A REGION OF INTEREST BASED SURVEILLANCE VIDEO CODING

[1] *Hung-Yi Chen (陳宏毅),* [2] *Jian-Jiun Ding* (*丁建均*), [3] *Chiou-Shann Fuh* (*傅楸善*)

[1] Graduate Institute of Communication Engineering, National Taiwan University, Taiwan
E-mail: r03942139@ntu.edu.com

[2] Graduate Institute of Communication Engineering, National Taiwan University, Taiwan
E-mail: djj@cc.ee.ntu.edu.tw

[3] Dept. of Computer Science and Information Engineering, National Taiwan University, Taiwan
E-mail: fuh@csie.ntu.edu.tw

ABSTRACT

H.264/AVC is the state-of-the-art in modern video compression standards. It has extremely high compression ratio to meet the requirements the video communication between portable devices. H.264/AVC is a block-based compression method. Given that it is a block-based compression method, which means that it perform the whole video frame with same coding strategy regardless of foregrounds and backgrounds. According to the Human Visual System (HVS) research, human vision is only able to focus on several areas in a frame, which is defined as Region of Interest (ROI). In most cases, region-of-interest which attracts human vision is foreground or moving objects. Therefore, this phenomenon gives us a chance to encode all macroblocks unequally. In video surveillance, we perform a ROI-detector to separate a video frame into ROIs and non-ROIs, and perform these two parts with different encoding processes, so that it could not only enhance the compression ratio but also sustain a great visual quality video for human beings. Otherwise, with the aid of proposed skip mode, the coding process could have time-efficiency.

Keywords *Motion Segmentation, Region of Interest extraction, ROI-based Signal Processing, Human Visual System (HVS), H.264/AVC*

I. INTRODUCTION

In recent years, a growth of wireless multimedia sensor applications in surveillance system is notable. In spite of great improvements in network and video coding technologies, we still need a time-efficient and coding efficient solution to fit our requirements in surveillance video coding.

In wireless multimedia sensor networks, the surveillance video is usually generated by a nearly-fixed camera, encoded in nearly real-time. Otherwise, the foregrounds or moving objects should be distinct enough so that some application such as object tracking or motion recognition could be easily done in the post-processing stage for the further analysis.

The most popular and effective video compression standard is H.264/AVC [1]. H.264/AVC is a block-based compression method which is regardless of the Human Visual System (HVS). In other words, H.264/AVC neglects the feature of foregrounds and backgrounds in the processing stage before the entropy encoder. Moreover, H.264/AVC also pays a lot efforts on encoding nearly unchanged backgrounds, which is not efficient in surveillance video coding, in that the time-efficiency is a key factor to judge the surveillance video encoder performance. The complete H.264/AVC includes I, P, B three frame types, and CAVLC, CABAC two entropy encoding method for choosing.

To lower the computation complexity, we adopt only I and P frame type in our implementation, and choose a CAVLC, which is a look-up table encoder to save computation time. Furthermore, we set the group-of-picture parameter to be 30, which means that only an I-frame is inserted in 30 input video frames. In our experiment, we would set the H.264/AVC control group have the same features we have previously mentioned, and our proposed ROI-based coding would act as an experimental group, and utilize the same rear end encoder which shares a same features with the H.264/AVC control group.

A similar work about the region-of-interest-based video coding method is [7], "An Region of Interest Based Video Compression for Indoor Surveillance" proposed by Ganfeng Qiang, Lijun Yue and Fengqi Yu. However, the method has too many constraints. First, it could only be performed under QP=34 with the average PSNR is about 32.2dB. Secondly, it seems no acceleration on the encoding process, which is the most time-consuming step in video coding.

In this paper, we first utilize a ROI detection algorithm that we have already proposed [2] but with marginal revise to seek for a better computation efficiency, then we separate an input video into ROIs and non-ROIs.

After that, we also provide a further compression technique to deal with the ROIs and non-ROIs with respective strategies. When the inputs are non-ROIs, we would construct a compromised background model, and assign this background model as an encoder input to represent several frames of non-ROIs. To cope with the ROIs, the ROI macro blocks are encoded and transferred in a minor modified H.264/AVC control group with PCM type and P-skip type. For the entropy coder we adopt the H.264 entropy designed by Abdullah Al Muhit, which is the H.264 codec with only I-frame, P-frame and CAVLC entropy coding method.

The details of the proposed algorithm would include as follows. The motivation and contribution of this paper would be shown in Section II. The Region-of-Interest-extraction and the background model construction would be described in Section III. The Region-of-Interest and constructed background model coding would be shown in Section IV. The entire encoding process would be shown in Section V. Several simulations and the discussions are performed in Section VI. Last but not least, we will conclude in Section VII. The acknowledgement and references would be winded up in the last section.

II. MOTIVATION AND CONTRIBUTION

A. Motivation

Just as we have previously mentioned in Section I, we hope that we could propose a specialized algorithm for encoding surveillance video, with the several characteristics that the block-based method such as H.264/AVC is lacking of. These features are: time-efficiency both in encoding and decoding, easy for extracting the region-of-interest after decoding, providing an acceptable visual equality for viewers.

Secondly, we already have developed an effective and accurate ROI-detection method [2]. We could have a reasonable hypothesis that we could make good use of this previously proposed algorithm to separate foregrounds and backgrounds. However, we still need a minor revise to fit the rear end application (ROIs and non-ROIs compression).

B. Contribution

In this paper, we proposed an efficient video codec (encoder and decoder) for the surveillance video. We utilize the Region-Of-Interest-based system structure that we first use an efficient and accurate moving object segmentation algorithm with minor revise as our ROI-detector.

Secondly, with the aid of proposed label map, a down-sampling matrix to detect whether the block in the input video is a foreground block or a background block. By using this auxiliary information, we could skip many computation works to save up our time and memory usage. For instance, encoding the whole rate-distortion curve of video sequence “hall.yuv” with six data cursors, H.264 takes about 950.09seconds on the Matlab R2014a/ Intel(R) Core(TM) i5-2400CPU@3.10GHz, whereas our method takes only 82.44seconds. It is a 11.6 times faster encoding process when compared with the H.264 standard.

Thirdly, we add the flexibility for the user to adjust their visual equality for the backgrounds and foregrounds. For instance, if the user accept to ignore the backgrounds but focus more on foregrounds, our work could afford them with this flexibility to retain the foreground more with the lower quantization parameter.

Last but not least, we reinforce what [7] is lacking of:

(1) We provide a ROI-based video coding process that the RD-curve performance could behave better than H.264 standard on very low bitrate (PSNR=30) to mid-level (PSNR=40). Compared with [7] only outperform H.264 on single data point (PSNR=32)

(2) Our method is an acceleration of video compression process: the smaller the foreground proportion in a frame, the lower computation time in our coding process. Take “hall.yuv” as an example, our method is roughly 11.5 times faster than the H.264 standard.

(3) The flexibility to provide user a difference viewing, such as encoding foregrounds with lower quantization parameter, so that the foregrounds is more clear for the human vision.

III. REGION-OF-INTEREST EXTRACTION

The proposed algorithm is shown below: ROI Extraction and Background Model, brief introduction and revise to “Real Time Sensing and Shadow Robustness Video Foreground Segmentation Algorithm” [2],

A. Background Model Construction

We perform the algorithm in [2] to separate the foregrounds (ROIs) and backgrounds (non-ROIs) for luminance variation analysis, to show that the ROI-detection and background/foreground separation is necessary in our algorithm. Figure1 is the one of the testing surveillance video “NTUMD5F.yuv” and Fig.2 is the experiment result of testing surveillance video “NTUMD5F.yuv” after the foreground/background separation. X-axis is the frame number whereas Y-axis is the mean luminance of each frame, and the unit is grey level value.

(a) Input 12th frame (b) Input 75th frame

Fig.1 The testing surveillance video "NTUMD5F.yuv"

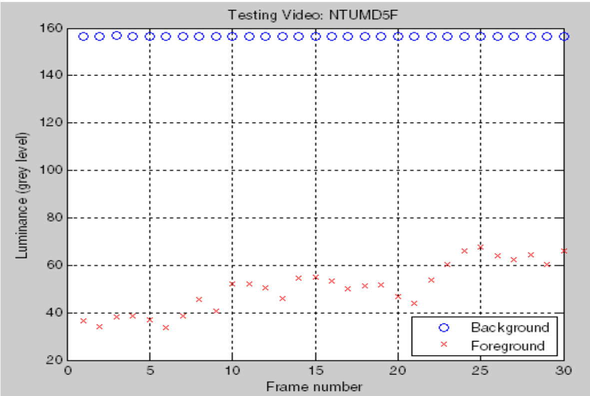


Fig. 2 The experiment result of Testing Video "NTUMD5F.yuv"
With frame number on the X-axis and
Y-axis refers to the mean frame luminance.

(a) Input 22th frame (b) Input 75th frame

Fig.3 The testing surveillance video "hall.yuv"

According to Fig.2, the result shows that: the luminance of foreground pixel fluctuate drastically even the moving object is in the same color. However, the background luminance is almost unchanged in a period of time. Therefore, it leads us a chance to build a background model to represent a sequence of nearly unchanged backgrounds (or called non-ROIs), which could not only lower the quantity of encoding data but also fasten the encoding process. The conclusion of Human Visual System also supports us to do so, because the human attention or vision is focus on the ROIs rather than the slight luminance variation in non-ROIs. The results of background model construction is shown in Fig.3 (a).

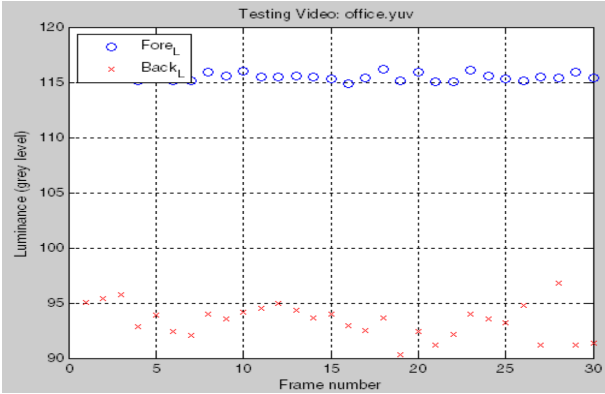
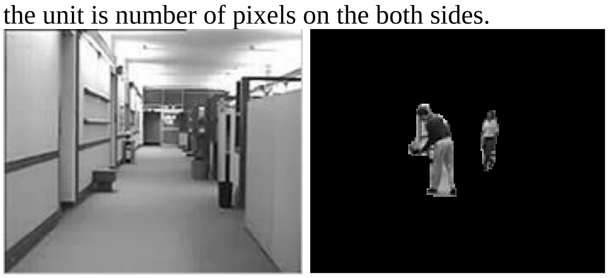


Fig. 4 The experiment result of Testing Video "hall.yuv"
With frame number on the X-axis and
Y-axis refers to the mean frame luminance.

The other input surveillance video "hall.yuv" is shown in Fig. 3, and its foreground/background separation analysis is shown in Fig.4. Although the backgrounds of hall.yuv exist some fluctuations, it is also acceptable to build a background model to represent several frames of backgrounds. And the luminance fluctuations of foregrounds still show that we should still apply a traditional motion estimation and compensation to encode the moving objects.

To be supplementary, two input surveillance videos (hall.yuv, NTUMD5F.yuv) share the same frame rate 30 frames/second, and the frame size of NTUMD5F.yuv is 128x192, whereas the frame size of hall.yuv is 144x176, the unit is number of pixels on the both sides.



(a) (b)

(a) refers to the constructed background model.
(b) refers to the foreground object extraction.

Fig. 5 The Extraction of ROIs and Non-ROIs by [2], the ROI-extraction result of hall_qcif

Besides, for the foregrounds (or called ROIs) we found out that the ROIs only occupy parts of the entire frame, which means that most of the macro blocks in the ROI-frame is no need to apply the motion estimation and compensation. Therefore, the encoding processing could be reduced greatly with minor investment computation time in the ROI-detection process.

B. Region-Of-Interest Extraction

There are lots of existing and proposed Region-of-interest methods.

Dan Grois and Ofer Hadar [3] have described several directions to perform Region-of-interest detection and tracking, such as object detection, face detection, skin detection and compressed-domain detection, and each direction has several previously proposed method to realize the ROI-extraction and build the background to implement a region-of-interest based video encoding. Otherwise, Z. Zivkoic [4], X. Zhou [5], and Yuan Shen [6] have also proposed some novel methods on performing video region-of-interest. However, it truly exists some drawbacks in these methods and models they adopt.

We expect that the whole foreground/background separation process would be nearly real-time to save the entire coding processing time. Therefore, we would adopt the ROI-detection method with least computation time but the method should be able to accurately separate the moving objects and nearly-still background, which is beneficial to the subsequent encoding stage. Thus, we adopt "Real Time Sensing and Shadow Robustness Video Foreground Segmentation Algorithm" [2] as our ROI-extraction method.

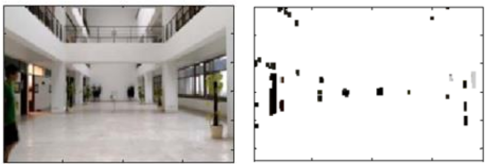
In this already proposed algorithm:

Firstly, we perform a Lab color transform for the simplicity of the subsequent ROI-extraction, otherwise performing ROI-extraction on Lab color transform is robust to the shadow effect.

Secondly, we adopt the self-defined frame difference to determine the definitely backgrounds and unknown backgrounds.

Thirdly, we perform an SLIC superpixel segmentation to separate the rest unknown background into certain foregrounds and backgrounds. The obtained backgrounds in this stage would be merged with the backgrounds we acquired in the previous stage.

Moreover, we will adopt the method in [2] with marginal revises, which is a previously proposed method by us. As the result shown in Figures 1 and 2, algorithm [4] could not extract the region-of-interest (ROI) only after the 15th frame and the shadow is misjudged as the foreground. Algorithm [5] needs more computation time and it is not robust to shadow effect. Lastly, algorithm [6] could not precisely extract the ROI-region, which is not suitable for the ROI-based video compression application.



(a)Input 12th frame (b) Algorithm [4]

(c) Algorithm [5] (d) Algorithm [6]

(e) Proposed Algorithm [2]

Fig. 6 The ROI Extraction Results of Several Known methods of 12th frame.

(a)Input 75th frame (b) Algorithm [10]

(c) Algorithm [12] (d) Algorithm [11]

(e) Proposed Algorithm [2]

Fig. 7 The ROI Extraction Results of Several Known methods of 75th frame.

In proposed ROI-based video coding, we will not only adopt the framework of [2] but also endow an improvement to fit the application. To lower the complexity of the ROI-extraction step, we will revise [2] by adopting a block-based computation to find out the background pixels, in that the foreground occupy only small part in a frame in the surveillance video coding process, by using the block-based (such as 2×2 or 3×3) for deciding the background is more efficient than the previous method. Moreover, we could build a background model by using the result of ROI-extraction by [2] and extract the moving object easily from the background. Therefore, we separate the input video sequence into ROIs and non-ROIs.

(Left) Constructed Background Model.

(Right) Foreground Object Extraction.
Fig. 8 The Extraction of ROIs and Non-ROIs by revised [2].

Table I The comparison of computation time in proposed and other known ROI-extraction methods.

Method	Computation Time	Development Environment
Proposed Method [2]	0.97 seconds 3.57 seconds	C/C++ Matlab
Zivkoic Method [10] 2004	3.79 seconds	C/C++
Augmented Lagrange [5] 2010	147.0 seconds	Matlab
Zhou Method [6] 2013	21 seconds	Matlab

Table I shows that the ROI-extraction method we adopt [2] would be very suitable to our application if we focus on the computation processing time feature. With the same testing environment, our ROI-extraction method is 16 times faster than other well-known methods.

IV. REGION-OF-INTEREST AND BACKGROUND CODING

At the section, we have already separated a video sequence into two parts: ROIs and Non-ROIs. That is to say, ROIs are the foreground pixels and the Non-ROIs are the background pixels. Subsequently, we would process these two parts with different tips to seek for a better performance.

A. Background Model Construction and Coding

Firstly we start with the Non-ROIs, given that the background normally change slightly, which means the pixel value change in background is very slow, so we could give them a lower frame rate to encode it. Furthermore, there are many cases that the background pixels didn’t change for a long time, such as many surveillance videos. In those cases, we could only use one image to represent the whole sequence’s background. This is the essential part where ROI-based video coding may be better than the traditional block-based solution. The results shown in Fig. 4 could strongly support the hypothesis we have previously mentioned: the background pixels seldom changes. Thus, we could follow the following steps to encode background sequence:

Let ***P(x,y,t)*** represent the pixel value on the coordinate ***(x,y)*** at the ***t***-th frame, and we pick every ***T*** frames as a set to generate the background model for these ***T*** frames.

1. Setting up the initial background model by:

$$BG(x,y) = \frac{1}{T}\sum_{i=1}^{T} P(x,y,t) \qquad (1)$$

2. Scanning through the whole video sequence since (L+1)-frame to the last frame

   If the PSNR(BG, next frame) < 37, we refer that the background has changed since that frame, then we move on the step 3

   Else, we skip the step 3 and start the step 4.

3. From the starting frame of the changed background frame, we start to generate another background model by this and the subsequent ***(T-1)*** frames.

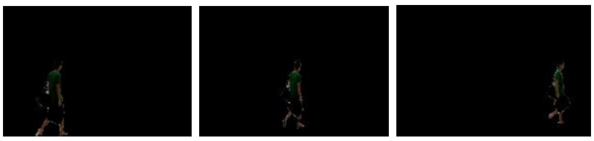
4. We encode the background model(s) and record the frame number which we should switch to another background model.

The reason why we set the encoding threshold to 37 dB is that this PSNR value means two image/video frame seems the same in normal human vision. Moreover, for the information of the background frames are similar, computing and encoding the difference (or called residue) is more efficient than encoding the original input frame.

(a) 40th frame. (b) 75th frame. (c) 125th frame.
Fig. 8 The background video sequence with extracted foreground pixels

B. Region-Of-Interest Coding

Secondly, let us turn to the ROIs. For the reason that the ROI only occupy part of the whole scenery, which means that we only have to encode the small part of information, which is shown in Fig. 5.



(a) 40th frame. (b) 75th frame. (c) 125th frame.
Fig. 9 The extracted foreground video sequence of “NTUMD5F.yuv”

Given that the H.264 standard encoder is a block-based method, we should transform our ROI-extraction elements into ROI-blocks, or we just call foreground blocks (FB), and the rules we follow to decide whether a block is:

Let ***P(x,y,t)*** represent the pixel value on the coordinate ***(x,y)*** at the ***t***-th frame, ***L(i,j,k)*** represent a label map to record whether the block***(i,j)*** on the ***k***-th frame is a FB (foreground block) or not. Suppose that the size of video frame is [***M,N,K***], the raw alpha map constructed in the ROI-extraction step ***A(x,y,t)***. Noted that the value of ***L(i,j,k)={0,1}***

1. Initialize an label map ***L*** to a zero matrix whose size is [***M/16,N/16,K***]. The value of ***L(i,j,k)*** would decide whether the initial video frame block ***P(16*i-15:16*i,16*j-15:16*j,k)*** is a foreground block or not. The threshold is set to 25 means that if the foreground label proportion is larger than 11.1%, we regard the block as a foreground block, which is a foreground block re-definition stage.

$$L(i,j,k)=1, if: \sum_{u=0}^{15}\sum_{v=0}^{15} A(i+u, j+v, k) > 25 \quad (2)$$

2. After getting the L(i,j,k), we could perform it as an foreground generation mask to extract the foreground block from every video frame by:

$$\prod_{u=0}^{15}\prod_{v=0}^{15} A'(i+u, j+v, k) = 1, if: L(i,j,k)=1 \quad (3)$$

$$Foreground(i,j,k) = \sum_{(i,j,k)\in\{M,N,K\}} P(i,j,k) * A'(i,j,k) \quad (4)$$

3. After obtaining the label map L(i,j,k), we follow the information of *L*(i,j,k) to encode the input foreground blocks. There are two types below: first is a coding process designed by Abdullah Al Muhit, which is the H.264 codec with only I-frame, P-frame and CAVLC entropy coding method. For simplicity, the encoder would be called Muhit_H.264 in short.

   The other is the proposed skip mode, which would skip the motion estimation and compensation process

4. Encoding the input block P(i:i+15,j:j+15,k) by Muhit_H.264, if L(i,j,k)=1.

5. Skip the encoding process and reconstruct the whole block by extracting the information from background model or previous block, if L(i,j,k)=1

The results above show that we only have to encode the few part of moving object in each frame. According to the Human Visual System (HVS) research, human vision usually focus more on one or some area(s) in a frame, and these areas are mostly consist of moving objects. Therefore, when encoding these area(s) we should try the best to keep the visual quality so that the reconstructed video is acceptable for human vision.

(a) 25th frame (b) 85th frame (c) 105th frame

Fig. 10 The reconstructed foreground blocks (FBs) of hall.yuv according to Label map ***L***.

(a) 25th frame (b) 65th frame (c) 85th frame

Fig. 11 The reconstructed foreground blocks (FBs) of NTUMD5F.yuv according to Label map ***L***.

Furthermore, we could observe that only the few regions is needed to be encoded, for the reason that most of the black region would be replaced by the constructed background model. So most of the region is no need to perform any computation such as motion vector estimation, compensation, mode decision and so on, which could not only reduced the data size but also save up many computation cost, and speed up the whole encoding process.

V. ENTIRE ENCODING PROCESS

As shown in Fig.12, the proposed encoding process is shown below.

To begin with, we adopt ROI-extraction process. Although we could integrate many types of ROI-extraction method, to seek for a better performance, we adopt [2] as our ROI-extraction method but with minor revise.

After separating foregrounds from the whole video frame, we implement a foreground block re-definition to get the block-based foreground/background blocks.

The foreground block re-definition result ***L*** would be provided as control reference information to decide whether the block encoding process should be skipped or not. The subsequent bit-stream output also depends on the foreground block re-definition result, which is also called label map in the previous section.

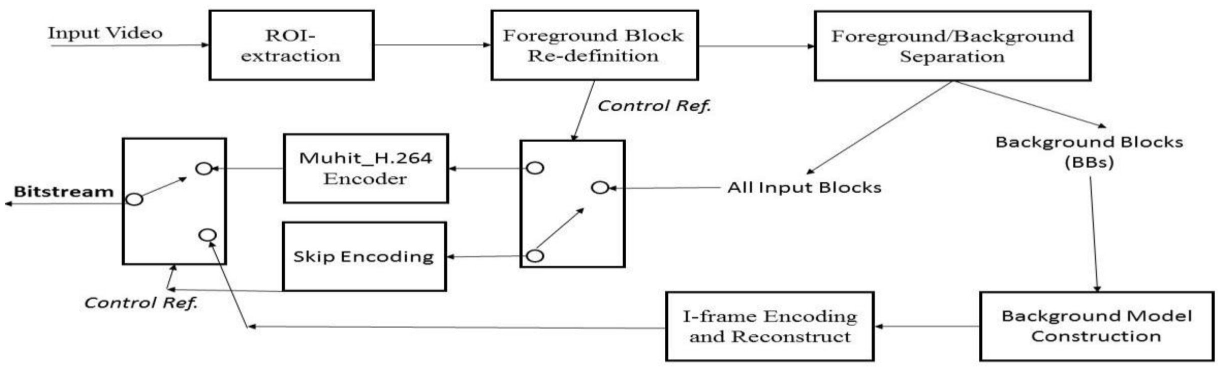


Fig. 12 The whole block diagram of proposed encoding process, with video sequence as input and bit-stream as output, every bit of output bit-stream∈{0,1}.

For the background blocks, we integrate all information and construct a background model, and use a normal I-frame coding process to encode the information. Then we trace back to the original input video to continue the encoding step.

Given that we have a down-sampled map to reference whether the block is a foreground block or not, we could use different strategy to encode. For the FBs, we just only follow the Muhit_H.264 to encode this type of input block. But for the non-FBs (BBs), we would skip the encoding process and take the information from background model or blocks of the previous frame as a reconstruction block.

To sum up, we proposed a nova video coding method with better compression ratio and time-efficient encoding process.

Besides, the proposed method also provides user a chance to generate different visual quality between foregrounds and backgrounds. In Fig. 13 we set background quantization parameter = 43, foreground quantization parameter = 29 so that the background is blurred but the user could still see the foreground man clearly. To be more precisely, foreground PSNR in Fig. 13 is 49.84, and background PSNR in Fig. 13 is 26.46dB.

Fig.13 Different visual qualities between foregrounds and backgrounds with background_QP=43 and foreground_ QP=29.

VI. SIMULATIONS AND THE DISCUSSIONS

A. Experimental Setup

To verify our method, we would follow the H.264 standards to design our entropy coder. In this experiment, we would adopt the H.264 entropy designed by Abdullah Al Muhit, Muhit_H.264, which is the H.264 codec with only I-frame, P-frame and CAVLC entropy coding method. Muhit_H.264 would act as a control group in this experiment whereas our proposed method would act as an experimental group. The whole experiment is implemented under Matlab R2014a/ Intel(R) Core(TM) i5-2400CPU@3.10GHz. We choose QPs of {39,36,34,28,24,15} to draw out the whole Rate-Distortion curve with kbps (number of thousand bits) on the x-axis and PSNR(dB) on the y-axis. The computation time unit would be second. The two input testing videos would be NTUMD5F.yuv and hall.yuv. NTUMD5F.yuv is a man walking horizontally across the whole video frame with several disturbance in the background. The hall.yuv is a testing video from Xiph.org Video Test Media [8], which is an office video that one man would walk far from the camera and another man would walk to the camera, which means that these two moving object would have a more complicated moving pattern, and the fluorescent lamp would keep shining in the office background.

B. Experimental Results

Fig. 14 and Fig. 15 are both the rate-distortion comparison result of two encoding method (proposed v.s H.264) where Fig.14 uses NTUMD5F.yuv as input sequence and Fig.15 uses hall.yuv as input sequence.

The result of two rate-distortion chart shows that the proposed method has better performance on coding surveillance videos. With the same rate (kbps), the difference between proposed and H.264 would be about +4.9dB. With the same PSNR, the proposed method only needs about 40% bit rate on average.

The reason why we could have such a great performance is that we use a long-term static background as background to represent the backgrounds in 20~30 video frames, but the constraint is that the proposed method has an upper bound in PSNR (which is about 44dB)

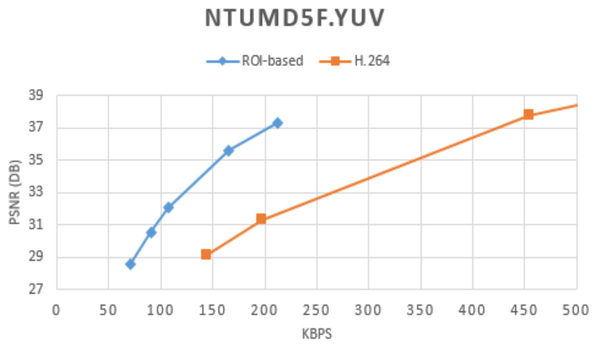


Fig. 14 The rate-distortion curve of proposed method (blue) and H.264 (orange), with the input data NTUMD5F.yuv.

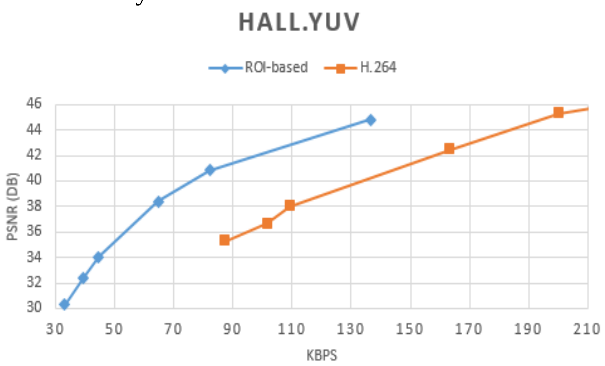


Fig. 15 The rate-distortion curve of proposed method (blue) and H.264 (orange), with the input data hall.yuv.

Table II The computation time of two methods on two video sequences with QPs of {39,36,34,28,24,15}.

Sequence	Proposed (sec)	H.264
NTUMD5F.yuv	150.1	1096.63
hall.yuv	80.44	934.09

Table II shows that our method is also computation time-efficient. The result shows that we have 7.31, 11.6 times faster than the H.264 standards when we adopting the same entropy coder, Muhit_H.264. For the reason that our method is no need to perform motion estimation and compensation of background blocks in every frame, we could skip many computation process proving that our method is really time-saving process.

VII. CONCLUSION

In this paper, we propose a surveillance video coding method with H.264 based entropy codec and a ROI-detector with several revises. In our method, the input video would be separated into foregrounds and backgrounds. The foreground blocks would just follow the H.264 standard coding scheme, whereas the backgrounds block would be used to construct a background model to represent for the whole video sequence. When the proposed encoder detect the parts where the skip mode is needed, it would skip the encoding process, including motion estimation and compensation and mode decision, and adopt the constructed background model as a reconstructed block. At the decoder, we could just combine the background model and the decoded foregrounds as a reconstructed video sequence. With the aid of the proposed coding scheme, we could upgrade the video compression dramatically but still retain affordable visual quality for viewers.

Besides, the great time-saving feature is also an important feature in our proposed method, which could solve the time-consuming encoding problem found in H.264/AVC standard. We also provide a chance for users to encode the foregrounds and backgrounds unevenly so that the user could see foregrounds more clearly and neglect the blurred backgrounds. For future works, we would focus on the accuracy of the background modeling and add the contour-based method to describe the foregrounds more accurately.

ACKNOWLEDGEMENT

The author than for the support of National Science Council of Taiwan under the contract 103-2221-E-002-121-MY3.

REFERENCES

[1] Iain E. Richardson, "The H.264 Advanced Video Compression Standard, 2nd Edition," 2010, WILEY, ISBN: 978-0-470-51692-8

[2] Jian-Jiun Ding, Yu-Chen Liu, Yao-Ren Chang and Hung-Yi Chen, "Real Time Sensing and Shadow Robustness Video Foreground Segmentation Algorithm," ICEAI 2016, Osaka

[3] Dan Grois and Ofer Hadar, "Recent Advances in Region-of-interest Video Coding," Ben-Gurion University of the Negev, Beer-Sheva, Israel

[4] Zivkoic, Z., "Improved adaptive Gaussian mixture model for background sub-traction," in International Conference on Pattern Recognition, vol. 2, 28-31, 2004

[5] Zhou, X., Yang, C., & Yu, W., "Moving object detection by detecting contiguous outliers in the low-rank representation," IEEE Trans. Pattern Anal. Mach. Intell., 35(3), 597-610, 2013

[6] Yuan Shen, Zaiwen Wen and Yin Zhang, "Augmented Lagrangian alternating direction method for matrix separation based on low-rank factorization"

[7] Ganfeng Qiang, Lijun Yue and Fengqi Yu, "An Region of Interest Based Video Compression for Indoor Surveillance." 157 – 160, ICITEC, 2014

[8] Xiph.org Video Test Media [derf's collection] https://media.xiph.org/video/derf/