CONTENT-AWARE H.264 ENCODING FOR TRAFFIC VIDEO TRACKING APPLICATIONS

E. Soyak [a], *S. A. Tsaftaris* [a,b] *and A. K. Katsaggelos* [a]

[a] Department of Electrical Engineering and Computer Science, Northwestern University
2145 Sheridan Rd., Evanston, IL 60208, USA
[b] Department of Radiology, Feinberg School of Medicine, Northwestern University,
737 N. Michigan Avenue Suite 1600, Chicago, IL 60611
email: {e-soyak, s-tsaftaris}@northwestern.edu, aggk@eecs.northwestern.edu

ABSTRACT

The compression of video can reduce the accuracy of tracking algorithms, which is problematic for centralized applications that rely on remotely captured and compressed video for input. We show the effects of high compression on the features commonly used in real-time video object tracking. We propose a computationally efficient Region of Interest (ROI) extraction method, which is used during standard-compliant H.264 encoding to concentrate bitrate on regions in video most likely to contain objects of tracking interest (vehicles). This algorithm is shown to significantly increase tracking accuracy, which is measured by employing a commonly used automatic tracker.

Index Terms— Urban traffic video tracking, ROI extraction, video compression, kurtosis

1. INTRODUCTION

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 bitrate requirements this means either expensive wired communication links or the video data being heavily compressed to not exceed the allowed communications bandwidth. Although MPEG-2 is the most common deployed standard for such applications, recently H.264 has started to be used, significantly reducing the link bandwidth requirement. However, most video compression algorithms are stil not optimized for traffic video data, nor do they take into account the possible data analysis that will follow at the control center. As a result of compression the visual quality of the data will suffer, but more importantly the tracking accuracy and efficiency are severely affected.

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 [7], and an inter-frame object association component such as the one developed in [3].

The subject of standard-compliant video compression specifically optimized for later tracking has been explored as early as [5] in the context of MPEG which focuses on concentrating (consolidating) bitrate on a Region of Interest (ROI). More recently in [9] a more elaborate approach that adds higher level elements such as motion field correction filtering is proposed in the context of H.263. In [6] 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 [11] 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 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 reqires 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.

In Section 2 we discuss the effects of video compression on the efficiency of tracking algorithms, focusing on the distortion of features commonly used in real-time video object tracking. We motivate the need for resource consolidation in the context of traffic video compression. In Section 3 we propose our method of kurtosis-based derivation of an ROI to guide video compression, for which we show experimental results in Section 4. Finally we present concluding remarks in Section 5.

2. EFFECTS OF COMPRESSION DISTORTION ON TRACKING

Compression artifacts are debilitating for tracking applications. In reviews of object tracking presented in [1] and [2] it is shown that most algorithms focus on three features in video to track objects: *spatial edges*, *color histograms* and *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 small search area. For example, heavily quantized but motionless areas such as the road surface will flicker with time, appearing as

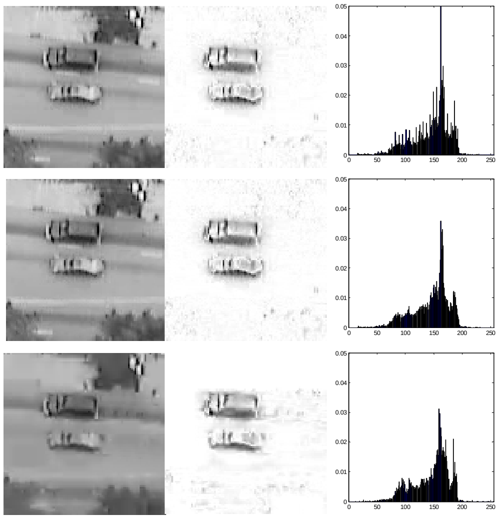


Fig. 1. Compression effects on vehicle tracking. The top row is a sample of uncompressed video, its error image vs. the background (median frame), and its intensity histogram respectively. The middle row video was compressed at a ratio of $3:10^2$, the bottom row at $3:10^4$.

having 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.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 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 new peaks introduced make histogram based tracking more difficult.

3. PROPOSED METHOD

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 due to passing cars).

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

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

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 [10], 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:

$$m_i = argmin\{w_i^d * Distortion + \lambda * Rate_i\} \tag{2}$$

where w_i^d is set equal to 0 for areas outside the ROI and equal to 1 for those within. Note that this step is necessary to code "zero motion" blocks outside the ROI – these blocks cannot simply be skipped given H.264 spatial motion vector prediction. 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.

4. EXPERIMENTAL RESULTS

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

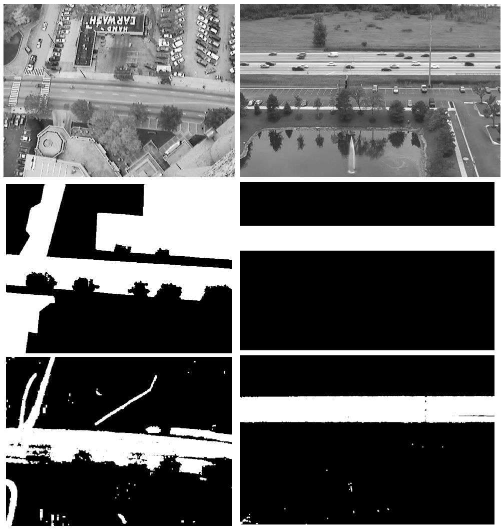


Fig. 2. Sample frames from "Camera6" and "I-90" sequences (top), their manually segmented ROI for analysis (center) and automatically extracted kurtosis-driven ROI for encoding (bottom).

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. 2 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 strength for the detector in that it focuses the ROI to region where activity has been reported and not a region where activity could take place.

In order to analyze total distortion to 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 [8] 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 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 [3] and implemented in the OpenCV project 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. 4 and 5 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 [4] 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.

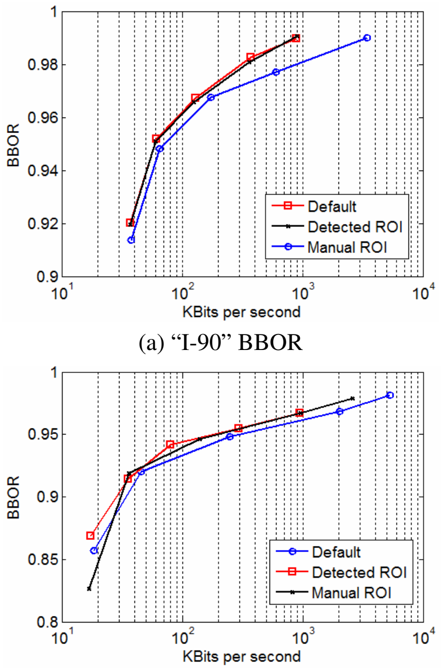
5. CONCLUSION

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.

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

6. REFERENCES

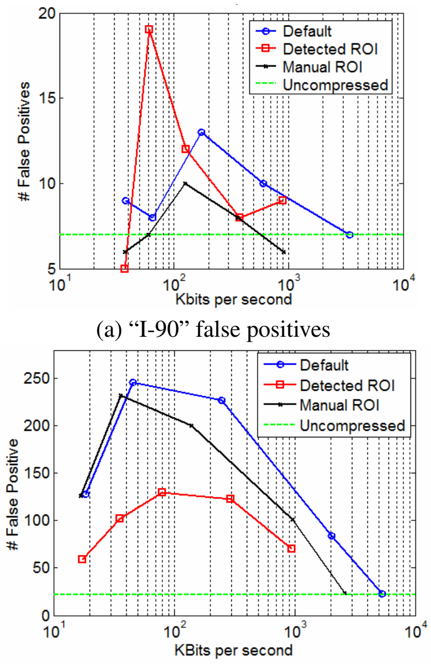
[1] A. Yilmaz, O, Javed, M. Shah, "Object Tracking: A Survey", *ACM Computing Surveys*, 2006, Vol. 38, No. 4, pp. 13.1-13.45



(a) "I-90" BBOR

(b) "Camera6" BBOR

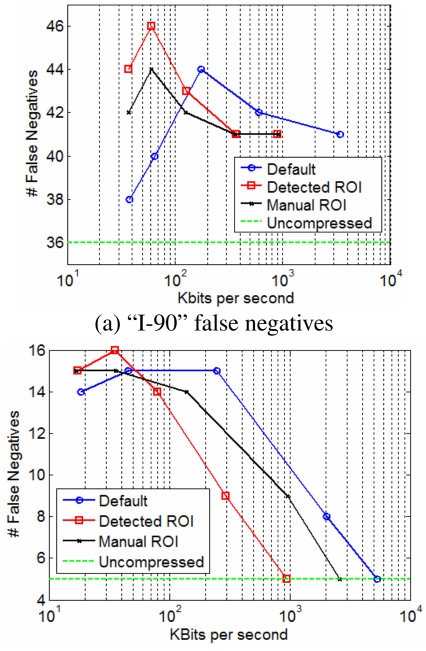
Fig. 3. Bitrate vs BBOR for "I-90" and "Camera6" sequences.



(a) "I-90" false positives

(b) "Camera6" false positives

Fig. 4. "I-90" and "Camera6" tracking false positive errors as a function of bitrate.



(a) "I-90" false negatives

(b) "Camera6" false negatives

Fig. 5. "I-90" and "Camera6" tracking false negative errors as a function of bitrate.

[2] P. F. Gabriel, J. G. Verly, J. H. Piater, A. Genon, "The State of the Art in Multiple Object Tracking Under Occlusion in Video Sequences", *Proc. ACIVS*, 2003, pp. 166-173

[3] D. Comaniciu, V. Ramesh, P. Meer, "Real-Time Tracking of Non-Rigid Objects Using Mean Shift", *Proc. CVPR*, 2000, Vol. 2, pp. 142-149

[4] C.E. Erdem, A. M. Tekalp, B. Sankur, "Video Object Tracking With Feedback Of Performance Measures", *IEEE Trans. on Circ. And Sys. for Video Tech.*, 2003, Vol. 13, pp. 310-324

[5] N. Zingirian, P. Baglietto, M. Maresca, M. Migliardi, "Customizing MPEG Video Compression Algorithms to Specific Application Domains: The Case of Highway Monitoring", *Proc. ICIAP*, 1997, Vol. II, pp. 46-53

[6] R. De Sutter, K. De Wolf, S. Lerouge, R. Van de Walle, "Lightweight Object Tracking in Compressed Video Streams Demonstrated in Region-Of-Interest Coding", *EURASIP Journal on Adv. in Sig. Proc*. Vol. 2007, Article 97845

[7] S. Cheung, C. Kamath, "Robust Techniques for Background Subtraction in Urban Traffic Video", *Proc. VCIP*, 2009, Vol. 5308, No. 1, pp. 881-892.

[8] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms", *IEEE Trans. on Systems, Man and Cybernetics*, 1975, Vol. 9, pp 62-66

[9] W. K. Ho, W. Cheuk, and D. P. Lun, "Content-Based Scalable H.263 Video Coding for Road Traffic Monitoring", *IEEE Trans. on Multimedia*, 2005, Vol. 7, No. 4

[10] A. Briassouli, V. Mezaris, I. Kompatsiaris, "Video Segmentation and Semantics Extraction from the Fusion of Motion and Color Information", *Proc. ICIP*, 2007, Vol. 3, pp. 365 - 368

[11] A. K. Kannur, B. Li "Power-Aware Content-Adaptive H.264 Video Encoding", *Proc. ICASSP*, 2009, Vol. 00, pp. 925-928