COMPOSITE-BLOCK MODEL AND JOINT-PREDICTION ALGORITHM FOR INTER-FRAME VIDEO CODING

Rong Ding, Fan Wang, Qionghai Dai, Wenli Xu and Dongdong Zhu
Department of Automation, Tsinghua University, Beijing 100084, P. R. China

ABSTRACT

Block-matching algorithm (BMA) is widely adopted in video coding. However, many blocks contain both background pixels and moving-object pixels, which often differ significantly in true physical motions. Using BMA for these blocks may lead to large prediction error and thus decreased coding efficiency. More accurate prediction can be achieved by object-based coding, which segments a video frame into different objects and codes these objects separately. However, additional bits are required for encoding the shape of the objects, and the computation complexity of object segmentation is often very high. In this paper, a composite-block model and a joint-prediction algorithm are proposed. This algorithm can get accurate prediction without shape coding, and its computation complexity is very close to that of the BMA. The experimental result shows that the new H.264 coder enhanced by our algorithm outperforms original H.264 coder by approximately 0.8~1.5 dB.

1. INTRODUCTION

In most video coding standards, a source frame is divided into rectangular blocks, and each block is predicted from the previously decoded frame with block-matching algorithm (BMA), in which all pixels in a block are assumed to undergo the same translation, and a motion vector (MV) is used to represent the displacement between the source block and the best matching block in the previously decoded frame [1]. If the elements of the prediction error block are small in intensity, high ratio compression could be achieved. In general, there are various objects in a video sequence. If a source block locates inside one of these objects, the assumption that all pixels in this block undergo the same translation is often acceptable. However, if the source block contains pixels from two different objects, the assumption might not be the truth. Typically, there are usually many blocks locating on the boundary between the background and a moving object. The background and the moving object often have very different true physical motions, so using BMA for these boundary blocks may lead to large prediction error.

For encoding such boundary pixels more efficiently, object-based video coding, in which the source sequence is segmented into different objects, for example, a background object and several moving objects, is a better choice. After the segmentation, a boundary block mentioned above is partitioned into two parts (one part belongs to the background object, and the other belongs to a moving object), and these two parts are predicted, transformed, quantized and binary coded respectively, therefore more precise prediction could be achieved and fewer bits are needed for the prediction error. However, a shape coder is needed in object-based video coding. There are two major classes of binary shape coders. One is bitmap-based coder [2] [3], which encodes for each pixel whether it belongs to the object or not; the other is contour-based coder [4] [5], which encodes the outline of the object. No matter which kind of shape coder is selected, additional bits for shape information are required for each object in addition to motion and texture.

In this paper, a novel video coding method based on composite-block model and joint-prediction algorithm is presented for providing natural and accurate prediction without requiring additional bits. Firstly, this method is described and analyzed in Section 2. Then the experimental results are presented in Section 3. Conclusions are drawn in Section 4.

2. COMPOSITE-BLOCK MODEL AND JOINT-PREDICTION ALGORITHM

In the composite-block model, each rectangular source block is partitioned into a background part (BP) and a moving-object part (MOP), and the two parts are predicted respectively to make accurate prediction. Let $S(k)$ represents a source frame, B is a block in $S(k)$, and $S(m,n,k)$ is a pixel in B; where m and n are the column and row indices respectively, and k is the frame number. The MOP of B is predicted from a traditional reference frame (TRF) $TRF(k)$ ($TRF(k)$ is actually the previously decoded frame, that is to say, $TRF(m,n,k)$ is the decoded value for $S(m,n,k-1)$); whereas the BP of B is predicted from a special reference frame, i.e. the background

reference frame (BRF) $BRF(k)$. $BRF(k)$ is also constructed from previously decoded frames, but since some background area appearing in $S(k)$ may have been overlapped by moving objects in $S(k-1)$ and thus in $TRF(k)$, more previous TRF are used to construct $BRF(k)$ as:

$$BRF(m,n,k)=\begin{cases} TRF(m,n,k), & \max\limits_{k-N+1\le k_1<k_2\le k}\left|TRF_Y(m,n,k_1)-TRF_Y(m,n,k_2)\right|\le T_1 \\ BRF(m,n,k-1), & otherwiese \end{cases} \quad (1)$$

where $TRF_Y(m,n,k)$ represents the intensity value of pixel $TRF(m,n,k)$ and T_1 is a threshold. Equation (1) shows that each BRF is set as a copy of the previous BRF at first, and then is refreshed by pixels which have changed little during N successive frames.

The motion of the BP is assumed to be zero in this paper. This assumption is usually acceptable in most video surveillance applications. For a video sequence whose background has more complex true physical motions, global motion compensation (GMC) can be the used, where the motion of the BP is specified by global motion vectors [6] [7] (GMV). Usually only a few additional bits are required for specifying the GMV. For example, in the advanced simple profile of the Mpeg4 visual standard, if the GMC is used, only four GMV are needed for each video object plane (VOP).

The motion of the MOP is characterized by a MOP-translational model, in which the MOP of B is assumed to be a translated version of the same part of the same moving object in $TRF(k)$, and a moving-object motion vector (MOMV), i.e. $MOMV_B$, is used to represent the displacement. Although the true physical motion of a moving object might be complex, the MOP is only a small part of the object and thus the MOP-translational model may be an acceptable approximation.

In order to choose a proper prediction method (be predicted from the $BRF(k)$ with zero motion vector, or be predicted from the $TRF(k)$ with the $MOMV_B$) for each pixel in B, a binary flag block (BFB) is constructed. $BFB(m,n,k)$ indicates whether $S(m,n,k)$ is predicted from the background (with $BFB(m,n,k)=0$) or from a moving-object (with $BFB(m,n,k)=1$). That is to say, the prediction value is decided by the BFB as:

$$P(m,n,k)=\begin{cases} TRF(m+\Delta m,n+\Delta n,k), & BFB(m,n,k)=1 \\ BRF(m,n,k), & BFB(m,n,k)=0 \end{cases} \quad (2)$$

where $P(m,n,k)$ is the predicted value for $S(m,n,k)$; and Δm and Δn are the column and row displacements of the $MOMV_B$, respectively. To avoid assigning additional bits for the BFB, a joint boundary and texture prediction is presented, in which the boundary separating the MOP from the BP is assumed to be a translation of some boundary in $TRF(k)$ with the same MV of the MOP, i.e. the $MOMV_B$. That is to say, the BFB is predicted from a binary flag frame (BFF) $BFF(k)$ as:

$$BFB(m,n,k)=BFF(m+\Delta m,n+\Delta n,k) \quad (3)$$

where $BFF(m+\Delta m,n+\Delta n,k)$ indicates whether $TRF(m+\Delta m,n+\Delta n,k)$ is a background pixel (with $BFF(m+\Delta m,n+\Delta n,k)=0$) or a moving-object pixel (with $BFF(m+\Delta m,n+\Delta n,k)=1$). Substituting Equation (3) into the right side of Equation (2), we obtain the final joint prediction equation as:

$$P(m,n,k)=\begin{cases} TRF(m+\Delta m,n+\Delta n,k), & BFF(m+\Delta m,n+\Delta n,k)=1 \\ BRF(m,n,k), & BFF(m+\Delta m,n+\Delta n,k)=0 \end{cases} \quad (4)$$

The $BFF(k)$ is constructed based on the differences between the intensity values of $TRF(k)$ and those of $BRF(k)$ as:

$$BFF(m,n,k)=\begin{cases} 0, & \left|TRF_Y(m,n,k)-BRF_Y(m,n,k)\right|\le T_2 \\ 1, & otherwise \end{cases} \quad (5)$$

where T_2 is a threshold.

In order to search for the proper $MOMV_B$ from a lot of candidate $(\Delta m,\Delta n)$, a joint motion estimation algorithm is proposed, where the best $(\Delta m,\Delta n)$ is the one that provides the minimum sum of absolute difference (SAD), i.e. $\sum_{S(m,n,k)\in B}\left|S_Y(m,n,k)-P_Y(m,n,k)\right|$.

After the joint prediction, the corresponding prediction error block for B is transformed, quantized and binary coded at the encoder as the same at the traditional video coding encoders. At the decoder, the decoded value for $S(m,n,k)$ is obtained from $P(m,n,k)$ and the prediction error. $P(m,n,k)$ is obtained based on $TRF(k)$, $BRF(k)$, $BFF(k)$ and $MOMV_B$; $BFF(k)$ is constructed from $TRF(k)$ and $BRF(k)$; $BRF(k)$ is constructed based on $BRF(k-1)$ and $TRF(k-N+1),\cdots TRF(k-1),TRF(k)$. Therefore, only $MOMV_B$ and the prediction error block are required to be coded and transmitted from the encoder to the decoder. Compared with traditional video coding algorithms using BMA, the algorithm described above needs no additional bits.

3. EXPERIMENTAL RESULTS

In the simulation, our algorithm is used to enhance the newest international video coding standard JVT/AVC/H.264 [8]. H.264 allows variable block size motion estimation, thus each inter-coded 16x16 pixel macro block may have seven candidate inter modes, i.e. 16x16, 16x8, 8x16, 8x8, 8x4, 4x8 and 4x4. The proposed algorithm is effective to all these modes. Therefore in the enhanced H.264, whichever the mode is 16x16 or 4x4, the corresponding sub-macro block is partitioned into a BP and a MOP.

The simulation is constructed over the main profile of the reference encoder JVT Model (JM9.0) of the H.264. All the notions of functions and files below are consistent with the notions in JM9.0. In the encoder, the codes for generating the BRF should be added firstly (for example, the function called "encode_one_macroblock" in the rdopt.c should be modified). Then the motion search should be modified by editing the function called "SetupFastFullPelSearch", the same as the motion compensation which needs to modify the function "OneComponentLumaPrediction4x4" and "OneComponentChromaPrediction4x4". In addition, a function used to generate the BFF is added to the program. As for the decoder, the codes for generating the BRF (modifying the function "decode_one_macroblock") and for motion compensation (modifying the function "decode_one_macroblock") are also applied, a function to generate the BFF is added as well.

In our enhanced H.264 algorithm, there are several parameters such as N and T_1 in Equation (1), T_2 in Equation (5), which may influent the performance of the encoder. These parameters cannot be set optionally. For example, if N is too small, improper BRF will be generated, while a large N will elongate the period of generating the BRF, which will also decrease the performance. It is similar in the case of T_1. A small T_1 makes it too hard to form the BRF, while a large T_1 easily generates a BRF which can not represent the true background. Fig.1 shows the increase of PSNR obtained by using the enhanced H.264 to replace the original H.264 under a constant bit rate (50kbps) and different combinations of N and T_1. A typical surveillance video sequence Dong1 (QCIF, 1400 frames, 25fps, a typical frame is shown in Fig.3) is used in the simulation. From Fig.1 we can see that the increase of PSNR reaches a maximum when $N = 8$ and $T_1 = 3$. Similarly, we can get different increase of PSNR when T_2 varies, and find that T_2 does not affect the performance much. We set it to 1 in the following simulation.

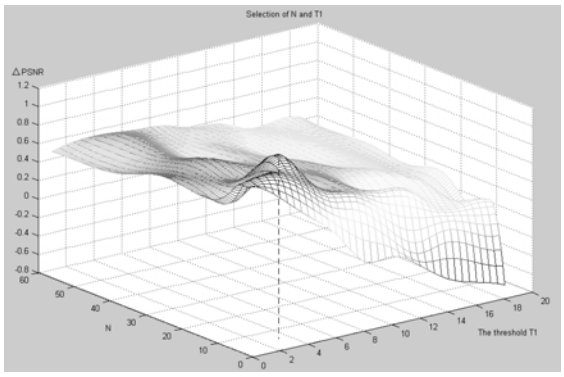

Fig. 1 Increase of PSNR obtained by using the proposed algorithm under different combinations of N and T1

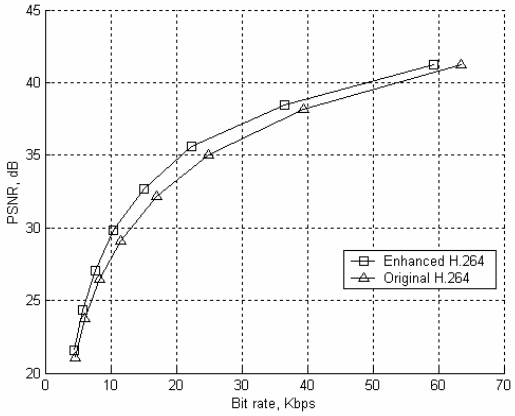

Fig. 2 Comparison of PSNR ~ Bit rate curve between original H.264 and enhanced H.264

By setting $N = 8$, $T_1 = 3$ and $T_2 = 1$, the PSNR ~ Bit rate curve of the enhanced H.264 is compared with that of the original H.264 in Fig. 2. The surveillance video sequence Dong1 is used again. It can be seen that for equivalent bit rate, approximately 0.8~1.5 dB improvement in PSNR is achieved. To illustrate our algorithm more explicitly, some typical frames in the encoding procedure are shown in Fig. 3, where "Original input frame" is Dong1, "binary bitmap" means the BFF, and "background frame" means the BRF.

The performance of the proposed algorithm is further examined by using four common test sequences, i.e. Carphone, Foreman, Salesman and Trevor. The PSNR and bit rate of the enhance H.264 and the original H.264 are compared in Table 1. The experiments are conducted for four quantization parameters, i.e. QP = 22, 26, 30 and 34. For simplicity, the parameters are also set to $N = 8$, $T_1 = 3$ and $T_2 = 1$. From Table 1 we can see that by applying the proposed algorithm, the enhanced H.264 coder can usually provide decreased bit rate and increased PSNR.

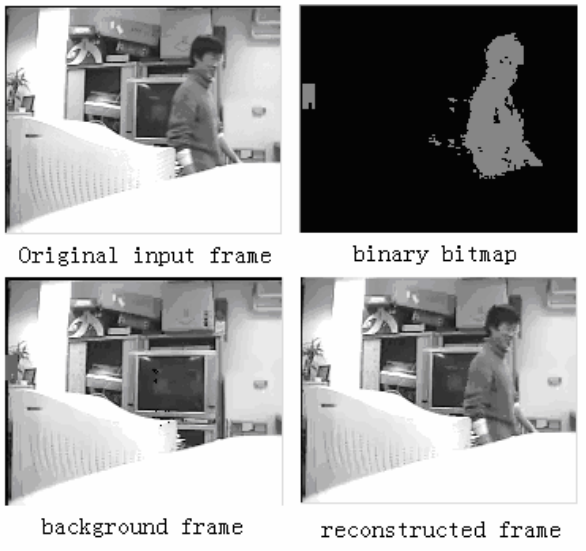

Fig. 3 Typical frames of Dong1 (the 33rd frame)

Table 1 Comparison of PSNR and Bit rate between enhanced H.264 and original H.264

Carphone, QCIF, 380 frames					
	QP	22	26	30	34
PSNR (dB)	Original	39.801	37.367	34.329	31.429
	Enhanced	40.657	37.549	34.503	31.695
	Δ PSNR	+2.15%	+0.5%	+0.5%	+0.8%
Bit rate (Kbps)	Original	393.84	242.95	136.07	72.424
	Enhanced	389.06	223.53	125.59	69.482
	Δ Bit rate	-1.2%	-7.9%	-7.7%	-4.1%
Foreman, QCIF, 400 frames					
	QP	22	26	30	34
PSNR (dB)	Original	39.529	36.401	33.585	31.031
	Enhanced	39.543	36.412	33.598	31.033
	Δ PSNR	+0.04%	+0.03%	+0.04%	+0.01%
Bit rate (Kbps)	Original	418.19	215.58	116.37	64.418
	Enhanced	419.44	212.86	111.86	64.192
	Δ Bit rate	+0.3%	-1.3%	-3.9%	-0.4%
Salesman, QCIF, 440 frames					
	QP	22	26	30	34
PSNR (dB)	Original	39.421	36.260	33.038	30.351
	Enhanced	40.087	36.780	33.680	30.868
	Δ PSNR	+1.7%	+1.4%	+1.9%	+1.7%
Bit rate (Kbps)	Original	115.97	63.544	34.218	21.150
	Enhanced	112.76	61.848	34.955	19.790
	Δ Bit rate	-2.8%	-2.7%	-2.15%	-6.4%
Trevor, QCIF, 150 frames					
	QP	22	26	30	34
PSNR (dB)	Original	40.934	37.582	34.852	32.487
	Enhanced	41.047	37.939	35.089	32.433
	Δ PSNR	+0.3%	+0.95%	+0.7%	-0.2%
Bit rate (Kbps)	Original	258.30	149.87	89.203	54.326
	Enhanced	257.23	149.81	88.163	51.789
	Δ Bit rate	-0.4%	-0.04%	-1.15%	-4.67%

4. CONCLUSIONS

In this paper, a composite-block model and a joint-prediction algorithm are proposed to improve the coding efficiency of the blocks locating on the boundary between the background and moving objects. By using the composite-block model in which each source block is partitioned into a BP and a MOP, more accurate prediction can be obtained. By using the joint-prediction algorithm, neither additional bits for shape information nor additional computation complexity for object segmentation is needed.

The motion of the BP is assumed to be zero in this paper, which is more suitable for coding surveillance video sequences. Our future work will focus on extend the algorithm to sequences whose background has more complex true physical motions by using globe motion vectors to specify the motion of the BP.

5. REFERENCES

[1] Yao Wang, Jörn Ostermann, and Ya-Qin Zhang, *Video Processing and Communications,* Prentice Hall, 2002.

[2] Brady Noel, and Bossen Frank, "Shape Compression of Moving Objects Using Context-based Arithmetic Encoding", *Signal Processing: Image Communications,* 15(7-8), pp. 601-617, 2000.

[3] Katsaggelos Aggelos K, Kondi Lisimachos P, Meier Fabian W, Ostermann Jorn, and Schuster Guido M, "MPEG-4 and Rate-distortion-based Shape-coding Techniques", *Proceedings of the IEEE,* 86(6), pp. 1126-1154, 1998.

[4] Saghri John A, and Freeman Herbert, "Source Analysis of the Precision of Generalized Chain Coders for the Representation of Planar Curves", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 3(5), pp. 533-539, 1981.

[5] Kim Jong I, Bovik Alan C and Evans Brian L, "Generalized Predictive Binary Shape Coding Using Polygon Approximation", *Signal Processing: Image Communication,* 15(7-8), pp. 643-663, 2000.

[6] Jozawa Hirohisa, Kamikura Kazuto, Sagata Atsushi, Kotera Hiroshi, and Watanabe Hiroshi, "Two-stage Motion Compensation Using Adaptive Global MC and Local Affine MC", *IEEE Transactions on Circuits and Systems for Video Technology,* 7(1), pp. 75-85, 1997.

[7] Rath Gagan B, and Makur Anamitra, "Iterative Least Squares and Compression Based Estimations for a Four-parameter Linear Global Motion Model and Global Motion Compensation", *IEEE Transactions on Circuits and Systems for Video Technology,* 9(7), pp. 1075-1099, 1999.

[8] Wiegand Thomas, Sullivan Gary J, Bjontegaard Gisle, and Luthra Ajay, "Overview of the H.264/AVC Video Coding Standard", *IEEE Transactions on Circuits and Systems for Video Technology,* 13(7), pp. 560-576, 2003.