FAST DISPARITY MOTION ESTIMATION IN MVC BASED ON RANGE PREDICTION

Xiaozhong Xu and Yun He

Department of Electronic Engineering, Tsinghua University
Tsinghua National Laboratory for Information Science and Technology

ABSTRACT

In order to achieve a better coding performance in Multi-view video coding (MVC), the search range for both motion estimation (ME) and disparity estimation (DE) are set twice as large as necessary for ME alone, which brings great computational complexity to the encoder. Based on the behavior analysis of disparity estimation, a simple yet effective fast search algorithm, inter-view search range prediction (ISRP), is proposed in this paper. By applying this technique to full search or other fast search algorithms, a majority portion of the disparity search time could be removed while the coding performance is still maintained.

Index Terms—Fast ME, MVC, Disparity Estimation, Search Range Prediction

1. INTRODUCTION

The video coding standard H.264/AVC [1] proposed by the JVT (Joint Video Team, jointly by the ITU-T and ISO/IEC) has achieved a great advancing in terms of coding efficiency for a wide range of applications. Based on its technology, Multi-view Video Coding (MVC) is an ongoing activity, tries to encode multiple video sequences (views) of the same scene, and also inherits the coding tools from H.264/AVC. The basic assumption of MVC is to encode at least one of the views by the standard H.264/AVC codec and use it as the reference view (base view); and then to explore the inter-view correlations among all other views (enhanced view). With the inter-view references, additional coding gains could be achieved beyond the H.264/AVC [2].

One key feature in H.264/AVC is the efficient motion compensation technique, which allows multiple reference pictures and also various block sizes. In order to find an optimal matching result, this kind of flexible compensation mechanism introduces great computational complexity to the encoder side. Motion estimation (ME) is the main bottleneck to speed up the encoder. Approximately 50% (1 reference) ~80% (5 references) time of the encoder is consumed by ME. Therefore a large number of contributions have been proposed in the literature to reduce the complexity of ME. According to the reported results, over 90% of the Full Search (FS, or exhaustive search) time could be removed, while the RD performance is nearly intact [3,4].

In MVC, the ME has been extended to ME (temporal references of the same view) and disparity estimation (DE, inter-view references across views). The behavior of motion across view is quite different than that in temporal direction. Sometimes a displacement of disparity over 100 pixels occurs in DE. In order to explore the possible coding performance gain from the benefit of DE, a large search range (±96 for SD resolution) is set for ME and DE in MVC [5]. This is approximately 4 times of complexity than simply doing temporal ME (±48 for SD). The simulations of MVC then become even tough and in this case over 99% encoding time is spent on ME and DE (for FS).

Although their performances in ME are excellent, the traditional fast algorithms do not take advantage of the DE character well. So even by using a fast search method, the majority of the encoder time is still consumed by ME and DE. Therefore further computational complexity reduction is still needed.

In this paper, the DE behaviors are studied and a new search range prediction method based on that is proposed. The rest parts of this paper are organized as follows: section 2 describes observations and analysis of DE; section 3 presents the proposed method; section 4 is the test results and the conclusion is drawn in section 5.

2. STUDY OF DISPARITY ESTIMATION

In ME, the motion direction for every block could be different but the tendencies of motion around neighboring blocks are highly correlated. This is the general assumption that many fast ME algorithms are based on. However, in DE, for the same time instance, there is no temporal “motion” occurs in the scene and the differences among pictures captured by different cameras are only caused by the displacement from one camera to another. Therefore the motion directions of all the blocks in the current picture should be the same (indicated by the camera movement).
The disparity of two views is decided by the distance between objects and cameras and also by the displacement between two cameras. For a sparse camera set-up and with a large range of object depths, the value of disparity vector could be about 30~50 pixels for background areas and up to 100 pixels for foreground objects. Fig. 1 shows the different disparities of the background and foreground objects between views.

Because the depth changes between objects are not smoothly transited, and even for the same object, it may have different disparities. The strategies of motion vector prediction by using neighboring coded blocks’ information often fail when the disparity change occurs. One possible solution is to enlarge the search range for disparity estimation in order not to miss the optimal-matched block. For instance, when encoding the picture of one view in Fig.1, using the picture of the other view as reference picture, the significant coding gain would continuously exist until the search range is increased to larger than 96. An example of the benefit of increasing the search range is shown in Table I.

### Table I: Comparison of corresponding bitrate reduction using different search ranges for race1 sequence (anchor data is search range = 16, QP = 22/27/32/37)

<table>
<thead>
<tr>
<th>Search Range</th>
<th>32</th>
<th>48</th>
<th>64</th>
<th>96</th>
</tr>
</thead>
<tbody>
<tr>
<td>△Bitrate</td>
<td>-6.72%</td>
<td>-10.49%</td>
<td>-11.82%</td>
<td>-12.11%</td>
</tr>
</tbody>
</table>

Fast algorithms may use search range prediction (based on neighboring blocks’ information), which could enable the search progress with a smaller search window. In inter-view cases, the prediction is less efficiently than in temporal cases due to the irregular disparity changes. Therefore, the fast algorithms need to be modified when applying them to DE.

### 3. PROPOSED SEARCH RANGE DECISION

Search range prediction (or decision) was first proposed in [6] to reduce the search window size of ME. The basic idea is that if the neighboring blocks are static (i.e., with small value of motion vector), the search range of the current block could be properly reduced. In ME, this assumption is right for the smooth and regular areas only. For strong or irregular motions, it may introduce significant RD performance drop. By the above analysis of DE in inter-view pictures, a new search range prediction method is introduced. When encoding the areas with small disparity, the search could be reduced. There are 3 main steps in the proposed algorithm.

#### 3.1. Disparity Map Storage and Refresh

The sampling density of a camera is 1/N second (N usually may be 24, 25, 30, or larger). Considering this dense sampling rate, normally the distribution of the current picture’s disparity and the one in the adjacent time instances will have strong correlation (between 2 fixed views). Unless strong motions happen temporally, this correlation is reliable. Disparities of temporal static objects will be the same in the incoming picture of the same view.

Therefore the disparity map of the current picture is recorded as a disparity predictor for the incoming inter-view predicted picture of the same view. For simplicity, only the disparity vectors of 16x16 blocks are stored for simulation in this paper. After encoding each inter-view picture, this disparity map is then refreshed by the disparities of the Macro-blocks in the current picture.

#### 3.2. Local Disparity Feature Decision

In the block level decision, three already obtained disparities, namely the disparity of co-located MB in the disparity map (dis_col), the disparity of the current 16x16 block (dis_16) and the disparity predicted by spatial neighboring blocks (dis_pred), are highly correlated with the current block. They are used to predict the current block’s disparity status. In case of the 16x16 mode, dis_16 is set to be 0. The intensity prediction of the current block’s disparity is decided in (1) and (2). The proposed search range prediction strategy categorizes the current block into 3 different modes (for 2 dimensions separately):
3.2.1. Low disparity mode
In this mode, the area is recognized as background and should result in “zero” disparity. In this mode, \( DV_x \) or \( DV_y \) is smaller than 2.

3.2.2. High disparity mode
In this mode, the area is recognized as foreground and should result in “large” disparity. In this mode, \( DV_x \) or \( DV_y \) is larger than 3. Large search range should apply to blocks in this mode.

3.2.3. Intermediate disparity mode
For those do not belong to the above 2 categories, they are identified as in intermediate mode, which has certain disparity magnitudes but not large.

3.3. Search Range Reduction
According to the decision made in section 3.2, 2 scale factors namely “vertical scale” (VS) and “horizontal scale” (HS) are set accordingly to reduce the search window size. The initial search range (SR) in x or y direction will be divided by these factors. Considering the disparity of 16x16 blocks will be used as predictor for other modes, the scale factor for it should be set conservatively. Therefore, for low, high and intermediate modes, the factor (VS or HS) is experimentally set to be 4, 1, 2 (for block 16x16) and 8, 1, 4 (for other block sizes) separately.

For ME, the motion could be in any direction. Therefore the search window is designed as a square. As is mentioned above, one observation on DE is that the directions of disparity are identical for all the blocks as long as the relative positions between cameras are not changed. And only the magnitudes of them are determined by the distance between the object and its camera. Therefore the symmetrical search window is redundant. Once the tendency of disparity direction is detected, the opposite direction (both in x-axis and y-axis) could be paid less attention to.

The above-mentioned direction is also called global disparity direction, which could be derived easily via summing up the already obtained disparity vectors and checking their signs in two dimensions separately.

The search range reduction is then further achieved by reducing the opposite directions of the search window. That is, the scale factors for two directions along x or y axis could be different. To design an unsymmetrical window, four factors are defined to the 4 directions of the search window. They are namely “Negative Vertical Scale” (NVS), “Positive Vertical Scale” (PVS), “Negative Horizontal Scale” (NHS) and “Positive Horizontal Scale” (PHS). The four factors are subject to (3) and (4):

\[
\begin{align*}
NVS \times PVS &= 16 \quad (3) \\
NHS \times PNS &= 16 \quad (4)
\end{align*}
\]

If the global disparity direction along x-axis (y-axis) is positive, then PVS = VS (PHS = HS); otherwise, NVS = VS (NHS = HS). So that the part of the search window in the opposite disparity directions could be properly reduced.

Finally, the initial search window size is reduced from: \((-SR, SR)\) to \((-SR/VNS, SR/VPS)\) vertically; and from \((-SR, SR)\) to \((-SR/HNS, SR/HPS)\) horizontally. An example of the reduced search window is shown in Fig.2.

4. SIMULATION RESULTS AND DISCUSSION
In order to study the disparity estimation purely in this paper, the mixed prediction structure (i.e. temporal and inter-view references co-exist) is disable. Only inter-view prediction is allowed in the simulations. As is shown in Fig.3, 2 views are encoded in the simulations while the P-coded view is compared under different disparity estimation algorithms. The initial search range for different methods is
set to be \( \pm 96 \). The QP values are \([22, 27, 32, 37]\) and 50 frames are coded for each view. The simulation platform is JMVM_6 [7] (reference software for JVT MVC). The distances between 2 cameras are selected as 10cm for ren sequence and 40cm for the other 2 (considered as a sparse set-up). All the sequences are of 640x480 resolution.

JMVM software uses an extended diamond search (XZS), which is based on MV prediction as start search points. In Table II, the “speedup” in columns means the ME speed comparison between the current method and the FS. For example, the ISRP DE is 11.94 times faster than FS DE for ballroom sequence. According to Table II, the average DE speedup of ISRP over FS is about 12.9; the average encoder speedup of ISRP over FS is about 12.4. The average DE speedup of JMVM over FS is about 45.8; the average encoder speedup of JMVM over FS is about 40.4.

Generally, ISRP could be applied into other fast search algorithms. And when combined with the JMVM in this simulation, the speedup is increased to 246 for DE and 142 for the encoder over FS. In another word, an additional speedup of 5.4 for DE and 3.5 for the encoder over JMVM is achieved. The complexity burden is greatly relieved. In terms of search quality, the corresponding PSNR loss of the proposed method relative to FS is negligible (less than 0.05dB).

For different DE algorithms, the rest of the encoding operations are still the same. Except DE, the encoding time spent on the rest parts of encoder is normally stable. The encoding time for the rest parts of the encoder are recognized as “1” unit in Table III. In this table, the encoding time comparison between DE and the rest of encoder are listed. We can see that FS DE spends over 99% of the encoding time and the JMVM reduces this number to about 90%. By combining the ISRP, only 58% encoding time is consumed by DE. This brings considerable benefit for simulations in MVC.

5. CONCLUSION

In this paper a study of disparity vector estimation is presented. Based on the special behavior of disparity distribution, a simple and efficient technique, ISRP, is proposed. The initial search range is therefore reduced into a proper size. According to the simulation results, a speedup of 12.9 over FS and 5.4 over the fast search algorithm in JMVM is achieved. The encoding speed of MVC is significantly improved therefore. By applying the combination of ISRP and JMVM, the DE performance is even better.

6. REFERENCES