A FAST INTER FRAME PREDICTION ALGORITHM FOR MULTI-VIEW VIDEO CODING

Xiaoming Li[1]*， Debin Zhao*[1]*, Xiangyang Ji*[2]*, Qiang Wang*[1]*, Wen Gao*[1]
{xmli, dbzhao, xyji, qwang, wgao}@jdl.ac.cn
[1]Department of Computer Science, Harbin Institute of Technology,   Harbin 150001, China
[2]Institute of Computer Technology, Chinese Academy of Science, Beijing 100080, China

ABSTRACT

The multi-view video coding improves the coding efficiency by utilizing motion-compensated prediction (MCP) and disparity-compensated prediction (DCP). However, the complexity of the inter frame prediction is very high, especially when the rate-distortion optimization is used. This paper presents a fast inter frame prediction algorithm to reduce the complexity. Firstly the prediction type is decided according to reference frames. Then some unuseful search regions in view direction are removed. Finally a fast inter mode decision strategy is proposed based on the relationship between MCP and DCP. Experimental results verify that the proposed algorithm can greatly increase the speed of prediction with negligible loss of coding efficiency.

Index Terms—multi-view video coding, fast inter frame prediction, motion estimation

1. INTRODUCTION

Recently, the video coding technology is developing very fast. H.264/AVC, developed by JVT, provides a very good coding efficiency [1].However, more video services, such as free-viewpoint video (FVV), free-viewpoint television (FTV), and 3DTV [2], are required. These new-born applications can be addressed by the key technology named multi-view video coding (MVC). To further improve the coding efficiency, MVC considers not only the correlation in the temporal direction for motion-compensated prediction (MCP), but also that in the view direction for disparity-compensated prediction (DCP). A lot of MVC algorithms have been proposed, whose common features are adaptively using MCP and DCP for predictive coding. Among all the schemes for MVC, the one proposed by HHI in [3] has achieved the best performance over the simulcast anchors. The basic coding scheme of HHI, as shown in Fig.1, uses the hierarchical B prediction structure for each view. The inter-view prediction is applied to every other view, and for the frames with inter-view prediction, the number of reference frames is up to 4. As described in [4], all views can be classified into two categories: main view (such as S0, S2, S4 and S6) which needs MCP only and auxiliary view (such as S1, S3, S5 and S7) which can reference the main view. The frames in the main views are predicted by MCP and those in the auxiliary views are predicted adaptively by MCP or DCP.

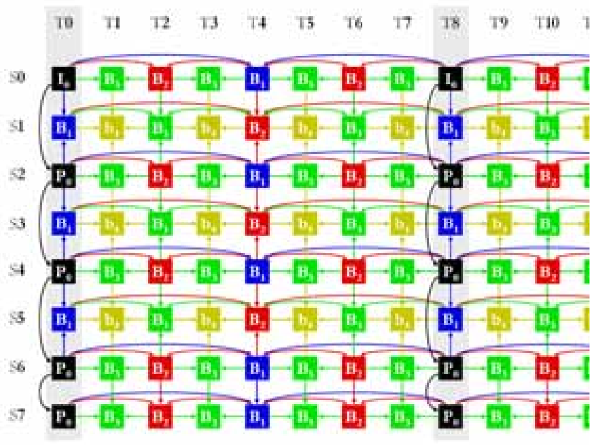

Figure 1. The HHI multi-view coding scheme

Some proposed fast algorithms for H.264/AVC can be used to speed up the prediction process in the main view direction such as [5-7]. The normal fast algorithms for hybrid video coding [5], by utilizing the local edge direction histogram from the edge map, some intra prediction modes can be skipped. In [6], the algorithm makes use of the spatial homogeneity and the temporal stationary characteristics of video objects to decide the best inter mode. The algorithm in [7], which is adopted in the MPEG-4 verification model (VM), is based on two observations. One is that a circular located at the zero motion position with a radius of two pixels will enclose most of the motion vectors. The second is that the block displacements of the real-world video sequences are mainly in the horizontal and vertical directions.

Because there are more reference frames in the MVC scheme, the auxiliary views are more complex than the main views. But there are few algorithms proposed to speed up the prediction process in the view direction. Considering the special characteristics of the MVC, we propose an algorithm to speed up the inter prediction process for frames in the auxiliary views. From statistics of some experiments, it is found that much of the computation for prediction is less effective. For example, only about 9.81% macroblocks in

1-4244-1437-7/07/$20.00 ©2007 IEEE

the auxiliary views of the *Ballroom* (640x480) sequence employ reference frames in view direction, but the computation time in that direction consumes up to 64.98%. For the *Exit* sequence only about 2.59% macrobocks in the auxiliary view employ the view direction. But the computation time in that direction is up to 65.04% of the total prediction process. Then, due to the correlation between MCP and DCP, some computation can be saved. For example, the mode employed by MCP can be utilized to guide the DCP to speed up the algorithm. In addition, because of the fixed relative positions of cameras, the coordinates of an object at the same time in different views also have correlation that can be used to reduce the computation load.

The rest of the paper is organized as follows. Section 2 describes the algorithm in detail. Section 3 presents the experimental results. Conclusions are given in Section 4.

2. THE FAST ALGORITHM FOR SPEEDING UP THE PREDICTION PROCESS

2.1. Fast Prediction Direction Decision for Auxiliary Views

As we know, HHI's scheme can get good performance because it chooses the best prediction result between the prediction in the temporal direction and that in the view direction. Inspired by [8], we present a method to judge which direction fits the object better. If the prediction in this direction is good enough, the computation load of the other direction can be saved.

It is well known that the motion vector in the temporal direction comes from the motion of the object. Better performance can be obtained in the temporal direction for homogeneous and stationary regions. The reason is that regions with fast motion may be predicted with small sub-block mode such as 4x4 and large motion vectors which decrease the coding efficiency. On the other hand, the disparity in the view direction which is determined by the relative positions of the object and cameras has little relation to the motion. So regions with fast motion prefer the prediction in the view direction as shown in Fig. 2, where (a) and (c) are original frames and (b) and (d) are the same frames as (a) and (c), but their blocks that employ the prediction in view direction are shown. It is denoted that most of the covered macroblocks belong to the regions whose motion are fast. From Fig.1 we can see the frame in the auxiliary views is in the middle of its two reference frames in the same view. The macroblock to be coded is more possible to have the same motion state as one of its collocated positions in two reference frames in the same view. And this macroblock is more possible to get better performance from the same direction as one of its collocated macroblocks. Statistics show that 84% macroblocks in the auxiliary views of the *Ballroom* sequence are prejudged correctly. For the *Exit* sequence, the correctly prejudged macrobocks in the auxiliary view are up to 86%. And in fact this assumption fits the background regions better.

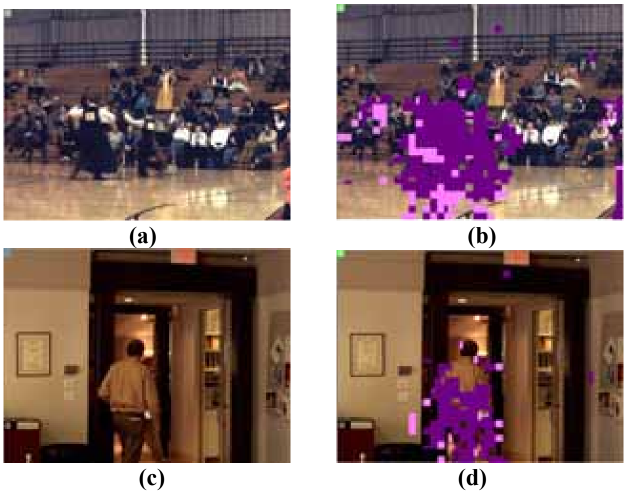

Figure 2. The distribution of blocks that employed the prediction in the view direction

Based on these observations, we can determine the prediction direction of the macroblock to be coded according to its collocated macroblocks in the temporal direction. If both of the collocated macroblocks employ the prediction in the view direction, the macroblock to be coded references frames in the view direction, otherwise the macroblock references frames in the temporal direction. If the performance is worse than a threshold, the second prediction stage in the other direction is performed for comparison to get the best performance.

Surely various methods can be used to define the threshold, and in this paper, when coding a macroblock, we define two thresholds. The one for view (or temporal) direction is defined as the average value of Rd-costs of the coded macroblocks in the coding frame which employed the prediction in view (or temporal) direction. The flow chart of the algorithm is shown in Fig.3.

2.2. Fast Motion Estimation in the View Direction

Certainly, the prediction process in the temporal and view direction can be sped up by fast algorithms for H.264/AVC video coding. We can further speed up the motion estimation in the view direction according to the location relationship between these views. The calculation process of the search range in the view direction is described as follows. In essence, an object's disparity between two frames at the same time in different views is generated by the displacement between two cameras. The position of pixel a and c are equal to |*ab*| and |*cd*| respectively, as shown in Fig.4, where *df* and *be* represent the screens of two cameras respectively, while *c* and *a* are the projection of object *O* on the two cameras. C_1 and C_2

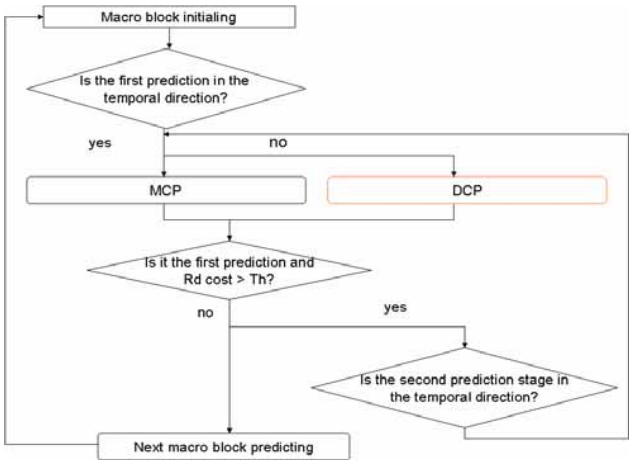
Macro block initialing
Is the first prediction in the temporal direction?
yes
no
MCP
DCP
Is it the first prediction and Rd cost > Th?
no
yes
Is the second prediction stage in the temporal direction?
Next macro block predicting

Figure 3. The flow chart of the fast prediction direction decision algorithm

present the positions of the two cameras. $|OH_1|$ is the distance between object O and the two camera screens. $|OH_2|$ is the distance between object O and the two cameras. Then the target for the motion estimate is to find out the position of pixel c which matches pixel a best, and the disparity value of object O is equal to $|cd|$ - $|ab|$ .According to Fig. 4, we have:

$$disparity\ (O) = |cd| - |ab| = |bd| - |ac| \qquad (1)$$

We can easily derive that

$$\frac{|ac|}{|C_1C_2|} = \frac{|Oa|}{|OC_2|} = \frac{|OH_1|}{|OH_2|} = \frac{|OH_2|-|H_1H_2|}{|OH_2|}. \qquad (2)$$

So the disparity has great relation to $|OH_2|$. Obviously, $|bd|$, $|C_1C_2|$ and $|H_1H_2|$ can be obtained from the parameter sets of the cameras, but unfortunately we can not get $|OH_2|$ only from $|ab|$. Because $|ca| < |C_1C_2|$ since OC_1 and OC_2 cross at O, and $|C_1C_2| = |db|$ since dC_1 parallels bC_2. We can say that $|ab|$ - $|cd|$ = $|ca|$ - $|db|$ < 0. Let z be a random position in the C_1 screen. If $|ab|$ - $|zd|$ > 0, then z is impossible to be the projection of O and such ranges that z belongs to can be skipped in the motion estimation process. Furthermore, this equation is always reliable though dC_1 may not parallels bC_2.

For the example in Fig.5, the car and the sun are at righter positions in Frame 3 and Frame 4 compared to those in Frame 1 and Frame 2 respectively. It is reasonable and always real because it is caused by the fixed set of cameras' position. So when we search d_1 or d_2 of one pixel in Frame 3 or Frame 4, only the left regions to its collocated position in Frame 1 or Frame 2 need to be searched.

2.3. Fast Mode Decision in the Second Prediction Stage

When the second prediction stage is needed, the best mode employed in the first prediction stage can be used to determine the candidate modes in the second prediction stage. It is because if the first prediction employed 16x16 mode, the macroblock to be encoded is a likely-harmony one. For example, if the best mode in the first prediction stage is 16x16, which means that the macroblock is relatively smooth, only 16x16, 16x8 and 8x16 are used as the candidate modes. Details of the candidate mode can be found in Table1.

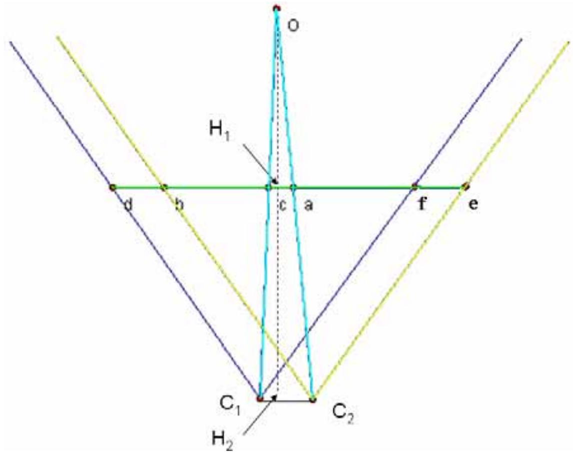

Figure 4. The generation of disparity

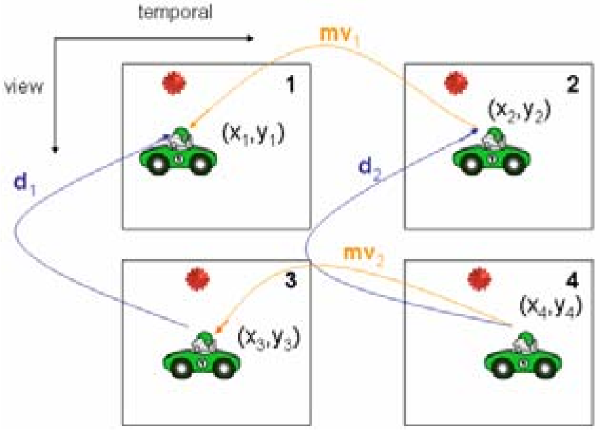

Figure 5. Relationship of mv_1, mv_2, d_1 and d_2

2.4. Fast Motion Estimation in the Second Prediction Stage

Besides the fast algorithm described above, the motion estimation process in the second prediction stage can be further sped up based on the motion correlation of different prediction types.

There are four corresponding pixels with respect to the same object in two views and two time points as shown in Fig. 5. Frame 1 and Frame 2 are in one view direction whereas Frame 3 and Frame 4 are in another view direction. Frame 1 and Frame 3 are at the same time whereas Frame 2 and Frame 4 are at the next temporal position. Intuitively, we can deduce one relation from the other three. The process to get by mv_1, mv_2 and d_1 is described below.

(x_3, y_3) can be obtained by

$$(x_3, y_3) = (x_4, y_4) + mv_2$$

Table 1. The candidate modes in the second prediction stage

The mode employed in the first prediction stage	Candidate modes in the second prediction stage
16x16	16x16 16x8 8x16
16x8	16x8 16x16
8x16	8x16 16x16
8x8 8x4 4x8 4x4	8x8 8x4 4x8 4x4 16x16 16x8 8x16

So

$$(x_1, y_1) = d_1 + (x_3, y_3).$$

As we know,

$$(x_2, y_2) = (x_1, y_1) - mv_1,$$

but we get neither (x_2, y_2) nor mv_1. There may be some pairs of pixels and motion vectors in Frame 2 that match (x_1, y_1) best in Frame 1. So (x_2, y_2) is much possible to be one of these pixels, and the search region for (x_2, y_2) is reduced.

And then

$$d_2 = (x_2, y_2) - (x_4, y_4).$$

Formally,

$$d_2 = f\,(mv_1, mv_2, d_1, (x_4, y_4)). \qquad (3)$$

Also,

$$mv_2 = g\,(mv_1, d_1, d_2, (x_4, y_4)). \qquad (4)$$

where functions f and g can be deduced as described above.

However, in existing MVC schemes, obviously not all pixels can be derived from (3) or (4). That is because if the prediction of (x_2, y_2) or (x_3, y_3) employed intra mode, the mv_1 or d_1 does not exist, and thus both (3) and (4) are invalidated. And also d_2 can be derived from d_1 and mv_2 can be derived from mv_1.

3. EXPERIMENTAL RESULTS

The proposed algorithm is implemented in JSVM 5_8. The employed sequences are *Ballroom* (640x480) and *Exit* (640x480). But limited by the time, we have not realized the method proposed in Subsection 2.4. Compared to the HHI's scheme, we list the average change of time, PSNR, and bits of frames in the auxiliary views in Table 2.

From the experimental results, we can see that *Exit* save more time than *Ballroom*. It is because we usually get high performance for stationary regions, where the second prediction stage is always skipped. And the prediction process of *Exit* with more stationary regions is sped up more.

4. CONCLUSION

This paper presents a fast algorithm for multi-view video coding based on HHI's scheme. From the experimental results we can see that considerable time is saved with only negligible loss of performance. It is because the predictions we do have the most possibility to get the best performance, whereas the predictions which seem to have little effect are abandoned. Furthermore, the four methods reducing the redundancy of computational load proposed in this paper can also be used jointly and separately.

Table 2. Experimental results

Sequence		△Time(%)	△Psnr(dB)	△Bits(%)
Ballroom	29	-65.931	0.023	3.316
	31	-64.121	-0.016	4.012
	32	-63.448	-0.078	3.276
	34	-62.516	-0.008	2.456
Exit	29	-69.148	-0.005	2.008
	31	-67.571	-0.017	2.822
	32	-67.624	-0.011	0.893
	34	-67.323	0.000	4.232

5. ACKNOWLEDGEMENT

This work has been supported by National Science Foundation of China (No.60672088), "985" construction project of Peking University, and the Research Fund for the Doctoral Program of Higher Education （No.20060213014）.

6. REFERENCES

[1] Information Technology—Coding of Audio-Visual Objects—Part 10: Advanced Video Coding, Final Draft International Standard, ISO/IEC FDIS 14 496-10, Dec.2003.

[2] "Report on 3DAV Exploration", ISO/IEC JTC/SC29/WG11, N5878, July 2003.

[3]"Description of Core Experiments in MVC", ISO/IEC JTC1/SC29/WG11, MPEG2006/W7798, January 2006.

[4]X. Guo,; Y, Lu.; F, Wu.; W, Gao, "Inter-View Direct Mode for Multiview Video Coding", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 16, No. 12, pp 1527-1532 , DECEMBER 2006.

[5]Feng Pan, Xiao Lin, Susanto Rahardja, Keng Pang Lim, Z. G. Li, Dajun Wu, and Si Wu, "Fast Mode Decision Algorithm for Intraprediction in H.264/AVC Video Coding", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 15, No. 7, pp 813-822, JULY 2005.

[6]D. Wu, F Pan, K. P. Lim, S. Wu, Z. G. Li, X. Lin, S. Rahardja, and C. C. Ko, "Fast Intermode Decision in H.264/AVC Video Coding", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 15, No. 6, pp 953-958, JULY 2005.

[7]S. Zhu and K.-K. Ma, "A new diamond search algorithm for fast block-matching motion estimation," in Proc. Int. Conf. Information, Communications, and Signal Processing, vol. 1, Sep. 1997, pp. 292–296.

[8] C. K, Cheung and L. M. Po, "Normalized partial distortion search algorithm for block motion estimation," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 10, pp 417-422, Apr. 2000.