Toward Building a Robust and Intelligent Video Surveillance System: A Case Study

Edward Y. Chang

Department of Electrical and Computer Engineering
University of California
Santa Barbara, CA 93106

Yuan-Fang Wang

Department of Computer Science
University of California
Santa Barbara, CA 93016

Abstract

With the proliferation of inexpensive cameras, availability of large-capacity disk storages, and ubiquitous presence of high-speed, broad-band communication networks, it is now economically and technically feasible to construct and deploy multi-camera video surveillance systems. In line with this is the need of intelligent, robust, (semi-)automated video analysis paradigms to assist the operators in scene analysis and event classification. In this paper, we summarize our current research work toward realizing such a multi-camera video surveillance system.

1 Introduction

Video cameras are becoming a ubiquitous feature of modern life, useful for surveillance, crime prevention, and forensic evidence. Many extended "eyes" are being installed at an unprecedented pace, yet the intelligence needed for interpreting video-surveillance events by computers is still rather unsophisticated. In a recent ACM video-surveillance workshop co-chaired by the authors [1], participating developers and practitioners emphasized the urgent need for an enhanced "brain" to match up with these multiple camera views for video analysis and query answering. We cannot solely rely upon human effort to watch and sift through hundreds and thousands of video frames for crime alerts and forensic analysis. That is a non-scalable task. We need a (semi-)automated video-analysis and event-recognition system that can provide timely warnings to alert security personnel, and that can substantially reduce the search space for forensic-analysis tasks.

A multi-camera surveillance task can be divided into two major phases: *data fusion* and *event recognition*. The data-fusion phase integrates multi-source spatio-temporal data to detect and extract motion trajectories from video sources. The event-recognition phase deals with classifying the events as to relevance for the search. The research challenges of the two phases are summarized as follows:

- *Data fusion from multiple cameras.* Observations from multiple cameras should be integrated to build spatio-temporal patterns that correspond to 3-dimensional viewing. Such integration is necessary to improve surveillance coverage, and to deal with object-tracking obstacles such as spatial occlusion and scene clutter.
- *Event recognition.* Event recognition deals with mapping motion patterns to semantics (e.g., benign and suspicious events). Most traditional machine-learning algorithms cannot be directly applied to such infinite-dimensional data, which may also exhibit temporal ordering. In addition, positive events (i.e., the sought-for hazardous events) are always significantly outnumbered by negative events in the training data. In such an imbalanced set of training data, the class boundary tends to skew toward the minority class and becomes very sensitive to noise.

In this paper, we summarize our current research on developing such a multi-camera video surveillance system. In particular, we discuss its hardware capability and its important software features.

2 Hardware Architecture

Fig. 1 depicts the system configuration. Multiple slave surveillance stations, each comprising a video camera connected to a host PC, are positioned at different physical locations to monitor the ground activities, say, in a parking lot. A camera is mounted on a PTZ (pan-tilt-zoom) platform, which allows its pose and aim to be dynamically controlled. This is advantageous if close-up views of moving objects are needed for identification.

The video stream obtained from each camera is encoded using standard encoding algorithms such as H.263, MPEG1 or MPEG4. Each stream is then relayed over wireless links to the master server for storage. The server indexes and stores video signal with their meta-data on RAID storage. The storage system provides real-time stream retrieval and supports scan operations such as rewind, forward, and slow-motion. Users of the system are alerted to unusual events and they can perform online queries to retrieve and inspect video clips of interest.

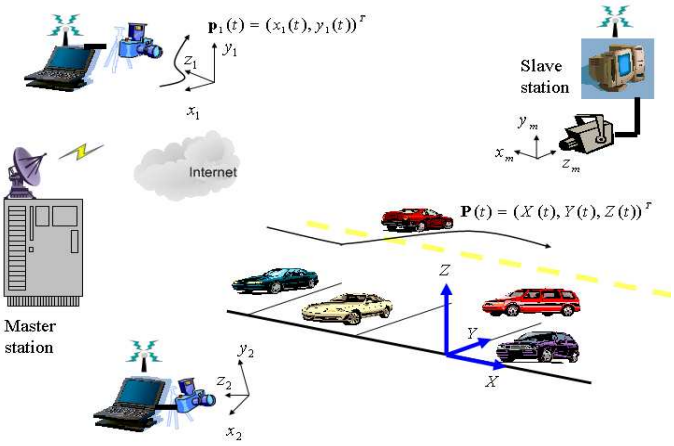

Figure 1: Surveillance system configuration

Because our system architecture is highly modularized, there is no restriction on the number of cameras used, their brand names, or their capability. The camera aim can be stationary, follows a fixed sweep pattern, or is remotely controlled by a human operator. The view volumes of different cameras can be totally disjoint or can overlap partially. Furthermore, the clocks on different slave stations need not be synchronized.

3 Software Architecture

Our software modules are geared toward automated detection and characterization of observed motion events. For event detection, we have focused on fusing sensor data for improving the reliability and robustness of event tracking. For event characterization, we have designed strategies for sequence data learning and biased learning. These activities are described in more details below.

3.1 Event Detection

The event detection stage aims at achieving optimal fusion of multi-sensor data spatially and temporally to derive a hierarchical and invariant description of the scene activities. Our sensor data fusion algorithm addresses both the bottom-up data integration problem and the top-down information dissemination problem in a coherent framework. Several issues are addressed in our event detection and sensor fusion framework: *variability in the spatial coverages* and *misalignment of the temporal time stamps* of multiple cameras, and *occlusion and missing data*.

Background Subtraction A widely-used technique for moving object segmentation is the background subtraction, which compares color or intensity of pixels in adjacent video frames. Significant differences are attributed to object motion. To increase the robustness of background subtraction, it is necessary to distinguish motion due to incidental environmental factors (e.g., lighting changes, shadow, and swing of vegetation) from those of interest (e.g., human and vehicular motions). Our approach uses an optimization scheme which maps the segmentation problem onto a recurrent stochastic binary network. We address the key issues in designing the energy function to take into consideration of both intensity and color change and optical flow information. The computation of this model is completely local and biologically plausible, and provides good segmentation results.

Spatial Registration This step is for determining the essential camera parameters and the pose and aim of a camera in the environment. While camera calibration usually needs to be done once and off-line (if the camera settings are not changed later), pose registration is a continuous process and is performed on-line for a mobile camera platform. The performance of pose registration is thus more critical and time sensitive. Theoretically, if the camera observes six landmarks, whose world coordinates are known, it is readily shown that a linear closed-form solution exists to pose registration. However, such a solution may not be satisfactory for real-world applications.

This is because in the real world scenario, given the general pose and aim of a mobile camera platform, finding six landmarks that (1) have known world coordinates, (1) are present in the field-of-view of the camera, and (3) remain visible even with dynamic camera aim, is a nontrivial task. Hence, alternative pose registration methods that require fewer landmarks for registration will be beneficial. Our alternative is to use an algorithm first developed by Earl Church back in 1945 for aerial photogrammetry. Church's algorithm is an iterative, nonlinear optimization technique that requires only three landmarks for pose registration. The solution is based on the condition that the face angle subtended by any two landmarks in space is equal to the face angle subtended at their corresponding image locations. Such constraints can be used to iteratively update the pose and aim of the camera. Our experience with Church's algorithm is that it is very accurate and efficient. It is possible to achieve thousands of pose updates per second using the current PC technology.

Temporal Alignment The same 3D trajectory is observed by multiple cameras, and hence, the same trajectory appears differently in different cameras' images because of the projective distortion. If we can somehow derive a unique, or invariant, "signature" of a 3D trajectory from its 2D projections, regardless of the particular way an image is generated, then we can correlate these invariant trajectories from different sensors to establish the time shift, and hence, solve the temporal registration problem.

Our invariant signature is designed based on two principles: First, it is well established in differential geometry that

a 3D curve is uniquely described (up to a rigid motion) by its curvature and torsion vectors with respect to its intrinsic arc length. Second, under the parallel projection model and the far field assumption (where the object size is small relative to the distance to the camera, an assumption that is generally true for outdoor surveillance applications), the affine projection model can be used to approximate the perspective model in image formation. And the affine model preserves the ratio of areas among different projections. Based on these two principles, we designed an invariant signature that uses normalized curvature and torsion ratios which are preserved under the affine model.

Sensor Data Fusion We developed a hierarchical master-slave fusion framework. Referring to Fig. 1, at the bottom level, each slave station tracks the movements of scene objects semi-independently. The local trajectories (each represented as a state vector comprising the estimated position, velocity, and acceleration of the tracked object) are then relayed to a master station for fusing into a consistent, global representation. This represents a "bottom-up" analysis paradigm. Furthermore, as each individual camera has a limited field of view, and occlusion occurs due to scene clutter, we also employ a "top-down" analysis module that disseminates fused information from the master station to slave stations which might lose track of an object.

At a slave station, we employ two different mechanisms for event detection; both are based on the powerful hypothesis-and-verification paradigm. The difference is in the number of hypotheses that are maintained. When the state prior and noise processes are modeled as uni-modal Gaussian processes, a single state is maintained. Kalman filter has proven to be very effective in such situations.

While Kalman filter is a simple and powerful mechanism for state estimation, its validity is challenged if the assumption on the prior and noise is not valid. Furthermore, there are situations where multiple hypotheses have to be kept until a later time when more visual evidence is gathered to validate some and discredit others. For example, if two or a group of persons enter the field of view of a camera in such a way that their silhouettes overlap, the tracking algorithm will not know in general whether such a moving region corresponds to a single person or multiple persons. Only when the group of people split later and head in different directions can the single person hypothesis be safely discarded.

Our approach here is to employ a robust, yet still real-time, control and fail-over mechanism—on top of low-level frame differencing- and correlation-based tracking—to deal with noise, scene clutter, short periods of absence and merging of silhouettes, and long periods of occlusion of activities from a camera's field of view—situations that can easily fail simple Kalman filter-based tracking.

Our formulation is based on the powerful hypothesis-and-verification paradigm. The utility of such a hypothesis-verification formulation, over traditional linear state estimation algorithms such as Kalman filtering, is that the noise processes do not have to be Gaussian and state propagation does not have to be unimodal. This allows multiple competing hypotheses to be maintained and contribute to the state estimation process. If we denote sensor data as $\mathbf{z}$, then multiple hypotheses allow us not to assume a particular parametric form of $p(\mathbf{x}|\mathbf{z})$. Instead, $p(\mathbf{x}|\mathbf{z})$ can be learned by sampling with multiple hypotheses using Bayesian rule ($p(\mathbf{x}|\mathbf{z}) \propto p(\mathbf{z}|\mathbf{x})p(\mathbf{x})$). In this sense, hypothesis-verification is akin to a Bayesian estimator instead of a maximum likelihood estimator.

We envision that future surveillance systems can be fairly "compartmentalized." A surveillance station can be made of a camera and associated mounting device, controlled by an inexpensive PC with a digitizer card and disk storage. A complete surveillance system then comprises many such stations to achieve scalability. In this scenario, We should take advantage of the processing power of individual surveillance stations to parallelly process video footage instead of using one big server to achieve integration from raw data. Our master-slave sensor-fusion scheme is ideally suited for such a distributed surveillance system.

3.2 Event Characterization

Raw trajectory data derived above are in terms of either local or global Cartesian coordinates. Such a representation is difficult for a human operator to understand. Our solution is to summarize such raw trajectory data using syntactic and semantic descriptors that are not affected by incidental changes in environmental factors and camera poses, and are easier for a human operator to interpret.

We first segment a raw trajectory fused from multiple cameras into fragments, using a constrained optimization approach under the EM (expectation-maximization) framework. We then label these fragments semantically. This is done by approximating the acceleration trajectory of a vehicle as a piecewise- constant (zeroth-order) or linear (first-order) function in terms of its direction and its magnitude. We then use that information to label each segment (e.g., as speed-up, slow-down, left-turn, and right-turn actions). Concatenation of successive segments according to some predefined Markov models provides an interpretation of the whole trajectory (e.g., successive turning and straight actions may signal a circling behavior).

Recognizing events on such descriptors must handle the ordered nature of the descriptors. Furthermore, in a surveillance setting, positive (suspicious) events are always significantly outnumbered by negative events. This imbalanced training-data situation can skew the decision boundary toward the minority class, and hence cause high rates of false negatives (i.e., failure to identify suspicious events). Our solution is to design a sequence-alignment kernel function to work with SVMs for correlating events. We then pro-

pose using kernel boundary alignment (KBA) to deal with the imbalanced training-data problem.

Sequence Alignment Learning We have labeled each segmented fragment of a trajectory with a semantic label and its detailed attributes including velocity and acceleration statistics. Now, the trajectory learning problem is converted to the problem of sequence-data learning with secondary variables. For this purpose, we construct a new *sequence-alignment kernel* that can be applied to measure pair-wise similarity between sequences with secondary variables. The sequence-alignment kernel will take into consideration both the degree of conformity of the symbolic summarizations and the similarity between the secondary numerical descriptions (i.e., velocity and acceleration) of the two sequences. Two separate kernels are used for these two criteria and are then combined into a single sequence-alignment kernel through *tensor product*.

More specifically, our idea is to first compare similarity at the symbol level. After the similarity is computed at the primary level, we consider the similarity at the secondary variable level. We then use the tensor product kernel to combine the similarity at the primary and secondary level.

Imbalanced Learning via Kernel Boundary Alignment Skewed class boundary is a subtle but severe problem that arises in using an SVM classifier—in fact in using *any* classifier—for real world problems with imbalanced training data. To understand the nature of the problem, let us consider it in a binary (positive vs. negative) classification setting. Recall that the Bayesian framework estimates the posterior probability using the class conditional and the prior. When the training data are highly imbalanced, it can be inferred that the state of the nature favors the majority class much more than the other. Hence, when ambiguity arises in classifying a particular sample because of similar class conditional densities for the two classes, the Bayesian framework will rely on the large prior in favor of the majority class to break the tie. Consequently, the decision boundary will skew toward the minority class.

While the Bayesian framework gives the optimal results (in terms of the smallest average error rate) in a theoretical sense, one has to be careful in applying it to real-world applications. In a real-world application such as security surveillance, the risk (or consequence) of mispredicting a positive event (a false negative) far outweighs that of mispredicting a negative event (a false positive). It is well known that in a binary classification problem, Bayesian risks are defined as:

$$\begin{aligned} R(\alpha_p|\mathbf{x}) &= \lambda_{pp}P(\omega_p|\mathbf{x}) + \lambda_{pn}P(\omega_n|\mathbf{x}) \\ R(\alpha_n|\mathbf{x}) &= \lambda_{np}P(\omega_p|\mathbf{x}) + \lambda_{nn}P(\omega_n|\mathbf{x}) \end{aligned} \quad (1)$$

where p and n refer to the positive and negative events, respectively, λ_{np} refers to the risk of a false negative, and λ_{pn} the risk of a false positive. Which action (α_p or α_n) to take—or which action has a smaller risk—is affected not just by the event likelihood (which directly influences the misclassification error), but also by the risk of mispredictions (λ_{np} and λ_{pn}).

For security surveillance, positive (suspicious) events often occur much less frequently than negative (benign) events. This fact causes imbalanced training data, and thereby results in higher incidence of false negatives. To remedy this boundary-skew problem, we propose an adaptive conformal transformation algorithm.

A conformal transformation, also called a conformal mapping, is a transformation T which takes the elements $X \in D$ to elements $Y \in T(D)$ while preserving the local angles between the elements after mapping, where D is a domain in which the elements X reside.

Kernel-based methods, such as SVMs, introduce a mapping function Φ which embeds the input space I into a high-dimensional feature space F as a curved Riemannian manifold S where the mapped data reside. A Riemannian metric $g_{ij}(\mathbf{x})$ is then defined for S, which is associated with the kernel function $K(\mathbf{x}, \mathbf{x}')$ by

$$g_{ij}(\mathbf{x}) = \left(\frac{\partial^2 K(\mathbf{x}, \mathbf{x}')}{\partial x_i \partial x'_j} \right)_{\mathbf{x}'=\mathbf{x}}. \quad (2)$$

The metric g_{ij} shows how a local area around $\mathbf{x}$ in I is magnified in F under the mapping of Φ. The idea of conformal transformation in SVMs is to enlarge the margin by increasing the magnification factor $g_{ij}(\mathbf{x})$ around the boundary (represented by support vectors) and to decrease it around the other points. This could be implemented by a conformal transformation of the related kernel $K(\mathbf{x}, \mathbf{x}')$ according to Eq. 2, so that the spatial relationship between the data would not be affected too much.

When the training dataset is very imbalanced, the class boundary would be skewed toward the minority class in the input space I. We hope that the new metric $\tilde{g}_{ij}(\mathbf{x})$ would further magnify the area far away from a minority support vector $\mathbf{x}_i$ so that the boundary imbalance is alleviated.

4 Concluding Remarks

This paper summarizes our framework for robust and intelligent video interpretation for video surveillance applications. Due to space limit, we presents only a brief overview. Experimental results and real-time video demo will be presented at the main conference.

Acknowledgment

We would like to acknowledge the support of NSF grants IIS-0133802 (CAREER), IIS-0219885, IIS-9908441, and EIA-0080134.

References

[1] Acm first internal workshop on video surveillance.