Adaptive Search Range Scaling for B Pictures Coding

Zhigang Yang[1], Wen Gao[1,2], Yan Liu[1], and Debin Zhao[1]

[1] Department of Computer Science and Technology,
Harbin Institute of Technology, Harbin 150001, China
[2] Institute of Computing Technology, Chinese Academy of Science,
Beijing 100080, China
{zgyang, wgao, liuyan, dbzhao}@jdl.ac.cn

Abstract. This paper presents a frame-level adaptive search range scaling strategy for B pictures coding for H.264/AVC from the hardware-oriented viewpoint. After studying the relationship between search range of P and B picture, a simple search range scaling strategy is proposed at first, which is efficient for normal or low motion video. After that, this strategy is extended to high motion video by using the information of intra prediction and motion vector of each P picture to restrict the search range of adjacent B pictures. This adaptive search range scaling strategy can not only reduce approximate 60% search area of B pictures, but also keep almost the same coding performance as the reference software.

Keywords: Video coding, adaptive search range scaling, B pictures, high motion video, H.264/AVC.

1 Introduction

In hybrid video coding, B pictures are encoded using both future and past pictures as references [1]. In the recent H.264/AVC standard [2], there are usually five prediction modes in B pictures coding, including forward prediction, backward prediction, bi-prediction, direct prediction and intra modes. The prediction signals of direct and bi-prediction modes are obtained by a linear combination of forward and backward prediction values based on motion compensation. The usage of such various modes is able to exploit the temporal correlation more efficiently between reference pictures and current B picture, especially for uncovering areas caused by zooming, non-linear motion etc. Therefore, B pictures can achieve higher compression ratio than P pictures. B pictures can also be coarsely quantized in some applications as B picture does not propagate errors when it is not used as a reference. In other words, B pictures become a substantial part in video coding in terms of both coding performance and video transmissions, especially in some real-time video transmissions services [3].

The high performance of B pictures is on the basis of various motion compensation predictions and complex motion estimation. In order to accelerate B pictures encoding for real-time applications, fast block-matching algorithm might be used. However, in hardware implementation, for example by means of application specified integrated

Y. Zhuang et al. (Eds.): PCM 2006, LNCS 4261, pp. 704 – 713, 2006.

circuits (ASIC) or field programmable gate arrays (FPGA), full search (FS) is usually adopted [4], [5] after comparing with other fast block-matching algorithms, because the specific hardware is usually designed for pipelines but not for uncertain condition branch operations. Another efficient way to reduce the computational complexity is search range adjustment for different motion-level video. When full search is adopted, the search range mainly determines the time spent. Furthermore, the processor can only deal with the data which have already been loaded into the high speed cache from outside. Therefore the size of search area also affects the cache hit rate and the data exchange rate between on-chip and extended memory [6]. In one word, a reasonable search range should not only reduce the encoding time but also keep the encoding performance.

The existing search range adjustment algorithms like [7], [8], [9], [10] are mainly focused on marcoblock level. That is to say, the motion information from four adjacent blocks (left, up-left, up, up-right) is used to estimate the current block's search range. Such methods can achieve good efficiency under the framework of general purpose processor, but they can not work well with full hardware design. Because these kinds of search range prediction bring serious correlations between the current block and its adjacent blocks, the current block's search area can not be determined until the prior blocks finish motion estimation. These correlations limit wide usages of background transfer and force the hardware architecture to process block serially. The actually efficient algorithm for hardware implementation is frame-based search range adjustment. Although [7] and [8] have mentioned some frame-level search range adjustment algorithms, they are not suitable for B pictures coding. In [7], the mean and variance of the entire motion vector in one frame are adopted to characterize the global motion activity, which is a part of whole search strategy. This frame-level algorithm is simply to help the macroblock-level algorithm to be more precise, and it can not give a good performance when working separately. In [8], the sum of absolute vector values and the sum of prediction errors in one frame are calculated to decide search range, which is based on fixed thresholds. In fact, using fixed thresholds is not practical, because the thresholds vary a lot on different video content, and even different coding tools can also affect the thresholds. Moreover, all the existing algorithms have not considered the relationship between P and B pictures.

The purpose of this paper is to find out a hardware-oriented search range scaling algorithm for B pictures coding based on the H.264/AVC to reduce the computational complexity while keeping the coding performance. The rest of this paper is organized as follows. In Section 2, the relationship between P and B pictures is first studied, and then simple search range scaling method is introduced. After analyzing the proposed simple method, an improved efficient adaptive search range scaling algorithm is presented in Section 3. Simulated results are demonstrated in Section 4 to show its effectiveness. Finally, the paper is concluded in Section 5.

2 Search Range Scaling (SRS)

The proposed strategy is on the basis of on the relationship between P and B pictures, so we first study the motion feature in P and B pictures.

2.1 Motion Feature in P and B Pictures

Based on H.264/AVC standard, we assume that MV is the motion vector of the current block, PredMV is the motion vector prediction of the current block, and MVD is the difference between MV and PredMV. The search center in the reference picture is pointed by PredMV, so MVD is actually restricted by the search range.

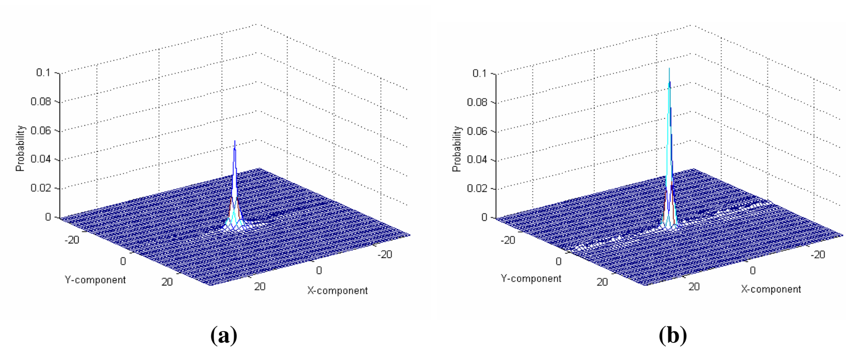

Fig. 1. MVD joint probability distributions of *bus* (condition: CIF, 150 frames, IBBPBBP, 2 reference, QP=24, full search ±32). (a) Distribution of P pictures. (b) Distribution of B pictures.

In order to get MVD statistic, *bus* sequence in CIF format was tested on conditions of full search in ±32 window and constant QP=24. Comparing the two typical MVD joint probability distributions of P and B pictures, respectively shown in Fig. 1(a) and Fig. 1(b), we can see that P pictures seem more "active" than B pictures. This is mainly because the temporal distance between B picture and its forward/backward reference picture is smaller than the distance between the forward and backward reference picture. Suppose that there is an object M moving from point A to point B, and the corresponding position in B picture is point C, shown in Fig. 2. Then

$$CA : CB : BA = tb : -tp : td \tag{1}$$

where *tb and tp* denote the temporal distance between the current B picture and the forward/backward reference picture respectively, and *td* denotes the temporal distance

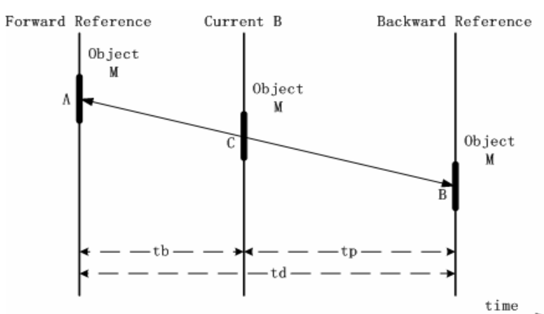

Fig. 2. Demonstration of different motion degree in P/B pictures

between the forward and backward reference picture. *CA* and *CB* represent the forward/backward motion vector of the blocks in object M in current B picture, and *BA* represents the motion vector of the blocks in object M in backward P picture.

2.2 Search Range Scaling (SRS) and Its Performance

Equation (1) indicates that the object's motion degree of B picture is lower than that of P picture. That is to say, the search range of B picture needs not to be as large as that of P picture. A simple search range scaling (SRS) algorithm for B picture can be obtained as follows:

$$SR_F = \lceil SR_{\max} \times tb / td \rceil \text{ (for the latest forward reference picture)} \quad (2)$$
$$SR_F = SR_{\max} \quad \text{(for other forward reference picture)}$$

$$SR_B = \lceil SR_{\max} \times tp / td \rceil \text{ (for the latest backward reference picture)} \quad (3)$$

where SR_F denotes the forward search range of B picture, SR_B denotes the backward search range of B picture, and SR_{max} denotes the max search range of P picture.

Six sequences with 30fps in CIF format are selected to evaluate SRS algorithm, including two high motion sequences *stefan* and *foreman*, two normal motion sequences *bus* and *mobile*, and two low motion sequences *news* and *paris*. The test conditions are listed in Table 1. Because this scaling method only focuses on the latest forward reference picture, so we only choose one forward reference which can show the changes of coding performance more clearly.

Table 2 shows the average PSNR gain and the average bit rate saving for only B pictures comparing with the original reference software. SRS algorithm works very

Table 1. Test conditions

Reference Software	H.264/AVC JM10.2
MV Resolution	1/4 pixel
RD Optimization	OFF
P References Number	2
B References Number	1 Forward, 1 Backward
Search Range	24
Motion Estimation	Fast Full Search
GOP Structure	IBBPBBP
QP	24,28,32,36
Loop Filter	ON

Table 2. Performance comparisons between SRS and the reference software

Sequences		stefan	foreman	bus	mobile	news	paris
B pict-ures	PSNR gain	-0.049	-0.011	0	-0.001	0	0
	BR saving	-26.29%	-3.315%	0.183%	0.105%	-0.016%	0.010%

Note: BR denotes bit rate.

well on normal and low motion sequences, but for high motion sequences, the performance losses quite a lot. Especially for *stefan* sequence, the bit rate of B pictures even increases more than 26%. So in the next part, we will analyze this phenomenon and improve the SRS algorithm.

3 Adaptive Search Range Scaling (ASRS)

In this section, we are going to find out why the bit rate increase so much, and then extend SRS algorithm to high motion video by adaptive prejudgment.

3.1 SRS Algorithm Analysis

SRS algorithm is based on the assumption that all the motion vectors of P pictures are restricted in the maximum search range. However, this requirement is not reachable all the time for high motion video; for example, there are two sudden camera motions in *stefan* sequence, and our experiments show that the peak difference between two P picture during the camera motion could be more than 100 pixels. Under such

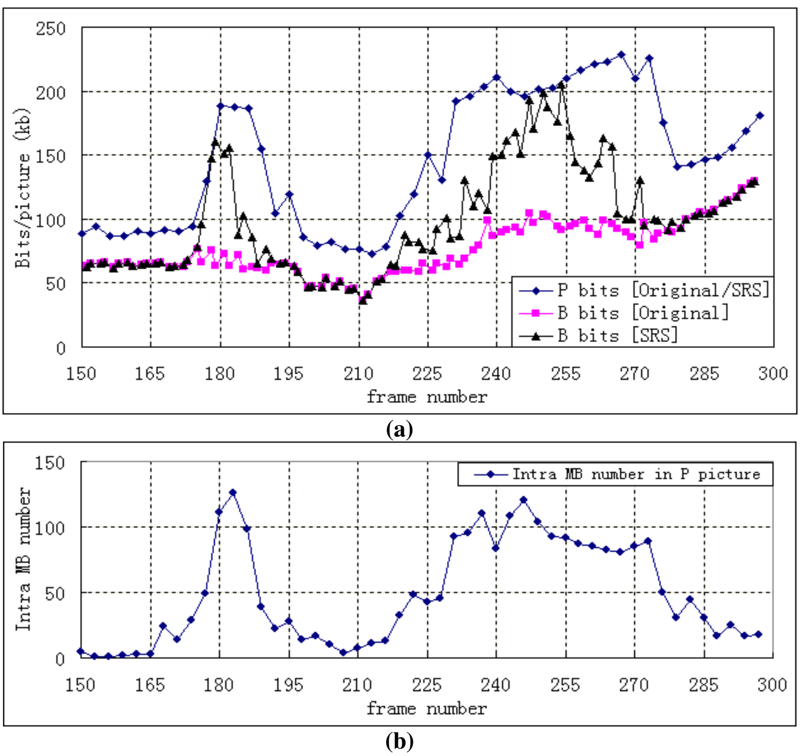

Fig. 3. SRS algorithm testing on *stefan* sequence when QP=24. (a) Coding bits comparisons between SRS algorithm and the original reference software. (b) Intra mode usage statistic for P pictures.

circumstance, still using SRS to cut down the search range of B pictures would only result in worse and worse coding performance. As shown in Fig. 3(a), coding bits per picture of *stefan*, there are two obvious bits increment of B pictures in SRS' curve, which happen at frame 176-191 and 220-278. These two parts exactly match the two periods of sharp motion in the original sequence.

In order to perform SRS under the proper condition, the encoder should decide whether the initial search range suits the motion level of current coded P picture or not. That is to say, if most of the blocks in current P picture can not find enough matched blocks in its references, the adjacent B pictures should avoid processing SRS.

The intra macroblock (MB) number of P picture is an efficient judgment. Generally speaking, the initial search range is suitable for most of the pictures, so intra MB does not often appear during P picture coding; but when the initial search range is too small, inter mode can not find the right matched block, then intra mode is selected for many MB. Fig. 3(b) shows the intra mode statistic for P pictures. Since the search range of P pictures is not changed during the tests in Section 2, the intra MB numbers of each P picture in two corresponding tests are the same all the time. Comparing the coding bits in Fig. 3(a) with the intra mode statistic in Fig. 3(b), we can find that the trend of bits per B picture coded with SRS is as same as the trend of the intra MB number of P picture. So according to the intra MB number of P picture, the encoder can make the decision that whether to perform SRS for next B pictures.

In next part, we are going to elaborate on how to decide the threshold of the intra MB number.

3.2 Adaptive Threshold Decision

First, we define some symbols shown in Fig. 4. w and h ($w \geq h$) are the width and height of the picture; $w_{mb}=w/16$ and $h_{mb}=h/16$ are the MB width and MB height of the picture. Δw and Δh are picture offsets in horizontal and vertical direction caused by the motion. OLD is the remained area from last P picture, and NEW is the new area of the picture caused by the motion. SR_{init} is the initial search range.

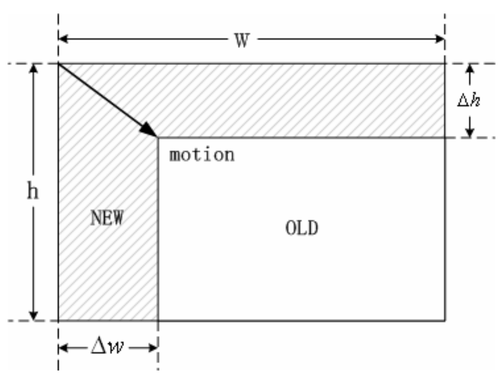

Fig. 4. Demonstration of picture division caused by motion

Then, we assume that all the MBs in NEW area are intra coded because there are no matched blocks in OLD area for them. The number of intra MB in NEW area is *intras*.

Next, we classify the motion into three cases:

(1) $CASE1 = (\Delta w \geq 2SR_{init})$ *or* $(\Delta h \geq 2SR_{init})$

$$intras \geq min(h_{mb} \times \Delta w/16\ ,\ w_{mb} \times \Delta h/16)$$
$$\geq min(h_{mb} \times 2SR_{init}/16\ ,\ w_{mb} \times 2SR_{init}/16)$$
$$\geq h_{mb} \times \lfloor SR_{init}/8 \rfloor = Threshold1$$

In this case, the motion exceeds duple search range, so the right matched position can not be found.

(2) $CASE2 = (CASE1 = FALSE)$ *and* $((SR_{init} \leq \Delta w < 2SR_{init})$ *or* $(SR_{init} \leq \Delta w < 2SR_{init}))$

$intras < Threshold1$, and

$$intras \geq min(h_{mb} \times \Delta w/16\ ,\ w_{mb} \times \Delta h/16)$$
$$\geq min(h_{mb} \times SR_{init}/16\ ,\ w_{mb} \times SR_{init}/16)$$
$$\geq h_{mb} \times \lfloor SR_{init}/16 \rfloor = Threshold2$$

In this case, the motion is less than duple search range but more than search range, so there must exist some MBs whose motion vector (mv_x, mv_y) satisfies $mv_x \geq SR_{ini}$ or $mv_y \geq SR_{ini}$ (motion vector could exceed search range in H.264/AVC, because MV=PredMV+MVD, PredMV and MVD are restricted by search range). We suppose that the number of such MB is at least h_{mb} (*Threshold3*). It should also be one important rule for the algorithm.

(3) $CASE3 = (CASE1 = FALSE)$ *and* $(CASE2 = FALSE)$

SRS algorithm can work well in this case.

Last, we can restrict the search range of B pictures in terms of *Threshold1*, *Threshold2 and Threshold3*.

3.3 Adaptive Search Range Scaling (ASRS)

As a summary, the algorithm description of adaptive search range scaling (ASRS) for B pictures coding is shown below.

```
SWITCH (image type):
{
   CASE I:
      Coding I picture;
      IF (first frame)
      THEN Backward_Scalable = TURE;
      ELSE Forward_Scalable = Backward_Scalable;
      BREAK;
   CASE P:
      Coding P picture;
      Stat. Num_intra : the number of intra MB;
      Stat. Num_mv : the number of MB whose MV exceeds the initial search range
                    SR_init;
```

Forward_Scalable = *Backward_Scalable*;
IF ($Num_{intra} \geq h_{mb} \times \lfloor SR_{init} / 8 \rfloor$)
THEN *Backward_Scalable* = *FALSE*;
ELSE
IF ($Num_{intra} \geq h_{mb} \times \lfloor SR_{init} / 16 \rfloor$ and $Num_{mv} \geq h_{mb}$)
THEN *Backward_Scalable* = *FALSE*;
ELSE *Backward_Scalable* = *TURE*;
BREAK;
CASE B:
IF (*Forward_Scalable* = *TRUE* and *Backward_Scalable* = *TRUE*)
THEN $SR_F = \lceil SR_{init} \times tb / td \rceil$, for the latest forward reference picture;
$SR_F = SR_{init}$, for other forward reference picture;
$SR_B = \lceil SR_{init} \times tp / td \rceil$, for the latest backward reference picture;
ELSE $SR_F = SR_B = SR_{init}$;
Coding B picture;
BREAK;
}

One more attention, we have another reason for collecting Num_{mv} to co-judge with Num_{intra}. There are some MBs coded with intra mode not due to the great motion, however Num_{mv} statistic can reduce such effect.

4 Simulated Results

To evaluate the performance of ASRS and also compare ASRS with SRS, we still choose the same sequences mentioned in Section 2, with 30fps in CIF (352x288) format, including *stefan*, *foreman*, *bus*, *mobile*, *news*, and *paris*. The test conditions are listed in Table 1. Fig. 5 shows that ASRS can make a right decision whether to

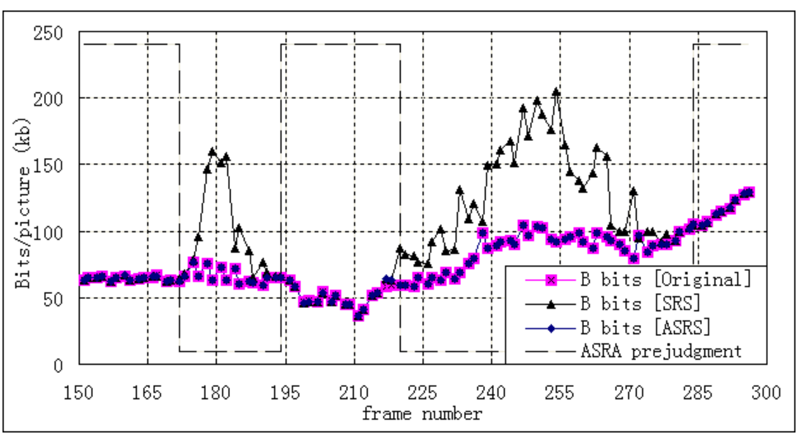

Fig. 5. ASRS vs. SRS testing on *stefan* sequence when QP=24

Table 3. Performance comparisons between ASRS and the reference software (CIF)

Sequences		stefan	foreman	bus	mobile	news	paris
All pict-ures	PSNR gain	-0.002	0	0	0	0	0
	BR saving	-0.354%	0.011%	0.254%	0.094%	0.038%	0.019%
B pict-ures	PSNR gain	-0.004	0	-0.001	0	0	0
	BR saving	-0.724%	0.022%	0.484%	0.172%	0.081%	0.032%
	SA saving	47.84%	41.68%	68.83%	71.30%	68.50%	71.30%

Note: BR denotes bit rate, and SA denotes search area.

Table 4. Performance comparisons between ASRS and the reference software (HD)

Sequences		city	cyclists	harbour	night	shuttl-estart	spinca-lendar
All pict-ures	PSNR gain	-0.001	-0.001	0	0	0	0.001
	BR saving	0.062%	0.065%	0.015%	0.021%	0.009%	0.071%
B pict-ures	PSNR gain	-0.001	-0.001	0	0	0	0.001
	BR saving	0.110%	0.115%	0.034%	0.043%	0.023%	0.120%
	SA saving	71.76%	52.11%	71.76%	25.12%	55.98%	71.61%

Note: BR denotes bit rate, and SA denotes search area.

scale the search range for each B picture or not. And Table 3 shows that ASRS has almost the same performance with the original reference software, however it can reduce average 61.6% search area. When comparing with Table 2, the results of SRS, ASRS improves the performance of high motion sequences a lot, such as for *stefan* and *foreman*.

To evaluate the performance of ASRS under difference picture size and different search range, we choose another six sequences with 60fps in HD (1280x720) format, including *city*, *cyclists*, *harbour*, *night*, *shuttlestart*, and *spincalendar*. The test conditions are: search range 48, QP 27, 30, 35, 40, UMHexagonS fast motion estimation, and the other conditions are same with Table 1. The detailed results are listed in Table 4, which further verifies ASRS algorithm to be efficient for B pictures in both maintaining the picture quality and reducing computational quantity.

5 Conclusions

In this paper, a frame-based adaptive search range scaling algorithm for B pictures is presented, which is suitable for hardware-designed motion estimation. The basic SRS algorithm is based on the relationship between search range of P and B picture. Through intra mode and motion vector statistics of P pictures, the improved ASRS algorithm can detect the motion degree of current coding status. If the motion is not exceed the expected range, search range of the following B pictures can be scaled.

This prejudgment can efficiently both keep the coding performance and reduce the computational complexity. Simulated results show the average search area can be reduced by about 60% from the original reference software.

The future work is to apply the adaptive search range scaling algorithm to the VLSI architecture of our designing real-time HD encoder for further optimization.

References

1. Markus Flierl and Bernd Girod, "Generalized B pictures and the draft H.264/AVC video-compression standard", *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 13, No. 7, pp. 587-597, July 2003.
2. "Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264/ISO/IEC 14 496-10 AVC," in Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVTG050, 2003.
3. Xiangyang Ji, Debin Zhao, Wen Gao, Qingmin Huang, Siwei Ma, and Yan Lu, "New bi-prediction techniques for B pictures coding", *ICME 2004*.
4. Jun-Fu Shen, Tu-Chih Wang, and Liang-Gee Chen, "A novel low-power full-search block-matching motion estimation design for H.263+", *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 11, No. 7, pp. 890-897, July 2001.
5. Lei Deng, Wen Gao, Ming Zeng Hu, and Zhen Zhou Ji, "An efficient hardware implementation for motion estimation of AVC standard", *IEEE Trans. Consumer Electron.,* Vol. 51, No. 4, pp. 1360-1366, Nov. 2005.
6. Zhigang Yang, Wen Gao, and Yan Liu, "Performance-Complexity Analysis of High Resolution Video Encoder and Its Memory organization for DSP Implementation", *ICME2006,* pp. 1261-1264, July 2006.
7. Prabhudev Irappa Hosur, "Motion adaptive search for fast motion estimation", *IEEE Trans. Consumer Electron.*, Vol. 49, No.4, pp. 1330-1340, Nov. 2003.
8. Toru Yamada, Masao Ikekawa, and Ichiro Kuroda, "Fast and accurate motion estimation algorithm by adaptive search range and shape selection", *ICASSP2005*, Vol. 2, pp. ii/897-ii/900, Mar. 2005.
9. Kuo-Liang Chung and Lung-Chun Chang, "A new predictive search area approach for fast block motion estimation", *IEEE Trans. Image Processing*, Vol. 12, No. 6, pp. 648-652, June 2003.
10. Yu-Chan Lim, Kyeong-Yuk Min, and Jong-Wha Chong, "A pentagonal fast block matching algorithm for motion estimation using adaptive search range", *ICASSP2003*, Vol. 3, pp. 669-672, Apr. 2003.