CHANNEL PROTECTION FOR H.264 COMPRESSION IN TRANSPORTATION VIDEO SURVEILLANCE APPLICATIONS

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

[a] Dept. of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL, USA
[b] Dept. of Radiology, Feinberg School of Medicine, Northwestern University, Chicago, IL, USA
email: {e-soyak, s-tsaftaris}@northwestern.edu, aggk@eecs.northwestern.edu

ABSTRACT

The compression of video and subsequent partial loss of the compressed bitstream can dramatically reduce the accuracy of automated tracking algorithms. This is problematic for centralized applications such as transportation surveillance systems, where remotely captured and compressed video is transmitted over lossy wireless links to a central location for tracking. We propose a low-complexity method for protecting compressed video against channel loss such that the tracking accuracy of decoded and concealed video is maximized. Our algorithm leverages a previous method of video processing that removes components of low tracking interest before compression to minimize bitrate, and uses some of the bitrate savings to introduce redundancy into the transmitted bitstream to reduce the probability of information loss. We show using a common tracker and loss concealment algorithm that our system allows for up to 100% increased tracking accuracy at a given bitrate, or 90% bitrate savings for comparable tracking quality.

Index Terms— Transportation video tracking, video compression, wireless channel, packet loss

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 assumes either expensive wired communication links or that the video data is being heavily compressed to not exceed the allowed communications bandwidth. Current transportation video solutions utilize older video compression standards and require dedicated wired communication lines. Recently H.264/AVC has started to be used in transportation applications, significantly reducing the link bandwidth requirement. However, most systems currently in the field are not optimized for traffic video data, nor do they take into account channel losses or the automated video analysis that will follow at the control center. As a result the visual quality of the data will suffer, but more importantly the tracking accuracy and efficiency are severely affected.

While the field of video object tracking contains a large variety of algorithms, most of these systems share some fundamental concepts. In a recent review of object tracking algorithms presented in [1] 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.

Where packet losses are expected, the encoder is expected to make coding decisions in a manner anticipating such loss, and taking into account the concealment strategy of the decoder. For example, the H.264 reference model (JM) in [2] offers the "frame copy" or "motion copy" concealment modes discussed in [3], which respectively copy colocated pixel values or scaled motion vectors and references for lost macroblocks. Fig. 1 shows sample loss-concealed video frames from the JM. The "trailing" (or "smearing") type artifacts shown in the figure are especially misleading for trackers, and are commonly seen when encoder/decoder mismatch due to packet losses is unsuccessfully concealed by the decoder.

There is significant interest in resource-distortion optimization given channel losses. In [4] such a framework along with an overview of packet-based video transmission is presented. In [5] an algorithm using H.264/AVC Flexible Macroblock Ordering and Redundant Slices to maximize the reconstructed Peak Signal to Noise Ratio (PSNR) in low-latency applications is presented.

However, these works focus on maximizing PSNR and do not consider the accuracy of object tracking. We propose to use redundant packet transmission in conjunction with a previous algorithm to minimize the bitrate for a given expected tracking accuracy in the presence of packet loss. We assume that camera nodes have limited computational capabilities as opposed to the centralized location which performs decoding and automated tracking.

Many traditional channel loss protection algorithms involve complex computation of the possible effects of packet loss in order to best allocate resources such as compression bits (encoder decisions), redundancy bits, and transmission power. Such algorithms are unsuitable for the remote (camera) nodes of transportation surveillance systems given the limited processing power available. The system presented herein is designed to be low in complexity and to be readily deployable as a simple modular add-on to remote nodes. 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 inter-operability.

The rest of this paper is organized as follows. In Section 2 we discuss our methodology for measuring the effects of video compression and packet loss on the efficiency of tracking algorithms. Subsequently we propose our method of minimizing the impact of packet losses on tracking accuracy via redundant slices, for which we show experimental results in Section 3. Finally we present concluding remarks in Section 4.

2. PROPOSED METHOD

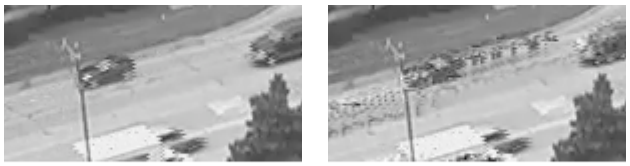
2.1. Modeling Channel Distortion of Tracking

We refer to the closeness of the match between targets tracked in the uncompressed and compressed videos as tracking accuracy, which we measure using the *Overlap* ($OLAP$), *Precision* ($PREC$) and *Sensitivity* ($SENS$) metrics presented in [6] and [7]:

$$OLAP = (GT_i \cap AR_i)/(GT_i \cup AR_i) \quad (1)$$

$$PREC = \overline{TP}/(\overline{TP} + \overline{FP}) \quad (2)$$

$$SENS = \overline{TP}/(\overline{TP} + \overline{FN}), \quad (3)$$



(a) without packet loss (b) with packet loss

Fig. 1. Sample concealment artifacts.

where in terms of the Ground Truth (GT) and Algorithm Result (AR), True Positives (TPs) are objects present in both the GT and AR, False Positives (FPs) are objects present in the AR but not in the GT, and False Negatives (FNs) are objects present in the GT but not in the AR. GT_i denote the segmented objects tracked in uncompressed video, AR_i those tracked in compressed video, $\cap$ the intersection of the two regions, and $\cup$ their union.

In order to jointly optimize for a combination of these metrics we define *tracking accuracy* A as

$$A = (\alpha * OLAP) + (\beta * PREC) + (\gamma * SENS), \quad (4)$$

where α, β and γ are weighting coefficients, such that $\alpha+\beta+\gamma = 1$.

We formulate the video compression, transmission and tracking processes as follows. The function f_{enc} mapping the raw video $\mathbf{V}$ to the compressed video $\hat{\mathbf{V}}$, with associated bitrate R and tracking accuracy A, is defined as

$$\{\tilde{\mathbf{V}}, R, A\} \overset{f_{enc}}{\leftarrow} \{\mathbf{V}\}. \quad (5)$$

The compressed video $\hat{\mathbf{V}}$ is then transmitted over a wireless channel with packet loss pattern `loss`, where the packets are protected against channel loss by the scheme `protect`, and the decoder conceals lost packets with scheme `conceal`. We define the function f_{xmit} mapping this input to the reconstructed/concealed video $\hat{\mathbf{V}}$ at the receiver, with associated bitrate R' and tracking accuracy A', as

$$\{\hat{\mathbf{V}}, R', A'\} \overset{f_{xmit}}{\leftarrow} \{\tilde{\mathbf{V}}, \texttt{loss}, \texttt{protect}, \texttt{conceal}\}. \quad (6)$$

Note that R' includes any additional bandwidth required by the channel protection scheme `protect`.

2.2. Core Algorithm

We propose a channel protection methodology using Redundant Slices (RS) to minimize the ultimate loss probability of each packet. Thus, in terms of Eqs. 5-6, our proposed algorithm only modifies the transmitter (i.e., `protect` in Eq. 6). We formulate our problem with the objective of maximizing the tracking accuracy A' on concealed/reconstructed video $\hat{\mathbf{V}}$ using bitrate R'. The channel model `loss` and decoder concealment `conceal` are not modified. Our system operates in two parts:

(a) suppress components in the compressed video that are of low tracking interest, thereby reducing the bandwidth requirement of the compressed bitstream

(b) use part of the bitrate savings from (a) to introduce redundancy into the transmitted packets in order to reduce their likelihood of being lost.

Note that part (a) removes components of low tracking interest from the bitstream, and thus all remaining data can be considered equally worth protecting against channel loss. This design is attractive for low-power remote surveillance nodes because it eliminates the need for processor-intensive Unequal Error Protection (UEP) algorithms.

For part (a) of our system, we make use of the filtering algorithm in [6]. This temporal filtering algorithm, referred to here as *Temporal Deviation Thresholding* (TDT), operates in two parts which we briefly describe here. The first part models, detects, and removes temporal events of low tracking interest (considered to be "noise") prior to compression, operating via an iterative filter described as

$$\mathbf{M}_t = |\mathbf{V}_t - \mathbf{V}_{t-1}| > C * n_t \quad (7)$$

$$\hat{\mathbf{V}}_t = \mathbf{M}_t * \mathbf{V}_t + \overline{\mathbf{M}_t} * \hat{\mathbf{V}}_{t-1} \quad \forall\ t \geq B, \quad (8)$$

where for frame t of video sequence $\mathbf{V}$, $\mathbf{M}_t$ is a logical bitmap, $\overline{\mathbf{M}_t}$ is its logical inverse, C is a constant, n_t an estimated noise standard deviation, B the number of buffered frames being analyzed, and $\hat{\mathbf{V}}_t$ the filtered output given to the encoder. The noise standard deviation is estimated by observing frame statistics over the last B frames. After the video is decoded at the receiver, n_t (which was transmitted with the bitstream) is used to synthesize and re-insert noise prior to tracking. Part (a) aims to minimize the bitrate requirement, while part (b) aims to improve post-compression tracking results. For further details see [6].

As shown in [6], TDT allows for up to 90% reduction in bitrate required for a given level of tracking accuracy. The majority of these savings originate from the fact that the encoder can use skip modes in areas TDT suppresses (i.e., where $\mathbf{M}_t$ = 0). Our goal is to use some of these bitrate savings to increase robustness to channel losses and maintain a high level of tracking accuracy. Therefore, for part (b) of our system, we make use of H.264 Redundant Slices to send each slice multiple times in order to reduce its probability of loss.

In order to model packet losses (i.e., `loss` in Eq. 6), due to wireless channel fading, we use a memoryless, uniformly distributed fading model. The video is packetized in fixed size (in bits) H.264/AVC slices. Each packet$_i$ is lost with equal probability P_l, and individual losses are determined independently (resulting in an independent and identically distributed loss pattern) based on the value of uniformly distributed fading random variable F_i as

$$\text{packet}_i \rightarrow \begin{cases} if\ F_i > 1 - P_l, & lost \\ else, & received \end{cases}. \quad (9)$$

We define RX_i, the number of redundant copies of slice i transmitted. While RX_i can be set adaptively based on slice content, given that TDT suppresses pixel activity of low tracking interest we set $RX_i = RX^* \ \ \forall\ i$. RX^* is set based on the expected packet loss probability P_l, which is usually available in a given system via channel state feedback. Based on our channel model P_l', the aggregate probability of loss including retransmissions, can be stated as

$$P_l' = (P_l)^{RX^*} \quad (10)$$

$$R' = R \cdot RX^*. \quad (11)$$

Note that additively increasing RX^*, and therefore the final bitrate R', reduces P_l' exponentially. Given reasonable latency constraints, redundant packets can be sent with some delay after the originals, thereby minimizing the effect of sustained channel fades on the correlation between the loss probabilities of redundant packets. Thus our memoryless channel model becomes more realistic as system latency tolerances allow greater time offsets between redundant data.

It is critical to note that our system offers a form of "built-in" UEP due to the suppression of pixel activity of low tracking interest during TDT processing, conserving valuable resources by (a) encoding fewer active pixels (saving bits and processing power) and (b) using a fixed RX^* (no UEP calculations). Such computational efficiency makes our algorithm highly desirable for low computational power remote nodes. Note that as shown in [6], TDT itself is a low-complexity algorithm, recycling encoder frame buffers to minimize its memory footprint and using simple filtering operations to limit its processing power requirement. When considering low complexity UEP algorithms it should be noted that at reasonable bitrates using RTP packet sizes (1400 bytes) each frame is coded in just a few slices, limiting the flexibility available for UEP. Reducing the slice size allows more flexibility, but also limits compression efficiency given that no prediction is allowed across slice boundaries.

3. EXPERIMENTAL RESULTS

To verify the gains possible using our algorithm a sample implementation was tested using multiple sequences with differing characteristics such as viewing angle, quality and type of vehicle traffic observed. Details for the sample implementation and experimental procedure in addition to test results are presented below.

The video compression for the experiments presented herein was performed using the open-source x264 H.264/AVC encoder [8] and the JM 16.0 H.264/AVC reference decoder [2] with the built-in concealment [3] for packet losses. The open-source OpenCV [9] "blobtrack" module was used as the object tracker, which relies on the Mean Shift object tracking algorithm [10].

The following parameters were used for our experiments. For TDT, we generated 4 realizations of Gaussian noise per sequence, which were used for all experiments performed using that sequence. As experimental constants we used threshold $C = 2$ and buffer size $B = 7$. Fixed QP rate control was used (i.e., no frame or macroblock level rate control). 1400-byte slices were used based on the common size of an RTP packet. For packet losses, 16 realizations of channel fading were used, and the average $\{R', A'\}$ were reported.

The following videos were used for testing. The "Golf" sequence (720x480) was shot on DV tape and is a relatively high fidelity source, showing a local road intersection with steady non-rush traffic. As part of the scene there are trees and parking lots for office buildings and a strip mall. The "Camera6" sequence (640x480) was used under the NGSIM license courtesy of the US FHWA. It shows an intersection with light traffic, with trees swaying in the wind and buildings casting reflections of passing cars as part of the scene. This video was MPEG4 intra-only compressed during acquisition and is thus significantly noisier than the "Golf" sequence.

Experimental results from our test framework are presented in Fig. 2. The experiments compare the performance of default coding (data marked by points) and TDT coding (data marked by crosses) over less reliable channels (packet loss probability $P_l = 0.1$, shown in the left column) and more reliable channels ($P_l = 0.01$, shown in the right column). Our proposed algorithm using TDT protected against losses with redundancy is evaluated with two parameters: $RX^* = 2$, marked by plus signs, and $RX^* = 3$, marked by stars.

Note that over relatively unreliable channels ($P_l = 0.1$, i.e., on average 10% packet loss) if the default H.264 compressed video is transmitted without channel protection, even when loss concealment is used at the decoder, the tracking accuracy is dramatically degraded. Note that the dashed lines showing the unprotected default and TDT cases for "Camera6" and "dt_passat" are flat, i.e., increasing the bitrate in these scenarios does not significantly impact tracking accuracy. This is because tracking is more challenging in "Camera6" due to noise and in "dt_passat" due to complex traffic flow, limiting tracking accuracy in the presence of channel loss for these scenes regardless of bitrate. However, in "Golf" targets to be tracked appear relatively large and clear, with easy to track trajectories, making this scene less susceptible resilient to loss. However in all cases, using $RX^* = 3$ protection with TDT video, a 100% increase in tracking accuracy is possible compared to default coding at the same bitrate and channel conditions, attaining a level of tracking accuracy not possible by default coding regardless of bitrate.

Observe in Fig. 2 that over relatively reliable channels ($P_l = 0.01$, i.e., on average 1% packet loss) unprotected transmission does not impair tracking accuracy as much as at higher loss rates. Even in such cases, $RX^* = 2$ allows for up to 90% reduction in bitrate compared to default coding for the same tracking accuracy in most cases. In the field the transmitter will use channel feedback to estimate P_l and decide on how to set RX^*.

A simple channel model was used to keep our experiments as general as possible in terms of the nature of channel loss – for a more realistic simulation of real-world wireless channel fading and subsequent data loss, a model such as the one in [11] can be used.

4. CONCLUSION

We have proposed a low-complexity method of protecting compressed video against channel loss such that the tracking accuracy of decoded and concealed video is maximized. Our algorithm leveraged a previous method of video processing that removes components of low tracking interest from compressed video to minimize bitrate, using some of the bitrate savings to introduce redundancy into the transmitted bitstream to reduce the probability of information loss. We have demonstrated using a common tracker and a concealment algorithm that our system allows for up to 100% increased tracking accuracy at a given bitrate, or 90% bitrate savings for comparable tracking quality over a variety of channel conditions. In the future we will explore how state-of-the-art loss concealment methods affect tracking accuracy, and suggest tracking-optimal concealment algorithms.

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

5. REFERENCES

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

[2] (2010, Dec.) The open-source H.264/AVC verification model. [Online]. Available: http://iphome.hhi.de/suehring/tml/

[3] S. K. Bandyopadhyay, Z. Wu, P. Pandit, and J. M. Boyce, "An error concealment scheme for entire frame losses for H.264/AVC," in *Sarnoff Symposium 2006*, Princeton, NJ, USA, Mar. 2006, pp. 1–4.

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

[5] 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. ICIP*, vol. 4, San Antonio, TX, USA, Jul. 2007, pp. 93–96.

[6] E. Soyak, S. A. Tsaftaris, and A. K. Katsaggelos, "Tracking-optimal pre- and post-processing for H.264 compression in traffic video surveillance applications," in *Proc. ICECS*, Athens, Greece, Dec. 2010, pp. 380–383.

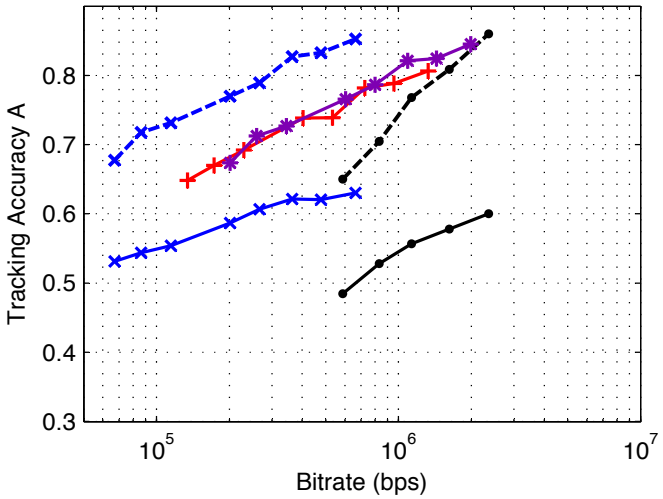
[7] M. B. A. Baumann, J. Ebling, M. Koenig, H. S. Loos, W. N. M. Merkel, J. K. Warzelhan, and J. Yu, "A review and comparison of measures for automatic video surveillance systems," *EURASIP Jour. on Image and Video Proc.*, vol. 2008, p. 30, 2008.

[8] (2010, Dec.) The open-source x264 video codec. [Online]. Available: http://www.videolan.org/developers/x264.html

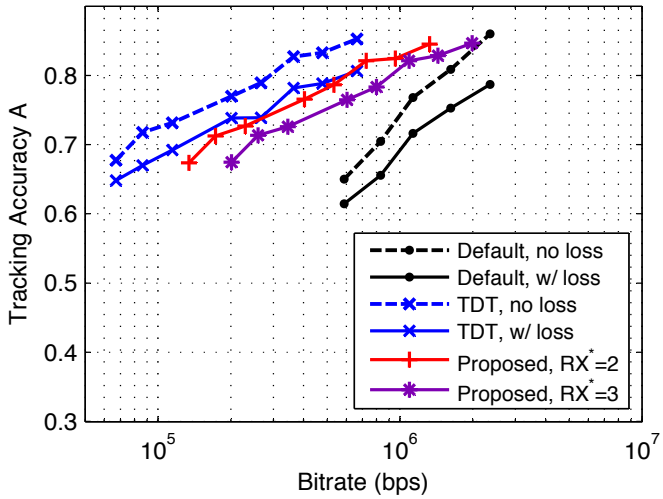
[9] (2010, Dec.) The OpenCV real-time computer vision library. [Online]. Available: http://opencv.willowgarage.com

[10] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Proc. CVPR*, vol. 2, Hilton Head, SC, USA, 2000, pp. 142–149.

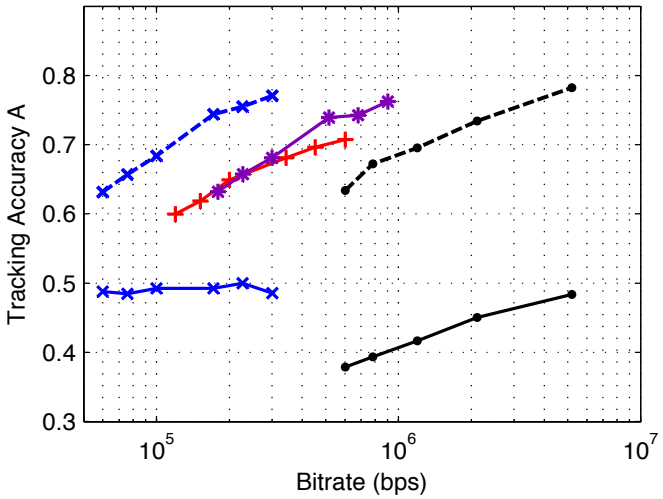
[11] E. Soyak, Y. Eisenberg, F. Zhai, T. N. Pappas, R. Berry, and A. K. Katsaggelos, "Channel modeling and its effect on the end-to-end distortion in wireless video communications," in *Proc. ICIP*, Singapore, 2004, pp. 3253–3256.



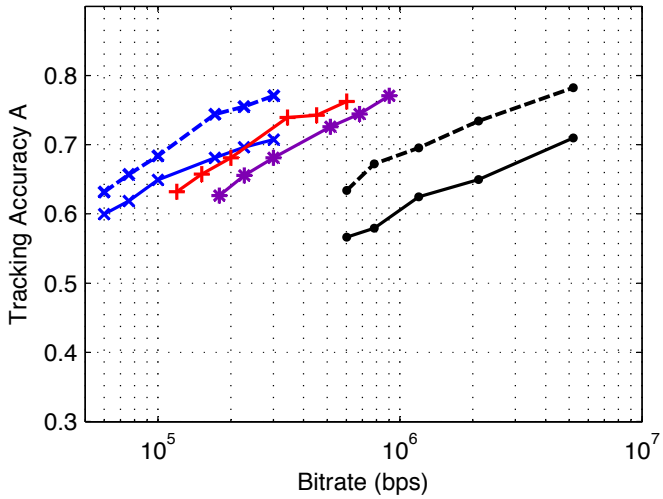
(a) "Golf" - 10% channel loss



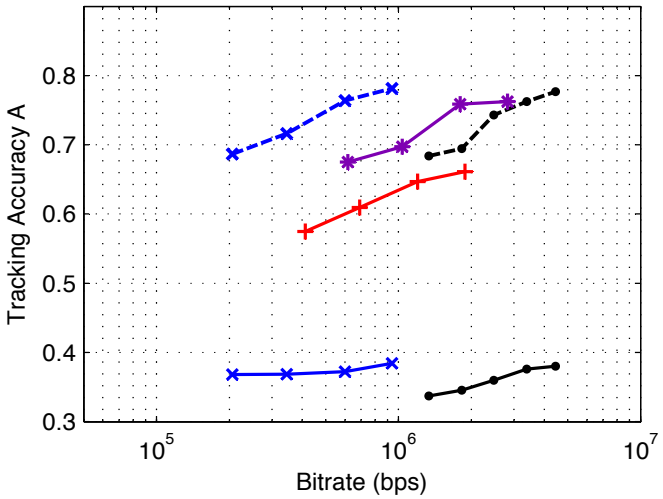
(b) "Golf" - 1% channel loss



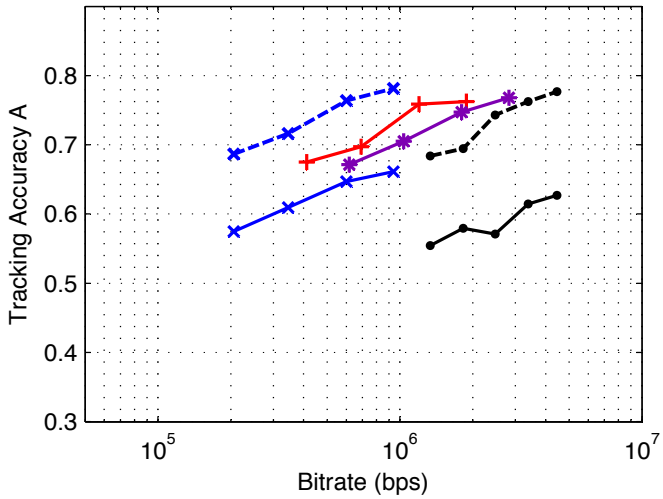
(c) "dt_passat" - 10% channel loss



(d) "dt_passat" - 1% channel loss



(e) "Camera6" - 10% channel loss



(f) "Camera6" - 1% channel loss

Fig. 2. Tracking accuracy across bitrates in the presence of packet loss. Results on the left and right columns correspond to channel loss probabilities of 10% and 1% respectively. Dashed lines identify cases without packet losses compressed using default H.264 from [8] and TDT from [6]. Sequences used are (a-b) "Golf", (c-d) "dt_passat", and (e-f) "Camera6".