FAST MOTION ESTIMATION AND EDGE INFORMATION INTER-MODE DECISION ON H.264 VIDEO CODING

Yu-Nan Pan and Tsung-Han Tsai

Department of Electronics Engineering, National Central University
No.300, Jhongda Rd., Jhongli City, Taoyuan County 32001, Taiwan, R.O.C.

ABSTRACT

In the upcoming video coding standard, MPEG-4 AVC/JVT/H.264, motion estimation (ME) is allowed to use seven kinds of block sizes to improve the rate-distortion performance. This new feature has achieved significant coding gain compared to coding a macroblock (MB) using fixed block size. However, ME is computational intensive with complexity increasing linearly to the number of allowed block sizes. In this paper, a combined fast motion estimation algorithm with Predict Hexagon Search (PHS) and Edge Information Mode Decision (EIMD) is proposed. EIMD uses the edge information to predict the best matching block size. It can determine the best MB type quickly. The PHS can search the candidate MV efficiently. The analysis results show that the speed improvement of our proposed algorithm over some popular fast motion estimation algorithm is about 2~15 times. Compared with JM10.2, the speed-up is marvelously enhanced as 20~40 times and can still keep good quality as JM10.2.

***Index Terms*—**H.264, fast mode decision, edge difference, edge information mode decision

1. INTRODUCTION

H.264 is a new international video coding standard of ITU-T and jointly made by ITU-T Video Coding Experts Group (VCEG) and ISO/IEC MPEG Video Group, named as Joint Video Team (JVT). The main goals of this standardization are to develop a simple and straightforward video coding design with enhanced compression performance. Comparing to MPEG-4 advanced simple profile, up to 50% of bit-rate reduction can be achieved. Such improvement comes from the prediction part [1], [2], [3]; it basically includes the ME at quarter-pixel with variable block sizes and multiple reference frames. With these novel techniques, the prediction errors are largely reduced.

There are many attractive characteristics of the ME on H.264. On the other hand, because of these characteristics, the complexity of H.264 is very high. In order to overcome this defect, many fast BMAs are have been proposed. Although the fast search algorithm can save a lot of search points, it is not enough to implement for real time constrain. Because the H.264/AVC inter-prediction is performed for different size such as 16x16, 16x8, 8x16, 8x8, 8x4, 4x8 and 4x4, for each MB all of the sizes are estimated and the one that leads to the least RD-Cost is selected. Nevertheless, this optimal decision is achieved at the expense of high computational complexity.

In this paper we propose a fast motion estimation algorithm. It combined with our previous work PHS [4] and EIMD algorithm. It can easily determine the candidate mode from the seven MB types and search the best candidate MV. This paper is organized as follows. In Section 2, the characteristics of MB types are analyzed. Section 3 describes the fast inter-mode decision algorithm and PHS. Section 4 presents the simulation results. Finally, Section 5 draws the conclusions.

2. CHARACTERISTICS OF MB TYPES

2.1. Tree Structure Motion Compensation

In order to achieve the best coding efficiency, H.264 uses Rate Distortion Optimization (RDO) to get the best coding result in terms of minimizing bit-rate and best quality. ME for each 16x16 macroblock can be performed using seven different kinds of block sizes. Each macroblock may be spilt into 4 kinds of sizes: 16x16, 16x8, 8x16, and 8x8. Each of the sub-divided regions is a macroblock partition. If the 8x8 mode is chosen, each of the four 8x8 may be spilt into 4 kinds of sizes: 8x8, 8x4, 4x8, and 4x4. The reason to use seven different kinds of block size in H.264 inter-prediction is to get the object movement more accurately. Fig. 1 shows an example of the optimum choice of MB partitions. It is observed that there exist a lot of homogenous regions which belong to the same video object or background, and there also exist a lot of edges. When the object or background moves, the homogeneous region may moves in a similar way. Thus, it is seldom to be split into small MB type. In addition, when edges exist in a MB, the edges may belong to the same object or not. Even if the edges belong to the same object they may move in the different ways. Thus, it is seldom to use large MB type. Generally, homogeneous regions existed in a MB are usually coded by large MB type. A MB contained the edge is more likely to be coded by smaller MB type.

2.2. Characteristics of Multiple Block Sizes

Table 1 illustrates the statistics of selected MB mode with different quantization parameter (QP) in percentage. From Table 1 (a), we can see that for QP=40, the percentage to select the block size 16x16 is about 60~77%. The percentage to select the block size 8x8 or the sub partition of block size 8x8 is about 2~9%. For QP=30, the percentage to select the block size 16x16 is about 38~52%. The percentage to select the block size 8x8 or the sub partition of block size 8x8 is about 12~34%, as show in Table 1 (b).

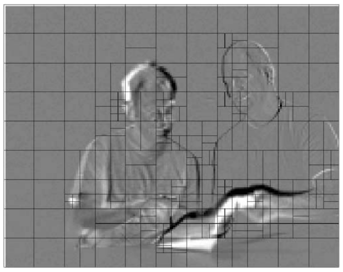


Fig. 1. The optimum choice of MB partitions [3].

Form Table 1 (c), as QP=20, the percentage to select block size 16x16 is about 20~44%. The percentage to select block size 8x8 or the sub partition of block size 8x8 is about 33~62%. From the analysis, we can see that the percentage to select block size 16x16 is rose, when changing the QP from 20 to 40. On the other hand, the percentage to select block size 8x8 or the sub partition of block size 8x8 is descended, when changing the QP from 20 to 40. It is because the Lagrangian cost function. The second term (Rate) of the Lagrangian cost function related to MB modes increases with which makes it much more difficult for low bit rate situations to select the partition modes with more side information.

3. EDGE INFORMATION MODE DECISION AND PHS ALGORITHM

3.1. Predict Hexagon Search (PHS) Algorithm

The well-known Full Search (FS) algorithm exhaustively evaluates all possible candidate motion vectors over a predetermined neighborhood search window to find the global minimum block distortion position. Although FS can get the best matching blocks but it expenses a high computational complexity. Thus, In order to overcome this defect, many fast BMAs are developed.

For the fast search algorithm, we developed a Predict Hexagon Search (PHS) [4]. PHS is a computational efficient algorithm, it depends on the characteristics of motion vector distribution; it can predict the object movement in horizontal and vertical direction with a novel search pattern. It can save more than 25% searching time compared with other fast search algorithm.

3.2. Edge Detection and Edge Difference

A region is homogeneous if the textures in the region have similar spatial property. There exist many techniques for detecting homogeneous regions in an image. Video object boundary usually exhibits strong edges, so edge information [9] is a way to determine homogenous region.

Here we use the Sobel operator to calculate the edge value. For a pixel $P_{i,j}$, in a luminance picture, we define the edge vector as

$$dx_{i,j} = P_{i-1,j+1} + 2\times P_{i,j+1} + P_{i+1,j+1} - P_{i-1,j-1} - 2\times P_{i,j-1} - P_{i+1,j-1} \quad (1)$$

$$dy_{i,j} = P_{i+1,j+1} + 2\times P_{i+1,j} + P_{i+1,j-1} - P_{i-1,j+1} - 2\times P_{i-1,j} - P_{i-1,j-1} \quad (2)$$

Where $dx_{i,j}$, $dy_{i,j}$ represent the degrees of differences in the vertical and horizontal directions. The edge value for each block size 4x4 is

$$EdgeValue_{4x4} = \sum_{i,j=0}^{3} (|dx_{i,j}| - |dy_{i,j}|) \quad (3)$$

Table 1. Statisics of Selected MB Modes in Percentages. (a) QP=40. (b) QP=30. (c) QP=20.

	16x16	16x8	8x16	8x8
Coastguard (352x288)	76%	10%	12%	2%
Stefan (352x288)	60%	19%	13%	8%
Silent (352x288)	68%	11%	19%	2%
Weather (352x288)	60%	13%	18%	9%
Tennis (176x144)	73%	13%	11%	3%
Foreman (176x144)	77%	11%	8%	4%
QP=40, Search Range=-16~15				

(a)

	16x16	16x8	8x16	8x8
Coastguard (352x288)	52%	18%	18%	12%
Stefan (352x288)	38%	14%	14%	34%
Silent (352x288)	47%	15%	21%	17%
Weather (352x288)	40%	15%	15%	30%
Tennis (176x144)	52%	17%	13%	18%
Foreman (176x144)	46%	15%	20%	19%
QP=30, Search Range=-16~15				

(b)

	16x16	16x8	8x16	8x8
Coastguard (352x288)	44%	10%	11%	35%
Stefan (352x288)	33%	16%	12%	39%
Silent (352x288)	32%	11%	14%	33%
Weather (352x288)	41%	9%	10%	40%
Tennis (176x144)	17%	12%	9%	62%
Foreman (176x144)	20%	10%	14%	56%
QP=20, Search Range=-16~15				

(c)

Because of the correlation between block size 4x4, 8x8 and 16x16, we can easily get the edge values of block size 8x8 and 16x16 by accumulating the edge value of block size 4x4. The edge value of block size 16x16 can be accumulated by sixteen $EdgeValue_{4x4}$, as

$$EdgeValue_{16x16} = \sum_{i=0}^{15} EdgeValue_{4x4}(i) \quad (4)$$

The edge value of block size 8x8 can be accumulated by four $EdgeValue_{4x4}$, as

$$EdgeValue_{8x8}(0) = \sum_{i=0}^{3} EdgeValue_{4x4}(i) \quad (5)$$

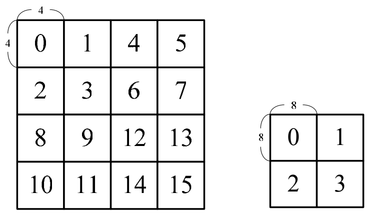
$$EdgeValue_{8x8}(1) = \sum_{i=4}^{7} EdgeValue_{4x4}(i) \quad (6)$$

$$EdgeValue_{8x8}(2) = \sum_{i=8}^{11} EdgeValue_{4x4}(i) \quad (7)$$

$$EdgeValue_{8x8}(3) = \sum_{i=12}^{15} EdgeValue_{4x4}(i) \quad (8)$$

The block number is shown in Fig. 2. If the edge value of the block is less than the threshold value, it is classified as homogeneous block, and is suitable for this kind of MB type. Otherwise it is a non-homogeneous block, and should be spilt into smaller MB type. Thus, we can use the edge value to determine whether the block size 16x16, 8x8 and 4x4 is the best MB type or not. Besides, if the $EdgeValue_{16x16}$ is larger than Th_{16x16}, and all the four $EdgeValue_{8x8}$ are smaller than $Th_{separate\_lar}$, we will use edge difference to determine whether this block is suitable for 16x8 and 8x16 mode or not.

If all the $EdgeValue_{16x16}$ and $EdgeValue_{8x8}$ are larger than the threshold value, but all the four $EdgeValue_{4x4}$ which belong to one 8x8 sub-block are smaller than $Th_{separate\_small}$, we will use edge difference to determine whether this block is suitable for 8x4 and 4x8 mode or not.



(a) (b)

Fig. 2. (a) Block number for each block with the size is 4x4. (b) Block number for each block with the size is 8x8.

Edge value means the degree of edge existing in a block. Thus, if the edge difference between two blocks is small, that means this two blocks may have same stuff. Here we propose the edge difference value for 16x8, 8x16, 8x4 and 4x8, as the equations below, and $x \in 0 \sim 3$

$$EdgeDifference_{16x8} = |EdgeValue_{8x8}(0) - EdgeValue_{8x8}(1)| + |EdgeValue_{8x8}(2) - EdgeValue_{8x8}(3)| \quad (9)$$

$$EdgeDifference_{8x16} = |EdgeValue_{8x8}(0) - EdgeValue_{8x8}(2)| + |EdgeValue_{8x8}(1) - EdgeValue_{8x8}(3)| \quad (10)$$

$$EdgeDifference_{8x4}(x) = |EdgeValue_{4x4}(4x) - EdgeValue_{4x4}(4x+1)| + |EdgeValue_{4x4}(4x+2) - EdgeValue_{4x4}(4x+3)| \quad (11)$$

$$EdgeDifference_{4x8}(x) = |EdgeValue_{4x4}(4x) - EdgeValue_{4x4}(4x+2)| + |EdgeValue_{4x4}(4x+1) - EdgeValue_{4x4}(4x+3)| \quad (12)$$

The $EdgeDifference_{16x8}$ and $EdgeDifference_{8x4}(x)$ mean the edge value difference in the horizontal way. The $EdgeDifference_{8x16}$ and $EdgeDifference_{4x8}(x)$ mean the edge value difference in the vertical way. If the edge difference in the horizontal way is smaller than the vertical way, that means the stuff in the horizontal way is more similar than the vertical way, and is suitable to use 16x8 or 8x4 MB types. Otherwise it is suitable to use 8x16 or 4x8 MB types.

3.3. Flow Chart for Edge Information Mode Decision (EIMD) Algorithm

As mentioned previously, if the MB has a lot of homogeneous regions, it is more likely using larger MB types. Otherwise, it should be spilt into smaller MB types. We can easily adopt the edge value and edge value difference to determine which MB type should be used. We develop a flow chart as follow:

Step 1. Calculate the $EdgeValue_{4x4}$, and accumulate the $EdgeValue_{4x4}$ to four $EdgeValue_{8x8}$ and $EdgeValue_{16x16}$.

Step 2. If the $EdgeValue_{16x16}$ is smaller than Th_{16x16}, the best MB type is 16x16. Otherwise the flow goes to step 3.

Step 3. If all of the $EdgeValue_{8x8}$ are smaller than $Th_{separate\_lar}$, the flow goes to the step 4. Otherwise the flow goes to the step 5.

Step 4. Calculate $EdgeDifference_{16x8}$ and $EdgeDifference_{8x16}$. If the $EdgeDifference_{16x8}$ is larger than $EdgeDifference_{8x16}$, the best MB type is 8x16. Otherwise the best MB type is 16x8.

Step 5. If the $EdgeValue_{8x8}$ is smaller than Th_{8x8}, then the best MB type for this sub-block is 8x8, and the flow goes to step 8. Otherwise the flow goes to step 6.

Step 6. If all of the four $EdgeValue_{4x4}$ which belong to one 8x8 sub-block are smaller than $Th_{seperate\_small}$, then the flow goes to step 7. Otherwise the best MB type for this sub-block is 4x4, then the flow goes to step 8.

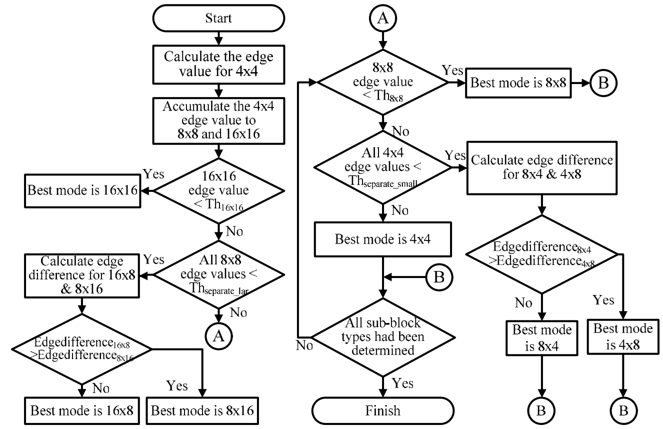


Fig. 3. Flow chart for proposed EIMD algorithm.

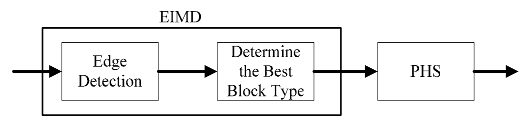


Fig. 4. System Flow for the fast motion estimation.

Step 7. Calculate $EdgeDifference_{8x4}$ and $EdgeDifference_{4x8}$. If the $EdgeDifference_{8x4}$ is larger than the $EdgeDifference_{4x8}$, the best MB type for this sub-block is 4x8 and the flow goes to step 8. Otherwise the best MB type for this sub-block is 8x4 and the flow goes to step 8.

Step 8. If all sub-block types had been determined, then it will finish. Otherwise it will start to determine the best MB type for next sub-block. Fig. 3 illustrates the overall scheme of proposed EIMD algorithm.

As we mentioned above, it has a correlation between QP and the type of block size. As QP is large, the percentage to select large kinds of block types is large. On the other hand, as QP is small, the percentage to select smaller kinds of block types is large. In our algorithm the Th_{16x16} is set to $1200xQP$ and the Th_{8x8} is set to $750xQP$. The $Th_{seperate\_lar}$ is set to $600xQP$ and the $Th_{seperate\_small}$ is set to $150xQP$.

3.4. System Flow for the Fast Motion Estimation Algorithm

With the PHS and EIMD algorithm, we develop a system flow for the fast motion estimation. As shown in Fig.4, at first, the proposed algorithm will enter into the edge detection and calculate the edge value. After the edge value has been calculated, the EIMD will use edge value and edge information to determine the best block mode. After the best block mode has been determined, it will enter into the PHS and search the best candidate MV.

4. SIMULATION RESULT

The proposed fast motion estimation algorithm is simulated using 3 types of video formats (SD, CIF, QCIF) and total 12 popular video sequences are used. These sequences consist of different degrees and types of motion contents, and some of them have been classified by [5]. The search window is ±16, the number of reference frames is 5, and the number of block types is 7. We construct the proposed algorithm and reconstruct the compared algorithms in JM.10.2 [6]. The

Table 2. Run Time(s) of ME and Speed-Up Ratio with Different Methods and Different Image Sequences.

	Night (720x480)		Sailormen (720x480)		Crew (720x480)		Harbour (720x480)		Coastguard (352x288)		Stefan (352x288)		Silent (352x288)		Weather (352x288)		Tennis (176x144)		Foreman (176x144)		Salesman (176x144)		Mobile (176x144)	
	time	Speed-up	time	Speed-up	time	Speed-up	time	Speed-up	time	Speed-up	time	Speed-up	time	Speed-up	time	Speed-up	time	Speed-up	time	Speed-up	time	Speed-up	time	Speed-up
JM10.2	4114.3	36.0	4244.3	40.0	4077.3	37.8	3541.0	27.5	1235.0	39.3	958.2	26.1	667.3	22.7	511.6	21.4	264.9	32.0	195.3	20.5	145.5	21.7	261.7	24.6
[7]	1528.1	13.5	1433.5	12.1	1436.1	13.9	1264.9	9.8	460.2	13.4	603.4	16.5	247.7	8.4	289.9	12.2	102.8	12.3	74.2	7.8	53.0	7.9	159.8	15.0
UMHexagoS	856.4	7.6	876.0	8.0	850.5	8.2	831.0	6.5	257.9	7.5	208.2	5.7	162.2	5.5	146.1	6.1	52.6	6.3	47.7	5.0	41.2	6.1	54.1	5.0
Simplified UMHexagonS	665.0	6.0	668.7	6.0	578.5	5.6	660.1/	5.1	181.1	5.3	173.7	4.8	91.2	3.1	71.8	3.3	38.8	4.7	35.4	3.7	19.5	3.0	41.9	3.9
EZPS	760.0	6.7	805.8	7.4	815.3	7.9	836.6	6.5	237.0	6.9	216.4	5.9	197.2	6.7	185.1	7.7	56.1	6.8	53.8	5.7	48.7	7.3	55.0	5.1
FMS+MDRPS	268.2	2.4	261.7	2.4	269.2	2.6	260.7	2.0	91.1	2.6	86.2	2.4	91.7	3.1	89.8	3.4	25.3	3.1	25.8	2.7	25.1	3.7	25.1	2.4
Proposed	112.7		109.1		103.7		128.7		34.5		36.7		29.5		23.9		8.3		9.5		6.7		10.7	

Table 3. Average PSNR per Frame with Different Methods and Different Image Sequences.

	Night	Sailormen	Crew	Harbour	Coastguard	Stefan	Silent	Weather	Tennis	Foreman	Salesman	Mobile
JM10.2	42.4	41.3	42.2	41.4	40.9	41.9	41.8	43.6	41.3	41.4	41.9	40.6
[7]	42.4	41.2	42.1	41.3	40.8	41.8	41.7	43.5	41.0	41.2	41.8	40.4
UMHexagoS	42.4	41.3	42.2	41.4	40.8	41.9	41.8	43.6	41.2	41.4	41.9	40.5
Simplified UMHexagonS	42.4	41.2	42.1	41.4	40.8	41.8	41.7	43.6	41.2	41.3	41.9	40.5
EZPS	42.4	41.3	42.2	41.4	40.8	41.9	41.8	43.6	41.2	41.4	41.9	40.5
FMS+MDRPS	42.4	41.2	42.2	41.4	40.4	41.6	41.7	43.4	41.0	41.2	41.8	40.4
Proposed	42.3	41.1	42.1	41.3	40.7	41.7	41.6	43.5	40.9	41.1	41.8	40.3

UMHexagonS, Simplified UMHexagonS and EZPS are the original construction in JM10.2.

Table 2 illustrates the run time of ME and the speed-up ratio with different methods. Based on the simulation results in Table 2, the run time of ME by the proposed algorithm is much smaller than JM10.2. Compared with other algorithms, our proposed algorithm requires the least run time. For the low motion sequences, such as "Salesman", JM10.2 needs 145.5(s). [7] needs 53.0(s), UMHexagonS needs 41.2(s) , Simplified UMHexagonS needs 19.5(s), EZPS needs 48.7(s), Fast Mode Selection (FMS)+Multi-Directional Rood Pattern Search(MPRPS) [8] needs 25.1(s). However, our proposed algorithm only needs 6.7(s). For the medium motion sequences, such as "Stefan", JM10.2 needs 958.2(s). [7] needs 603.4(s), UMHexagonS needs 208.2(s) , Simplified UMHexagonS needs 173.7(s), EZPS needs 216.4(s), Fast Mode Selection (FMS)+Multi-Directional Rood Pattern Search(MPRPS) [8] needs 86.2(s). Our proposed algorithm only needs 36.7(s). For high resolution sequences, as "Sailormen", JM10.2 needs 4244.3(s). [7] needs 1433.5(s), UMHexagonS needs 876.0(s) , Simplified UMHexagonS needs 668.7(s), EZPS needs 805.8(s), Fast Mode Selection (FMS)+Multi-Directional Rood Pattern Search(MPRPS) [8] needs 261.7(s). Our proposed algorithm only needs 109.1(s). Based on the simulation results, we can see that no matter the sequence is low motion or high motion, our proposed algorithm can save a lot of run time in comparison with these existing techniques. For the speed-up ratio, compare with the original JM10.2, our proposed algorithm can has 20~40 times. Even compared with other fast motion estimation algorithm, our proposed algorithm can has 2~15 times. Table 3 shows the average PSNR per frame. The proposed EIMD has almost the same average PSNR compared with [7], FMS, and JM10.2.

5. CONCLUSION

H.264 is the latest standard for image compression which is targeted for high video quality and low bit rate applications. Because the ME is the critical part in H.264, in this paper, we propose a new combined fast motion estimation algorithm with the PHS and EIMD. The proposed EIMD uses the edge information to predict the homogenous region in a MB. It applies some simple equations to calculate the edge difference to determine the same stuff region and can easily predict the best matching MB type. Our previous work, PHS, can search the best candidate MV efficiently. Obviously, it can save lots of ME run time in the lower resolution sequences. Even for the higher resolution sequences, our method can trace the high motion movement well and still maintain the high quality. The speed improvement of our proposed algorithm over some popular fast motion estimation algorithm is about 2~15 times. Compared with JM10.2, the speed-up is about 20~40 times.

6. REFERENCES

[1] *Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification*, ITU-T Rec. H.264 and ISO/IEC 14 496-10 AVC, Joint Video Team, Mar. 2003.

[2] "H.264/MPEG-4 Part 10: Overview," H.264/MPEG-4 Part 10 White Paper, http://www.vcodex.com.

[3] "H.264/MPEG-4 Part 10: Inter Prediction," H.264/MPEG-4 Part 10 White Paper, http://www.vcodex.com.

[4] T. H. Tsai and Y. N. Pan, "A Novel 3-D Predict Hexagon Search Algorithm for Fast Block Motion Estimation on H.264 Video Coding," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 16, no. 12, pp. 1542-1549, Dec. 2006.

[5] *MPEG-4 Video Verification Model Version 18.0*, ISO/IEC JTC1/SC29/WG11 N 3908, Jan. 2001.

[6] Joint Video Team software JM10.2, 2006.

[7] D. Wu, F. Pan, K. P. Lim, S. Wu, Z. G. Li, X. Lin, S. Rahardja, and C. C. Ko, "Fast Intermode Decision in H.264/AVC Video Coding," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 15, no. 7, pp. 953-958, July 2005.

[8] G. L. Li, M. J. Chen, H. J. Li, and C. T. Hsu, "Efficient Search and Mode Prediction Algorithms for Motion Estimation in H.264/AVC," *in Proc. IEEE Circuits and Systems Conf.*, vol. 6, May 2005, pp. 5481-5484.

[9] Gonzale, R. C. and R. E. Woods 2002 " Digital Image Processing. " 2nd Ed. Prentice Hall, Inc.