OPTIMIZED REAL TIME H264 ENCODER FOR VIDEOSURVEILLANCE APPLICATIONS.

Laurent PRIMAUX, Jenny BENOIS-PINEAU
LaBRI CNRS UMR 5800/Université Bordeaux 1
{primaux, jenny.benois}@labri.fr

ABSTRACT

The paper describes an optimized real-time H.264 encoder for videosurveillance applications. The method is based on a fast block- and motion–based analysis of frames and extraction of a motion mask which is compatible with H.264 block sizes. The encoder performs at 3 fps at CIF resolution without increase of the bit-rate and with PSNR compared to reference H.264 encoder.

1. INTRODUCTION

Real time video transmission at a low bit-rate is a very active research area. Needs and constraints imposed by Internet remote monitoring or mobile video communications are a high compression rate (with an acceptable distortion), and real-time computing. The ITU-T Recommendation H.264 [1] is most adapted for low bit-rate communications with an acceptable visual quality. A lot of works [2,3] have been recently devoted to various optimization aspects of H.264/AVC such as rate-distortion optimization which is a core of the codec [2], or development of the possibility of random access to regions of interest [3]. The coding complexity of the H.264 recommendation is an obstacle to real-time processing on general-purpose computers. The knowledge of the area of interest in video frames can help to optimize the rate-distortion step of H.264 (in terms of operational costs). The highest encoding quality could only be required in the area of interest and the full rate-distortion process would be replaced by an adapted analysis.

This paper is focused on videosurveillance applications with a fixed camera. In this case, the object extraction tool can be easily developed based on motion detection with H.264 adapted frame decomposition. The paper is organized as follows: first a general description of the optimization system is proposed, then we present the motion detection method and the computation of motion masks. The articulation of analysis results with H.264 encoder is then described, and finally the results of the method are presented for videosurveillance applications in a low density environment.

2. GENERAL SYSTEM DESCRIPTION

One of the main differences of H.264 Recommendation from previous coding standards is in the very flexible modes for the encoding of 16x16 macroblocks. In fact, both in Intra and Inter mode, a macroblock can be encoded by various combinations of 4x4 and 8x8 blocks. In Inter mode, one macroblock can be divided into either one 16x16 block, or two 16x8 (8x16) blocks, or four 8x8 blocks (see Figure 1). Each 8x8 block in its turn can be divided into either one 8x8 block, or two 8x4 (4x8) blocks or four 4x4 blocks (see Figure 2). Together with “Skipped” mode this leads to a total of 20 prediction possibilities, which makes the motion compensation process extremely fine but rather costly. Thus, to choose the best rate-distortion prediction mode for a given macroblock the encoder has to try each of the 20 prediction mode combinations.

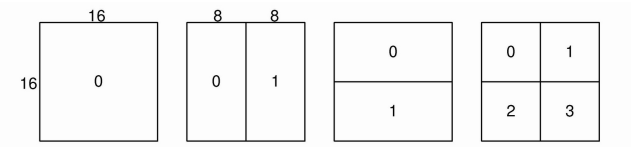


Figure 1 Macroblock segmentation modes

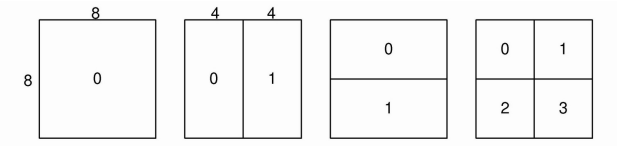


Figure 2 8x8 block segmentation modes

In our work, we used the Reference Codec Software (RCS) [4] from the JVT standardization committee. We noticed that the best prediction modes selected by this codec were strongly related to motion. Hence we set up a motion detection algorithm based on the maximum likelihood decision. Our method detects motion areas at three resolution levels (4x4, 8x8 and 16x16) allowing a direct correspondence with H.264’s blocks. Therefore, we directly obtain a motion mask expressed as one of the 20 combinations of prediction modes allowed by H.264. As shown in Figure 3, this modification is only located in the prediction part of the codec and does not affect other encoding parts (transform, quantizing and compression).

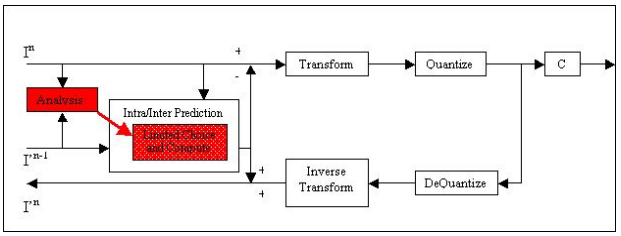


Figure 3 Localization of modifications (in dark grey) in a "closed-loop" encoder.

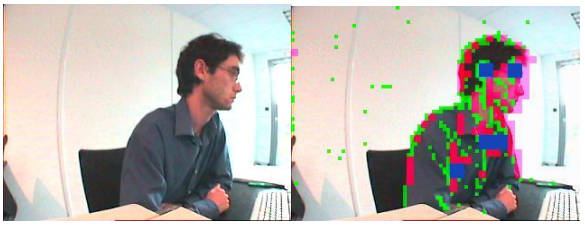
In the rest of this paper, we will see that this method enhances execution time and keeps image quality and compression ratio at the same level as the RCS in the optimal mode.

3. MULTIRESOLUTION AND MULTISCALE MOTION DETECTION

The motion detection method is based on the comparison of the current frame with the reference frame that can be either the previous frame grabbed from the capture device, or the encoded and decoded previous frame. Moving objects can be detected because they modify the luminance of traversed pixels. Hence, to detect luminance changes we consider the following hypothesis: let A_1, A_2 be image blocks centered on (x_0, y_0) both at t and t-dt; H0 (first hypothesis) the two blocks have the same grey-level Gaussian distribution (no temporal change), H1 (second hypothesis) distributions are different (motion between frames). The method consists of maximizing likelihood ratio and comparing it with a threshold. So, with the constant grey level model: $(A) \Leftrightarrow I(x,y) \approx N(m, s^2)$ we obtain for each point (x,y) in A_1 and A_2 $I(x,y) \approx N(m_0, s^2)$ with H0 hypothesis, and for each point (x,y) in A_1 $I(x,y) \approx N(m_1, s^2)$ and in A_2 $I(x,y) \approx N(m_2, s^2)$ with $m_1 \neq m_2$ in H1 hypothesis. Let $R = \ln \frac{L(H1)}{L(H0)} \underset{H0}{\overset{H1}{\gtrless}} l$ the likelihood ratio and the threshold l, we deduce the following decision criteria $\sqrt{R} = \frac{n}{2s} |\hat{m}_1 - \hat{m}_2| \underset{H0}{\overset{H1}{\gtrless}} l$' (see [5]). Then we introduce $FD(x,y,t+dt) = I(x,y,t+dt) - I(x,y,t)$, which gives the decision criteria $\frac{1}{2ns} \left| \sum_A FD(x,y,t+dt) \right| \underset{H0}{\overset{H1}{\gtrless}} l$', that we reinforce as $\frac{1}{2ns} \sum_A |FD(x,y,t+dt)| \underset{H0}{\overset{H1}{\gtrless}} l$", here s is the standard deviation of the noise.

This technique allows building two kinds of masks, a *global motion mask* and an *object mask*. The later one is computed in an "open loop" mode, when two successive frames from original grabbed video are used for motion detection. The detection of motion for object mask is fulfilled at the highest resolution level, normally with 4x4 blocks. The example of object mask for sequence "Laurent" at CIF and 3 fps resolutions is given in Figure 6. The *global motion mask* is computed in a "closed-loop" mode when the reference frame for motion detection (in the current original frame I^t) is the coded and decoded previous frame $\hat{I}^{t-1}$. On the contrary to the *object mask* the *global motion mask* allows detection of progressive light changes in the background as well. By using these two masks simultaneously we can distinguish modified blocks of object of interest from the background. Therefore we can focus the prediction quality on the object of interest, while setting the prediction mode of background blocks to 16x16 mode without the loss of neither quality nor compression ratio.

Once the *global motion mask* of the highest resolution is built (4x4 blocks), a multiresolution grouping is realized to obtain the *global motion masks* of a lower resolution (8x8 and 16x16). Lower resolution blocks are obtained from several higher resolution blocks by reporting the number of motion blocks in the higher level (one 8x8 is made of four 4x4 sub-blocks, one 16x16 block (macroblock) is made of four 8x8 blocks). Thus, when one block or macroblock has a value of 4, it is considered a moving one. Figure 4 shows the three multiresolution levels: 4x4 motion blocks are shown in light-grey, 8x8 ones in grey and 16x16 ones in dark-grey. This multiresolution strategy, in case of a still background, will allow a fine block-based representation on occluding borders and more rough inside a moving object.



(a)

(b)

Figure 4 Original frames (left) and corresponding multiresolution masks (right); (a)Sequence "Laurent", (b)Sequence "Michel"

No supplementary detection tests are needed to obtain block-based decomposition at a rougher resolution.

4. TOPOLOGICAL OBJECT MASK CONSTRUCTION

The *object mask* obtained by the motion detection is often not simply connected due to the constance of the signal in flat areas. In order to obtain the mask for the full object of interest we had to extend the initial *object mask*. The first step consists in strengthening the borders of object, and then the internal empty areas are filled. To do these tasks we perform a raster scan with a structuring element (Figure 5) on the initial detection mask. To fill the borders a block is marked if at least two of its neighbors are also marked. To fill the internal areas a block is marked if at least four of its neighbors are also marked (newly marked blocks are considered as originally marked blocks, thus ensuring the causal propagation of the object label).

N	N	N
N	CB	N
N	N	N

Figure 4 The 8 neighbors (N) of the current block (CB)

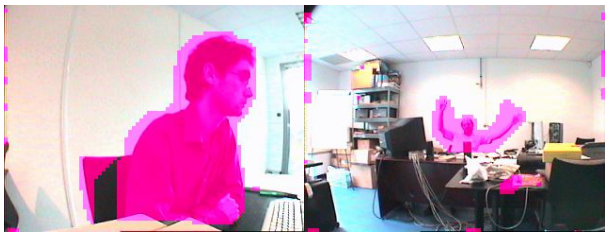


Figure 5 Object masks filled for sequence "Laurent" (left) and sequence "Michel" (right)

The method gives an overestimation of the object area (due to the causality effect). Bit-rate and quality will not be affected, but the computation time will be slightly increased. However, the time loss in this rare situation is lower than the time needed for an optimal filling of borders.

Figure 6 shows an *object mask* example, the effect of causality can be observed on the bottom right of the face. As it can be seen, this is a slight overestimation of the object area. The overdetection on the left is due to the uncovered background, so this area has to be updated making this detection a "good" one. This final object mask will be used to make fast distinction between objects and background in our optimized coding process.

5. OPTIMIZATION OF PREDICTION MODES DEFINITION IN H.264

In this part we present the way we drive the reference H.264 encoder and how the two presented masks allow an important reduction of the coding complexity. With the *global motion mask* we get three segmentation levels (4x4, 8x8 and 16x16) which allow an easy identification of 8x16 (or 16x8) and 4x8 (or 8x4) segmentation cases. To build the prediction error frame, for all macroblocks we can directly submit at most 2 prediction mode combinations to the encoder instead of the 20 combinations as in RCS. We will consider two sets of macroblocks; the first one consists of macroblocks which do not contain any blocks of any size from the object mask (we call them "background macroblocks"); the rest are "object macroblocks":

- For a background macroblock with no motion: "skipped" mode.
- For a background macroblock with motion: "16x16" mode.
- For an object macroblock with no motion: "16x16" mode.
- For an object macroblock with motion: *global motion mask*-deduced mode and "16x16" mode.

The *global motion mask*-deduced mode means: in case the the number of motion 8x8 blocks is less than 4 the mode proposed to H.264 encoder will be chosen from three possibilities (16x8, 8x16 and 8x8). In the later case, for each of four 8x8 blocks, the mode will be chosen in the same manner with 8x4, 4x8 and 4x4 blocks.

In "Inter-coded" frames, it is also possible to encode some macroblocks in Intra mode. In the reference coder there are four Intra prediction modes for 16x16 blocks and nine for each sub-block (4x4). In our case, when a 8x4 or 4x8 or 4x4 blocks are involved in *global motion mask*-deduced mode, we do not test Intra prediction modes in 16x16 because there are small textured areas in the macroblock.

6. RESULTS AND FURTHER PERSPECTIVES

In this section we present results on the 100 frames-length videosurveillance sequence "Michel" in a low density environment (only few moving objects) in CIF format at 3 fps and 4:2:0 chroma format. We tested 3 encoding modes: the optimal mode of RCS (referred as ***best***), the quick mode of the RCS (referred as ***16x16***) which involves only 16x16 blocks for motion compensation as in MPEG2 and our optimized codec (referred as ***SPC***). Figure 7 shows the time spent to encode each frame (in ms), Figure 8 shows the bit-rate associated to encoded frames (in bits), and Figure 9 displays the corresponding PSNR.

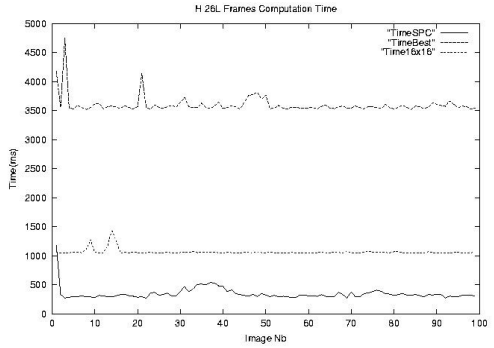


Figure 6 Computation time (ms per frame)

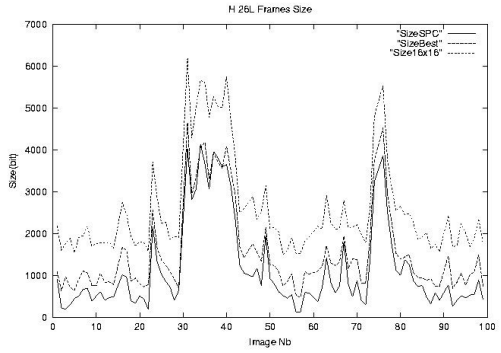


Figure 7 Frames bit-rate (bits)

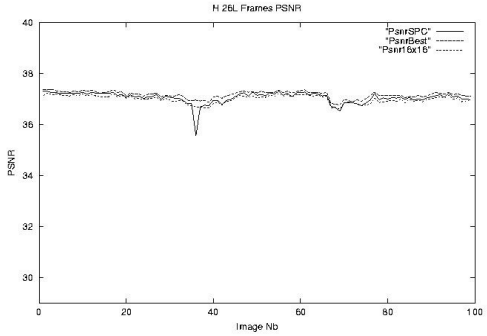


Figure 8 PSNR for each frame

The average per-frame encoding time obtained on Pentium 1.2GHz with our modified codec is around 300ms that is 3 fps, while RCS allows only 0.3 fps. The associated bit-rate is the lowest (see Figure 8), while keeping an identical PSNR to the RCS almost everywhere (see Figure 9). The average PSNR in H.264 RCS ***Best*** mode is of 37.2299 dB, in quick ***16x16*** mode it is of 37.0526 dB. Our ***SPC*** scheme supplies a 37.1431 dB average PSNR. This value doesn't take care of strong lightning change frames, because they are not representative of the global quality. So, decreases of PSNR can be observed in case of global lightning changes (frames 33-38).

Thus in this paper we proposed a real-time implementation of H.264 encoder on general purpose PCs due to the motion multiresolution detection and fast object extraction. This approach will also be used in case of a moving camera completed by camera motion estimation. Further works will also consist in optimizing "Intra-coded" frames processing by interpolation of prediction modes.

7. ACKNOWLEDGEMENTS

This work has been supported by the grant FEDER jointly with VisualPix LTD, we thank Mr. Michel STEMPIN (VisualPix) for his technical assistance during this project.

8. REFERENCES

[1] ITU-T Rec. H264/ISO/IEC11496-10 : "Advanced Video Coding", Final Committee Draft – March 2003.

[2] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini and G. Sullivan : "Rate-Constrained Coder Control and Comparison of Cideo Coding Standards", IEEE Transactions on Circuits and Systems for Video Technology, ISSN 1051-8215, Section 3: Video Coder Control, p 692.

[3] Miska M. Hannuksela, Ye-Kui Wang and M. Gabbouj : "Random Access using Isolated Regions", ICIP 14-17 September 2003, Barcelona, Spain.

[4] "H264/AVC Reference Codec Software", JM 6.0a, http://bs.hhi.de/~suehring/tml/

[5] P. Bouthemy, P. Lalande. "Recovery of moving object masks in an image sequence using local spatiotemporal contextual information**"**. Optical Engineering, 32(6):1205-1212, June 1993.