Fast Algorithms for Inter-view Prediction of Multiview Video Coding

Rong Pan, Zheng-Xin Hou and Yu Liu
School of Electronic Information Engineering, Tianjin University, Tianjin, China
Email: panrong@tju.edu.cn, zhengxinhou@163.com, liuyu@tju.edu.cn

Abstract—The compression efficiency and complexity should be simultaneously taken into consideration in Multiview Video Coding (MVC). In order to reduce the complexity while maintaining the efficiency of MVC, we proposed in this paper two algorithms called Adaptive Search Region Adjustment (ASRA) and Adaptive Selection of Inter-view References (ASIR). In ASRA, we simplified the exiting method to obtain the global disparity vector (GDV), using which the correlations between the disparity vector (DV) and predicted vector were exploited. The statistical property of the absolute difference value of DV and GDV was analyzed and the results were applied to adjust the search range dynamically. In ASIR, we first used the prediction information of the lower temporal layer in the hierarchical B picture prediction structure to reduce the number of inter-view reference pictures, and then applied the results of reference selection after inter_16×16 prediction mode decision to revise our method. The experimental results demonstrate that the algorithms suggested in this paper can reduce the computation up to 25% with little decrease of compression efficiency.

Index Terms—Multiview Video Coding (MVC), Inter-view Estimation, Global Disparity Vector (GDV), Hierarchical B Picture Prediction Structure (HBPS)

I. INTRODUCTION

There is strong evidence that multiview videos will be widely used in the applications such as 3DTV, free viewpoints video [1] and high performance imaging [2]. The distinguishing characteristic of multiview videos captured by a set of cameras is the existence of strong correlations between different views, which requires and enables developing new technologies to efficiently compress the large quantities of data of multiview videos [3]. Recently, the draft of Multiview Video Coding (MVC) as Annex H of H.264 [4] has been submitted to JVT (Joint Video Team of ISO/IEC MPEG & ITU-T VCEG) for discussion [5].

The MVC scheme based on H.264 in [5] is proposed by HHI [6]. To exploit the redundance between different views, the methodology of inter-frame prediction in H.264 is extended to inter-view prediction in MVC, i.e., pictures in other views can be used as references of the pictures being encoded in the current view. Besides, the temporal Hierarchical B Picture Prediction Structure (HBPS) proposed in Scalable Video Coding (SVC, annex G of H.264 [4]) is also used in MVC to provide temporal scalability. The technologies such as multiple references and various block sizes from 16×16 to 4×4 inherited from H.264 bring not only high compression performance to MVC, but also high computational complexity, especially in the context of the multiplied data of multiview videos. Thus, it is necessary to develop efficient fast algorithms for MVC to promote the applications of multiview videos in some systems such as mobile stereoscopic multimedia system in power-constrained devices and real-time multiview video broadcasting system.

Motion estimation (ME) and mode decision (MD) are two main time-consuming processes of video coding as well as MVC. ME can be sped up by fast search algorithms or dynamic search range reduction, and MD can be sped up by technically selecting the most probable modes partly among which the best one is determined. A number of methods [7]–[9] developed to reduce the high complexity of ME and MD in single-view video coding technologies such as H.264 can be directly used in MVC, e.g., Peng and Jiang, etc. [10], [11], applied the fast MD algorithms just based on classical mode analysis. But the fast algorithms for single-view video coding are not efficient enough for MVC, since the complexity of disparity estimation (DE) and MD between different views has not been considered yet. By now, a number of fast algorithms especially employing to MVC based on exploiting the correlations between different views have been developed.

Lu and Cai, etc. [12], proposed an epipolar geometry-based fast DE algorithm to greatly reduce the search range and effectively track large and irregular disparity vector (DV) for inter-view prediction in MVC. Y. Kim and J. Kim, etc. [13], applied the geometry of camera arrangement to acquaint the disparity reliability of different views and a pair of predictive vectors (PV) according to the relationship between DV and motion vector (MV), and decided the accuracy of the PV to adaptively adjust the search range for both intra-view and inter-view predictions. The two methods above have the weak-point that the camera configuration and calibration information should be known in advance. Cernigliaro and Jaureguizar, etc. [14], used depth maps to estimate the likely structure of the motion field for fast MD in intra-view prediction, and applied the MD results of neighboring views together with the depth information to make the results more reliable. But it was complicated to compute depth maps or expensive to obtain those using depth cameras. Xu and He, etc. [15], recorded the disparity maps of encoded pictures as disparity predictors to adaptively adjust the search range for the incoming
inter-view-predicted pictures of the same view, but this method was no longer feasible when the encoded picture was not temporally adjacent with the incoming picture, and the results were given only under I-P-P-P coding structure. Yu and Peng, etc. [16], used global disparity vector (GDV) to find the corresponding blocks in other views of the current one being encoded, and selected the sub-optimal mode according to the modes of the corresponding blocks for intra-view prediction.

In this paper, we proposed two complementary fast algorithms. In the first algorithm, an approximate GDV was obtained by an improved method in order not to influence the coding time too much. Different from the method proposed by Yu and Peng, etc. [16], we used the GDV to dynamically adjust the search range to speed up DE rather than MD, by finding the statistical correlation among DV, GDV and PV which were bounded by the triangle inequality. It was the main contribution of the second algorithm to use the characteristics of HBPS to mark off a region containing the blocks most probably applying inter-view references, reducing the total times of using inter-view prediction. Our methods were developed only for the time saving of inter-view prediction in MVC, and the first one was suitable for encoding multiview videos captured by parallel cameras, while the second one applied to all videos encoded with HBPS.

In Section II, the prediction structure of MVC was introduced and some definitions for this paper were made. Section III proposed a fast algorithm to get the GDV, and then dynamically adjusted the disparity search range to reduce the searching time by analyzing the correlations between the real DV, GDV and PV. Section IV proposed another algorithm to reduce the number of reference pictures according to the characteristics of HBPS. The experimental results were presented and discussed in Section V, and Section VI concluded this paper.

II. Prediction Encoding Structure for MVC

The prediction structure of MVC shown in Fig. 1 stems from the HBPS used in SVC [17]. In an HBPS, the reference picture list 0 and 1 are restricted to the temporally preceding and succeeding picture respectively with a temporal layer identifier (TL in Fig. 1) smaller than that of the current predicted picture, making the temporal scalability of HBPS [18] possible.

Only three views are shown in Fig. 1 since the encoding process for three views can fully represent the characteristics of MVC. The length of Group of Pictures (GOP) of the coding structure shown in Fig. 1 is 12, and the indicator POC denotes the picture order count, i.e., picture display order, and Vi, i = 0, 1, 2, is the view number. Pictures in an HBPS can be represented by PVi,POCj, i = 0, 1, 2, j = 0, 1, 2, ..., 12, e.g., the top-left picture is P0,0,0, and its temporal layer identifier is TL0. In Fig. 1, the HBPS has five temporal layers, TL0, ..., TL4. The set of {TL0, ..., TLi} may be decoded independently of other layers with temporal layer identifiers larger than TLi, so different temporal resolution can be selected to provide streams with various bitrates. Specially in MVC, the reference pictures of blocks in the predicted picture may be from any other views, that is, more strictly, the reference picture lists 0 and 1 not only include pictures in the same view, but also pictures in the backward and forward views with the same POC of the predicted picture. In Fig. 1, P1,1,0,6 has four reference pictures that can be selected: P1,0,0,0 (list 0), P1,1,1,0 (list 1) in the same view and P0,0,6,6 (list 0), P2,2,6,6 (list 1) in other views with the same POC.

The four concepts ME, MV, DE and DV mentioned in this paper are correlative. For example in Fig. 1, the process of searching the best matches in P1,1,0,1 or P1,1,1,1 for blocks in P1,1,1, in the same view is ME, and the vector obtained through ME is MV. Accordingly in different views, if P0,0,6 or P2,2,6 are the reference pictures, the process is called DE, and the corresponding vector is DV. Especially, the reference pictures in the same view of the current predicted picture are called intra-view references in this paper, and the ones in other views are inter-view references.

III. Adaptive Search Range Adjustment Based on Global Disparity

An important characteristic of multiview videos is the existence of global disparities generated by different camera shooting angles between different views. We can technically find the corresponding points in stereo matching pictures through multiview geometry, and obtain the exact value of DV from one pixel to its correspondence. However, we cannot get an exact value of GDV because of different object depths, but only an approximate one by calculating a pair of pictures with the same POC in different views. In the following, we propose a fast algorithm called Adaptive Search Range Adjustment (ASRA) based on the GDV obtained by an improved method.

A. Global Disparity Vector

In this subsection, we propose an improved method to obtain the GDV which is not very exactly but quickly as well as sufficiently as the basis of ASRA. Lee and Park, etc. [19], proposed a full-pixel search algorithm to obtain GDV when the optimum matching areas of two corresponding pictures were determined. Yang and Huo, etc. [20], applied eight-pixel searching step instead of one-pixel to develop an improved algorithm, which has been

![Figure 1. Hierarchical B Picture Prediction Structure for MVC. POC, V and TL denote the Picture Order Count, View Number and Temporal Layer, respectively. Each picture can be denoted by a pair of V and POC, as: PVi,POCj.](image-url)
integrated into the MVC test model JMVM 8.0 [21] to obtain GDV for the coding tool of motion skip. Though the latter algorithm is obviously not as accurate as the former, it is much faster and the accuracy of the result is enough for application. The calculating formulas of GDV proposed in [20] are as (1) and (2):

$$GDV = (x, y) = \arg \min_{-SR \leq x, y \leq SR} \{MAD(8x, 8y)\} \tag{1}$$

$$MAD(x, y) = \frac{1}{(h - y)(w - x)} \sum_{i=0}^{w-1-x} \sum_{j=0}^{h-1-y} \left| I_r(i + x, j + y) - I_c(i, j) \right| \tag{2}$$

In (1), $SR$ is the search range. In (2), $h$ is the height of a picture, and $w$ is the width. $I_r$ and $I_c$ are the values of luma samples in current picture and in one of its inter-view references respectively. As shown in Fig. 2, the matching regions indicated by $O_{curr}$ and $O_{ref}$ in the current picture $P_{curr}$ and reference picture $P_{ref}$ respectively are enclosed by the bold black solid lines. $O_{curr}$ and $O_{ref}$ have the same size and correlative positions, varying according to the absolute values and directions of $x$ and $y$, e.g., the regions enclosed by dashes form another matching pair. The $MAD$s of all pairs of matching regions are calculated by (2), and a proper GDV is determined with the minimum $MAD$. GDV has the unit of eight-pixel as well as $SR$, $h$ and $w$, and is in the range of $(-SR, SR)$.

The acquisition of GDV is an additional and time-consuming process. In order not to greatly influence the efficiency of ASRA, we select a much smaller pair of matching regions as the shaded area in Fig. 2 to further reduce the calculation amount but get a decent GDV for our algorithm. The size of the shaded area is denoted by $B_x$ and $B_y$, which can be selected with proper values to provide a tradeoff between time and accuracy of the GDV. Another time control strategy is that this GDV estimation process is only executed in the anchor pictures such as $P_{V1,POC0}$ and $P_{V1,POC12}$ in Fig. 1.

Figure 2. Acquisition of GDV: $P_{curr}$ and $P_{ref}$ denote the current picture and the reference respectively. When the MAD of the shaded areas of the match region $O_{curr}$ and $O_{ref}$ reaches to the minimum, GDV = $(x, y)$ is obtained.

### B. Adaptive Search Range Adjustment

As processed in the ME of H.264, a predicted vector (PV) of the current block under inter-view prediction is formed based on calculated DVs of nearby previously coded blocks [22]. One of the features of PV is providing a searching start point for current DE. The search terminates at the boundary of the search region, which may be narrowed if we know the approximate place the finally obtained DV points to. We find that there is strong correlation between $DV$ and $GDV$, whose difference value is relatively small in most cases. The difference value of $DV$ and $GDV$ is defined as residual vector (RV):

$$RV_i = |DV_i - GDV_i| \tag{3}$$

where $i = x, y$, the indexes of two components of the vectors, the same as below. Through statistical analysis of some sequences introduced in Section V, the distribution of the values of $RV_i$ is obtained and shown in Fig. 3. It can be seen that $RV_i$ is distributed within a small interval. The finally obtained $DV$ cannot be known in advance, but its approximate value can be deduced by $RV_i$. The absolute difference of $DV$ and $PV$ is restricted by the triangle inequality as follows:

$$|DV_i - PV_i| \leq |DV_i - GDV_i| + |PV_i - GDV_i| = RV_i + |PV_i - GDV_i| \tag{4}$$

In (4), values of $PV_i$ and $GDV_i$ are already known, and the overwhelming majority of values of $RV_i$ fall into a small range just as indicated in Fig. 3. So the search range can be reduced by the following:

$$Curr_{SR_i} = \text{TRUNC}_{SR}((RV_i + |PV_i - GDV_i|) >> 2) \tag{5}$$

The operator $\text{TRUNC}_{SR}$ is:

$$\text{TRUNC}_{SR}(x) = \begin{cases} 
\text{Orig}_{SR} & \text{if } x > \text{Orig}_{SR} \\
 x & \text{otherwise}
\end{cases} \tag{6}$$

If the references are inter-view ones, $Curr_{SR_i}$ in (5) is calculated and applied as the current search range, where the operation $>> 2$ means the unit is transformed from 1/4 pixel to one pixel. Though the original search region is a square with the searching start point as its center, the current region is a rectangle because its width and height are calculated by (5) separately.

### IV. Adaptive Selection of Inter-view References Based on HBPS

#### A. Encoding Characteristics of MVC

The inter-view prediction structure of MVC is determined by the camera arrangement, and the selection of reference views obeys the principle of proximity. Differently, pictures in the intra-view, i.e., inter-frame prediction structure introduced in Sec. II as HBPS are
encoded layer by layer. With the increment of the layer identifier, the temporal distance between current picture and any one of its intra-view references is decreasing, which means the correlation between these two pictures is stronger. Therefore, the pictures with larger TLs tend to use intra-view references with higher possibility. We propose here both statistical and theoretical analyses to illustrate these characteristics using which to develop efficient approaches for the acceleration of inter-view prediction.

1) Statistical Analysis: We do our statistical analysis using Fig. 4 and 5, and reach two conclusions: first, the proportion of blocks applying inter-view references in each picture decreases while the layer identifier increases, and second, blocks applying inter-view references are much fewer than the ones applying intra-view references.

The first conclusion can be drawn from Fig. 4. For example, the proportions of blocks applying inter-view references in \( P_{V1,POC6} \), \( P_{V1,POC3} \) and \( P_{V1,POC9} \) are 23\%, 17\% and 18\% respectively. The layer identifiers of \( P_{V1,POC3} \) and \( P_{V1,POC9} \) are both 2, higher than that of \( P_{V1,POC6} \) which, however, has a higher proportion of blocks applying inter-view references. The reason is that the distance between a pair of pictures with lower adjacent TLs is greater, and vice versa, e.g., the distance between \( P_{V1,POC0} \) (TL = 0) and \( P_{V1,POC6} \) (TL = 1) is 6, and that between \( P_{V1,POC6} \) (TL = 1) and \( P_{V1,POC3} \) (TL = 2) is 3. We can draw from Fig. 1 that the intra-view references of a picture are located in its temporal adjacent lower layer, e.g., \( P_{V1,POC0} \) is one of the references of \( P_{V1,POC6} \), and \( P_{V1,POC6} \) is one of the references of \( P_{V1,POC3} \). Larger distance results in lower degree of similarity between two pictures, in which one picture has less probability to find optimum corresponding blocks in the other. In other words, it is more likely for blocks in a predicted picture to find their correspondences in inter-view references rather than intra-view ones whose distances to the predicted picture are larger. In addition, for pictures in the base temporal layer, \( P_{V1,POC0} \) and \( P_{V1,POC12} \), the proportions are above 90\% since these pictures don’t have any intra-view references but only inter-view ones.

The second conclusion increases the possibility for us accelerating the coding process if we predict whether the blocks need intra-view or inter-view references in advance by some strategies. In Fig. 5, the blocks marked with red are the MBs using inter-view references, and their number decreases with the increment of TL—the picture with TL = 2 has more blocks applying inter-view references than the one with TL = 3. These blocks mainly scatter on the moving and especially rotating objects in the pictures,
and the others applying intra-view references mostly locate in the background. Generally, the background and objects in translation movement in a picture cover most of the area, and pictures captured by different cameras have different projective distortion, making the proportion of blocks applying intra-view references much larger in a picture. For continuous motion, the moving objects in one picture can be traced by their positions in the temporal adjacent pictures. Thus the prediction information of the encoded pictures in the lower temporal layers is able to be used to pre-judge the possible reference pictures for the ones in the higher layers, and then reduce the number of reference pictures. Statistical analyses for other test sequences have similar results as for Ballroom shown in Fig. 4 and 5.

2) Theoretical Analysis: In this analysis, we intend to prove that most of the blocks applying inter-view references locate on the rotating objects in the sequences. To simplify the analysis, we assume two cameras at work are fixed, and the 3D object being recorded is rigid in both translation and rotation movements. The 2D points corresponding to the 3D object at the image plane of a camera is expressed as $P_{2D}$:

$$P_{2D} = P_r \cdot P_{3D}$$  \hspace{1cm} (7)

In (7), $P_r$ denotes the projective matrix of the first camera integrated with the scene depth, and $P_{3D}$ is the coordinates of the 3D object. If replace $P_r$ with $\tilde{P}_r$, the projective matrix of the second camera, we obtain the 2D points $\tilde{P}_{2D}$ on the other image plane:

$$\tilde{P}_{2D} = \tilde{P}_r \cdot P_{3D}$$  \hspace{1cm} (8)

Let $P'_{3D}$ be the 3D coordinates of the object after movement, expressed as [23]:

$$P'_3 = R \cdot P_{3D} + T_{3D}$$  \hspace{1cm} (9)

In (9), $R$ and $T_{3D}$ denote the rotation matrix and translation vector for 3D movement respectively. We then get the 2D points $P'_{2D}$ of the object projected to the first camera after movement by substituting $P'_{3D}$ into (7):

$$P'_{2D} = P_r \cdot R \cdot P_{3D} + T_{3D}$$  \hspace{1cm} (10)

In (10), $T_{2D}$ is the 2D translation vector expressed as:

$$T_{2D} = P_r \cdot T_{3D}$$  \hspace{1cm} (11)

To illustrate the effect of the 3D movement, we assume $P_r$ and $\tilde{P}_r$ are such matrices which maintain the values of $x$ and $y$ of the 3D coordinates with a multiplicative factor $a$ but set the value of $z$ (the depth direction) as 0, i.e., $P_{2D} = a \cdot P_{3D,z=0}$, $\tilde{P}_{2D} = a \cdot P_{3D,z=0}$, and $P'_{2D} = a \cdot R \cdot P_{3D,z=0} + a \cdot T_{3D,z=0}$. Neglect the influence of translation on the shape change such that $P''_{2D} = a \cdot R \cdot P_{3D,z=0}$. It has to be noted that the 2D points obtained above should be rounded off and the same points should be merged in order to remove the occlusion part on the image. Let the visible part of a 3D object opposite to a camera have the area of $p$, if the 3D object rotates about the axis parallel to the image plane with angle $\theta$, then the remainder of visible part corresponding to the camera is $p_r = b \cdot p$, i.e., $P'_{2D} = b \cdot P_{2D}$, $b \propto 1/\theta$. Since $P_{2D} = P_{3D}$, it is better for the picture in the second view to be the reference of the blocks locating on the rotating object in the picture captured by the first camera at the same instant.
Consider the influence of 3D translation. Let the 3D object move uniformly along the $x$ axis, and then the translational distance on the image plane is obtained as $x_{2D} = c \cdot x_{3D}$, where $c = P_i(0, 0)$. In general, $x_{2D}$ determines the size of the range within which we can track the moving object, and is proportional to the distance of two pictures in a view:

$$x_{2D} \propto |POC_i - POC_j|$$  \hspace{1cm} (12)

According to the analysis, it is possible for us to pre-judge the moving object suited to apply inter-view references, and decrease the number of references to reduce the calculating amount. In fact, we only need to track the rotating objects and in most of the real scenes the $x_{2D}$ has a much smaller value, illustrated in Subsection IV-B.2. In practice, we use the method of 3D warping [24] to realize the projection from 3D to 2D or the reverse, and so called quaternion [25] to accomplish warping [24] to realize the projection from 3D to 2D or the information of the current predicted picture after applying inter-view references in the previous encoded pictures in the lower layer of HBPS is firstly applied to obtain $IIVR$ with an approximate range, and the ones in $OIVR$ only apply intra-view references. The main target is to obtain a proper division of $IIVR$ and $OIVR$ for a predicted picture in order to reduce the prediction time without loss of coding efficiency. In our method ASIR, The prediction information of encoded pictures in the lower layer of HBPS is firstly applied to obtain $IIVR$ with an approximate range, and the information of the current predicted picture after inter$_{16 \times 16}$ mode decision is used for amendment.

1) Prediction Information of Pictures in the Lower Layer: We define part of $IIVR$ as $IIVR_A$, which is the dilatation of the positions of corresponding MBs applying inter-view references in the previous encoded picture nearest to the current predicted picture. The strict definition of $IIVR_A$ is as follows:

$$IIVR_A = \{(TRUNC_L(x), TRUNC_L(y))\}$$ \hspace{1cm} (13)

The operator $TRUNC_L$ is defined as:

$$TRUNC_L(z) = \begin{cases} 
0 & z < 0 \\
\frac{1}{l} & 0 \leq z < \frac{1}{l} \\
\frac{1}{l} - 1 & z \geq \frac{1}{l}
\end{cases}$$ \hspace{1cm} (14)

In (14), the value of $l$ is obtained as follows:

$$l = \begin{cases} 
picture_width_{\text{in}_{\text{pixel}}} & z = x \\
picture_height_{\text{in}_{\text{pixel}}} & z = y
\end{cases}$$ \hspace{1cm} (15)

In (13), $x$ and $y$ should match the following condition:

$$\begin{cases} 
|x - x_{LF}| & \leq TH_1 \\
|y - y_{LF}| & \leq TH_1
\end{cases}$$ \hspace{1cm} (16)

(13) indicates that $IIVR_A$ is made up of some MBs with coordinates of $TRUNC_L(x)$ and $TRUNC_L(y)$. The region outside $IIVR_A$ is called $OIVR_A$. In (16), $(x_{LF}, y_{LF})$ are the coordinates of MBs in one of the intra-view references of the current one, and its layer identifier $TLr = TLc - 1$. $TLc$ is the layer identifier of the current picture, and $TLc > 0$, i.e., this approach is not applied to the anchor pictures. $TH_1$ is a threshold used to restrict the value of $(x, y)$, and can be dynamically adjusted by the distance of the current predicted picture and its reference according to (12). That is, $TH_1$ is suggested to be larger in order to contain as many as MBs requiring inter-view estimation when the temporal distance increases, but practically, we use a fixed value since it is adequate for tracking a rotating object and the $IIVR_B$ described in next subsection can be an effective supplement of $IIVR_A$.

2) Prediction Information of Inter$_{16 \times 16}$ Mode: However, cases such as rapid or sudden movement of the objects may result in great loss of MBs constituting $IIVR$, which should be averted by some supplementary means. We notice that if a MB applies inter-view references after inter$_{16 \times 16}$ prediction mode decision, partitions of this MB may apply inter-view references very likely in the end. The coincidence proportion is above 80% as shown in Fig. 7. So we process the inter$_{16 \times 16}$ prediction at first, and then define $IIVR_B$ according to whether the inter-view or intra-view references are applied:

![Coincidence percentage of blocks applying inter-view references](image-url)
Figure 8. Pinhole camera model, by which we find in the 3D space the corresponding length \(mbh\) of \(mbs = 16\) pix, the size of a MB in the image plane, and find the maximum translation velocity that can be traced by assigning 1 to \(TH_1\) in the method ASIR.

\[
IIVR_B = \{(TRUNC_L(x), TRUNC_L(y))\} \quad (17)
\]

In (17), \(x\) and \(y\) should match the following condition:

\[
\begin{cases}
| x - x_{16 \times 16, OV} | \leq TH_2 \\
| y - y_{16 \times 16, OV} | \leq TH_2
\end{cases} \quad (18)
\]

In (18), \((x_{16 \times 16, OV}, y_{16 \times 16, OV})\) are the coordinates of the MBs applying inter-view references after inter\(16 \times 16\) prediction in \(OIVR_A\) of the predicted picture. \(TH_2\) is a threshold value restricting the range of \((x, y)\), and \(TH_2 \geq 0\). The final \(IIVR\) is then obtained by:

\[
IIVR = IIVR_A \cup IIVR_B \quad (19)
\]

\(OIVR_B\) is the region outside \(IIVR_B\). And the final \(OIVR\) is:

\[
OIVR = OIVR_A \cap OIVR_B \quad (20)
\]

When simply assign 1 to \(TH_1\) and \(TH_2\), we obtain in Fig. 5 the \(IIVR_A\) in the first picture, \(IIVR_B\) in the second, and \(IIVR\), the combination of \(IIVR_A\) and \(IIVR_B\), in the third. For the 3rd and 4th pictures in Fig. 5, all blocks applying inter-view references are in the range of \(IIVR\), which means if the MBs in \(IIVR\) are regularly predicted with both intra-view and inter-view references while others predicted with only intra-view references, the coding time will be reduced without any PSNR loss. It seems that 1 is a sufficient value for \(TH_1\) and \(TH_2\) determining a proper range of \(IIVR\). This phenomenon also arises when testing other sequences, explained as follows.

It is above 80% MBs applying inter-view references finally when they find their optimum matching blocks in other views after inter\(16 \times 16\) prediction, thus it is sufficient to assign 1 to \(TH_2\). In the following we mainly discuss why \(TH_1 = 1\) is enough to track the blocks applying inter-view reference. We use the pinhole camera model to illustrate what’s the corresponding length of a MB when back projected into the 3D space, as shown in Fig. 8, where the object in the distance \(s\) with length \(h\) is captured by a camera with a view angle \(\alpha\), and projected onto the image plane with length \(d\). Besides in Fig. 8, \(mbs = 16\) pixels, the size of MB, and its corresponding length in the 3D space is \(mbh\). Our work is to analyze the corresponding relation between \(mbs\) and \(mbh\) when the two conditions \(\alpha\) and \(s\) change. The results are shown in Tab. I.

### Table I. Parameter values of a pinhole camera under different conditions

<table>
<thead>
<tr>
<th>(\alpha) (deg)</th>
<th>(s) (m)</th>
<th>(h) (m)</th>
<th>(d) (pix)</th>
<th>(d) (pix)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(60)</td>
<td>(15)</td>
<td>(17.3)</td>
<td>(0.43)</td>
<td>(0.27)</td>
</tr>
<tr>
<td>(80)</td>
<td>(6)</td>
<td>(10.1)</td>
<td>(0.25)</td>
<td>(0.16)</td>
</tr>
<tr>
<td>(10)</td>
<td>(30)</td>
<td>(36.4)</td>
<td>(0.91)</td>
<td>(0.57)</td>
</tr>
</tbody>
</table>

Based on the real camera calibration, we let \(\alpha\) vary from 80, 60 to 40 degrees, and \(s\) from 6, 15 to 50 meters accordingly, since cameras with narrower view angles can focus on objects in longer distance. If we let \(d\) be 640 and 1024 pixels, then \(mbs = 1/40\) and \(1/60\) of \(d\) respectively, and these two ratios of \(mbh\) to \(h\) are the same according to triangle geometry. \(h\) can be obtained by \(h = \tan(\alpha/2) \cdot s \cdot 2\). In Tab. I, \(v\) is the maximum translation velocity of the 3D object in movement which can be reflected by the size of a MB in a image. For example, in Fig. 1, the test sequence has 30 frames per second, and the largest distance between two pictures using ASIR with adjacent \(TLs\) is 3, e.g., the distance from \(PV_{1, 0C3}\) to \(PV_{1, 0C6}\). If the mapping of the 3D object moves from \(PV_{1, 0C3}\) to \(PV_{1, 0C6}\) in 16 pixels, the velocity is \(mbh/(3/30)\) (m/s). The largest value in short distance of \(v\) is 2.5 m/s, which is the velocity of a man walking fast, and the largest in long distance is 9.1 m/s, which is the velocity of a sprinter. So generally, we can use one MB size to trace most of the translation movement. But how about the exceptions, such as a car in a rally race? We realize that MBs applying inter-view references mostly locate on the objects in rotation but not in translation movement, as analyzed in Subsection IV-A. Objects in rotation wouldn’t have high speed in translation movement very likely. And eventually, we use \(IIVR_B\) as an effective supplement to ensure the accuracy of ASIR to prevent the influences of extreme cases.

Finally, it has to be noted that 1) the \(IIVR\) is determined with MB but not smaller unit size, since the unit size of \(IIVR_B\) is \(16 \times 16\) naturally, and \(IIVR\) is a somewhat rough but not such precise region just for prediction, which means \(IIVR_A\) doesn’t need the precision of sub-partitions of a MB; and 2) though the data of Fig. 4 and 7 is drawn from Ballroom, the statistical laws shown in these figures are applicable to other sequences.
The improved algorithm in Subsection III-A is used to estimate the GDVs as the basis of ASRA. We should obtain proper values of $B_x$ and $B_y$ at first, by testing on matching regions with different sizes to get a series of corresponding GDVs. We compare these GDVs with the ones obtained by the original algorithm in terms of Sum of Absolute Difference (SAD) calculated by (21):

$$SAD_{GDV} = \sum_{i=1}^{7} \sum_{j=1}^{2} |GDV_{ijk,prop} - GDV_{ijk,orig}|$$

(21)

In (21), $i$ is the index of all the seven $640 \times 480$ test sequences, $j$ and $k$ denote the indexes of two reference lists and two different components of GDV, respectively. The first variable between the signs for absolute value on the right-hand side is obtained by the proposed algorithm, and the second obtained by the original. The plot of $SAD_{GDV}$ vs. scaling factors of $B_x(y)$ to $O_{curr(ref)}$ with different values in Fig. 2 is shown in Fig. 9, indicating the $SAD_{GDV}$ is steadily increasing when the scaling factor is smaller than 1/2. Therefore we assign $B_x$ and $B_y$ with the values of half the sizes of $O_{curr}$ and $O_{ref}$, reducing nearly 1/4 calculation amount of the algorithm proposed in [20]. GDVs obtained by the proposed algorithm and the original one respectively are shown in Tab. III. It can be drawn from Tab. III that the GDVs obtained by the improved algorithm are almost all the same as those obtained by the algorithm proposed in [20], and the accuracy is enough for ASRA. A pair of values separated by backslashes in Tab. III are GDVs obtained between current view and its backward/forward views respectively. No view of Flamenco2 has a forward reference, thus the forward GDV of flamenco is not available.

### B. Rate-Distortion Curves

Four rate-distortion (RD) curves are plotted in Fig. 10 with Peak Signal to Noise Ratio (PSNR) vs. bitrate after encoding the test sequence Ballroom by different methods separately. The methods are ASRA, ASIR, the combination of ASRA and ASIR, and the original algorithms integrated in JMVM 8.0. The four curves are completely coincident with each other from subjective observe, indicating that the compression performance is maintained when using our fast algorithms.

After encoding the other test sequences, we get highly similar results in terms of RD curves compared with those of Ballroom, i.e., these curves overlap each other, thus only the plot of Ballroom is proposed here. It has to be noted that ASRA is quite effective for all of the sequences with spatial resolution $640 \times 480$, but not so effective for the $1024 \times 768$ ones, resulting in lower curves of ASRA than others for sequences with $1024 \times 768$ resolution. But it is not because of the seeming differences in their resolution, but different camera arrangements. We can draw from Tab. II that $640 \times 480$ sequences are all produced by cameras with parallel arrangement (it is array or cross for Race1, Akko & Kayo, and Flamenco2, but at the horizontal and vertical direction respectively, it is parallel), and the $1024 \times 768$ sequences are not. Non-parallel arrangement of cameras makes the multiview videos having no such accurate GDVs used.
in ASRA as the basis to analyze the statistical properties of $RV$. So the ASRA performs not so well and the RD curves of $1024\times768$ sequences obtained by ASRA are distinguishable lower than the curves by JMVM 8.0, i.e., ASRA is not able to maintain the coding efficiency for $1024\times768$ sequences which are produced by non-parallel cameras. The curves are not presented here too, since we choose ASIR only for encoding the $1024\times768$ sequences in our experiment, and the amount of time saved is still satisfactory.

C. Bjontegaard Method

Though R-D curve is popular in measuring the efficiency of a video compression tool, it is not suitable when coding results have only minor differences. In order to have a finer objective evaluation of our results, we use Bjontegaard method [27] to calculate the average $PSNR$ differences $\Delta PSNR$ and the amount of $bitrate$ change $\Delta bitrate$ between R-D curves obtained by the proposed algorithms and the original algorithm in JMVM 8.0. We integrate our algorithms ASRA (indicated by I), ASIR (II), the combination of ASRA and ASIR (I&II) into JMVM 8.0 separately and obtain three sets of values in terms of $PSNR$, $bitrate$ and $time$ after encoding all the test sequences. Bjontegaard method is then used to calculate the $\Delta PSNR$ and $\Delta bitrate$ between these three sets of values and those obtained by the original algorithm in JMVM 8.0 respectively. Let $\Delta time$ denote the ratio of the encoding time reduced by the fast algorithms, calculated by (22):

$$\Delta time = \frac{1}{\sum_{Q_P} time(Q_P)_{prop} - time(Q_P)_{orig}} \times 100\%$$

### Table II. Coding Parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>MERL</th>
<th>HII</th>
<th>Race1</th>
<th>KDDI</th>
<th>Microsoft</th>
<th>Nagoya_u</th>
<th>Akko &amp; Kayo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame size</td>
<td>640×480</td>
<td>1024×768</td>
<td>640×480</td>
<td>640×480</td>
<td>1024×768</td>
<td>640×480</td>
<td>640×480</td>
</tr>
<tr>
<td>Arrangement</td>
<td>parallel</td>
<td>convergent</td>
<td>parallel(array)</td>
<td>parallel(cross)</td>
<td>arc</td>
<td>parallel</td>
<td>parallel(array)</td>
</tr>
<tr>
<td>$GOP$</td>
<td>12</td>
<td>12</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>$Basic QP$</td>
<td>22 27 32 37</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\Delta QP$</td>
<td>0 3 4 5 6 7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table III. GDV Obtained by Simplified Algorithms

<table>
<thead>
<tr>
<th>GDV_Algorithm</th>
<th>Ballroom</th>
<th>Exit</th>
<th>Vassar</th>
<th>Nagoya_u</th>
<th>Rena</th>
<th>Akko &amp; Kayo</th>
<th>KDDI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>(2.0)(-1.0)</td>
<td>(4.0)(-3.0)</td>
<td>(1.0)(-1.0)</td>
<td>(2.0)(-2.0)</td>
<td>(0.0)/(0.0)</td>
<td>(0.3)/NA</td>
<td>(3.0)/(2,-3)</td>
</tr>
<tr>
<td>Proposed</td>
<td>(1.0)(-1.0)</td>
<td>(4.0)(-4.0)</td>
<td>(1.0)(-1.0)</td>
<td>(2.0)(-2.0)</td>
<td>(0.0)/(0.0)</td>
<td>(0.3)/NA</td>
<td>(3.0)/(2,-3)</td>
</tr>
</tbody>
</table>

1 (2.0) is the GDV between view 1 and 0 of Ballroom, and (-1.0) is that between view 1 and view 2. Others are similar.

### Table IV. Experimental Results Obtained by Bjontegaard Method

<table>
<thead>
<tr>
<th>Parameters</th>
<th>MERL</th>
<th>Ballroom</th>
<th>Exit</th>
<th>Vassar</th>
<th>Nagoya_u</th>
<th>Rena</th>
<th>Akko &amp; Kayo</th>
<th>KDDI</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Delta PSNR$ (dB)</td>
<td>-0.03</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.01</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>$\Delta bitrate$ (%)</td>
<td>0.90</td>
<td>0.02</td>
<td>0.01</td>
<td>-0.19</td>
<td>-0.09</td>
<td>-0.37</td>
<td>0.18</td>
<td>-0.05</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Akko &amp; Kayo</th>
<th>Rena</th>
<th>Microsoft</th>
<th>Breakdancers</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Delta PSNR$ (dB)</td>
<td>0.00</td>
<td>0.00</td>
<td>0.01</td>
<td>0.02</td>
</tr>
<tr>
<td>$\Delta bitrate$ (%)</td>
<td>-0.10</td>
<td>-0.05</td>
<td>-0.16</td>
<td>-0.46</td>
</tr>
<tr>
<td>$\Delta time$ (%)</td>
<td>-19.65</td>
<td>-11.55</td>
<td>-23.61</td>
<td>-11.00</td>
</tr>
</tbody>
</table>

1 Only the first algorithm ASRA is used. 2 Only the second algorithm ASIR is used. 3 Both ASRA and ASIR are used to speed up the prediction.

© 2011 ACADEMY PUBLISHER
TABLE V. THE ABSOLUTE TIME UNDER DIFFERENT QPs

<table>
<thead>
<tr>
<th></th>
<th>22</th>
<th>27</th>
<th>32</th>
<th>37</th>
</tr>
</thead>
<tbody>
<tr>
<td>JMVM 8.0 (s)</td>
<td>163.03</td>
<td>154.23</td>
<td>145.42</td>
<td>135.83</td>
</tr>
<tr>
<td>ASRA &amp; ASIR (s)</td>
<td>120.69</td>
<td>112.92</td>
<td>109.56</td>
<td>108.08</td>
</tr>
</tbody>
</table>

Where \( n = 4 \), the number of QPs with different values. \( \text{time}(QP)_{\text{prop}} \) and \( \text{time}(QP)_{\text{orig}} \) denote the absolute encoding time with a specific QP when using JMVM 8.0 and the proposed fast algorithms respectively. As an example, the absolute time of encoding Ballroom with a GOP length using JMVM 8.0 and the combination of ASRA and ASIR are proposed in Tab V. All the results in terms of \( \Delta \text{PSNR} \), \( \Delta \text{bitrate} \), and \( \Delta \text{time} \) are shown in Tab IV. The values of the parameters in Tab IV are reduced by our algorithms if they are negative represented, and vice versa. Only ASIR is used to accelerate the processes of encoding the 1024×768 sequences provided by HHI and Microsoft. Here in ASRA, Orig=SISR = 32 (in the unit of one pixel), \( RV_x = 60 \), \( RV_y = 20 \) (in the unit of a quarter pixel), which are drawn from Fig. 3. In ASIR, both \( TH_1 \) and \( TH_2 \) are assigned with 1, which is sufficient for the \( IIV \) containing most of the blocks needing inter-view prediction, as discussed in Subsection IV-B.2.

It can be drawn from Tab IV that the time saving is up to 25% for Janine1 applying only ASIR. ASIR also has a good performance when used to encode other 1024×768 sequences except Breakdances. The reason is that in the prediction process of encoding the sequences as Janine1, etc., only a few blocks use inter-view references in the end. For example, the proportions of blocks applying inter-view references for \( PV_1, POC \) (\( j = 0, 1, 2 \cdots, 12 \)) of Janine1 are \{90.13%, 10.13%, 15.41%, 16.52%, 11.53%, 9.36%, 22.86%, 8.98%, 10.99%, 18.09%, 9.58%, 16.06%, 90.06%\}, the values of which are smaller compared with those of Ballroom shown in Fig. 4. So ASIR is able to play a more important role in reducing the coding time. Additionally, we use ASRA to encode Breakdances, and the \( \Delta \text{PSNR} \) is -0.13 dB, unacceptable in the sense of quality, detailed discussed in Subsection V-B.

To reduce the encoding time is not the only target for a fast algorithm, whose impact on compression efficiency should also be taken into consideration. In Tab IV, the maximum reduced \( \Delta \text{PSNR} \) is only 0.03 dB, which can be ignored in a real system. Even for some sequences, such as Rena when using ASRA and ASIR jointly, its \( \Delta \text{PSNR} \) has a litter gain: 0.02 dB. In some cases, searching for matching blocks in the prediction process may fall into a local optimum, and our algorithms may overcome this shortcoming at some time and find better matching blocks. But this gain can also be ignored. It should be noted that when we use the fast algorithms ASRA and ASIR separately, the sum of the \( \Delta \text{time} \) is more than that reduced by jointly using these two algorithms, because ASIR reduces the number of inter-view references, which limits the range of application of ASRA.

Some results among those reported in [10]–[16] may outperform ours, but the methods may have limitations: the camera calibration information should be used, or they are only applied to I-P-P-P prediction structure, etc., detailed discussed in Section I. Our methods are only used to reduce the time of inter-view prediction, and some others have better results since they are used to reduce both intra-view and inter-view predictions.

Overall, the ultimate purpose of developing fast algorithms for MVC is to construct real-time multiview video systems where require low computational complexity and decent coding efficiency. The absolute time shown in Tab V is far away from constructing a real-time system, but the results calculated by Bjontegaard method and the relative time reduction shown in Tab IV indicate our algorithms may push on this proceeding.

VI. CONCLUSION

In this paper, we propose two fast algorithms for MVC, the first one (ASRA) tries to adjust the search range based on global disparity, and the second (ASIR) selects reference pictures for inter-view estimation when the blocks fall into a special region called \( IIV \) determined by previous estimation information. In ASRA, we use GDV to analyze the correlations between \( DV \) and \( PV \), and the statistical property of the absolute difference value of \( DV \) and \( GDV \) is applied in this method to dynamically adjust the search range. In ASIR, we first use the prediction information of the lower temporal layer in HBPS to reduce the number of inter-view reference pictures, and then apply the result of references selection after inter_16×16 MD to revise our method. The experimental results show that these two algorithms work well. Furthermore, algorithms in this paper can be integrated with others proposed in some literatures and substantially reduce the encoding complexity of MVC while maintaining the compression performance in accordance with the standard.

In the future, we intend to deeply analyze the influence of non-parallel camera setting on ASRA, and develop more robust algorithm using GDV suitable for all kinds of multiview videos. And another research point is to track the rotating objects more precisely and quickly in ASIR, further reduce the range of \( IIV \) and the coding time.

ACKNOWLEDGMENT

The authors would like to thank the National Natural Science Foundation of China (no. 60602033 & 60902063) and Natural Science Foundation of Tianjin (no. 07JCZDJC06800) for their supports on this work.

REFERENCES


Rong Pan received his B.S. and M.S. degrees in signal & and information processing from Tianjin University, China, in 2005. He is currently a Ph.D. candidate at Tianjin University. His research interests include multimedia processing and digital video coding.

Zheng-Xin Hou was born in Tianjin, China in 1945. He received the B.E. degree in electrical engineering from Peking University, China, in 1969 and the M.S. degree in signal & information processing from Tianjin University, China, in 1982. Now he is a professor and supervisor of Ph.D. students in the School of Electronic Information Engineering, Tianjin University, China. His research interests include digital filtering theory, signal processing theory, image & video processing, wavelet analysis and its applications, etc.

Yu Liu received his Ph.D. degree in signal & information processing from Tianjin University, China, in 2005. Now he is an associate professor in the School of Electronic Information Engineering, Tianjin University. His research interests include image and video compression, HDTV and multimedia systems.