Fast Encoding Techniques for Multiview Video Coding

S. Khattak[a], R. Hamzaoui[a,*], S. Ahmad[a], P. Frossard[b]

[a] *Centre for Electronic and Communications Engineering, De Montfort University, United Kingdom*
[b] *Signal Processing Laboratory - LTS4, Ecole Polytechnique Féderale de Lausanne, Switzerland*

Abstract

Multiview Video Coding (MVC) is a technique that permits efficient compression of multiview video. MVC uses variable block size motion and disparity estimation for block matching. This requires an exhaustive search process that involves all possible macroblock partition sizes. We analyze the time complexity of MVC and the methods that have been proposed to speed up motion and disparity estimation. We then propose two new methods: Previous Disparity Vector Disparity Estimation (PDV-DE) and Stereo-Motion Consistency Constraint Motion and Disparity Estimation (SMCC-MDE). PDV-DE exploits the correlation between temporal levels and disparity vectors to speed up the disparity estimation process while SMCC-MDE exploits the geometrical relationship of consecutive frame pairs to speed up motion and disparity estimation. We build a complete low complexity MVC encoding solution that combines our two methods with complementary previous methods to speed up motion and disparity search. We evaluate the complexity of our solution in terms of encoding time and number of search points. Our experimental results show that our solution can reduce the encoding time and number of search points of the standard MVC implementation (JMVM 6.0) using the fast TZ search mode by up to 93.7% and 96.9%, respectively, with negligible degradation in the rate-distortion performance. Compared to the best published results, this is an improvement of up to 11% and 7%, respectively.

*Corresponding author. Tel.: 44 116 207 8096; Fax: +44 116 207 8159
Email addresses: shadan.khattak@myemail.dmu.ac.uk (S. Khattak), rhamzaoui@dmu.ac.uk (R. Hamzaoui), sahmad@dmu.ac.uk (S. Ahmad), pascal.frossard@epfl.ch (P. Frossard)

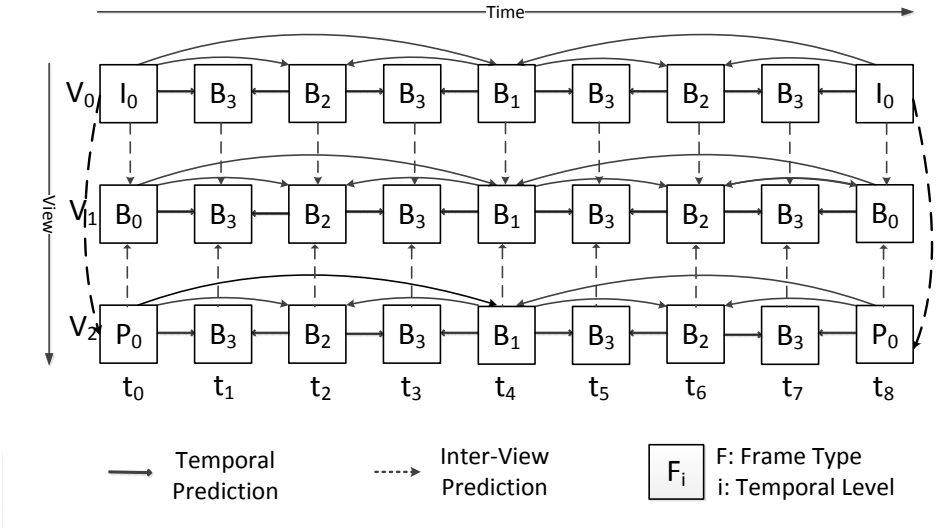


Figure 1: Typical MVC prediction structure. V_0, V_1, and V_2 represent three views while $t_0, t_1, \ldots, t_8$ represent nine successive frames. In each view, the first frame of the Group of Pictures (GOP) is said to be at Temporal Level 0 (TL0). All the frames that use frames at TL0 as references belong to Temporal Level 1 (TL1). Similarly, the frames that use frames at TL1 as references belong to Temporal Level 2 (TL2), etc.

Keywords: Multiview Video Coding, Disparity estimation, Motion estimation.

1. Introduction

Multiview Video is a technology that uses multiple cameras to simultaneously capture a scene from different view points. Multiview Video is used in applications such as 3D Television and Free View-point Television [1]. While Multiview Video gives a richer viewing experience than conventional video, it produces a huge amount of data. However, since the data from all cameras relates to the same scene, it is highly redundant. This has led to the development of Multiview Video Coding (MVC) [2], the multiview extension of the latest video coding standard, H.264/AVC [3]. In MVC, reference frames for block matching are taken from neighboring views (Disparity Estimation) as well as across the temporal axis (Motion Estimation). A typical prediction structure [4] of MVC is presented in Fig. 1.

MVC uses variable block size motion and disparity estimation, which requires an exhaustive search for motion and disparity vectors using all the available block sizes. Unfortunately, this makes the MVC encoder very complex. Therefore, reducing the time complexity of the MVC encoder is very important.

Complexity reduction can be achieved in several ways, e.g., by reducing the number of candidate modes, the number of reference frames, the number of directions for prediction, or the search range. Such reductions must, however, be done with smallest penalty on rate-distortion (RD) performance. State-of-the-art methods address specific parts of the problem, but, to the best of our knowledge, there is no global solution yet. Moreover, most of the published work relies only on CPU time saving as the evaluation metric for low complexity encoding methods. This may not necessarily reflect the efficiency of the methods themselves, as it is dependent on the particular implementation and the test platform.

The contributions of this paper are as follows. We define four levels of complexity in the MVC encoder and identify the best previous fast encoding techniques at each level. We then combine these techniques in a unique framework in which savings in complexity add up. We observe that the performance of our combination largely depends on motion and disparity in the video sequence as well as the encoding bitrate. The complexity savings are larger for low motion content. They are also larger at low bitrates than at high bitrates. To improve the performance for high motion content and high bitrates, we propose two new fast encoding techniques: (i) Previous Disparity Vector Disparity Estimation (PDV-DE), which exploits the correlation between temporal levels and disparity vectors and (ii) Stereo-Motion Consistency Constraint Motion and Disparity Estimation (SMCC-MDE), which exploits the geometrical relationship of consecutive frames in the multiview video sequence. Our contribution builds on our previous paper [5] in which PDV-DE was introduced.

We analyze the performance of our global solution using two metrics: CPU time and number of search points. Experimental results show that our solution can save up to 93.7% in encoding time and 96.9% in number of search points compared to JMVM 6.0 [6] using the fast TZ search mode [7] without significant penalty in terms of bitrate or distortion. This is an improvement of over 11% and 7%, respectively, compared to the best published method [8]. This method uses inter-view mode and motion vector correlations to reduce the complexity of the mode decision, the reference frame selection, and the

Table 1: Complexity levels in Multiview Video Coding.

Level 1: Mode	Level 2: Prediction Direction	Level 3: Reference Frame	Level 4: Block Matching
1. SKIP			
2. INTER16X16	1. Forward	1. ME	
3. INTER16X8			
4. INTER8X16			
5. INTER8X8	2. Backward		Search for the best rate-distortion match
6. INTER8X4			
7. INTER4X8		2. DE	
8. INTER4X4	3. Bi-directional		
9. INTRA16, INTRA8, INTRA4			

block matching process.

The remainder of the paper is organized as follows. In Section 2, we give an overview of the encoding complexity framework for MVC and identify its main bottlenecks. We then discuss the solutions that have been proposed to address them. In Section 3, we present our low-complexity solution, which combines PDV-DE and SMCC-MDE with state-of-the-art methods. Section 4 contains our experimental results. Section 5 provides our conclusions and suggests future research directions.

2. MVC encoding complexity

To analyze the complexity of MVC encoding, we focus on JMVM, which is the reference software for MVC. For efficient compression, JMVM offers multiple ways of encoding a macroblock. These include a choice of different macroblock partition sizes, prediction directions, reference frames and search window sizes. In the standard implementation, all the possible options are exhaustively checked, and the ones resulting in the lowest rate-distortion cost are finally selected. We identify the following four levels of complexity in JMVM (Table 1).

- Level 1 - **Mode Selection**: Several modes are checked in sequence to find the best rate-distortion match for the current macroblock. These modes are: (i) SKIP, (ii) INTER16x16, (iii) INTER16x8, (iv) INTER8x16, (v) INTER8x8, (vi) INTER8x4, (vii) INTER4x8, (viii) INTER4x4, (ix) INTRA16, (x) INTRA8, and (xi) INTRA4. When a macroblock is encoded using the SKIP mode, no motion or residual data is transmitted. The macroblock is reconstructed with the help of motion

vectors from the spatially neighboring macroblocks. In INTER16x16 mode, a single motion/disparity vector along with the residual data is transmitted. In INTER16x8 and INTER8x16 modes, a macroblock is partitioned into two partitions of sizes 16x8 and 8x16, respectively, and for each partition, a separate motion/disparity vector is transmitted. In INTER8x8 mode, a macroblock is partitioned into four partitions of size 8x8 and four motion/disparity vectors are transmitted. Each 8x8 size partition can be further divided into three possible sub-macroblock partitions of sizes 8x4, 4x8, and 4x4. For each sub-macroblock partition, a separate motion/disparity vector is transmitted.

- Level 2 - **Prediction Direction Selection**: For each INTER mode in Level 1, a best match is sought in: (i) past frames (forward prediction), (ii) future frames (backward prediction), and (iii) a combination of one past and one future frame (bi-directional prediction).

- Level 3 - **Reference Frame Selection**: For each prediction direction selected in Level 2, the JMVM reference software searches reference frames from different views to find the best match through block-matching. These frames can be: (i) from the same view (using ME), (ii) from the two neighboring views (using DE).

- Level 4 - **Block Matching**: For each reference frame, a best match is sought in a search window of size $n \times n$, where n denotes the number of pixels. For good compression efficiency, usually a large window size is used. An important element in the search process is the determination of the motion vector predictor. The motion vector predictor determines the starting point for the search process. The more accurate the predictor is, the more probable it is to find the best match in a smaller search area.

Several methods have been proposed to reduce the encoding complexity of MVC. We briefly review them below.

Mode Selection: A fast mode selection method exploiting the correlation between the modes of neighboring views is proposed in [9]. To predict the mode of the current macroblock, the modes of the corresponding macroblock in the neighboring view and its eight spatially neighboring macroblocks are taken into consideration. Weights are assigned to each mode and macroblocks are classified according to the average weight. If the average is

less than 0.125, then the current macroblock is called *Simple* and only SKIP mode and INTER16x16 modes are considered. If the average is greater than 0.125 and smaller than 0.25, then the current macroblock is called *Normal* and additionally, INTER16x8 and 8x16 are also considered. Finally, if the average weight is greater than 0.25, the macroblock is called *Complex* and all modes are considered. Zeng, Ma, and Cai [10] extend the work of [9] by increasing the number of macroblock types to five.

A fast mode decision method based on rate-distortion costs is presented in [11]. It uses inter-view rate-distortion cost correlation of optimal modes to reduce the number of candidate modes in the current view.

Early detection of SKIP mode reduces the complexity of the encoder significantly as macroblocks encoded in SKIP mode do not require block matching. An early SKIP mode detection method is proposed in [12]. The detection is based on the analysis of SKIP mode decisions of the nine corresponding neighbors in the neighboring right view.

Prediction Direction Selection: In JMVM, most of the pictures are of B type. For macroblocks in B pictures, motion and disparity estimation are done using forward, backward and bi-directional prediction. Zhang et al. [13] observe that the prediction direction that results in the lowest rate-distortion cost for INTER16x16 is also the one that results in the lowest rate-distortion cost for the other INTER modes. So they propose to save encoding time by selecting for all modes the prediction direction that results in the lowest rate-distortion cost for INTER16x16.

Reference Frame Selection: Zhang et al. [13] restrict block matching to the reference frame that gives lowest rate-distortion cost for INTER16x16. Another fast reference frame selection method is presented in [14]. Frames are divided into regions with homogeneous motion (*homogeneous* regions) and regions with complex motion (*complex* regions). The classification is based on forward motion vectors for 4x4 pixel blocks in four view-neighboring macroblocks (i.e., corresponding macroblock in the neighboring right view together with its left, upper, and upper-left macroblocks). The authors observe that in homogeneous regions inter-view prediction is rarely used and thus propose to disable DE in those regions.

Another adaptive disparity estimation method is proposed by Shen et al. [8]. This method enhances the method of [14] by defining a third class of regions, namely *medium homogeneous* regions. Moreover, the classification is refined by involving all nine view-neighboring macroblocks. DE is disabled in homogeneous regions, as well as in medium regions if the rate-distortion

cost of the motion vector predictor (initial prediction of the motion vector) is smaller than that of the disparity vector predictor (initial prediction of the disparity vector).

While the method in [13] reduces the number of reference frames for smaller macroblock partitions to one in each prediction direction, all the reference frames are still checked for the INTER16x16 mode. Similarly, the method in [8] reduces the number of reference frames in each prediction direction to one in homogeneous regions. But in complex regions, two reference frames are checked in each direction.

Block Matching: Shen et al. [8] observe that the best block (for ME and DE) is usually found close to the current macroblock for homogeneous regions, far away from the current macroblock for complex regions, and somewhere in between for medium homogeneous regions. Therefore, the search range for a macroblock is adjusted according to the region type. For homogeneous regions, the search range is limited to a quarter of the full search range. For medium homogeneous regions, it is limited to half the full search range. For complex regions, full search is used. The spatio-temporal correlation of the disparity fields is studied in [15]. The authors find that the search range for disparity estimation can be reduced if multiple candidates are considered as search centers. However, they do not exploit the correlations between disparity vectors at different temporal levels, which can be used to further reduce the search ranges. Similarly, they do not study the effect of the type of macroblock on this correlation.

Deng et al. [16, 17] use Stereo-Motion Consistency Constraint (SMCC) to reduce the complexity of motion and disparity estimation for stereo video coding. They use an iterative search strategy in which SMCC geometry is exploited to get base motion and disparity vectors. To reduce the effect of macroblock boundary mismatches on the performance of their algorithm, they extend the search around the base motion and disparity vectors iteratively. Because base motion and disparity vectors are not very accurate, a large search region is required around them during the successive iterations.

Combinations: Finally, some combinations of methods at different levels of the encoding scheme have been proposed. Zhang et al. [18] combine a rate-distortion cost threshold fast mode decision technique with the multiple reference frame selection method in [13]. While this algorithm speeds up the mode decision and reference frame selection processes, redundancies still exist in prediction direction selection and block-matching. Shafique, Zatt, and Henkel [19] take into consideration texture classification and rate-distortion

cost of a macroblock to predict the mode and prediction direction. However, the reference frame selection and block matching steps are not modified.

3. Proposed framework

To achieve maximum reduction in encoding complexity, it is important to simplify the processes involved at all levels of the encoding process. To the best of our knowledge, there is no method that simultaneously reduces the complexities present at all these levels. So the motivation for our work is to reduce the complexity at all levels and thus present a complete low-complexity solution for MVC encoding. To do this, we first combine into a novel framework state-of-the-art methods ([12], [13] and [8]) that target different levels of the encoding. We notice that the gains add up. We then observe that, while our combination speeds up the overall MVC encoding process, its performance at high bitrates and for content with high motion can still be improved. Thus, we propose two new complexity reduction techniques. The first one (PDV-DE) reduces the search range by exploiting the correlation of the disparity fields of successive frames at different temporal levels. The second one (SMCC-MDE) exploits the geometric constraint between motion and disparity vectors of two consecutive stereo pairs to reduce the area where a potential best rate-distortion match lies. We detail our techniques in the next two sub-sections and summarize the complete framework in Section 3.3.

3.1. Previous Disparity Vector Disparity Estimation (PDV-DE)

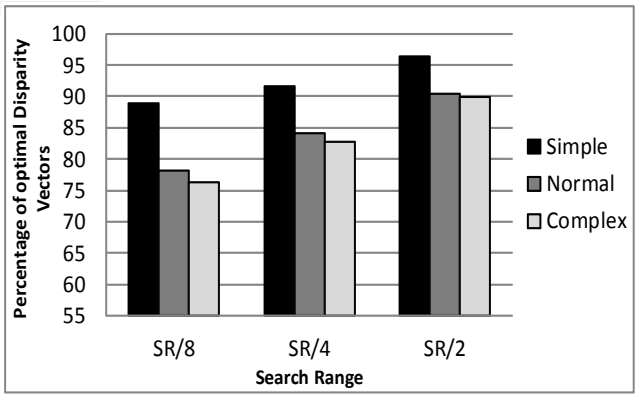
While DE consumes as much time as ME, the probability that it is used for prediction is generally low [14, 8]. In this section, we propose to adjust the search range for DE according to the temporal level of the frame and the type of the macroblock (simple, normal, or complex).

The search for the best match starts in the search center. If the search center is close to the best match, we may reduce the size of the search range and still find the best match. Because one does not know a priori where the best match for the current macroblock will be found, JMVM uses median prediction. In median prediction, the median of the vectors of the top, top-right, and left macroblocks are used as the search center. The same procedure is used for both ME and DE. However, the nature of disparity is different from that of motion. Indeed, even in the presence of motion, the

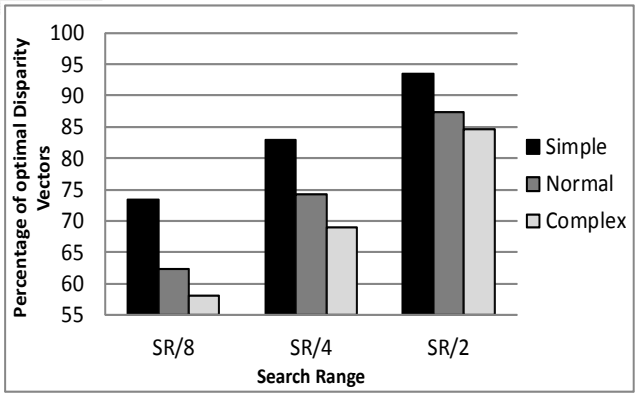
source of disparity (i.e., the camera arrangement) is usually fixed. Thus, disparity is not as difficult to predict as motion. Moreover, the disparity fields of successive frames are highly correlated [15]. Thus, if the search process is started from the position identified by the disparity vector of the corresponding macroblock in the previous frame, it is expected that the best match will be found very early in the process.

To validate this assumption, we modify the procedure for finding the search center. We use the disparity vector of the corresponding macroblock in the temporally preceding frame as the search center, instead of the median prediction. We call this vector Previous Disparity Vector (PDV). The initial search range is set to 64. Then, we determine the proportion of macroblocks that find their best match in various search ranges: 1/8th of the initial search range (SR/8), 1/4th of the initial search range (SR/4), and half of the initial search range (SR/2). We note that the way the best match is spread across the search range depends on the temporal level of the frame as well as on the macroblock type. The number of temporal levels depends on the length of a GOP. A GOP of length l has $\lceil \log_2(l) \rceil + 1$ temporal levels. In our experiments, the GOP length is 16, so the highest temporal level is 4 (TL4), while the second highest is 3 (TL3). For frames at higher temporal levels, the best match is usually found in a smaller area than for those at lower temporal levels. For example, for frames at TL4, the best match is found in a smaller area than for those at TL3 (Fig. 2). Also for simple macroblocks, the best match is found in a smaller area than for normal and complex macroblocks. This indicates that if the previous disparity vector is used as the search center, the search range can be reduced adaptively according to both temporal level and macroblock type.

Based on the observations and motivations in this subsection, we formulate our new search strategy of disparity estimation, which we call PDV-DE. During disparity estimation, the search center is set to PDV (Fig. 3) and two conditions are checked: (i) Does the frame belong to TL3 or TL4? (ii) Is the macroblock simple, normal or complex? If the frame belongs to TL4 and the macroblock is of type 'simple', the search range is reduced to 1/8th of the initial search range. At the same temporal level, the search range for macroblocks of type 'normal' is reduced to 1/4th of the initial search range, and for macroblocks of type 'complex', it is reduced to half the initial search range. Since at lower temporal levels, the correlation between disparity vectors decreases, a slightly different search strategy is used to maintain similar rate-distortion performance to that of JMVM. So at TL3, if the macroblock is



(a) Temporal Level 4



(b) Temporal Level 3

Figure 2: Percentage of macroblocks for which the best disparity vectors are found if Previous Disparity Vector (PDV) is used as the search center. Results are shown for TL4, TL3, and various search ranges (SR). The percentages represent averages for the Ballroom and Exit sequences over quantization parameter (QP) values 24, 28, 32, and 36.

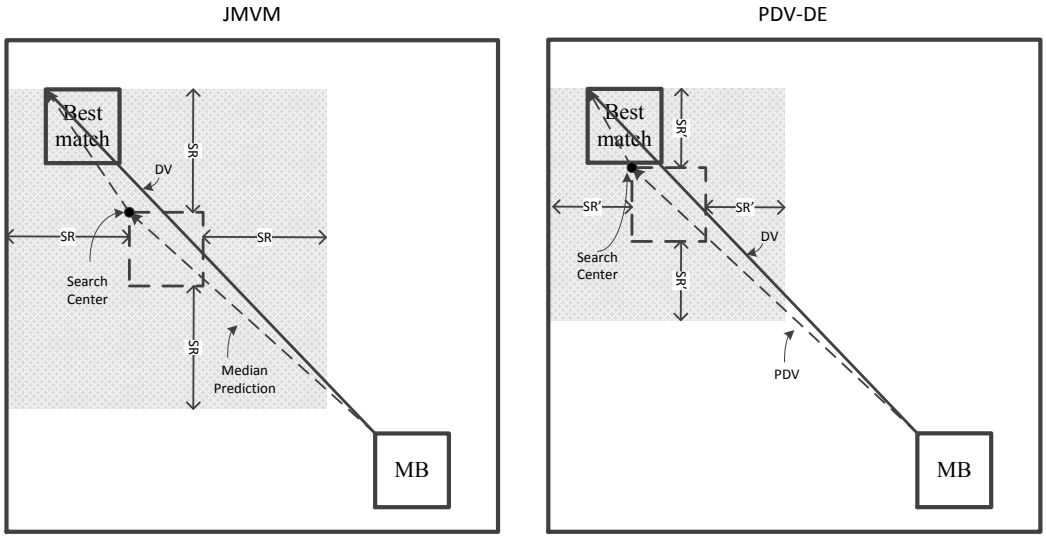


Figure 3: Reducing the search range in PDV-DE. The search centre is set to the Previous Disparity Vector (PDV) and the search range (SR) is reduced to SR'.

of type 'simple', the search range is reduced to a quarter of the initial search range, while for 'normal' macroblocks, it is reduced to half the search range. The search range is not reduced for 'complex' macroblocks. The complete search strategy of PDV-DE is shown in Fig. 4.

3.2. Stereo Motion Consistency Constraint Motion and Disparity Estimation (SMCC-MDE)

Stereo Motion Consistency Constraint (SMCC) is a geometrical constraint between the motion and disparity fields of two stereo pairs of video [20]. It is a pixel-based method where vectors are associated with pixels and denote the difference between the coordinates of corresponding pixels in different frames. SMCC is used to speed up the pixel matching process by providing a prediction for the optimal motion and disparity vectors.

In this section, we extend SMCC to block-based MVC. Fig. 5 and Fig. 6 illustrate our method. Four frames ($F_{1,t}, F_{0,t}, F_{1,t-1}$, and $F_{0,t-1}$) from two neighboring views (V_0, V_1) and two consecutive time instances ($t-1$: previous, t: current) are considered. The goal is to predict the motion and disparity vectors $MV_{1,t}$ and $DV_{1,t}$ for the current macroblock (MB).

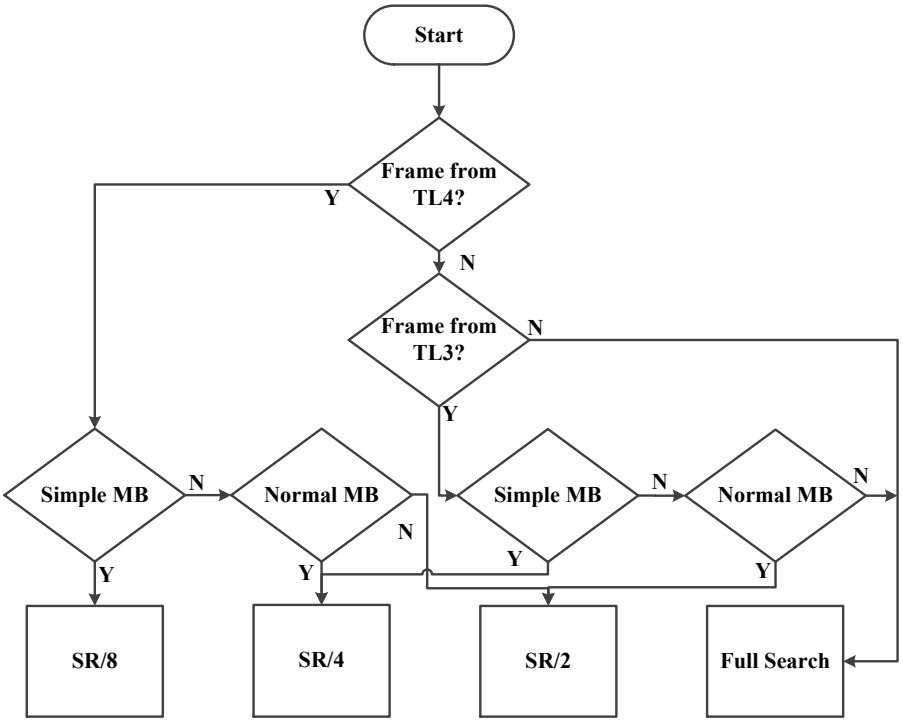


Figure 4: Disparity Vector Search in PDV-DE. Here TL = Temporal Level, SR = Search Range, and MB = Macroblock.

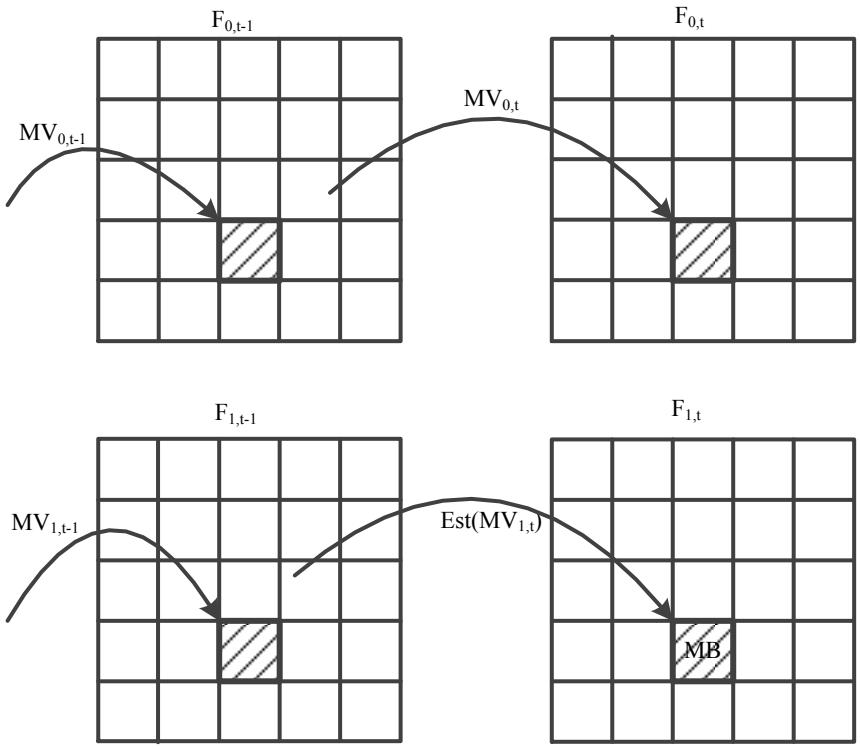


Figure 5: Estimation of the Motion Vector for macroblock MB.

We define $MV_{0,t}$ as the motion vector of the corresponding macroblock in $F_{0,t}$, $MV_{0,t-1}$ as the motion vector of the corresponding macroblock in $F_{0,t-1}$, and $MV_{1,t-1}$ as the motion vector of the corresponding macroblock in $F_{1,t-1}$ (Fig. 5). Then we exploit the correlation between the motion fields of neighboring views and obtain an estimate $Est(MV_{1,t})$ of the motion vector $MV_{1,t}$ as

$$Est(MV_{1,t}) = MV_{1,t-1} + MV_{0,t} - MV_{0,t-1} \tag{1}$$

To find an estimate of $DV_{1,t}$, we define $B_{1,t-1}$ as the macroblock with maximum overlap with the optimal match in $F_{1,t-1}$ for the current macroblock.

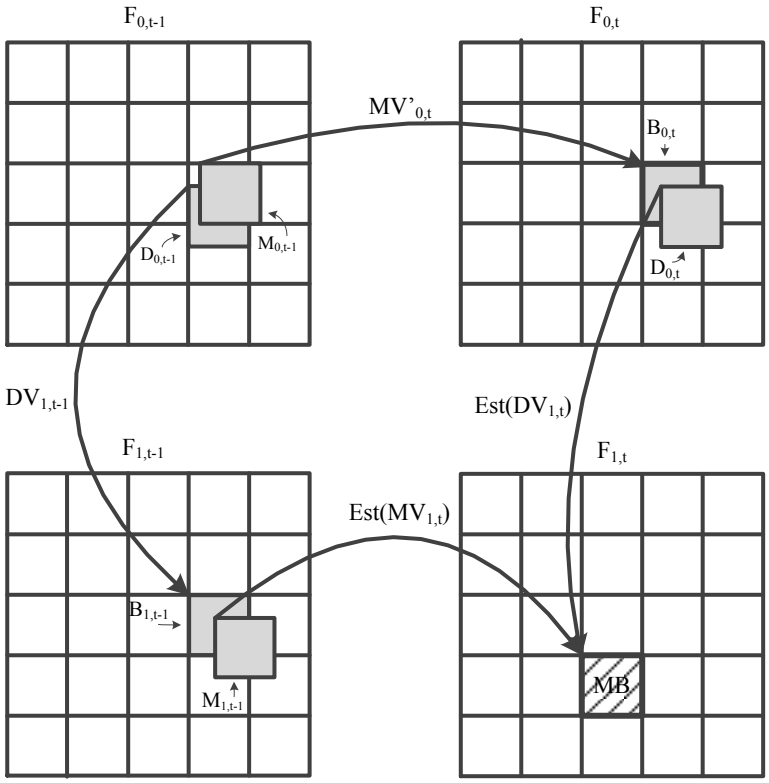


Figure 6: Estimation of the Disparity Vector for macroblock MB.

Then we use $DV_{1,t-1}$, the disparity vector of $B_{1,t-1}$, to obtain a macroblock $D_{0,t-1}$ in frame $F_{0,t-1}$. Next, we find the macroblock $B_{0,t}$ in frame $F_{0,t}$ whose motion vector $MV'_{0,t}$ is associated with the macroblock $M_{0,t-1}$ in $F_{0,t-1}$ with maximum overlap with $D_{0,t-1}$ (Fig. 6).

If the motion and disparity compensated macroblocks in frames $F_{0,t}$, $F_{1,t-1}$, and $F_{0,t-1}$ are perfectly aligned on macroblock boundaries, then, by analogy with pixel-based stereo motion consistency, we have

$$MV'_{0,t} + Est(DV_{1,t}) = Est(MV_{1,t}) + DV_{1,t-1} \tag{2}$$

Thus, given $Est(MV_{1,t})$, $DV_{1,t-1}$, and $MV'_{0,t}$, one can use (2) to find an estimate of $DV_{1,t}$.

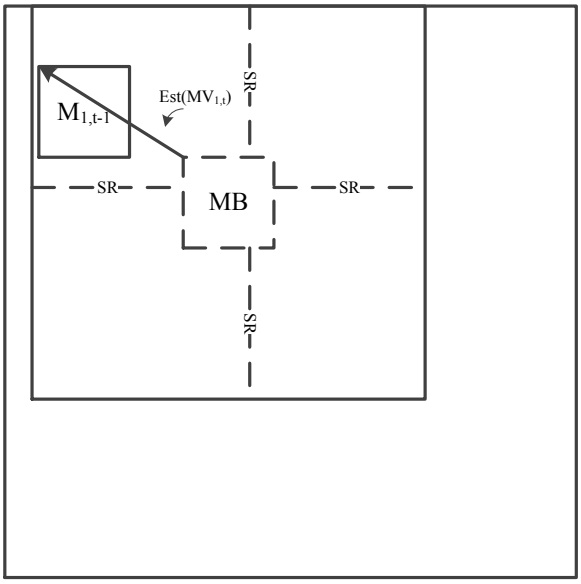


Figure 7: Search for the optimal motion vector. The search range is set to $SR = \lceil \max(|Est(MV_{1,t}x)|, |Est(MV_{1,t}y)|) \rceil$.

The next step is to refine the estimated motion vector (Fig. 7). This is done by setting the search range to

$$SR = \lceil \max(|Est(MV_{1,t}x)|, |Est(MV_{1,t}y)|) \rceil$$

We apply a similar procedure to refine the estimated disparity vector.

3.3. Complete Framework

In this section, we present a framework that enables us to reduce the encoding complexity at all levels. Our framework combines the methods in [12], [13], and [8] with the two new methods described in Sections 3.1 and 3.2. A flowchart is shown in Fig. 8.

1. **Input:** Three views V_0, V_1 and V_2. A macroblock MB in V_1 to encode.
2. **Pre-Processing:** Encode V_0 and V_2 using JMVM. Find the macroblock in view V_2 defined by the Global Disparity Vector (GDV) [21]. GDV represents the average disparity in MB units ($\pm$ 16 integer pixel units) between the current frame and the frame of a reference view.

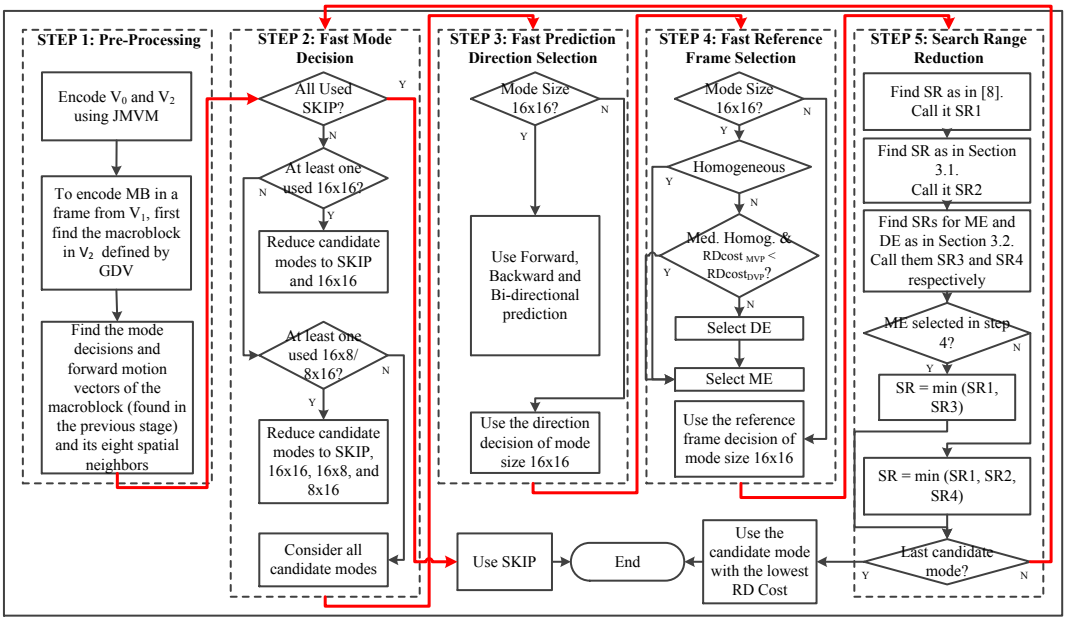


Figure 8: Block diagram of Complete Low-Complexity Multiview Video Coding (CLCMVC).

Obtain the mode size decisions and forward motion vectors (at 4x4 block level) of this macroblock and its eight spatial neighbors.

3. **Fast Mode Decision:**

   (i) If the nine macroblocks in the view neighborhood use the SKIP mode, then encode MB with the SKIP mode.

   (ii) If at least one of the nine macroblocks in the view neighborhood is encoded using the 16x16 mode, then check the SKIP and 16x16 modes.

   (iii) If at least one of the nine macroblocks in the view neighborhood is encoded using the 16x8 or 8x16 mode, then check the SKIP, 16x16, 16x8, and 8x16 modes.

   (iv) In all other cases, check all the mode sizes.

4. **Fast Prediction Direction Selection:**

   (i) If the mode size is 16x16, search using Forward, Backward and Bi-directional Prediction.

   (ii) For all other mode sizes, use the prediction direction decision of mode size 16x16.

5. **Fast Reference Frame Selection:**
   (i) If the mode size is 16x16 and
   (a) motion is homogeneous: disable DE and search the remaining (temporal) reference frame(s).
   (b) motion is medium homogeneous and the RD cost of Motion Vector Predictor (MVP) is less than that of Disparity Vector Predictor (DVP), disable DE and search the remaining (temporal) reference frames. MVP and DVP are the initial (basic) motion and disparity vectors during motion and disparity estimation.
   (c) in all other cases, search all the reference frames.
   (ii) For all other mode sizes, use the reference frame decision of mode size 16x16.
6. **Search Range Reduction:** In the identified reference frames, use the following criteria to determine the search range (SR).
   (i) Motion homogeneity: Use the search strategy proposed in [8] to determine the search range and call it SR1.
   (ii) Temporal level of the frame and mode complexity of the MB: Use PDV-DE (Section 3.1) to determine the search range and call it SR2.
   (iii) Stereo motion consistency constraint: Use SMCC-MDE (Section 3.2) to determine the search range and call it SR3 for ME and SR4 for DE.
   (iv) If both the reference frame and the current frame belong to the same view (Intra-View), then SR = min (SR1, SR3). Otherwise (Inter-View), SR = min (SR1, SR2, SR4)

4. Results

4.1. Setup

We extensively tested all algorithms on the standard sequences Ballroom, Exit, Vassar and Race1 as recommended by the Joint Video Team (JVT) [22]. Ballroom and Race1 are examples of sequences with high motion content while Vassar and Exit represent sequences with low motion content. Ballroom and Exit are also representative of sequences with large disparities. In our experiments, three views of the test sequences are used for simulations. The second view was encoded using the proposed algorithm, and the first and the third views were used as reference views. For all sequences, the GOP size was set to 16, and the maximum search range was $\pm$ 64. Results are

presented for QP values 20, 24, 28, 32, and 36. The simulations were run on a machine with Intel Core i5 dual core 2.67 GHz CPU and 4 GB RAM.

We considered two indicators of complexity reduction: CPU time saving and number of search points saving compared to the fast TZ search method [7] in JMVM. The number of search points is the number of times the rate-distortion cost is checked during motion and disparity estimation. The following formulas are used to calculate the percentage time saving, the percentage number of search points saving, the percentage additional bitrate and the difference in Peak Signal-to-Noise Ratio (PSNR), respectively:

$$\Delta T(\%) = \frac{T_{JMVM} - T_{METHOD}}{T_{JMVM}} \times 100$$

$$\Delta N(\%) = \frac{N_{JMVM} - N_{METHOD}}{N_{JMVM}} \times 100$$

$$\Delta B(\%) = \frac{B_{METHOD} - B_{JMVM}}{B_{JMVM}} \times 100$$

$$\Delta PSNR(dB) = PSNR_{METHOD} - PSNR_{JMVM}$$

Here $T_{JMVM}, N_{JMVM}, B_{JMVM}$, and $PSNR_{JMVM}$ represent the encoding time, the number of search points, the bitrate, and the PSNR obtained using the JMVM algorithm, while T_{METHOD}, N_{METHOD}, B_{METHOD}, and $PSNR_{METHOD}$ represent the encoding time, the number of search points, the bitrate, and the PSNR obtained using the proposed method. The number of search points is calculated by dividing the number of search points for the whole sequence by the number of frames in the sequence. The result is further divided by the number of macroblocks in the frame to obtain the average number of search points per macroblock.

Since JMVM does not use inter-view prediction for the first and third views (except for the first frame of the third view), we calculate ΔT and ΔN only for the second view (V_1 in Fig. 1).

4.2. PDV-DE

Table 2 shows the results for PDV-DE. On average, PDV-DE achieves a time saving of over 35% compared to the TZ search mode of JMVM 6.0 while maintaining similar rate-distortion performance. The time saving does not vary much for different QP values. This is because PDV-DE exploits frames at the highest and second highest temporal levels and, for the same GOP size, the number of such frames is not affected by the QP value. Table 2 also

Table 2: Performance of PDV-DE compared to JMVM 6.0.

	QP	Δ PSNR	ΔB	ΔT	ΔN
Ballroom					
	20	0.01	0.12	35.29	38.66
	24	0.01	-0.19	35.52	39.15
	28	0.00	-0.48	34.63	38.20
	32	0.01	-0.62	32.95	37.69
	36	0.04	-0.12	32.08	36.88
	Avg.	0.01	-0.26	34.09	38.12
Exit					
	20	0.02	0.40	35.50	37.55
	24	0.02	0.48	37.59	39.90
	28	0.01	0.41	36.95	39.71
	32	0.02	0.28	36.09	39.04
	36	0.05	0.47	36.24	39.44
	Avg.	0.03	0.41	36.47	39.13

shows that the time saving for the Exit sequence is slightly larger than that of Ballroom. This is because for frames at the same temporal level, PDV-DE reduces the search range primarily for 'simple' macroblocks, and the number of such macroblocks increases when there is less motion content.

4.3. SMCC-MDE

The results of SMCC-MDE are presented in Table 3. SMCC-MDE achieves a time saving of over 41% on average. The time saving increases with decreasing QP value. For example, for both sequences, for a QP value of 20, at least about 7% additional time saving is achieved compared to the time saving at a QP value of 36. One reason for this is that the algorithm uses estimated motion vectors to set the search range for ME. These estimated motion vectors depend on the difference between motion vectors of consecutive frames (see (1)). With fine quantization (low QP values), the difference between motion vectors of consecutive frames is small while with coarse quantization (high QP values), this difference is large. The rate-distortion performance of the

Table 3: Performance of SMCC-MDE compared to JMVM 6.0.

	QP	Δ PSNR	ΔB	ΔT	ΔN
Ballroom					
	20	0.00	0.30	46.42	48.19
	24	0.00	0.53	46.09	47.66
	28	−0.01	0.81	45.28	48.48
	32	−0.01	0.74	43.87	47.96
	36	0.00	0.77	38.20	45.57
	Avg.	0.00	0.63	43.97	47.57
Exit					
	20	−0.01	0.04	42.25	46.62
	24	0.00	0.30	40.64	40.88
	28	−0.01	0.57	40.39	40.67
	32	−0.02	0.05	39.00	39.68
	36	0.01	0.31	35.73	37.73
	Avg.	0.00	0.25	39.60	41.12

algorithm is very similar to that of JMVM. The slight increase in bitrate is due to the more frequent use of small mode sizes, which increases the number of motion vectors.

4.4. CLCMVC

In this section, we present detailed results for our complete solution, which we call Complete Low-Complexity MVC (CLCMVC). We also show, step by step, how the addition of each constituent method of CLCMVC reduces the number of search points and thus increases the overall time saving.

Table 4 shows the savings in time and number of search points for the four test sequences. The results for 'Exit' sequence show that JMVM searches on average 13,900 search points for different QP values, before a block is selected. By reducing the number of candidate modes as in [9], the number of search points can be reduced to about 1,100. This translates into an average time saving of around 68%. When the candidate modes reduction and selective disparity estimation methods are combined as in [14], the average time saving

Table 4: Comparison of fast MVC encoding techniques. S denotes the average number of search points per macroblock. ΔN and ΔT denote the percentage number of search points saving and the percentage time saving compared to JMVM 6.0.

Exit															
Method/QP	36			32			28			24			20		
	S	ΔN	ΔT	S	ΔN	ΔT	S	ΔN	ΔT	S	ΔN	ΔT	S	ΔN	ΔT
JMVM	13906			13911			13901			13893			13882		
[9]	1101	92.08	76.92	1434	89.69	73.93	1751	87.40	71.48	2292	83.50	67.47	4284	69.14	53.73
[14]	555	96.01	87.96	647	95.35	85.67	735	94.71	85.2	931	93.30	83	1913	86.22	77.5
[8]	452	96.75	89.34	539	96.13	87.04	612	95.60	85.94	773	94.44	83.91	1458	89.5	79.5
[8]+[12]	314	97.74	90.31	466	96.65	88.41	547	96.07	86.44	731	94.74	84.45	1448	89.57	79.46
[8]+[12]+[13]	162	98.84	94.18	232	98.33	93.95	266	98.09	93.44	354	97.45	92.41	691	95.02	89.86
[8]+[12]+[13] + PDV-DE	113	99.19	95.38	165	98.81	95.11	205	98.53	94.6	273	98.03	93.69	514	96.30	91.46
CLCMVC	109	99.22	95.42	162	98.84	95.37	189	98.64	95.05	246	98.23	94.33	439	96.84	92.27

Ballroom															
Method/QP	36			32			28			24			20		
	S	ΔN	ΔT	S	ΔN	ΔT	S	ΔN	ΔT	S	ΔN	ΔT	S	ΔN	ΔT
JMVM	13908			13900			13892			13882			13868		
[9]	2141	84.61	67.16	2734	80.33	64.58	3387	75.62	61.31	4108	70.41	56.33	5202	62.49	49.77
[14]	1113	92.00	78.98	1297	90.67	77.28	1513	89.11	76.27	1807	86.98	74.83	2384	82.81	72.93
[8]	1023	92.64	80.72	1208	91.31	78.73	1378	90.08	77.99	1626	88.29	76.25	2091	84.92	74.56
[8]+[12]	947	93.19	82.04	1132	91.86	79.38	1322	90.48	78.05	1585	88.58	76.9	2080	85.00	74.3
[8]+[12]+[13]	391	97.19	91.31	454	96.73	91.03	522	96.24	90.74	611	95.60	90.28	812	94.14	88.06
[8]+[12]+[13] + PDV-DE	305	97.81	93.15	359	97.42	92.90	473	96.60	92.22	562	95.95	90.92	651	95.31	90.01
CLCMVC	287	97.94	93.61	341	97.55	93.45	400	97.12	93.11	448	96.77	92.63	546	96.06	91.62

Vassar															
Method/QP	36			32			28			24			20		
	S	ΔN	ΔT	S	ΔN	ΔT	S	ΔN	ΔT	S	ΔN	ΔT	S	ΔN	ΔT
JMVM	13917			13916			13912			13905			13893		
[9]	664	95.23	80.46	914	93.43	76.88	1164	91.63	76.36	1741	87.42	73.08	4208	69.71	58.59
[14]	395	97.16	90.41	502	96.39	89.57	577	95.85	88.94	823	94.08	87.73	2122	84.73	81.51
[8]	308	97.79	92.26	412	97.04	91.51	480	96.55	90.75	663	95.23	89.41	1699	87.77	83.79
[8]+[12]	143	98.97	93.55	227	98.37	92.62	231	98.34	91.64	621	95.53	89.94	1690	87.84	83.81
[8]+[12]+[13]	111	99.20	94.79	178	98.72	94.19	187	98.66	94.37	325	97.66	93.38	738	94.69	91.07
[8]+[12]+[13] + PDV-DE	59	99.58	95.70	123	99.12	94.78	118	99.15	94.91	234	98.32	93.96	501	96.39	91.44
CLCMVC	36	99.74	95.84	57	99.59	95.58	47	99.66	95.26	201	98.55	94.22	376	97.29	92.06

Race1															
Method/QP	36			32			28			24			20		
	S	ΔN	ΔT	S	ΔN	ΔT	S	ΔN	ΔT	S	ΔN	ΔT	S	ΔN	ΔT
JMVM	13833			13806			13779			13772			13753		
[9]	1761	87.27	71.96	2177	84.23	69.35	2701	80.40	67.20	3365	75.57	62.75	4428	67.80	56.40
[14]	1064	92.31	81.36	1234	91.06	79.59	1448	89.49	78.42	1708	87.60	76.22	2120	84.59	73.61
[8]	927	93.30	81.54	1113	91.94	80.17	1351	90.20	78.57	1601	88.37	76.49	1973	85.65	73.62
[8]+[12]	762	94.49	86.52	987	92.85	84.79	1175	91.47	82.62	1537	88.84	79.49	1918	86.05	75.67
[8]+[12]+[13]	393	97.16	91.38	481	96.52	90.74	529	96.16	90.05	643	95.33	88.81	754	94.52	87.90
[8]+[12]+[13] + PDV-DE	271	98.04	93.55	334	97.58	93.24	354	97.43	92.58	422	96.94	91.60	526	96.18	90.39
CLCMVC	265	98.08	93.67	314	97.73	93.41	326	97.63	92.92	397	97.12	92.07	476	96.54	91.17

is further increased to about 83.8%. Combining candidate modes reduction and selective disparity estimation methods with the search range reduction method as in [8] takes the time saving to over 85%.

To the best of our knowledge, the state-of-the-art results are reported in [8]. Our results first show that our novel framework, which combines state-of-the-art methods ([12], [13] and [8]) of different levels, achieves a significant reduction in encoding complexity compared to [8]. Table 4 shows that this combination can achieve, on average, a time saving of over 91.5%. Compared to [8], the complexity reduction is larger (13%) for sequences with high motion content (Ballroom, Race1) than for sequences with low motion content (6% for the Exit and Vassar sequences). There are two reasons for this. First, as inter-view redundancies are mainly found in still and low-motion regions, and all methods in [8] exploit inter-view redundancies, the room for improvement is small. Second, unlike the methods in the combination [8], which depend highly on inter-view redundancies, the method in [13] exploits the redundancies within a macroblock and the measure of such redundancies is not affected by the type of motion. Thus compared to [8], our novel framework achieves higher gains for high-motion content.

The addition of PDV-DE further increases the time saving by about 1.5%, compared to the combination [8]+[12]+[13]. More time saving is achieved for sequences with high motion content and at high bitrates. For example, the increase in time saving is, on average, over 2% for the Ballroom and Race1 sequences, which are representative of high motion sequences.

The addition of SMCC-MDE (CLCMVC) saves, on average, another 0.6% of the total time, the maximum being 1% for the Ballroom sequence. That takes the overall time saving to over 93.6%, which is an improvement of over 11% in encoding time saving compared to the state-of-the-art [8]. Compared to the method in [8], CLCMVC saves more time at high bitrates. This is because the method in [8] relies heavily on the type of region to reduce the search range for ME and DE, while CLCMVC just uses it as one of the many indicators of the search range (others being the temporal levels and SMCC). At lower bitrates, the proportion of simple regions is bigger, so [8] is very successful. But as the bitrate increases, the proportion of simple regions decreases and so does the efficiency of [8]. Compared to the method in [8], our method saves on average around 32 s per GOP on our test platform. At high bitrates, the saving reaches about 40 s. The saving exceeds 43 s for video sequences with a high level of motion. The addition of SMCC-MDE and PDV-DE to the combination of [8], [12], and [13], results in saving an

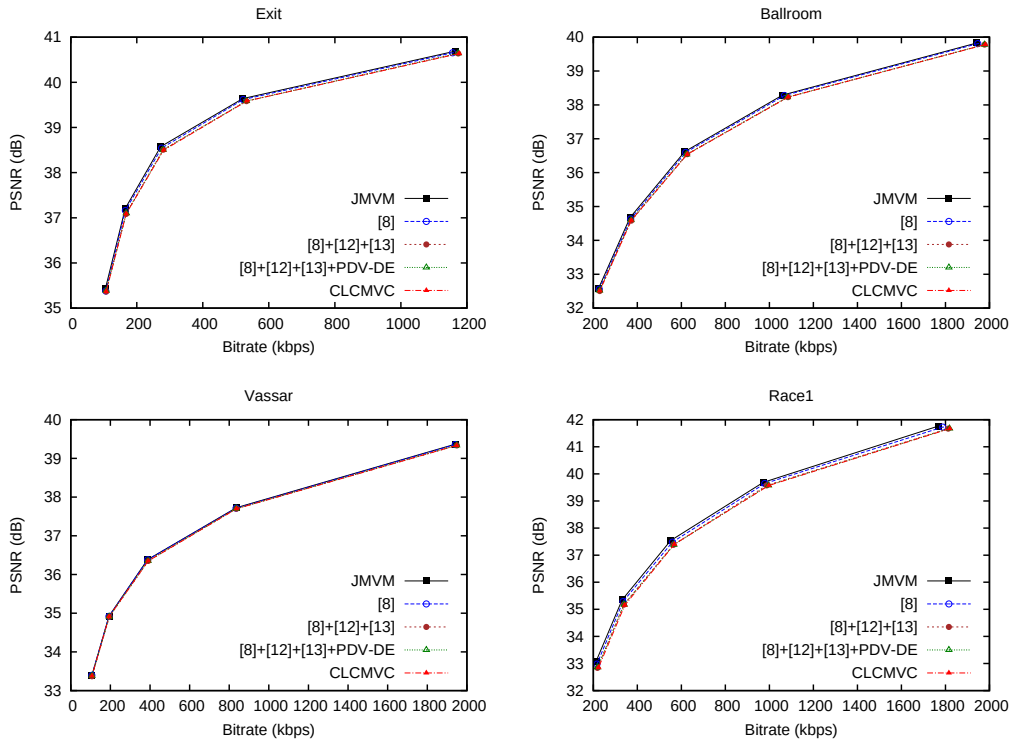


Figure 9: Rate-distortion performance of fast MVC encoding techniques.

additional 4.8 s per GOP. The percentage saving in number of search points also corresponds to the behavior given by the time saving values.

Fig. 9 compares the rate-distortion performance of our method, JMVM, and state-of-the-art methods. The results show that our method does not penalize the rate-distortion performance.

5. Conclusion

We provided a framework for low-complexity MVC. We split the encoding process into four levels (mode decision, prediction direction selection, reference frame selection, block matching) and identified previous relevant techniques at each level. By combining these techniques, we were able to reduce the encoding complexity of JMVM 6.0 with the fast TZ search by about 91.5% on average. We also proposed two new techniques: PDV-DE and SMCC-MDE. PDV-DE exploits the correlation between disparity vectors at high temporal levels in the same view to reduce the search range

for disparity estimation. SMCC-MDE exploits the stereo motion consistency constraint to reduce the search range for both motion and disparity estimation. Integrating PDV-DE and SMCC-MDE in our encoding framework reduced the encoding time and number of search points of JMVM 6.0 by about 93.7% and 96.9%, respectively. This was achieved at a negligible cost of 0.05 dB decrease in PSNR and 1.46% increase in bitrate.

We expect our method to be particularly useful in applications characterized by high motion and high bitrates as in 3D TV sports since it is there that the improvement over the state-of-the-art [8] was the most significant in our performance study. In the future, we plan to extend our work on MVC to 3DVC [23] to jointly exploit texture and depth data in low complexity rate-distortion optimized encoders.

References

[1] A. Smolic, K. Müller, P. Merkle, C. Fehn, P. Kauff, P. Eisert, and T. Wiegand, 3D video and free viewpoint video - technologies, applications and MPEG standards, in Proceedings of ICME 2006, Toronto, Ontario, Canada, July 2006.

[2] M. Flierl and B. Girod, Multi-view video compression - Exploiting inter-image similarities, IEEE Signal Processing Magazine 24 (6) (2007) 66–76.

[3] ITU-T Rec. & ISO/IEC 14496-10 AVC, Advanced Video Coding for generic Audio Visual services, 2005.

[4] P. Merkle, A. Smolic, K. Müller, and T. Wiegand, Efficient prediction structures for multi-view video coding, IEEE Transactions on Circuits and Systems for Video Technology 17 (11) (2007) 1461–1473.

[5] S. Khattak, R. Hamzaoui, S. Ahmad, and P. Frossard, Low-complexity multiview video coding, in Proceedings of PCS'2012, Picture Coding Symposium, Krakow, Poland, May 2012.

[6] ISO/IEC JTC/1 SC29/WG11 and ITU-T SG16 Q.6, JMVM 6.0 software, JTV-Y208, Oct. 2007.

[7] J. Reichel, H. Schwarz, and M. Wien, Joint Scalable Video Model 8, ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, JVT-U202, July 2006.

[8] L. Shen, Z. Liu, T. Yan, Z. Zhang, and P. An, View-adaptive motion estimation and disparity estimation for low complexity multiview video coding, IEEE Transactions on Circuits and Systems for Video Technology 20 (6) (2010) 925–930.

[9] L. Shen, T. Yan, Z. Liu, Z. Zhang, P. An, and L. Yang, Fast mode decision for multiview video coding, in Proceedings of IEEE ICIP, pp. 2593–2596, Cairo, Egypt, Nov. 2009.

[10] H. Zeng, K.-K. Ma, and C. Cai, Fast mode decision for multiview video coding using mode correlation, IEEE Transactions on Circuits and Systems for Video Technology 21 (11) (2011) 1659–1666.

[11] C.-C. Chan and C.-W. Tang, Coding statistics based fast mode decision for multi-view video coding, Journal of Visual Communication and Image Representation (2012). Available at: `http://dx.doi.org/10.1016/j.jvcir.2012.01.004`.

[12] L. Shen, Z. Liu, T. Yan, Z. Zhang, and P. An, Early SKIP mode decision for MVC using inter-view correlation, Signal Processing: Image Communication 25 (2)(2010) 88–93.

[13] Y. Zhang, S. Kwong, G. Jiang, and H. Wang, Efficient multi-reference frame selection algorithm for hierarchical B pictures in multiview video coding, IEEE Transactions on Broadcasting 57 (1) (2011) 15–23.

[14] L. Shen, Z. Liu, S. Liu, Z. Zhang, and P. An, Selective disparity estimation and variable size motion estimation based on motion homogeneity for multi-view coding, IEEE Transactions on Broadcasting 55 (4) (2009) 761–766.

[15] W. Zhu, X. Tian, F. Zhou, and Y. Chen, Fast disparity estimation using spatio-temporal correlation of disparity field for multiview video coding, IEEE Transactions on Consumer Electronics 56 (2) (2010) 957–964.

[16] Z.-P. Deng, Y.-L. Chan, K.-B. Jia, C.-H. Fu, and W.-C. Siu , Fast motion and disparity estimation with adaptive search range adjustment in stereoscopic video coding, IEEE Transactions on Broadcasting 58 (1) (2012) 24–33.

[17] Z.-P. Deng, Y.-L. Chan, K.-B. Jia, C.-H. Fu, and W.-C. Siu, Iterative search strategy with selective bi-directional prediction for low complexity multiview video coding, Journal of Visual Communication and Image Representation 23 (3) (2012) 522–534.

[18] Y. Zhang, S. Kwong, G. Jiang, X. Wang, and M. Yu , Statistical early termination model for fast mode decision and reference frame selection in multiview video coding, IEEE Transactions on Broadcasting 58 (1) (2012) 10–23.

[19] M. Shafique, B. Zatt, and J. Henkel, A complexity reduction scheme with adaptive search direction and mode elimination for multiview video coding, in Proceedings Picture Coding Symposium, Krakow, Poland, May 2012.

[20] I. Patras, N. Alvertos, and G. Tziritas, Joint disparity and motion field estimation in stereoscopic image sequences, in Proceedings of ICPR 1996, Vienna, Austria, Aug. 1996.

[21] H.-S. Koo, Y.-J. Jeon, and B.-M Jeon, MVC Motion Skip Mode, ISO/IEC JTC1/SC29/WG11 and ITU-T Q6/SG16, Doc. JVT-W081, Apr. 2007.

[22] ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, Common test conditions for multiview video coding, JVT-T207, Klagenfurt, Austria, Jul. 2006.

[23] ISO/IEC JTC1/SC29/WG11, Overview of 3D Video Coding, Doc. N9784, Archamps, France, May 2008.