QUANTIZATION OPTIMIZED 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 post-compression tracking algorithms. This is problematic for centralized applications such as traffic surveillance systems, where remotely captured and compressed video is transmitted to a central location for tracking. We propose a low complexity optimization framework that automatically identifies video features critical to tracking and concentrates bitrate on these features via quantization tables. Using the H.264 video coding standard and two commonly used state-of-the-art trackers we show that our algorithm allows for over 60% bitrate savings while maintaining comparable tracking accuracy.

Index Terms— Urban traffic video tracking, video compression, optimal quantization

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 video imaging solutions utilize older video compression standards and require dedicated wired communication lines. Recently H.264 has started to be used in transportation applications, significantly reducing the link bandwidth requirement. However, most video compression algorithms are 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 one developed in [1], and an inter-frame object association component such as the one developed in [2, 3].

Most tracking algorithm models account only for the native statistics of video objects, and as a result distortion of these statistics by noise sources, such as compression, severely degrade their accuracy. In [4], special consideration is given to post-compression tracking, and a novel method of spatio-temporally concentrating bitrate via a Region of Interest derived according to statistical behavior is presented. The algorithm presented herein optimizes tracking accuracy for a given bitrate by concentrating available bits in the frequency domain on the features most important to tracking.

Given the special requirements of centrally controlled traffic surveillance systems, it is necessary to limit resource requirements, such as memory and processing power, for any technique seeking to counter the effects of video distortion on tracking. The algorithm presented herein 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 with other standard-compliant 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. In Section 3 we propose our method of bitrate concentration on critical frequencies to guide video compression, for which we show experimental results in Section 4. We present concluding remarks in Section 5.

2. COMPRESSION DISTORTION OF TRACKING

While the active field of video object tracking contains a large variety of algorithms, most of these systems share some fundamental concepts. In reviews of object tracking presented in [5] and [6] 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. 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

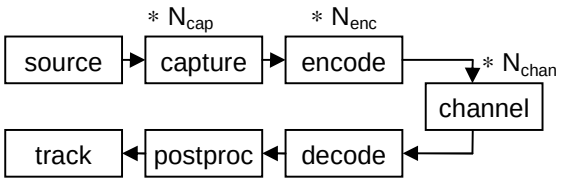

Fig. 1. Typical centrally controlled tracking system. Video of objects to be tracked is acquired (with capture noise N_{cap}) at a remote location, compressed (with encoding distortion N_{enc}), and transmitted over a channel (with channel distortion N_{chan}). At the receiver the transmission is decoded, post-processed and passed on to tracker.

(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 compression and lossy 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 network 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 [7] 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,

$$OLAP = \frac{GT_i \cap AR_i}{GT_i \cup AR_i}, \tag{1}$$

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

$$PREC = \frac{\overline{TP}}{\overline{TP} + \overline{FP}}, \tag{2}$$

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

$$SENS = \frac{\overline{TP}}{\overline{TP} + \overline{FN}}, \tag{3}$$

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

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

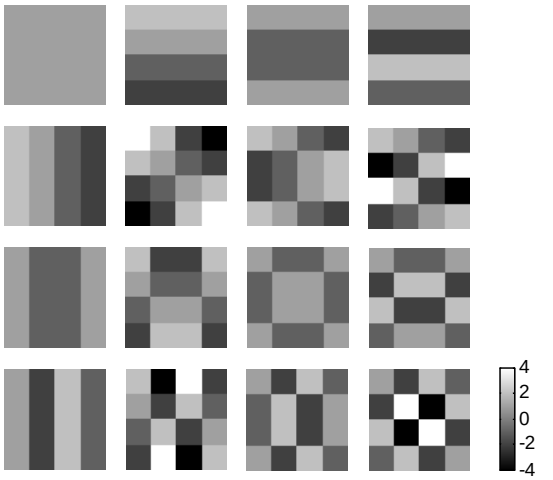

Fig. 2. Transform coefficients represented as per-coefficient basis functions applied to the source 4x4 block. From left to right, top to bottom, the coefficient indices are numbered 0,1,2..15.

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

3. PROPOSED METHOD

The proposed algorithm is an iterative gradient search. For each iteration, the encoder quantization scheme of each individual frequency is modified, and tracking accuracy is measured for a sample clip of video. From these results, only the more favorable modifications in the rate vs. A sense are kept, and subsequent iterations proceed cumulatively. Details for the algorithm are presented below.

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. 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 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 re-

sulting from the 4x4 spatial transform depicted in Fig. 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.

The H.264/AVC standard specifies quantization for a given transform coefficient index q_{idx} in terms of the quantization point (QP) and the QT as

$$\begin{aligned} QT &= [t_0, t_1, t_2, ...t_{15}] \\ QP_{idx} &= QP * \left(\frac{1}{16}\, QT[idx]\right). \end{aligned} \tag{5}$$

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 a gradient search. We will coarsen quantization of frequencies iteratively found to be less useful to tracking, thereby saving more bits per accuracy reduced than if we simply coarsened quantization uniformly across all frequencies.

The gradient search 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 search is said to converge when the set of iteration 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,

$$\begin{aligned} S_{opt} &\rightarrow (A_k < A_{k+n} | R_k < R_{k+n}) \; \forall \; n, k \\ S_{opt}[0] &= argmin\{R_k\} \; \forall \; k, \end{aligned} \tag{6}$$

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

$$\Phi\{T, idx, C\} = T[t_0, t_1, t_2, ... \frac{t_{idx}}{C}, ...t_{15}]. \tag{7}$$

To initialize our search we generate the OPs obtained by updating each entry in T_{init} and applying the result across a given range of quantizers. Of these results we choose the iteration 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 updating entries in their QTs, forming the set of OPs S_i from which the optimal set $S_{i,opt}$ is drawn. Refer to Fig. 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 at 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. 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.

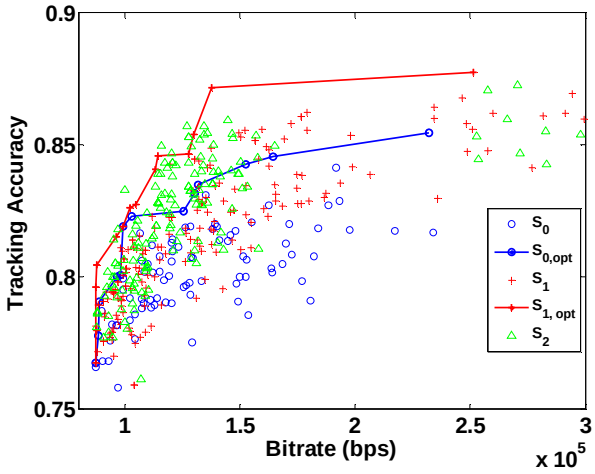

Fig. 3. An example showing the first three iterations of the optimization process in the rate-accuracy domain.

Note that the search 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 quantizated 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 reinitialization 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 will be unacceptably high for most low-cost embedded devices.

An implementational point to note is that the algorithm requires access to the ground truth for operation. In a centrally controlled system such as described in Fig. 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.

Note that while our algorithm involves online iterative searching, a generic "tracking friendly" QT can generated by performing the search over a variety of video content offline. Such a generic QT would likely not be able to match the bitrate gain a QT tailored to the specific scene could deliver. However, it would have the advantage of not requiring a full-duplex channel between the remote and

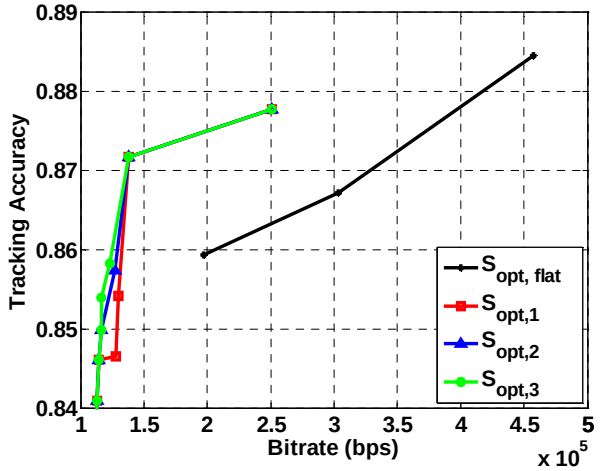

Fig. 4. Rate-accuracy results for the "I90" sequence and "Mean Shift" tracking.

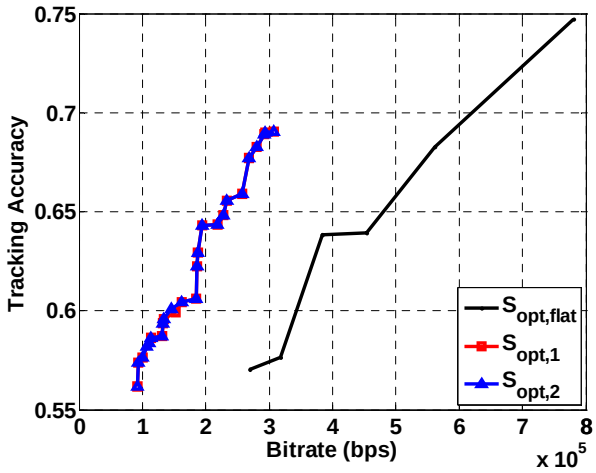

Fig. 5. Rate-accuracy results for the "Golf" sequence and "Connected Component" tracking.

central locations to implement, and it would require no additional startup time or additinal resources at the remote node to operate.

4. EXPERIMENTAL RESULTS

The video compression experiments presented herein have been performed using the open-source H.264/AVC encoder x264 [8]. 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 Sec. 3 to generate the "optimal" uniform quantization performance curve.

For our experiments, the post-processing block shown in Fig. 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 [9] "blobtrack" module was used as the object tracker. In order to keep our work general we defined equal accuracy components weights (i.e. $\alpha = \beta = \gamma = \frac{1}{3}$).

Refer to Fig. 4 for results from experiment using the "I-90" sequence (lightly congested highway traffic) and the Mean Shift tracker described in [2]. 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. 5 for results from experiment using the "Golf" sequence (average congested local intersection) and the "Connected Component" tracker described in [3]. The algorithm was allowed to run for 3 iterations, evaluating a total of 447 OPs. The lower overall tracking accuracies compared to those if Fig. 4 are due to more challenging tracking video being used. 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 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 gradient search can be done offline and needs to be performed only once per system initialization or reset (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.

5. CONCLUSION

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.

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

6. REFERENCES

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

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

[3] A. Senior, A. Hampapur, Y.-L. Tian, L. Brown, S. Pankanti, R. Bolle, "Appearance Models for Occlusion Handling", *Proc. 2nd IEEE Workshop on PETS*, Kauai, Hawaii, USA, Dec. 2001

[4] E. Soyak, S. A. Tsaftaris, A. K. Katsaggelos, "Content-Aware H.264 Encoding for Traffic Video Tracking Applications", *Proc. ICASSP*, March 2010

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

[6] 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

[7] A. Baumann, M. Boltz, J. Ebling, M. Koenig, H. S. Loos, M. Merkel, W. Niem, J. K. Warzelhan, J. Yu, "A Review and Comparison of Measures for Automatic Video Surveillance Systems", *EURASIP Jour. on Im. and Vid. Proc.*, Vol. 2008, Article ID 824726

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

[9] http://opencv.willowgarage.com