Adaptive Multi-resolution Motion Estimation Using Texture-based Search Strategies

Jie Liu, Xianghu Ji, Chuang Zhu, Huizhu Jia, Xiaodong Xie, WenGao, Fellow, IEEE
Engineering Lab for Video Technology, School of EECS, Peking University
Email: {liuzimin, xhji, czhu, hzjia, xdxie, wgao}@jdl.ac.cn

Abstract—Motion estimation is the most complex module which contributes nearly 70% of computation resources in a hardware-based video encoder. This huge computational complexity limits the performance of HD video encoders in terms of encoding speed and power consumption. This paper presents a hardware oriented multi-resolution motion estimation algorithm using adaptive search strategies to reduce computational complexity. The spatial homogeneity and the temporal stationarity characteristics of video sequences are adaptively detected to determine search range and down-sampling rate. Homogeneous regions are detected by using Sobel edge operators and stationary regions are detected by using temporal information. These texture-based search strategies make motion estimation more concise under fixed computational complexity constraint. Experimental results show that the proposed algorithm achieves better performance and reduces computation cost by 40% compared with previous works.

Index Terms — Multi-resolution motion search, search strategies, homogeneity, stationarity, computational complexity constraint.

I. INTRODUCTION

In many video coding standards, such as H.264/AVC [1], motion estimation (ME) plays a key role in the block-based hybrid coding framework. Integer motion estimation aims at reducing temporal redundancies between the current frame and the reference frame. There are some new tools such as variable block size motion estimation (VBSME), multiple reference frames for real-time motion estimation in high definition (HD) video encoder. As a result, the complexity and computation cost increase greatly.

Full-search block matching algorithm (FSBMA) [2] is widely used for hardware ME design due to its superior performance and high regularity. However, FSBMA needs lots of computation due to many candidate blocks to be matched. Many fast ME algorithms, including SEA [3] and DSA [4], were proposed to reduce high computation complexity. But these software-oriented approaches cannot be used in the hardware-based encoder. Multiresolution motion estimation algorithm (MMEA) [5]–[7] is developed with a coarse-to-fine search hierarchy. The MMEA is suitable for hardware implementation with its highly regular data flow. And, it can reduce the computational complexity by decreasing the number of computations.

However, traditional MMEA [5]–[7] only use fixed search range and the same down-sampling is applied to all the image area without discrimination. Thus, firstly, although it performs well for small and uniform motions, the resulting performance degradation is not negligible when the motion is complex. The basic idea of traditional MMEA is that potential match candidates are obtained from a large search area at the coarse level and the candidates become the search center in the lower fine levels. But for the sequences with complex texture, the MV in coarse search may be an incorrect result. It will yield search error directly passed on to the next level. If using large search range in fine level without down-sampling, these methods will have a high computational cost because the fine level contains large amount of calculation. Secondly, for flat region, since the texture in this region has similar spatial property, performance degradation caused by downsampling is negligible. Downsampling search at coarse level will be accurate in flat region. In this situation, downsampling search should be applied to reduce computational cost. Thirdly, when the current macroblock (MB) is not moving in adjacent video frames such as background, much computational complexity will be wasted since it also searches in the large search window.

In this paper, an adaptive multi-resolution motion estimation algorithm (AMMEA) by using texture-based search strategies is proposed. It makes search strategies customized for each block and reduces computational cost by 40% compared with traditional MMEA [5] [6]. For different kinds of video sequences, it applies adaptive search range and down-sampling rate based on stationary and homogeneous features of current MB. The proposed algorithm use Sobel edge operators to detect homogeneous regions and stationary regions are detected by using temporal information.

The remainder of this paper is arranged as follows. Section II gives a brief introduction of the typical MMEA and its challenges. The AMMEA is proposed and explained in Section III. The experimental result are given in Section IV. Finally, conclusions are drawn in Section V.

II. TRADITIONAL MMEA AND ITS CHALLENGES

The previous MMEA [5]–[7] includes three levels, as is illustrated in Fig. 1. In level 2 (most down-sampled), the search window is the largest and centered on original point (0, 0). In level 1, four search windows are centered on three candidates selected from level 2 and a candidate selected through predicted motion estimation (PMV), respectively. The 4:1 down-sampling is adopted in level 1. Level 0 is a fine level without data subsampling and VSBME is used in this level. The supported block size is larger than or equal to 8x8.

The MMEA only has fixed search range and the same down-sampling for all blocks of a sequence uniformly. Fig. 2
Considering the stationary MB may move in small range, it is
in the reference frame. In this paper, spatial homogeneity and
the difference of the current MB and its co-located counterpart
MB’s edge intensity, and temporal video objects. Spatial homogeneity of a MB is based on the
decide the best mode in inter coding by using the spatial
second is that the adaptive multi-resolution motion estimation
algorithm into two parts to analyse. The first is the adaptive search strategies based on feature of block, the
more likely to move around (0, 0) with a small search range. But it
should be (0, 0) according to FSBMA. For the current block
with complex texture, the MV in coarse search is an incorrect
result. And, it yields error match directly passed on to the next
level. Therefore, performance degradation is inevitable in
MMEA due to downsampling sequences with complex texture.
Another problem is that the temporal stationary blocks are
more likely to move around (0, 0) with a small range. But it
also searches in the large search window and the much computational complexity will be wasted.

III. ADAPTIVE MMEA USING SEARCH STRATEGIES

In this section, we divide adaptive multi-resolution motion
estimation algorithm into two parts to analyse. The first is the
adaptive search strategies based on feature of block, the
second is that the adaptive multi-resolution motion estimation
algorithm using the search strategies.

A. Search Strategies Based on Feature of MB

In [8], it proposes a fast inter mode decision algorithm to
decide the best mode in inter coding by using the spatial
homogeneity and the temporal stationarity characteristics of
video objects. Spatial homogeneity of a MB is based on the
MB’s edge intensity, and temporal stationarity is decided by
the difference of the current MB and its co-located counterpart
in the reference frame. In this paper, spatial homogeneity and
temporal stationarity are applied to adjust search strategies.
Considering the stationary MB may move in small range, it is
centered at the current MB position in the reference frame that
is the origin (0, 0) and the search range(SR) will be set to be
very small for reducing calculation. When MBs in the picture
are considered as homogeneous blocks, performance
degradation caused by downsampling searching at coarse level
is negligible. Thus, it is reasonable to search in large window
down-sampling. In contrary, for nonhomogeneous blocks,
if downsampling pixels are selected for final MV, the ME
quality loss is inevitable. In this situation, it will abandon
searching in coarse level. Then it will only search in fine level
with a larger search range so as to achieve significant coding
gain using VBSME.

Before the hierarchical ME process, there are two steps for
stationary regions determination and homogeneous regions
detection to determine search strategies. Stationary regions
refer to non-moving regions in the temporal dimension. In this
paper, we use stationary regions detection to determine search
range. If current block is regarded as stationary region, we will
only search around (0, 0) with a small search range. In natural
video sequences, there are correlations between current frame
and reference frame. A method for detecting stationary region
is proposed using temporal information [8]. The difference
between current MB and reference MB can be computed by
using (1). Here C[i, j] and P[i, j] are respectively the
luminance values in the current MB and reference MB. The
image is 8 bit per pixel, and setting the threshold to 200
achieves good performance as suggested in [8].

\[ \text{Diff} = \sum_{i,j \in \text{MB}} \text{abs}(C[i,j] - P[i,j]) \]  

(1)

In addition, we also utilize homogeneous region
determination to adjust search strategies. Homogeneous region
refers to the regions having similar texture in the spatial
domain. Edge information can represent texture complexity.
According to the analysis on the texture complexity on image,
homogeneous regions will be detected. As analysis in [8],
there are many techniques for detecting edge information.
Using the Sobel edge operators to obtain the edge information
is a balance between computational expense and performance.
The Sobel edge operators have two 3x3 convolution kernels to
calculate approximations of the derivatives, one for horizontal
changes, and another for vertical. For a pixel, in a luma picture,
we define the corresponding edge vector, \( \vec{D}_{ij} = (dx_{ij}, dy_{ij}) \) as

\[
dx_{ij} = p_{i+1,j-1} + 2 \times p_{i+1,j} + p_{i+1,j+1} - p_{i-1,j-1} - 2 \times p_{i-1,j} - p_{i-1,j+1} \\
dy_{ij} = p_{i+1,j-1} + 2 \times p_{i-1,j} + p_{i-1,j+1} - 2 \times p_{i+1,j+1} - p_{i-1,j-1} \\
\]  

(2)

where \( x \) and \( y \) represent the degree of difference in vertical
and horizontal directions respectively. Therefore, the
amplitude of the edge vector can be computed by

\[
\text{Amp}(\vec{D}_{ij}) = \sqrt{dx_{ij}^2 + dy_{ij}^2} \\
\]  

(3)

The current block size is 16x16, so the homogeneity size of
a block is the same. The sum of the amplitude of the edge
vectors in the block is divided into three categories by two
thresholds Thd1, Thd2. Three categories correspond to
different search strategies. The details will be further
discussed in Section III-B. The \( r \) and \( c \) are the indices of the

Fig. 1. Three level multi-resolution motion estimation shows an example frame from the 720P sequence
"Spincalindar". In Fig. 2(a), three best candidates after the
level 2 search and PMV (0, 0) are chosen as the search center for
level 1. The winner candidate of the level 1 is selected as the
center for level 0 in Fig. 2(b). Fig. 2(c) shows that the final
MV is achieved by using MMEA. However, the best MV
should be (0, 0) according to FSBMA. For the current block
with complex texture, the MV in coarse search is an incorrect
result. And, it yields error match directly passed on to the next
level. Therefore, performance degradation is inevitable in
MMEA due to downsampling sequences with complex texture.

Fig. 2. The best MVs chosen after each level
row and column of the block. The sum amplitude of one MB is represented as (4). The adaptive search strategies of MB(r,c) are defined as follows:

\[ H(r,c) = \sum_{i,j \in N} \text{Amp}(D_{i,j}) \]

(4)

\[ \text{MB}(r,c) = \begin{cases} \text{level 0:on, level 1:off, level 2:off}, & \text{if } H(r,c) = \text{Thd}_1 \\ \text{level 0:on, level 1:on, level 2:off}, & \text{if } \text{Thd}_2 < H(r,c) < \text{Thd}_1 \\ \text{level 0:on, level 1:on, level 2:on}, & \text{if } H(r,c) < \text{Thd}_2 \end{cases} \]

(5)

B. Adaptive Multi-Resolution Motion Estimation Algorithm

Fig. 3. Adaptive MMEA flow chart

Fig. 3 provides the algorithmic flow chart. To begin with, it will determine whether the current MB is stationary or not. If the current MB is stationary, the proposed algorithm adopts the search strategies that abandon searching at level 2 and level 1. Considering the stationary MB may move in small range, it is centered on (0, 0) and the SR is set to \([-SR_x^{\text{Static}}, SR_x^{\text{Static}}] \times [-SR_y^{\text{Static}}, SR_y^{\text{Static}}]\) at level 0, as shown in Fig. 4(a). Further more, if it is not a stationary MB, we need to determine whether the current MB is homogeneous or not. As analyzed in Section III-A, there are two threshold Thd1 and Thd2 dividing the amplitude of the edge vectors into three categories (5). According to the experimental results, Thd1=20000 and Thd2=16000 will achieve good performance for all kinds of test sequences. The three categories are introduced as follows:

a) \(H(r,c) > \text{Thd}_1\) The current MB is regarded as highly complex texture. In this situation, the resulting performance degradation is not negligible if downsampling is still used. Highly complex MB has a strong chance to be encoded using VBSME. VBSME is unsuited to be utilized at coarse levels. Therefore, it doesn’t search at level 2 and level 1. Since the current MB is nonstationary, it is centered at PMV and the SR is set to \([-SR_x^{\text{Non}}, SR_x^{\text{Non}}] \times [-SR_y^{\text{Non}}, SR_y^{\text{Non}}]\) at level 0, Fig. 4(b). This search strategies make sure that VBSME be regarded in finest level with larger SR.

b) \(\text{Thd}_2 < H(r,c) < \text{Thd}_1\) The current MB is regarded as modestly complex texture. Since level 2 is the coarsest level that 1:1 downsampled from level 0, it only searches at level 1 and level 0. In level 1, it is centered at PMV and the SR is set to \([-SR_x^{\text{Non}}, SR_x^{\text{Non}}] \times [-SR_y^{\text{Non}}, SR_y^{\text{Non}}]\). The 4:1 downsampling is applied in this level. After the level 1 search, the MV with minimum cost is selected as the center for level 0 search window with the SR is \([-SR_x^{0}, SR_x^{0}] \times [-SR_y^{0}, SR_y^{0}]\), Fig. 4(c).

c) \(H(r,c) < \text{Thd}_2\) The current MB is a homogeneous MB. In [8], the current MB will most probably be encoded using 16x16 mode. To a homogeneous MB, the resulting performance degradation caused by downsampling is negligible and the MV from the coarse level can be thought of relatively accurate. It is reasonable to use 16x16 mode at level 2 and level 1. In this situation, the search strategy is the same as three level MMEA [5][6] in Fig. 1. The only difference is that it is enough to search with a very smaller SR\(([-SR_x^{1}, SR_x^{1}] \times [-SR_y^{1}, SR_y^{1}])\) at level 0.

For different kinds of video sequences, the proposed algorithm uses flexible search range and down-sampling rate. Compared with previous works, the computational cost is reduced while maintaining a better performance. The details of the experimental result will be described in Section IV.

IV. EXPERIMENTAL RESULTS

Our proposed algorithm was implemented to support AVS Jizhun profile. According to the specifications provided in [5][6], the test conditions are as follows: 1) Table 1 shows search window comparison. 2) SAD is used as the matching distortion criterion. 3) Reference frame number equals to 2. 4) VLC is enabled. 5) MV resolution is 1/4 pel. 6) GOP structure is IPPBP. 7) Inter block mode from 16x16 to 8x8. 8) The number of frames in a sequence is 32.

<table>
<thead>
<tr>
<th>Design</th>
<th>MMEA[5][6]</th>
<th>Proposed algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level2</td>
<td>(SR_x = 128)</td>
<td>(SR_x = 128)</td>
</tr>
<tr>
<td>Level1</td>
<td>(SR_y = 96)</td>
<td>(SR_y = 96)</td>
</tr>
<tr>
<td>Level0</td>
<td>(SR_x = 8)</td>
<td>(SR_y = 8)</td>
</tr>
<tr>
<td></td>
<td>(SR_x = 8)</td>
<td>(SR_y = 8)</td>
</tr>
<tr>
<td>Level0</td>
<td>(SR_x = 10)</td>
<td>(SR_y = 10)</td>
</tr>
<tr>
<td></td>
<td>(SR_x = 4)</td>
<td>(SR_y = 4)</td>
</tr>
<tr>
<td></td>
<td>(SR_x = 4)</td>
<td>(SR_y = 4)</td>
</tr>
<tr>
<td></td>
<td>(SR_x = 12)</td>
<td>(SR_y = 12)</td>
</tr>
<tr>
<td></td>
<td>(SR_x = 12)</td>
<td>(SR_y = 12)</td>
</tr>
</tbody>
</table>

In Table I, the search range and down-sampling rate of the proposed algorithm is observed through experimental results to achieve a better performance while reducing computation cost by 40% compared with [5][6]. Keeping fixed computation cost, AMMEA adjustable search range depends on different search strategies in Table I. According to fully re-configurable
parallel processing element (PE) array structure [5], there are totally 6×2 parallel four-pixel PEs in this architecture. The computational cost to determine search strategies is negligible. The total cycle consumption for three levels are given as follow:

\[
T_{\text{me}} = \frac{(2 \times SR_x / 4) \times (2 \times SR_y / 4)}{4} + (2 \times SR_x \times 2) \times (2 \times SR_y \times 2)
\]

(6)

Table II shows the total cycle consumption of the proposed algorithm and the comparison with previous works.

**TABLE II IMPLEMENTATION COST COMPARISON WITH PREVIOUS ME**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Ref. Number</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Search Range</td>
<td>256×192</td>
<td>256×192</td>
<td>128×64</td>
<td>196×128</td>
<td>128×64</td>
<td>65×65</td>
<td></td>
</tr>
<tr>
<td>Number of PEs</td>
<td>512</td>
<td>512</td>
<td>N/A</td>
<td>2048</td>
<td>128×9</td>
<td>16×16</td>
<td></td>
</tr>
<tr>
<td>Data Latency (Cycles)</td>
<td>576×512</td>
<td>848</td>
<td>N/A</td>
<td>960</td>
<td>1536</td>
<td>5216</td>
<td></td>
</tr>
<tr>
<td>Working Frequency (MHz)</td>
<td>150</td>
<td>220</td>
<td>108</td>
<td>200</td>
<td>108</td>
<td>260</td>
<td></td>
</tr>
</tbody>
</table>

Different sequences under different resolutions (1080P, 720P, D1, CIF) are chosen for the test. Every resolution selects some sequences with different characteristics. These features include not only complex texture and high motion, but also homogeneous region and small motion. The PSNR degradation of proposed algorithm and [5][6] compared with FSBMA are shown in Table III. The metrics is BD-PSNR of these sequences with complex texture and small motion, such as "BasketballDrive" and "Tractor". Meanwhile, Fig. 5. shows the PSNR curves for better performance than [5][6], such as "BasketballDrive" and "Tractor". From Table III, we find that the feature of these sequences with complex texture and small motion, such as "Spinnaclendar" and "Fireworks", have high PSNR gain. For high motion sequences, the proposed algorithm still have better performance than [5][6], such as "BasketballDrive" and "Tractor". Meanwhile, Fig. 5. shows the PSNR curves for different resolutions.

**TABLE III THE PSNR PERFORMANCE COMPARISON**

<table>
<thead>
<tr>
<th>Resolution</th>
<th>Sequence</th>
<th>Complex Texture</th>
<th>Simple Motion</th>
<th>Complex Motion</th>
<th>Proposed (dB)</th>
<th>[5][6] (dB)</th>
<th>Diff (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1080P</td>
<td>Fireworks</td>
<td>✓</td>
<td>✓</td>
<td>-0.10</td>
<td>-0.17</td>
<td>0.07</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BasketballDrive</td>
<td>✓</td>
<td>✓</td>
<td>-0.06</td>
<td>-0.07</td>
<td>0.01</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tractor</td>
<td>✓</td>
<td>✓</td>
<td>-0.06</td>
<td>-0.07</td>
<td>0.01</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Cactus</td>
<td>✓</td>
<td>✓</td>
<td>-0.01</td>
<td>0.02</td>
<td>0.01</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MobalCfer</td>
<td>✓</td>
<td>✓</td>
<td>0.00</td>
<td>0.01</td>
<td>0.01</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Crowndan</td>
<td>✓</td>
<td>✓</td>
<td>0.00</td>
<td>0.04</td>
<td>0.04</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Spinnaclendar</td>
<td>✓</td>
<td>✓</td>
<td>-0.07</td>
<td>-0.13</td>
<td>0.06</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sheriff</td>
<td>✓</td>
<td>✓</td>
<td>0.00</td>
<td>-0.01</td>
<td>0.01</td>
<td></td>
</tr>
<tr>
<td></td>
<td>City</td>
<td>✓</td>
<td>✓</td>
<td>0.03</td>
<td>0.07</td>
<td>0.04</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Optis</td>
<td>✓</td>
<td>✓</td>
<td>0.00</td>
<td>0.01</td>
<td>0.01</td>
<td></td>
</tr>
<tr>
<td>720P</td>
<td>Mobalcalendar</td>
<td>✓</td>
<td>✓</td>
<td>-0.05</td>
<td>-0.14</td>
<td>0.09</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Flowergarden</td>
<td>✓</td>
<td>✓</td>
<td>-0.00</td>
<td>0.05</td>
<td>0.04</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mobile</td>
<td>✓</td>
<td>✓</td>
<td>-0.00</td>
<td>-0.18</td>
<td>0.18</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Foreman</td>
<td>✓</td>
<td>✓</td>
<td>0.04</td>
<td>-0.11</td>
<td>0.07</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Kiel</td>
<td>✓</td>
<td>✓</td>
<td>0.05</td>
<td>0.21</td>
<td>0.16</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Crew</td>
<td>✓</td>
<td>✓</td>
<td>0.02</td>
<td>-0.07</td>
<td>0.05</td>
<td></td>
</tr>
</tbody>
</table>

V. CONCLUSION

In this paper, a hardware oriented fast motion estimation algorithm is proposed by using MB's texture and stationarity characteristics to determine search strategies. The proposed algorithm and architecture of IME also can be used for H.264/AVC and HEVC. Compared to FSBMA, the proposed algorithm reduce the computational complexity with a negligible average PSNR loss of 0.03 dB. It has better performance and reduces computation cost by 40% compared with other hierarchical motion search algorithm. The proposed algorithm reaches a good balance between computational complexity and performance.

REFERENCES