VOL. 10, NO. 6, APRIL 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

# AN IMPROVED MOTION ESTIMATION SEARCH ALGORITHM FOR H.264/AVC STANDARD

Vani Rajamanickam[1], Sangeetha Marikkannan[2] and Sharmila Ganesan[2]
[1]Department of Electronics and Communication Engineering, Meenakshi College of Engineering, Anna University, Tamil Nadu, India
[2]Department of Electronics and Communication Engineering, Karpaga Vinayaga College of Engineering and Technology, Anna University, Tamil Nadu, India
Email: rvani18@gmail.com

**ABSTRACT**

The virtual role of science and technology in modern life demands compression in multimedia application as it involves transfer a large amount of data. Motion estimation is one of the most important and complex block of all the existing video coding standards. In the video coding standard H.264/AVC, Motion Estimation is allowed to search multiple reference frames and the ME process is much more complex due to variable block size with quarter pixel accuracy. Therefore, efficient motion estimation algorithm is required to reduce the computational complexity. The improved Search algorithm is suitable for stationary, quasi-stationary and fast moving video sequences and computationally less complex. The results show that the proposed algorithm requires very few number of search points for finding the best matched block with almost negligible loss in video quality. As compared to the existing ME algorithm, the simulated results of the proposed algorithm achieved an average of 11.145 search computations with less time and an average PSNR of 23.41dB for a frame rate of 15fps.

**Keywords:** motion estimation, H.264, search pattern, algorithm, search computations.

## 1. INTRODUCTION

H.264/AVC is one of the most usable video coding standards used in various video processing applications. It has been jointly developed by ITU-T Video Coding Expert Group (VCEG) and ISO/IEC 14496-10 AVC Moving Picture Expert Group (MPEG) in 2003. The H.264 video compression standard has been incorporated into MPEG-4 as MPEG-4 Part10 (ISO/IEC 14496-10). This means MPEG-4 now has two video compression standards available. However, these two video compression standards are non-inter operable, with each standard using different methods to compress and represent the data i.e. an MPEG-4 Part 10 decoder cannot decode an MPEG-4 Part 2 bit stream and vice versa. H.264 reduces the bit-rate 39%, 49% and 64% as compared to MPEG-4, H.263 and MPEG-2, respectively. The improvement in coding performance is due to variable block size motion estimation with quarter-pixel accuracy, multiple reference frame and improved prediction mode [3, 9, 11]. Table-1 shows some of the key features related to H.264/MPEG-4 (AVC). Block based motion estimation is used in all the existing MPEG-X and H.26X video coding standards due to ease in implementation and good trade-offs between complexity and accuracy. It is used for removal of the temporal redundancy between the current frame and reference frames. Full search block based motion estimation is the best Motion Estimation (ME) algorithm for finding the best possible matched block in the search window of the reference frame because it searches each and every possible blocks in the search window, but it consumes more time. Variable block size and multiple reference frames in H.264/AVC video coding standard add more complexity in ME process [4, 14]. There are a number of evaluation criteria for evaluating the best match between the current frame and the reference frames. The Sum of Absolute Difference (SAD) sums the absolute values of differences between pixels in the current frame and the corresponding pixels in the reference frame. This metric is the norm of difference image and the lowest SAD estimates the best position of prediction within the search image. The SAD metric is calculated as shown in Equation (1).

$$d_1(I_j, P) = \sum_{i=1}^{n} \left| I_{i,j} - P_i \right| \qquad (1)$$

where Ii, j denotes current frame and Pi denotes the reference frame.

**Table-1.** Key features in H.264 standard.

| Category | H.264 |
|---|---|
| Bit rate | 40-50% bit rate reduction quality compared to MPEG-2 |
| Specification | Supports up to 4k (4096x2304) Supports up to 59.9 fps |
| Compression Model | Hybrid Spatial-temporal prediction Model |
| Block structure | Macro block structure with maximum block size supporting of 16x16 |
| Intra prediction directional modes | 9 Directional modes |
| Improvements | Led growth HD content for broadcast and online |

VOL. 10, NO. 6, APRIL 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

Further, variable block size with quarter pixel accuracy ME consumes more than 90% time of the H.264/AVC baseline encoder. Several techniques such as reduction of bit width, MV prediction, successive elimination, hierarchical search and reduction of the checking points were proposed by researchers to accelerate the ME process. Nowadays, the traffic caused by video applications in network is high due to the growing popularity of High Definition (HD) Video and beyond HD videos. Personal computers, tablets and even mobile devices need to receive and display HD videos, and this become a severe challenge on today's network. Network traffic can be minimized by increasing video compression in all existing applications and issues of H.264/AVC. It particularly focuses on two key issues (a) increased video resolution (b) improve the coding performance. The most essential part in video coding standards like MPEG-2, H.264/AVC is Motion Estimation (ME). But ME consumes more than 50% of coding complexity or time to encode. To reduce the computational complexity, the time required for Motion Estimation should be reduced. In [5], experiments are performed on various types of video sequences using full search ME algorithm and found 74.76% motion vector lie on the cross-center-biased portion. So, instead of using LDSP as the initial step, they used Cross-Search Pattern (CSP) as the initial step. The CSP searches total nine positions, one at origin and eight at (0, ±1), (0, ±2), (±1, 0) and (±2, 0)) positions around the origin of the search window. In [6], small cross search pattern is used as the initial step for further speeding up the CDS algorithm with negligible loss in visual quality. But these algorithms are complex because they use more number of search patterns like small cross search, CSP at origin and LDSP at the origin [7][8]. The proposed improved search algorithm is suitable for stationary, quasi-stationary and fast moving video sequences and computationally less complex. The simulation is done using MATLAB and the results show that the algorithm proposed reduces the computational complexity than the existing algorithms.

The paper is organized as follows: Section II explains about the existing search methodologies. In section III, the proposed algorithm for motion estimation has been analysed. Section IV gives the simulation results of the existing and the proposed motion estimation algorithm using various search patterns. The last section will conclude with future work to be done.

## 2. EXISTING SEARCH METHODOLOGIES

This section explains about the existing search patterns such as Diamond Search (DS), Cross Search and Hexagon Search (HS) algorithm. The search patterns are used to find the global minimum points. The search patterns come in different forms such as small cross, diamond grids. These patterns are used so as to reduce the computational complexity for finding Motion Vector (MV).

### A. Diamond search pattern

As the video sequences are naturally the combination of stationary, slow moving and fast moving blocks, so there is a need to develop a single algorithm which is suitable for all types of video sequences. DS is one of the most popular fast ME algorithm. It gives good PSNR value with less number of search points as compared to other fast ME algorithms like Three Step search (TSS) [13], New Three Step Search (NTSS), Hexagon based Search (HS) [10] etc. DS algorithm has two types of diamond search patterns: Large diamond-search pattern (LDSP) and Small diamond search pattern (SDSP) as shown in Figure-1. LDSP searches nine points while SDSP searches only five points. The first step is the LDSP and it continues until either the motion search reaches the boundary of the search range or best matched block is found on the center of the LDSP. After finding best matched block using LDSP, SDSP is used for further refinement and it executes only once [1, 12].

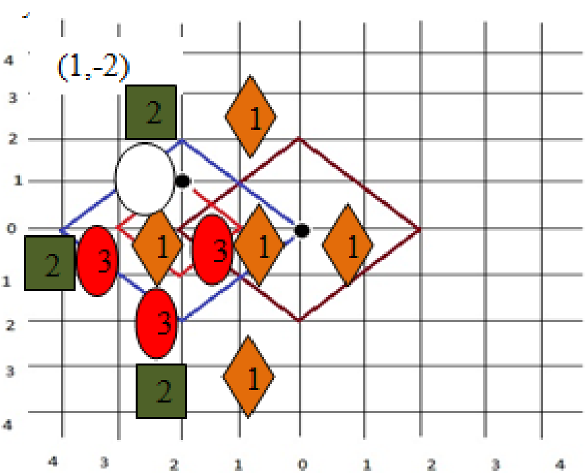


**Figure-1.** Diamond search.

### B. Hexagon search pattern

Hexagon pattern save around 23% of search window when compared to diamond pattern [2]. There are two basic hexagon search pattern. Horizontal hexagons is good for horizontal motion and have poor performance for vertically moving objects, whereas Vertical hexagons is good for vertical motion and have poor performance for horizontally moving objects. In order to cope with these patterns without loss of performance, we consider rotating hexagon patterns as it incorporates both vertical and horizontal motion. Hexagon pattern helps reducing the computational complexity to a great extent, but there is still complexity in coding. Figure-2 shows the hexagon search pattern.

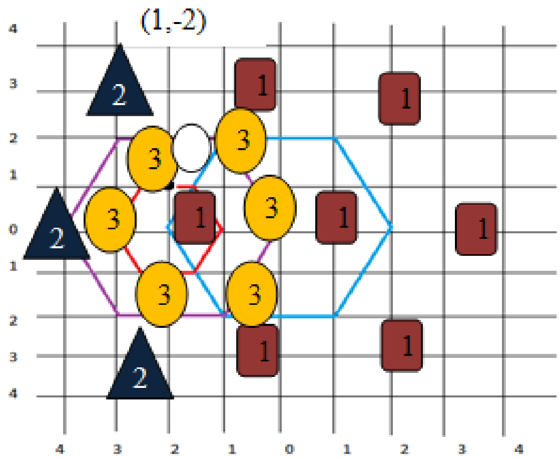


**Figure-2.** Hexagon pattern.

## 3. PROPOSED ALGORITHM

The Improved Search algorithm searches for the best motion vectors in a course to fine search pattern. The new ME algorithm is the combination of small cross search pattern and diamond search pattern. The proposed algorithm is suitable for all types of video sequences. It uses two half way stop techniques: one after initial step and the other after second step. The small cross pattern is used as the initial step, which is good for stationary and quasi stationary blocks. If block is neither stationary nor quasi stationary, LDSP is used for further refinement which gives better performance for fast moving objects. The algorithm is suitable for all types of video sequences. It uses small cross in initial step, and gains the advantage of speed up in the algorithm. The step by step procedure of the proposed algorithm is shown in Figure-3 and is explained as follows:

**Step-1:** Initially we assume the best match is at location (1,-2). The first step of the algorithm is small cross search pattern. It uses five points of the small cross-search pattern initially and finds the minimum block distortion using SAD metric. If minimum cost is found at the center of the small cross-search pattern, the search stops. Else go to step 2. Here the best match is at (0,-1).

**Step-2:** The Cross Search with ±1 distance is used to find the point with minimum distortion by using the minimum point from the previous step as the center point. It uses 3 points for its evaluation i.e., pixels at the location (-1,-1), (0, - 2), (1,-1). If the center is found to be minimum cost, stop the search else go the next step. The best match is at (0,-2).

**Step-3:** With the minimum cost from the step-2 as the center, the search starts. This step uses large diamond pattern and starts to find the best match. Here 1 pixel is reused and 3 pixels from the neighbor are used to search its best match. The match is at the same location. If the minimum cost lies in the center, then go to step-4.

**Step-4:** Finally small diamond search pattern is used with minimum point from the previous search as its center. Here 3 pixels from the nearest neighbour decide the final cost. This search gives the final best matched block for the current block with in the search window of the reference frame. The algorithm uses 14 points to find the match at the exact location.

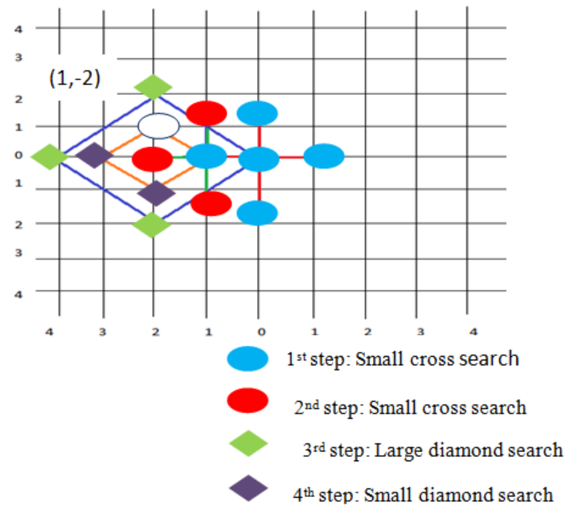


**Figure-3.** Steps involved in the proposed algorithm.

The steps involved in the algorithm can be understood by means of the flowchart shown in Figure-4. The improved search algorithm gives better performance for fast moving objects and it is suitable for all types of video sequences.

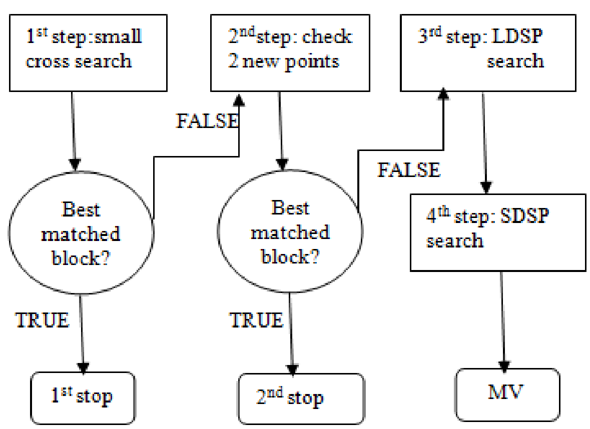


**Figure-4.** Flow chart of the proposed algorithm.

## 4. SIMULATION RESULTS

The result of motion estimation is done using the MATLAB environment. Peak Signal to Noise Ratio (PSNR) is used to evaluate the performance of the proposed algorithm. PSNR is easily defined via the Mean Squared Error (MSE) as given in equations (2) and (3). We have performed experiments on various video sequences for both existing and the proposed search ME algorithm. Parameters used are block size16x16, metric SAD and at frame rate 10fps, 15fps.

$$MSE = \frac{\sum_{M,N} [I_1(m,n) - I_2(m,n)]^2}{M * N} \qquad (2)$$

$$PSNR = 10 \log_{10}\left(\frac{R^2}{MSE}\right) \qquad (3)$$

where M and N denotes the number of rows and columns in the input images. $I_1$ (m, n) represents the original image pixel values and $I_2$ (m, n) represents the reconstructed image pixel values. PSNR is then computed in decibels between two images and R is the maximum fluctuation in the input image data type. A higher PSNR indicates that reconstruction is of higher quality. YUV test sequences named 'MOBILE', 'BQMALL' of resolution 832x480 and 'PEOPLE ON STREET' AND 'TRAFFIC' of resolution 2560 x 1600 is used for analysis as shown in Figure-5. The YUV sequence is converted to sequence of frames and is read from storage location using MATLAB.


**Figure-5.** Test sequences.

Motion estimation is done for a total of 10 frames per second and 15 frames per second. The average time, PSNR and computations is calculated with different search patterns. The existing and the proposed algorithms were simulated with different video sequences. The motion estimation time is calculated for each frame of the sequence and the motion estimation time compared with different patterns. We consider the previous frame as the reference frame with block size 16×16. The results show that the proposed pattern has lower ME time at all points compared to other search patterns. The proposed method has minimum search points and also provides low computation than other search patterns. Table-2 and Table-3 provides computations obtained for different algorithms such as DS, HS, and New Hexagon Search (NHEXS) [10] and the proposed algorithm.

**Table-2.** Motion estimation computations for different search pattern for frame rate 10fps.

| Test sequences | NHEXS | DS | HS | Proposed |
|---|---|---|---|---|
| Mobile | 13.88 | 15.86 | 11.92 | 11.72 |
| BQmall | 11.27 | 14.94 | 11.31 | 11.11 |
| Traffic | 11.35 | 15 | 11.58 | 10.8 |
| Mobisode | 13.03 | 18.15 | 12.66 | 11.79 |
| Average | **12.383** | **15.988** | **11.868** | **11.355** |

**Table-3.** Motion estimation computations for different search pattern for frame rate 15fps.

| Input sequences | NHEXS | DS | HS | Proposed |
|---|---|---|---|---|
| Mobile | 12.67 | 14.16 | 10.93 | 10.87 |
| BQmall | 11.27 | 15.1 | 11.33 | 11.09 |
| Traffic | 11.35 | 15.01 | 11.58 | 10.82 |
| Mobisode | 13.09 | 18.21 | 12.7 | 11.8 |
| Average | **12.095** | **15.62** | **11.635** | **11.145** |

Table-4 gives the ME time and computations for Mobile sequence at different block sizes 16x16, 4x4 and 8x8. This show the algorithm works for different block sizes at a frame rate of 10fps. Table-5 provides the PSNR obtained for different search patterns for various sequences. It is found that the proposed search algorithm has better PSNR when compared to other algorithms.

**Table-4.** ME Time, computations of variable block size obtained for mobile sequence.

| Variable block size | No. of frames/sec | Time in seconds | Computations |
|---|---|---|---|
| 4x4 | 10 | 102.84 | 11.72 |
| 8x8 | 10 | 29.13 | 11.31 |
| 16x16 | 10 | 10.52 | 10.85 |

**Table-5.** PSNR obtained for different search patterns.

| Sequence | NHEXS | DS | HS | Proposed |
|---|---|---|---|---|
| Mobile | 14.08 | 15.96 | 15.16 | 19.98 |
| BQmall | 22.22 | 22.86 | 23.01 | 23.01 |
| Traffic | 24.05 | 24.37 | 24.48 | 24.5 |
| Mobisode | 24.01 | 22.91 | 23.25 | 26.15 |
| Average PSNR | **21.09** | **21.525** | **21.475** | **23.41** |

The simulated results of the proposed algorithm achieved an average of 11.5325 search computations with less time and an average PSNR of 23.41dB for frame rate of 10fps. Figure-6 and Figure-7 shows the plot of computation variations and PSNR for a frame rate of 15fps in different test sequences. It is understood that the proposed algorithm has less computations and less ME time.

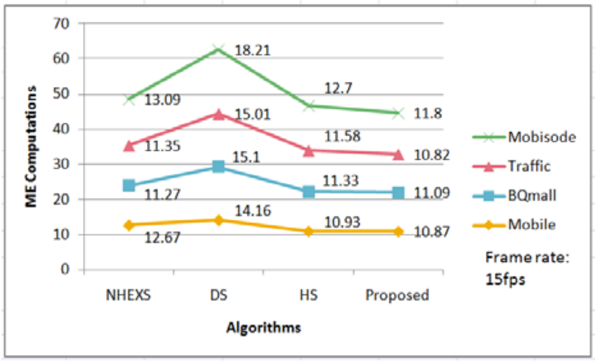


**Figure-6.** Comparison of different algorithms in terms of ME time at a frame rate 15fps.

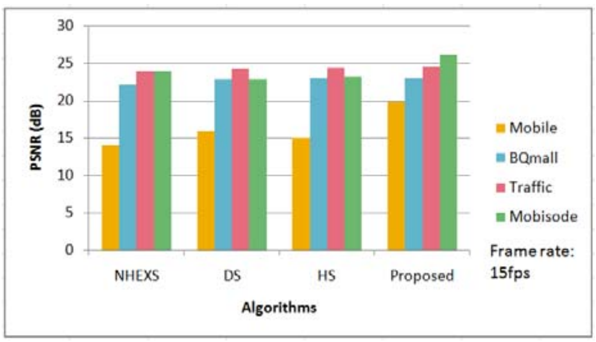


**Figure-7.** Comparison of different algorithms in terms of PSNR at a frame rate 15fps.

H.264 has a high complexity in computational time. So as to reduce the computational complexity, the number of computations should be reduced. This is achieved by limiting the search points of search patterns. The new pattern proposed in H.264 has minimum search points and so thus provide low computation than other search patterns. Plot of average computations and average PSNR for the existing and the proposed algorithm is shown in Figure-8 and Figure-9.

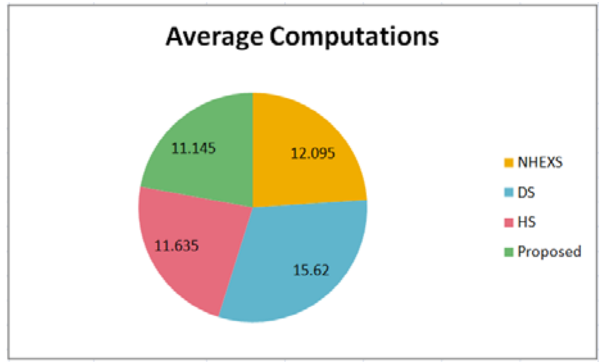


**Figure-8.** Average computations for the proposed and the existing search algorithms.

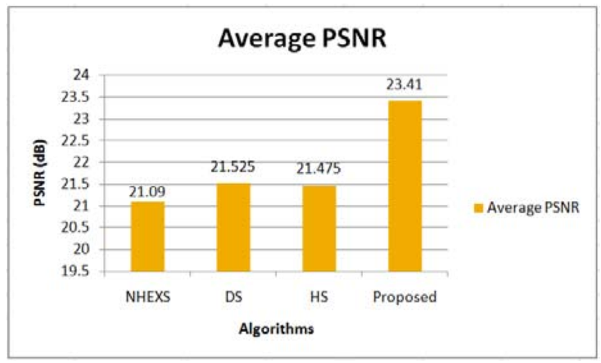


**Figure-9.** Average PSNR for the proposed and the existing search algorithms.

## 5. CONCLUSIONS

In this paper, we proposed a new fast improved search motion estimation algorithm for H.264/AVC video coding standard. The proposed algorithm reduces the number of checking points and thereby makes the motion estimation process faster. It outperforms other fast motion estimation algorithms like DS, HS, and NHEXS. Simulation results show that the proposed algorithm requires very few number of search points for finding the best matched block with almost negligible loss in video quality. Further, with less computation, there is a possibility for the architecture of the proposed algorithm to be computationally efficient. As compared to the existing ME algorithm, the simulated results of the proposed algorithm achieved an average of 11.355 search computations with less time in 10 fps and 11.145 search computations in 15 fps and an average PSNR of 23.41dB. The future work leads to the modification of the proposed algorithm for High Efficiency Video Coding standard and its hardware implementation.

## REFERENCES

[1] Karam Singh, Shaik Rafi Ahamed. 2013. A modified small cross-diamond search motion estimation algorithm for H.264/AVC. Annual IEEE India Conference. 1-5.

[2] C.H. Cheung, L.M. Po. 2002. A novel cross diamond search algorithm for fast block motion estimation. IEEE Trans. Circuits Syst. Video Technology. 12: 1168-1177.

[3] Y.W. Huang, B.Y. Hsieh, S.Y. Chien, S.Y. Ma, L.G. Chen. 2006. Analysis and complexity reduction of multiple reference frames motion estimation in H.264/AVC. IEEE Trans. Circuits Syst. Video Technol. 16: 507-522.

[4] Xing Wen, Lu Fang, Jiali Li. 2011. Novel RD-Optimized VBSME with Matching Highly Data Re-Usable Hardware Architecture. IEEE Transactions on Circuits and Systems for Video Technology. 21: 206-219.

[5] Huitao Gu, Shuming Chen, Shuwei Sun, Shenggang Chen, Xiaowen Chen. 2010. Multiple Search Centers based Fast Motion Estimation Algorithm for H.264/AVC. International Conference on Parallel and Distributed Computing, Applications and Technologies. pp. 346-350.

[6] Haithem Chaouch, Salah Dhahri, Abdelkrim Zitouni, Rached Tourki. 2010. Low Power Architecture of Motion Estimation and Efficient Intra Prediction based on Hardware Design for H.264. International conference on Design and Technology of Integrated systems in Nanoscale Era. pp. 1-6.

[7] C.W. Lam, L.M. Po, C. H. Cheung. 2003. A new cross diamond search algorithm for fast block matching motion estimation. Proc. Int. Conf. Neural Networks Signal Processing. 2: 1262-1265.

[8] Qiang Tang, Panos Nasiopoulos. 2010. Efficient Motion Re-Estimation with Rate-Distortion Optimization for MPEG-2 to H.264/AVC Transcoding. IEEE Trans. Circuits Syst. Video Technology. 20:262-274.

[9] O. Ndili, T. Ogunfunmi. 2011. Algorithm and architecture co-design of hardware oriented, modified diamond search for fast motion estimation in H.264/AVC. IEEE Trans. Circuits Syst. Video Technology. 21: 1214-1227.

[10] Kamel Belloulata Shiping Zhu, ZaikuoWang. 2011. A Fast Fractal Video Coding Algorithm Using Cross-Hexagon Search for Block Motion Estimation. International Scholarly Research Network Signal Processing. pp. 1-4.

[11] T. Wiegand G.J., Sullivan G. Bjøntegaard, A. Luthra. 2003. Overview of the H.264/AVC video coding standard. IEEE Transactions of Circuits and System for Video Technology. 13: 560-576.

[12] S. Zhu K. K. Ma. 2000. A new diamond search algorithm for fast block matching motion estimation. IEEE Transactions of Image Processing. 9: 287-290.

[13] Li Ren-xiang Zeng Bing, Liou M. L. 1994. A new three step search algorithm for block motion estimation. IEEE Transactions on Circuits and Systems for Video Technology. 4: 438-442.

[14] Puri X., Chen A. Luthra. 2004. Video coding using the H. 264/MPEG-4 AVC compression standard. Signal process. Image communication. 19: 793-849.