CENTER FOR THE COMMERCIALIZATION OF INNOVATIVE
TRANSPORTATION TECHNOLOGY
(CCITT)

USDOT UNIVERSITY TRANSPORTATION CENTER

NORTHWESTERN UNIVERSITY

FINAL REPORT

iTRAC: Intelligent Video Compression for Automated Traffic Surveillance Systems

Principal Investigators
Sotirios A. Tsaftaris, Research Assistant Professor
Aggelos K. Katsaggelos, Professor

Student
Eren Soyak

Department of Electrical Engineering and Computer Science

August 1 2010

Disclaimer

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the Department of Transportation University Transportation Centers Program, in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof.

This work was funded by the Northwestern Center for the Commercialization of Innovative Transportation Technology (CCITT).

CCITT (http://www.ccitt.northwestern.edu) is a University Transportation Center funded by the Research and Innovative Technology Administration (http://www.rita.dot.gov/) of USDOT operated within the Northwestern University Transportation Center in the Robert R. McCormick School of Engineering and Applied Science (http://www.mccormick.northwestern.edu).

Prof. Sotirios A. Tsaftaris is with the Northwestern University Departments of Electrical Engineering & Computer Science and Radiology. He can be reached at s-tsaftaris@northwestern.edu.

Prof. Aggelos K. Katsaggelos is with the Northwestern University Department of Electrical Engineering & Computer Science. He can be reached at aggk@eecs.northwestern.edu.

Eren Soyak is with the Northwestern University Department of Electrical Engineering & Computer Science. He can be reached at e-soyak@northwestern.edu.

Chapter 1

Introduction

1.1 Project Summary

Non-intrusive video imaging sensors are commonly used in traffic monitoring and surveillance. For some applications it is necessary to transmit the video data over communication links. However, due to increased requirements of bitrate this means either expensive wired communication links or the video data being heavily compressed to not exceed the allowed communications bandwidth. Current video imaging solutions utilize aging video compression standards and require dedicated wired communication lines. Recently H.264 (a newer standard) has been proposed to be used in transportation applications. However, most video compression algorithms are not optimized for traffic video data and do not take into account the possible data analysis that will follow either in real time at the control center or offline. As a result of compression, the visual quality of the data may be low, but more importantly, as our research efforts in vehicle tracking has shown, the tracking accuracy and efficiency is severely affected. iTRAC aims to inject highway content-awareness in the H.264 encoding standard. Our technology operates within the computational limits of consumer grade hardware equipment. With the possible reduction in bitrate we envision that we can provide a portable, easy to deploy, low cost, low power, wireless video imaging sensor.

1.2 Project Goals

Non-intrusive video imaging sensors are commonly used in traffic monitoring and surveillance [1]. They are the only cost effective solution that yields information on a large field of view that allows for real time monitoring of video feeds and video archiving for forensic or traffic analysis applications. Other imaging solutions (ie., Autosense Solo) can only count and identify vehicles and measure instantaneous speed without providing any information on the path a vehicle took in a area of interest. Video imaging is the only modality that observes a vehicle's trajectory (path), which subsequently allows us to study driver behavior and its possible effects on congestion. Recently, automated video analysis has been suggested for the extraction of a vehicle's trajectory, speed, and type (car, truck, etc) for a variety of applications [2, 3]. Video data are compressed to reduce the amount of information being transmitted or stored. Even with recent video standards (H.264) the bit-rate is high forcing the use of dedicated wired lines (T1 or fiber optic lines). A low cost, low power, wireless video imaging sensor that could be easily deployed over areas of interest, will enable transportation officials to monitor these areas without a large investment in infrastructure and time-consuming planning. If the video feed is post-processed by computers to extract the trajectories of each vehicle the quality of the data have a large impact on the accuracy of the tracking. Therefore, it is critical to maintain tracking efficiency in the presence of compression.

Through our research we have identified that the quality of the transmitted/archived video is critical for the accurate detection and tracking of vehicles, humans, or even animals. Video parameters such as resolution, frame rate, and data rate are quite critical and each have a direct impact on the performance of many target tracking algorithms. For example, if the resolution is small, and the camera has a wide field of view, targets can become too small to be tractable [4, 5]. In addition, weather and lighting conditions can affect the accuracy of tracking algorithms.

Herein we propose iTRAC for H.264, an intelligent algorithmic module to be used in conjunction with the H.264 encoding standard. Compression algorithms in general tend to be content agnostic, aiming to minimize the video data rate while maintaining requested video quality as expressed by an objective quality metric (e.g., mean squared error). We move away from this common approach and provide a content aware system based on the H.264 codec that is designed to minimize the compressed video data rate

while maintaining detection accuracy. iTRAC places special focus on moving objects or targets of interest and compresses them with such quality that detection and tracking accuracy are maintained at high levels. The question the encoder has to answer is how much data can be removed such that the decoder can still detect and track the objects of interest as if there were no compression at all. So in our case quality is defined simply as the accuracy of the tracking result. In fact, the Federal Highway Administration defines quality as "the fitness of data for all purposes that require it" [6]. We should note that even if a human will monitor the video feed in real-time, our proposed approach will assist them since our video data will provide higher visual fidelity on the moving targets (vehicles) as compared to a content-agnostic H.264 implementation.

1.3 Outline

In this work we discuss the various technologies that when used individually or in conjunction with each other implement the iTRAC system. This report is organized as follows. In Chapter 2 we introduce the problem of optimal video compression for video surveillance of vehicle traffic, and review existing work in the literature concerning traffic video tracking and content-specific video compression. In Chapter 3 we present a method of spatial resource concentration via Region of Interest (ROI) coding for video compression; this work has appeared in [7]. In Chapter 4 we present an algorithm to optimize tracking accuracy for a given bitrate by concentrating available bits in the frequency domain on the features most important to tracking; this work is to appear in [8]. Finally in Chapter 5 we present concluding remarks.

Chapter 2

Background

In this chapter we will present a review the state of the art in Traffic Surveillance Systems, focusing on the areas of Video Compression and Video Object Tracking as relevant to the field. By such reviews we lay the groundwork for our novel algorithms and proposed future work in subsequent chapters.

2.1 Real-world Traffic Surveillance Systems

As a natural extension of modern urbanization, increasing vehicle traffic in populated areas has created a need for automated regulation. The current trends in urban traffic volume indicate that surveillance and control systems capable of a diverse range of tasks need to be made available at most medium- and high-utilization roads. The high level needs such systems must address include the following:

- gathering low-complexity statistics such as congestion or average vehicle velocity
- gathering high-complexity statistics such as driver behavior or road conditions
- recording events of interest for purposes such as security, accident documentation or law enforcement
- automatic responses to predefined events such as speed limit violations or accidents.

By this definition it is clear that the desired systems must possess the capability for higher-order tasks such as identifying vehicles or responding to "risky" driver behavior. Such capability will require an architecture capable of complex tasks yet affordable enough to make the required wide-scale deployment feasible. A sample study of capabilities expected from an intelligent surveillance system is presented in [9].

Current traffic surveillance systems for the most part make use of mature solutions such as inductor cables embedded in roads to count passing cars and fixed or handheld radar units for speed detection. Newer technologies that have seen recent deployment include video surveillance systems to record and respond to low-complexity events such as red light infractions or improper safety lane usage. However, even these newer systems are limited in the range of tasks they can accomplish, and do not possess the capability to address most of the needs described above.

The system for which algorithms will be proposed in this work is a "centrally controlled" traffic surveillance application. Such a system is comprised of a nodular structure, with low-cost, easy to install remote camera units whose small size allows them to blend with the rest of the urban infrastructure. These remote nodes capture and compress video for transmission over a wireless link to a central processing station, where the bulk of the processing capability of the system resides. Such a system, given its centrally located processing power, is unconstrained in the complexity of tasks it can undertake, and yet is still relatively affordable and easy to deploy with sufficient coverage given the simplicity of its remote nodes. The parameters for the system would be as follows:

- Low-power nodes requiring only a power connection in terms of physical links.
- Low-cost and easy to deploy wireless remote nodes, mountable on existing infrastructure such as poles or traffic signals.
- Full-duplex wireless communication channel between central processing and remote nodes, carrying compressed video on the uplink (remote to base) and control information on the downlink (base to remote).
- Powerful central processing station where records are kept, statistics are gathered and automatic responses are generated. Remote nodes are controlled from here, allowing the system to adapt to changing

conditions such as weather, day/night or even new functionality (implemented via a software update).

Such systems are very difficult to build with existing technology due to the poor performance of computer vision algorithms on compressed video. Given the bandwidth limitations of wireless channels, the limited processing power available at remote nodes and the real-time operating constraints of many desirable traffic surveillance applications the compressed video that is transmitted to the central processing station is typically quite poor in quality. Moreover, most computer vision algorithms that are to be used rely on models based on the nature of the video content they seek to process. Such models may no longer be realistic for video distorted by compression.

On the other hand, the bandwidth requirement to send video compressed at quality that is acceptable for tracking algorithms is not commonly available in wireless environments, typically requiring expensive dedicated channels or hard to install and costly to maintain landlines – in [10] a study discussing typical costs of video surveillance is presented. Given these parameters, wide scale deployment of effective traffic surveillance systems are not feasible due to the cost of installation and maintenance. In [11] an example can be seen of how even modest gains in the compression subsystem can make drastic changes to the feasibility of real-world traffic surveillance systems.

2.2 Video Compression for Traffic Surveillance

Compression artifacts are debilitating for tracking applications. In reviews of object tracking presented in [12] and [13] it is shown that most algorithms focus on the following three features in video to track objects:

- spatial edges
- color histograms
- detected motion boundaries.

Coding artifacts introduced by motion compensated video compression impact all three of these features – color histograms are distorted, true edges are smeared, and artificial edges are introduced. As a result the estimated motion field of pixels is sometimes significantly distorted. Other artifacts attributed to heavy quantization are contouring and posterizing (in otherwise

smooth image gradients), staircase noise along curving edges, and "mosquito noise" around edges. Artifacts attributed to the time aspect of video are motion compensation errors and quantization drift. Compensation errors arise from the fact that motion compensation does not aim at finding the true motion of objects but rather the most similar object in a limited search area. For example, heavily quantized but motionless areas such as the road surface will flicker with time, appearing to have different intensity. Subsampling of chroma components (typically from 4:4:4 to 4:2:0) in the YUV colorspace further reduces the accuracy of color histogram based tracking.

These artifacts and distortions decrease the accuracy of computer vision based tracking algorithms. Fig. 2.1 offers examples of such distortions. The left column shows sample images from video sequences, the top being uncompressed, the center compressed at a ratio of $10^2 : 3$, and the bottom at a ratio of $10^4 : 3$. For each video a background model is computed by taking the median intensity of each pixel over time, which is then subtracted from each frame to give an error image (shown in center column). This error is used to locate objects in each frame, even if they have not moved since the previous frame. The pixel intensity histograms of the images (shown in right column) are used to associate objects from different frames, thereby tracking each object across time. Note that blocking artifacts due to quantization are much more pronounced in the higher compression ratio video. Distorted edges and artificial smudges in the difference data impair gradient based tracking efforts. The intensity histogram is seen to be significantly distorted for the $10^4 : 3$ case – the artificially introduced peaks make histogram based tracking more difficult.

The subject of standard-compliant video compression specifically optimized for subsequent tracking has been explored as early as [14] in the context of MPEG compression, where the focus is on concentrating (consolidating) bitrate on a Region of Interest (ROI). More recently in [15] a more elaborate approach that adds higher level elements such as motion field correction filtering is proposed in the context of H.263. In [16] a method of using automatic resizing of ROIs detected by video encoder motion estimation in conjunction with object tracking is presented, where the ROI detection relies on motion estimation capturing true motion (and not for example best block match) for good results. In [17] a method of using ROIs to focus limited processing power on highest gain encoder components in the context of H.264 is presented. In [18] an algorithm that specifically does not track individual vehicles, but rather operates in the compressed domain to detect traffic con-

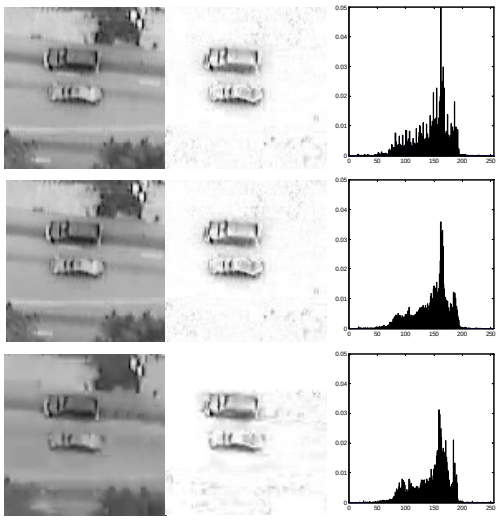


Figure 2.1: Compression effects on vehicle tracking. The top row is a sample of uncompressed video, its error image vs. the background (median frame), and its intensity histogram respectively. The middle row video was compressed at a ratio of $3 : 10^2$, the bottom row at $3 : 10^4$.

gestion. These methods are all low in complexity, but rely on information generated by the encoder (such as motion vectors or macroblock types) to limit computation.

2.3 Vehicle Tracking

The field of video object tracking is quite active, with various solutions offering strength/weakness combinations suitable for different applications. For urban traffic video tracking most applications involve a background subtraction component for target acquisition such as the one developed in [19], and an inter-frame object association component such as the ones developed in [20, 21].

Each of these algorithms has its own strengths and weaknesses, and there is no universally accepted gold-standard object tracking algorithm even in the specific context of traffic surveillance. The computational complexity of object tracking algorithms is the main motivation for our work: if such algorithms were simple enough to deploy on low-cost embedded systems it would

be feasible to perform object tracking directly on raw video data without the need for compression and transmission to a central processing location. In [22] an in-depth study of the processing burden of state-of-the-art video tracking systems, including those proposed in [20] and [21], is presented. The reported complexity of tracking systems analyzed in this study is helpful in illustrating the unfeasibly high cost of implementing such functionality on a multitude of remote nodes.

Applications similar to traffic tracking are also relevant to our discussion. In [23] a survey of video processing techniques for traffic applications is presented, some of which are directly relevant to the pre-processing methods proposed in this work. In [24] a method of vehicle counting for traffic congestion estimation is presented, a capability that would be very useful where only an estimate of congestion (but not higher order statistics) is required. In [25] a review of on-road vehicle detection techniques is presented, where the camera acquiring the video for tracking is not statically elevated over the road but instead located within a vehicle in motion on the road itself. Clearly to be of value such methods need to both be realizable in real-time and to be of complexity manageable by embedded systems feasible for deployment on individual vehicles, making it quite relevant to our low-complexity pre-compression algorithms. In [26] a method of lane estimation and tracking is presented. Lane extraction is of interest to our work in that it can be used to focus our compression resources on video regions of greatest interest, and can even be used to guide compression itself as in [15]. In [27] the presented method of road extraction in aerial images can serve as an example of the challenges and complexity of the problem of road extraction in complex images.

Chapter 3

Utility-Based Segmentation and Coding

3.1 Introduction

In this chapter we propose a method of spatial resource concentration for video compression which is designed to operate on remote nodes in our target system. Given that these remote nodes have low processing power and memory, our algorithm maintains low requirements for both resources. Our target technology is a video tracker, and therefore this algorithm seeks to optimize for tracker performance while minimizing the bitrate required to transmit the compressed video from the remote node to the central processing station.

The subject of standard-compliant video compression specifically optimized for later tracking has been explored as early as [14] in the context of MPEG which focuses on concentrating (consolidating) bitrate on a Region of Interest (ROI). More recently in [15] a more elaborate approach that adds higher level elements such as motion-field correction filtering is proposed in the context of H.263. In [16] a method of using automatic resizing of ROIs detected by video encoder motion estimation in conjunction with object tracking is presented, where the ROI detection relies on motion estimation capturing true motion for good results. In [17] a method of using ROIs to focus limited processing power on highest gain encoder components in the context of H.264 is presented. These methods are all low in complexity, but rely on information generated by the encoder (such as motion vectors or

macroblock types) to limit computation.

We propose a computationally efficient ROI extraction method, which is used during standard-compliant H.264 encoding to consolidate bitrate in regions in video most likely to contain objects of tracking interest (vehicles). The algorithm is low in complexity and requires limited modification of the video compression module. Thus it is easily deployable in non-specialized low processing power remote nodes of centralized traffic video systems. It makes no assumptions about the operation of the video encoder (such as its motion estimation or rate control methods) and is thus suitable for use in a variety of systems.

3.2 Kurtosis-based Region Segmentation

The proposed algorithm optimizes bit allocation for video compression such that the available bitrate is consolidated on regions that are expected to contain objects of tracking interest. The algorithm derives (and maintains) the ROI by a non-parametric model based on the temporal distribution of pixel intensities. The goal is to isolate a map of pixels which in a given analysis window show a sharp intensity variation. Rather than regions undergoing constant change, such as trees, fountains or reflections of the sky, we are interested in regions undergoing periods of dramatic change such as roads, whose intensity changes primarily due to passing cars.

In order to detect such regions we use the kurtosis of intensities for each pixel position over time, defined as

$$\kappa(x) = \frac{\mu_4}{\sigma^4} = \frac{\frac{1}{n}\sum_{i=0}^{n-1}(x_i - \overline{x})^4}{(\frac{1}{n}\sum_{i=0}^{n-1}(x_i - \overline{x})^2)^2} - 3. \tag{3.1}$$

where x is the intensity of a pixel over time at the same spatial position over n samples, and $\overline{x}$ is the mean value of the intensities. By this normalzied definition the Gaussian distribution has an *excess* kurtosis of 0. A higher kurtosis value indicates that the variance of a given distribution is largely due to fewer but more dramatic changes, whereas a lower value indicates that a larger number of smaller changes took place. In this aspect kurtosis, used for a similar method of feature extraction in [28], is a better indicator of the desired behavior than variance.

To identify a threshold that will help us in isolating areas of interest we follow a probabilistic approach in modeling areas of interest. Video capture

noise is modeled as additive Gaussian, which is known to have a kurtosis of 0. Therefore, regions of the scene without motion should have excess kurtosis 0. Movement due to objects such as trees is modeled as a Mixture of Gaussians (excess kurtosis of 0 by the additive property of kurtosis). The desired type of motion will be modeled as a Poisson process, which is commonly used for traffic analysis and is distributed exponentially (with excess kurtosis 6). Therefore we set our model as $X = N + M$, where N is Gaussian noise and M is any movement that occurs on top of it. M is classified as V (motion to be tracked, such as vehicles) or T (motion to be ignored, such as trees). We set $M = \{T\ if\ \kappa(X) \leq threshold,\ else\ V\}$. The ROI is set to 1 for V and 0 for T type pixel positions.

While an online optimization to set the kurtosis threshold is possible within a hypothesis testing framework, given the low computational cost requirement of the system a fixed threshold approach is proposed. We therefore propose to use the threshold of 3, the midpoint between the two models excess kurtosis. Note that this method of modeling traffic as a Poisson process is suitable for common urban and highway traffic, but will not perform well in extreme cases of bumper to bumper congested traffic.

During encoding, for each frame the extracted ROI is used to suppress the Displaced Frame Difference (DFD) that is encoded. This is done by implementing the following change in the rate distortion optimization:

$$m_i = argmin\{w_i^d * Distortion_i + \lambda * Rate_i\} \tag{3.2}$$

where w_i^d is set equal to 0 for areas outside the ROI and equal to 1 for those within. Note that simply skipping macroblocks outside the ROI will cause the decoder to possibly infer motion for these regions given H.264 spatial motion vector prediction. Therefore this step is necessary to code "zero motion" blocks outside the ROI, limiting motion prediction across ROI boundaries via explicitly coded zero motion vectors . While such a binary scheme is not necessarily optimal compared to one with more degrees of flexibility, it is preferable due to the negligible extra computation it adds to the overall system.

3.3 Experimental Results

The video compression experiments presented herein have been performed using original and modified versions of the JM (H.264/14496-10 AVC Refer-

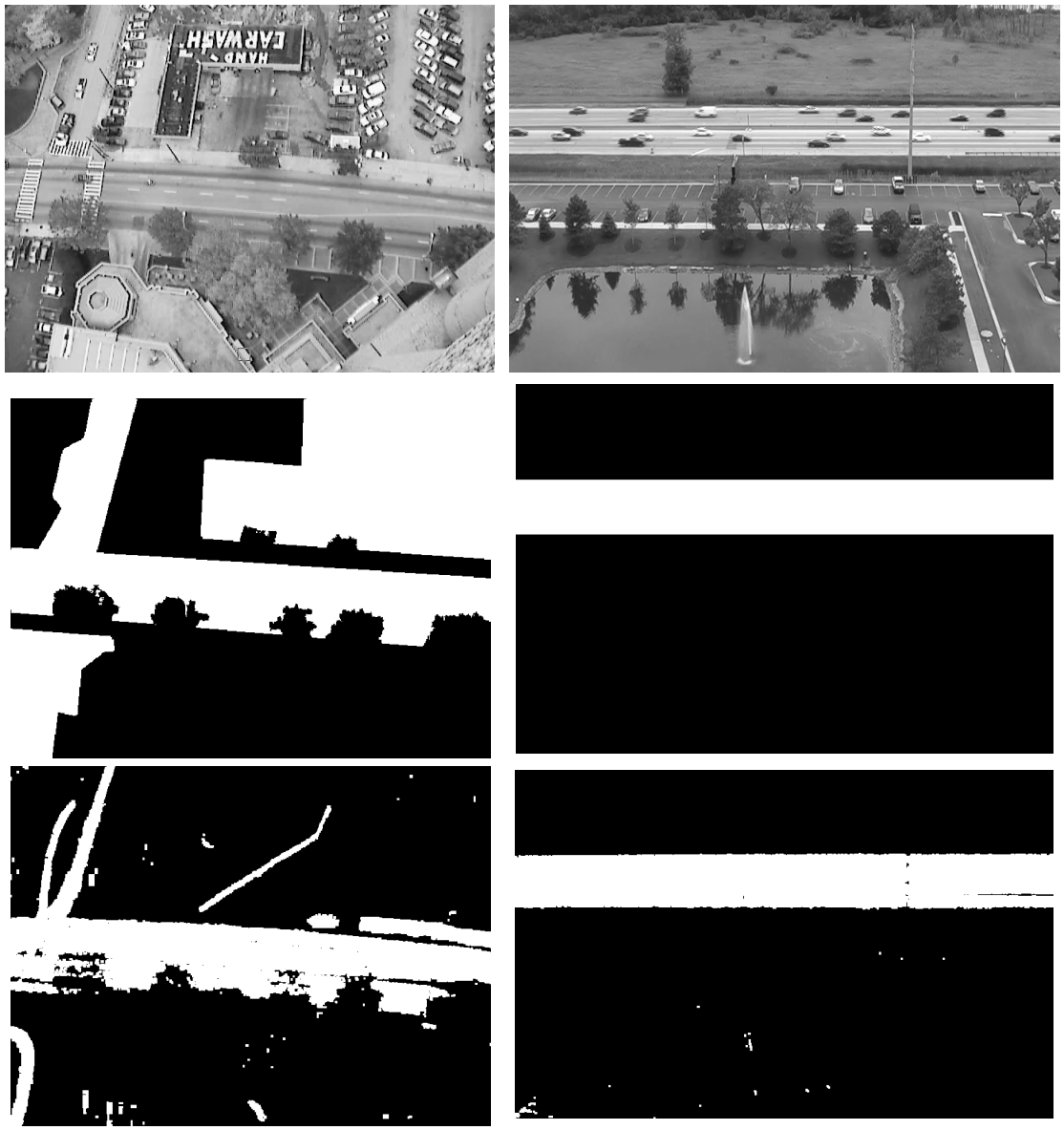


Figure 3.1: Sample frames from "Camera6" and "I-90" sequences (top), their manually segmented ROI for analysis (center) and automatically extracted kurtosis-driven ROI for encoding (bottom).

ence Software) v16.0. Given that the primary interest is in tracking vehicles, in our experiments the reconstructed results are analyzed for performance within the manually derived ROI.

The "I-90" sequence (720x480 @30Hz) was shot on DV tape and is therefore high quality. The "Camera6" content (640x480 @15Hz) was acquired under the NGSIM license courtesy of the US FHWA and was MPEG4 compressed during acquisition, and is significantly noisier. Kurtosis estimation was initialized and updated using 3 second windows (one update per temporal window). While the experiments were executed in MATLAB, the computation and memory requirements are low enough for mobile and embedded platform implementations. The modifications to the H.264 encoder were compartmentalized enough to make adding the algorithm to mature products feasible.

In Fig. 3.1 we show some sample detected and manually extracted ROI. Note that in the figure "I-90" has a detected ROI much closer to the manually extracted version than "Camera6" – this is because the observer manually extracting the ROI was asked to mark "areas of interest to urban traffic", whereas the kurtosis-based ROI detection algorithm accumulates areas where cars have actually been to within its analysis window. This difference is a benefit for the detector in that it focuses the ROI to a region where activity has been reported and not a region where activity could theoretically take place in the future.

In order to analyze total distortion in tracking we focus separately on two separate metrics: one to measure the degradation of a trackers ability to find targets on each frame and the other to its ability to associate these targets as the same object across frames. For the first the "Bounding Box Overlap Ratio" (BBOR) metric is used. This metric maintains a simple median background model (updated once per window), which it uses for background subtraction. The resulting foreground on each frame is thresholded using the method presented in [29, 30] and processed with morphological operators before bounding boxes (BB) are extracted. For comparing sequences S_1 (baseline) and S_2 (compressed), the BBOR is defined as $BBOR = \frac{|BB(S_1) \cap BB(S_2)|}{|BB(S_1)|}$, where $\cap$ denotes the intersection and $||$ the cardinality of the sets. Since our main interest is in tracking vehicles, the manual ROI, which corresponds to regions vehicles can be found such as roads and parking lots, is used to mask the video after compression. In our experiments this simulates a specialized tracker which targets only vehicles.

A higher value of the BBOR indicates that targets (not necessarily the same targets from frame to frame) were found in more similar spatial locations between the two sequences being compared. In Fig. 3.2 BBOR results comparing pre-compression performance to that of default encoding vs. encoding focusing on detected and manual ROIs are presented. Note that at higher bitrates our algorithm provides significant bitrate reduction given encoder sensitivity to noise and peripheral "uninteresting" motion (trees, fountains) – bitrate savings of up to 75% for "I-90" and 50% for "Camera6" were seen with negligible difference in BBOR. While such large savings are not maintained at lower bitrates, even at the lowest analyzed bitrate results never show below 5-10% savings. The larger savings seen in "I-90" compared to "Camera6" can be attributed to "I-90" having a simpler and smaller ROI and with smaller disparity between the detected and manually extracted ROIs.

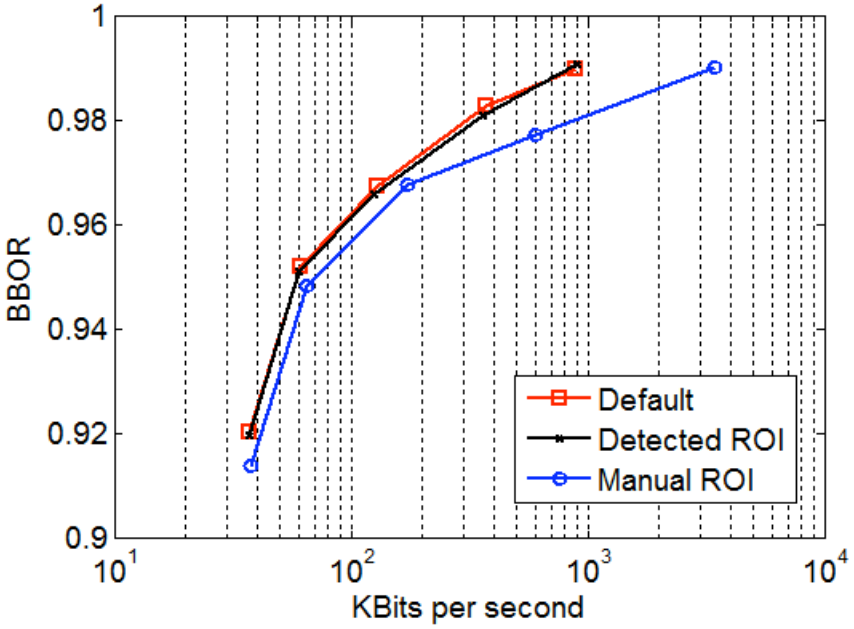
For the second analysis the "Mean Shift" tracking method proposed in [20] and implemented in the OpenCV project available in [31] is used. The metrics used in this case are number of "false positives" and "false negatives". Given that various traffic tracking applications can prefer one type of error to the other a separate analysis is presented for each. Note that the measurements for these metrics are done on an observation basis, and while the experiments have been controlled by averaging repeated tests some degree of subjective variability is expected. In Figs. 3.3 and 3.4 the number of errors in sample Mean Shift tracking in uncompressed and compressed sequences are shown. Note that in all cases an increase in errors is observed for the mid-range bitrates, where the error numbers go up from high to mid rates and then back down for the low rates. This behavior can be attributed to the smoothing effect of coarse quantization removing error-causing features from the video as the bitrate goes down. It is interesting to observe that the increase in errors corresponds to 100Kbps - 1Mbps range, which is the operating space that would be commonly used for acceptable visual quality applications. Also note that for the "Camera6" sequence, where the detected and manual ROIs differ, the detected ROI mostly outperforms the manual ROI.

In [32] a quality metric is proposed for tracking that combines scores for edge sharpness, color histogram preservation and motion boundary sharpness of tracked silhouettes. While this score also covers all features most significantly degraded by video compression, our metrics were chosen for their simplicity. Complex metrics which analyze the sharpness of target segmentation or the stability of inter-frame association are available but not

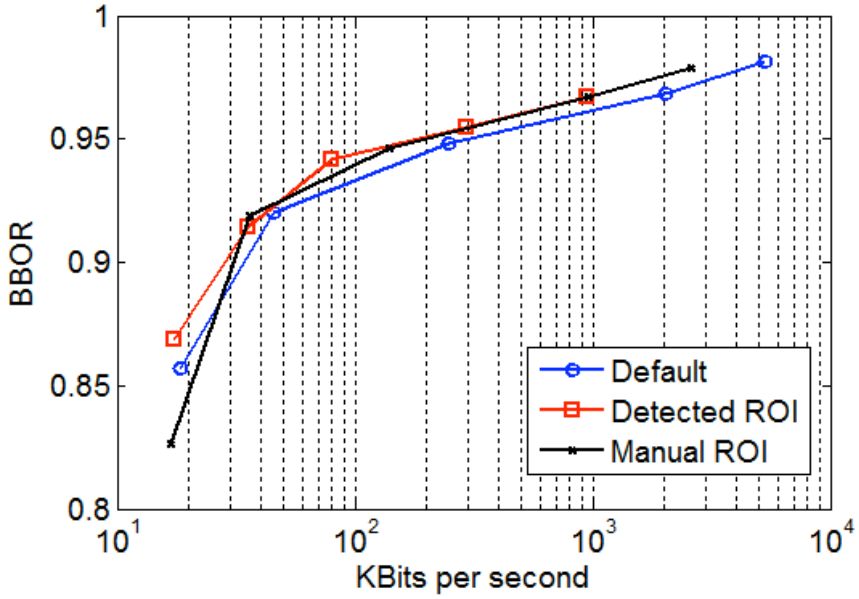
universal.

3.4 Conclusions

We have proposed a novel method of using pixel intensity kurtosis to consolidate video compression bitrate on an ROI incorporating tracked object trajectories. We have demonstrated that such an approach can lead to up to 75% bitrate savings for comparable tracking performance, and have shown that an ROI derived by our method of extraction results in performance close to a manually derived one. The reduction in required bandwidth coupled with its relatively low processing and memory overhead make the algorithm attractive for deployment on remote nodes of centralized traffic video tracking applications. The next step is the derivation of online low-complexity optimization methods for the kurtosis threshold and the number of frames needed in the analysis window.

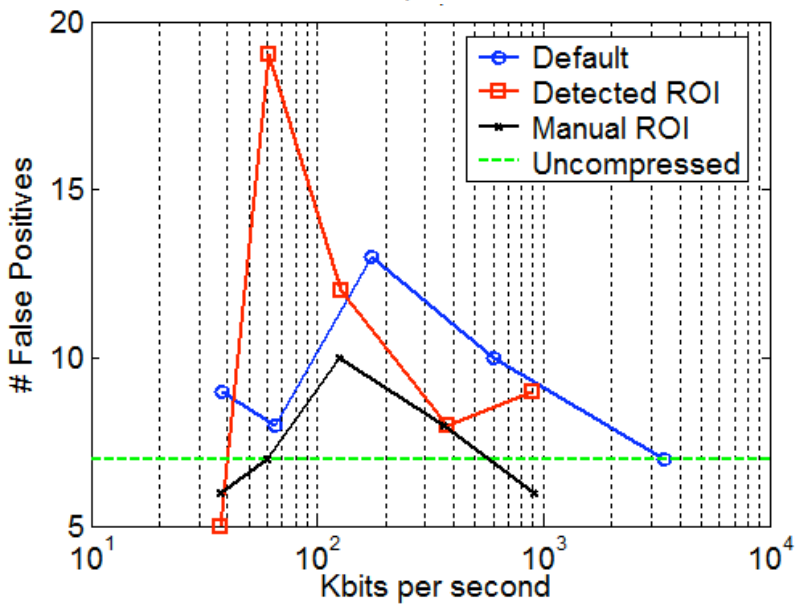


(a) "I-90" BBOR

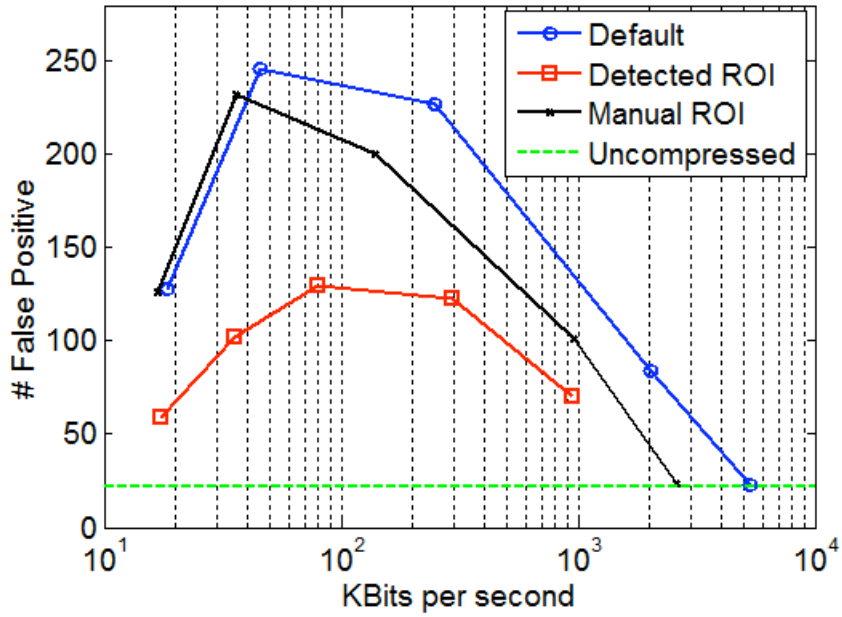


(b) "Camera6" BBOR

Figure 3.2: Bitrate vs BBOR for "I-90" and "Camera6" sequences.

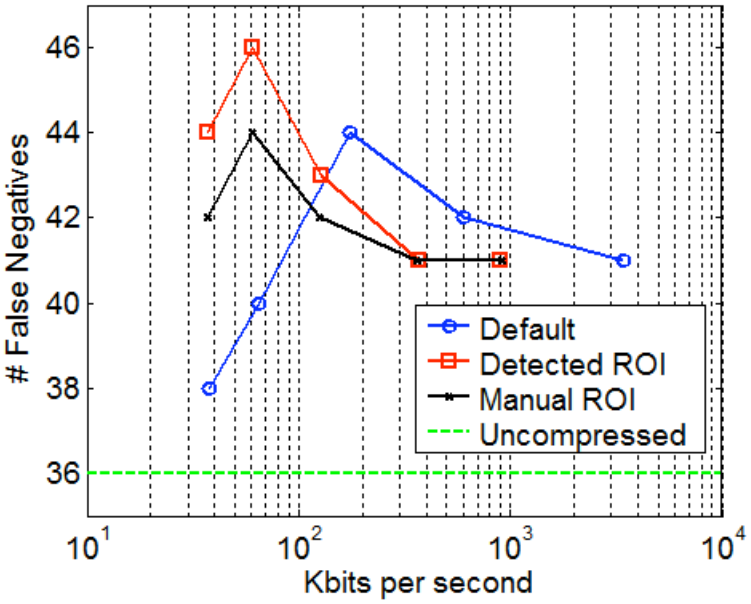


(a) "I-90" false positives

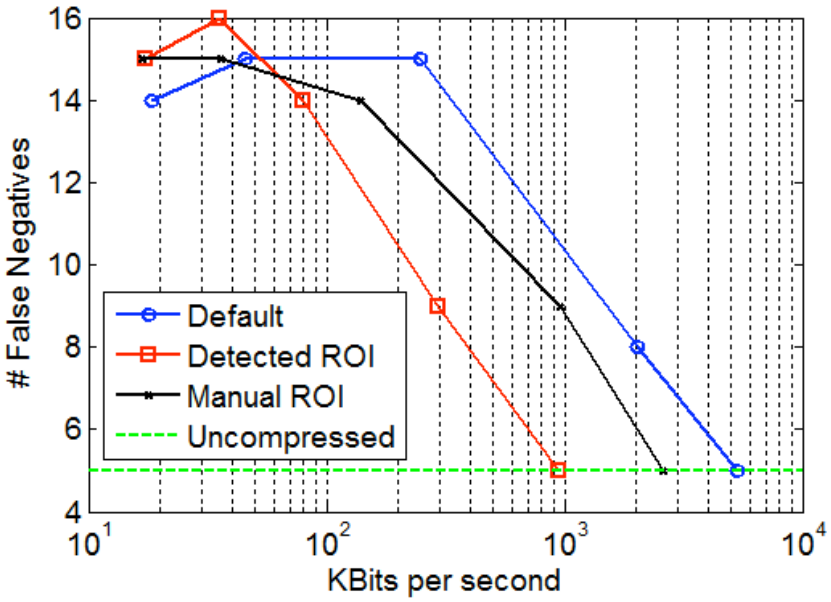


(b) "Camera6" false positives

Figure 3.3: "I-90" and "Camera6" tracking false positive errors as a function of bitrate.



(a) "I-90" false negatives



(b) "Camera6" false negatives

Figure 3.4: "I-90" and "Camera6" tracking false negative errors as a function of bitrate.

Chapter 4

Tracking-Optimal Transform Coding

4.1 Introduction

In this chapter we present an algorithm to optimize tracking accuracy for a given bitrate by concentrating available bits in the frequency domain on the features most important to tracking. We also present a tracking accuracy metric which is more advanced than that used in Chapter 3, combining multiple pertinent metrics into a single measure which we use to iteratively drive optimization. Our proposed algorithm is similar to the trellis-based R-D optimization presented in [33] in that it seeks to optimize for a given target by manipulating quantized transform coefficients. However in our work we optimize for tracking accuracy rather than fidelity, and work on a sequence level as opposed to an individual transform level. This work is to appear in [8].

Given the special parameters of centrally controlled traffic surveillance systems, it is necessary to limit resource requirements, such as for memory and processing power, for any technique seeking to counter the effects of video distortion on tracking. Our algorithm is low in complexity and is readily deployable as a simple modular add-on to low processing power remote nodes of centralized traffic video systems. It makes no assumptions about the operation of the video encoder (such as its motion estimation or rate control methods) and is thus suitable for use in a variety of systems. The resulting bitstreams are standard-compliant, thereby guaranteeing interoperability

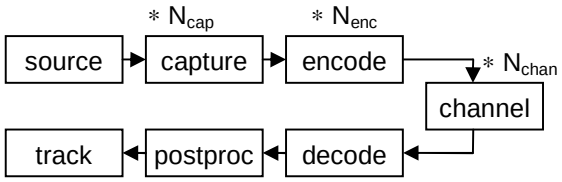


Figure 4.1: Typical centrally controlled tracking system. Video of objects to be tracked is acquired (with capture noise N_{cap}) at a remote location, compressed (with encoding distortion N_{enc}), and transmitted over a channel (with channel distortion N_{chan}). At the receiver the transmission is decoded, post-processed and passed on to tracker.

with other standard-compliant systems.

4.2 Frequency Decomposition of Tracking Features

The active field of video object tracking contains a large variety of algorithms, yet most of these systems share some fundamental concepts. In reviews of object tracking presented in [12] and [13] it is shown that most algorithms operate by modeling and segmenting foreground and background objects. Once the segmentation is complete and the targets located, the targets are tracked across time based on key features such as spatial edges, color histograms and detected motion boundaries. The segmentation models and key features for a particular tracking application are chosen based on the application's goals and parameters. For example, color histograms can be useful when tracking highway vehicle activity during the day, but can be less useful under low light conditions at night.

Compression artifacts are especially debilitating for video tracking applications. In a scenario where the video is distorted, the performance of the tracking algorithm may suffer as the foreground/background models become not as realistic and key tracking features difficult to identify. In Fig. 4.1 a typical centrally controlled tracking system is shown, where the video is captured at a remote location and must be transmitted to a central location for processing. Here the compressed video stream is decoded and post-processed to remove as much distortion as possible, and then tracking

is performed. Such a separation of the capture and processing locations of video is required in systems where many sources of video exist (streets, intersections, strategic locations) yet the processing power required to process the video on-site at each location would be prohibitively costly. Therefore a central processing location where all the video is sent is required. While the distortion N_{cap} from the video acquisition process is inherent to any video system, the distortion introduced by the video compression and lossy channel transmission (N_{enc} and N_{chan}) are specific to such centrally controlled systems.

The introduction of measures to alleviate the effects of distortion during encoding, transmission and post-processing is challenging given the different types of distortion, the parameters of which may also vary across time. In the highway vehicle tracking example, N_{cap} and N_{enc} may vary based on lighting conditions, and if a non-dedicated channel such as WiFi is used N_{chan} will vary based on signal reception and traffic congestion. Therefore any measures meant to alleviate distortion effects need to either account for all such variations in advance or be adaptive to each variation.

In order to optimize for tracking quality a metric to measure tracking accuracy is required. In [34] a state-of-the-art review for video surveillance performance metrics is presented. Due to their pertinence in traffic surveillance for our work we choose the Overlap, Precision and Sensitivity metrics presented therein. *Overlap* (OLAP) is defined in terms of the ratio of the intersection and union of the Ground Truth (GT) and Algorithm Result (AR) objects,

$$OLAP = \frac{GT_i \cap AR_i}{GT_i \cup AR_i}, \tag{4.1}$$

where GT_i are the segmented objects tracked in uncompressed video, the AR_i those tracked in compressed video, $\cap$ the intersection of the two regions and $\cup$ their union. *Precision* (PREC) is defined in terms of the average number of True Positives (TPs) and False Positives (FPs) per frame as

$$PREC = \frac{\overline{TP}}{\overline{TP} + \overline{FP}}, \tag{4.2}$$

where TPs are objects present in both the GT and AR, while FPs are objects present in the AR but not in the GT. An FP is flagged if an object detected in the AR does not overlap and equivalent object in the GT

($OLAP(AR_i, GT_i) = 0$). *Sensitivity* (SENS) is defined in terms of TPs and False Negatives (FNs) as

$$SENS = \frac{\overline{TP}}{\overline{\overline{TP} + \overline{FN}}}, \tag{4.3}$$

where FNs are objects present in the GT but not in the AR. An FN is flagged if an object detected in the GT does not overlap and equivalent object in the AR ($OLAP(GT_i, AR_i) = 0$). We define the aggregate tracking accuracy A as

$$A \;\; = \;\; (\alpha * OLAP) + (\beta * PREC) + (\gamma * SENS), \tag{4.4}$$

where α, β and γ are weighting coefficients. Given that OLAP, SENS, PREC are all in the range [0 1], no normalization of A is necessary as long as $\alpha + \beta + \gamma = 1$.

4.3 Iterative Quantization Table Optimization

The proposed algorithm seeks to optimize video compression in the system to adaptively maximize performance under the varying effects of distortion. To limit the scope of our discussion we will consider only N_{cap} and N_{enc}, disregarding N_{chan}. We assert that any given tracking algorithm uses one or more features that play a greater role in its success than other features. Each of these features is subject to N_{cap} and N_{enc}, possibly as governed by different functions based on the nature of distortion – for example, a blurring N_{cap} may impact edges but not color histograms. We further assert that there exist undesirable features (such as those introduced by noise) that confuse tracking efforts and actively detract from tracking accuracy while still consuming bits to be represented in the compressed video. All of these features are each coherently represented in the frequency domain by one or more of the spatial transform filters used in hybrid video coding, an example of which is shown in Fig. 4.2. The basis functions shown in the figure are those used for the 4x4 transform in the H.264/AVC video coding standard – observe that each coefficient's corresponding basis sharpens vertical and/or horizontal edges to varying degrees, with the exception of the 0-index "DC" basis which sets the mean value. Also observe that by their nature each basis will represent some

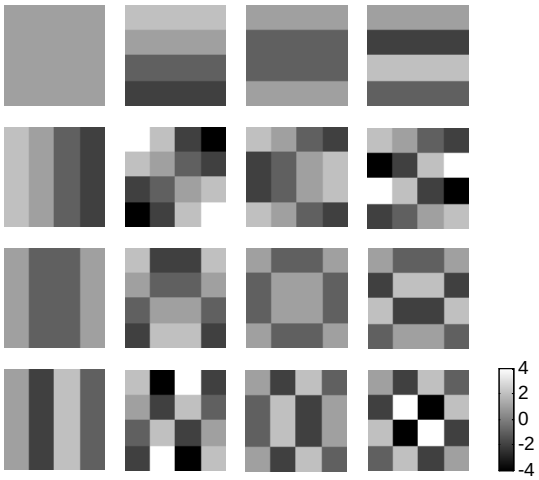


Figure 4.2: Transform coefficients represented as per-coefficient basis functions applied to the source 4x4 block. From left to right, top to bottom, the coefficient indices are numbered 0,1,2..15.

feature more effectively than others, while at the same time not representing other features at all – this observation will be key to our optimization.

Our algorithm automatically identifies and concentrates compression bitrate on frequencies useful to tracking, at the cost of bitrate allocated to frequencies confusing or useless to tracking. We perform our optimization by manipulating the quantization of coded transform coefficients. The quantization scheme is varied via the Quantization Table (QT) specified as part of the Sequence and Picture Parameter Set structures in the H.264/AVC video compression standard. Each entry of the QT is used to quantize a coefficient resulting from the 4x4 spatial transform depicted in Fig. 4.2 – the goal is to spend the fewest bits on those coefficients containing the least useful information pertaining to the features used by the tracker.

Refer to [35] for a description and to [36] for a detailed explanation of the H.264/AVC frequency transform. The standard specifies quantization for a given transform coefficient index q_{idx} in terms of the quantization point (QP) and the QT as

$$\begin{aligned} QT &= [t_0, t_1, t_2, ...t_{15}] \\ QP_{idx} &= QP * \left(\frac{1}{16}\, QT[idx]\right). \end{aligned} \tag{4.5}$$

Integers in the range [0-255] (8 bits) are allowed for each entry to signify a multiplicative per-coefficient modification in the range $[\frac{1}{16}\ 16]$. The probability space for our optimization is therefore of dimension 256^{16} for a single quantizer. Given the large number of costly evaluations that would have to be tried in an exhaustive approach we proceed using Lagrangian optimization. Based on a chosen set of tracking accuracy criteria, we will iteratively coarsen quantization of frequencies less useful to tracking, thereby saving more bits per accuracy reduced than if we simply coarsened quantization uniformly across all frequencies.

The optimization is performed by iteratively generating a set of operating points (OPs), characterized by their bitrate R and accuracy A, and selecting a subset of these considered superior in performance. These "iteration optimal" OPs form the basis of the subsequent iteration, whose OPs are generated by modifying the parameters of the previous iterations optimal OPs. The algorithm is said to converge when the set of optimal OPs does not change between two subsequent iterations. The ultimate goal is to generate a rate-accuracy curve allowing the user to specify a bitrate and receive a QT which will maximize tracking accuracy.

We define the uniform QT $T_{init} = [255, 255...255]$, which attenuates all frequencies at the maximum allowed level. The iteration optimal set S_{opt} is defined as the strictly increasing set of rate-accuracy pairs which include the lowest bitrate in the set,

$$\begin{aligned} S_{opt} &\rightarrow (A_k < A_{k+n} | R_k < R_{k+n}) \ \forall\ n, k \\ S_{opt}[0] &= argmin\{R_k\} \ \forall\ k, \end{aligned} \tag{4.6}$$

where k and $k+n$ are indices into the set of available OPs. The QT relaxation function Φ is defined as

$$\Phi\{T, idx, C\} = T[t_0, t_1, t_2, ... \frac{t_{idx}}{C}, ...t_{15}]. \tag{4.7}$$

To initialize our optimization set we generate the OPs obtained by relaxing each entry in T_{init} and applying the result across a given range of quantizers. Of these results we choose the optimal subset $S_{0,opt}$, which forms the basis of the first iteration. For each subsequent iteration i, each point on $S_{i-1,opt}$ is revisited by relaxing entries in their QTs, forming the set of OPs S_i from which the optimal set $S_{i,opt}$ is drawn. Refer to Fig. 4.3 for a

sample iteration. The set of OPs S_0 (circles) are generated, and only the elements of S_0 which lie on the strictly increasing $S_{0,opt}$ curve are revisited to form S_1 (crosses). Thereafter only those members of S_1 which lie on $S_{1,opt}$ are revisited for S_2 (triangles). The resulting set S_2 contains OPs superior to those on $S_{1,opt}$, and therefore the algorithm will continue to iterate a third time using an $S_{2,opt}$ to populate S_3.

Given that each iteration only a single QT entry can be modified per OP, the theoretical worst-case convergence bound will involve a maximum of $\frac{255}{C}$ iterations. Each iteration i can evaluate a maximum of 16^i OPs. While this worst case set already involves close to 20 orders of magnitude fewer evaluations than the exhaustive search, given the highly unlikely nature of the worst case it is expected for our algorithm to converge with significantly fewer evaluations than the worst case allows for. Where a strict convergence time requirement shorter than the worst case exists, the number of iterations allowed can be set to a fixed ceiling for a faster resolution guarantee.

Note that the optimization must be performed simultaneously for a range of base quantizers, as tracking is a nonlinear process subject to different distortions at each quantization level. It is possible that a finer quantized OP may result in worse tracking performance due to the introduction of noise elements which were effectively filtered out with coarser quantization. Any non-iterative effort to optimize quantization in this sense would require accurate models of the video content and all sources of distortion, taking into account all variations across time. Our iterative process allows for per-coefficient quantization optimization without such difficult and error-prone modeling.

A core assumption of our algorithm is that the distortion process of key tracking features is stationary for a given video source, at least over sufficiently long periods of time where re-initialization of the optimization to rebuild the optimal QT each time the distortion process changes is feasible. Such change detection would need to be provided externally, for example via light sensors to detect nightfall or via frame histograms to detect inclement weather.

One limitation of our search method is that it is "greedy," considering only single hop modifications to $S_{i-1,opt}$ when populating S_i. This limitation introduces sparsity in the set of OPs that can be reached, making it possible for the converged S_{opt} to be suboptimal compared to an exhaustive solution. While this issue can be readily circumvented by allowing for multi-hop projections when populating S_i, the additional computational burden to do so

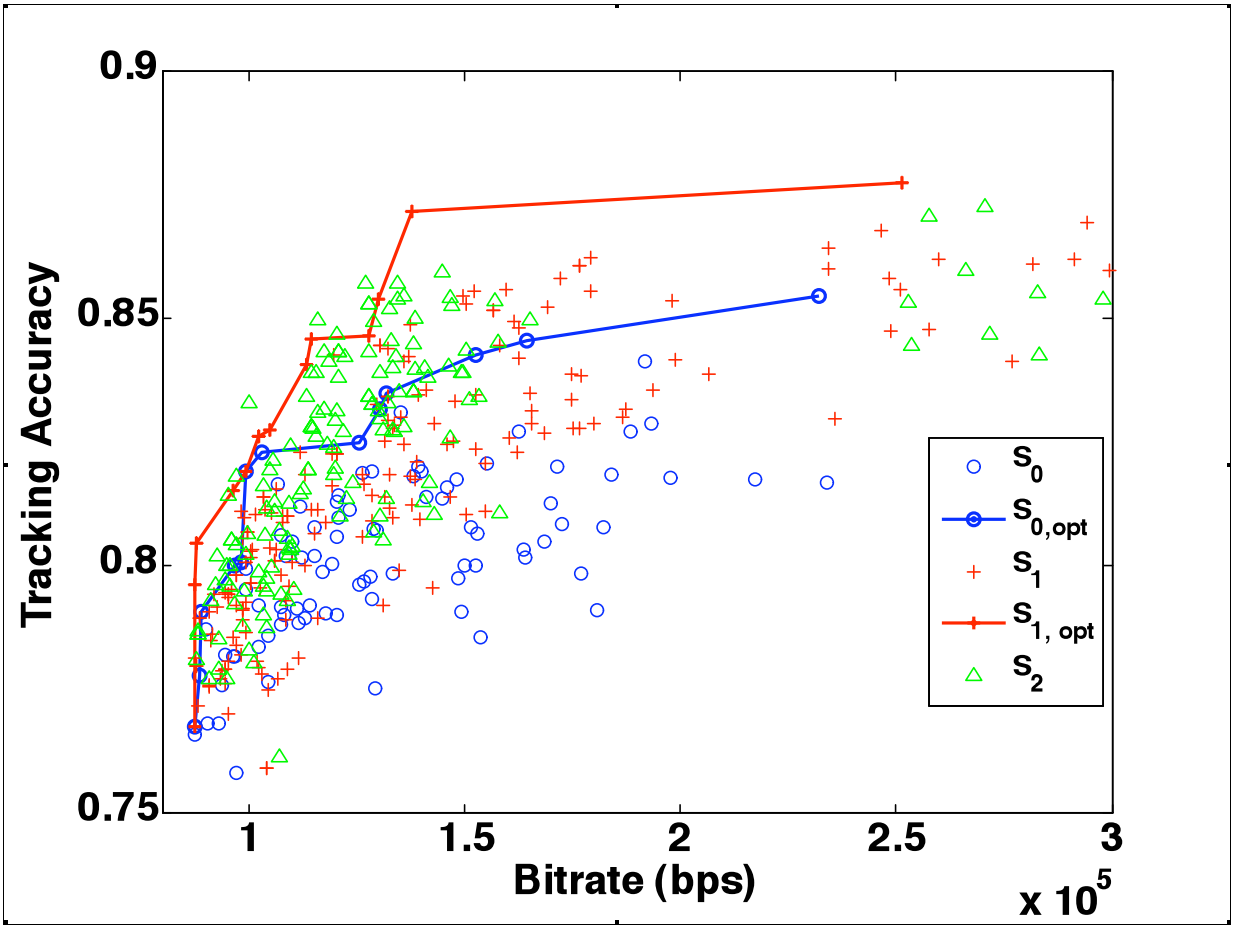


Figure 4.3: An example showing the first three iterations of the optimization process in the rate-accuracy domain.

will be unacceptably high for most low-cost embedded devices.

An point related to implementation is that the algorithm requires access to the ground truth for operation. In a centrally controlled system such as described in Fig. 4.1 this will not be available. However, a very close approximation can be obtained by compressing the video sample at high bitrates and transmitting it at channel capacity over a slower than real-time interval before starting the optimization. If this is done such a process would have to run in series with the optimization, thus adding to the initialization time requirement.

4.4 Experimental Results

The video compression experiments presented herein have been performed using the open-source H.264/AVC encoder x264 [37]. The "I-90" and "Golf" sequences (720x480 @30Hz) were shot on DV tape and are therefore high quality sources. 600 frames (20 seconds) of each sequence were compressed using a common QP set of [25, 26, 27, 28, 29, 30] and uniform QTs $T_j = 16 \rightarrow j = [0, 1, ...15]$. The resulting video was used for tracking, and the results were put through an "iteration optimal" criterion as described in Section 4.3 to generate the "optimal" uniform quantization performance curve.

For our experiments, the post-processing block shown in Fig. 4.1 involves manually segmenting the road to help automated tracking – segmentation is performed once and used for all cases where the content was utilized. The open-source OpenCV "blobtrack" module available at [31] was used as the object tracker.

Refer to Fig. 4.4 for results from experiment using the "I-90" sequence (lightly congested highway traffic) and the Mean Shift tracker described in [20]. The algorithm was allowed to run for 4 iterations, evaluating a total of 587 OPs. Note that at the higher bitrates close to 40% bitrate savings for comparable accuracy tracking is possible using our algorithm. Also note the gradual improvement in performance among curves $S_{opt,1}$, $S_{opt,2}$ and $S_{opt,3}$, each increasingly superior to the uniform quantized OPs of $S_{opt,flat}$.

Refer to Fig. 4.5 for results from experiment using the "Golf" sequence (average congested local intersection) and the "Connected Component" tracker described in [21]. The algorithm was allowed to run for 3 iterations, evaluating a total of 447 OPs. The lower overall tracking accuracies compared to those in Fig. 4.4 are due to more challenging tracking video being used.

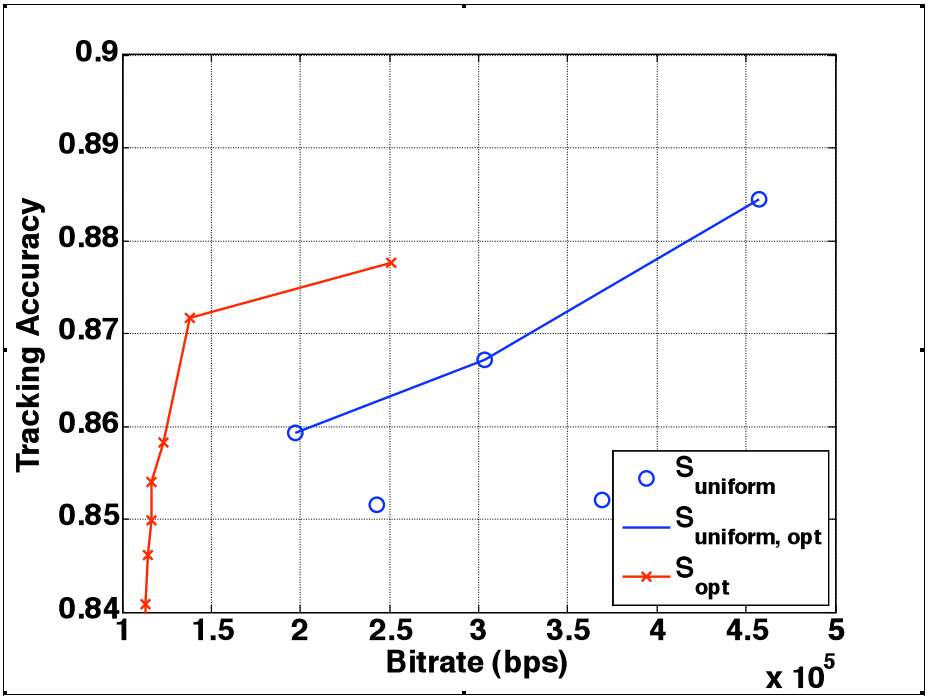


Figure 4.4: Rate-accuracy results for the "I90" sequence and "Mean Shift" tracking.

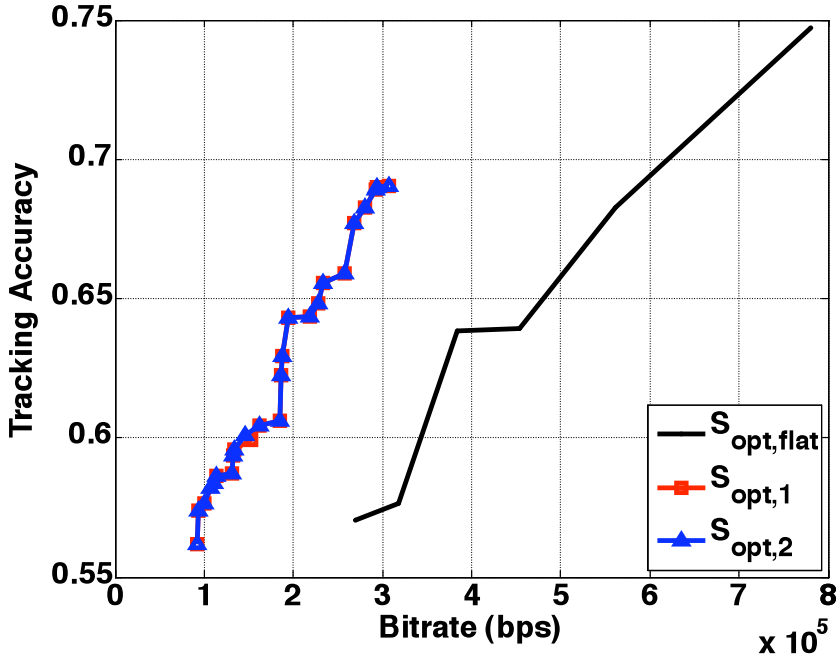


Figure 4.5: Rate-accuracy results for the "Golf" sequence and "Connected Component" tracking.

Note that at lower bitrates savings exceeding 60% in bitrate can be realized with just 3 iterations, and that as early as $S_{opt,2}$ the algorithm has almost converged. Also note that here a completely different tracker than the one in Fig. 4.4 has been used on content of a different nature (hard to track traffic intersection as opposed to easier to track highway content). Consistent improvement across such different content and trackers clearly demonstrates the adaptability of the algorithm.

The computation and memory requirements of the algorithm are low enough for mobile and embedded platform implementations. Given that the Lagrangian search can be done offline and needs to be performed only once per system initialization or reset (triggered manually or due to a large change in conditions), any system that can perform real time encoding at remote nodes and tracking at the central node can reasonably complete the optimization process in a matter of minutes.

4.5 Conclusions

We have proposed a novel method of optimizing object tracking quality in compressed video through quantization tables. We have demonstrated using two common object tracking algorithms that our algorithm allows for over 60% bitrate savings while maintaining comparable tracking quality.

Chapter 5

Conclusion

In this report we have discussed the various technologies that when used individually or in conjunction with each other implement the iTRAC system. Used individually each algorithm can provide up to 75% savings in bitrate required to transmit traffic surveillance video with comparable automated tracking quality. Therefore for real-world traffic surveillance applications featuring automated tracking, the bitrates required by systems using iTRAC could be deployed over existing 3G or WiMAX wireless links, allowing ubiquitous coverage at reasonable cost.

The results for this project were published in [7, 8, 38, 39].

Bibliography

[1] A. Chatziioanou, S. L. M. Hockaday, S. Kaighn, and L. Ponce, "Video image processing systems: applications in transportation," *Vehicle Navigation and Information Systems Conference*, vol. 38, pp. 17–20, 1995.

[2] A. Chatziioanou, S. L. M. Hockaday, S. Kaighn, and L. Ponce, "Video content analysis moves to the edge," tech. rep., IMS Research, January 2007.

[3] M. Kyte, A. Khan, and K. Kagolanu, "Using machine vision (video imaging) technology to collect transportation data," *Innovations in Travel Survey Methods*, vol. Transportation Research Record No. 1412, 1995.

[4] V. Kovali, V. Alexiadis, and P. Zhang, "Video-based vehicle trajectory data collection," *Transportation Research Board Annual Meeting*, 2007.

[5] N. Zingirian, P. Baglietto, M. Maresca, and M. Migliardi, "Customizing MPEG video compression algorithms to specific application domains: The case of highway monitoring," *Transportation Research Board Annual Meeting*, pp. 46–53, 1997.

[6] J. Versavel, "Traffic data collection: Quality aspects of video detection," *Transportation Research Board Annual Meeting*, 2007.

[7] E. Soyak, S. A. Tsaftaris, and A. K. Katsaggelos, "Content-aware H.264 encoding for traffic video tracking applications," *Proceedings of ICASSP*, March 2010.

[8] E. Soyak, S. A. Tsaftaris, and A. K. Katsaggelos, "Quantization optimized H.264 encoding for traffic video tracking applications," *(to appear) Proceedings of ICIP*, September 2010.

[9] A. J. Lipton, C. H. Heartwell, N. Haering, and D. Madden, "Critical asset protection, perimeter monitoring, and threat detection using automated video surveillance," tech. rep., ObjectVideo Inc., 2002.

[10] P. Eaton, "The hidden costs of video surveillance," tech. rep., Recon Systems Inc., 2007.

[11] "Roads in oakland county are safer and less congested thanks to wi4 fixed solutions," tech. rep., Case Study by the Road Commission for Oakland County, 2007.

[12] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Computing Surveys*, vol. 38, pp. 13.1–13.45, 2006.

[13] P. F. Gabriel, J. G. Verly, J. H. Piater, and A. Genon, "The state of the art in multiple object tracking under occlusion in video sequences," *Proceedings of ACIVS*, pp. 166–173, 2003.

[14] N. Zingirian, P. Baglietto, M. Maresca, and M. Migliardi, "Video object tracking with feedback of performance measures," *Proceedings of ICIAP*, vol. 2, pp. 46–53, 1997.

[15] W. K. Ho, W. Cheuk, and D. P. Lun, "Content-based scalable h.263 video coding for road traffic monitoring," *IEEE Transactions on Multimedia*, vol. 7, no. 4, 2005.

[16] R. D. Sutter, K. D. Wolf, S. Lerouge, and R. V. de Walle, "Lightweight object tracking in compressed video streams demonstrated in region-of-interest coding," *EURASIP Journal on Advances in Signal Processing*, 2007.

[17] A. K. Kannur and B. Li, "Power-aware content-adaptive h.264 video encoding," *Proceedings of ICASSP*, vol. 00, pp. 925–928, 2009.

[18] F. Porikli and X. Li, "Traffic congestion estimation using HMM models without vehicle tracking," *IEEE Proceedings on Intelligent Vehicles*, 2004.

[19] S. Cheung and C. Kamath, "Robust techniques for background subtraction in urban traffic video," *Proceedings of VCIP*, vol. 5308, no. 1, pp. 881–892, 2009.

[20] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," *Proceedings of CVPR*, vol. 2, pp. 142–149, 2000.

[21] A. Senior, A. Hampapur, Y.-L. Tian, L. Brown, S. Pankanti, and R. Bolle, "Appearance models for occlusion handling," *Proceedings of the 2nd IEEE Workshop on PETS*, December 2001.

[22] T. P. Chen, H. Haussecker, A. Bovyrin, R. Belenov, K. Rodyushkin, A. Kuranov, and V. Eruhimov, "Computer vision workload analysis: Case study of video surveillance systems," *Intel Technology Journal*, vol. 9 (12), May 2005.

[23] V. Kastrinaki, M. Zervakis, and K. Kalaitzakis, "A survey of video processing techniques for traffic applications," *Image and Vision Computing*, vol. 21, no. 4, pp. 359–381, 2003.

[24] E. Bas, A. M. Tekalp, and F. S. Salman, "Automatic vehicle counting from video for traffic flow analysis," *IEEE Transactions on Intelligent Transportation Systems*, June 2007.

[25] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection: A review," *IEEE Transactions on Pattern Analysis And Machine Intelligence*, vol. 28, May 2006.

[26] J. McCall and M. M. Trivedi, "Video based lane estimation and tracking for driver assistance: Survey, system, and evaluation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, pp. 20–37, March 2006.

[27] M. Barzohar, L. Preminger, T. Tasdizen, and D. B. Cooper, "Robust method for completely automatic aerial detection of occluded roads with new initialization," *Proceedings of SPIE, Infrared Technology and Applications XXVII*, vol. 4820, pp. 688–698, January 2003.

[28] A. Briassouli, V. Mezaris, and I. Kompatsiaris, "Video segmentation and semantics extraction from the fusion of motion and color information," *Proceedings of ICIP*, vol. 3, pp. 365 – 368, 2007.

[29] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 9, pp. 62–66, 1975.

[30] M. Luessi, M. Eichmann, G. Schuster, , and A. Katsaggelos, "Framework for efficient optimal multilevel image thresholding," *Journal of Electronic Imaging*, vol. 18, Jan. 2009.

[31] "http://opencv.willowgarage.com."

[32] C. Erdem, A. M. Tekalp, and B. Sankur, "Video object tracking with feedback of performance measures," *IEEE Transactions on Circuits And Systems for Video Technology*, vol. 13, pp. 310–324, 2003.

[33] J. Wen, M. Luttrell, and J. Villasenor, "Trellis-based R-D optimal quantization in H.263," *IEEE Transactions on Image Processing*, vol. 9, pp. 1431–1434, August 2000.

[34] M. B. A. Baumann, J. Ebling, M. Koenig, H. S. Loos, W. N. M. Merkel, J. K. Warzelhan, and J. Yu, "A review and comparison of measures for automatic video surveillance systems," *EURASIP Journal on Image and Video Processing*, 2008.

[35] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, July 2003.

[36] H. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky, "Low-complexity transform and quantization in H.264/AVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 598–603, July 2003.

[37] "http://www.videolan.org/developers/x264.html."

[38] E. Soyak, S. A. Tsaftaris, and A. K. Katsaggelos, "Tracking-optimal pre- and post-processing for H.264 compression in traffic video surveillance applications," *Proc. ICECS*, Dec. 2010.

[39] E. Soyak, S. A. Tsaftaris, and A. K. Katsaggelos, "Low-complexity video compression for automated transportation surveillance," *submitted, IEEE Transactions on Circuits And Systems for Video Technology.*