provided by the PostgreSQL database engine. The tracked object trajectories are transformed to a path geometric primitive in the database. The query counts the number of route nodes the object's trajectory intersects with. This allows a level of discrimination between ambiguous choices for route classification. The '?#' operator in the SQL statue is a logical operator that returns true if the object trajectory intersects with polygon region of a route node.



```
select routeid, count(nodeid)
from routenodes r, objects o
where camera=2
and o.trajectory ?# r.polyzone
and o.videoseq =87
and o.trackid =1
group by routeid
```

Figure 5. (a) Example of route classification, (b) SQL query to find route that intersect with an object trajectory

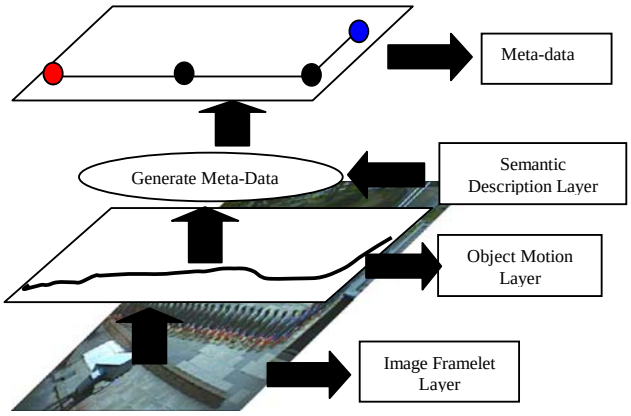


Figure 6. Information flow for online meta-data generation.

2.1.4 Metadata Layer

The multi-layered database allows the video content to be annotated using an abstract representation. It is possible to generate metadata online when detected objects are stored in the image framelet and object motion layers. In figure 6 the data flow is shown from the input video data to the metadata generated online. Initially, the video data and object trajectory is stored in the image framelet and object motion layers. The object motion history is then expressed in terms of the models stored in the semantic description layer to produce a high-level symbolic description of the object's activity. The metadata contains information for each detected object including: entry point, exit point, time of activity, and the routes taken

through the field of view, along with the time spent in each route node. This information is tagged to each object detected by the system.

3. Applications

In order to evaluate the performance of the hierarchical database we have run the system continuously over a twenty-four hour period using a camera network consisting of six intelligent cameras. The majority of the tracking data is generated by cameras 5 and 6, which overlook a road that has regular flows of vehicle traffic. The peak data transmission rate is at around 5pm, which is consistent with the time of rush hour traffic in London. The network traffic generated is much lower than that required to transmit the original video from six cameras over the network.

The metadata provides better indexing of the object motion and image framelet layers of the database, which results in improved performance for various types of activity queries. This point is illustrated in Figure 7, where an activity query was run to identify object motion between various pairs of entry and exit zones within a specific time interval. In Figure 7 objects moving between entry zone B and exit zone A are shown. The meta-data generation results in compact video content summaries of each object detected by the surveillance system. The meta-data can be assessed for video content analysis of the underlying low-level video data. Another example of the results returned by a spatial temporal query is shown in Figure 8. An activity query was executed to return objects that have followed a certain path over a specific time interval. The results show objects that have followed two of the paths in one of the camera views.

4. Conclusion

We have presented a hierarchical database that can be employed to capture and store tracking data in real-time and generate video content summaries. One key novelty of our system is that the surveillance database contains semantic scene models that are generated automatically by post track analysis of object tracking data. The main benefit of the framework is that it is possible to execute high-level object activity queries using a SQL database. The meta-data reduces the response times of activity queries from several minutes to a few seconds. In future work we plan to generate more complex activity queries and use probabilistic methods to recognize different types of object interactions.

Figure 7. Visual representation of results returned by spatial temporal activity queries: objects moving from entry zone B to exit zone A

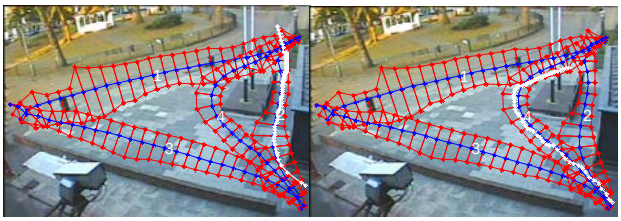


Figure 8. Example of results returned by spatial temporal activity queries

References

[1] Remagnino P, Jones GA, Classifying Surveillance Events and Attributes and Behavior, British Machine Vision Conference (BMVC2001), Manchester, September 2001, 685-694.

[2] Ivanov YA, Bobick AF, Recognition of Visual Activities and Interactions by Stochastic Parsing, Pattern Analysis and Machine Intelligence (PAMI), Vol. 22, No. 8, August 2000, 852-872.

[3] Katz B., Lin J., Stuaffer C., Grimson E. Answering Questions about Moving Objects in Surveillance Videos. Proceedings of 2003 AAAI Spring Symposium on New Directions in Question Answering, March 2003.

[4] Trivedi M., Bhonsle S., Gupta A. Database Architecture for Autonomous Transportation Agents for On-scene Networked Incident Management (ATON). International Conference on Pattern Recognition (ICPR2000), Barcelona, Spain, 2000.

[5] Xu M, Ellis TJ, Illumination-Invariant Motion Detection Using Color Mixture Models, British Machine Vision Conference (BMVC 2001), Manchester, September 2001, 163-172.

[6] Xu M, Ellis TJ, Partial Observation vs Blind Tracking through Occlusion, British Machine Vision Conference (BMVC 2002), Cardiff, September 2002, 777-786.

[7] Black J, Ellis T.J., Multi View Image Surveillance and Tracking, IEEE Workshop on Motion and Video Computing, Orlando, December 2002, 169-174..

[8] Makris D, Ellis T.J., Automatic Learning of an Activity-Based Semantic Scene Model, IEEE Conference on Advanced Video and Signal Based Surveillance (AVSB 2003), Miami, July 2003, pp183-188