ABSTRACT

The direct mode used in the bi-predictive pictures (B-pictures) can efficiently improve the coding performance of B pictures, because it has small overhead and obtains a predictive picture from two reference pictures. The traditional temporal direct mode (TDM) derives the motion vector of the current block by scaling the motion vector of the co-located block in the backward reference picture. However, when the current block and its co-located block in backward reference picture belong to different objects with different motion directions, the prediction efficiency of TDM is drastically reduced. In this paper, we propose an improved direct mode prediction method. In the method, a virtual reference picture is generated using the pixel projection technique. Then the virtual reference picture is used to predict the direct mode blocks in B pictures. The proposed method can enhance the prediction performance of the direct mode blocks and achieve a higher coding efficiency.

1. INTRODUCTION

Bi-directional prediction used for bi-predictive pictures (B-pictures) is a very efficient tool to improve coding efficiency. But a considerably higher percentage of bits are needed for encoding motion information. It causes the problem that the coding efficiency is decreased even if the prediction efficiency is improved. In the H.264, H.263 and MPEG-4 [1][2][3] video coding standards, the direct mode has been proposed to solve this problem. In the temporal direct mode (TDM), two motion vectors for bidirectional prediction are computed using the time continuity of motion. The direct mode does not need bits for the motion vectors, therefore, the overhead is quite small. And direct mode prediction generates a predictive image from the forward and the backward reference pictures, which leads to even further performance benefit.

In TDM of B-pictures, the forward and backward motion vectors are derived from the motion vector used in the co-located block of the backward reference picture [4], as shown in Fig.1. The forward motion vector $MV_F$ and backward motion vector $MV_B$ of direct mode block were originally calculated as follows:

$$MV_F = (tb/td) \times MV_C$$  \hspace{1cm} (1)
$$MV_B = ((tb - td)/td) \times MV_C$$  \hspace{1cm} (2)

but were later replaced with equations:

$$X = 16384 + \text{abs}(td/2)/td$$  \hspace{1cm} (3)
$$\text{ScaleFactor} = \text{clip}(-1024,1023,(tb \times X + 32) >> 6)$$  \hspace{1cm} (4)

$$MV_F = (\text{ScaleFactor} \times MV_C + 128) >> 8$$  \hspace{1cm} (5)
$$MV_B = MV_F - MV_C$$  \hspace{1cm} (6)

where $MV_C$ represents the motion vector of the co-located block in the backward reference picture, $td$ is the temporal distance between the backward reference frame $RL_b$ and forward reference picture $RL_0$, and $tb$ is the temporal distance between the current B frame and the forward reference frame $RL_0$. Variables $X$ and ScaleFactor are pre-computed at the slice or picture level to reduce the number of divisions required [5].

TDM can efficiently exploit the temporal redundancy between adjacent pictures. However, the current block is not always located in the motion trajectory of its co-located block, particularly when the current block and its co-located block in backward reference picture belong to different objects with different motion directions. In order to improve the motion vector accuracy of direct mode block, Xiangyang Ji [6] proposed a motion vector tracking scheme in B pictures. As shown in Fig. 2, block $M_{t+1}$ in the backward reference picture $P_{RL}^{t+1}$ has in its motion vector $MV_{t+1}^{MC}$ projecting to block $M_{t}$, which is the reference block of block $M_{t+1}$ in the forward reference picture $P_{RL}^{t}$. $M_t$ is the projection block of $M_{t+1}$ in the B picture. For every block $M_{t+1}$ in the backward reference picture, there is always a projection block $M_t$ in the B picture. The motion vector of the block $M_t$ which covers the largest part of the block $M_{t+1}$ in current B picture is selected to derive the forward and backward motion vector of block $M_C$.

Fig.1. Temporal direct mode used in H.264/AVC

Fig.2. Motion vector tracking (the basic block size in the picture is $4 \times 4$)
The scheme in [6] can bring more accurate motion vector for direct mode block \( M_C \) in B picture. But sometimes the block \( M_C \) is consisted of more than one moving object. When the different moving objects in the block \( M_C \) have different motion vectors, it is not accurate to derive the forward and backward motion vectors of block \( M \), only by the motion vector of block \( M_C \), which is projected from block \( M_{t+1} \).

In this paper, we propose a virtual reference picture, which is located in the same temporal position as current B picture. And we present a pixel projection algorithm to generate the virtual reference picture using the temporally subsequent and previous predictively-coded pictures (P pictures). The virtual reference picture is used to predict the direct mode block in B pictures to bring better prediction performance.

The rest of the paper is organized as follows: section 2 describes the formation of the virtual reference picture and its utilization to obtain direct mode blocks in B pictures. The simulated results are presented in section 3. Finally, section 4 concludes this paper.

2. VIRTUAL REFERENCE PICTURE FOR DIRECT MODE CODING

2.1. The location of the virtual reference picture

The H.264 standard has adopted multiple reference pictures to predict the values in B picture [7]. Fig. 3 illustrates the basic idea of multiple reference pictures, some previous and one future P (or I) picture in display order are used for B picture prediction. Excepting the existing reference pictures, a virtual reference picture is proposed by us. The virtual reference picture exists in a virtual formation and is only used to predict the direct mode blocks in the B picture. As shown in Fig. 3, the dotted line is the virtual reference picture \( P_v \). The temporal position of the virtual reference picture \( P_v \) is the same as that of the current B picture. Only the integer pixel values are stored in the virtual reference picture, so the size of the virtual reference picture is the same as that of the other reference pictures.

2.2. The formation of the virtual reference picture

The virtual reference picture is formed after the coding of the temporally subsequent P picture \( P_{t+1} \), and just before the coding of the current B picture. As shown in Fig.4, since the block \( M_{t+1} \) in the temporally subsequent P picture \( P_{t+1} \) has its motion vector \( M_{Vt+1} \) pointing to the temporally previous P picture, and the virtual reference picture \( P_v \) is located in the same temporal position as current B picture \( B_t \), there is an intersection block \( M_t \) between the virtual reference picture \( P_v \) and the projection block of \( M_{t+1} \). After finding the projection block \( M_t \) in the virtual reference picture, the forward and backward motion vector of block \( M_t \) can be obtained by scaling the motion vector \( M_{Vt+1} \). Then the pixel values in the block \( M_t \) can be obtained by bi-prediction. The size of the block unit used for pixel projection in the temporally subsequent P picture \( P_{t+1} \) is different among different macroblocks. It is the same size as that of the motion compensation block. In \( 16 \times 16 \) mode macroblocks, there is only one block used for pixel projection (such as block \( M_{t+1} \) in P picture \( P_{t+1} \)). In \( 8 \times 8 \) mode macroblocks, there are four blocks used for pixel projection (such as block \( \bar{M} \) in P picture \( P_{t+1} \), its size is \( 8 \times 8 \)). In intra mode macroblocks, the whole macroblock is not used for pixel projection.

![Image of multiple reference pictures](fig3.png)

**Fig.3.** Multiple reference pictures used in H.264

The pixel projection process, we generate the motion vector trajectory of projection from the up-left corner of each block \( M_{t+1} \) in the temporally subsequent P picture \( P_{t+1} \) as shown in Fig.4. The position and motion vector of the projection block \( M_t \) in the virtual reference picture \( P_v \) is calculated as follows:

\[
MV_t = ((td - th) / td) \times MV_{t+1}
\]

(7)

\[
MV_m = (MV_t, >> 2) \times 4
\]

(8)

\[
CP_t(x, y) = CP_{t-1}(x, y) + MV_m
\]

(9)

\[
MVB_m = ((tb - td) / td) \times MV_{t+1}
\]

(10)

\[
MVF_m = MV_{t+1} + MVB_m
\]

(11)

where \( td \) is the temporal distance between the subsequent P frame \( P_{t+1} \) and the previous P picture, \( tb \) is the temporal distance between the virtual reference picture \( P_v \) and the previous P picture, \( M_{t+1} \) is the motion vector of the block \( M_{t+1} \) pointing to one block in the previous P picture. \( MV_t \) is the scaled motion vector of block \( M_{t+1} \). \( M_{Vt} \) is intended to point to the projection block in the virtual reference picture \( P_v \). But the accuracy of \( MV_{t+1} \) and \( M_{Vt} \) is in units of one quarter of the distance between luma samples [8], and the virtual reference picture only has integer pixel value, so \( MV_t \) is modified to \( MV_m \) according to equation (8). Motion vector \( MV_m \) points to the integer pixel projection block \( M_t \) in the virtual reference picture. \( CP_{t-1}(x, y) \) denotes the up-left corner of the block \( M_{t+1} \). \( CP_t(x, y) \) is the up-left corner of the block \( M_t \). The accuracy of \( CP_{t-1}(x, y) \), \( CP_t(x, y) \) and \( MV_m \) is also in units of one quarter of the distance between luma samples. Block \( M_t \) is the same size as block \( M_{t+1} \). \( MVF_m \) and \( MVB_m \) denote the forward and backward motion vectors of \( M_t \), respectively.

The virtual reference picture is divided in \( 4 \times 4 \) blocks. After the pixel projection of every block in the temporally subsequent P picture \( P_{t+1} \), for some \( 4 \times 4 \) blocks in the virtual reference picture, not every pixel within it has a bidirectional predicted value (such as block A in picture \( P_v \)), because some pixels are not covered by projection block. For some \( 4 \times 4 \) blocks in the virtual reference picture, some pixels within it have more than one bidirectional predicted value (such as block B in picture \( P_v \)), because it is covered by more than one projection block. We can deal with these cases as follows:

![Image of pixel projection for virtual reference picture](fig4.png)

**Fig.4.** Pixel projection for virtual reference picture

In the pixel projection process, we generate the motion vector trajectory of projection from the up-left corner of each block \( M_{t+1} \) in the temporally subsequent P picture \( P_{t+1} \), as shown in Fig.4. The position and motion vector of the projection block \( M_t \) in the virtual reference picture \( P_v \) is calculated as follows:

![Diagram of pixel projection](diagram.png)
1) If every pixel in the $4 \times 4$ block has only one bidirectional predicted value (such as block $C$ in picture $P_v$), the pixel values of the $4 \times 4$ block are set to the bidirectional predicted values.

2) If some pixels in the $4 \times 4$ block have more than one bidirectional predicted value, the pixel values are set to the last bidirectional predicted values in projection order.

3) If some pixels in the $4 \times 4$ block have no bidirectional predicted value, we derive all the pixel values of the $4 \times 4$ block by spatial motion vector prediction. The forward and backward motion vectors of the block are separately obtained by median prediction technique [9]. The motion vector of current block is the median of the three motion vectors from the left, the upper, and the upper-right blocks.

### 2.3. Obtaining direct mode block from virtual reference picture

For every $B$ picture, there is a virtual reference picture in its same temporal position. The pixel values of virtual reference picture are obtained by bidirectional prediction just before the coding of the $B$ picture. When coding the current $B$ picture, for every direct mode block (the size of direct mode block is $16 \times 16$ in $16 \times 16$ mode macroblocks or $8 \times 8$ in $8 \times 8$ partitions type macroblocks), we can obtain its values using the co-located block in the virtual reference picture, as shown in Fig. 5. For the other modes in the $B$ picture, the coding method is unchanged.

![Fig. 5. Adopting co-located block in the virtual reference picture as the direct mode block in current B picture](image)

### 3. EXPERIMENTAL RESULTS

The proposed method is implemented based on the H.264/AVC reference software JM98 [10]. The test sequences with CIF format, including *foreman, tempepe, bus, container* and *paris*, are 160 frames with the frame rate of 30fps. The main test conditions are shown in Table 1. To evaluate the average rate distortion performance, we employ the method described in [11].

The performance comparison of our method with TDM in H.264 is shown in Table 2. In our method, for sequences *foreman, tempepe, bus, container* and *paris*, the bitrate reduction is separately 4.38%, 6.95%, 4.56%, 6.28% and 4.48%. The performance comparison of scheme [6] with TDM in H.264 is also shown in Table 2. It can be observed from the results that, for sequences with constant motion and sequences with irregular motion, the proposed method is always better than scheme [6].

![Table 1. Test conditions](image)

![Table 2. Performance comparison](image)

Fig. 7 shows the Rate Distortion curves of the proposed method in sequences *foreman* and *tempepe* compared with TDM in H.264. We can observe that the benefits of the proposed method tend to increase at lower bitrates, which is to be expected considering that motion information at these bitrates tends to take an even larger percentage of the coded information. In the proposed methods, all the pixel values of the virtual reference picture are obtained by bidirectional prediction along the motion vectors of moving objects in the temporally subsequent $P$ picture. The pixel values of the virtual reference picture are closer to the actual pixel values in the $B$ picture. Therefore, for every direct mode block in the $B$ picture, especially the block which has different moving objects with different motion vectors, obtaining the direct mode block values from co-located block in the virtual reference picture can bring better prediction performance. The coded residual bits in the direct mode block is reduced, the coding efficiency of whole picture is enhanced.

As for the complexity, in the pixel projection process, our method needs to calculate the location of the projection block, and the size of the block unit used for pixel projection in the temporally subsequent $P$ picture is in accordance with the size of the motion compensation block (from $16 \times 16$ to $4 \times 4$). The scheme [6] needs to calculate the location of the projected motion vector, and the size of the block unit used for motion vector tracking in the temporally subsequent $P$ picture is $4 \times 4$. So the number of projection calculation in our scheme is less than that of [6]. Our method needs to read the pixel values from temporally previous $P$ picture and temporally subsequent $P$ picture to virtual reference picture, then read the pixel values from virtual reference picture to direct mode block in current $B$ picture. One more read operation is needed in our scheme, but it is not too much time consuming. The virtual reference picture requires some increase of the memory. It can be used to predict the other block modes of the $B$ picture in the future study.

### 4. CONCLUSION

When the current block and its co-located block in backward reference picture belong to different objects with different motion directions, or there is more than one moving object with different motion vectors in current block, the traditional TDM can not bring accurate motion vector for current block in $B$ picture. To resolve this problem, this paper paper have proposed a virtual reference picture to predict the direct mode blocks in $B$ pictures, and present a pixel projection technique to generate the virtual reference picture using the temporally previous and subsequent $P$ pictures. Experimental results show that the proposed method can bring better coding efficiency than TDM scheme and motion tracking scheme [6]. In the next step, we will further study the utilization of the virtual reference picture to predict the other block modes in the $B$ pictures.
5. ACKNOWLEDGEMENT

This work has been Supported by Special Foundation of President of The Chinese Academy of Sciences under Grant No. 20064020

6. REFERENCES


Fig.6. Number of blocks coded with direct mode in each B frame

Fig.7. Rate-distortion curves of proposed scheme for sequences foreman and tempete (compared with TDM in H.264)

Table 2. Performance evaluation of the proposed scheme and scheme [6] (compared with TDM in H.264)

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Foreman Ave PSNR gain</th>
<th>Foreman Ave BR saving</th>
<th>Tempete Ave PSNR gain</th>
<th>Tempete Ave BR saving</th>
<th>Bus Ave PSNR gain</th>
<th>Bus Ave BR saving</th>
<th>Container Ave PSNR gain</th>
<th>Container Ave BR saving</th>
<th>Paris Ave PSNR gain</th>
<th>Paris Ave BR saving</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed scheme</td>
<td>0.267</td>
<td>4.38%</td>
<td>0.372</td>
<td>6.95%</td>
<td>0.281</td>
<td>4.56%</td>
<td>0.346</td>
<td>6.28%</td>
<td>0.283</td>
<td>4.48%</td>
</tr>
<tr>
<td>Motion tracking scheme [6]</td>
<td>0.053</td>
<td>0.89%</td>
<td>0.205</td>
<td>3.92%</td>
<td>0.072</td>
<td>1.09%</td>
<td>0.172</td>
<td>3.23%</td>
<td>0.114</td>
<td>1.81%</td>
</tr>
</tbody>
</table>