

## Adaptive inter frame compression using image segmented technique

Ebtesam Fadhel Kanger, Ban Abed –Al-Rizak, Ban Sabah Ismeel, Issraa Jimeel

Department of Astronomy and Space, College of Sciences, University of Baghdad

E-mail: ban-alsabah77@yahoo.com

### Abstract

The computer vision branch of the artificial intelligence field is concerned with developing algorithms for analyzing video image content. Extracting edge information, which is the essential process in most pictorial pattern recognition problems. A new method of edge detection technique has been introduced in this research, for detecting boundaries.

Selection of typical lossy techniques for encoding edge video images are also discussed in this research. The concentration is devoted to discuss the Block-Truncation coding technique and Discrete Cosine Transform (DCT) coding technique. In order to reduce the volume of pictorial data which one may need to store or transmit, the research modifies a method for video image data compression based on the two-component code; in this coding technique, the video image is partitioned into regions of slowly varying intensity. The contours separating the regions are coded by DCT, while the rest image regions are coded by Block-Truncation Coding. this hybrid coding technique called segmented image coding (SIC). Also in this paper A modify of the four step search for motion Estimation technique was produce. for searching scheme has been introduced which is contributed in decreasing the motion estimation searching time of the successive inter frames.

### Key words

Video compression,  
Image segmented,  
motion estimation

### Article info

Received: Oct. 2012

Accepted: Apr. 2013

Published: Sep. 2013

## تقنية مطورة لضغط الصور الفيديوية باستخدام تقنية التجزئة للصورة

ابتسام فاضل خنجر، بان عبد الرزاق، بان صباح اسماعيل، اسراء جميل

قسم الفلك والفضاء، كلية العلوم، جامعة بغداد

### الخلاصة

يعنى فرع المرئيات في الكمبيوتر بمجال الذكاء الصناعي الذي يهتم بتطوير وسائل لتحليل محتويات الصور الفيديوية. وبعد استخلاص معلومات الصورة الأجراء الأساسي في معظم مشاكل تعريف أشكال الصورة الفيديوية. في هذا البحث تم تبني تقنية محددة. تم التركيز على مناقشة تقنية ترميز تشذيب الزمر وتحويل الجيب تمام المنفصل (DCT). وفي النهاية وللحصول على أقل كمية من البيانات المرئية التي قد نحتاجها للخرن او النقل، تم في هذا البحث تطوير طريقة لضغط بيانات الصورة الفيديوية اعتمادا على ترميز مكونات الصورة. عن طريق تجزئة الصورة الى مناطق تتغير فيها الشدة ببطء. ويتم ترميز المحيط الذي يفصل هذه المناطق بواسطة الـ DCT، بينما يتم ترميز المناطق الباقية من الصورة بواسطة الـ B.T.C. إن طريقة الترميز الهجينة هذه تدعى ترميز قطع الصورة (SIC).

**Introduction**

The demand for communications with moving video picture is rapidly increasing. Video is required in many remote video conferencing systems, and it is expected that in near future cellular telephone systems will send and receive real-time video. A major problem in a video is the high requirement for bandwidth. A typical system needs to send dozens of individual frames per second to create an illusion of a moving picture. For this reason, several standards for compression of the video have been developed. Digital video coding has gradually increased in importance since the 90s when MPEG- 1 first emerged [1]. The degree of data reduction achieved by a compression process or algorithm is called compression ratio, given by[2]:

$$\text{compression ratio} = \frac{\text{Uncompressed File Size}}{\text{Compressed File Size}} \quad (1)$$

Therefore, measuring compression ratio as a function of the reproduced image quality define the efficiency characteristics of the adopted lossy compression technique, rather than a signal compression ratio in the decompressed pictures.

The equivalent definition to the mean square (SNR)<sub>ms</sub> is the Peak Signal-to-Noise Ratio (PSNR) defined as [2,3]

$$\text{PSNR} = \frac{[\text{Peak to Peak of } x_i]^2}{\text{MSE}} \quad (2)$$

The PSNR can be represented in decidable (dB) unit as:

$$\text{PSNR} = 10 \log_{10} \left[ \frac{\text{gray scale of image}}{\text{MSE}} \right]^2 \quad (3)$$

Interframe predictive coding is often used to eliminate the large amount of temporal and spatial redundancy that exists in video sequences. It is also helps in achieving compression purposes. In conventional predictive coding the difference between the current frame and the

predicted frame, which is based on the previous frame , is coded and transmitted. The better the prediction, the smaller the error and hence the lowest transmission bit rate. However, when there is motion in a sequence, then a pel on the same part of the moving object is a better prediction for the current pel. the use of the knowledge of the displacement of an object in successive frames is called Motion Compensation [4].This called Block Matching Algorithm, this algorithm estimates the amount of motion as blocks basis, i.e., for each block in the current frame, a block from the previous frame is found, that is said to match this block, based on a certain criterion .Two of them are:

The Mean Absolute Difference (MAD), defined as[1,5,6] :

$$\text{MAD}(dx, dy) = \frac{1}{mn} \sum_{i=-\frac{n}{2}}^{\frac{n}{2}} \sum_{j=-\frac{m}{2}}^{\frac{m}{2}} F(i, j) - G(i + dx, j + dy) \quad (4)$$

Where  $F(i,j)$  represents an  $(m \times n)$  macro block within the current frame,  $G(i,j)$  represent the corresponding macro block within reference frame (past or future),  $(dx,dy)$  a vector representing the search location .

The Mean-Squared Difference (MSD) cost function is defined as:

$$\text{MSD}(dx, dy) = \frac{1}{mn} \sum_{i=-\frac{n}{2}}^{\frac{n}{2}} \sum_{j=-\frac{m}{2}}^{\frac{m}{2}} [F(i, j) - G(i + dx, j + dy)]^2 \quad (5)$$

In this research, modifies a method for video image data compression based on the two-component code; which is called Segmented Image Coding (SIC) is relatively an image compression technique. In SIC, the image is partitioned into regions of slowly varying intensity, and the edge (contours) separating the regions. The actual performance of SIC depends highly on the edge detection algorithm used. For coding purposes, the edge detection algorithm should have the

property that the contour (or edge) should be smooth so that they can be coded efficiently. At high compression ratios, SIC yields a better subjective image quality than block transform coding (BTC) methods such as discrete cosine transform coding, or block truncation coding method, because the objectionable block distortion is avoided. For this reason, SIC is ideal for every low bit rate coding and progressive transmission of images and is considered a possible candidate for inclusion in the pcoming MPEG – 4 standard.

In what follows we shall explain the procedures involve in our segmented image coding *SIC*. The adaptation imposed, in our present work, on those likely based published techniques will be demonstrated by the following [6,7,11].

### Edge Detection

The first step in most video image processing application is to detect sharp transitions called edges, and then connect these edges to outline the desired boundaries. In short, extracting edge information and constructing boundaries by connecting edges and line segments can be considered as the essential process in most pictorial analysis and pattern recognition problems, since subsequent measurements of shape size (area or land perimeters) and texture can then be taken. Moreover, image data compression by separating the image into two parts “i.e. edges and all the rest”, can also be achieved as a sub – sequent process to the edge detection. A Maximum of Neighboring Difference Technique [11]. The maximum of neighboring difference method (MND) is an algorithm for edge detection, which can be used for most applications. The MND algorithm is performed by computing the maximum difference between the  $3 \times 3$  mask central points with their 8 – neighbors. Edge image, then, defined by assigning edge point to all

image pixels having difference values more than or equal to a predefined threshold value. In our present study, the threshold adopted was the image differences mean value [9-11].

### Image Block Subdividing Step

The next step in our coding algorithm is to group the binary edge image elements, which obtained from the first step into similar size blocks, each is considered as to be an individual image region. In our present work, different size ( $N \times N$ ) block are used; i.e.  $4 \times 4$ ,  $8 \times 8$ ,  $16 \times 16$ . Then we used a suitable threshold value to compare with the number of edge point in each block to decided if this block is an edge region or not and as follows; <sup>[11]</sup> :-

1. No. of edge. Point  $\geq$  Th, then the block is edge region.
2. No. of edge. Point  $\leq$  Th, then the block is not edge.

Accordingly, experiments showed that larger threshold value results in an important loss of edge information, which must be avoided.

### Encoding Process

Up to this point, the segmented image regions are classified into two types, these are; edge region or rest region. Our major aim was directed toward the development of a coding scheme based on segmenting the digital image. In our present work, the encoding process is adopted by used two compression image data techniques to encoded the same digital image, so we adopt the discrete cosine transform coding on edge region, while the block truncation method is used to encode the rest region. The two compression techniques mentioned above are implemented on the original image, so the binar (edge detected) image is used to decided if this region is edge region or rest region only. However, certain flags are used to distinguish it the block was

coded by the block traction or by the DCT method[10] .

**Data image compression using Block Truncation method**

This coding technique is designed to preserve image details, using the two image statistical parameters; i.e. mean and variance. The method significantly reduces the amount of computation required by most traditional coding methods such as transform coding, yet it still produces an acceptable quality of reproduction. However, the block truncation coding (BTC) technique retains the sharp edges, which may appear ragged in the reconstruction, the BTC This technique can be summarized by the following[1,12] :-

- Subdividing an input image into uniform blocks, (such as, 4×4 pixels in size).
- For each block, the mean ( $\mu$ ) and the standard deviation ( $\sigma$ ) are computed.

All pixels with values greater than or equal to block’s mean are assigned a value "1" while pixels with values less than the block’s mean assigned a value "0", then the bit plane contains two quantified levels i.e. "a" and "b" are

$$a = \mu - \sigma \sqrt{\frac{q}{(n-q)}} \tag{6}$$

$$b = \mu + \sigma \sqrt{\frac{(n-q)}{q}} \tag{7}$$

where n is the number of pixels in the block, q is the number of pixels greater than or equal to the block mean.

- The image can, then reconstructed by replacing all "0" values by "a" and those of "1" values with "b".

**Data image compression using DCT Coding**

The following five main steps can describe the lossy coding which is adopted in our work.

Step 1: Transformation process: by which the discrete cosine transformation is Applied to each of the"8 x 8" blocks separately. The output of each DCT is an 8 × 8 matrix of DCT coefficients

Any gray-scale 8 × 8-pixel block can be fully represented by a weighted sum of these 64 basis functions. The DCT operation therefore transforms a block of pixels into the set of DCT coefficients that represent the block in the spatial frequency domain. This is itself does not give compression since the information is merely being represented in different form [1,2].

The two-dimensional DCT of an 8 × 8 block of pixel values is described by the following formula, where the pixel value are represented as f (x, y), and the transform coefficients as F (u, v) [9]

$$F(u, v) = \frac{C(u)}{2} \cdot \frac{C(v)}{2} \sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16}, \tag{8}$$

where

$$C(u), C(v) = \left\{ \begin{array}{ll} \frac{1}{\sqrt{2}} & \text{for } u \text{ or } v = 0 \\ 1 & \text{for } u \text{ or } v > 0 \end{array} \right\} \tag{9}$$

The  $F (0,0)$  coefficient, is called the " DC coefficient ", and the remaining 63 coefficients are called "AC coefficients".

Step 2: Quantization process: by which the less significant DCT coefficients are wiped out. The result of this lossy transformation is an integer produced by dividing each of the coefficients in the 8 x 8 DCT block by a weighted value taken from a specified table. If all the weights are “1”, means the transformation does nothing. However, if the weights increase sharply from the origin, higher spatial frequencies are dropped quickly. the uniform quantization is performed according to the following equation[1]:

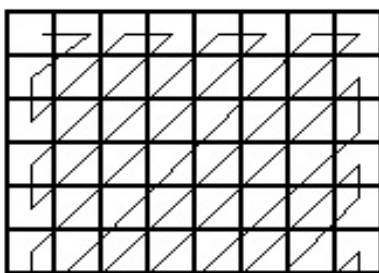
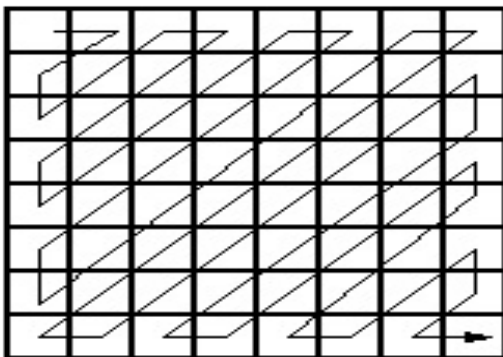
$$F_q(u, v) = \text{Round} \left[ \frac{F(u, v)}{Q(u, v)} \right] \tag{10}$$

where  $Q(u, v)$  are the quantization coefficients specified in the quantization table. Each element  $Q(u, v)$  is an integer from 1 to 255, which specifies the step size of the quantizer for its corresponding DCT coefficient [10]. After this step we will presented a new two techniques for quantization of coefficients.

i) **Retaining Transformed Quantized Coefficients:** In our present technique, a number of significant transformed coefficients are selected, while non-significant coefficients are set to zero. The retained coefficients, within each transformed block, have been selected according to their distance from the DC-coefficient (using Zig-Zag scan follower).

ii) **Energy Coefficients Retaining Method:** The selected number of coefficients in this method followed the percentage amount of energy preserved in these transformed coefficients. This has been done by counting (percentile), the retained coefficient values with respect to the total values within each block.

Step 3: **Zig-Zag-Scan process:** except for the DC coefficient, which is treated separately, the 63 AC coefficients are read using a zigzag scan as shown in Fig.1 in order to transform matrix into a flow of data best suited for the next coding steps [1,2,9].



Step 4: **Entropy Encoding:** in this step the selected quantized coefficients are transformed in binary codes, suited for storage or transmitting through certain transmission channel. Huffman encoder is one of the most common encoding methods used to perform this encoding process

In summary, the Huffman coding algorithm consists of the following steps.

- Arrange all source symbols in such away in which their occurrence probabilities are in a non-increasing order.
- Combine the two least probable source symbols:
  1. Form a new source symbol with a probability equal to the sum of the probabilities of the two least probable symbols.
  2. Assign a binary “0” and a binary “1” to the two least probable symbols.
- Repeat until the newly created auxiliary source alphabet contains only one source symbol.
- Start from the source symbol in the last auxiliary source alphabet and trace back to each source symbol in the original source alphabet to find the corresponding code words[1,2,9]

**DCT video data image Decompression Processes**

In the DCT sequential decoding process, all the steps of the encoding process are inverted (i.e. implemented in reverse order), first, an entropy decoding process (i.e. Huffman Inverse Process) is implemented on the received compressed image data. The binary sequence is converted to a symbol sequence using Huffman Table.

Dequantization process, then, is implemented using the following

function:

$$F(u, v) = F_q(u, v) \times Q(u, v) \quad (11)$$

Where  $Q(u, v)$  is the quantization coefficient obtained from the quantization table.

Then, the Inverse Discrete Cosine Transform ( IDCT ) is implemented on the dequantized coefficients to convert the coefficients from frequency domain into the image spatial domain. The IDCT equation is defined as [1,2] :

$$f(x, y) = \frac{1}{4} \left[ \sum_{u=0}^7 \sum_{v=0}^7 C(u)C(v)F(u, v) \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16} \right] \quad (12)$$

where:

$C(U)$  ,  $C(V)$  as defined in equation (9).

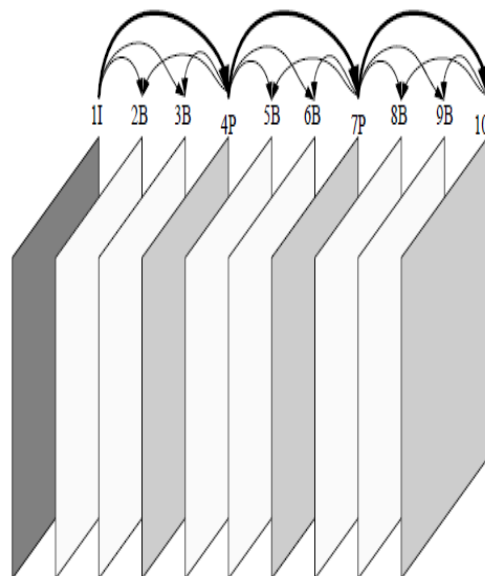
### Motion Estimation

The goal of video compression is to remove the redundancy in the video signal. In still image coding, the main task is to improve the transform and entropy coding efficiency. However, the optimal performance of an individual video frame in video coding is upper bounded by the entropy of its residual frame. Motion estimation plays an important role in video coding by taking advantage of the temporal Redundancy across the sequence. The general goals of motion estimation methods are to improve the prediction accuracy, or to reduce the implementation complexity [4].

### Interpolation (IBBP)

Interpolation is a simple yet efficient and important method in image and video compression. In image compression The IBBP frame model is anew feature of video coding which consists of three pictures (one p-picture and two BB-pictures). The P-pictures are predicted from previous decoded (I). The BB-pictures are bidirectional predicted both from the previous decoded (I) and P-picture. This model has been used in this work [9,12].

$$f(x, y, t) = \frac{I_2}{I_1 + I_2} f(x, y, t_1) + \frac{I_1}{I_1 + I_2} f(x, y, t_2) \quad (13)$$



**Fig.2:** The BB-pictures are bidirectional predicted both from the previous decoded ( I ) and P-picture[3,13]

### Adaptive Four Step Search for motion estimation technique (AFSS)

An image, by this method, is partitioned into a set of non –overlapped, equally spaced, fixed size, small rectangular blocks, and the translation motion within each block is assumed to be uniform. Although this simple model consider translation motion. this search, anew mechanism is introduced, it is based on dividing the image into 8x8 blocks, implementing the search for each block separately, such that each search will give minimum values matching criterion; i.e (MAD and MSD) with different threshold (4,8 and12):

Step 1: Nine values for the cost function calculated.

Step 2: Six new cost function are calculated are surrounding for the minimum cost at location in previous step.

Step 3: The location with the smallest cost function is selected as a new (center location) and three cost function are calculated surrounding this location.

Step 4: The smallest value is the (center location) and a new nine cost function are calculated surrounding this location [7,8,12,13]. The smallest value is the final estimate of the motion vector.

The procedure of (AFSS) as shown in Fig.3.

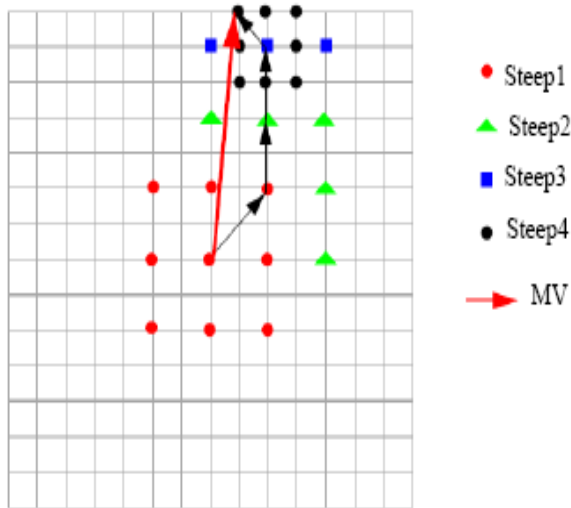


Fig.3: Show the adaptive four step search for motion Estimation technique.

**Experimental results of the mentioned coding**

In this section, we shall demonstrate the obtained results by the mentioned encoding algorithm. Samples of two images have been chosen to demonstrate the coding results, with different block size. The new efficient video image compression using block truncation with different block size is applied to Man the image size (128x128) and reference block size is (4x4),( 8x8) and (16x16) pixel on smooth area and DCT Technique on edge image. Also was applied

adaptive search for motion Estimation technique and reference block size is( 8x8) pixel. We will show the usefulness of the motion estimation in interframe compression., these will be illustrated in the following figures.Fig.4 shows the original frames (I),(P) and there when when we apply edge detection technique by(MND). Fig.5 shows the reconstruction images (I),(P) when we applied Block Truncation technique, block size (4x4) on the smooth area and applied DCT technique for edge image. Fig.6 show the block different for reconstruction frames (I), (P) when we apply (AFSS) used (MAD) criteria with thr.4. Fig.7 show the block different between the reconstruction frames (I), (P) when we apply (AFSS) used (MAD) criteria with thr.12. Fig.8 represent the interpolation frames (B1), (B2).Fig.9 shows the (I), (B1), (B2), (P) respectively.

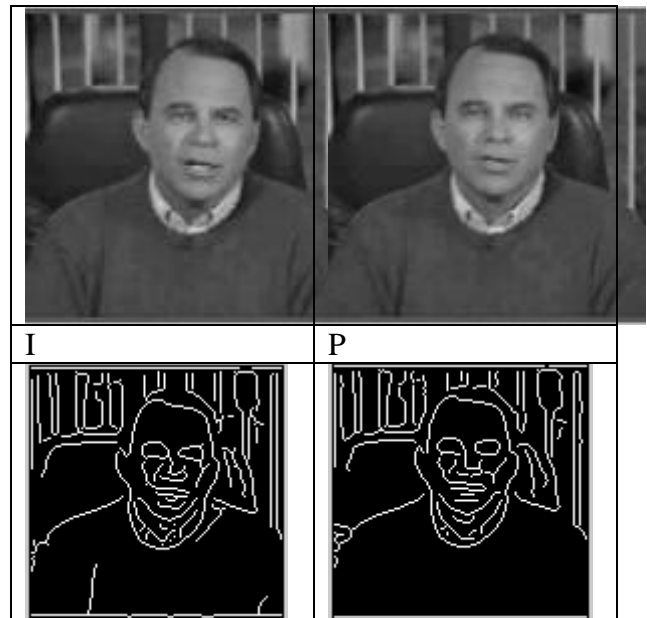


Fig.4: Shows the original frames (I),(P) and there when we apply edge detection technique by(MND)



**Fig.5:** Shows the reconstruction images (I),(P) when we applied Block Truncation technique, block size (4x4) on the smooth area and applied DCT technique for edge image.



**Fig.6:** Show the block different for reconstruction frames (I), (P) when we apply (AFSS) used (MAD) criteria with thr.4.



**Fig.7:** Show the block different between the reconstruction frames(I),(P) when we apply (AFSS) used (MAD) criteria with thr.12.





Fig.8: Represent the interpolation frames (B1),(B2).



Fig.9: Represented the frames (I), (B1),(B2) and (P) respectively Where BB-pictures are bidirectional predicted both from the previous decoded (I) and P-picture

Table1: Numerical result when we apply motion estimation (AFSS) with different criteria (MAD) and (MSD) with block truncation technique block size (4x4)

Comp. ratio (I) = 5.6 PSNR=27.6

Threshold	Tim of search(sec)	No. of block change	Over all comp. ratio	PSNR (B1)	PSNR (B2)	PSNR (P)
4 (MAD)	0.395	89	19.5	24.8	24.5	26
8 (MAD)	0.211	83	21.8	24.8	24.5	26
12 (MAD)	0.159	51	21.5	24.8	24.5	26
4 (MSD)	0.395	76	22.3	24.3	23.7	24.9
8 (MSD)	0.211	57	35.2	24.3	23.7	24.9
12 (MSD)	0.161	39	39.3	24.3	23.7	24.9

Table 5: Numerical result when we apply motion estimation (AFSS) with different criteria (MAD) and (MSD) with technique block size (8x8). Comp. ratio (I)=6.9 PSNR=25.7

Threshold	Tim of search(sec)	No. of block change	Over all comp. ratio	PSNR (B1)	PSNR (B2)	PSNR (P)
4 (MAD)	0.395	89	22.7	23.2	22.7	24.3
8 (MAD)	0.211	83	25.6	23.2	22.7	24.3
12 (MAD)	0.161	51	31.2	23.2	22.7	24.3
4 (MSD)	0.395	76	27.1	22.3	22.1	23.2
8 (MSD)	0.211	57	37.5	22.3	22.1	23.2
12 (MSD)	0.161	39	40,8	22.3	22.1	23.2

**Table3: Numerical result when we apply motion estimation (AFSS) with different criteria (MAD) and (MSD) with technique block size (16x16).  
Comp. ratio (I) =9.2 PSNR=23.2**

Threshold	Tim of search(sec)	No. of block change	Over all comp. ratio	PSNR (B1)	PSNR (B2)	PSNR (P)
4 (MAD)	0.395	89	25.7	21.8	21.1	22.2
8 (MAD)	0.211	83	28.9	21.8	21.14	21.2
12 (MAD)	0.161	51	33.8	21.8	21.14	21.2
4 (MSD)	0.395	76	36.9	21.8	20.1	20.9
8 (MSD)	0.211	57	40.2	21.3	20.1	20.9
12 (MSD)	0.161	59	44.7	21.3	20.1	20.9

### Conclusions

From the result we were found the best image reconstruction(PSNR) when we used block truncation at block size (4x4) with good value of compression ratio but lowest C.R than block size (8x8)and (16x16). The PSNR lowest at image reconstruction at block size (16x16). We have presented motion estimation technique, Adaptive Four Step Search for motion estimation technique (AFSS), The experimental results show that this new method can significantly reduce the computational complexity and low search time while achieving comparable performance for many video sequences. Also we have found the criteria (MAD) was gave higher number of block different than criteria(MSD) then low C.R when we used(MAD) than (MSD).Threshold (4) was gave highest number of block different than other thresholds but lowest C.R.

### References

[1] R.C. Gonzalez and R.E. Woods, "Digital Image Processing", Prentice Hall, 2<sup>3d</sup> edition, 2008.  
[2] Zhen Li "New methods for motion estimation with applications to low complexity video compression" Ph.D Thesis, Faculty of Purdue University, December 2005.

[3] J.Eric Balster "Video compression and Rate control methods Based on the wavelet Transform" Dissertation Degree of Doctor the Ohio State University, 2004.

[4] N. Kanvel, and Elwin Chandra Monie "Adaptive lifting based image compression scheme for narrow band transmission system" International Journal of Physical Sciences, 4, 4(April,2009) 194-164

[5] A. Barjatya, Student Member, IEEE" Block Matching Algorithms for Motion Estimation" IEEE Trans. on Circuits and Systems for Video Technology, 2004.

[6] T. Kondo, Waree Kongprawechnon "Robust motion estimation methods using gradient orientation information" doi: 10.2306/scienceasia1513-1874.2009.35.196 ScienceAsia, 35 (2009).

[7] V. V. Gohokar ,V. N. Gohokar "Selection of Motion Estimation Block Size for Rate -Distortion Optimization" International Journal of Computer Applications (0975 – 8887) 17,4 (March 2011)

[8] S. Momcilovic, Nuno Roma and Leonel Sousa "An ASIP Approach for Adaptive AVC Motion Estimation" International Journal of Computer Science, issues2007.

[9] Yun, Q "Image and Video Compression for multimedia engineering", New Jersey Institute of Technology Newark, NJ,2000.

[10] D. L. Gall, "MPEG: A Video compression standard for multimedia application", Commune ACM, Vol. 34, 1991.

[11] N. Ethan, Sh. Sara, H. Hamid, "Edge Detection Techniques Evaluations and Comparison" *Applied Mathematical Sciences*, 2, 31(2008) 1507-1520.

[12] D. Mohammed, and Fatma Abou-Chadi (senior member, IEEE.) "Image Compression Using Block Truncation Coding", *Cyber Journals: Multidisciplinary Journals in Science and Technology* ,

*Journals of Selected Areas in Telecommunications (JSAT)*, February Edition, 2011.

[13] W.Ahmad Zheng, J. Luo, and M, Liou "A Fast Adaptive Motion Estimation Algorithm" *IEEE Transactions on Circuits and Systems for Video Technology*, 16, 3, (March 2006).