7-2010

Analytics-Modulated Coding of Surveillance Video

Lai-Tee CHEOK
Singapore Management University, LAITEECHEOK@smu.edu.sg

Nikhil Gagvani

ANALYTICS-MODULATED CODING OF SURVEILLANCE VIDEO

Lai-Tee Cheok, Nikhil Gagvani

Cernium Corporation
Email: ltcheok@gmail.com, ngagvani@cernium.com

ABSTRACT

Video surveillance systems increasingly use H.264 coding to achieve 24x7 recording and streaming. However, with the proliferation of security cameras, and the need to store several months of video, bandwidth and storage costs can be significant. We propose a new compression technique to significantly improve the coding efficiency of H.264 for surveillance video. Video content is analyzed and video semantics are extracted using video analytics algorithms such as segmentation, classification and tracking. In contrast to existing approaches, our Analytics-Modulated Compression (AMC) scheme does not require coding of object shape information and produces bitstreams that are standards-compliant and not limited to specific H.264 profiles. Extensive experiments were conducted involving real surveillance scenes. Results show that our technique achieves compression gains of 67% over JM. We also introduced AMC Rate Control (AMC RC) which allocates bits in response to scene dynamics. AMC RC is shown to significantly reduce artifacts in constant-bitrate video at low bitrates.

Keywords— H.264, object-based coding, video analytics, video surveillance, rate control

1. INTRODUCTION

Object-based coding techniques for selective video compression already exist in the MPEG-4 standard where a video sequence is treated as a collection of one or more objects of arbitrary shape. An alpha binary map is computed where each pixel belonging to the foreground object is defined as black and pixel corresponding to the background region outside the object is defined as white. A binary shape encoder is then used to encode the alpha binary map and the encoded shape information is transmitted together with the coded video data.

H.264/AVC is the latest video coding standard developed jointly by MPEG and VCEG that offers significantly higher compression gains than MPEG-4. However, one functionality that is absent in the H.264/AVC standard is the ability to code arbitrary shaped objects. Recent work progresses along the direction to incorporate shape coding functionality in H.264/AVC. In [1] and [2], the authors used Flexible Macroblock Ordering (FMO) which is supported only in the H.264 Baseline Profile (BP). Conventional video standard allows the encoding of macroblocks (MB) in raster scan order, whereas with FMO, the slices of macroblocks can be grouped into slice groups freely. In [1], the authors defined separate slice groups for foreground (fg) objects and background (bg) objects. Bg slice groups have a larger QP value than the fg slice groups, resulting in significant bit-rate reduction. The work in [2] adopts a similar approach, however, the fg QP value is determined by the native rate controller [3], whereas an equation is proposed to derive the corresponding bg QP value as a function of additional parameters that determine the sharpness of the quality difference between the fg and bg object. In [4], the authors re-designed the H.264 slice group structure and a new binary shape coder for H.264 video, following a strategy similar to that used in MPEG-4. All the video frames are encoded as I frames and the compressed Binary Alpha Blocks (BABs) are delivered via the auxiliary information stream of the H.264/AVC bitstream.

Our proposed scheme, Analytics-Modulated Coding (AMC), does not require coding of shape information (unlike [4]) and produces bitstreams that are standards-compliant and are not limited to H.264 Baseline Profile (BP). [1] and [2] works only for H.264 BP that supports FMO. Furthermore, we use Structural Similarity (SSIM) [5] instead of PSNR since SSIM correlates better to human perceptual quality. In AMC, video analytics algorithms such as segmentation and classification are utilized to analyze the video content and extract scene semantics for identifying target objects of interest (fg objects) and to determine when target objects appear in the scene or disappear from the scene. The bg and/or non-ROI regions are encoded using fewer bits relative to the identified ROI. A typical video surveillance scene involves one or a few moderately sized target objects and the scene is static most of the time. Adopting our proposed scheme for encoding surveillance video will significantly reduce the bit-rate without compromising quality. The GOP size as well as other encoding parameters (e.g. motion vector search range) are adapted to the scene content for improving compression. Instead of adapting the spatial and temporal resolution which may result in discontinuity in perceptual quality, encoding parameters are adjusted at finer granularity to yield smooth, high-quality video at low bit rate. An object tracking module can also be utilized to yield motion information for optimal I/P/skipped mode decision and for faster motion estimation to reduce computational overhead.

We have also designed and implemented a new Rate Control scheme (AMC RC) that allocates bits in response to scene dynamics. It is worthwhile to mention that AMC does not focus only on selective coding of image regions, but presents other in-

978-1-4244-7493-6/10/$26.00

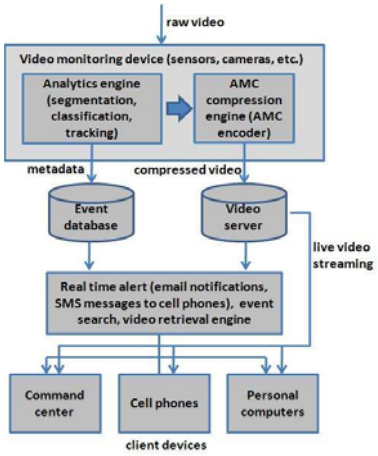


Fig. 1. Architecture of a video surveillance system incorporating AMC

novative ways of exploiting video analytics modules to improve other aspects of compression for achieving higher compression and computational efficiency than H.264. It also significantly reduces storage and bandwidth requirements and offers a competitive solution to surveillance systems that record and stream video continuously.

This paper is organized as follows: Section 2 provides an overview of our video surveillance solution employing Analytics-Modulated Coding (AMC). Section 3 introduces the design of AMC. The new rate control module (AMC RC) is proposed in section 4. We present and discuss the experimental results in section 5 and draw conclusions from our work in section 6.

2. ARCHITECTURE OF A VIDEO SURVEILLANCE SYSTEM INCORPORATING ANALYTICS-MODULATED CODING (AMC)

Figure 1 illustrates an architecture of a video surveillance and video compression solution employing AMC. Both the AMC encoder and the video analytics components run on a single device (e.g., IP cameras, sensors, etc.). Video input to the device is processed by the analytics engine which comprises of a segmentation, a classification and a tracking module.

The segmentation module produces foreground blobs which are inter-connected clusters of pixels. Our current surveillance system uses a variant of the background subtraction technique [6] in which a background model is constructed and updated. Each subsequent frame is compared with the background model to yield foreground blobs which are classified by assigning a category to each of the blobs. The classification engine uses image features to discriminate one class from another, for instance, a multi-class classifier [7] separates object blobs into one of multiple classes, e.g., object is a person, a vehicle, or an animal. Classified objects are then tracked [8] across multiple video frames by establishing a correspondence between the object blobs in consecutive frames. These correspondences can be

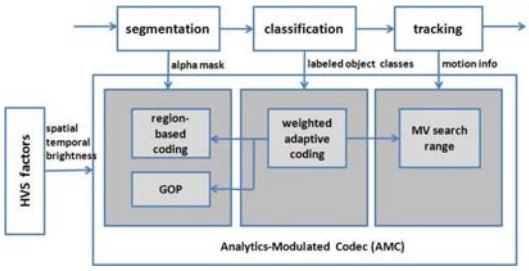


Fig. 2. Architecture of AMC

used for scene interpretation and for behavior/activity recognition. As shown in the figure, metadata from the analytics engine is stored in an event database while the encoded video from the compression engine is stored on a video server. Our surveillance system generates real time alerts of activities of interest to the users, by posting SMS messages or sending email notifications to client devices (e.g., users' cell phones, PCs, or the command center). Users can interact via a web-based interface such as querying for a specific event from the event database and have event clips delivered to their devices.

In existing systems, the analytics engine runs independently of the compression engine (encoder). However, our AMC encoder utilizes output from the analytics engine (shown as block arrow in Figure 1), for scene-adaptive coding to produce high quality low bit-rate H.264 video for storage on video servers and/or delivery to client devices.

3. ANALYTICS-MODULATED CODING (AMC)

Our proposed Analytics-Modulated Coding (AMC) scheme aims to leverage analytics components to enable scene-adaptive coding to achieve better compression and computational efficiency than H.264. As shown in Figure 2, the binary alpha mask generated by the segmentation module is used to identify the regions of interest (ROI). A ROI may correspond to regions where a target person is identified in the scene/picture. The GOP size is adaptively adjusted based on whether the segmented object enters or leaves the scene. The segmented objects are classified as belonging to 1 of N classes of objects through the classification process. Weights are assigned to these classes to define relative priorities among the classes for bit allocation. For example, blobs belonging to one class (e.g., person class), use a greater fraction of the bit budget compared to blobs belonging to another class (e.g., trees class). The classified objects are tracked over successive frames by establishing correspondence between blobs. The tracking module yields motion information for each object that is utilized to determine a suitable motion vector search range. The interactions between the modules are described in more details in the following subsections.

3.1. Region-based Coding by Modulating QP Value

The mask from the segmentation module is used to identify ROIs so that non-ROIs are coded with relatively fewer bits in

such a way that the overall bitrate is reduced without compromising quality. As mentioned, unlike existing approaches, our scheme does not require object shape information to be encoded, is not limited to specific profiles in H.264 and produces H.264 bitstreams that are compliant to the standard.

We have implemented our proposed scheme into H.264 JM reference software (JM 15.1) [9], and integrated with our analytics engine. As shown in Figure 2, the alpha mask derived from the segmentation module is used to distinguish foreground (fg) objects (i.e., ROIs) from background (bg) objects (non-ROIs). In our surveillance application where the aim is to detect persons, the fg objects refer to detected person targets, while the bg objects refer to objects other than persons, such as vehicles, animals, moving tree foliage, etc. Each class of object can be further assigned a weight and a different QP value could be associated with different object class based on their relative priorities. The mask defining the boundary of the ROI are in pixel units. We have mapped the ROI from pixel to macroblock (MB) units and defined an array indexed by macroblock address to keep track of MBs belonging to fg and bg objects. Extensive experiments have been conducted using two hour long real surveillance video and are described in more details in Section 5. The proposed feature discussed in this section allows AMC to operate in VBR mode. AMC in CBR mode involving RC is covered in Section 4.

3.2. Adaptive I-Frame Placement (Adaptive GOP Interval)

The analytic modules can be used to detect scene changes. An initial large GOP interval is used and a frame is encoded as an I-frame whenever scene change occurs. This has the effect of adaptively varying the GOP size depending on the scene content. An example of a scene change is when a target object (e.g., a person) is detected entering or leaving the scene. Our implementation uses both the segmentation and classification modules to detect scene change. Frames are coded as P frames up to the moment that an object enters the scene. A maximum GOP size is specified such that an I-frame can be inserted when the period of inactivity exceeds certain predetermined duration or predetermined criterion (e.g., number of frames of inactivity). A minimum GOP size is specified such that two I frames are at least a certain duration apart. We have encoded several surveillance videos with a combination of minimum and maximum GOP sizes. Details of the experiments are given in Section 5.

3.3. I/P/Skipped Mode Decision

Existing algorithms on I/P/Skipped mode decision rely on the computation of Mean Absolute Difference (MAD) and comparing that against a threshold to determine if the MB should be an I, P or skipped MB. Such an approach, however, incurs higher computation cost. Furthermore, the threshold chosen may not guarantee that all fg MBs will be coded as I MBs. Our proposed approach allows the use of segmentation and classification output to directly determine the I/P/skipped mode selection. When

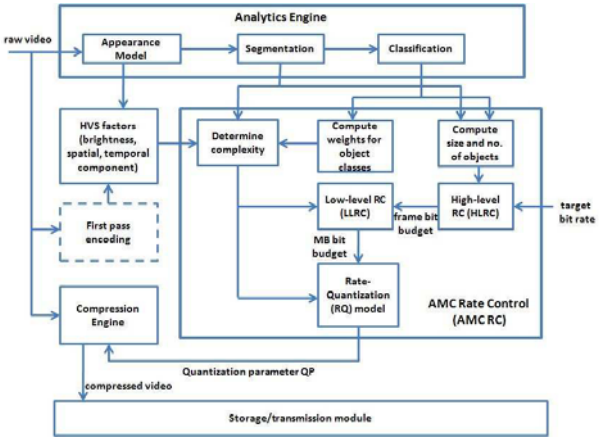


Fig. 3. Analytics-Modulated Coding with Rate Control (AMC RC)

a scene change occurs, the fg MBs are coded as I MBs. bg MBs in regions having movement (e.g., foliage, fountain, etc.) are coded as P MBs whereas MBs corresponding to static bg regions are coded as skipped MBs. Results show about 40 - 50% compression gains by coding static bg MBs as skipped MBs and fg MBs as I MBs.

3.4. Determining Motion Vector Search Range

A motion trajectory for an object blob from the tracking module can be used to determine a suitable MV search range with a good trade-off between accuracy and complexity. Such motion information can also be used to derive a predictive motion vector used in predictor-based ME algorithms. The MV search range can also be determined based on the result of the classification engine. As an example, for segmented blobs classified as a person, the lower half of the ROI corresponding to the leg movement of the detected person target can be used to determine the search range, which can further be adaptively changed depending on the motion history of objects.

4. ANALYTICS-MODULATED CODING WITH RATE CONTROL (AMC RC)

We have also designed and implemented a new rate control scheme by leveraging analytics. Figure 3 illustrates the AMC RC module and its relationship with the analytics and compression engines. Raw video is processed by the appearance model [10] in the analytics engine to yield brightness, spatial and temporal information that contribute to the HVS factors. Alternatively, these HVS factors can be derived through first-pass encoding as shown encased within the dotted box. The segmentation module of the Analytics engine identifies foreground blobs which are then categorized by the classification module. Within the AMC RC block, the inputs to the High-level RC are target bitrate, weights for object classes, number of objects and size of objects. The latter three are derived from the segmentation and classification modules. Complexity is computed based on

segmentation output and the weight for each object class. Complexity is used to drive the Low-level RC and the RQ model which is described in detail in the following sections.

4.1. AMC RC High-Level Rate Control (AMC RC - HLRC)

At the HLRC, the target bit allocation for each frame can be dependent on the number of remaining frames in the GOP, number of remaining bits available for the GOP as well as scene complexity, which is expressed as a function of the number, the size and the type of objects in the scene. These three quantities are derived from the analytics module. The mathematical formulations for calculating frame budget at the HLRC, while incorporating analytics are as follows:

$$T_i = \frac{X_i}{((N-i+1)X_i)+c}\left(R_t - \sum_{j=1}^{i-1} R_j\right) \quad (1)$$

$$X_i = \sum_{k=1}^{n} \alpha_{k,j} F_{k,j}, \quad F_{k,j} \epsilon \{a_i, b_i\} \quad (2)$$

$$a_i = \frac{\sum_{g=1}^{m_i} S_{g,i}}{S}, \quad b_i = \frac{m_i}{M} \quad (3)$$

For a GOP with N frames, the bit budget for frame i, T_i, is computed by taking a proportion of bits remaining in the GOP as shown in equation (1). R_t refers to the total bit budget whereas R_j gives total bits used to encode frame j and c is a constant. X_i is the complexity of frame i due to multiple features and, as an example, is expressed as weighted combination of the normalized size of objects, a_i, and normalized number of objects, b_i, in the scene as shown in equations (2) and (3). $\alpha_{k,j}$ is the weight associated with $F_{k,j}$ which denotes feature k of frame i. In equation (3), $S_{g,j}$ refers to the size of object g (in pixels) in frame i; m_i is the number of objects in frame i, while S and M are the total number of pixels in the video frame and the expected maximum number of objects respectively.

$$T_{i\_final} = \gamma TB_i + (1-\gamma)T_i \quad (4)$$

The final target bit budget, T_{i_final}, is allocated for each frame as shown in equation (4). T_{i_final} is the weighted sum of the bit budget computed based on buffer occupancy (i.e., TB_i), and the bit budget predicted based on complexity (i.e., T_i), as derived from equation (1). γ indicates how much to weight each component (TB_i and T_i) of the sum. We adopted the buffer occupancy scheme in JM.

4.2. AMC RC Low-Level Rate Control (AMC RC - LLRC)

At the LLRC, bits for each macroblock (MB) are allocated as a fraction of the frame bit budget and the MB complexity. In our scheme, the MB complexity is calculated based on analytics components.

$$r_j = (x_j / \sum_{j=1}^{p} x_j) T_{i\_final} \quad (5)$$

$$x_j = \sum_{k=1}^{n} \lambda_{k,j} f_{k,j} \quad (6)$$

$$c_j = (1-g_j) * 1/S_j \quad (7)$$

$$d_j = (1-g_j) * 1/M_j \quad (8)$$

$$e_j = (1-g_j) * 1/I_j \quad (9)$$

$$h_j = g_j w_j, \quad (10)$$

As shown in equation (5), the bit budget for each MB, r_j, is computed as a fraction of frame bit budget, T_{i_final}, based on MB complexity, x_j. p is the total number of MBs in each frame. x_j is expressed as a weighted linear combination of features $f_{k,j}$, where $f_{k,j} \epsilon \{c_j, d_j, e_j, h_j\}$ and $\lambda_{k,j}$ are weights associated with each of these features (equation (6)). These features include HVS factors such as brightness, spatial activities and temporal activities, denoted as normalized quantities I_j, S_j and M_j respectively (equations (7) - (9)). Using h_j, analytics are incorporated into the calculation of the MB complexity. g_j, in equation (10), indicates whether the MB belongs to the foreground or background object and is derived based on the segmentation module. The classification module is used to compute the normalized weight for each object class, w_j, $j \epsilon \{1, \ldots, L\}$, where L is the number of object classes. For example, a larger weight can be assigned to a PERSON object and a smaller weight can be assigned to a CAR/VEHICLE object and/or other objects in an application used to detect target persons in the scene. This directly incorporates weights based on object class into the rate control technique used for compression. We used the quadratic R-Q model in equation (11) to derive MB QP.

$$r_j = K_i * \frac{x_j}{QP} + K_2 * \frac{x_j}{QP^2} \quad (11)$$

r_j and x_j, are derived from equation (5) and (6) respectively. K_1 and K_2 are model parameters. In contrast to schemes that uses MAD between pixels in an original image and pixels in a predicted image, a combination of segmentation and classification results from analytics is used, as well as HVS factors to compute the MB complexity.

5. RESULTS AND DISCUSSION

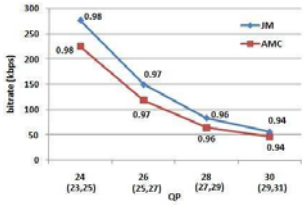
We implemented our AMC encoder based on the H.264 JM reference software (JM 15.1) [9] and ran experiments on real surveillance video containing both indoor and outdoor scenes with different levels of motion activity. Each two hour long video is encoded at 30fps with CIF resolution (352x240). We have also conducted experiments on HD video and found that the compression gain is higher (approx. up to 10%). We use SSIM since it correlates better to subjective quality compared to PSNR. We present results of 3 video sequences in the following subsections and evaluated the performance of our proposed

(a) Building Entrance

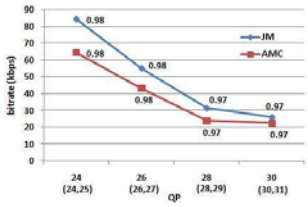
(b) Trailer

(c) Parked Car

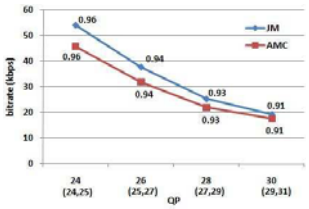
Fig. 4. Real Surveillance Scenes used for Experiments



(a) Building Entrance



(b) Trailer



(c) Parked Car

Fig. 5. Results on Region-based coding using QP modulation

compression scheme against H.264 JM baseline profile. The 3 video sequences consist of the scenes of a building entrance, a trailer, and a parked car as shown in Figure 4.

5.1. Experimental Results for Region-based Coding by Modulating QP

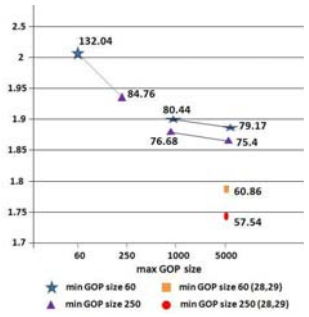
Figure 5 shows the bitrate, QP and SSIM values of video encoded using JM and AMC for the 3 video scenes. SSIM values are shown as data labels on the points along the curves. AMC (fg,bg) QP values appear in brackets beside the JM QP values along the x-axis. In Figure 5(a), the building entrance scene is first encoded using JM with a QP value of 24. The resulting bitrate is 276.64kbps and SSIM value is 0.98. The same video is encoded using AMC with a (fg,bg) QP value of 23 and 25. The resulting bitrate at the same SSIM value is 224.84kbps giving a bitrate savings of 18% over JM. The bitrate and corresponding SSIM for QP of 24, 26, 28 and 30 are shown. Results for other scenes are illustrated in Figures 5(b),(c) and (d). As shown, AMC via QP modulation achieves compression gain of up to 24%.

5.2. Experimental Results for Adaptive I-Frame Placement

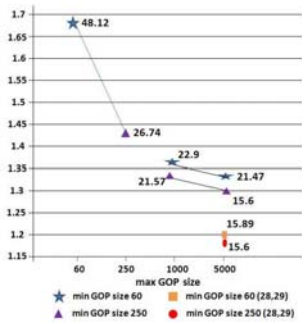
Figure 6 describes results from experiments conducted on the surveillance scenes shown in Fig. 4, with low and high activity, using a minimum GOP size of 60 or 250 and a maximum GOP size of 60, 250, 1000 and 5000. The results are presented in the graph using data symbols * and 'triangle'. The 'square' and 'circle' symbols represent results that will be discussed in the next section. For JM, the min and max GOP size is the same, i.e., min, max size of (60,60) refers to using JM at a GOP size of 60. As shown in Figure 6(a), JM bitrate decreases from 132.04kbps to 84.76kbps as GOP size increases from 60 to 250. AMC bitrate reduces from 80.44kbps to 79.17kbps when max GOP size increases from 1000 to 5000 at the same min GOP size of 60. Similar results are achieved at min GOP size of 250 and for other video scenes in Figure 6(b) and 6(c). The amount of bitrate savings is dependent on scene activity. Relatively static scenes yields higher savings than scenes with more activity. Overall, bitrate reduction as high as 55% is achieved.

5.3. Combined Results

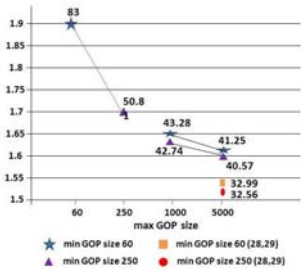
The results of combining Region-based coding via QP modulation and adaptive I-frame placement are shown represented as 'square' and 'circle' symbols in Figure 6. The y-axis indicates bitrate (kbps) in log scale. Bitrate also appears as data labels alongside the symbols. The "square" corresponds to results ob-



(a) Building Entrance



(b) Trailer



(c) Parked Car

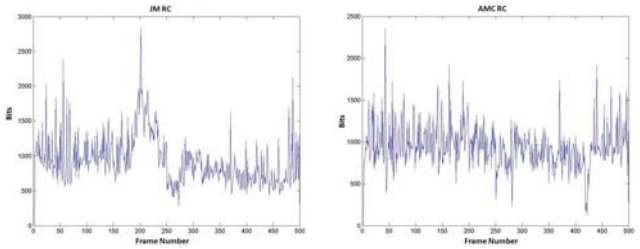
Fig. 6. Combined Results on Region-based coding (QP Modulation) and Adaptive I-Frame Placement

tained after encoding video using AMC encoder with a minimum GOP size of 60, a maximum GOP size of 5000, and using fg and bg QP values of (28, 29). The results obtained using a

minimum GOP size of 250, a maximum GOP size of 5000, and fg and bg QP values of (28, 29) is indicated by the "circle" symbol. As shown, combining both features yields bitrate savings of up to 67%.

5.4. AMC RC Results

Figure 7 compares results of JM RC and AMC RC. As shown, AMC RC yields better bit allocation compared to JM RC which produces spikes and dips over frame interval 180 - 300.



(a) JM RC (b) AMC RC

Fig. 7. Results of AMC RC

6. CONCLUSION

We proposed a new compression scheme leveraging analytics to achieve better compression than H.264. Video content is processed by analytics and scene semantics are extracted to adaptively vary the GOP size and to allow selective coding of regions. The proposed AMC technique directly incorporates object-based coding capability into H.264 without the drawbacks suffered by existing approaches. Experimental results show that AMC can achieve approximately 67% improvement in coding efficiency/gain. We have also developed AMC RC to allow AMC to operate in CBR mode. For future work, we will extend AMC to adaptively vary the number of reference frames, MV search range, as well as to improve intra-prediction, and to combine AMC technique with SVC.

7. REFERENCES

[1] Y.Dhondt, "Flexible macroblock ordering as a tool to ease video adaptation," *Sixth FirW PhD Symposium, Ghent Univ*, vol. 95, Nov. 2005.

[2] W.A.C.Fernando P.Sivanantharasa and H.K.Arachchi, "Region of interest video coding with flexible macroblock ordering," in *First Intl Conf. on Industrial and Information Systems, ICIIS*, Aug. 2006.

[3] F.Pan S.W.Ma G.N.Feng K.P.Lim X.Lin S.Rahardja H.Q.Lu Z.G.Li, W.Gao, "Adaptive rate control with hrd consideration," in *Proc. 8th JVT meeting*, Geneva, Switzerland, May 2003, pp. 23–27.

[4] H.E. Bez X. Li, E.A. Edirisinghe, "Selective compression of video with h.264/avc," in *Proc. 6th IASTED Intl Conf. on Visualization, Imaging and Image Processing*, Palma de Mallorca, Spain, Aug. 2006.

[5] H.R.Sheikh E.P.Simoncelli Z.Wang, A.C.Bovik, "Image quality assessment: from error visibility to structural similaritg," *IEEE Trans. Image Processing*, vol. 13, no. 4, Apr. 2004.

[6] C. Stauffer and W.E.L. Grimson, "Adaptive background mixture models for real-time tracking," in *CVPR99*, 1999, pp. II: 246–252.

[7] L.M.Brown, "View independent vehicle/person classification," in *Proc. ACM 2nd Int"l Workshop on Video Surveillance and Sensor Networks*, New York, USA, 2004, pp. 114–123.

[8] M.Shah A.Yilmaz, O.Javed, "Object tracking: A survey," *ACM Computing Surveys*, vol. 38, Dec. 2006.

[9] JVT, "H.264/avc reference software (jm15.1)," in *http://iphome.hhi.de/suehring/tml/*, Jan. 2009.

[10] D.Snow P.Viola, M.J.Jones, "Detecting pedestrains using patterns of motion and appearance," *Int"l Journal of Computer Vision*, vol. 63, pp. 153–161, 2005.