Application-Aware Approach to Compression and Transmission of H.264 Encoded Video for Automated and Centralized Transportation Surveillance

Zhaofu Chen, Sotirios A. Tsaftaris, *Member, IEEE*, Eren Soyak, *Member, IEEE*, and Aggelos K. Katsaggelos, *Fellow, IEEE*

***Abstract*—In this paper, we present a transportation video coding and wireless transmission system specifically tailored to automated vehicle tracking applications. By taking into account the video characteristics and the lossy nature of the wireless channels, we propose video preprocessing and error control approaches to enhance tracking performance while conserving bandwidth resources and computational power at the transmitter. Compared with current state-of-the-art H.264-based implementations, our system is shown to yield over 80% bitrate savings for comparable tracking accuracy.**

***Index Terms*—Error concealment (ERC), forward error correction (FEC), H.264/AVC, object tracking, preprocessing, surveillance centric coding, transportation video.**

I. Introduction

Remote imaging sensors are commonly deployed for transportation monitoring and surveillance [1], [2]. To reduce the associated infrastructure cost and increase urban coverage, inexpensive but computationally constrained remote sensors are deployed for video capturing and wireless transmission. Therefore, video compression technologies have to be applied such that the compressed bitrate can be accommodated by the modern wireless channels.

Recently, the state-of-the-art H.264 [3] standard has been proposed for transportation video-related applications [4]–[6]. However, H.264 is a generic video coding procedure, whose direct application in transportation systems may lead to degraded performance, e.g., tracking accuracy, because of the application-agnostic information removal. Another challenge faced by the transportation video transmission system is the lossy nature of the wireless channels. Despite the significant interest in resource-distortion optimization given channel losses [7], [8], these works, as others, are not application aware or tailored to transportation object tracking.

Recently, it has been shown that it is possible to maintain tracking accuracy with reduced bitrate by focusing resources in an application-aware context [9]. In this paper, motivated by the findings of [9] and [10], we propose a transportation video transmission system that integrates components at both the transmitter and receiver sides to

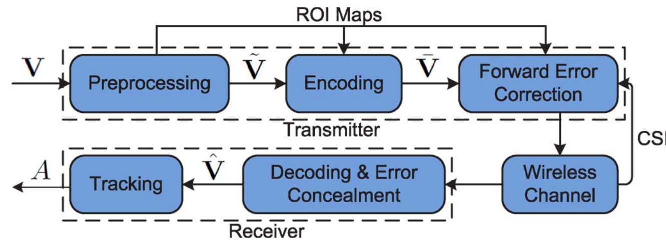

Fig. 1. System block diagram.

increase tracking accuracy while minimizing bitrate, given channel losses and remaining standard compliant. The contributions of this paper are centered on optimizing bit allocation within video frames, error protection schemes for video packets, and concealment strategies in the case of losses while maximizing tracking accuracy at the receiver's end and minimizing computational load at the encoder's side. Although other works discuss video quality-rate tradeoffs [7], [8], this is the first treatment of identifying beneficial error mitigation and concealment strategies from the viewpoint of the performance of automated video analysis.

The rest of this paper is organized as follows. In Section II, we provide an overview of the proposed system and present the employed performance metrics. Section III describes the system components and discusses their relative merits and limitations. In Section IV, we demonstrate the effectiveness of the proposed system using real-life test videos. For comparable tracking accuracy, our proposed systems yield over 80% reduction in bitrate. Finally, this paper is concluded in Section V.

II. System Overview and Performance Metrics

The complete system design is illustrated in Fig. 1. At the transmitter, the input raw video $\mathbf{V}$ is preprocessed to identify image regions of low tracking interest, and signal composition in those regions is altered such that the subsequent generic video compressor can encode it with fewer bits. This preprocessing step provides bitrate savings, as well as useful region-of-interest (ROI) information, to the subsequent channel protection. For channel protection, we specifically consider forward error correction (FEC) because of the stringent delay requirement. FEC utilizes the channel state information feedback to determine the appropriate protection schemes.

The received bitstream is decoded with error concealment (ERC) to yield the reconstructed $\hat{\mathbf{V}}$, which is used as input for subsequent applications such as tracking [11]. Note that the discrepancy between $\hat{\mathbf{V}}$ and $\mathbf{V}$ because of all the processing/losses may severely affect the tracking performance.

To quantitatively analyze the impact of the aforementioned operations on tracking performance, we compare the trajectories of the modified video [i.e., the algorithmic result (AR)] with those of the input raw video $\mathbf{V}$ [i.e., the ground truth (GT)]. It is known that both tracking and trajectory analyses are nonlinear operations. An analytical estimator that could accept as input a video stream and as output a numerical estimate of tracking accuracy without performing tracking would be ideal; however, to the best of our knowledge, there is no such effective analytical estimator so far.

In [12], a review of the state of the art for video surveillance performance metrics is presented. We choose the Overlap, Precision, and Sensitivity metrics presented in [9] due to their relevance. Overlap $(OLAP)$ is defined as the ratio between intersections and unions of frame regions covered by GT and AR, averaged over all detected

Manuscript received November 8, 2012; revised March 11, 2013 and May 2, 2013; accepted June 3, 2013. Date of publication June 20, 2013; date of current version November 26, 2013. This work was supported in part by the Commercialization of Innovative Transportation Technology, Northwestern University. The Associate Editor for this paper was Q. Kong.

Z. Chen and A. K. Katsaggelos are with the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL 60208 USA (e-mail: zhaofuchen2014@u.northwestern.edu; aggk@eecs.northwestern.edu).

S. A. Tsaftaris is with the IMT Institute for Advanced Studies, 55100 Lucca, Italy (e-mail: s.tsaftaris@imtlucca.it).

E. Soyak is with AirTies Wireless Networks, 34394 Istanbul, Turkey (e-mail: erensoyak@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2013.2266861

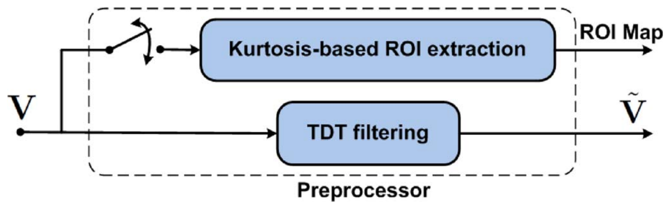

Fig. 2. Preprocessing block diagram.

objects. Precision $(PREC)$, defined as the number of true positives over the total number of objects detected in AR, measures how credible the AR is in terms of object identity. Similarly, sensitivity $(SENS)$ is defined as the number of true positives over the total number of objects detected in GT.

In order to jointly consider the above metrics, tracking accuracy A is defined as a linear combination of $OLAP$, $PREC$, and $SENS$, where the weights incorporate the relative importance of the accuracy components. In applications when no prior knowledge is assumed on the relative importance, we can set $A = (OLAP + PREC + SENS)/3$, as has been done in this paper.

III. Proposed System Components

A. Preprocessor

The preprocessor shown in Fig. 2 adopts a two-branch design. The lower branch filters the input video to consolidate encoded bits on video information that is most important for tracking, and the upper branch consists of an optional ROI extraction component. Both branches should be computationally efficient. Moreover, the lower branch should have fast adaptability to per-frame changes in order to minimize the bits spent on encoding noiselike intensity variations. The upper branch should provide homogeneous outputs valid for a long time, as is required by the subsequent encoding and channel protection. As is explained shortly, the complementary behaviors of these two branches jointly satisfy the set of design requirements.

1) TDT Filtering: The lower branch filtering is performed using the temporal deviation thresholding (TDT) algorithm [9]. TDT seeks to suppress noiselike per-pixel intensity variations before encoding. Such variations are usually imperfectly represented in the compressed video, which, after decoding, may contain artifacts that are misleading to the tracker. TDT suppresses such variations by iteratively generating frame mask $\mathbf{M}_t$ and output frame $\tilde{\mathbf{F}}_t$ as

$$\begin{aligned} \hat{\sigma}_t &= \text{mode}\left(\text{std}\left(\{\mathbf{F}_{t-T+1}, \ldots, \mathbf{F}_t\}\right)\right) \\ \boldsymbol{\Delta}_t &= |\mathbf{F}_t - \mathbf{F}_{t-1}|, \mathbf{M}_t = \boldsymbol{\Delta}_t > \tau\hat{\sigma}_t \\ \tilde{\mathbf{F}}_t &= \mathbf{M}_t \star \mathbf{F}_t + (1 - \mathbf{M}_t) \star \tilde{\mathbf{F}}_{t-1} \end{aligned} \tag{1}$$

where $\mathbf{F}_t$ is the tth input frame, $\star$ denotes the elementwise matrix multiplication, and $\tau > 0$ is a threshold multiplier. Since irrelevant intensity variations in the TDT output have been suppressed, it was shown that the filtered video yields a better rate-accuracy tradeoff.

Note that TDT utilizes simple first-order statistics and hence is able to adapt on a frame basis, which makes TDT effective as a video filter. However, as discussed in [9], the TDT output contains a few isolated misclassified pixels. Although such pixels do not incur significant bitrate increase because of their small spatial extent, they limit the use of TDT for ROI extraction, which requires a homogeneous map. Therefore, we rely on a nonparametric algorithm complementary to TDT to obtain a homogeneous ROI map.

2) Kurtosis-Based ROI Extraction: The primary motivation for ROI extraction is the subsequent unequal error protection (UEP), which is implemented using the flexible macroblock (MB) ordering (FMO) option in H.264. The FMO encoding requires a homogeneous ROI map valid for a large number of frames, which cannot be obtained from TDT.

The identification of the ROI is based on the excess kurtosis of the temporal pixel intensity distribution [13]. Briefly explained, using T training frames, the per-pixel excess kurtosis can be estimated. ROI extraction is achieved by thresholding the estimate at the midpoint between the two models representing: 1) noise and periodic movements of objects (such as trees and snow falls), which are characterized by a mixture of Gaussians (MoG) and 2) the desired type of motion due to moving vehicles, modeled as exponentials.

In practice, if the scene is relatively fixed (e.g., the camera is mounted on a pole), a single ROI can be used for a long time, and T is much less than the length of the entire sequence. Note also that the ROI is not updated on a frame basis and hence has a low amortized computational complexity.

3) Discussion on Preprocessing: As we have seen, TDT and the kurtosis-based approach complement each other and serve the purpose of video filtering and ROI extraction, respectively. TDT has fast adaptability and identifies (and suppresses) noiselike pixel variations. The kurtosis-based approach uses higher order statistics, and the generated homogeneous ROI maps unveil image areas where events of tracking interest are likely to occur in a long period of time.

B. Encoder

In this system, we use the state-of-the-art H.264 as the source encoder [3], with the FMO feature optionally enabled for UEP. Specifically, when UEP is desired, each video frame is divided into two slice groups (SGs), with one for the highlighted areas in the ROI and the other for the remaining areas. Since the SG information is conveyed in the picture parameter set as specified by H.264, it does not allow fast per-frame changes and any update to it costs bits. Thus, the "freshness" of the ROI information is limited by H.264 and the bitrate requirements.

C. FEC

The output of the H.264 encoder is a set of packets, each containing a slice of MBs. The FEC module adds redundancies to the packets for improved error resilience. Due to the limited computational power at the remote nodes, the simple yet effective redundant slices (RSs) are used to minimize the overall packet loss probability, where multiple copies of a slice are independently transmitted.

With the RS scheme, equal error protection (EEP) assigns a uniform protection level to all the packets, whereas UEP assigns possibly different levels to different packets. The apparent advantage of EEP is its inherent simplicity. Moreover, due to TDT filtering, most protection bits are devoted to the important video signal even if EEP is applied because intensity variations of low tracking interest have been suppressed by the preprocessor.

In other cases when the input frames exhibit complex motion of mixed types, UEP has the potential to make better utilization of the bandwidth resources. Specifically, let $\mathcal{I}_H$ denote the set of packets of high tracking interest and $\mathcal{I}_L$ the set of remaining packets. H and L denote the respective assigned protection levels, whose selection can be based on the maximum supported bitrate $\hat{R}$ and a target overall loss probability P_{target}. Let R_i denote the size of the ith packet. Algorithm 1 can be carried out to determine H and L.

Algorithm 1 Determining UEP Error Protection Levels

1: $c_{\text{target}} = \lceil \log_{P_{\text{unprot}}}(P_{\text{target}}) \rceil$; /* P_{unprot} is the unprotected packet loss probability */
2: **if** $c_{\text{target}} \sum_{i\in\mathcal{I}_H} R_i + \sum_{i\in\mathcal{I}_L} R_i > \hat{R}$ **then**
3: $\quad H = \lfloor(\hat{R} - \sum_{i\in\mathcal{I}_L} R_i)/\sum_{i\in\mathcal{I}_H} R_i\rfloor$;
4: $\quad L = 1$;
5: **else**
6: $\quad H = c_{\text{target}}$;
7: $\quad L = \lfloor(\hat{R} - c_{\text{target}}\sum_{i\in\mathcal{I}_H} R_i)/\sum_{i\in\mathcal{I}_L} R_i\rfloor$;
8: **end if**

The underlying assumption here is that $\hat{R}$ can support at least one copy of each packet. If this assumption is violated, we can prioritize important packets as discussed in [14].

D. Decoder and ERC for Tracking

By design of FEC, the information contained in a lost packet will not be retransmitted and must be estimated at the decoder, a process known as ERC. An important issue in ERC is the encoder/decoder mismatch. In transportation videos, due to the approximately translational motion of objects, such mismatch usually creates trailing artifacts that are particularly distracting to trackers. Thus, it is desirable to identify ERC strategies that minimize such types of distortion.

In general, ERC utilizes the spatial or temporal correlation between the lost information and its neighbors [15]. A typical example of ERC based on spatial correlation is the boundary matching algorithm (BMA) [16], which interpolates the video content using reliably reconstructed spatial neighbors to minimize the discrepancy of the boundary surrounding the lost region. The minimization is based on metrics such as peak signal-to-noise ratio and mean squared error, which are not application aware (i.e., tracking aware in our problem).

On the other hand, there exist ERC schemes that explicitly use temporal correlation. A straightforward but intuitive example is the motion-copy (MC) algorithm [17], which uses the motion information from colocated MBs in the previously decoded frames. The MC algorithm is potentially more capable of accurately recovering the lost translational motion information. It is particularly suitable for the preprocessed video because the pixel variations, not due to translational object motion, have been suppressed by TDT. In the numerical examples below, we will show several examples of trailing artifacts and demonstrate that the MC algorithm indeed outperforms the spatial BMA scheme in terms of reduced number of false positives and improved accuracy.

IV. Experiments

A. General Framework

To verify the gains made possible by the proposed schemes, we test the system using multiple known and privately collected sequences with different characteristics such as viewing angles, quality, and type of observed vehicle traffic. Results presented below refer to two publicly available sequences. Experiments on other sequences demonstrate similar behavior and are not shown here for brevity.

The "Camera6" sequence [18] shows an intersection with light traffic, with trees swaying and buildings casting reflections of passing cars. The "dt_passat" sequence [19] shows a busy intersection with traffic interrupted by a signal light and an urban rail crossing. Both sequences contain significant capture noise.

The proposed system is implemented with JM 16.2 [20] reference software, with the FMO feature optionally enabled for UEP-related experiments. The JM decoder is modified to enable the MC strategy, whereas the built-in BMA is used for performance comparison. The open-source OpenCV 2.3 [21] "blobtrack" module is used as the object tracker, which relies on the mean shift object tracking algorithm [22].

In the following experiments, we consider an independent identically distributed (IID) channel model. When channel conditions change fast or when packet interleaving is possible, an IID channel leading to independent packet losses is a reasonable approximation.

B. Effect of Preprocessing

To illustrate the effect of the proposed preprocessing filter in terms of bitrate consolidation and accuracy improvement, we take the raw video encoded using H.264 as the benchmark. In addition, we consider a popular background (BG) segmentation algorithm introduced in [23]. Despite its popularity for BG segmentation, this algorithm has not been used for application-aware video filtering similar to what is considered herein. We therefore adopt the principles from the original algorithm and adapt it as a reference video filtering approach.

Briefly explained, in this reference approach [23], the pixel intensity is modeled as a MoG distribution, where the classification of pixel is based on its matching to one of the Gaussian components. When all pixels in a frame are classified, a mask $\mathbf{M}_t$ is accordingly generated, which, in turn, is used to generate the filtered frame

$$\tilde{\mathbf{F}}_t = \mathbf{M}_t \star \mathbf{F}_t + (1 - \mathbf{M}_t) \star \mathbf{F}_{\text{static}} \tag{2}$$

where $\mathbf{F}_{\text{static}}$ is a static image taken from the video to represent the BG.

There are two notable differences between the TDT and MoG algorithms. First, TDT and MoG differ in their treatment for BG. For TDT, the BG regions are replaced by the collocated image signal from the previous filtered frame; thus pixels classified as BG are considered unchanged between two consecutive frames. For MoG, the classification is done by clustering, and therefore, the BG pixels may not resemble their temporal predecessors. This implies that TDT yields a more visually appealing BG that is updated and adaptable to changes. Our testing showed that letting MoG mimic TDT in BG treatment and use the previous filtered output to replace the BG will result in trailing artifacts. Second, TDT enjoys lower computational complexity. MoG maintains a multicomponent model for each pixel, whereas TDT maintains a single-component model for all pixels within a frame. Hence, TDT requires less computation to update its statistical model and uses less space to store the model than MoG. This difference is important in light of the constrained computational power and memory capacity at the remote nodes. We will see shortly that TDT, despite its relative simplicity, has satisfactory performance compared with the reference MoG algorithm.

For the remainder of our experiments for TDT filtering, we set the threshold τ to be 2 and buffer size T to be 7. For MoG filtering, five Gaussian components are used and the first frame in a video sequence is used as the static image for MoG.

In Fig. 3, we present sample outputs from the preprocessing steps from typical frames that exhibit moving objects of tracking interest (e.g., vehicles and pedestrians), as well as noise and irrelevant motion. As can be seen, both TDT and MoG unveil the moving objects of tracking interest. TDT misclassifies some isolated pixels in the BG as movement, leading to false positives. Although these may not hurt bitrate performance noticeably, they are evidence as to the difficulty of fusing multiple TDT masks to create a homogeneous ROI map. To the contrary, the ROI maps extracted by the kurtosis-based algorithm clearly highlight the streets and are thus well suited for UEP. However, if these ROI maps were to be used for filtering, some bits would be

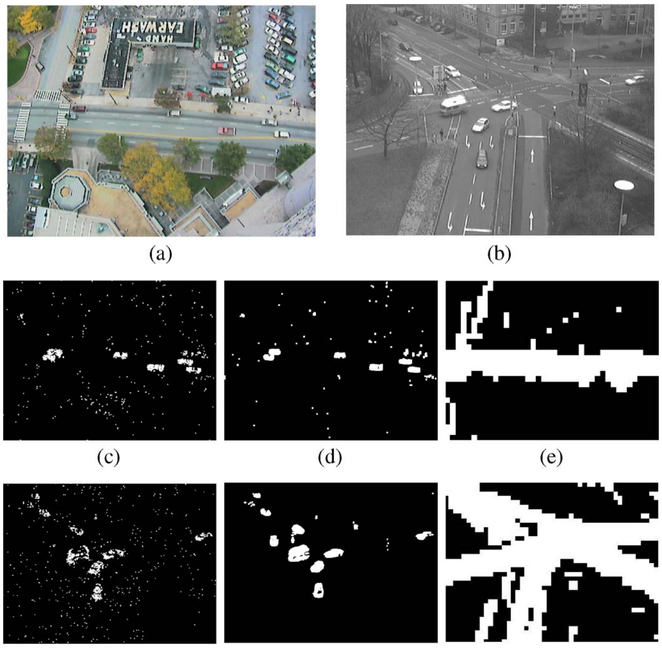

Fig. 3. Effects of preprocessing. [(a) and (b) Original frames in “Camera6” and “dt_passat”; (c)–(e) TDT mask, MoG mask, and kurtosis-based ROI for “Camera6” frame; (f)–(h) TDT mask, MoG mask, and kurtosis-based ROI for “dt_passat” frame.]

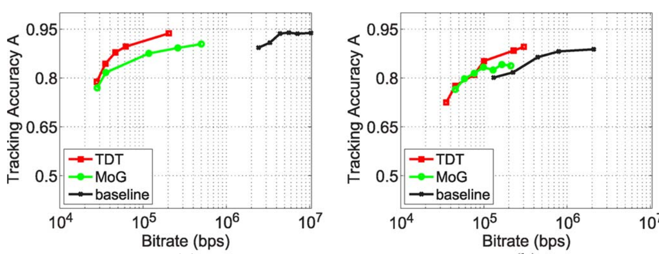

Fig. 4. Comparison between filtering approaches. (a) “Camera6” sequence. (b) “dt_passat” sequence.

spent on encoding static information. Thus, our proposed design takes advantage of the complementary behaviors of the two algorithms.

Quantitatively, we examine the effectiveness of the TDT and MoG algorithms in bitrate consolidation and compare their performance with unpreprocessed video encoded using the baseline H.264. It is evident from Fig. 4 that TDT effectively consolidates bitrate on the video content important for tracking. Compared with baseline, TDT significantly reduces the encoded bitrates while maintaining satisfactory tracking accuracy. Compared with its MoG counterpart, TDT demonstrates comparable or even improved performance despite its computational simplicity.

C. Effects of ERC

To investigate ERC performance, the encoded video sequences were subject to random packet losses, and the received bitstreams were decoded with BMA and MC, respectively. The results averaged across multiple random realizations are presented below.

Fig. 5 shows zoom-in views of the concealed image regions. Visually, the MC-concealed frames exhibit less severe trailing artifacts than the BMA-concealed frames due to the better utilization of the motion information embedded in the bitstream.

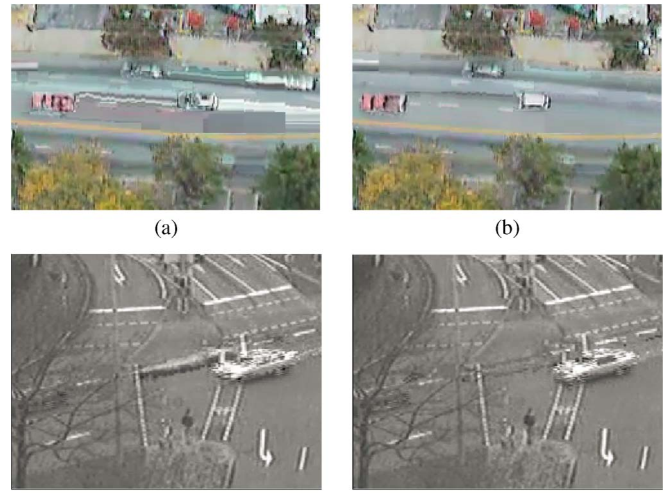

Fig. 5. Sample concealed frame for (a) and (b) “Camera6” sequence and (c) and (d) “dt_passat” sequence. (a) BMA-concealed frame. (b) MC-concealed frame. (c) BMA-concealed frame. (d) MC-concealed frame.

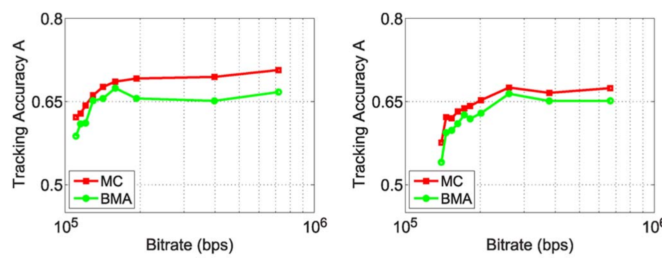

Fig. 6. Tracking performance comparison between ERCs. (a) “Camera6” sequence. (b) “dt_passat” sequence.

TABLE I
AVERAGE TRACKING PERFORMANCE WITH MC AND BMA (IN PERCENT); SEQUENCE 1 IS “Camera6”; SEQUENCE 2 IS “dt_passat”

Seq.	Overlap		Precision		Sensitivity		Accuracy	
	MC	BMA	MC	BMA	MC	BMA	MC	BMA
1	66.4	64.3	50.2	44.4	84.0	83.4	66.8	64.0
2	52.7	50.8	74.4	71.1	64.8	63.6	64.0	61.8

Quantitatively, we show the tracking accuracy values for the concealed video sequences at various bitrates, where the average packet loss probability is maintained at 0.05. As shown in Fig. 6, MC-concealed sequence in general yields higher tracking accuracy values. When examining the individual components of tracking accuracy shown in Table I, we see that when BMA is used, a 12% decrease in precision is observed compared with MC. By the definition of precision, this implies that BMA results in an increased number of false positives, which is directly attributed to the more severe trailing artifacts.

Note that channel impairments (even under moderate channel conditions) do affect tracking accuracy negatively, explaining the overall low tracking accuracy values shown above. Thus, bitrate consolidation and channel protection must be introduced to improve overall system performance.

D. Overall System Performance

In the previous paragraphs, we showed that some preprocessing (TDT or MoG) can consolidate bitrate and that different ERC strategies have different effects on tracking accuracy. In this section,

TABLE II
SYSTEM COMPONENT SETTINGS

System ID	Input Filtering	ROI Enabled	FMO Enabled	FEC Scheme	ERC Strategy
Proposed-1	TDT	No	No	EEP(3)	MC
Proposed-2	TDT	Yes	Yes	UEP(3,2)	MC
Reference	MoG	No	No	EEP(3)	MC
Baseline-1	None	No	No	None	MC
Baseline-2	None	No	No	EEP(2)	MC
Baseline-3	None	No	No	EEP(3)	MC

we combine the proposed TDT preprocessing, FEC (EEP and UEP), and ERC modules into a complete system and compare its performance with a MoG-based reference system, as well as variations of the baseline H.264 implementations. From all possible system settings, we present those in Table II to illustrate the combined gains of the proposed approach.

Different system settings are referred to by their respective IDs. The numbers in column 5 denote the protection levels. Note that both Proposed-1 and Proposed-2 use TDT. However, Proposed-1 employs EEP and hence does not require explicit ROI extraction or FMO-enabled encoding. The reference system employs MoG for input filtering, which replaces the BG regions with a static image and hence spends minimum bits on encoding such regions. In this case, EEP is used for its channel protection as UEP leads to minimal gains. For the baseline implementations, we use the H.264 encoder with the unpreprocessed video. Since channel errors have a major impact on the decoded video quality, unprotected bitstreams (Baseline-1) yield exceptionally low tracking accuracy. In order to make the comparison of results more interesting, we apply EEP with two levels to the baseline encoded bitstreams and record their respective tracking accuracy after transmission and decoding.

The performance comparisons shown in Fig. 7 make it evident that the baseline implementations, despite their great efficiency in generic video compression, are less effective in this application-specific environment. The proposed TDT filtering removes video content unimportant for tracking and saves bitrate for protection purposes. The FEC module utilizes the bitrate savings to provide protection to the video information most relevant for tracking. At the decoder side, tracking-aware ERC strategy is employed to recover the lost motion information and therefore further improves the overall system performance. Quantitatively, the proposed system yields an 80% reduction in bitrate for the same tracking accuracy compared with the protected baseline implementation.

The proposed system has similar performance as the reference system while being computationally more efficient. This is a significant advantage in our constrained environment. Comparing between Proposed-1 and Proposed-2, we see that when the scene contains complex motion of mixed types, gains are possible by using UEP because the ROI map it relies on is derived using higher order statistics.

From this numerical analysis, we see that the proposed system is effective in: 1) bitrate consolidation; 2) channel degradation mitigation; and 3) lost information recovery, while always maintaining a low computational cost profile at the encoder side and retaining standard compliance.

V. CONCLUSION AND FUTURE WORK

In this paper, we have presented a video coding and transmission system specifically tailored to automated transportation surveillance and monitoring. The characteristics of the video and the lossy nature of the wireless channels were considered in the system design. The effectiveness of the proposed system was demonstrated using real-life video sequences. The current system includes individual designs for the transmitter and receiver. By incorporating information about the receiver in the transmitter design, it is possible to achieve more specialized preprocessing and channel protection. As an example, if the preprocessor has information as to what the ERC module can recover, it can intentionally discard such information whenever there is a rate-accuracy benefit.

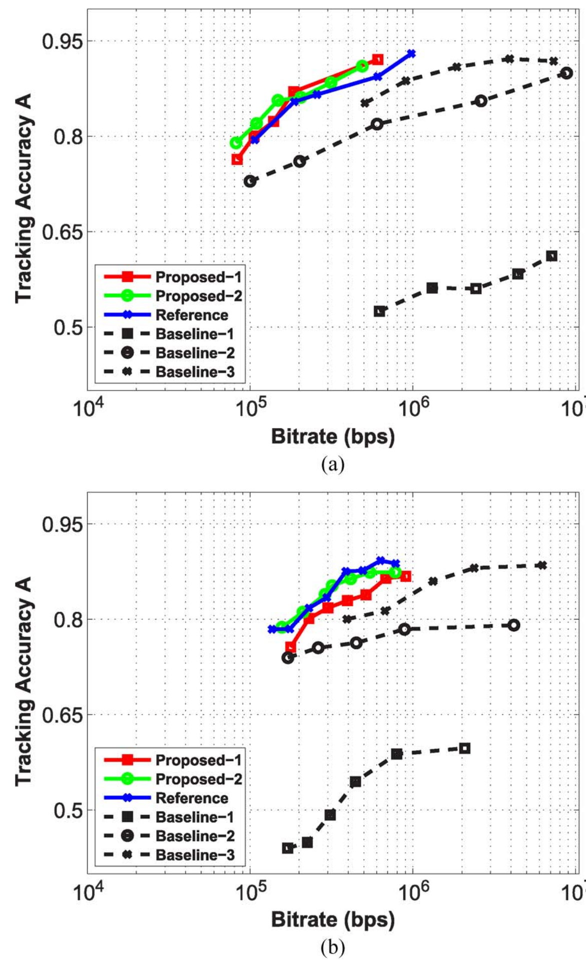

Fig. 7. Comparison of system performance. (a) "Camera6" sequence. (b) "dt_passat" sequence.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their constructive comments and help for improving the manuscript.

REFERENCES

[1] M. Kyte, A. Khan, and K. Kagolanu, "Using machine vision (video imaging) technology to collect transportation data," *Transp. Res. Rec., Innovations Travel Survey Methods*, no. 1412, pp. 23–32, 1993.

[2] F. Jiang, Y. Wu, and A. Katsaggelos, "A dynamic hierarchical clustering method for trajectory-based unusual video event detection," *IEEE Trans. Image Process.*, vol. 18, no. 4, pp. 907–913, Apr. 2009.

[3] H.264: Advanced Video Coding for Generic Audiovisual Services, Jun. 2011. [Online]. Available: http://www.itu.int/rec/T-REC-H.264/

[4] C. Erdem, "Video object tracking with feedback of performance measures," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 4, pp. 310–324, Apr. 2003.

[5] R. Sutter, K. DeWolf, S. Lerouge, and R. de Walle, "Lightweight object tracking in compressed video streams demonstrated in region-of-interest

coding," *EURASIP J. Appl. Signal Process.*, vol. 2007, no. 1, p. 59, Jan. 2007.

[6] A. Kannur, "Power-aware content-adaptive H.264 video encoding," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2009, pp. 925–928.

[7] A. Katsaggelos, Y. Eisenberg, F. Zhai, R. Berry, and T. Pappas, "Advances in efficient resource allocation for packet-based real-time video transmission," *Proc. IEEE*, vol. 93, pp. 135–147, Jan. 2005.

[8] P. Baccichet, S. Rane, A. Chimienti, and B. Girod, "Robust low-delay video transmission using H.264/AVC redundant slices and flexible macroblock ordering," in *Proc. IEEE Int. Conf. Image Process.*, Jul. 2007, vol. 4, pp. IV-93–IV-96.

[9] E. Soyak, S. Tsaftaris, and A. Katsaggelos, "Low-complexity tracking-aware H.264 video compression for transportation surveillance," *IEEE Trans. Circuits Syst. Video Technol.—Special Issue Video Analysis Resource-Limited Systems*, vol. 21, no. 10, pp. 1378–1389, Oct. 2011.

[10] E. Soyak, S. Tsaftaris, and A. Katsaggelos, "Channel protection for H.264 compression in transportation video surveillance applications," in *Proc. IEEE ICIP*, 2011, pp. 2237–2240.

[11] Y. Wang and Q. Zhu, "Error control and concealment for video communication: A review," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 86, no. 5, pp. 974–997, May 1998.

[12] A. Baumann, M. Boltz, J. Ebling, M. Koenig, H. Loos, M. Merkel, W. Niem, J. Warzelhan, and J. Yu, "A review and comparison of measures for automatic video surveillance systems," *EURASIP J. Image Video Process.*, vol. 2008, no. 3, p. 1, Jun. 2008.

[13] E. Soyak, S. Tsaftaris, and A. Katsaggelos, "Content-aware H.264 encoding for traffic video tracking applications," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Mar. 2010, pp. 730–733.

[14] P. Pahalawatta, R. Berry, T. Pappas, and A. Katsaggelos, "Content-aware resource allocation and packet scheduling for video transmission over wireless networks," *IEEE J. Sel. Areas Commun.—Special Issue Cross-Layer Optimization Wireless Multimedia Communications*, vol. 25, no. 4, pp. 749–759, May 2007.

[15] Y. Wang, S. Wenger, J. Wen, and A. Katsaggelos, "Error resilient video coding techniques," *IEEE Sig. Process. Mag.*, vol. 17, no. 4, pp. 61–82, Jul. 2000.

[16] W. Lam, A. Reibman, and B. Liu, "Recovery of lost or erroneously received motion vectors," in *Proc. IEEE ICASSP*, Apr. 1993, vol. 5, pp. 417–420.

[17] S. Bandyopadhyay, Z. Wu, P. Pandit, and J. Boyce, "An error concealment scheme for entire frame losses for H.264/AVC," in *Proc. IEEE Sarnoff Symp.*, Mar. 2006, pp. 1–4.

[18] Next Generation Simulation (NGSIM), Jan. 2013. [Online]. Available: http://www.ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm

[19] KOGS/IAKS Karlsruhe Image Sequence Server, Feb. 2012. [Online]. Available: http://i21www.ira.uka.de/image_sequences/

[20] The Open-Source H.264/AVC Verification Model, Feb. 2012. [Online]. Available: http://iphome.hhi.de/suehring/tml/

[21] The OpenCV Real-Time Computer Vision Library, Dec. 2010. [Online]. Available: http://opencv.willowgarage.com

[22] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2000, vol. 2, pp. 142–149.

[23] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 1999, pp. 2246–2252.