

Surveillance Video Coding with Quadtree Partition Based ROI Extraction

Peiyin Xing^{1,2}

¹The Shenzhen Key Lab for Cloud Computing
Technology & Application (SPCCTA)
Shenzhen Graduate School, Peking University
Shenzhen 518055, P.R. China
pyxing@pku.edu.cn

Yonghong Tian², Tiejun Huang², Wen Gao²

²National Engineering Lab for Video Technology,
School of EE & CS, Peking University
Beijing, 100871, P.R. China
{yhtian, tjhuang}@pku.edu.cn

Abstract—To reduce the surveillance video coding cost, it is intuitive to encode surveillance videos by dealing with the foreground objects and the background separately. One widely used method following this strategy is Region-of-Interest (ROI) based coding. To achieve significant improvement for the coding efficiency of ROI based methods, this paper presents a surveillance video coding method with High Efficiency Video Coding (HEVC) quadtree partition based ROI extraction. With automatically generated foreground mask and modeled background frame, a ROI extraction following the block partition in HEVC's quadtree structure is firstly performed. Afterwards, surveillance videos can be compressed by coding two-layer videos. One is the *ROI-layer video* generated by merging ROIs and background data in each frame together. The other is the *background-layer video* produced by subtracting the ROIs from the original input video. Results show our method can achieve remarkable total bit-rate saving and significant bit-rate cost reduction on ROIs.

Keywords: *surveillance video, High Efficiency Video Coding (HEVC), region of interest (ROI), background modeling*

I. INTRODUCTION

Surveillance videos, playing an important role in safety and communication domains, are usually captured by stationary cameras at a fixed location for a long time. In surveillance applications, video archives are always stored for a long time, which leads to large storage and bandwidth cost. Due to the rapid development of the video coding standardization, significant improvements have been shown in video compression capability, which enables the storage and transmission cost reduction. Recently, High Efficiency Video Coding (HEVC) [1], the latest video coding standard in which quadtree block partition is applied, can achieve about 50% bit-rate reduction against its predecessor H.264/AVC. However, surveillance videos have their own specialized characteristics and none of the coding standard is especially designed for them. Thereby more efficient encoding strategies can be used in surveillance domain. Considering the long-time static background characteristic, a background modeling based coding scheme was proposed by our previous work [2], which can achieve significant bit-rate reduction while encoding the surveillance videos.

Nevertheless, there is still much room for further bit-rate reduction and subjective quality improvements. Intuitively, surveillance videos should be encoded by dealing with the foreground objects and background separately. Region-of-

Interest (ROI) based coding is the widely used method to encode the ROIs and background respectively. ROIs, the interest parts detected from a given scene, can be extracted from the video sequences. The main idea to encode the ROI in the video is to reduce the bit-rate by a degradation of the visual quality of the non-ROI area. Recently, various ROI encoding techniques have been investigated for video communication systems. A novel ROI based rate control algorithm was proposed by Yang et al [3], which determines the quantization parameter (QP) for the ROI according to the user defined interest level, and allocates bits between ROI and non-ROI regions adaptively. Detecting ROI regions using texture contrast and motion features to meet the low power requirement of portable application was introduced by Wang et al [4]. A dynamic parameter allocation scheme to reduce the computational complexity was applied after getting ROIs in [4]. Liu et al. [5] presented region based computational power and bit allocation by adjusting encoding parameters adaptively after using frame difference and skin-tone to detect ROI.

However, all above schemes focused on reducing the computational encoding complexity and most of the traditional detected ROI were square regions. The bitrate reduction is much less than our previous work [2]. To realize more bit-rate reduction for ROI based coding, we introduce a surveillance video coding method with HEVC quadtree partition based ROI extraction in this paper. Firstly, we embed background modeling into our method to generate a background frame, which will be encoded into stream for long-term prediction. Secondly, Gaussian Mixture Model (GMM) algorithm is used to produce an initial foreground mask for each original frame. Afterwards, the foreground mask and the modeled background frame are used to automatically extract the ROIs in each frame with the quadtree block partition of HEVC coding structure. According to the finally recognized ROIs, it is reasonable to compress videos in forms of code-streams for ROIs and background data respectively. To achieve a higher ROI coding efficiency, we propose to compress ROIs by using the modeled background frame as long-term reference to encode the so-called *ROI-layer video*, which is generated by merging ROIs and background data in reconstructed background frame together. For a high-efficiency background compression, background data are compressed by encoding the background-layer video which is produced by subtracting ROIs from the original input video, also with long-term background reference.

While decoding such video streams, in addition, we can reconstruct scalable videos including the decoded ROI-layer videos with unchanged background and the original videos which merge the ROI-layer videos and the background-layer video together. It should be noted that, while merging the background-layer and ROI-layer videos, we will fill the ROI data into the background-layer according to whether the co-located data in background-layer is empty. Experimental results show that, compared with the HEVC test Model (HM) which performs better coding efficiency than the state-of-the-art ROI based methods, our method can averagely achieve scalable surveillance video coding with 50% total bit-saving and 15% bit-saving on ROIs.

The rest of this paper is organized as follows: Sec. II introduces our video coding framework in detail, Sec. III presents the experimental results, and Sec. IV concludes this paper.

II. THE PROPOSED METHOD

Our method engages to improve the surveillance video coding efficiency of ROI based methods. The first problem of traditional ROI based methods is that, the typical characteristics of the long-time static background is not fully exploited to remove the background coding redundancy and optimize ROI extraction precision. Thus in this paper, background modeling and modeled background based long-term prediction are employed to realize better background prediction for surveillance video coding and foreground mask generation for ROI extraction. The second problem lies in that traditional ROIs are usually described by square regions, which not only takes too many uninterested data in account but also produces an obstacle for adopting the quadtree coding structure of HEVC for high coding efficiency. Fig. 1 shows the example of ROI description in traditional method. Thereby, we further propose a not-square-but-quadtree-partition based ROI extraction in this paper. Moreover, to further remove the background redundancy, we perform video coding after dividing the input video into ROI layer and background layer. Fig. 2 shows video coding framework of our method.



Original frame Traditional ROI
Figure 1. Example of the ROIs in traditional method

As shown in Fig. 2, a background frame is firstly modeled from the original input sequence for Foreground Mask Generation, Background Encoding. Afterwards, the foreground mask for each frame is produced for the following Automatic ROI Extraction and the background frame is encoded into stream to reconstruct the long-term background reference. Thirdly, the ROI Layer and Background Layer Coding utilize the reconstructed background and the extracted ROIs of each frame to construct and encode the so-called ROI-layer and background-layer videos. In the encoding procedures, both of them employ the reconstructed background as long-term reference for each frame. At last, the code-streams of the above two-layer videos

are merged together. At the decoder side, customers can selectively only decode the ROI-layer video in which the background data is almost static or compensate both two videos together in which the background data is more realistic.

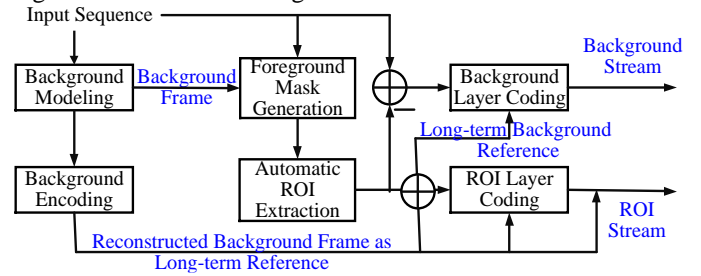


Figure 2. Coding framework of our method

A. Background Modeling, Encoding and Updating

We take our previous low complexity segment-and-weight based running average in [6] as the background modeling algorithm in our method. In general, the method can be summarized as the following five steps: (a) initializing average values and corresponding weights, (b) calculating the threshold for temporal segmenting, (c) creating a new segment or widening the current segment, (d) updating the average values and (e) calculating the final background value. The background frame should be updated periodically. We still follow [6] to update each background frame every super group of frames to avoid the bit-allocation problem and realize a no-delay coding. In order to produce a high-quality background and guarantee the decoding match, the modeled background frame is encoded into the encoding stream with lower QP. Supposing qp is the QP for ordinary frame coding, we use $qp-10$ to compress the modeled background frames.

B. Foreground Mask Generation

Firstly, we use GMM algorithm [7] to generate an initial foreground which may contain many noise pixels for each original frame. However, as shown in Fig. 3, these noise pixels are always isolated, so we can use connected region division algorithm to generate the foreground objects. In our method, we use four-connected region filling method. By searching every pixel's 4 adjacent pixels, this method can get a considerable optimal foreground objects. The four-connected region filling method can be described by Algorithm 1 which recursively recognizes ROI from foreground pixel at start position (x, y) . Among the recognized ROIs by Algorithm 1, only the ROIs having more than 64 pixels are the finally extracted. In such way the noise and uninterested objects can be removed.

Algorithm 1
<p>Procedure FourFill ($x, y, ImgW, ImgH$: integer) var isForeground: bool; isForeground=CurrentPixelIsForeground(x, y) if (isForeground && ($x < ImgW$ && $y < ImgH$) && (x, y) is available and never visited) SetCurrentPixelVisited(x, y); FourFill ($x+1, y, ImgW, ImgH$); FourFill ($x-1, y, ImgW, ImgH$); FourFill ($x, y+1, ImgW, ImgH$); FourFill ($x, y-1, ImgW, ImgH$); endif</p>

Furthermore, with the help of the modeled long-time static background frame by Background Modeling, a more-frequent GMM model updating method is conducted to weaken the influence of short-term stationary foreground for mask generation. The modeled background frame is used as the baseline for whether a stationary foreground appears in the frame.



Original frame Initial mask
Figure 3. Example of the initial mask in our method

C. Automatic ROI Extraction

In order to deal with the foreground objects and background separately, we try classifying the to-be-encoded coding units (CUs) into different categories. An analysis experiment is conducted on the HM10.0. After a background frame is generated, we categorize the CUs into Background CUs (BCUs) or Foreground CUs (FCUs) through the foreground or background property of its inside Basic Units (BUs, which are 4x4 blocks).

Donating b_i as the i -th pixel value of current BU and bg_i as the i -th pixel value at the corresponding position in the modeled background frame, then we can calculate the property $P(b)$ of a BU b as foreground F or background B by

$$P(b) = \begin{cases} F, & \sum_{i=1}^{i=16} abs(b_i - bg_i) > \alpha \\ B, & \sum_{i=1}^{i=16} abs(b_i - bg_i) \leq \alpha \end{cases} \quad (1)$$

This means the current BU b will be judged as foreground if the sum of difference exceeds the threshold value α (80 in our experiment). Otherwise, it will be judged as background. With each BU's property, we get each CU's category $C(c)$ through calculating and comparing the proportion of foreground BUs of current CU c . Supposing $\|X\|$ represents the size of a set X , $b(i)$ is the i -th BU in c , and $2N \times 2N$ is the size of c , then the calculation process is

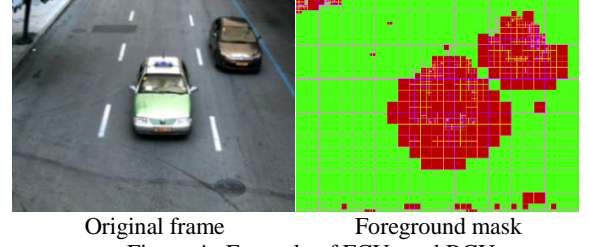
$$C(c) = \begin{cases} FC, & 16 \times \frac{\| \{i | P(b(i)) = F\} \|}{N^2} > \varepsilon \\ BC, & 16 \times \frac{\| \{i | P(b(i)) = F\} \|}{N^2} \leq \varepsilon \end{cases} \quad (2)$$

where ε is practically set to 0.0625. All the thresholds are obtained from the analysis experiment. Under our constant threshold, the BCUs and FCUs will be more consistent with the scene. This means if the foreground BUs proportion is no more than 1/16, the current CU will be categorized as BCU; otherwise it will be one FCU. Fig. 4 shows an example of the different CUs, in which the green represent the BCUs and the red are the FCUs.

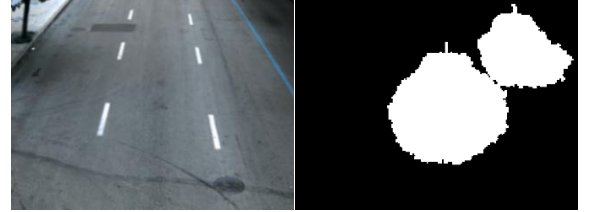
Inspired by the CU classification, we introduce an HEVC quadtree block partition based ROI extraction. With the initial foreground mask produced by GMM and modeled background frame as input, we can modify the mask $M(c)$ of current CU c . Supposing $gmm(i,j)$ is the initial mask value at (i,j) in c and $2N \times 2N$ is the size of c , then the modification process is

$$M(c) = \begin{cases} 1, \exists i, j, 0 \leq i, j < 2N \rightarrow gmm(i, j) = 1 \vee C(c) = FC \\ 0, \forall i, j, 0 \leq i, j < 2N \rightarrow gmm(i, j) = 0 \wedge C(c) = BC \end{cases} \quad (3)$$

$M(c) = 1$ means all the pixels in c will be foreground, which forms the final quadtree ROI. Fig. 5 shows the example of the final foreground mask. Compared with the traditional square ROI regions, ROIs in our method contain less background and are consistent with the quadtree block partition.



Original frame Foreground mask
Figure 4. Example of FCUs and BCUs



Modeled background Extracted ROIs
Figure 5. Example of the mask in our method

D. Encoding Procedure

After ROI is extracted, the input sequence is divided into ROI layer and background layer through corresponding foreground mask. Supposing p is the current pixel to be layered, $m(i)$ is the mask value of pixel i , then the layer of current pixel $L(p)$ is

$$L(p) = \begin{cases} ROI & , m(p) = 1 \\ Background & , m(p) = 0 \end{cases} \quad (4)$$

For the ROI layer, the background will be replaced by the reconstructed modeled background frame. Besides, the corresponding ROI positions will be set to zero for the background layer.

After layering, different encoding strategies are applied to different layers. And as discussed above, for the ROI layer, we only retain the ROI parts of a frame, other parts will be replaced by the reconstructed modeled background frame. Note that, the frames without any ROIs will be dropped totally. We just need to encode a "picture distance" to the ROI stream to indicate how many frames have been dropped between two frames that contain ROIs. The ROI layer and background layer are encoded into ROI stream and background stream respectively. To guarantee the efficiency and decoding match, the modeled background frames are encoded into the two streams for long-term prediction.

E. Scalable Video Reconstruction

At the decoder side, we can reconstruct scalable videos. It depends on the users' preferences or terminal capabilities.

a) If users want to check the video scene, we just need to decode the first background frame in the background stream;

b) If users want to check to passing-by objects, we can obtain ROIs with constant background by adding ROIs and the static background together, since the modeled background frame is encoded into the stream.

c) If users want to fully reconstruct the original video, we should reconstruct ROIs with realistic background pixels by

decoding the two streams and adding the decoded pictures together. Of course, we never add the background data on the background data of ROI-layer video. The detailed adding procedure compensates the empty data in background layer with the co-located ROI data in ROI layer video. This means no mask data are required and the total bitrate is saved.

III. EXPERIMENTAL RESULT

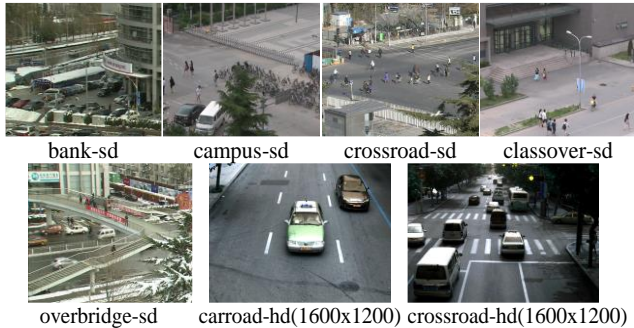


Figure 6. Surveillance videos used to evaluate our method

A. Experimental Setup

To verify our method, the original HEVC encoder (denoted by HMO) and optimized encoder which utilizes key frame (the first I-frame) as long-term reference (denoted by HMLT) are utilized as the anchors for comparison. Because HMO and HMLT always perform better coding efficiency than the traditional ROI based coding, we can indirectly estimate the improvement of our ROI based methods. The experiment is implemented on HM10.0 under the low-delay common test conditions [8] for the real-time surveillance videos with BD Rate/PSNR as metrics and QPs of {22, 27, 32 and 37}. Moreover, the experimental dataset are seven SD&HD surveillance videos with 900 frames. Fig. 6 gives the example frames for the surveillance videos with different motion characteristics, captured by static cameras.

B. Results

As shown in Tab. I, reconstructing the original video by merging the decoded ROI layer and background layer, compared with HMO, our method can achieve 42.17% (SD)/50.29% (HD) bit-rate reduction averagely. Compared with HMLT, our method can achieve 26.40% (SD)/36.80% (HD) bit-rate reduction averagely. The main reason of the encoding efficiency improvements is that, we carry out the ROI-layer and background-layer coding using long-time static background to remove the redundancy. In detail, we can also find that smaller bit-rate saving will be achieved on the sequences with larger foreground objects. For example, no clean background can be modeled for the crossroad-hd with lots of tightly-moving cars, so it has the least bit-saving.

In addition, we use the traditional square ROI to replace the quadtree ROI in our method as another anchor to evaluate the efficiency of our quadtree ROI. As shown in Tab. II, comparing the ROI layer streams, with similar PSNR on ROIs at each QP, our method saves about 15% bit-rate averagely on coding the ROIs, because our ROI regions contain less background. The result is very meaningful for the users who only require the ROI streams.

TABLE I. CODING EFFICIENCY COMPARISON WITH HMO AND HMLT

definition	sequence	Proposal vs HMO		Proposal vs HMLT	
		BD Rate	BD PSNR	BD Rate	BD PSNR
SD	bank	-46.75 %	0.891 dB	-27.04 %	0.654 dB
	campus	-48.28 %	1.003 dB	-30.94 %	0.650 dB
	classover	-30.60 %	0.643 dB	-13.61 %	0.322 dB
	crossroad	-32.37 %	1.019 dB	-24.22 %	0.645 dB
	overbridge	-52.85 %	1.655 dB	-36.19 %	1.018 dB
	average	-42.17 %	1.042 dB	-26.40 %	0.658 dB
HD	carroad	-75.78 %	2.370 dB	-68.40 %	1.588 dB
	crossroad	-24.80 %	0.865 dB	-5.20 %	0.267 dB
	average	-50.29 %	1.618 dB	-36.80 %	0.928 dB

TABLE II. BITRATE REDUCTION OVER TRADITIONAL ROI EXTRACTION

sequence	ROI Coding Bitrate Reduction				
	QP=22	QP=27	QP=32	QP=37	Average
bank-sd	-24.17 %	-16.64 %	-16.83 %	1.92 %	-13.93 %
campus-sd	-27.42 %	-22.93 %	-23.88 %	0.61 %	-18.41 %
classover-sd	-19.82 %	-14.55 %	-13.37 %	-0.83 %	-12.14 %
crossroad-sd	-23.84 %	-17.04 %	-14.96 %	-8.44 %	-16.07 %
overbridge-sd	-22.47 %	-17.90 %	-18.93 %	-3.80 %	-15.78 %
carroad-hd	-31.45 %	-23.37 %	-4.50 %	-6.28 %	-16.40 %
crossroad-hd	-18.69 %	-11.41 %	-7.26 %	-10.23 %	-11.90 %

IV. CONCLUSION

In this paper, we propose a surveillance video coding method with HEVC quadtree partition based ROI extraction. In our method, ROI-layer and background-layer videos are produced with the help of background modeling and ROI extraction, and then encoded into ROI stream and background stream respectively. At the decoder side, we can reconstruct scalable videos with ROIs combined with static background and ROIs with realistic background pixels. Results show that our method can achieve remarkable total bit-rate saving and significant bit-rate cost reduction on ROIs. For future work, we will focus on the accuracy of ROI extraction and background modeling.

ACKNOWLEDGEMENT

This work is partially supported by grants from the Chinese National Natural Science Foundation under contract No. 61035001, 61121002 and 61176139.

REFERENCES

- [1] G. J. Sullivan, W. Han, "Overview of the High Efficiency Video Coding (HEVC) Standard," IEEE Trans. Circuits Syst. Video Technology, vol. 22, no. 5, pp. 1649-1668, Dec. 2012.
- [2] X. Zhang and et al., "An efficient coding scheme for surveillance videos captured by stationary cameras," in VCIP, July, 2010
- [3] L. Yang, L. Zhang, S. Ma and D. Zhao, "A ROI quality adjustable rate control scheme for low bitrate video coding," in Proc.PCS, pp.1-4, 2009
- [4] M. Wang and et al., "Region-of-interest based dynamical parameter allocation for H.264/AVC encoder," in Proc.PCS, pp.1-4, 2009
- [5] Y. Liu, Z. G. Li, and Y. C. Soh, "Region-of-interest based resource allocation for conversational video communication of H.264/AVC," IEEE Trans. Circuits Syst. Video Technol., pp. 134-139, 2008
- [6] X. Zhang, T. Huang, Y. Tian and et al., "Low-complexity and high-efficiency background modeling for surveillance video coding," in VCIP, Nov., 2012
- [7] Stauffer, Chris and et al., "Adaptive background mixture models for real-time tracking." Computer Vision and Pattern Recognition, IEEE Computer Society Conference on Vol. 2. IEEE, 1999.
- [8] JCT-VC, "HM Common Test Conditions and Software Reference Configurations," JCTVC-L1100, Jan. 2013