

**Military Technical College
Kobry El-Kobbah,
Cairo, Egypt**



**6th International Conference
on Electrical Engineering
ICEENG 2008**

An Efficient Block Matching Binary Motion Estimation Algorithm Using One Level Decomposition of Discrete Wavelet Transform

By

Dr. Atalla I.Hashad* Prof.Dr. Hazem H.Ali** Eng. Ajnadeen Khalil**

Abstract:

This thesis proposes a Discrete Wavelet Transform (DWT) based Block Matching Binary Motion Estimation (BME) algorithm. By using the DWT analysis, the size of data involved in the algorithm is enormously reduced. On the process of extracting the motion vector (MV), the block matching is performed only on one of the sub-frames of the DWT of the original frame. Motion vector is then found by means of minimum Sum of Absolute Differences (SAD). The proposed algorithm skips unnecessary steps of computations so it decreases the computational load with successive elimination of non-candidate blocks in the search window.

The proposed technique outperforms conventional full search block matching method, which is described in MPEG-4 verification mode, in terms of both peak signal to noise ratio and computational complexity. It is shown that the proposed algorithm is simple and efficient while keeping the same regularity and good performance.

Keywords:

Binary Motion Estimation (BME), Discrete Wavelet Transform (DWT), Motion Vector (MV).

* Computer Engineering Department, AAST, Cairo, Egypt

** Electronics & Communication Engineering Department, AAST, Cairo, Egypt

1. INTRODUCTION

A video sequence consists of a series of frames, however it has huge redundant information. To achieve compression, the temporal redundancy between adjacent frames can be exploited. The common technique to achieve this goal is known as motion estimation (ME).

Motion estimation has proved to be an efficient technique for exploiting the temporal redundancy in video sequences and is therefore an essential part of MPEG and H.263 compression standards[1, 2,3, 4].

In Full search (FS) or the Exhaustive Search method, the best predicted representative of the current block is searched by computing the matching criterion between the current block and all blocks in the search area.

The motion vector value corresponds to the candidate block, which yields the minimum Sum of Absolute Differences (SAD). The FS algorithm is a very simple algorithm which nevertheless produces good results. This type of BM can give least possible residue for video compression; however, the computation complexity is overwhelming and time intensive due to the large amount of candidates to evaluate.

To overcome this large number of computations , we propose a fast Binary Motion Estimation (BME) algorithm based on the analysis of Discrete Wavelet Transform (DWT). The DWT analyzes the signal at different frequency bands with different resolutions by decomposing the signal into a coarse approximation and detail information.

The DWT decomposition divides each frame of the video sequence of size $N \times M$ by a factor of 2 in each coordinate into a set of 4 ($N/2 \times M/2$) sub-frames that represent details at different scales. The motion vector calculation is only performed on the most efficient frame quarter, this minimizes the size of the data involved within the algorithm to $1/4$ compared to the Full Search.

Using Binary Motion Estimation (BME) within this algorithm efficiently skips the highly unlikely search positions [5,6,7,8]. Converting the sub-frame to a binary image, the algorithm performs the block matching only on the blocks having the same class. Then the motion vector is calculated with the measure of SAD.

Combining the advantages of the two techniques, DWT and BME, yields to a better estimation with a superior reduction of the data involved in the algorithm and with a high efficiency when compared to the normal full search.

In next section, an introduction to Discrete Wavelet transform is proposed. section 3, describes the proposed algorithm, while section 4, presents the simulation results of the proposed algorithm, and finally, In section 5, the conclusion is submitted.

2. 2-D DISCRETE WAVELET TRANSFORM

The DWT provides sufficient information both for analysis and synthesis for the original signal, with a significant reduction in the computation time [9,10,11].

The process starts with passing the signal $x[n]$ through a half band digital low pass filter with impulse response $h[n]$. [12]

Then subsampled by 2, this procedure can mathematically be expressed as

$$y[n] = \sum_{k=-\infty}^{\infty} h[k].x[2n - k] \quad (1)$$

Also, the original signal $x[n]$ is passed through a half band high pass filter $g[n]$ to generate the details coefficients, and a low pass filter $h[n]$ to generate approximation coefficients .

For many signals, the low frequency content (Approximation coefficients) is the most important and rich part of the signal. It has the most important information about the signal. While the high frequency content (Details coefficients) imparts flavor or nuance. After the filtering, half of the samples can be eliminated. This constitutes one level of decomposition and can mathematically be expressed as follows:

$$y_{high}[k] = \sum_n x[n].g[2k - n] \quad (2)$$

$$y_{low}[k] = \sum_n x[n].h[2k - n] \quad (3)$$

Where $y_{high}[k]$ and $y_{low}[k]$ are the outputs of the HPF and LPF, after subsampling by 2.

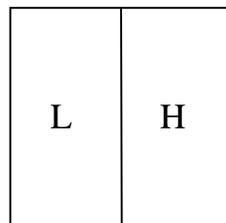
The filtering and subsampling will result half the time resolution and half the frequency band spanned (and hence doubles the frequency resolution).

The reconstruction of the original signal is very easy. The above procedure is followed in reverse order. The signals are upsampled by two, passed through the synthesis filters g' [n], and h' [n] (high pass and low pass, respectively), and then added. The interesting point here is that the analysis and synthesis filters are identical to each other, except for a time reversal. Therefore, the reconstruction formula becomes (for each layer) as follows

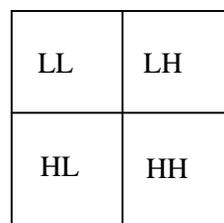
$$x[n] = \sum_{k=-\infty}^{\infty} (y_{high}[k].g[-n+2k] + (y_{low}[k]h[-n+2k])) \quad (4)$$

Under certain conditions it is possible to find filters that provide perfect reconstruction. The most famous ones and which we used in our research are Daubechies' wavelets.

In summary, DWT is performed first on the row direction and then on the column direction. When performing a two-level DWT to an image, the image is broken down into four sub-images; the approximation coefficients LL and details coefficients LH, HL and HH. With each having a size of 1/4 of the original image size. Fig. 1 shows the DWT four sub-images.



1st level in the row direction



2nd level in the column direction

Figure(1): DWT sub images for a two level decomposition.

3. PROPOSED ALGORITHM

a- Binary Motion Estimation

The FS algorithm simply applies the matching criterion to all possible motion vectors and finds the vector that gives the smallest error. The number of candidates to evaluate are $(2S_x+1)*(2S_y+1)$; where S_x & S_y are the search area in the horizontal and vertical aspects. This makes the number of computations is very enormous due to the great number of comparisons in the search area.

The BME reduces the number of candidate blocks to be compared with the current block. In the search area, the blocks are classified and those with the same class as the current block are considered as candidate blocks.

BME first begins with converting the frame into a binary image having a one bit binary value; either 0 or 1, then the classification process to each block is taken place depending on the number of ones it has. Table 1 shows the number of 1's in each class.

The matching process using BME is much easier and faster. As it deals only with 1 or 0, so the block matching is regarded as a comparison of the number of "1" contained in each reference block with that of the current block. In [5], the block matching process is performed between the blocks having the same class.

b- Motion Vector Extraction

The proposed algorithm starts with the DWT analysis, each frame in the input sequence is divided to four sub-frames each having a size of $\frac{1}{4}$ of the original frame size. Resulting into four sequences LL, LH, HL and HH, which represent the DWT coefficients of the frames at different scales.

In the process of extracting the MV, only the LL-frame, having the most important details about the original frame which yields to a good estimation, is used, that would reduce the time which was needed before in the FS to get MV.

First, we classify the current block (8×8) in the current LL-frame I_{LLC} to a class according to number of 1's it has. Then for each search position in the search window, we also classify the reference block in the reference frame I_{LLR} to a class. If a match is found, and by match we mean if the reference block have the same class or an adjacent class to the one of the current block, we start to calculate SAD by comparing the current block located at (x, y) to the reference block located at a displacement of $(\Delta x, \Delta y)$ as described in equation (5).

$$SAD(x,y) = \sum_{i=1}^8 \sum_{j=1}^8 |I_{LL}^c(x+i, y+j) - I_{LL}^r(x+\Delta x+i, y+\Delta y+j)| \quad (5)$$

Otherwise, we skip that position and start looking for the next search position. Hence, we can skip unneeded SAD computations between way different classes.

For each current block, the best match is chosen such that it has the minimum SAD. The motion vector MV represents the displacement of the current block from the reference block as described in equation (6).

$$MV(x, y) = (\Delta x, \Delta y) ; \{x \in R_x, y \in R_y\} \quad (6)$$

Where R_x is the search area in the x – direction and can have a value of $x - 8$: $x + 8$; and R_y is the search area in the y – direction and can have a value of $y - 8$: $y + 8$.

With the knowledge of the MV from LL-frame, each of the sub-frames LL, LH, HL, and HH, residuals are computed as

$$RES_{LL} = I_{LL}^c(y: y+7, x: x+7) - I_{LL}^r(y: y+\Delta y+7, x: x+\Delta x+7) \quad (7)$$

RES-LH, RES-HL, and RES_HH are calculated in the same way.

These four residuals together with the information of the MV are sent to the decoder to be used in the reconstruction process. The decoder gathers the four residuals with the reconstructed reference sub-frames I'_{LLr} , I'_{LHr} , I'_{HLr} , and I'_{HHr} respectively, resulting into four current sub-frames I'_{LLc} , I'_{LHc} , I'_{HLc} , and I'_{HHc} , which represent the DWT coefficients of the current frame.

An inverse DWT is performed so we can get the retrieved frame $I'c$. Also, the four reconstructed sub-frames are stored in a buffer to be used later as reference frames for the coming frames.

The reconstruction error is defined as the difference between the original current frame and the reconstructed current frame as

$$\begin{aligned} Rec. Error &= I_{LL}(n) - I'_{LL}(n) \\ &= (RES_{LL} + I_{LL}(n-d)) - (RES_{LL} + I'_{LL}(n-d)) \\ &= I_{LL}(n-d) - I'_{LL}(n-d) \end{aligned} \quad (8)$$

So, to minimize the reconstruction error, in the reconstruction process, we can replace the original reference frame $I_{LL}(n-d)$ with reconstructed reference frame $I'_{LL}(n-d)$, so that the error becomes zero.

4. SIMULATION RESULTS

Different QCIF and CIF test sequences have been used for experimental investigations. Twelve QCIF test sequences, each has 90 frames of six seconds duration, 15 frame/sec, data rate 1114 kbps and each frame of 144×176 pixels. And, eight CIF test sequences, each has 90 frames of six seconds duration, 15 frame/sec, and each frame of 288×352 pixels were used[13, 14].

Figure 2 shows an example of calculating the motion vector difference MVD for Foreman sequence between the motion vectors calculated using LL, LH, HL and HH, all compared with the FS. Figure (2.a) shows MVD on the x aspect (column), while figure (2.b) shows the MVD on the y aspect (row).

The horizontal axis presents the number of frames, while the vertical axis shows the average MVD all over the sequence and the values are per pixel. The graphs show that the nearest DWT sub-frame MV to the FS MV and with minimum error is the LL, and that is what made us choose extracting the MV from LL in the proposed algorithm.

The quality of the reconstructed frame is measured by the Peak Signal to Noise Ratio PSNR which is defined as:

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \text{ dB} \quad (9)$$

And

$$MSE = \frac{1}{NM} \sum_{i=1}^N \sum_{j=1}^M (I_c(i, j) - I'_c(i, j))^2 \quad (10)$$

Where N & M are the number of pixels in each of the row and column respectively.

The search ratio indicates the ratio of the number of SAD computations in the proposed algorithm to those in the case of the typical FS algorithm with the search area 17×17 candidate blocks in each comparison. Table 2 shows PSNR and search ratio of the retrieved test sequences using the proposed algorithm.

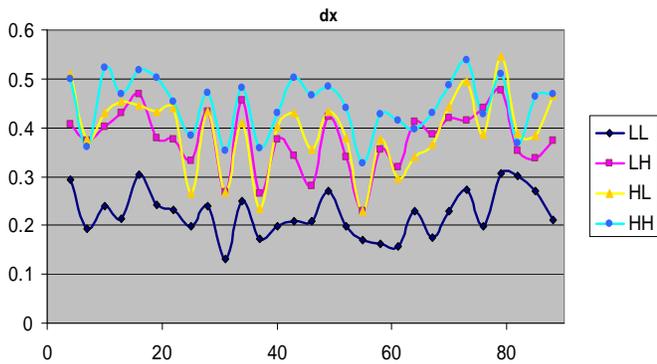


Fig 2.a

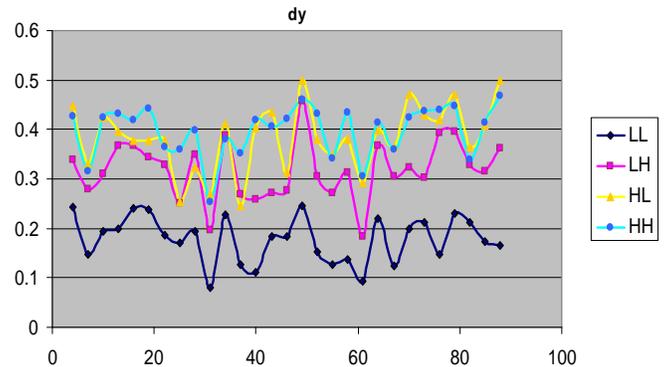


Fig 2.b

Fig 2. Average MVD of Foreman sequence on the x and y aspects.

5. CONCLUSION

The FS method is efficient but very computationally intensive. The proposed algorithm applies the search only on the LL sub-frame of the DWT of each sequence frame. Based on the results, we showed that using the LL coefficients yields the best efficiency to extract the motion vector and so reconstruct the original sequence with good efficiency when compared to the FS.

On the other side, the LH, HL and HH give also good results compared to the FS but not as high as the LL. The comparison between current and reference blocks is done after the BME technique to the blocks in the search area.

The results indicate that the quality of the reconstructed sequence is very high with PSNR > 60 dB in all the retrieved sequences. The computation process is significantly reduced and represents $\leq 2\%$ of the computation in the case of the FS algorithm.

TABLE (1):PSNR AND SEARCH RATIO OF THE PROPOSED ALGORITHM WITH STEP DIFFERENCE BETWEEN THE CURRENT AND REFERENCE FRAME 3 AND 5.

Sequences	Frame rate	no. of frames	PSNR (step diff. 3)	Saving search ratio %	PSNR (step diff. 5)	Saving search ratio %
<i>foreman_qcif</i>	15	90	67.4919	1.28	67.1863	1.244
<i>akiyo_qcif</i>	15	90	121.6145	1.48	121.665	1.446
<i>carphone_qcif</i>	15	90	71.4762	1.488	69.2167	1.4449
<i>grandma_qcif</i>	15	90	121.3814	1.366	121.432	1.333
<i>mobile_qcif</i>	15	90	68.8966	1.185	67.8798	1.1563
<i>mother and daughter_qcif</i>	15	90	71.2246	1.359	71.9827	1.327
<i>tempeste_qcif</i>	15	90	66.9495	1.22	67.0252	1.20
<i>table_qcif</i>	15	90	66.9495	1.518	64.6112	1.048
<i>container_qcif</i>	15	90	120.7548	2.201	120.8051	2.15
<i>news_qcif</i>	15	90	120.5736	1.5188	120.6244	1.48
<i>silent_qcif</i>	15	90	70.4	1.509	71.3169	1.46
<i>suzie_qcif</i>	15	90	67.4283	1.19	66.18	1.13
<i>foreman_cif</i>	15	90	68.3953	1.305	67.8388	1.27
<i>waterfall_cif</i>	15	90	74.0662	1.058	72.7908	1.044
<i>mobile_cif</i>	15	90	69.6584	1.08	67.0532	1.044
<i>mother and daughter_cif</i>	15	90	71.8222	1.32	70.7191	1.286
<i>tempeste_cif</i>	15	90	67.6143	1.234	67.5316	1.21
<i>container_cif</i>	15	90	122.381	1.58	122.4313	1.546
<i>news_cif</i>	15	90	74.8102	1.49	72.8219	1.45
<i>stefan_cif</i>	15	90	67.1791	1.18	66.0982	1.1124

6. REFERENCES

- [1] ISO/IEC CD 14496/2 (MPEG-4 Video), "Information technology---Coding of natural/visual objects, Part 2: Visual"1999.
- [2] Y. Noguchi, J. Furukawa, and H. Kiya, "A Fast Full Search Block Matching Algorithm for MPEG-4 Video," in Proc. IEEE Int. Conf. Image Processing, vol. 1, pp.61–65, 1999.
- [3] "Motion estimation tutorial," [http:// www. Cobweb .ecn.purdue.edu](http://www.Cobweb.ecn.purdue.edu).
- [4] T. Toivonen, Number Theoretic Transform -Based Block Motion Estimation, Master's Thesis, University of Oulu, Finland (2002)
- [5] E.Al-Qaralleh, and T.S.Chang, "An efficient binary motion estimation algorithm and its architecture for mpeg-4 shape encoding," IEEE Trans., VOL.16, NO. 7, JULY 2006.
- [6] Tsung-Han Tsai, Chia-Pin Chen, "A fast binary motion estimation algorithm for MPEG-4 shape coding." IEEE Trans, VOL. 14, Issue 6, June 2004.
- [7] Chung-Bin Wu, Chin-Yuan Yao, Bin-Da Liu, Jar-Ferr Yang, " DCT-based adaptive thresholding algorithm for binary motion estimation" IEEE Trans, VOL. 15, Issue 5, June 2005.
- [8] Seung-Hawn Kim, Yo-Sung Ho, "Fine granular scalable video coding using context-based binary arithmetic coding for bit-plane coding," IEEE Trans, VOL. 17, NO.10, October 2007.
- [9] Al-Bovik, "Handbook of Image & Video Processing," Academic Press, Austin-Texas, 2000.
- [10] Abhijeet Golwelkar, John W.Woods, "Motion-compensated temporal filtering and motion vector coding using biorthogonal filters," IEEE Trans, VOL. 17, NO.4, April 2007.
- [11] S.Mallat, "A Wavelet tour of signal processing", Academic Press, 1998.
- [12] <http://users.rowan.edu/~polikar/WAVELETS/WTtutorial.html>. The dwt wavelet tutorial.
- [13] <http://trace.eas.asu.edu/yuv/index.html>.
- [14] <http://eeweb.poly.edu/~yao/VideobookSampleData/doc/sample/H263.html>.