A Background Memory Update Scheme for H.263 Video Codec

Kui Zhang and Josef Kittler
Centre for Vision, Speech and Signal Processing,
School of EEIT&M,
University of Surrey,
Guildford GU2 5XH, United Kingdom
e-mail: K.Zhang@surrey.ac.uk, J.Kittler@surrey.ac.uk

ABSTRACT

In the majority of video conference applications, the camera position and its focus are fixed and the scene in the picture is stationary. When the subject moves, some occluded background reappears in the scene. Such uncovered background part usually has to be spatially encoded. In this paper, we propose a background memory update scheme for H.263 video codec which can use the previously transmitted background scene as the reference so that a higher coding efficiency can be achieved. The proposed method only uses available information in the macroblock level to update the background memory. Thus the extra computational load is small and the synchronisation of the background memory is easy to achieve. The experimental results show that the proposed scheme gives improved performance both in terms of objective and subjective criteria.

1 Introduction

In the majority of video conference and video-phone applications, the focus and the position of the camera are fixed. This implies that the background scene is stationary and updating of the background region can be kept to a minimum. However, when the subjects enter the scene or make large movements, many regions have to be encoded in the INTRA mode which normally requires a higher bit consumption than that of the INTER mode coding. Such a burst of bit increment will lead to either a lower image quality (because the rate control mechanism will increase the quantization step) or a lower frame rate. The reduced resolution update mode (Annex. Q) in H.263+ can partly alleviate the problem [3, 4]. A further analysis of the situation shows that the requirements for the INTRA mode come from two different sources: one is for the newly appearing objects and the other is for the uncovered background. However, as the latter may have already been transmitted in previous frames there is a chance to recover such uncovered background efficiently. Various background memory techniques were proposed to improve coding efficiency for uncovered background region[1, 6, 5, 2]. The key to the success of the scheme depends on how to accurately obtain the background memory. Both methods proposed in [1] and [6] require explicit foreground/background segmentation which is both time consuming and unreliable in real scene sequences. In [2], Hepper proposed to use change detection to obtain background information and used a counter to determine the background memory updating strategy. The change detection requires extra computational load and its quality will determine the quality of the background memory. To synchronise the background memory at both encoder and decoder, all three schemes require either to send extra information or to use exactly the same segmentation or change detection algorithm both in the encoder and decoder. To get around the problem of obtaining reliable background memory and the synchronisation of the background memory on the encoder and decoder sides, Wiegand et.al. proposed a long-term memory scheme in [5] which uses up to 50 previously decoded images to locate the best matching motion vectors. The scheme not only performs well to encode uncovered background regions but also gives high coding efficiency in sequences with repeated scene changes such as “News” sequence when sufficient memories are used. The reported results are very impressive but at the expense of much increased memory requirement and computational load.

Here, we propose a background memory update scheme similar to the idea in [2]. We maintain the ability to store new background quickly and to update it slowly. Our scheme aims to provide a good background memory with the minimum increase in computational load. The proposed scheme only uses the information available in a coded macroblock and therefore the extra overhead to obtain background is minimum and the synchronisation of the background memory is easy to achieve. The proposed background memory update scheme may be used in conjunction with reduced resolution update mode(H.263 Annex. Q) to improve the coding efficiency and image quality for certain types of applications.

2 Background Memory Update Scheme

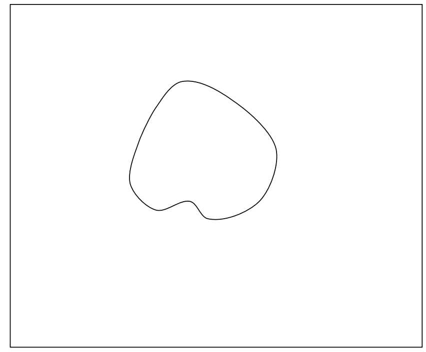
The advantage of using background memory as an alternative reference picture is that previously registered background can be used when a previously occluded background region reemerges as a consequence of moving foreground objects. Therefore, no INTRA coding will be necessary for such kind of reappearing regions. Figure 1 shows how background memory can help encoding occluded background regions. In Figure 1 (c), the reappearing region can be directly copied from background memory instead of using spatial coding so that a better coding efficiency can be achieved. If the object moves back and forth in front of a highly textured background, the background memory can save a considerable number of bits that would otherwise be needed to encode the repeatedly reappearing background regions.

2.1 Replenish and Update Background Memory

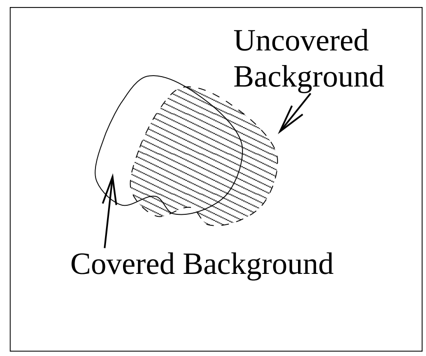
The successful use of background memory for efficient coding of an uncovered background region largely depends on the ability to reliably create and update the background memory. It is also important that the background memory at both the encoder and decoder sides of the codec are kept the same. To avoid sending background memory through the bitstream, it is normally obtained from the reconstructed image. Furthermore, with the progress of the frames, some previously covered background may appear in the scene and should be put into background memory. We also noted that in a fixed bit rate channel, the first INTRA frame is normally coarsely coded (using large quantization step) to avoid buffer overflow and to reduce the coding delay. Such INTRA frame is normally used as the initial background memory. Therefore, it may be of poor image quality. Here, we propose a background update scheme aiming at solving the above problem. The scheme only uses information obtainable from the bit stream so that the synchronisation of the background memory does not need any extra overhead. We also address the background memory quality improvement problem by using quantization step size to control the rate of updating the background memory. The whole scheme consists of three steps:

1. When a first INTRA frame or a scene change frame appears, the whole background memory is flashed with that frame. A quantization map matrix in which each element corresponds to an 8×8 area in the background memory is created and initialized to the quantization step used for INTRA frame coding.

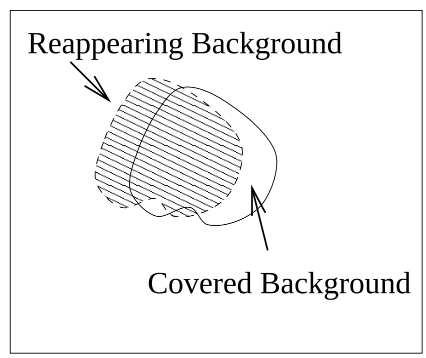
2. For successive frames, if the macroblock is not coded, the corresponding background memory will stay unchanged. Otherwise if the motion vector for an 8×8 region is zero and there are DCT coefficients to be transmitted, the quantization step used to encode the DCT coefficients(QP_C) is compared with that in the background quantization



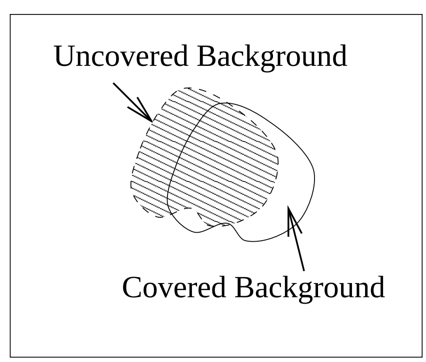
(a) Initial position of the object



(b) Object moving up-left



(c) Object moving down-right



(d) Object moving down-right further

Figure 1: Object motion and the background memory

map matrix(QP_{BG}). If $QP_C < QP_{BG}$, the background memory is updated using

$$BG = (\alpha * BG + Rec)/8 \quad (1)$$

where BG is the pixel value in the background memory, Rec is the pixel value of the reconstructed image, α is the background updating factor. The corresponding quantization map entry is also updated by

$$QP_{BG} = (\alpha * QP_{BG} + QP_C)/8 \quad (2)$$

If $QP_C \geq QP_{BG}$, the background memory will remain unchanged.

3. If an occluded region can not be found in the background memory (an INTRA coding is required for such a macroblock), the background memory is replenished by

$$BG = (BG + \beta * Rec)/8 \quad (3)$$

where β is the background replenishing factor. The quantization step for the background memory is also updated using

$$QP_{BG} = (QP_{BG} + \beta * QP_{Rec})/8 \quad (4)$$

where QP_{BG} and QP_{Rec} are quantization steps for background memory and reconstructed INTRA codec macroblock respectively.

In the above equations, the weighting factors α and β control the speed of updating and replenishing respectively. In our experiment, we choose

$$\alpha = \beta = 7$$

which gives an updating rate of 12.5% and a replenishing rate of 87.5%. Because of the use of quantization step information, the quality of the background memory can only be improved and any low quality updating is avoided.

2.2 Application of Background Memory in Uncovered Background Prediction

Once we have successfully created the background memory, we can use it to improve the coding efficiency on uncovered background regions as illustrated in Figure 1. The use of background memory as a reference picture for Uncovered Background Prediction (UBP) can be controlled by either internal or external means. As our aim is to provide an efficient way to encode uncovered background for sequences with stationary background, the UBP mode should be switched off for sequences with non-stationary background to reduce the bit overhead and computational load. The UBP mode may be used in complement to Global Motion Compensation (GMC) mode which can efficiently encode sequences with camera motion[7]. Therefore, one of possible internal control strategies may simply use a global motion detector to toggle between UBP and GMC mode. Alternatively, one can use external means to select a suitable option for the codec.

When the UBP mode is selected, the Sum of Absolute Difference (SAD) between background memory and the original input(SAD_{UBP}) is calculated and the result is compared with SAD of zero motion vector($SAD_{0,0}$). If SAD_{UBP} is less than $SAD_{0,0}$, it is then used in place of $SAD_{0,0}$, noted as SAD_{zero}. If SAD_{zero} is less than $SAD_{MC} + 100$ and SAD_{UBP} is less than $SAD_{0,0}$, the background memory will be used as the reference instead of the reconstructed previous frame.

The procedure is operated at the macroblock level so that the overhead to indicate the use of background memory is kept to 1 bit per macroblock.

2.3 Bitstream Syntax Change

The use of the UBP mode requires some modification in the bit stream syntax of H.263 Version 2. Both picture head and macroblock level need to be modified to accommodate the use of background memory.

In the picture head, the Bit16 of optional PLUSPTYPE (OPPTYPE) can be used to indicate the use of the Uncovered Background Prediction (UBP) mode in which a 1 will enable the mode and a 0 will disable the mode. The use of background memory update may be negotiated through external means.

In the macroblock level, a one bit Background Memory Prediction (BMP) field is added after the Coded macroblock indication (COD) bit when UBP is 1 in OPPTYPE. A "0" in BMP indicates that the normal reference is used for the macroblock being coded; a "1" in BMP indicates that the background memory is used as the reference picture. Please note that BMP bit should be checked even if COD=1 (macroblock not coded) so that the right reference picture can be selected for the macroblock. The extra overhead for using UBP mode is 1 bit per macroblock which may significantly increase the bit rate for very low bit rate applications with high frame rate(e.g., it will result in a 16.5% extra overhead for CIF sequences coded at 24kbps at 10 fps).

3 Experimental Results

The experiments were carried out using an implementation of H.263+ coder with modifications to use the background memory update mode. Two CIF sequences with some degree of occlusion are used in the test: "Children" and "Silent voice". The rate-distortion results of the experiments are given in Figure 2 and Figure 3 respectively. In the figures, "Codec A" is the codec with uncovered background prediction enabled and "Codec B" is the same codec with uncovered background prediction disabled. Both sets of results show that the use of uncovered background prediction improves both the image quality (higher PSNR) and coding efficiency (lower bitrate). The results also show that the use of the proposed scheme tends to improve the image quality at the lower end of bit rate and it gives a better coding efficiency at the higher end of bit rate. The maximum gain is achieved in the middle range of the bit rate. The results proved our analysis in the previous section that the one bit per macroblock extra overhead hinders any improvements in coding efficiency in low bit rate applications. When the bit rate is sufficiently high, the gain from the use of UBP outpaces the overhead. This results in a more significant improvement in coding efficiency.

4 Conclusions

The proposed background memory update scheme is effective for sequences with stationary background and large local motions involving uncovered background regions. The bit overhead and computational load of the proposed scheme are relatively small but it requires an extra memory equivalent to the size of the image frame. The experimental results show that the use of uncovered background prediction with the proposed background memory update scheme can improve both the image quality for uncovered background regions as well as coding efficiency.

Acknowledgements

This work was supported by the European Union ACTS Project Scalar and by the Engineering and Physical Sciences Research Council Grant GR/K42776.

References

[1] F. Dufaux and F.Moscheni. Background mosaicking for low bit rate video coding. In *ICIP-96, Lausanne, Switzerland*, volume I, pages 673–676, 16-19 September, 1996.

[2] D. Hepper. Efficient analysis and application of uncovered background prediction in a low bit rate image coder. *IEEE Transaction on Communications*, 38(9):1578–1584, 1990.

[3] ITU-T. *Draft ITU-T Recommendation H.263: Video Coding for Low Bitrate Communication.* International Telecommunication Union, 26 September, 1997.

[4] Akira Nakagawa. *Comparison of Annex Q and MRR.* ITU-T SG16 Q.15 Meeting, Portland, USA, July, 1997.

[5] T.Wiegand, X.Zhang, and B.Girod. Motion-compensating long-term memory prediction. In *ICIP-97, Santa Barbara, CA, USA*, volume II, pages 53–56, 26-29 October, 1997.

[6] Y.A.Jeong, S.H.Han, and J.S.Choi. Very low bit video coding algorithm using uncovered region memory. *IEEE TRANSACTIONS ON CONSUMER ELECTRONICS*, 43(3):286–294, 1997.

[7] K. Zhang and J. Kittler. Global motion estimation and robust regression for video coding. In *ICASSP-98,Seattle, USA*, 12-15 May, 1998.

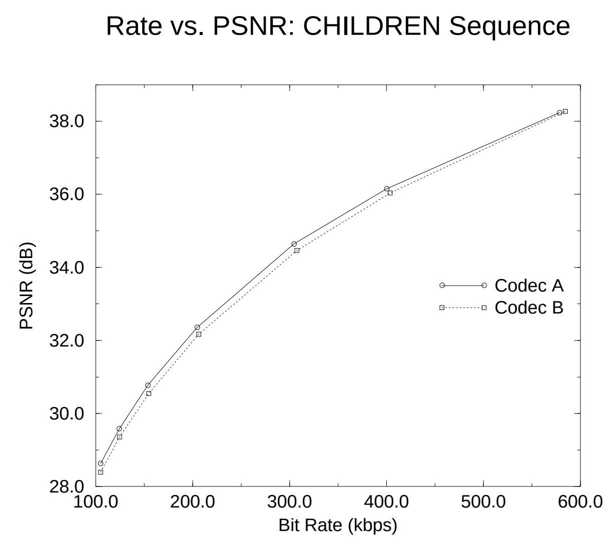


Figure 2: Test results for the sequence *Children*

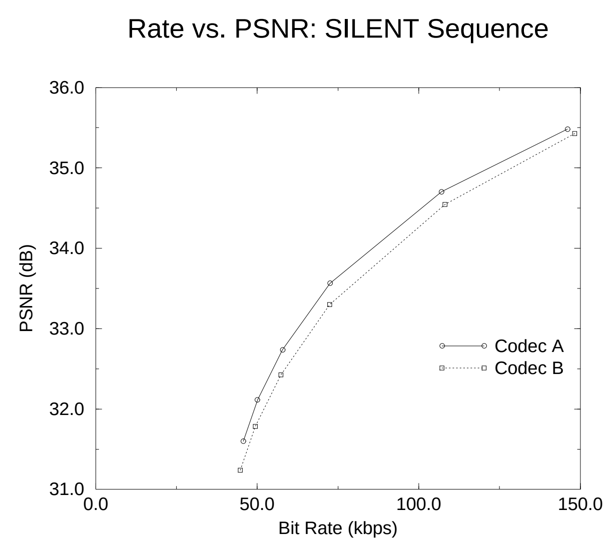


Figure 3: Test results for the sequence *Silent*