Video Clip Compression Using DCT Technique

A Thesis
Submitted To the Council of College of Science
University of Babylon
In Partial Fulfillment of the Requirement for the degree of Master in Computer Science

BY

Wafaa Hassan AL-Marsomi

٢٠٠٦ - May ١٤٢٧-Jamad Al-awal

بسم الله الرحمن الرحيم

الله نور السموات والأرض مثل نوره كمشكاة فيها
مصباح المصباح في زجاجة الزجاجة كأنها كوكب دري
يوقد من شجرة مباركة زيتونة لا شرقية ولا غربية يكاد زيتها
يضيء ولو لم تمسسه نار نور على نور يهدي الله لنوره من
يشاء ويضرب الله الأمثال للناس والله بكل شيء عليم

صدق الله العلي العظيم

سورة النور (٣٥)

Supervisor Certification

We certify that this thesis was prepared under our supervision at the Department of Computer Science/College of Science/Babylon University , by **Wafaa hassan Alwan** as partial fulfillment of the requirements for the degree of Master of Science in Computer Science.

Signature:	Signature:
Name: **Dr. Nabeel Hashem Kaghad**	Name**: Dr. Tawfiq A. Abbas**
Title: **Professor**	Title: **Assistant Professor**
Date: / /٢٠٠٦	Date: / /٢٠٠٦

In view of the available recommendations, I forward this thesis for debate by the examination committee.

Signature:

Name: **Dr. Abbas Mohson Al-Bakry**

Title: **Head of the department of Computer Science, Babylon University**

Date**:** / /٢٠٠٦

Certification of the Examination Committee

We chairman and members of the examination committee, certify that we have studied the thesis entitled **(Video Clip Compression Using DCT Technique)** presented by the student **Wafaa Hassan Alwan** and examined her in its contents and in what is related to it, and we have found it worthy to be accepted for the degree of **Master of Science in Computer Science** with **Degree.**

Signature:	Signature:
Name: **Sattar B. Sadkhan**	Name: **Jassim Sarsouh**
Title: **Professor Assistant**	Title: **Professor Assistant**
Date**:** / /٢٠٠٦	Date: / /٢٠٠٦
(Chairman)	**(Member)**

Signature:

Name: **Baheja K.Shukur**
Title: **Professor Assistant**
Date: / /٢٠٠٦
(Member)

Signature:	Signature:
Name:**Dr. Nabeel Hashem Kaghad**	Name**:Dr. Tawfiq A. Abbas**
Title: **Professor**	Title: **Assistant Professor**
Date: / /٢٠٠٦	Date: / /٢٠٠٦
(Supervisor)	**(Supervisor)**

Signature:

Name:**Dr. Oda M.Y. Al-Zamely**
Title: **Dean of College of Science in Babylon University**
Date: / /٢٠٠٦

الإهـــداء

إلى من علم نفسي الصفـــاء

ومن منحني بمـرونـه البهـــاء

ورفع رأسي بين الناس بكبريـــاء

إلى من أسمـــاني وفـــاء

والـــدي الحبيب حسن المرسومي

إلى نبض قلبي ونور عيني الحبيبة أمي

إلى رمز الأمتنان زوجي

إلى ينبوع الحنان عائلتي

إلى زهرة الأقحوان ساهر

إلى بناة مستقبل العراق (أطفال العراق)

وفـــاء

Acknowledgment

I would like to express my deep appreciation to my supervisors Prof. Dr. Nabeel Hashem and Dr. Tawfiq Al-Asadi For their invaluable guidance , supervision and untiring efforts during course of this work.

Sincere appreciation is due to my family especially my Dad Mr. Hameed AL-Ahraji and my lover mother for their patience, encouragement and help during the work.

I would like to declare my deep thanks and appreciation to my husband Hydar for all his support and encouragement which I have felt through the preparation of this work. Great thanks also go to my lovely brothers & sisters.

Special thanks to my close friends Samaher, Aminaa, Hedhab, Balasim & Firas for every thing they presented for me.

I would like to thank all my dear colleagues in the Msc.Course and to whole staff of the Computer Science Department in Babylon university and spatial thanks go to my university kerbala .

Finally I would like to thank all the kind ,helpful and lovely people who helped me directly or indirectly to complete this work and apologize to them for not being able to mention them by name here but they are in my heart.

Abstract

A video Compression technique plays a vital role in storage & transmission through limited bandwidth. Because of video files contain repeated sequence of still image as a result each second represented by٢٤ image or more which is represent the motion in video file therefore that repeated sequence provide us large amount from temporal redundancy through similarities of sequence image with few difference between them. In addition to spatial redundancy contained in image data itself . Our proposed system achieves compression ratio close to the best known video compression techniques MPEG between (٢٠- ٤٠).

The proposed system can be summarized into four stages the first stage is opening Audio/Video Interleaved file (AVI) Video file which is called RIFF. AVI, second stage is compress sequence of video by using techniques presented in lossy compression in video Codec through used of transform technique which is presented by discrete cosine transform to reduce spatial redundancy in image data itself and to reduce temporal redundancy by using motion compensation technique which is employed associated with discrete cosine transform to error image produced by motion technique based on skew measurement and at last of this stage we used efficient method to code them, the third stage we inverse all hybrid compression techniques to obtain the decompressed sequence image of video file to use. At the fourth stage which is used to reconstruct video file of RIFF .AVI type. Results are generated for compression ratio with PSNR for each image component in video with average PSNR for Video file to compare between quality of decompressed video file and original video. All these file tested on ٢١٠٠ GHz Pentium ٤ with ٢٥٦ MB of main memory. The programming language was VB ver.٦. The proposed system was achieved high results for compression ratio with lost of the data which was clearly from PSNR values for these movie file.

List Of Contents

List Of Abbreviations

Symol	**Meaning**
B-Frame	Bidirectional-Frame
BDA	Block Distortion Algorithm
BMC	Block Motion Compensation
BPP	Bit Per Pixel
CATV	Cable Television
CF	Compression Factor
CIF	Common Intermediate Format
CP	Compression Performance
CR	Compression Ratio
CS	Color Space
CBR-VBR	Constant Bit- Rate , Variable Bit- Rate
Codec	Coder/Decoder
DCT	Discrete Cosine Transform
DIP	Digital Image Processing
DFT	Discrete Fourier Transform
DWT	Discrete Wavelet Transform
DVD	Digital Video Data
eRMS	Root-Mean- Square Error
FC	Fidelity Criteria
FPS	Frame Per Second
FSBM	Full Search Block Matching
HDTV	High Definition Television
I-P-	Intra-Inter -Frame
ISDN	Integrated Services Digital Network
ISM	Interchange Storage Media
ISO	International Standard Organization
ITU	International Telecommunication Union
JPEG	Joint Picture Experts Group
JPEG٢٠٠٠	Joint Picture Experts Group ٢٠٠٠
MAD	Mean Absolute Distortion
MC	Motion Compensation
ME	Motion Estimation
MP	Matching Pursuit
MSD	Mean Square difference Distortion

MPEG	Moving Picture Expert Group
MPEG٤AVC	MPEG٤ Advanced Video Control
OBMC	Overlapped Block Motion Compensation
PCD	Pel difference Classification Distortion
PSNR	Peak Signal –to- Noise Ratio
QCIF	Quarter Common Intermediate Format
RIFF	Resource Interchange File Format
RGB	Red , Green , Blue
RLE	Run Length Encoding
SIF	Standard Intermediate Format
SQ	Scalar Quantization
SNR *rms*	Signal –to- Noise Ratio
Sub-QCIF	Sub- Quarter Common Intermediate Format
VQ	Vector Quantization
YCC	Luminance/Chrominance components

List Of Figures

List Of Tables

Chapter One

Introduction

١.١ Introduction

Digital video has become very important form of information technology and is now used in many different areas, such as board casting, teleconferencing, mobile telephone, surveillance, and entertainment. People now expect to be able to access video through a wide range of different devices and over various networks. To provide these kinds of services we must know what is video compression for storage and transmission [١].
Compression of video imagery has become a necessity and very important because of transmission and storage of uncompressed video would be extremely costly and impractical. For instance, a video sequence running for (٩٠) minutes, at (٢٥) frames per second, at standard resolution of (٧٢٠ *٥٧٦), and with (٢٤) bit per pixel would require (١٦٧٩٦١٦٠٠٠٠٠) bit or approximately ١٥٦.٤٣ gigabytes". This is not a problem if we only wish to access and deal with video through high - end systems with a lot of storage and network band width. However, since it is desired that video will be accessible from a wide range of devices having different capabilities and connected to different networks, therefore some form compression for digital video is required [٢]. Compression refers to the process of reducing the number of bits required to represent the image and video comes in two forms lossless and lossy[٣]. In lossless compression ,is a process to reduce image or video data for storage and transmission while retaining the quality of original image (i.e. the decoded image quality is required to be identical to image quality prior encoding). In lossy compression, on the other hand, some information present in the original image or video is discarded so that the original raw representation of image or video can only be approximately reconstructed from the compressed representation with high compression

ratio. For more compression can be achieved with approximate quality to source image lossy compression is almost usually used than lossless compression to compress digital video [٤].

Generally speaking, in video compression, the video sequences contain significant amount of statistical and subjective redundancy within and between frames. The ultimate goal of video source coding is the bit-rate reduction for storage and transmission by exploring both spatial and temporal redundancy and to encode “minimum set” of information using entropy coding techniques .This is usually results in compression of coded video data compared to original source data [٥].

In natural images or in frames of natural video, pixels that are close to one another tend to have similar values and tend to be very correlated – if we randomly take from natural images samples consisting of pixels spatially close to one another and then plot of those samples in a multidimensional space each axis of which gives the value of one pixel in a sample, the distribution of points produced would fit well to a line .The reason is that most of the time, samples would be taken of pixels lying within the boundary of some object or region of semantic significance ,and pixels within these boundaries would very likely be similar and correlated simply because it is very property that allow us as human beings to identify visually such objects in our surroundings. This correlation between pixels implies the presence of redundancy called spatial redundancy, in the uncompressed representation of an image because the tendency of images to have correlated pixels means in the set of all possible images, some images are far more likely to occur than others, which lowers the entropy of the probability distribution of images such that if all images are represented in the same number of bits per pixel, as in uncompressed images then on average, significantly more bits than this entropy are used –thus the redundancy in the uncompressed representation . Further more, for lossy compression ,in which a form of quantization is used to discard some detail in the representation of image samples, the correlation means that less quantization cells are required to partition the sample space for given distortion than would be required if the images were purely, and less cells require less bits to encode, improving compression efficiency further. In addition the redundancy makes techniques such as transform coding useful – without the redundancy, transform coding would have no benefit – after which other clever methods can be used to code the resulting transform coefficients efficiently [٦]. A video sequence is essentially an ordered set of frames are displayed in quick succession to a human viewer, they give the visual illusion of motion .This illusion is achieved by keeping the semantic content mostly the same in each frame of group of successive frames - so that the viewer is able to identify the same critical objects or critical elements of the video in each frame –and by introducing only small changes to this content between frames – by moving objects relative to one another or by panning, or zooming or moving the camera, etc . Therefore the fact that the video is intended for human viewer is important because it allow us to assume things about the nature of the video that greatly aid in compression. To give this illusion of motion, successive frames in the video must have very similar semantic content, and this effectively means that successive frames often have very similar pixel have moved slightly between successive frames. As a result there is a lot of information that is shared between frames in sequence, and this information, ideally, should only be encoded once in uncompressed representation of the video sequence, this creates a redundancy called temporal redundancy [٦].

The performance of the video compression techniques depends on the amount of spatial and temporal redundancy contained in video data as well as on actual compression techniques used for encoding – with practical code schemes a trade – off between coding performance (high compression with sufficient quality) and implementation complexity is targeted – thus one of the most method which is used for exploiting spatial redundancy is the transform coding and for exploiting temporal redundancy used motion estimation methods [٥], all of these method used in video codec which are explained in figure (١-١)[٧].

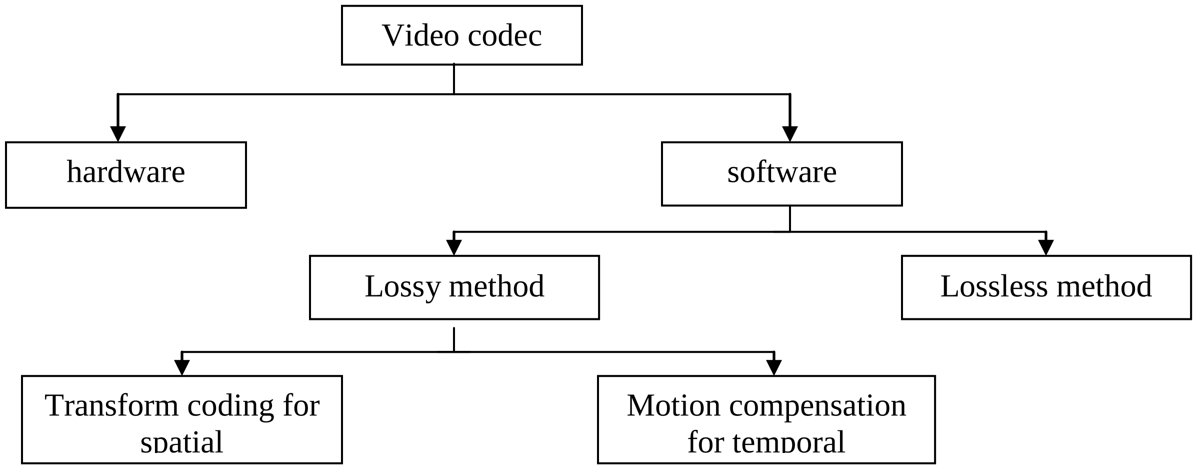

Video codec
hardware
software
Lossy method
Lossless method
Transform coding for spatial
Motion compensation for temporal

Figure (١-١): Types of video codec

Video Codec is a codec i.e. computer program that compress/ decompress digital video data according to given file format[٨]. And can be either software application or part of hardware as explained in figure (١-١). They process the video through complex algorithm that compress and decompress video files. Video codec implemented in hardware is most efficient way to code and decode video file, they are faster and demands much less processor power than a software codec. hardware codec are often use at video conferencing where the equipment at the sender and receiver are configured at the same way. Software codecs are cheaper and there are a lot of codecs on the internet to use for free such as (MPEG-٢, MPEG-٤, H.٢٦١ and H.٢٨٣) . Many of these codecs give good result but demand a lot of processor power. Another disadvantage is the slow process to analyze a video file and to compress it. In general video codecs can be divided into two groups depending on what technique they use " temporal compression" and spatial compression, which are both work with lossy compression which means information over whelming or not noticed by the viewer is deleted [٧]. After we are introduced to redundancy which is contained in image sequence of video file we are used DCT to reduce spatial redundancy, and differences between successive frames to reduce temporal redundancy by using skew measurement to measure asymmetry between them by calculating standard deviation and mean which tell us about contrast and brightness respectively to compress video file and stored it in another form . To construct video file again we are needed to store all necessary information about AVI header and keep the same structure of AVI file .

١.٢ General Concepts

١.٢.١ Digital Image Processing (DIP):

DIP is one of the important fields in signal processing that is concerned with computer processing of images. these images come from many sources such as digital camera, scanners, and are stored as a file of specific format. In general the purpose of digital image processing is to enhance or improve the image in some way, or to extract information from it [٩]. Typical operations are to :

١. remove the bluer from image.
٢. smooth out the graininess.
٣. segment on image into regions such as objects and background.

٤. remove distortion from an image.

٥. improve the contrast on other visual properties of an image prior to display it.

٦. magnify, minimize, or rotate an image.

٧. code the image into some efficient way for storage and transmission.

١.٢.٢ Color Space (CS):

CS is transformation of RGB channels into luminance / chrominance to remove subjective redundancy contain in video data because human eye is more sensitive to brightness changes [١٠].

١.٢.٣ Motion Compensation (MC):

MC is powerful tool to reduce temporal redundancies between frames and is used extensively in MPEG-١ and MPEG-٢ video coding standards as a prediction technique for temporal coding the concept of motion compensation is based on the motion estimation of motion between successive video frames, (i.e. if all elements in a video scene are approximately spatially displayed), the motion between frames can be described by limiting number of motion parameters (i.e. by motion vectors for translator motion of pixels)[٥].

١.٢.٤ Discrete Cosine Transform (DCT):

DCT is most usual video compression algorithm that tests an image at regular intervals, by converting spatial amplitude into spatial frequency data. Lower frequencies contributes with more picture information than the higher frequencies does. This results in possibility to change an image to frequent components and throw away a lot of higher frequencies[١١] .

١.٢.٥ Fidelity Criteria (FC):

FC is measurement of image fidelity which can be divided into “subjective” and “objective”[١٢].

١. Subjective Fidelity Criteria :

Which require the definition of a qualitive scale to assess image quality. This scale can then be used by human test subjects to determine image fidelity. Subjective testing is performed by creating a data base of image to be tested, gathering group of people that are representative of desired population, and then evaluate the image according to predefined scoring criterion.

٢. Objective Fidelity Criteria :

Are borrowed from digital signal processing and information theory and provide us with equation that can be used to measure the amount of error in the reconstructed (decompressed) image. These equation are :

١. Root – Mean – Squared Error (***e RMS***):[٣]

$$eRMS = \left[\frac{1}{MN} \sum_{X=0}^{M-1} \sum_{Y=0}^{N-1} \left[\widehat{f}(x,y) - f(x,y) \right]^2 \right]^{\frac{1}{2}} \quad \dots (١.١)$$

f̂ (x ,y) decompressed image, ***f(x ,y)*** original image, ***MN*** number of pixels in row and column, ***x*** is the location row, ***y*** is the location column .

٢. Signal –To- Noise Ratio (***SNR rms***):

$$SNRrms = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \widehat{f}(x,y)^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left[\widehat{f}(x,y) - f(x,y) \right]^2} \quad \dots (١.٢)$$

٣. Peak Signal-To- Noise Ratio (***SNR peak***):[١٣].

$$PSNR = 10 \log_{10} {(255^2)}/{eRMS} \quad \dots (١.٣)$$

١.٢.٦ Compression Performance (CP):

Several quantities are commonly used to express the performance of compression method - which means high compression with sufficient quality- that explain below [١٤] .

١. Compression Ratio (***CR***) which is also called Bit Per Bit (BPB) is defined as :

CR= size of output stream /size of input stream ***… (١. ٤)***

٢. Compression Factor (***CF***) is the inverse of the compression ratio defined as :

CF= size of input stream/ size of output stream ***… (١. ٥)***

٣. The expression: ***١٠٠*(١-CR).*** ***… (١. ٦)***

٤. Bit Per Pixel (***BPP***) is commonly used in image compression ,it is equals the number of bits needed ,on average , to compress one pixel of the image . ***BPP= number of bits / number of pixels*** ***…(١. ٧)***

١.٢.٧ Skewness

The skew is to measure asymmetry about the mean in the gray level distribution . it is defined as

$$SKEW = \frac{1}{\sigma g^{3}} * \sum_{g=0}^{L-1} (g - \hat{g})^{3} * p(g) \qquad \text{…(١.٨)}$$

where ***σg*** is the standard deviation , ***ĝ*** is the mean , ***p(g)*** is the probability of histogram Another method to measure the skew uses mean, mode, and standard deviation where the mode is defined as the peak, or highest value :

$$SKEW = \hat{g} - \text{mod}e \big/ \sigma g \qquad \text{…(١. ٩)}$$

This method of measuring the skew is more computationally efficient, especially considering that, typically, the mean and standard deviation have already been calculated [١٢].

١.٢.٨ Fundamental Concepts of Digital Video

In the area of digital video – which is defined as a type of video system that work by using digital representation of the brightness and color of each pixel of the image, black and white also possible- there are several functions that make video clips as small as possible, which is essential for streaming. They are all part of the most basic concepts in digital video for understanding and the knowledge in this subject you should therefore be familiar in what these words mean : Fps, Key Frame, Image Size, Bit Depth, Video System and Data rate .

a- Frame Per Second (FPS)

With the expression " frame per second ", it means the number of images or frames that are shown per second in a video clip. fps is also used to synchronize image and sound [١٥]. This is important for movie quality since the higher images per second, the softer and the pleasant movie experience. When talking about fps and streaming video it is important higher image per second also means bigger file size. For that reason there are a rule of how high fps that can be used depending on the internet connection speed, in general ordinary TV program (Europe) is in ٢٥ fps, movie in full size in ٣٠ fps and digital video for the internet about ١٥ – ٣٠ fps [١٦].

b-Key Frame and Delta Frame

Is a technique that is used to reduce file size when compressing video. The information describing an image / frame is stored in a key frame. The changes in the following frames are then stored in delta frame [١٧].The key frame is then used as a reference for the delta frame to compare with. Normally when using key frames this is used each tenth or fifteen image, depending on how much the content change [١٨] .

c- Image Size

The image size on a general video clip is ٦٤٠* ٤٨٠ pixels and has been this from the old standard of the VGA screens. Therefore a full screen video is ٦٤٠*٤٨٠ pixels. If you then divide the size in half you will get ٣٢٠*٢٤٠ which is called half screen, even if it really mean a fourth screen for VGA. Some image size that are ordinary on internet by streaming video are the middle format ٢٤٠*١٨٠ and the quart image size ١٦٠*١٢٠ [١٩]. But with increasing broad band connections the image size ٣٢٠*٢٤٠ is getting more usual. The high broadband connections size ٤٠٠*٣٠٠.

d- Bit Depth

Bit depth means how many different color shades, which can be showed at the same time. The higher bit depth, the larger the video clip will be in the size. Normally a color depth of ١٦ bits used for video clips, but today it is more common that modern codecs use ٢٤ bit color depth see table (١-١).

Table (١-١) number of color as a function of bit depth [١٩] .

Bit depth	١	٢	٨	١٥	١٦	٢٤
No. of color	٢	٤	٢٥٦	٣٢٧٦٨	٦٥٥٣٦	١٦٧٧٧٢١٦

e- Video System

With different broad cast of video there are different standards a round the world of video system: Europe PAL, American NTSC, French SECAM, PAL and SECAM uses ٦٢٥ lines that are shown on screen ٢٥ times per second. NTSC uses instead ٥٢٥ lines which are shown ٣٠ times per second [٢٠].

f-Interlaced and Progressive Scan

The technique for drawing lines in video images can be done by either interlace or progressive scan. Interlace is used for TV screens and works by first drawing even lines of the image and then filling in odd lines. Progressive scan is on the other hand the technique used for computer screens and in difference draws the image entirely from top to bottom [٢١].

g-Data Rate (Bit Rate)

Data rate or bit rate is measurement of the number of data which is transported per second in a video clip. To calculate how much data rate an uncompressed video clip will take, it can be done with the following formula [١٩].

Data Rate= image size *bit depth * frame per second ***.... (١.١٠)***

as an example of this can be seen in table (١- ٢)

Table (١-٢) Typical values of Data Rate

Image size	Bit depth	Frame per second	Data rate
٧٢*٥٧٦	٢٤	٢٥	٣١,١٠٤ MB/S
٦٤٠*٤٨٠	٢٤	٢٥	٢٣,٠٤ MB/S
٤٠٠*٣٠٠	٢٤	٢٥	٩,٠٠ MB/S
٣٢٠*٢٤٠	٢٤	٢٥	٥,٧٦ MB/S
١٦٠*١٢٠	٢٤	٢٥	٢٨٨ KB/S

Lets say that we use the example four from table ١-٢ with most use image size for streaming : ٣٢٠*٢٤٠, ٢٥ fps, and ٢٤ bit depth. To calculate the Data Rate we use the form and multiply :

٣٢٠*٢٤٠ *(٢٤/٨)*٢٥ = ٥٧٦٠٠٠٠ Byte / S (٥,٧٦ MB / S). Data rate is one important key why video compression is needed the high data rate is required for uncompressed clip with an image size ٧٢٠*٥٧٦ is far too high to be played on modern computer today. limiting data rate by compressing the video clip is therefore essential for playback and in term of streaming, where the data rate is controlled by the users. Internet connection, of course lower data rate means lower quality since less data information is processed in each second [١٩] .

١.٣ Literatures Survey

Neff and Zakhor (February-١٩٩٩) have applied the Matching Pursuits MP technique to code the motion prediction error signal, because of in hybrid compression system after predicts each frame from its neighboring frames, then compress the prediction parameter and produces the prediction error frame, which is code by using DCT, although DCT video coding is efficient, it introduces undesirable blocking artifacts, especially at low bit – rates and some blocks are represented by one or small number of coarsely quantized transform coefficients resulting in artifacts known as ringing and mosquito noise. The MP coder divides each motion residual error into blocks, and measures the energy of each block. The center of the block with the largest energy value is adopted as an initial estimate for inner product search. A dictionary of Gabor basis vector is then exhaustively matched to an S*S window around the initial estimate.

The location, basis vector index, and the value of the target quantized inner product are then coded together. Video sequence coded using MP do not suffer from either blocking or ringing artifacts since the basis vector are only coded when they are well matched to residual error, but the distortion appears in MP by increasing blurriness or loss some detail[٢٢].

Neri Merhav & Vasudev Bhaskaran (٢٠٠٠) proposed new algorithm converts motion compensated compressed video into sequence of DCT-domain blocks corresponding to the spatial domain blocks of the current frame alone, without prediction based on other frame, i.e. removing the inter – frame element of compression /decompression. This algorithm used for undoing the motion compensated operation in the DCT – domain, therefore the algorithm receives as the input DCT blocks of the motion compensated compressed video, and provide DCT blocks of the corresponding spatial domain, blocks of the current frame alone without Reference to past or future frames. This operation of canceling motion compensation enable video compositing in the DCT compressed domain as well as several composite operations, e.g. scaling, overlapping , translation and filtering, etc. This proposed algorithm save ٤٧٪ computations compared to brute – force approach – which is used to composite compressed video streams from several resources into a single composite video stream – which can be replaced with the proposed algorithm i.e. compositing pixel- by – pixel in the spatial domain, and recompressing the composite stream [٢٣].

Stefan Einerman (December -٢٠٠١) submit thesis in residual coding which is lossless video compression. He proposed method of residual video coding such that the residual sequence is obtain by differencing between original video and its MPEG-٢ coded version :

Residual Sequence = Original – MPEG- ٢ ... ***(١.١١)***

The resulting residual sequence will contain what can be called error frames, the resulting histogram of the residual or error frame will be highly peaked around zero, so that a distribution can easily be further compressed by an entropy coder such as Huffman. The compression ratio of residual coding is measured by dividing size of original on (size of MPEG-٢ code plus size of residual code) which reach ١.٧ for an average video clip , which tend to be better to existing lossless method[٢٤].

Amir Said (may-٢٠٠٤) proposed new coding system, in this system the encoder perform coding depend on coding image block classification method that is very simple and low computational – complexity, so that the compound image or video data is divide into blocks of certain size (like ٨*٨), and then classification method is applied to blocks to decide which compression method is to be used to code its pixels, the classification information is also coded before coding the block pixels, and usually it depends only on the values of pixels inside a block and possibly the decision on the neighboring blocks, but also first encodes some information about the number of blocks that have the same classification (group), then code the classification, and finally codes whole group of block. In this proposed encoder using two techniques first is classification based on color (i.e. if block of pixels have small number of colors than threshold (T= ٩)) , then he use lossless method to encode what colors in the block and then what is location of these colors as well as regions with small colors have normally small entropy for coding. Second technique is the identification of block that need high quality such as smoother transition in the edges make the DCT coding more efficient and artifacts less pronounced, thus the proposed detection method by using y-color component and taking the absolute difference to get information about block activity and presence of the edge then analyzing histogram of differences, the block with text and graphics normally have histogram containing a few isolated peaks therefore defined entropy function, which is in classification and coding by putting these two techniques together[٢٥] .

Ayman Darwish (November-٢٠٠٥) submitted thesis in scalable video coding, in it he used DWT to reduce Spatial Redundancy at ١,٢,٣,٤-level and use F which is quantization factor at ٨,١٦,٣٢, when increasing F result in more quantization and worse quality reconstructed image. To reduce Temporal Redundancy use Overlapped Block Motion Compensation(OBMC) which uses same number of block in Block Motion Compensation (BMC)and each block has same center but each block has been enlarged to twice for each width and height of its counter part in BMC. Therefore the enlargement naturally creates overlapped between blocks such that each block overlap a quadrant wise with all eight of immediate neighbors. For example the top-left quadrant of block overlaps with left, top-left, and top neighbors of the block. Therefore any given pixel incurrent frame belongs to four of these overlapping blocks . OBMC instead of simply copying block from reference into prediction frame , it first multiplies each block from reference frame into wise with a window function and then add the resultant overlapping blocks together in prediction frame, so that each pixel values taken from different location in reference frame. DWT can not be applied to small block of BMC effectively, but must be applied to as a large area of image as possible to reap full benefit of DWT by maximizing its efficiency and energy compaction effect[٦].

١.٤ Thesis Aim

This thesis aims to design and implement a coder/decoder software for video clip file using lossy digital video compression and AVI movie file - as a type of file under which windows was treated with- to reduce the size of AVI video file for storage and transmission through the limited bandwidth by converting source video file which is a sequence of frames into another form which can be represented in fewer bit, and then change it back for using as AVI movie file again .

١.٥ Thesis Layout

Thesis has been structured as five chapters:

- Chapter one gives an introduction to the digital video compression with general concepts, literature survey and thesis aim .
- Chapter two explains the general techniques which are used in lossy compression algorithm for video .
- Chapter three shows some video clip image compression algorithm such as MPEG-٢, H.٢٦X, and structured AVI file .
- Chapter four (practical section) shows suggested system .
- Chapter five shows proposed system results ,conclusions, future works.

Chapter Two

General Techniques

٢.١ Introduction

In this chapter we explain some general techniques which are used for compression to reduce the redundant information contain in images and video sequence which are come in various formats that can be defined by color space, sampling, spatial resolution and temporal resolution for video.
The first technique is color space which are used to decorrelate image pixel with sampling to reduce size of color component.
The second technique is used motion estimation /compensation which are used to find similarities between successive frames and differences.
The third technique is used DCT to convert image data into frequency domain with quantization process to produce large number of zero coefficients.

٢.٢ Color Space (CS)

To describe color space, we must know what is color which is defined as the brain reaction to a specific visual stimulus. although we can precisely describe color by measuring its spectral power distribution (the intensity of visible Electro – magnetic radiation at many discrete wave lengths) this leads to large degree of redundancy. The reason for this redundancy is that the eye's retina samples color using only three board bands, roughly corresponding to red, green, and blue light [٢٦][٢٧].
Actual information stored in digital image is the brightness information in each band (i.e. red, green, and blue), when image display, the corresponding brightness information is displayed on screen by picture element that emit light energy corresponding to that particular color ,therefor to use image in some application such as compression we need to transform color from RGB

into mathematical color space that de-couples the brightness information from color information, this is done by using transformation of color space, the image information consist of a ١-D brightness, or luminance, and ٢-D color space, now ٢-D color space does not contain any brightness information, but typically contains information regarding the relative amounts of the different colors.

٢.٢.١ Computer Graphics Of Color Space

Traditionally the CS used in computer graphics have been made designed for specific devices: RGB for the CRT displays and CMY(K) for printers. Most computer graphics CS is commonly used :

١. Computer RGB Color Space

This is CS produced on CRT (or similar) display when pixel values are applied to graphics card. RGB space may be visualized as cube with the three axis corresponding to red, green, and blue. The bottom corner, red = green = blue = ٠ is black, while the opposite top corner, where red = green = blue =٢٥٥ (for ٨ bit per channel display system) is white. RGB is frequently used in most computer applications since no transform is required to display information on the screen. For this reason it is commonly the base CS for most applications. The RGB CS is not efficient representation for compression because there is significant correlation between color component in typical RGB images for compression, a luminance – chrominance representation such as (YCbCr, YIQ, YUV, etc) is considered superior to RGB representation because of lower inter- component correlation. RGB images are acquired from color cameras and scanners are therefore transformed to one of the luminance – chrominance spaces prior to compression. These conversion introduces some error caused by rounding of the results of arithmetic operations, therefore RGB space may be used for strictly lossless compression with law compression ratio [٢٨].

٢. TV Color Space

Is the television transmission CS or transmission primaries YUV and YIQ are analogue for NTSC and pal system respectively while YCbCr is digital standard, these CS separate RGB into luminance – chrominance information and are useful in compression applications(both digital and analogue). The luminance in the range (٠ – to – ٢٥٥) and chrominance is bipolar (positive as well as negative) [٢٦]: -

a- European YUV (EBU)

European TV (PAL and SECAM coded) uses YUV components y is similar to perceived luminance, U and V carry color information and some luminance information and are bipolar. the coding equation for non-linear signals are [٢٧]:-

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} ٠.٢٩٩ & ٠.٥٨٧ & ٠.١١٤ \\ ٠.١٤٧ & -٠.٢٨٩ & ٠.٤٣٦ \\ ٠.٦١٥ & -٠.٥١٥ & -٠.١٠٠ \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

... (٢.١)

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} ١ & ٠.٠٠٠ & ١.١٤٠ \\ ١ & -٠.٣٩٦ & -٠.٥٨١ \\ ١ & ٢.٠٢٩ & ٠.٠٠٠ \end{bmatrix} * \begin{bmatrix} Y \\ U \\ V \end{bmatrix}$$

b- American YIQ

American TV (NTSC coded) uses YIQ component. Again y is similar to perceived luminance, I and Q is carry color information and some luminance information. The coding equation for non-linear signals are [٢٧]:-

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} ٠.٢٩٩ & ٠.٥٨٧ & ٠.١١٤ \\ ٠.٥٩٦ & -٠.٢٧٤ & ٠.٣٢٢ \\ ٠.٢١٢ & -٠.٥٣٢ & -٠.٣١١ \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

... (٢.٢)

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} ١ & ٠.٩٥٦ & ٠.٦٢١ \\ ١ & -٠.٢٧٢ & -٠.٦٤٧ \\ ١ & -١.١٠٥ & ١.٧٠٢ \end{bmatrix} * \begin{bmatrix} Y \\ I \\ Q \end{bmatrix}$$

It is possible to define a transformation matrix between EBU YUV and NTSC YIQ, the non- linear connecting equations are :-

$$\begin{bmatrix} I \\ Q \end{bmatrix} = \begin{bmatrix} -٠.٥٤٧ & ٠.٨٤٣ \\ ٠.٨٣١ & ٠.٥٤٧ \end{bmatrix} * \begin{bmatrix} U \\ V \end{bmatrix}$$

... (٢.٣)

$$\begin{bmatrix} U \\ V \end{bmatrix} = \begin{bmatrix} -٠.٥٤٧ & ٠.٨٤٣ \\ ٠.٨٣١ & ٠.٥٤٧ \end{bmatrix} * \begin{bmatrix} I \\ Q \end{bmatrix}$$

c- Digital YCbCr

This is international standard for digital coding of TV pictures. It deals only with digital representation of RGB signals in YCbCr form. the non-linear coding matrices are [٢٩]:-

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} ٠.٢٩٩ & ٠.٥٨٧ & ٠.١١٤ \\ -٠.١٦٩ & -٠.٣٣١ & ٠.٥٠٠ \\ ٠.٥٠٠ & -٠.٤١٨ & -٠.٠٨١ \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

... (٢.٤)

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} ١ & ٠.٠٠٠ & ١.١٤٠\ ٢١ \\ ١ & -٠.٣٤٤١ & -٠.٧١٤٢ \\ ١ & ١.٧٧١٨ & ٠.٠٠٠ \end{bmatrix} * \begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix}$$

٢.٢.٢ Sampling

Sub-Sampling reduces the number of samples temporally or spatially and this results in the loss of higher frequency information [٢٩]. After conversion from RGB signals to luminance – chrominance system which take the advantage of the human eye's which is lower sensitivity to color information, The chrominance components are known as color components are mostly sub-sampled with respect to luminance component[٢٨]. Therefore When working with RGB images, we use exactly the same number of bits to store the three color components. When working with YCC video, by taking this advantage of the peculiarity of human perception previously mentioned- the eye is much more sensitive to changes in the luminance of an image than to the color (chrominance) So, instead of storing the same amount of information for each of the YCC components, professional video only needs to store half as much color information as it does luminance information. This is also known as ٤:٢:٠ color, which means that for every four samples of luminance values, there are only ٢ samples of each color signal. See Figure (٢-١). This helps save bandwidth during analog transmission, as well as storage space in the digital realm. YCC can be reduced even further to what is known as ٤:١:١ color. DV cameras save video in the ٤:١:١ space in order to reduce storage needs [٢٩]. The basic video chrominance sampling schemes are shown in figure (٢- ١) [٢٨][٣٠].

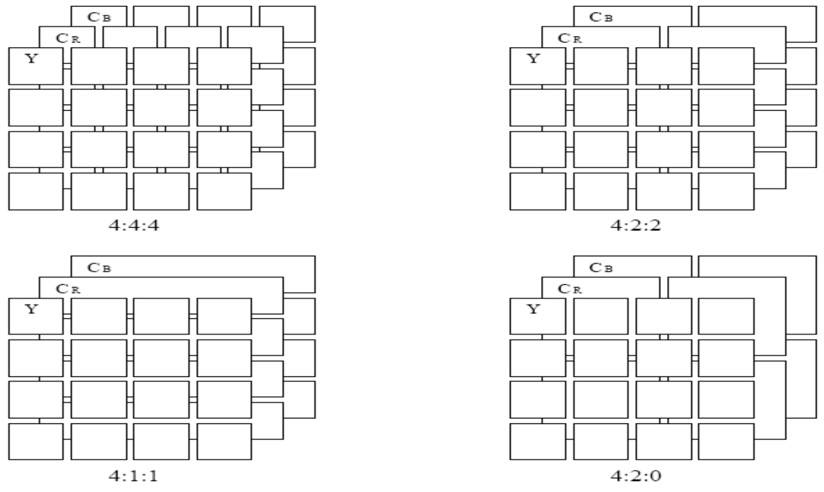


Figure (٢- ١) -Illustration of SAMPLING TYPE

The ٤:٢:٢ sampling scheme is sub- sampled to one half in the horizontal dimension and is used in high – fidelity applications like contribution – quality television while ٤:٢:٠ which is sub-sampled to one - half in both vertical and horizontal dimension is very common high compression applications. Basic video formats are summarized in table (٢-١) ,SIF (Standard Intermediate Format) ,CIF (Common Intermediate Format) and QCIF (Quarter CIF) , and Sub-QCIF (sub- Quarter CIF) are defined. QCIF operation is mandatory, while CIF operation is optional [٣٠].

Table (٢-١) Illustrate Sampling Type in vertical and horizontal dimension with Video Formats (SIFif & CIF) .

Sampling Type	SIF	CIF	QCIF	Sub-QCIF
	٤:٢:٢	٤:٢:٠	٤:٢:٠	٤:٢:٠
Lum. format	٧٢٠*٥٧٦	٣٥٢*٢٨٨	١٧٦*١٤٤	١٢٨*٩٦
Chro. format	٣٦٠*٥٧٦	١٧٦*١٤٤	٨٨*٧٢	٦٤*٤٨
Frame rate	٢٥-٣٠	٢٥-٣٠	٢٥-٣٠	٣٠

٢.٣ Motion Compensation / Estimation

In recent year video compression has played a vital role in data storage and transmission as we said previously, video compression is involved with removal of the spatial and temporal redundancy, therefore intra – frame which are used to remove spatial redundancy and inter –frame is used for to remove temporal redundancy, by using this fact the current frame is often not entirely different from the previous one. Many part of blocks of a frame are identical to corresponding block the previous frame [٣١][٣٢] .

Coding efficiency is achieved with prediction from the previous reference frame, prediction consist of two process motion estimation and motion compensation. ME is process that attempts to find the best match of a block of pixels with the previous frame [٣٣].

ME can be classified into three types which are optical flow equation based methods, peel recursive methods, and block matching methods which are seemingly used by the video compression standards because it is achieve a good balance between complexity and coding efficiency [٣٣]. The goal of ME is to find a vector that points to a block of pixels in previous frame [٣٤].

There are many type of ME – block matching methods which divide the frame into macro blocks of size (١٦ * ١٦) pixel, also assume the macro block is fit entirely within previous frame (i.e. not hang over the edge) and the motion vector is the relative coordinates between the upper- left -hand corner of the reference block and the candidate block. Then, the upper- left-hand pixel of the candidate block must fit within the area denoted in gray in the figure (٢-٢) , a suitable macro block that fits gray area in figure (٢-٢) , this give possibility of ١٦١ *١٢٩, or, ٢٠٧٦٩ motion vectors checked per macro block, with ٩٩ macro block there are over ٢ million motion vector to be checked per a single frame [٣٥] .

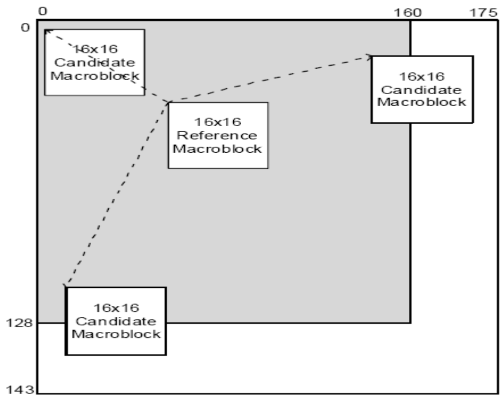

0
160
175
0
128
143
16x16 Candidate Macroblock
16x16 Reference Macroblock
16x16 Candidate Macroblock
16x16 Candidate Macroblock

Figure (٢-٢) Illustration of ME computational complexity of entire QCIF

Usually ,then the area of motion estimation is restricted to what is called a search area .this greatly limits the number of possible candidate blocks from impractical statistics mentioned previously .For example search area is limited to[-١٦,١٥] under normal operation [٣٥]. Figure (٢-٣) illustrates this case .

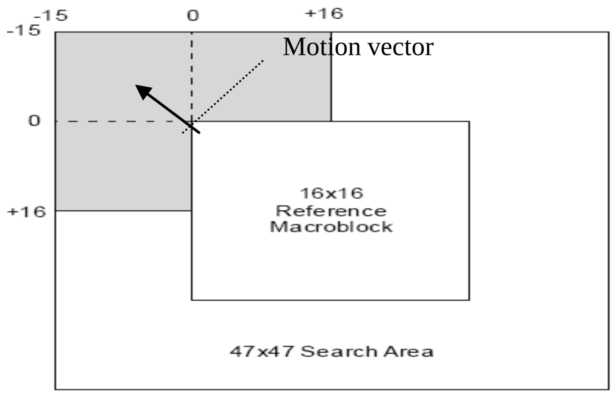


Figure (٢-٣) Illustration of Motion Estimation Under H.٢٦٣

Again, the upper-left-hand corner of the reference block can be " shifted " any where inside the gray area. The entire search area is ٤٧ *٤٧ pixels to accommodate all the motion vectors. Now there are only ٣١ *٣١ or ٩٦١ motion vectors to be checked per macro block and ٩٥١٣٩ per frame [٣٥].

٢.٣.١ Matching Criteria

When trying to find the best matching block, there has to be some way to compare the two block. Matching criteria, e.g., a Block Distortion Algorithm (BDA), is what is used to do this it is very important that the distortion function is fast it is used in every comparison and if the distortion function is slow then the whole block matching will be slow, mean square difference(error), mean absolute difference, pel difference classification , and sum absolute difference function:[٣٤]

١. Mean Absolute Distortion Function (*MAD*)

$$MAD(A,B) = \frac{1}{mn}\sum_{p=0}^{m-1}\sum_{q=0}^{n-1}\left|A(p,q) - B(p,q)\right| \qquad \text{... (٢.٥)}$$

where ***A(p,q)*** is luminance intensity of the pixel at the location ***(p,q)*** in the original block, ***B(p,q)*** is luminance intensity of the pixel at the location in the target block, ***m*** and ***n*** are the width and the height of the block respectively measured in pixels.

٢. Mean –Square – Difference (*MSD*)

$$MSD(A,B) = \frac{1}{mn}\sum_{p=0}^{m-1}\sum_{q=0}^{n-1}\left(A(p,q) - B(p,q)\right)^2 \qquad \text{...(٢.٦)}$$

٣. Pel Difference Classification

If the difference is less than some predetermined threshold then there is match. The number of matches is then counted with the help of the Ord(n). Ord(n) is one if n is true and zero if n is false.The smaller the difference is the greater the number of matching pixels, is the batter is the match [٣٤].

$$PCD(A,B) = \sum_{p=0}^{m-1}\sum_{q=0}^{n-1}\left[Ord\left(\left|A(p,q) - B(p,q)\right| < T\right)\right] \qquad \text{... (٢.٧)}$$

where ***T*** is the threshold

٢.٣.٢ Motion Estimation Algorithms:

٢.٣.٢.١ Full-Search Block-Matching Algorithm

The Full-Search Block-Matching (FSBM) algorithm is the most straightforward motion estimation operation. Simply speaking, it calls for searching the entire allowable search area for the best motion vector-the one with the best MAD rating. Obviously, this algorithm always finds the optimal motion vector for the given search area. It should also be obvious this algorithm suffers from the largest computational complexity. For example, considering the search area of [-٧, +٧], there are ٢٢٥ MAD calculations that must be performed for each macro block in a frame. Again with ٩٩ macro blocks in a QCIF-sized frame, that translates to ٢٢,٢٧٥ MAD calculations per frame of video to perform motion estimation[٣٥].
The advantages of implementing FSBM as the motion estimation algorithm is achieving the optimum results for any given search window [٣٤]. Disadvantages of the FSBM algorithm lie in the large amounts of area required to implement the systolic arrays, the high throughput requirements to calculate all the MADs and the large amount of power required to drive the systolic arrays [٣٥].

٢.٣.٢.٢ ٢ -D Logarithmic Search

It was the first block matching algorithm that utilized the quadrant monotonic principle – which assumes that the value of distortion function increase as the distance from the best matching block increase, that algorithm is divided into number of steps [٣٤].

First step:

If the maximum displacement is d and the step size is **s**, then the first five locations to be examined are [٠,٠], [٠,**s**], [**s**,٠], [٠,-**s**], and [-**s**,٠] . in the figure (٢-٤) the first points to be examined are marked with number ١. Maximum displacement is the largest number of pixels away for the starting point that the algorithm looks for the best matching block. In this figure this ٨ along the vertical axis and ٨ along the horizontal axis sometimes the maximum displacement is larger along the horizontal axis because of the assumption that the objects generally move more along that direction. Step size in the first step set to ٤ .

Second step :

The search is now centered on the location that produced the least distortion. If the location is the center position then the step size is halved. In the example shown in figure (٢-٤), the location marked number ١ with the least distortion is located at the point (٠,٤), therefore the step size remains the same and the second search is centered on point (٠,٤), but the search number three has the least distortion in the middle search location (-٤,٤), which means that the step size is reduced by factor ٢, thus becoming ٢ .

Third step:

If the step is equal to ١, the search area is minimal, then all the eight locations around the center are examined to find the best match for the block .

	-٨	-٧	-٦	-٥	-٤	-٣	-٢	-١	٠	١	٢	٣	٤	٥	٦	٧	٨
-٨					٣				٢								
-٧																	
-٦			٥		٤												
-٥		٦	٦	٦													
-٤	٣	٦	٤	٦	٢		٤		١				٢				
-٣		٦	٦	٦													
-٢			٥		٤												
-١																	
٠					١				١				١				
١																	
٢																	
٣																	
٤									١								
٥																	
٦																	
٧																	
٨																	

Figure(٢- ٤) : ٢-D Logarithmic Search Example Searches

٢.٣.٣ Motion Compensation (MC)

Is the process that is take Y component of a frame and divided into macro block ١٦ *١٦ , which used to find motion vector by using one of the motion estimation algorithm that is used then in motion compensation to create motion compensated version of the current block of pixels. This new macro block is basically the difference of the two "matching "macro blocks and It is defined by this equation :

$$E(x,y,t) = I(x,y,t) - I(x-u,y-v,t-١) \qquad \text{... (٢.٨)}$$

In equation (٢.٨), the ***(u, v)*** respectively represent the ***x*** and ***y*** coordinates of the motion vector found during motion estimation. the third dimension is a time , hence ***I(x, y, t)*** represent the intensity at the position ***x*** and y at a time ***t*** , thus the previous frame block is being subtracted from the current block of pixels. The out put of motion compensation block can then be transformed by using DCT to remove spatial correlation and then transmitted with motion vector cross the channel to receiver.

The decoder can inverse this operation by performing its own motion compensation to retrieve the current block of pixels, albeit degraded by the quantization factor and any losses from the DCT inverse transform . (Refer to Figure ٢-٥ for a picture of the encoder and decoders). Compression is achieved if the blocks “match up” well, so maximum compression depends on the success of motion estimation. [٣٤][٣٥] .

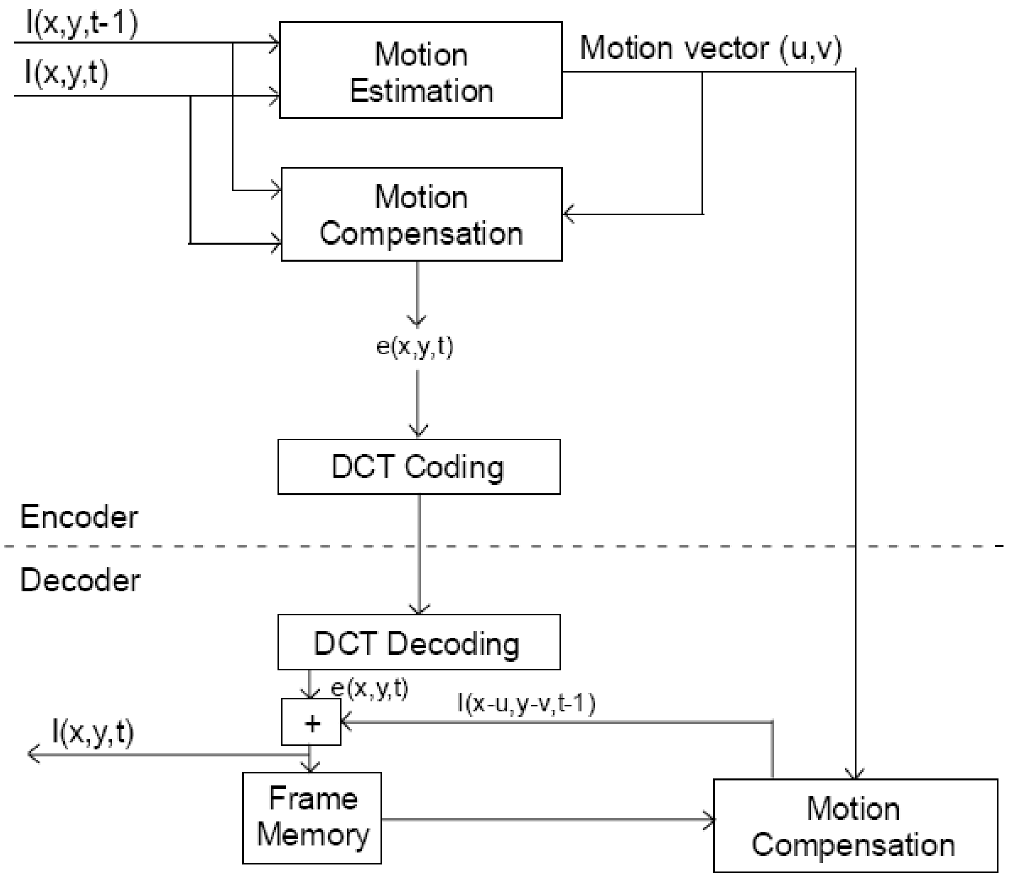


Figure (٢-٥) : Generic Video Encoder and Decoder Supporting Motion Estimation/Compensation

٢.٤ Discrete Cosine Transform (DCT)

٢.٤.١ Transform

The transform maps image data into different mathematical space via transformation equation [١٢]. DCT is widely used transform in image processing, especially for compression. Some of the applications of two ٢-Dimensional DCT involve still image compression and compression of individual video frames, while multi- Dimensional is mostly used for compression of video streams and volume spaces [٣٦].

٢.٤.٢ Discrete Cosine Transform DCT

Like other transforms, the Discrete Cosine Transform (DCT) attempts to decorrelate the image data. After decorrelation each transform coefficient can be encoded independently without losing compression efficiency [٣٧]. This section describes the DCT and some of its important properties.

٢.٤.٢.١ The One -Dimensional DCT

The most common DCT definition of a ١-D sequence of length N is

$$C(u) = \alpha(u)\sum_{x=0}^{N-1} f(x)\cdot Cos\left[\frac{\Pi(2x+1)u}{2N}\right] \quad ...(٢.٩)$$

for u = ٠, ١ , ٢, ...,N- ١ .Similarly, the inverse transformation is defined as

$$f(x) = \sum_{u=0}^{N-1}\alpha(u)\cdot C(u)\cdot Cos\left[\frac{\Pi(2x+1)u}{2N}\right] \quad ...(٢.١٠)$$

for u= ٠, ١ , ٢, ...,N- ١. In both equations**(٢.٩)** and **(٢.١٠)**$\alpha(u)$ is defined as

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}} & for\ u=٠ \\ \sqrt{\frac{2}{N}} & for\ u\neq ٠ \end{cases} \quad ...(٢.١١)$$

It is clear from **(٢.٩)** that for $u=0, C(u=0)=\sqrt{\frac{1}{N}}\sum_{x=0}^{N-1} f(x)$. Thus, the first transform coefficient is the average value of the sample sequence. In literature, this value is referred to as the *DC Coefficient*. All other transform coefficients are called the *AC Coefficients*. To fix ideas, ignore the *f(x)* and *α(u)* component in **(٢.١٠)** .The plot of $\sum_{x=0}^{N-1} f(x)\cdot Cos\left[\frac{\Pi(2x+1)u}{2N}\right]$ for *N* =٨ and varying values of *u* is shown in Figure (٢-٦). In accordance with our previous observation, the first the top-left waveform (*u* = ٠) renders a constant (*DC*) value, whereas, all other waveforms (*u* =١, ٢, …,٧) give waveforms at progressively increasing frequencies [٣٨]. These waveforms are called the *cosine basis function*. Note that these basis functions are orthogonal. Hence, multiplication of any waveform in Figure (٢-٦) with another waveform followed by a summation over all sample points yields a zero (scalar) value, whereas multiplication of any waveform in Figure (٢-٦) with itself followed by a summation yields a constant (scalar) value. Orthogonal waveforms are independent, that is, none of the basis functions can be represented as a combination of other basis functions [٣٩]. Fig (٢-٦) show below

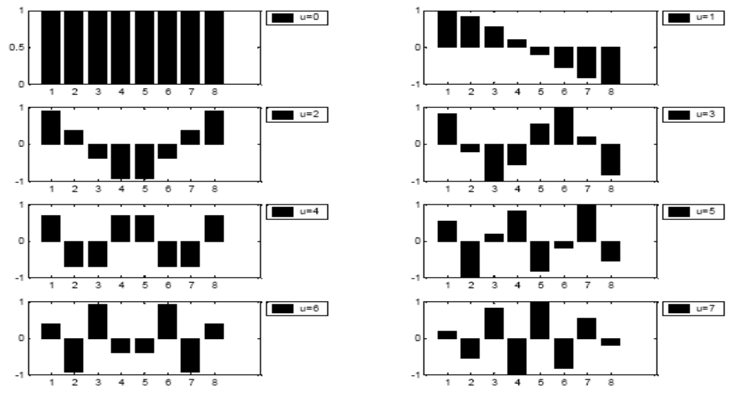


Figure (٢-٦) . One dimensional cosine basis function (*N*=٨).

If the input sequence has more than *N* sample points then it can be divided into sub-sequences of length *N* and DCT can be applied to these chunks independently. Here, a very important point to note is that in each such computation the values of the basis function points will not change. Only the values of *f(x)* will change in each sub-sequence. This is a very important property, since it shows that the basis functions can be pre-computed offline and then multiplied with the sub-sequences. This reduces the number of mathematical operations (i.e., multiplications and additions) thereby rendering computation efficiency.

٢.٤.٢.٢ The Two -Dimensional DCT

The ٢-D DCT is a direct extension of the ١-D case and is given by :

$$C(u,v) = \alpha(u)\alpha(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cdot Cos\left[\frac{\Pi(2x+1)u}{2N}\right]\cdot Cos\left[\frac{\Pi(2y+1)v}{2N}\right] \quad \textbf{...(٢.١٢)}$$

for *u, v* =٠,١,٢, , …, *N* - ١ , and *α(u)* and *α(v)* are defined in **(٢.١١)**. The inverse transform is defined as

$$f(x,y) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1}\alpha(u)\alpha(v)\cdot C(u,v)\cdot Cos\left[\frac{\Pi(2x+1)u}{2N}\right]\cdot Cos\left[\frac{\Pi(2y+1)v}{2N}\right] \quad \textbf{...(٢.١٣)}$$

for *x* , *y* =٠,١,٢,٣ ,……, *N* – ١ [٣٧] . The ٢-D basis functions can be generated by multiplying the horizontally oriented ١-D basis functions (shown in Figure (٢-٦)) with vertically oriented set of the same functions [٣٨]. The basis functions for *N* = ٨ are shown in. Again, it can be noted that the basis functions exhibit a progressive increase in frequency both in the vertical and horizontal direction. The top left basis function of results from multiplication of the DC component in Figure (٢-٧) with its transpose. Hence, this function assumes a constant value and is referred to as the DC coefficient.

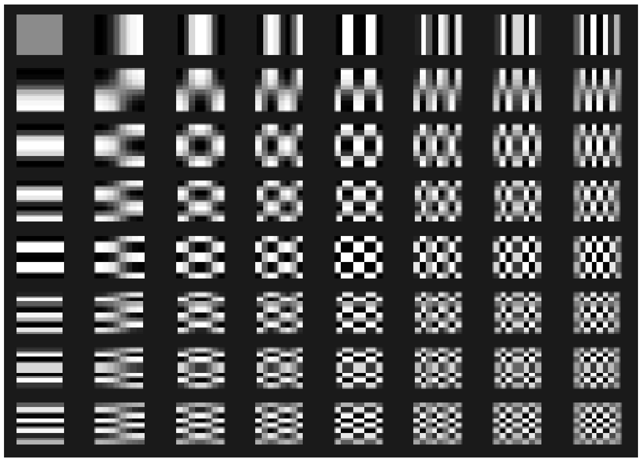


Figure ٢-٧: Two dimensional DCT basis functions (*N* = ٨). Neutral gray represents zero, white represents positive amplitudes, and black represents negative amplitude [٣٨].

As we said previously the basis function can be pre - computed offline for n= ٨ and the ٢-D DCT which is the amongst block transform that partitions an input image into non – overlapped blocks, then maps them into blocks of coefficients so as it can be used in JEPG & MPEG compression successfully at decorrelating and concentrating the energy of pixel data into spatial domain [٤٠]. Given an image of any array that partition into ٨*٨ blocks is then the DCT applied separately on ٨*٨ blocks of data so that the equation (٢.١٢) (٢.١٣) can be written as[٤٠]:

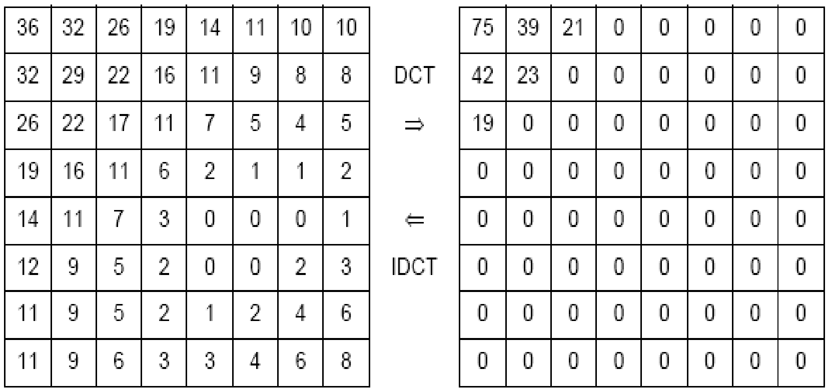
$$C(u,v)=\frac{\alpha(v)}{2}\cdot\frac{\alpha(u)}{2}\sum_{y=0}^{7}\sum_{x=0}^{7} f(y,x)\cdot Cos\left[\frac{\Pi(2x+1)u}{16}\right]\cdot Cos\left[\frac{\Pi(2y+1)v}{16}\right] \quad \text{...(٢.١٤)}$$

For *u, v* = ٠ ,١ ,٢ ,… *n*- ١.

$$f(x,y)=\sum_{v=0}^{7}\frac{\alpha(v)}{2}\cdot\sum_{u=0}^{7}\frac{\alpha(u)}{2}\cdot C(v,u).Cos\left[\frac{\Pi(2x+1)u}{16}\right]\cdot Cos\left[\frac{\Pi(2y+1)v}{16}\right] \quad \text{...(٢.١٥)}$$

for *x* , *y* = ٠ ,١ ,٢ ,…… *n*- ١. Where $\alpha = \begin{cases} ١/\sqrt{٢} & \text{if } u = ٠ \\ ١ & \text{if } u <> ٠ \end{cases}$

The DCT make it possible to compress the image by concentrating most of the image information in the lower spatial domain, so that the ٦٤ coefficients produced by that equation is represent as in figure (٢-٨). The coefficient of the top -left- corner is called DC which represent the average of whole image, and it is low frequency, the other ٦٣ coefficients is high frequency which is called AC [٣٩] .



36	32	26	19	14	11	10	10		75	39	21	0	0	0	0	0
32	29	22	16	11	9	8	8	DCT	42	23	0	0	0	0	0	0
26	22	17	11	7	5	4	5	⇒	19	0	0	0	0	0	0	0
19	16	11	6	2	1	1	2		0	0	0	0	0	0	0	0
14	11	7	3	0	0	0	1	⇐	0	0	0	0	0	0	0	0
12	9	5	2	0	0	2	3	IDCT	0	0	0	0	0	0	0	0
11	9	5	2	1	2	4	6		0	0	0	0	0	0	0	0
11	9	6	3	3	4	6	8		0	0	0	0	0	0	0	0

Figure ٢-٨ : The Effect of a Discrete Cosine Transform

Given this interpretation of the DCT , the way to lose the unimportant image information is to reduce the size of ٨*٨ numbers of coefficient especially the ones of the right bottom .

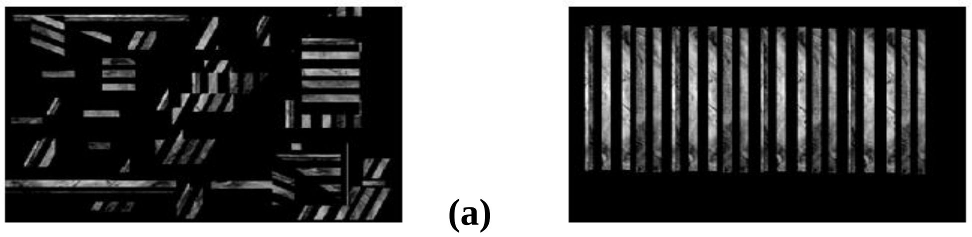
There is a chance that this won’t degrade the image quality much. In general these coefficients must be quantified and then compress using one of compression lossless method such as run length coding, Huffman, and etc as explain in section ٢-٥ .

٢.٤.٢.٣ Properties of DCT

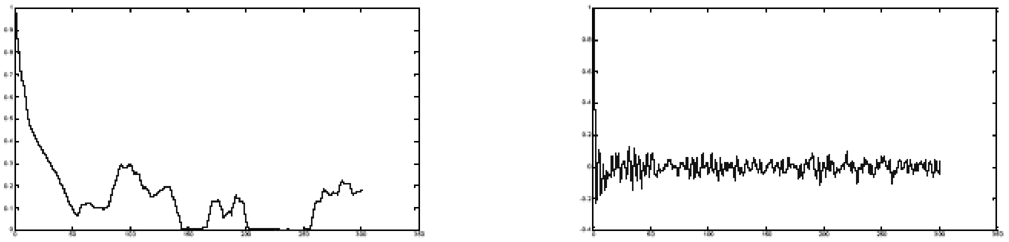
Discussions in the preceding sections have developed a mathematical foundation for DCT. In This section outlines(with examples) some properties of the DCT which are of particular value to image processing applications[٣٧].

a- Decorrelation

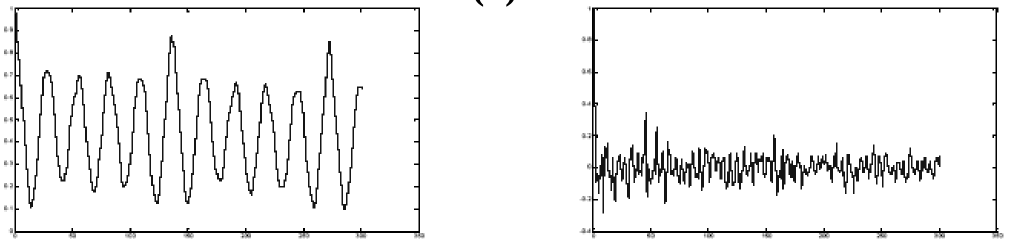
As discussed previously, the principle advantage of image transformation is the removal of redundancy between neighboring pixels. This leads to uncorrelated transform coefficients which can be encoded independently. Let us consider the example in Figure (٢-٩). The normalized auto correlation of the images before and after DCT is shown in that Figure. Clearly, the amplitude of the auto correlation after the DCT operation is very small at all lags. Hence, it can be inferred that DCT exhibits excellent decorrelation properties.



(a)



(b)



(c)

Figure ٢.٩: (a) Original image (b) Normalized autocorrelation of uncorrelated image before and after DCT; (c)Normalized autocorrelation of correlated image before and after DCT.

b- Energy Compaction

Efficacy of a transformation scheme can be directly gauged by its ability to pack input data into as few coefficients as possible. This allows the quantizer to discard coefficients with relatively small amplitudes without introducing visual distortion in the reconstructed image. DCT exhibits excellent energy compaction for highly correlated images. Let us again consider the two example images of Figure (٢-٩) . In addition to their respective correlation properties discussed in preceding sections, the uncorrelated image has more sharp intensity variations than the correlated image. Therefore, the former has more high frequency content than the latter. Figure (٢-١٠) shows the DCT of both the images. Clearly, the Uncorrelated image has its energy spread out, whereas the energy of the correlated image is packed into the low frequency region (i.e., top left region). We classify image into number of types such as image contain large areas of slowly varying intensities. These images can be classified as low frequency images with low spatial details. A DCT operation on these images provides very good energy compaction in the low frequency region of the transformed image. Second type of image contains a number of edges (i.e., sharp intensity variations) and therefore can be classified as a high frequency image with low spatial content. However, the image data exhibits high correlation which is exploited by the DCT algorithm to provide good energy compaction. Third type of image with progressively high frequency and spatial content. Consequently, the transform coefficients are spread over low and high frequencies. The last type of image with periodicity shown therefore the DCT contains impulses with amplitudes proportional to the weight of a particular frequency in the original waveform.

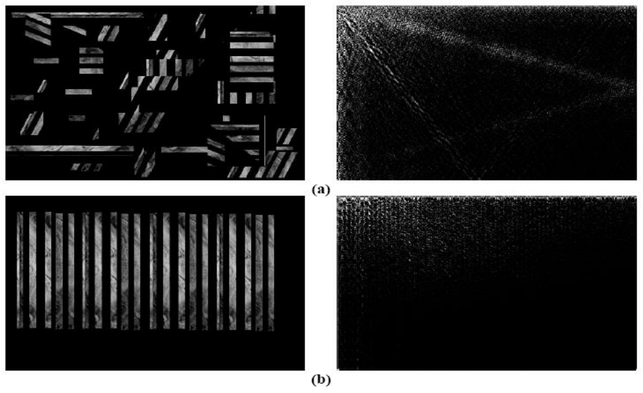


Figure ٢-١٠: (a) Uncorrelated image and its DCT; (b) Correlated image and its DCT.

Hence, it can be inferred that DCT renders excellent energy compaction for correlated images. Studies have shown that the energy compaction performance of DCT approaches optimality as image correlation approaches one i.e., DCT provides (almost) optimal decorrelation for such images [٤١].

c- Separability

The DCT transform equation **(٢.١٤)** can be expressed as[٣٦],

$$C(u,v) = \alpha(u).\alpha(v)\sum_{x=0}^{N-1} Cos\left[\frac{\Pi(2x+1)u}{2N}\right].\sum_{y=0}^{N-1} f(x,y).\sum_{x=0}^{N-1} Cos\left[\frac{\Pi(2y+1)v}{2N}\right] \quad \textbf{...(٢.١٦)}$$

For *u* ,*v* = ٠ , ١ , ٢ ,.. , *n*-١. This property, known as *separability*, has the principle advantage that *C*(*u*, *v*) can be computed in two steps by successive ١-D operations on rows and columns of an image. This idea is graphically illustrated in Figure (٢-١١) . The arguments presented can be identically applied for the inverse DCT computation (٢.١٥).

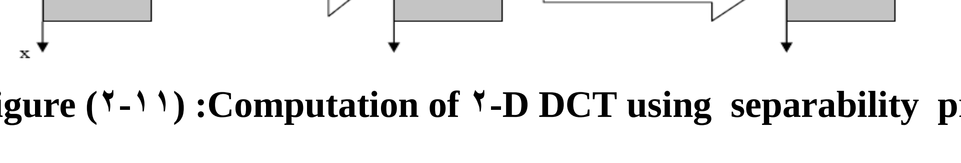


Figure (٢-١١) :Computation of ٢-D DCT using separability property.

d- Symmetry

Another look at the row and column operations in Equation **(٢.١٦)** reveals that these operations are functionally identical. Such a transformation is called a *symmetric transformation*. A separable and symmetric transform can be expressed in the form [٤٢]

$$\boldsymbol{T = AfA} \qquad \ldots(٢.١٧)$$

where A is an $N \times N$ symmetric transformation matrix with entries $a\ (i\ ,\ j)$ given by:

$$a(i,j) = \alpha(j)\sum_{j=0}^{N-1} Cos\left[\frac{\Pi(2j+1)i}{2N}\right]$$

and f is the $N*N$ image matrix.

This is an extremely useful property since it implies that the transformation matrix can be pre computed offline and then applied to the image thereby providing orders of magnitude improvement in computation efficiency.

e- Orthogonality

In order to extend ideas presented in the preceding section, let us denote the inverse transformation of **(٢.١٧)** as

$$f = A^{-1}TA^{-1}\,. \qquad \ldots(٢.١٨)$$

As discussed previously, DCT basis functions are orthogonal Thus, the inverse transformation matrix of A is equal to its transpose i.e A^{-1} = AT. Therefore, and in addition to its decorrelation characteristics, this property renders some reduction in the pre-computation complexity.

٢.٥ Quantization Process

Is the process of reducing the number of possible value of quantity, thereby reducing the number of bit needed to represent it. Therefore there is a trade – off between image quality and degree of quantization. A large quantization step can produce unacceptably large image distortion. This effect is similar to quantization Fourier series coefficients to coarsely, large distortion would result .

Unfortunately, finer quantization leads to lower compression ratio, therefore how can quantize the DCT coefficient most efficiently. Because of human eye's light natural high frequency roll – off, these frequencies play less important role than low frequencies [٤٣].

After each ٨*٨ matrix of DCT coefficients *c (u ,v)* is calculated, it is quantized, this step where information is loss (except for some unavoidable loss because of finite precision in other steps) occurs [١٤]. The quatizer divides the DCT coefficient by its corresponding quantum. The quantization matrix is ٨*٨ matrix of step sizes called quantums and then rounds to the nearest integer. Large quantum's drive small coefficient down to zero [٤٣]. the type of quantum matrix which are used by different compression algorithm is :

١. Default Quantiztion Tables :

One such table, for luminance (gray scale) component, is the result of many experiments preformed by the JEBG committee as shown in table (٢-٢) .

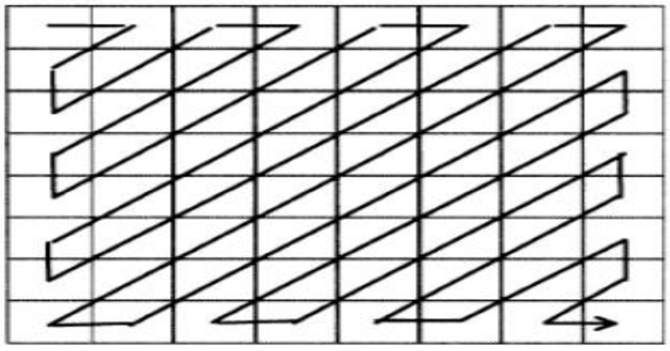
Table (٢-٢) – standard table of JEBG

١٦	١١	١٠	١٦	٢٤	٤٠	٥١	٦١
١٢	١٢	١٤	١٩	٢٦	٥٨	٦٠	٥٥
١٤	١٣	١٦	٢٤	٤٠	٥١	٦٩	٥٦
١٤	١٧	٢٢	٢٩	٥١	٨٧	٨٠	٦٢
١٨	٢٢	٣٧	٥٦	٦٨	١٠٩	١٠٣	٧٧
٢٤	٣٥	٥٥	٦٤	٨١	١٠٤	١٣	٩٢
٤٢	٦٤	٧٨	٨٧	١٠٣	١٢١	١٢٠	١٠١
٧٢	٩٢	٩٥	٩٨	١١٢	١٠٠	١٠٣	٩٩

٢. A simple quantization table Q

This is a table computed, based on one operate parameter R supplied by the user. A simple expression such as Qij = ١+(i+j)*R guarantees that QCs start small at the upper – left – corner and get bigger toward the bottom right corner [١٤] .

After quantization, it is not unusual for more than half of the DCT coefficients to equal zero, thus we can use zigzag type scan pattern shown in figure (٢-١٢) which re-order the coefficient so that gathering the zero coefficient one often another ,which is useful in lossless compression methods .



Figure(٢-١٢) ZIGZAG –SCAN TYPE

Chapter Three

Data Compression and Video Compression Techniques

٣.١ Introduction

There are many type of codec-coder /decoder which are used to describe any thing which turns data into another form for storage and transmission, then change it back for use - used in video compression.
In traditional broad casting, a codec is physical device which turn analog video and audio data into digital form to be sent out the air, it is also capable of turning received digital information back into analog format.
in computer a codec is used away of compressing video, images, and audio to more manageable size. The majority of codecs use a lossy method of compression, but some are lossless . Lossless codec such as MSU or HUFFyuv, reproduce the original video exactly with no subsequent loss if the video is re- encoded. The more common lossy codecs lose varying degrees of information, but can save substantial amounts of space [٤٤] .
The term video refer to the technology of processing electronic signals representing moving picture. A major application of video technology is television, but is also widely used in engineering, scientific, manufacturing, and security applications, the word video come from Latin " I see" [٤٥].
In this chapter we introduce to different compression method used in lossless and lossy image compression, in addition to video compression standard methods.

٣.٢ Lossless Compression

In which original image can be recreated exactly from the compressed data, so that called lossless because no data lost. For example image in these method are limited to compressing the image file to about one half to one third of its original size (٢:١ or ٣:١) , often the achievable compression is much less. For simple images such as text only image, lossless method may achieve much higher compression [١٢] .

٣.٢.١ Huffman Coding

A Huffman encoder takes a block of input characters with *fixed* length and produces a block of output bits of *variable* length. It is a fixed-to-variable length code

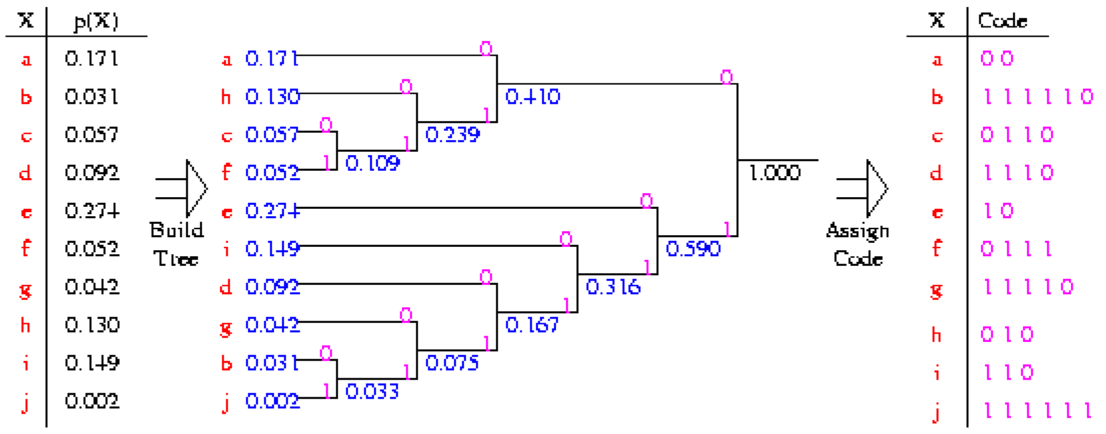
Huffman coding, uses a table of bit sequences which maps a variable length bit sequence to a particular token. For Huffman coding to compress data, shorter variable length codes must be used for tokens with the highest probability of occurring in a particular bit-stream. For this reason, Huffman coding usually requires the bit-stream to be pre-scanned to determine the relative probabilities of tokens in the bit-stream. Alternatively, several samples may be scanned to develop an average histogram of the probability of certain tokens occurring in a bit stream. Arithmetic coding uses a similar principle to Huffman coding, but does not require a lookup table for encoding or decoding [٤٦].

A Huffman code is designed by merging together the two *least probable* characters, and repeating this process until there is only one character remaining. A code tree is thus generated and the Huffman code is obtained from the labeling of the code tree. An example of how this is done is shown below in table (٣-١) [٤٧].

Table (٣-١) Example of Huffman input with probabilities

X	A	B	C	D	E	F	G	H	I	j
P(x)	٠.١٧١	٠.٠٣١	٠.٠٥٧	٠.٠٩٢	٠.٢٧٤	٠.٠٥٢	٠.٠٤٢	٠.١٣٠	٠.١٤٩	٠.٠٠٢

The final static code tree is given below in figure ٣-١:



X	p(X)
a	0.171
b	0.031
c	0.057
d	0.092
e	0.274
f	0.052
g	0.042
h	0.130
i	0.149
j	0.002

X	Code
a	0 0
b	1 1 1 1 1 0
c	0 1 1 0
d	1 1 1 0
e	1 0
f	0 1 1 1
g	1 1 1 1 0
h	0 1 0
i	1 1 0
j	1 1 1 1 1 1

Figure (٣-١) Example of Huffman with its code

- It does not matter how the characters are arranged. I have arranged it above so that the final code tree looks nice and neat.
- It does not matter how the final code tree are labeled (with ٠s and ١s). I chose to label the upper branches with ٠s and the lower branches with ١s.
- There may be cases where there is a *tie* for the two least probable characters. In such cases, any tie-breaking procedure is acceptable.
- Huffman codes are not unique.
- Huffman codes are optimal in the sense that no other lossless fixed-to-variable length code has a lower average rate.

٣.٢.٢ Run Length Encoding RLE

Huffman coding is poorly suited for image data. Although you could get general probabilities for various pixel intensities by sampling a variety of images, the compression rate would not be very high and the algorithm would not be as effective for all images. However, there are generalizations one can make about images. Images often have long runs of the same color due to objects having a consistent color. Images also have runs where the color is inconsistent. These runs usually occur at edges. Run length encoding (RLE) takes advantage of this by encoding pixels as runs [٤٧]. One of the simplest methods for compressing data is known as Run-Length Encoding (RLE).This method takes advantage of repeating data by encoding it as a tuplet consisting of the number of repetitions and the repeating symbol or number. Run-length encoding used exclusively generally only performs well on highly correlated data [٤٦].

The First byte sent indicates the type of run (one bit) and the length of the run. In the event of a run of a single color the following bytes record the color. Otherwise the following bytes are the colors of the pixels in the run. Typically the color channels are treated separately because the color channels vary independently. This technique can be taken a step further to a bit oriented run length encoding where each bit plane of each color is treated separately. Although the compression ratios will vary from image to image, you can expect a compression ratio of ٢:١ for byte oriented run length encoding .

٣.٢.٣ Arithmetic Coding

The main steps of arithmetic coding are as follows[٤٨] :-

*. start by defining the “current interval” as [٠,١) .

*. repeat the following two steps for each symbol S in the input stream :

. divide the current interval into subintervals whose size are proportional to the symbols probabilities .

. select subinterval for S and define it as the new current interval.

*. when the entire input stream has been processed in this way, the output should be any number uniquely identifies the current interval.

The symbols and the probabilities are written on the output stream before any of the bits of the compressed code, and this will be the first thing input by the decoder. the encoding process starts by defining two variables LOW and HIGH which define an interval [LOW , HIGH) as follow :-

LOW = LOW +RANGE* LOWRANGE (X)

HIGH = LOW +RANGE*HIGHRANGE (X)

Where RANGE= HIGH- LOW, LOWRANGE (X) and HIGHRANGE (X) indicate the low and high limits of the range of symbol X, respectively. The method reads the input stream symbol by symbol and appends more bits to the code each time a symbol is input and processed .

An example of how this is done is shown below.

Given three symbols a١,a٢,a٣ with probabilities p١=٠.٤,p٢=٠.٥ , and p٣=٠.١, respectively. The interval [٠,١) is divided among the three symbols by assigning each subinterval proportional in size to its probability. The three symbols are assigned the subintervals [٠,٠.٤), [٠.٤,٠.٩), and [٠.٩,١). To encode the string "a٢, a٢, a٢ ,a٣" we start with the interval [٠,١).

The first symbol a٢ reduce [٠,١) to [٠.٤,٠.٩) as follow :

LOW = ٠+(١-٠)*٠.٤ HIGH =٠+(١-٠) *٠.٩

The second symbol a٢ reduces [٠.٤,٠.٩) to [٠.٦,٠.٨٥) as follows:

LOW = ٠.٤+(٠.٩-٠.٤)*٠.٤ =٠.٦ HIGH =٠.٤+(٠.٩-٠.٤) *٠.٩=٠.٨٥

The third symbol a٢ reduces [٠.٦,٠.٨٥) to[٠.٧,٠.٨٢٥) as follows :

LOW = ٠.٦+(٠.٨٥-٠.٦)*٠.٤ =٠.٧ HIGH =٠.٦+(٠.٨٥-٠.٦) *٠.٩=٠.٨٢٥

The fourth symbol a٣ reduces [٠.٧,٠.٨٢٥) to [٠.٨١٢٥,٠.٨٢٥٠) , and this is the final code as follows :

LOW= ٠.٧+(٠.٨٢٥-٠.٧) ٠.٩ =٠.٨١٢٥ HIGH =٠.٧+(٠.٨٢٥-٠.٧) *١=٠.٨٢٥

٣.٣ Lossy Compression

In order to achieve compression ratios with complex images and video sequence, lossy compression methods are required. as we said previously lossy compression provides tradeoffs between image quality and degree of compression, which allows the compression algorithm to be customized to the application. with some of more advanced methods, images can be compressed with ١٠ to ٢٠ times with virtually no visible information loss and ٣٠ to ٥٠ times with minimal degradation, image enhancement and restoration techniques can be combined with lossy compression schemes to improve appearance of the decompressed image .

Lossy compression is performed in both spatial and transform domains such as gray-level run –length coding, differential predictive coding, and block truncation coding and vector quantization in spatial domain, transform coding in transform domain .

٣.٣.١ Gray –level Run Length Coding RLC

The Run Length Coding can also be used for lossy image compression by reducing the number of gray levels value and then applying RLC techniques. As with lossless techniques , for for this method to be effective, the reduced image data (in natural code)needed to be mapped to a gray code, where adjacent numbers differs only in one bit.

A more sophisticated RLC algorithm for encoding gray-level images is called the dynamic window-based RLC. This algorithm relaxes the criterion of the runs being the same value and allows for the runs to fall within gray-level range, called the dynamic window range. This range is dynamic because it starts out larger than actual gray-level window range, and maximum and minimum values are narrowed down to the actual range as each pixel value is encountered. This process continues until a pixel is found out of the actual range. The image is encoded with two values one for the run length and one to approximate the gray-level value of the run. This approximation can simply be the average of all the gray-level values in the run or a more complex method may be used to calculate the representative value [٤٠] .

An example of how this is done is shown below.

Assume the following pixels in sequence ٦٥ ٦٧ ٦٦ ٦٤ ٦٣ ٨٦ ٧٠ and window range of ٥

The first value is called the reference value(in this case ٦٥).

A dynamic window range is then defined which,

MINIMUM = reference- (window length-١) and,

MAXIMUM = reference+ (window length-١) and,

In this case dynamic window is [٦٥ - (٥-١)] to [٦٥+(٥-١)]= ٦١ to ٦٩ .

The next value encountered ٦٧ is used to adjust this range. The range based on this value alone is from ٦٣ to ٧١.

The new dynamic range is based on the intersection of the range from this new value with the previous range, so the new range is ٦٣ to ٦٩.

This process continues until the of ٨٦ is countered. At this point the range has been narrowed down to ٦٣ to ٦٧ so ٨٦ is out of range.Run Length= ٥ , Gray-level =(٦٥+٦٧+٦٦+٦٤+٦٣)/٥=٦٥ .

٣.٣.٢ Vector Quantization

For many years, vector quantization has been remain an important processing stage for lossy image compression, which is typified by using only subset of original data to approximate the input image data.

This make the technique ideal to break down the content of an image for the creation of an efficient indexing key. However, several problem exist with these techniques. Heuristic search methods, which attempt to perform a lengthy search of all data points to determine the closest ones.ctor quantization is the process of mapping a vector that can have many values to a vector that have a smaller(quantized) number of values. for image compression, the vector corresponds to a small sub-image, or block .

Vector quantized treats the entire sub-image (vector) as a single entity and quantizes it by reducing the total number of bits required to represent sub-image this is done by utilizing a codebook, which stores a fixed set of vectors, and then coding the sub-image by using the index (address) into the code book, this provides ١٦:١ compression, but note that this assumes that the codebook is not stored with the compressed file. However, the codebook will need to be stored unless a generic codebook that could be used for a particular type of image is devised, then we need only store the name of that particular codebook file .In the general case, better results will be obtained with a codebook that is designed for a particular image . In this case, including the codebook cut the compression in a half, from ١٦:١ to ٨:١ .

Now, how do we decide which vectors will be stored in the codebook ? This is typically done by a training algorithm that finds a set of vectors that best represents the blocks in the image[٤٩] .

This set of vectors is determined by optimizing some error criteria.

٣.٣.٣ Predictive Coding

Predictive coder directly utilize the fact that picture elements in local regions are highly correlated with one another. Predictive coding is an image compression technique, which uses a compact model of an image to predict pixel values of an image based on the value of neighboring pixel. A model of an image is a function model (x, y) , which computes (predict) the pixel value at coordinate (x, y)of an image , given the values of some neighboring of pixel (x, y), where neighbors are pixels whose values are known. Typically, when processing an image in raster scan order neighbors are selected from the pixels above and to left of the current pixel for example, common set of neighbors used for predictive coding is the set {(x-١ ,y-١),(x,y-١),(x+١,y-١),(x-١,y)}. Linear predictive coding is a simple, special case of predictive coding in which takes the average of the neighboring values .There are two expected sources of compression in the predictive coding based image compression. First, the error signal for each should have a smaller magnitude than the corresponding pixel in the original image (therefore they are requiring fewer bits to transmit the error signal).

Second, the error signal should have less entropy than the original.To complete the compression, the error signal is compressed by using an entropy coding algorithm such as Huffman coding or arithmetic coding [٤٠] .

٣.٣.٤ Transform Coding

Is a form of block coding done in the transform domain the image divided into blocks, or sub-images, and the transform is calculated for each block .Any of the previously defined transforms can be used, frequency (e.g.Fourier), or sequency (e.g. Walsh), but it has been determined that DCT transform is optimal for images. After transform has been calculated, the transform coefficients are quantized and coded . The primary reason this method is effective because the frequency / sequency transform of images efficiently puts most of information into relatively few coefficients so that many of high- frequency coefficients can be quantized to ٠. The simplest form of transform coding is achieved by filtering, there are several type of transform such as Discrete Fourier Transform, Discrete Cosine Transform which explain in details in chapter two, Discrete Wavelet Transform. In Discrete Wavelet Transform DWT the idea of wavelet theory is same to Fourier idea-that the signal such as image can be decomposed into linear combination of set of basis functions in wavelet theory the basis functions are function of two variable, translation and scale instead of one variable, frequency as in Fourier, resulting in a transformation to representation with varying degrees of frequency (scale) resolution and time (translation) resolution. The wavelet basis functions $\psi_{a,b}$ are called translated and scaled versions of function ψ called the mother wavelet function. That is

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi(\frac{t-b}{a}) \qquad ...(٣.١)$$

where a is the scaling parameter and b is the translating parameter ,where $\frac{1}{\sqrt{a}}$ is normalization of the energy of all wavelets.

This transform is used as alternative to DCT transform to provide decompressed image with out artifacts [٦].

٣.٤ Compression Video Techniques

There are several Video compression methods such as MJPEG, MJPEG٢٠٠٠, H.٢٦x and MPEG group (١,٢,٤) that explain in following sections, but we must know the basic principles of image compression firstly.

٣.٤.١ Principles of Image Compression

A typical lossy image compression scheme is shown in Figure (٣-٢) The system consists of three main components, namely, the source encoder, the quantizer, and the entropy encoder. The input signal (image) has a lot of redundancies that needs to be removed to achieve compression. These redundancies are not obvious in the time domain. Therefore, some kind of transform such as discrete cosine, Fourier, or wavelet transform is applied to the input signal to bring the signal to the spectral domain.

The spectral domain output from the transformer is quantized using some quantizing scheme. The signal then undergoes entropy encoding to generate the compressed signal [٥٠].

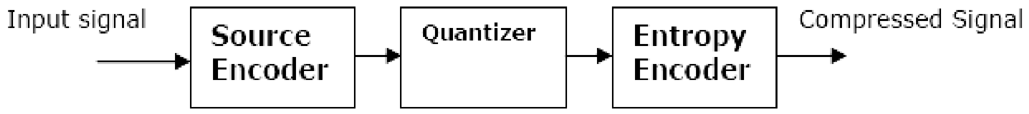


Figure (٣-٢) explain Principles of Image Compression

Source Encoder An encoder is the first major component of image compression system. A variety of linear transforms are available such as Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT), and Discrete Wavelet Transform (DWT).

Quantizer A quantizer reduces the precision of the values generated from the encoder and therefore reduces the number of bits required to save the transform co-coefficients. This process is lossy and quantization can be performed on each individual coefficient. This is known as Scalar Quantization (SQ). If it is performed on a group of coefficients together then it is called Vector Quantization (VQ).

Entropy Encoder An entropy encoder does further compression on the quantized values. This is done to achieve even better overall compression. The various commonly used entropy encoders are the Huffman encoder, arithmetic encoder, and simple run-length encoder. For improved performance with the compression technique, it’s important to have the best of all the three components.

Depend on this principles of image compression, there are two commonly known algorithm are called JPEG which is used DCT transform coding in source coding in figure (٣-٢) while JPEG٢٠٠٠ which is used DWT transform coding in it .

In JPEG & JPEG٢٠٠٠, Joint Photographic Experts Group , was tasked by the International Telecommunication Union (ITU) and International organization for Standardization (ISO) to develop of standard for the digital compression and coding of continuous-tone still image, therefore JPEG is suited to the compression of images that have been sampled from a more or less continuous function, and these are typically images of naturals and photographic nature, but not suited to the compression of schematics, line drawings, cartoons, diagrams, or any other kind of image consisting largely of sharply contrasted edge and large plain areas with little texture [١١][١] .

٣.٤.٢ Motion JPEG(M- JPEG)

A digital video sequence can be represented as a series of JPEG pictures, the advantages are the same as with single still JPEG-Picture flexibility both in terms of quality and compression ratio. The main disadvantage of M-JPEG is that since only series of still picture it make no use of video compression techniques. The result is slightly lower compression ratio for video sequence compared to “real” video compression techniques [٥١].

٣.٤.٣ Motion JPEG٢٠٠٠(M- JPEG٢٠٠٠)

It is also can be used to represent a video sequence. The advantages is that the blockiness of M-JPEG is removed, but replaced with a more over all fuzzy picture, whether this fuzziness of M-JPEG٢٠٠٠ is preferred compared to the “ blockiness” of M-JPEG is a matter of personal preference, also it is better compression ratio compared to M-JPEG, but at the price complexity. The disadvantage of its does not take any advantage of video sequence compression this result lower compression ratio compared to real video compression techniques [٥١].In these two method M-JPEG & M-JPEG٢٠٠٠ compress each frame of video sequence using JPEG & JPEG٢٠٠٠ continuous –tone still image compression standard as shown in figure (٣-٣). As such its an intra-frame compression scheme, it is use each frame as still image convert it from RGB to digital YCbCr, then sub-sampled by using ٤:٢:٢ and source input format of image SIF (٣٥٢*٢٤٠), then used the same principles of compressed image in section ٣.٣.١ [٥٢] .

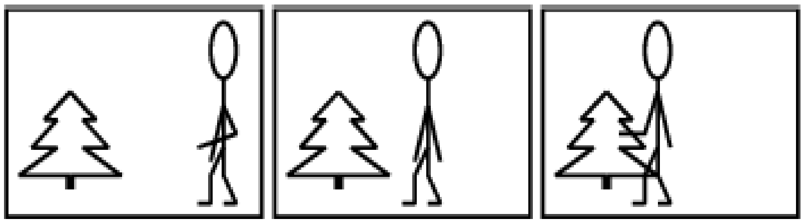


figure (٣-٣) Example of a sequence of three complete JPEG images

٣.٤.٤ H.٢٦x

Video conferencing and video telephony are the intended applications for H.٢٦١, H.٢٦٣ , and H.٢٦٤ by ITU , it is also designed for develop primarily for low bit – rate Integrated Services Digital Networks (ISDN) services, to allow for low- cost implementations , H.٢٦١ fixes many of the system parameters , only YUV color component separation with ٤:٢:٠ sampling ratio is allowed by the standard [٥٢]. In addition allow for only two frame size CIF(٣٥٢*٢٨٨) & QCIF(١٧٦*١٤٤) [٥٣] . All these techniques based on motion compensation ,DCT transform for video coding standard ,therefore two type of frames are defined ; key frame that are coded independently and non- key frames that coded with respect to previous frame , i.e. key frame coded as intra- frame , while non-key frame as inter-frame [٥٢].The data rate of coding algorithm was designed to be able to set between ٤٠kbits/s and ٢mbits/s [٥٤].

٣.٤.٥ Video Compression – MPEG

One of the best-known audio and video streaming techniques is the standard called MPEG (initiated by the *Motion Picture Experts Group* in the late ١٩٨٠s).These algorithm developed from H.٢٦x and JPEG algorithms it use YUV color component for PAL & YQI for SECAM color component separation with ٤:٢:٠ sampling ratio is allowed by the MPEG standard, it accept any frame size such as SIF, CIF, QCIF, and Sub-QCIF also these algorithm based on motion compensation, DCT transform for video coding standard [٥٢].Sequence of video frames are compressed by using three type of image I-, B-, and P-frames may look as below in figure (٣-٤) , which are defined as[٥٥]:

*. Intra –picture Coding

Intra –picture, or I –picture are coded using only information present in the picture itself like JPEG.

*. Predictive –picture Coding

Predicted pictures, P –pictures are coded with respect to the nearest previous I- or P- picture, this technique is called forward prediction .

*. Bidirectional –picture Coding

Bidirectional –picture, or B-picture that use both a past and future picture as a reference, this technique is called Bidirectional prediction .

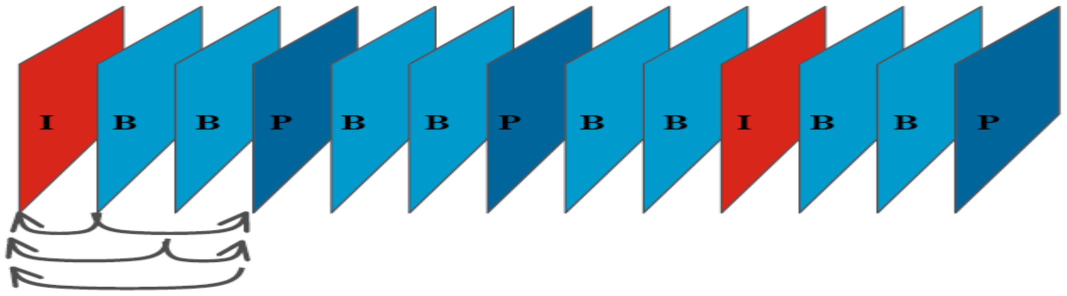


Figure (٣-٤) sequence of frames compressed in three type with MPEG

To compare MJPEG with MPEG the second algorithm is better than MJPEG because of the result of applying MPEG video compression is that the amount of data transmitted across the network is less than that of Motion JPEG. This is illustrated below where only information about the differences in the second and third frames is transmitted as illustrated in figure (٣-٥) i.e .only the motion is compressed [٥٢] .

Figure (٣-٥) illustrate differences between successive frame

The MPEG algorithm can be classified into three types.

٣.٤.٥.١ MPEG-١

The MPEG-١ standard was released in ١٩٩٣ with the target application of storing digital video onto CDs. Therefore, most MPEG-١ encoders and decoders are designed for a target bit-rate of about ١.٥Mbit/s at CIF resolution. For MPEG-١, the focus is on keeping the bit-rate relatively constant at the expense of a varying image quality, typically comparable to VHS video quality. The frame rate in MPEG-١ is locked at ٢٥ (PAL)/٣٠ (NTSC) fps. [٥٣]

٣.٤.٥.٢ MPEG-٢

MPEG-٢ was approved in ١٩٩٤ as a standard and was designed for High Quality Digital Video (DVD), Digital High-Definition TV (HDTV), Interactive Storage Media (ISM), Digital Broadcast Video (DBV), and CAble TV (CATV). The MPEG-٢ project focused on extending the MPEG-١ compression technique to cover larger pictures and higher quality at the expense of a lower compression ratio and higher bit-rate. MPEG-٢ also provides additional tools to enhance the video quality at the same bit-rate; thus producing very high image quality when compared to other compression technologies. The frame rate is locked at ٢٥ (PAL)/٣٠ (NTSC) fps, just as in MPEG-١ [٥٤].

٣.٤.٥.٣ MPEG-٤

The MPEG-٤ standard was approved in ٢٠٠٠ and is a major development from MPEG-٢. In this section we'll take a brief look MPEG-٤ to fast understand terms and aspects such as:

١• MPEG-٤ profiles

٢• MPEG-٤ short header and MPEG-٤ long header

٣• MPEG-٤ and MPEG-٤ AVC

٤•MPEG-٤ Constant Bit-Rate (CBR) and MPEG-٤ Variable Bit-Rate (VBR)

٥• Positioning MPEG-١, MPEG-٢, and MPEG-٤

As a major development from MPEG-٢, there are many more tools in MPEG-٤ to lower the bit-rate needed to accomplish a certain image quality for a certain application or image scene.

Furthermore, the frame rate is not locked at ٢٥/٣٠ frames per second.

It's important to note however, that most of the tools used to lower the bit-rate are relevant today only for non-real time applications. This is because some of the new tools need so much processing power that the total time for encoding and decoding (i.e. the latency) makes them impractical for applications other than studio movie encoding, animated movie encoding, and the like. In fact, most of the tools in MPEG-٤ that can be used in a real time application are the same tools that are available in MPEG-١ and MPEG-٢.

Another enhancement with MPEG-٤ is the larger number of profiles and profile levels (explained below) that cover a wider range of applications, everything from low bandwidth streaming to mobile devices, to applications with extremely high quality and almost unlimited bandwidth demands. The making of studio movies is one such example[٥٦].

١. *MPEG-٤ Profiles*

Because of the large number of techniques (tools) available in MPEG (especially MPEG-٤) to reduce the bit-rate, the varying complexity of these tools, and the fact that not all tools are applicable to all applications, it would have been unrealistic and unnecessary to specify that all MPEG encoders and decoders should support all available tools. Therefore subsets of these tools for different image formats and target bit-rates have been defined[٥٣].
There are a number of different subsets defined for each of the MPEG versions. Such a subset of tools is called an ***MPEG Profile***. A specific MPEG Profile specifies exactly which tools the MPEG decoder shall support. In fact the requirements are in the decoder and the encoder does not have to make use of all available tools.
Furthermore, each profile exists at different *Levels*. The Level specifies parameters such as maximum bit-rate and supported resolutions. By specifying the MPEG Profile and Level, it's possible to design a system that only uses the tools in MPEG that are applicable to the target application.
MPEG-٤ has a number of different profiles. Among them Simple Profile and Advanced Simple Profile are the most commonly used in security applications. While many tools are used by both of these profiles, there are some differences. For example, Simple Profile supports I-, and P- (frames), while I-, B-, and P- (frames) need to be supported by Advanced Simple Profile. Another difference between Simple Profile and Advanced Simple Profile is the supported range of resolutions and bit-rates, denoted by the Level. While Simple Profile goes up to CIF resolution and ٣٨٤ kbit/s (at the L٣ level) , Advanced Simple Profile goes up to ٤CIF resolution and ٨٠٠٠ kbit/s (at the L٥ level).

٢. *MPEG-٤ short header and long header*

Some network video streaming systems specify support for “MPEG-٤ short header,” so it’s important to understand this term. In fact it’s nothing more than an H.٢٦٣ video stream encapsulated with MPEG-٤ video stream headers.
MPEG-٤ short header does not take advantage of any of the additional tools specified in the MPEG-٤ standard. MPEG-٤ short header is only specified to allow backwards compatibility with the H.٢٦٣ recommendation for video conferencing over ISDN and LAN. For practical purposes, MPEG-٤ short header is identical to H.٢٦٣ encoding/decoding, which gives a lower quality level than both MPEG-٢ and MPEG-٤ at a given bit-rate.
The image and video quality in “short header” is not close to that of true MPEG-٤, since it does not make use of techniques that allow it to filter out picture information that is not visible to the human eye.To clarify a network video streaming system specification, support for MPEG-٤ is sometimes denoted “MPEG-٤ long header,” which in other words is the mode where the MPEG-٤ compression tools are being used[٥٦].

٣.*MPEG-٤ part ١٠ (Advanced Video Control)*

MPEG-٤ AVC, also referred to as H.٢٦٤, is a further development in which MPEG has a completely new set of tools that add more advanced compression techniques to further reduce the bit rate at a given image quality. Being more complex also adds performance requirements & cost, especially for the encoder, to the network video streaming system[٥٣].

٤•*Constant bit-rate (CBR) and Variable bit-Rate (VBR)*

Another important aspect of MPEG is the bit-rate mode that is used. In most MPEG systems, it’s possible to select if the bit rate should run in CBR mode or VBR mode. The optimal selection depends on the application and available network infrastructure. With only limited bandwidth available, the preferred mode is normally CBR, since this mode generates a constant and predefined bit-rate. The disadvantage is that image quality will vary and while it will remain relatively high when there is no motion in the image scene, the quality will significantly decrease with increased motion.

The VBR mode on the other hand will maintain a high, if so defined, image quality regardless of motion or no motion in the image scene. This is often desirable in security and surveillance applications when there is a need for high quality, especially if there is motion in the picture. Since the bit-rate in VBR may vary, even though an average target bit-rate is defined, the network infrastructure (available bandwidth) for such a system needs to have this capacity[٥٢].

٥• Positioning MPEG-١, MPEG-٢, and MPEG-٤

The illustration below shows MPEG-٤'s much wider scope relative to MPEG-١ and MPEG-٢, which were developed for more specific applications. While MPEG-١ was developed for digital video on CD-ROM, MPEG-٢ was developed with DVD and high-definition television in mind. MPEG-٤ on the other hand does not have such specific target applications and may be appropriate from the studio down to cellular phone applications, so that the horizontal axis represent group of MPEG methods which are developed for different application which represent the quality of application in vertical axis [٥٦].

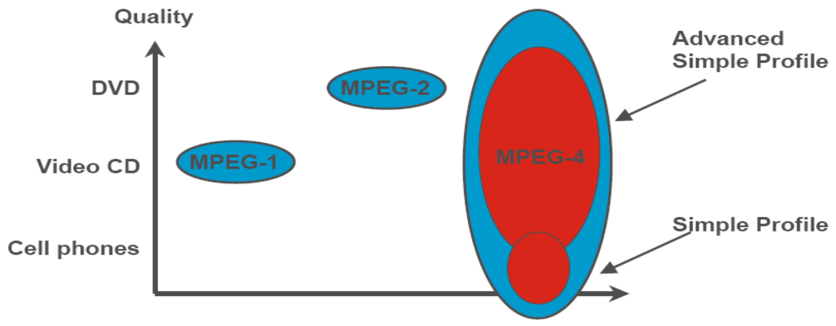


Figure ٣-٦ illustrate Positioning MPEG-١, MPEG-٢,and MPEG-٤

٣.٤.٥.٤ Advantages and Disadvantages for M-JPEG,

MPEG-٢ and MPEG-٤ Due to its simplicity, M-JPEG is a good choice for use in many applications. JPEG is a widely available standard in many systems often by default.

It’s a simple compression/decompression technique, which means the cost, in both system time and money, for encoding and decoding is kept low. The time aspect means that there is limited delay between image capturing in a camera, encoding, transfer over the network, decoding, and

finally display at the viewing station. In other words, M-JPEG provides low latency due to its simplicity (image compression and complete individual images), and for this reason it’s also well suited for when image processing is to be performed, for example video motion detection

or object tracking.

Any practical image resolution, from mobile phone display size up to full video (٤CIF) image size, is available in M-JPEG. It also gives a guaranteed image quality regardless of movement or complexity of the image scenes. It still offers the flexibility to select either high image quality (low compression) or lower image quality (high compression) with the benefit of lower image file sizes, thus lower bit-rate and bandwidth usage. At the same time the frame rate can be easily controlled, providing a means to limit bandwidth usage by reducing the frame rate, but still with a guaranteed image quality.

Since M-JPEG doesn’t make use of a video compression technique, it generates a relatively large amount of image data that is sent across the network. For this reason, at a given image compression level (defining the image quality of the I-frame and JPEG image respectively), frame rate, and image scene, the amount of data per time unit sent across the network (bit-rate) is less for MPEG compared to M-JPEG, except at low frame rates as described below.

This following neatly summarizes the benefit of MPEG: the ability to give a relatively high image quality at a lower bit-rate (bandwidth usage). This can be especially important if the available network bandwidth is limited, or if video is to be stored (recorded) at a high frame rate and there are storage space restraints. The lower bandwidth demands come at the cost of higher complexity in encoding and decoding, which in turn contributes to a higher latency when compared to M-JPEG. One other item to keep in mind: Both MPEG-٢ and MPEG-٤ are subject to licensing fees. The graph below shows in figure (٣-٧) in principle how bandwidth use between M-JPEG and MPEG-٤ compares at a given image scene with motion. As can be seen, at lower frame rates, where MPEG-٤ compression cannot make use of similarities between neighboring frames to a high degree, and due to the overhead generated by the MPEG-٤ streaming format, the bandwidth consumption is actually higher than M-JPEG[٥٦].

It is clear from the graph when the frame rate is low the MPEG-٤ can not use the similarities between neighboring frames therefore require higher bandwidth than the M-JPEG to same frame rate ,but when frame rate is high the MPEG-٤ does not require to higher bandwidth than the M-JPEG, because it use the similarities at higher degree as shown in graph.

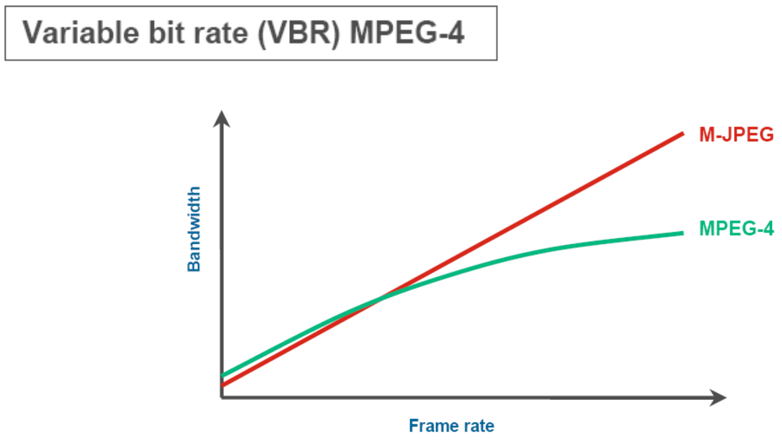


Figure (٣-٧) illustrate bandwidth use between M-JPEG and MPEG-٤

٣.٥ AVI Movie File

The Microsoft AVI file format is a RIFF(resource interchange file format) which is a document format file specification used with applications that capture, edit, and play back audio-video sequences. In general, AVI files contain multiple streams of different types of data. Most AVI sequences use both audio and video streams. A simple variation for an AVI sequence uses video data and does not require an audio stream.[٥٧][٥٨] .

AVI (Audio Video Interleave)

The Audio Video Interleave file format was originally developed by Microsoft for Intel. The AVI format is now marketed as “Video for Windows”. Due to the wide user base of Microsoft Windows, the AVI format has become very popular. Extension: .avi

AVI RIFF Form

AVI files use the AVI RIFF form. The AVI RIFF form is identified by the four-character code ``AVI '' – which A FOURCC (four-character code) is a ٣٢-bit unsigned integer created by concatenating four ASCII characters. The AVI file format uses FOURCC codes to identify stream types, data chunks, index entries, and other information-. All AVI files include two mandatory LIST chunks. These chunks define the format of the streams and stream data. AVI files might also include an index chunk. This optional chunk specifies the location of data chunks within the file. An AVI file with these components has the following form:

```
RIFF ('AVI '
    LIST ('hdrl' ... )
    LIST ('movi' ... )
    ['idx١' (<AVI Index>) ]
   )
```

The 'hdrl' list defines the format of the data and is the first required LIST chunk. The 'movi' list contains the data for the AVI sequence and is the second required LIST chunk. The 'idx١' list contains the index. AVI files must keep these three components in the proper sequence.

AVI Main Header

The 'hdrl' list begins with the main AVI header, which is contained in an 'avih' chunk. The main header contains global information for the entire AVI file, such as the number of streams within the file and the width and height of the AVI sequence. The main header chunk consists of the **AVIMAINHEADER** structure defines global information in an AVI file.

```
typedef struct _avimainheader {
   FOURCC fcc;
   DWORD  cb;
   DWORD  dwMicroSecPerFrame;
   DWORD  dwMaxBytesPerSec;
   DWORD  dwPaddingGranularity;
   DWORD  dwFlags;
   DWORD  dwTotalFrames;
   DWORD  dwInitialFrames;
   DWORD  dwStreams;
   DWORD  dwSuggestedBufferSize;
   DWORD  dwWidth;
   DWORD  dwHeight;
   DWORD  dwReserved[٤];
} AVIMAINHEADER;
```

Fcc: Specifies a FOURCC code. The value must be 'avih'.

Cb: Specifies the size of the structure, not including the initial ٨ bytes.

dwMicroSecPerFrame : Specifies the number of microseconds between frames. This value indicates the overall timing for the file.

dwMaxBytesPerSec : Specifies the approximate maximum data rate of the file. This value indicates the number of bytes per second the system must handle to present an AVI sequence as specified by the other parameters contained in the main header and stream header chunks.

dwPaddingGranularity: Specifies the alignment for data, in bytes. Pad the data to multiples of this value.

dwFlags: Contains a bitwise combination of zero or more of the flags
dwTotalFrames:Specifies the total number of frames of data in the file
dw InitialFrames :Specifies the initial frame for interleaved files. Non interleaved files should specify zero.
dwStreams : Specifies the number of streams in the file. For example, a file with audio and video has two streams.
dwSuggestedBufferSize:Specifies the suggested buffer size for reading the file.
dwWidth :Specifies the width of the AVI file in pixels.
dwHeight: Specifies the height of the AVI file in pixels.
dwReserved : Reserved. Set this array to zero.

AVI Stream Headers

One or more 'strl' lists follow the main header. A 'strl' list is required for each data stream. Each 'strl' list contains information about one stream in the file, and must contain a stream header chunk ('strh') and a stream format chunk ('strf'). In addition, a 'strl' list might contain a stream-header data chunk ('strd') and a stream name chunk ('strn').

The stream header chunk ('strh') consists of an AVI Stream Header structure .

AVIOLDINDEX Structure

The AVIOLDINDEX structure describes an AVI ١.٠ index ('idx١' format). New AVI files should use an AVI ٢.٠ index ('indx' format).syntax

```
typedef struct _avioldindex {
  FOURCC  fcc;
  DWORD   cb;
  struct _avioldindex_entry {
    DWORD   dwChunkId;
    DWORD   dwFlags;
    DWORD   dwOffset;
    DWORD   dwSize;
  } aIndex[];
} AVIOLDINDEX;
```

Members

fcc : Specifies a FOURCC code. The value must be 'idx١'.

Cb : Specifies the size of the structure, not including the initial ٨ bytes.

dwChunkId : Specifies a FOURCC that identifies a stream in the AVI file. The FOURCC must have the form 'xxyy' where *xx* is the stream number and *yy* is a two-character code that identifies the contents of the stream: such as db means Uncompressed video frame , dc means Compressed video frame , and wb means Audio data

dwFlags : Specifies a bitwise combination of zero or more of the flags:

dwOffset : Specifies the location of the data chunk in the file. The value should be specified as an offset, in bytes, from the start of the 'movi' list; however, in some AVI files it is given as an offset from the start of the file.

dwSize : Specifies the size of the data chunk, in bytes.

Chapter Four

Proposed System

٤.١ Introduction

In the previous chapters, we introduce a number of video compression methods and introduce the AVI data structure under which windows deals with, in this chapter we shall introduce to the proposed compression system which is shown in figure (٤-١).

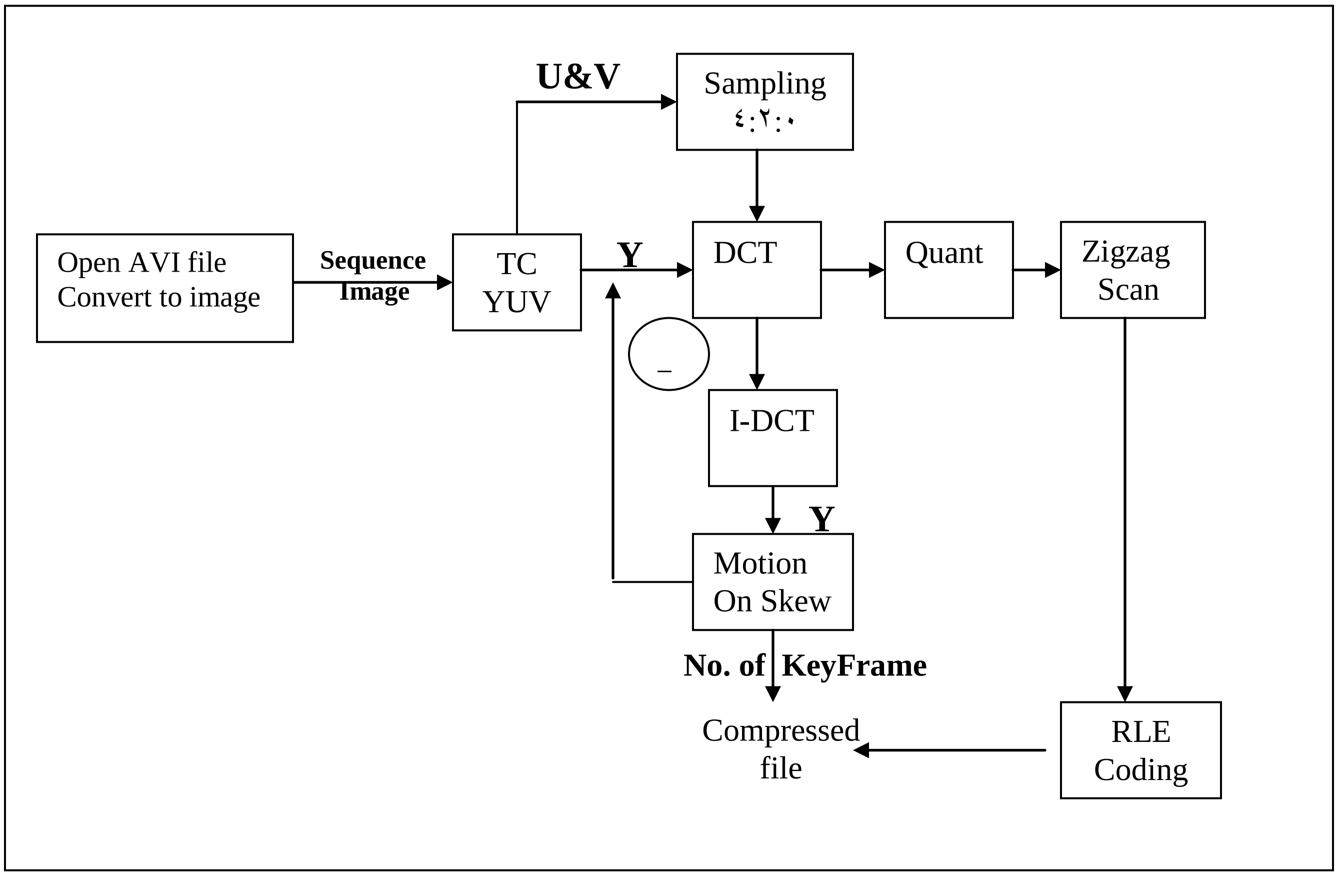


Figure (٤-١) block diagram of Coder in the proposed system

The proposed system consist of two parts : the first part is the Coder which contain several stages as shown in figure (٤-١) which represent the compression of movie file and the second part is Decoder which contain several stages as shown in figure (٤-٢), in the next sections we explain each part in details .

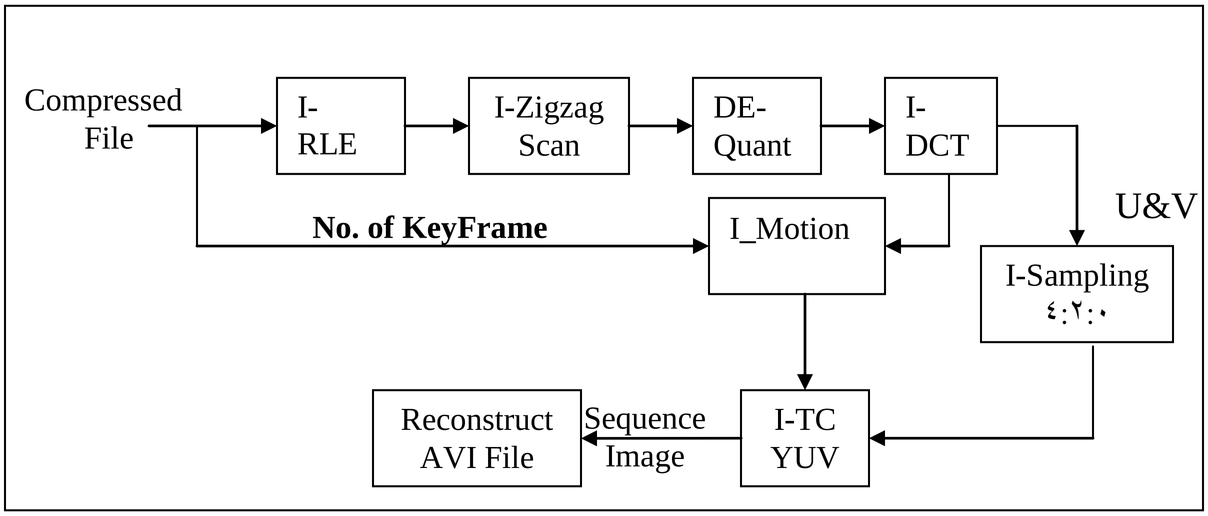


Figure (٤-٢) block diagram of Decoder in the proposed system

٤.٢ Coder Part

This part consist of several algorithms which are needed to compress video data into another form stored in compressed file. These algorithms will be explained as follows :

٤.٢.١ Open AVI File & Convert to Image

In this stage we proposed algorithm to open AVI structure and reach to list movie (i.e. sequence frames / images) – which are needed in the next stages to compress it – and we can summarize the open AVI file in few steps :

Algorithm ٤.١ Open AVI File & Convert To Image

```
Input :-  AVI Video File .
Output:- Sequence image stored in three matrices called Red (row, col), Green (row, col), and Blue (row, col) image.
step ١: call  convert function ASCII to long
step ٢: open riff .AVI file
step ٣: read AVI  header , read AVI index
step ٤: search for the Movie list
step ٥: for I = ١ to frames  number          Do
          search for initiate "٠٠db"
```

```
        for Row = height  downto ١        Do
         for Col = ١ to width         Do
            read RGB pixels with split it into three images RED, GRN ,and
            BLU as
            RED[Row , Col ] = RGB pixel .R
            GRN[Row , Col  ] = RGB pixel .G
            BLU[Row , Col ] = RGB pixel .B
STEP ٦: END .
```

In this algorithm the function (ASCII to long) was used to convert FCC ASCCII character to long for example convert FCC RIFF to long by calling ASCII to long (“R”,”I”,”F”,”F”) to open AVI file, then after that take information from AVI header such as total frame , height , and width of the image and then search for list movie using search function about key word “MOVI”, when we reach make a loop from one to sequence of image (frames) then search about keyword “٠٠db” which is the initial of each frame, then make two loops form height and from width respectively to read RGB pixel image, and split it into three bands that save it into three individual matrices called RED,GRN, and BLU which are used in the next stage called Transform Color .

٤.٢.٢ Transform Color YUV

The ‘Transform Color YUV’ perform algorithm to convert RGB pixels into Luminance / Chrominance components by using YUV European which convert Red into Luminance called Y and the value lies in [٠,٢٥٥] while convert Green and Blue to color information called Chrominance U and V respectively which is a bipolar value (i.e. positive & negative), the range for U is [-١١٢ , ١١٢] and V is [-١٥٧ , ١٥٧], this conversion is efficient in compression. The following algorithm describe the process of conversion :

Algorithm ٤.٢ Transform color YUV

Input:- Red (row, col), Green (row, col), and Blue (row, col) image matrices.
Output:- Y(row, col) luminance image, U (row, col) & V (row, col) chrominance image which carry color informations.

```
STEP ١: for I = ١ to frames number          Do
            for row = ١ to height            Do
             for col = ١ to width            Do
              Let R = RED[I , row , col ]
              Let G = GRN[I , row , col ]
              Let B = BLU[I , row , col ]
STEP ٢:    Y = ٠.٢٩٩ * R + ٠.٥٨٧ * G + ٠.١١٤ * B
             If Y < ٠ then
                Y=٠
              Else if Y>٢٥٥ then
                Y=٢٥٥
              End if
STEP ٣:    U = ٠.١٤٧ * R  -٠.٢٨٩ * G  + ٠.٤٣٦ * B
             If U < -١١٢ then
                U=-١١٢
              Else if U>١١٢ then
                U=١١٢
              End if
STEP ٤:    V = ٠.٦١٥ * R - ٠.٥١٥ * G - ٠.١٠٠*B
              If V < -١٥٧ then
                V=-١٥٧
              Else if V>١٥٧ then
                V=١٥٧
              End if
STEP ٥:     Let RED[I , row , col ] = Y
              Let GRN[I , row , col ] = U
              Let BLU[I , row , col ] = V
STEP ٦:END .
```

٤.٢.٣ Sub - sampling ٤:٢:٠

The sub- sampling take only U and V components (i.e. chrominance) which carries color information and apply the algorithm which reduce each color component to half by dividing by two in both direction height and width to reduce the size that take in compressed file

Algorithm ٤.٣ Sub - sampling ٤:٢:٠

Input:- Two color component U (row, col) & V (row, col)matrices.

```
Output:- Sampling U (row/٢, col/٢) & V (row/٢, col/٢) into half size.
STEP ١: Let DH = height / ٢ : DW = width / ٢
STEP ٢: for I = ١ to frames number  Do
            for row =  ١  to  DH          Do
             for col = ١  to  DW          Do
               Let P١ = GRN[I , row * ٢  , col * ٢   ]
               Let P٢ = GRN[I , row * ٢+١ , col * ٢   ]
               Let P٣ = GRN[I , row * ٢  , col * ٢+١  ]
               Let P٤ = GRN[I , row * ٢+١, col * ٢+١]
               Let P٥ =  BLU[I , row * ٢  , col * ٢   ]
               Let P٦ =  BLU[I , row * ٢+١ , col * ٢   ]
               Let  P٧ =  BLU[I , row * ٢  , col * ٢+١  ]
               Let P٨ =  BLU[I , row * ٢+١, col * ٢+١]
STEP ٣:      Let GRN١ =(P١+P٢+P٣+P٤) /٤
               Let BLU١ =(P٥+P٦+P٧+P٨) /٤
STEP ٤: END
```

٤.٢.٤ Motion Based on Skew

The motion based on skew take only Y luminance component of sequence frame and calculate the skew for each image (frame), then make absolute difference for successive frames and compare the result with some threshold -the threshold value will be indicated by the user in the proposed system, the threshold value is ٠.٠١- which is chosen to split sequence image into scenes based on skew and comparison, then in each scene copy the first frame as key frame and then make the difference between key frame and successive frame to reduce pixel value because the successive frame contain many identical information, after that stored the number of key frame of each scene in compressed file, we can explain motion algorithm in few steps :

Algorithm ٤.٤ motion based on skew

```
Input:-   Y (row, col) luminance image matrix .
Output:-  Differences of successive Y (row, col) image with key frame
number.
STEP ١: Let key frame (١) =١ : count of key frame  = ٢
STEP ٢: for I = ٢ to frames number          Do
```

```
            If  skew (I)  - skew(I-١) >= Threshold   then
             ١. key frame (count of key frame  ) = I
             ٢. increment count of key frame  by ١
            END if
          END for
STEP ٣ : store key frame in compressed file
STEP ٤:Copy Of Key Frame In Each Scene
          For nframs = ١ to count of key frame  Do
            For row = ١ to  height         Do
             For col =  ١ to width           Do
              Let MRED (key frame (nframs),row ,col) =  RED(key frame
              (nframs),row ,col)
             END for Col
           END for  Row
STEP ٥:Calculate Of The Differences
          For I = (key frame (nframs)+١) to (key frame (nframs+١))
           If  I <>  (key frame (nframs+١)) then
           For row = ١ to  height
            For col =  ١ to width
             Let MRED (I,row ,col) =  RED(I,row ,col)- RED(I-١,row ,col)
            END for Col
           END for  Row
           Else
           Goto  STEP ٧
          End if
STEP ٦: END For I
STEP ٧: END For nframes
STEP ٨: END.
```

٤.٢.٥ DCT Transformation & Quantization

The transform DCT apply algorithm to the three Y, U, and V images matrices to transform image data from spatial domain to frequency domain by dividing each frame into block of ٨*٨ pixels and apply DCT transform to each block ٨*٨ pixel value to produce ٦٤ coefficients. And Then apply algorithm of quantization to ٨*٨ block of DCT coefficients to increase the number of zero coefficient by increasing R factor which is used in (١+(row + col)*٢٠) to divide DCT coefficients and to avoid negative coefficients by using normalized operation by computing the maximum coefficient and minimum coefficient can be obtained by the DCT transform in block and then difference the minimum value from the maximum to produce positive value of coefficients.

٤.٢.٦ Zigzag Scan & RLE

The process of zigzag scan apply algorithm to convert quantized coefficients block ٨*٨ into one dimensional vector of ٦٤ coefficient scan in zigzag type which is useful to accumulate the coefficients that closely same together, and then run length encoding algorithm is applied to ٦٤ coefficients of one-dimensional read in zigzag scan type, to reduce the repetition of the coefficient value which are similar especially zero coefficients that produced by DCT transform and Quantization by counting the number of repetition of coefficients of the same value which represent the counter followed by coefficient value such as ٨٨ ٨٨ ٨٨ ٨٨ ٨٨ ٨٨ stored as (٦ ,٨٨), Lyr in RLE represent number of red, green or blue the RLE can be summarized in few steps:

Algorithm ٤.٥ RLE

```
Input:-   ١-D block of ٦٤ coefficients DCT .
Output:-  pairs of RLE (count ,value) of block coefficients.
STEP ١: let Curpos =٠ : Count =٠
STEP ٢:  DO
          Let cChar= ZRed[Curpos]
          call sub BYRUN (Curpos , ZRed[Curpos], run, ZRed[ ])
STEP ٣:if run = ١ then
            ١. Let temp (Count) =cChar :   ٢. increment Count by ١
           Else If Count <>٠ then
           Call sub write run ٢ (Temp , Count , Lyr Curpos)
            Let Count =٠
           End if
          Call sub write run ١ (cChar , run , Lyr Curpos)
         End if
STEP ٦: loop until Curpos >=٦٣
STEP ٧: PUT FRLE (Lyr , Curpos) in compressed file
STEP ٨:END.
```

The sub BYRUN used to count the repetition of coefficient value in each ٦٤ – coefficients of ١-D, while write run٢ used to store coefficient of different value in each ٦٤ – coefficients of ١-D and write run ١ used to store coefficient of same value in each ٦٤ – coefficients of ١-D .

٤.٣ Decoder Part

This Part consist of several algorithms which are needed to decompress video data from compressed file (i.e. inverse compression operation) as in figure ٤-٣ ,thus we explain the work of this part in details in following sections .

٤.٣.١ Inverse of RLE & Inverse of Zigzag Scan

The inverse run length coding algorithm read from the compressed file ,byte after byte ,the first byte represent count , if count not equal to ٦٤ then read data and decompress repetition by the following steps :

Algorithm ٤.٦ Inverse-RLE

```
Input:-   pair of RLE (RLEcount , value).
Output:- ١-D block of ٦٤ coefficients of DCT.
STEP ١: let Cur  =١ : Count =١
STEP ٢: read from compressed file  length
STEP ٣: DO while Cur <= length
           ١. read from compressed file  RLECount
           ٢. increment Cur by ١
STEP ٤:if RLECount > ٦٤ then
           ١.  RLECount=RLECount-١٩١
           ٢.   read from compressed file  cChar
           ٣.  increment Cur by ١
           ٤.for J = ١to RLECount
           ٥. FRLE (Lyr,Count) =cChar
           ٦.increment Count by ١
           ٧. End for
           Else
           ١.for J = ١to RLECount
           ٢. read from compressed file  cChar
```

```
            ٣.increment Cur by ١
            ٤.  FRLE (Lyr,Count) =cChar
            ٥.increment Count by ١
            ٦. End for
           End if
STEP  ٥:loop While
STEP ٦: END.
```

After that we apply the inverse zigzag scan take ٦٤ coefficients of ١-D from FRLE (Lyr, Count) produced by inverse RLE, and apply inverse zigzag algorithm to convert them from ١-D of ٦٤ coefficients to ٢-D of ٨*٨ block.

٤.٣.٢ De - Quant & Inverse of DCT

In this process which is the inverse of quantization take the ٢-D coefficients of block ٨*٨ from the inverse zigzag scan and multiply by factor R in equation (١+(row + col)*٢٠) to increase the value of coefficients to reach nearly from source value coefficient or same it that produced by DCT transform, since rounding in quantization process lead to that lose , also we can inverse the normalization to obtain the negative value of coefficient ,then apply process if Inverse of DCT ,in this process the quantized coefficients of block ٨*٨ are take as input to inverse DCT transform to produce RED,GRN, and BLU image matrices which are used in the next algorithm.

٤.٣.٣ Inverse of Sub-Sample ٤:٢:٠

This process take U& V only color component (i.e. GRN ,and BLU image matrices) to make inverse of sub- sample by increasing U& V by multiplying by two in both direction horizontal and vertical , next algorithm represent it :

Algorithm ٤.٧ Inverse of Sub - sampling ٤:٢:٠

```
Input:-   Image matrix of U (row/٢, co/٢) & V (row/٢, col/٢) which sampling into half size .
Output:- Image matrix of U (row, col) & V (row, col) which inverse sampling into full size
STEP ١: Let DH = height / ٢ : DW = width / ٢
STEP ٢: For I = ١ to frames number   Do
            For row =  ١  to  DH          Do
             For col = ١  to  DW          Do
              Let P١ = GRN[I , row    , col   ]
              Let P٢ =  BLU[I , row  , col    ]
STEP ٣:   Let  GGRN [I , row *٢   , col*٢     ]=P١
             Let  GGRN[I , row * ٢+١ , col * ٢   ]=P١
             Let  GGRN[I , row * ٢  , col * ٢+١  ]=P١
             Let GGRN[I , row * ٢+١, col * ٢+١]=P١
             Let  BBLU[I , row * ٢  , col * ٢   ]=P٢
             Let  B BLU[I , row * ٢+١ , col * ٢   ]=P٢
             Let  BBLU[I , row * ٢  , col * ٢+١  ]=P٢
               Let   BBLU[I , row * ٢+١, col * ٢+١]=P٢
STEP ٤: END
```

٤.٣.٤ Inverse of Motion

This process only read the key frame number from compressed file and compare it with frame number output from I-DCT process. If it is equal copy, the frame is as key frame otherwise add the output frame of I-DCT process to previous frame this process apply only on Y luminance component in the next algorithm :

Algorithm ٤.٨ Inverse-motion based on skew

```
Input:-  key frames numbers from compressed file with differences image.
Output:- construct sequence image by adding difference image to key image.
STEP ١: read  key frame from compressed file
STEP ٢: Let  count = keyframe number
STEP ٣: Let keyframe = number of frame
STEP ٤: Copy Of Key Frame
          For nframs = ١ to count               Do
          For row = ١ to  height                    Do
           For col =  ١ to width                    Do
            Let RRED  (key frame (nframs),row ,col) =  RED(key frame
            (nframs),row ,col)
           END For Col
          END For Row
STEP ٥: calculate the addition of successive frame
         For I = (key frame (nframs)+١) to (key frame (nframs+١))
          If  I <>  (key frame (nframs+١)) then
           For row = ١ to  height                              Do
            For col =  ١ to width                              Do
             Let RRED (I,row ,col) =  RED(I,row ,col)+ RED(I-١,row ,col)
            END For Col
           END For Row
          Else
           Goto  STEP ٧
          End if
STEP ٦: END For  I
STEP ٧: END For nframes
STEP ٨: END.
```

٤.٣.٥ Inverse of TC YUV to RGB

This algorithm take Y ,U , and V three image matrices and apply the equation that transform them from YUV to RGB pixel channel which merge together to produce sequence of image in small algorithm :

Algorithm ٤.٩ Inverse-Transform color YUV to RGB

```
Input:- Y (row, col) luminance image ,U (row, col) & V (row, col)
chrominance  image which carry color informations.
Output:- :-   Red (row, col), Green (row, col), and Blue (row, col) image
matrices.
STEP ١: for I = ١ to frames number           Do
          for row =  ١ to height              Do
           for col = ١ to width               Do
            Let Y = RRED[I , row , col ]
            Let U = GGRN[I , row , col ]
            Let V = BBLU[I , row , col ]
```

```
STEP ٢:   R = Y +١.١٤*V
          If  R< ٠ then                    R=٠
           Else if R>٢٥٥ then              R=٢٥٥
           End if
STEP ٣:   G = Y- ٠.٣٩٥*U- ٠.٥٨١*V
           If  G < ٠ then                  G=٠
           Else if G>٢٥٥ then              G=٢٥٥
           End if
STEP ٤:   B = Y+٢.٠٣٢*U
          If  B < ٠ then                   B=٠
           Else if B>٢٥٥ then              B=٢٥٥
           End if
STEP ٥:   Let RED[I , row , col ] = R :   Let GRN[I , row , col ] = G
           Let BLU[I , row , col ] = B
STEP ٦: END .
```

٤.٣.٦ Reconstruct AVI

Write AVI block is the last stage of the Decoder , in this stage we form the structure of AVI movie file to play back again by using sequence images that obtained from previous stages of the Decoder which extract them from compressed file by using two process load AVI header which are used to store the all information needed to write AVI file and write AVI file algorithm which explain in details below :

Algorithm ٤.١٠ Write AVI

```
Input:- information of AVI header and structure with sequence image.
Output:- AVI movie file.

STEP ١: Let frame size = AVIPACK . bmpsize

STEP ٢: searchstr =٠

STEP ٣: DO

        Read from AVI long str
          If  AVIstr =long str then
            ١. searchstr =loc(AVIf)
            ٢. exit DO
          end if
STEP ٤:loop until EOF(AVIf)
STEP ٥: Put in AVIf AVI PACK
STEP ٦:Let AVIIndx.offset = offset value
 STEP ٧: Put in AVIf AVI Indx
              For I = ١ to Indxsize- len(AVIIndx)
               Put in AVIf b
             END for
STEP ٨: Put in AVIf  FCCLIST: Put in AVIf  CLng (size of list )
              Put in AVIf FCCodml: Put in AVIf FCCdmlh
              Put in AVIf CLng (size dmlh):Put in AVIf AVI PACK.total frame
 STEP ٩ :for I = ١ to size of list - ٤
                Put in AVIf Cbyte (٠)
               End for
STEP ١٠: Put in AVIf FCC Junk
               Let Junksize = CLng(offset value -loc(AVIf) -١٦
               Put in AVIf  Junksize
               For  I = ١ to Junksize
                 Put in AVIf   Cbyte(٠)
               End for
STEP ١١: Put in AVIf  FCCLIST
               Let Mloc=loc(f): Put in AVIf  CLng(١)
                Movielist size = loc(f): Put in AVIf  FCCmovi
                Let ix٠٠pos= loc(f): Put in AVIf  IX
                Let c= IX٠٠pos+١٦+AVIIndxsize
                For I = ١ to IX.ENteries
                 Put in AVIf  CLng( c) : Put in AVIf  frame size
               Enteries (I,١)= CLng( c): Enteries (I,٢)=frame size
```

```
        Let c= c+frame size+٨
          End for
STEP ١٢: for I = ١ to IX.Enteries
          Put in AVIf  FCC ٠٠db :   Put in AVIf  frame size
          For row =AVIPACK.bmp height  to ١decrease by ١
          For col = ١ to  AVIPACK.bmp width  increase by ١
           Put in AVIf    Cbyte BBLU(I ,row,col)
           Put in AVIf    Cbyte GGRN(I ,row,col)
           Put in AVIf    Cbyte RRED(I ,row,col)
           End for :End for
           End for
STEP ١٣: Let movie listsize =loc(f)- movielistsize
         Put in AVIf   FCCIdx١:    Put in AVIf   CLng(indxsize+١٦)
         Put in AVIf   FCC٧fxx:    Put in AVIf   CLng(٠)
         Put in AVIf   IX٠٠pos: Put in AVIf   CLng(٠)
STEP ١٤:for I = ١ to IX.enteries
         Put in AVIf   FCC٠٠db: Put in AVIf   CLng(١)
         Put in AVIf   Enteries (I,١): Put in AVIf   Enteries (I,٢)
        End for
STEP ١٥: Let RIFFsize = loc(f)
          AVIPACK.RIFFsize= CLng (RIFFsize)
          AVIPACK.AVIlistsize=CLng(offset value -٣٢ )
           Put in AVIf in first location  AVIPACK.RIFFsize
           Put in AVIf in( Mloc+١)CLng(movilistsize)
STEP ١٦: END .
```

Chapter Five

Results, Conclusions, Future Works

٥.١ THE Experimental Result:

٥.١.١ Result of Video Clip Movie Files

In this chapter we test the proposed compression system on different video clip files that represent in Table (٥.١) to obtain CR with PSNR to measure the quality of de compressed movie file by using :-

$$APSNR= \left(10\log_{10} \left(255^2\right) / eRMS \right) / \text{Total Frame}$$

To calculate the average PSNR of movie file ,these seven video clip were vary therefore have different CR with different PSNR . The results of test were summarized below.

Table (٥.١)Shows CR, average PSNR for each of the three channel component R, G, B , average PSNR of RGB of whole video clip & Total frame .

No.	Movie sample	Source file	Compress file	Compression ratio	No. of Frame	Apsnr. R	Apsnr. G	Apsnr. B	Apsnr. RGB
١		١٣٣٣١٦٠	٣٣١٠١	٤٠.٢٨:١	١٦	٣٧.٨٤	٣٧.٨٤	٣٧.٨٤	٣٧.٨٤
٢		٧٨٩٥٣٠٤	٢٤١٧٦٣	٣٢.٦٦:١	١٠٢	٢٩.٧١	٢٩.٦٥	٢٩.٣٠	٢٩.٥٥
٣		١٠٧٩٤٨٥٦	٣٤٣٥١٧	٣١.٤٣:١	١٤٠	٢٧.٣٤	٢٧.٦٩	٢٦.٩٢	٢٧.٣١
٤		٦٢٤٣٢٥٦	٢٠٤٦٠٦	٣٠.٥٢:١	٨١	٣٥.٠١	٣٦.٨٧	٣٤.٣٠	٣٥.٣٩
٥		٥٣٧٧٢٧٢	١٨٢٩٧٩	٢٩.٣٩:١	٦٩	٢٩.٤٨	٣٠.٤٥	٢٩.٦٩	٢٩.٨٧
٦		١٩٧٦٠٢٧٦	٧٥٤٧٤٤	٢٦.١٩:١	٣١٠	٢٩.٨٤	٣١.٢١	٣٠.٩٠	٣٠.٦٥

٧		١٩١٨٨٢٩٦	٧٨٤٦١٧	٢٤.٤٦:١	٢٥٩	٢٩.٤١	٢٩.٥٣	٢٨.٨٩	٢٩.٢٨

In the proposed system numbers of video file movies were taken which are difference in natural complexity, scenes quality, and size. These movie implemented with proposed system give the results which explain in table (٥-١), and we noticed the compression ratio increased if the files have few complexity with stable scene (i.e. the file contain one scene) such as the movie no.١ which contain ١٦ frames has high CR that equal to (٤٠:١) and PSNR(٣٧.٨٤) while the movie no. ٧ has which contain ٢٥٩ frame distributed over several scenes has low CR (٢٤.٤٦:١) and PSNR(٢٩.٢٨) , this mean that when scenes were varying and increased in video file the compression ratio decreased. In addition the PSNR is in negative with compression ratio for the same movie file, for example when reduce the value of (R)in quantization process to ١٠ instead of ٢٠ for movie no.١ the compression ratio will decreased to (٣٥.٥:١) and PSNR value will increased to (٤١.٢٩) , also depend on the complexity of the movie, such as movie no.١ because have simple image without complexity the PSNR is high while movie no.٧ because has natural image with some complexity the PSNR is decreased in spite of the CR is low.

٥.١.٢ Comparison with other Video Clip Compressor

In this section we compare our proposed system with standards systems based on DCT such as MPEG-١ and H.٢٦١ can be summarized in several points as differences between them as follow:

١. It differ by using run length encoding(RLE) instead of zero run length merge with Huffman coding .

٢. It differ by accept in all image dimension in (row ,column) instead of (٣٥٢,٢٤٠)in H.٢٦١ or only video format(DTV ,CIF ,QCIF ,sub-QCIF) as MPEG .

٣. It differ by using scalar quantization which depends on value of (R) such that when increase value of (R) the compression ratio increased instead of use quantization table as MPEG .

٤. It differ in motion compression such that our algorithm split movie into scenes depends on the value of SKEW for each frame and then make simple direct difference between sequence frames in individual scene with store number of key frames which present the initial frame of each individual scene in compressed file instead of using motion compression algorithm which is called Motion Estimation /Compensation that are require complex operation , time and store Motion vectors for macro blocks in each individual frame.

In spite of all these differences between the proposed system and standard systems for video compression but our algorithm reach to Compression Ratio lies in
(٢٠ -٤٠) instead of (٢٠ -٣٠) as MPEG-١.

٥.٢ Conclusions

In this section we will present some conclusions:

١. The proposed system can split any movie file into scenes by calculating skew measurement for each frame and comparing the skew value of sequence frame by using a specific threshold for that purpose.

٢. Natural of movie file contained temporal redundancy i.e . the sequence frames are identical with some differences ,because of the motion make the whole frame is similar to the previous except only some regions change which represent the motion in that frame , thus the differences between frames due to smaller pixel value. Therefore the coefficients produced through DCT tend to zero which make the compression operation by using Run Length Coding very high .

٣. The quantization process not used to exclude the real value of coefficients but also to exclude negative value from it by using normalization which calculate the maximum and minimum value of each block of DCT coefficients and subtracted them to obtain positive value only , that makes Run Length Coding effective.

٤. The compression ratio based on the number of frames in each scene so that the relation between them is positive because each frame in that scene is similar to the previous one with few differences between them, so that calculate motion based on skew will increase compression ratio while the relation is in negative between the compression ratio and number of scene (i.e. when increase number of scene by decreasing threshold which is used with skew measurement the number of frame in each scene will decreased ,therefore increase the key frames and decrease the delta frames so that the differences will decreased and subsequently compression ratio will decreased .

٥.٣ Future Work

١. Using another transform technique instead of Discrete Cosine Transform such as Discrete Wavelet Transform.

٢. Applying this proposed system to movie file contain video / audio.

٣. Using image enhancement methods to enhance the decompressed sequence frames of movie file .

٤. Applying another data compression methods such as Huffman, arithmetic, and embedded zero tree wavelet .

References

١. K.Holtz, "Image and Video Compression Techniques"
site :// http // www . AutoSophy.com / video com.html ,٢٥/١/٢٠٠٦

٢. B.Tower,"H.٢٦٣ video compression : the standardized advantage " March Network corporation®, September ٢٠٠١.

٣. N. Schaab, " Compressed Video In the Mobile Environment ", Virginia Polytechnic Institute and University © by Chris N.Schaab, ٢٠٠٤.

٤. L. Kizewsk ," Image De-blocking Using Local Segmentation" School of Computer Science and Software Engineering, Monash University, November, ٢٠٠٤.

٥. T.sikora,"MPEG-١ and MPEG-٢ digital video coding standard ",digital consumer electronic hand book, Mc Grow - Hill Book Company,Ed.R.Jurgens,١٩٩٨.

٦. A. Darwish ,"Scalable Video Coding " , Clyton School of Information Technology, Monash University, November, ٢٠٠٥.

٧. P. Lindmark ,"Optimizing Video Compression and Streaming Media for E-Learning" MS_SC thesis IT university of GÖteborg, Sweden ٢٠٠٥.

٨. Bamboo Web Dictionary, "Video Codec",
site:// http // www.BambooWeb.com/articls/v/i/videocodec.html,
٢٤/١٢/٢٠٠٥

٩. W. Niblack, "An Introduction of Digital Image processing" ,prentice hall international ,١٩٨٦.

١٠. E .Scott ,"Computer Vision and Image Processing : A practical approach using CVIP tools " Prentice –Hill , NewJersy, ١٩٩٨.

١١. B. Andrew "Image Compression Using Discrete Cosine Transform ", Mathematics Journal vol.٤, no.١, pp ٨١-٨٨, ١٩٩٤.

١٢. R. Gonzalez & R. Woods ,"Digital Image Processing (٢nd addition) " NewJersy , prentice –Hill ,٢٠٠٢.

١٣. L. shen ,"Image Analysis and Processing " , Computer and Information Science U Mass Dartmouth, ٢٠٠٥.
site://http//www.cis.umassd.edu/ ~lshen/ courses/ ٢٠٠٥_spring_cis٤٦٧/notes.

١٤. D. Salomon ,"Data Compression" ,dept of computer science , California University, Verlag NewYourk,١٩٩٨.

١٥. B. Digan , & Koyanagi (١٩٩٧) ,"Streaming Video "
site :// http // www.Is.unc.edu/video / quality.html (٣-١-٢٠٠٥).

١٦. C. Nilsson ,& B. Marianne (١٩٩٨) "Introduction Till Digital Video "
site ://http//www.Mosda .VXU.Se /multimedia/ introdigi.html, ٩-٣-٢٠٠٥ .

١٧. D.G Jameson ," Multimedia in Teaching Space (١٩٩٨)" R
site :// http // www.agocg.ac.uk / reports / mmedia/casestdy/alc ,٣-١-٢٠٠٥

١٨. Anderson ,& Pertal (١٩٩٧) "dator och rÖrliga bilder",
site :// http // www٢. educ.umu .se / ~pamac / video ,٣-١-٢٠٠٥ .

١٩. C. Charlotta, "Multimedia Pa WWW",
site :// http // www.iml.umu.se / stadie information / arbeten / multimedia-www / kapitel ٥-٦ .html, ٣-١-٢٠٠٥ .

٢٠. D.Cunningham , & F. Niel (٢٠٠١)," An Introduction to Streaming Video:"
site :// http // www.cultivate– int org /issue٤/video, ٣-١-٢٠٠٥ .

٢١. W. Anthony & M. berry (٢٠٠٢)," Interlace & Progressive Scan (٢٠٠٢)",
site ://http//www. aluminums studios.com /digital video /advanced / interlaced/interlaced .html, (٣-١-٢٠٠٥).

٢٢. O. K. AL-Shayk, E. Milo, T. Nomura, R. Neff and A. Zakhor ," Video Compression Using Matching Pursuits", Department of Electrical

Engineering & Computer Science, California University, IEEE Transaction on circuits & system for video technology , February ١٩٩٩.

٢٣ . N. Merhav & V. Bhaskaran ," A Fast Algorithm for DCT – domain Inverse Motion compensation ", Computer Systems laboratory & Hewlett – Packard laboratories (HP) in Isreal Science Center, Technion city , Haifa -٣-٢٠٠٠.

٢٤. S. Einerman ,"Residual Coding " ,Department of signal ,Sensors and Systems, Royal Institute of Technology (KTH), December ٢٠٠١ .

٢٥ A. Said , "Compression of Compound Images and Video for Enabling Rich Media in Embedded Systems", Imaging System laboratories HP laboratories Palo Alto, © Hewlett – Packard company, May ٢٠٠٤.

٢٦. A. Ford & A. Robert , "Color Space Conversions", August ١٩٩٨ .
site :// http://www.poynton.com /poynton-color.html.

٢٧ . D. F. Rogers ," Procedural Elements For Computer Graphics ", United State Naval Academy, Annapolis, Md McGraw – Hill Book Company,١٩٨٥.
٢٨. Adobe Dynamic Media Group, "A digital Video Primer ", Updated & Enhanced , ٢٠٠٢ .
Site://http://www.adobe.com/products/aftereffects/demodnld.html

٢٩. Y. Wang, " Video Coding Basics" , Polytechnic University, Brooklyn, , ٢٠٠٥ .

٣٠. Y. Wang, J. Ostermann, Y. Q. Zhang, "Video Processing and Communications", Prentice Hall, ٢٠٠٢.

٣١. B. Girod & M. Flierl ," Multi- Frame Motion Compensated Video Compression For The Digital Set-Top Box", Information System Laboratory, Dept. of Electrical Engineering , Stanford University, September ٢٠٠٢ .

٣٢. S .Aljosch, S . Heik & W. Thomas ,"Long –Term Global Motion Compensation For Advanced Video Coding ", Heinrich –Hertz – Institute, Berlin, Germany, ٢٠٠٤.

٣٣. S. Soongsathitanon & S. S. dlay, "A New Orthogonal Logarithmic Search Algorithm For Fixed Block-Based Motion Estimation For Video

Coding “, Dept. of Electrical & Electronic Engineering, University of New Castle, ٢٠٠٠ .

٣٤. D. Martin, “ Predication Based Block Matching for Video Encoding”, Royal Institute of Technology, Dept.of Signal, Sensors & System Signal Processing, December, ٢٠٠١.
٣٥. S. Richmond ,” A Low-Power Design of Motion Estimation blocks for Low Bit –Rate Wireless Video Communication “ , MSc, Electrical Engineering, Virginia a Polytechnic Institute & State University, March, ٢٠٠١ .

٣٦. D. Babic, “Discrete Cosine Transform Algorithm for FPGA Devices “, Zagreb, April, ٢٠٠٣ .

٣٧. S. A. Khayam ,”The Discrete Cosine Transform Theory & Application “, Dept. of Electrical & Computer Engineering, Michigan State University, March, ٢٠٠٣.

٣٨. W.B. Pennebaker & J.L. Mitchell ,” JEPEG–Still Image Data Compression Standard “, New York: International Thamson publishing ١٩٩٣.
٣٩.G.Strang,”The Discrete Cosine Transform“, SAIM Review, Vol.٤١, No.١, PP.١٣٥-١٤٧, ١٩٩٩.

٤٠. T. A. Abbas , “A Hybrid Algorithm For Images Compression “, Depart. Of Computer Science of Technology University, April ٢٠٠٤.

٤١. R.J. Clarck, ”Transform Coding Of Images ”, New York : Academic Press١٩٨٥.

٤٢. R. Hayder, ”Lecture Notes: ECE ٨٠٢- Information Theory & Coding”, IEEE Computer Graphics and Application , January, ٢٠٠٣.

٤٣. M. Akuri , "Image Compression Using Discrete Cosine Transform", Dept of Electronics &Computer Technology ,Indiana State University ,December,٢٠٠٤.

٤٤. B. McGuigan ,” What Is a Codec “ copyright by Wise Geek ,٢٠٠٥.

٤٥. Bamboo Web Video ,”Video“
site :// http // www.bambooweb .com /articles/v/i/video .html ٢٤-١٢-٢٠٠٥

٤٦.R. Hawkins ," Digital Stereo Video: Display, Compression & Transmission ", Australian Nation University , February ,٢٠٠٢.

٤٧. D. A. Huffman, ``A Method for the Construction of Minimum Redundancy Codes,

site://http://www.vectorsite.net/ttdcmp١.htmlCopyright©ScienceOxygen. com all rights reserved, ٢٠٠٤.

٤٨. M. Nelson & J. Gailly, "The Data Compression Book ", ٢nd edition .M & TBooks,١٩٩٦.

٤٩. G. Lu & S. Teng ,"A Noval Image Retrieval Technique Based on Vector Quantization ", processing of int'١conference on computational Intelligence for Modeling Control & Automation,١٩٩٩.

٥٠. F.Iqbal ," Wavelet Transform based Image Compression on FPGA " , Mc thesis in Collage of Engineering Florida State University, Spring ٢٠٠٤.

٥١.White paper Axis communication, "compression techniques ", site :// http // www.axis .com© Axis communication, ٢٠٠٢.

٥٢. P. Bahl & P. S. Gauthier , " Software Only Compression : Redering, & Playback of the Digital Video", Digital Technical Journal, Vol.٧,No.٤, ١٩٩٥ .

٥٣.Array Microsoft Systems .Inc ," Video Compression " ,١٩٩٧,
site :// http // www.Array .com /compress . pdf .

٥٤. BambooWeb Dictionary ,"H.٢٦١",
site :// http // www.bambooweb. Com/articles /h/H٢٦١.html, ٢٤-١٢-٢٠٠٥

٥٥. C-CUBE ,"Compression Technology : An MPEG Overview ",
site :// http // www.C-CUBE .com, September ١٩٩٩ .

٥٦. Lund head office White Paper communication in Sweden , "Digital Video Compression: Reviewing Methodologies & Standards to Use for Video Transmission & Storage " ,
site ://http // www.axis .com© Axis communication, June ٢٠٠٤.

٥٧. Microsoft DirectShow ٩.٠," AVI RIFF File Reference" , site//http://msdn.microsoft.com/library/default.asp,©٢٠٠٦Microsoft Corporation.

٥٨. Open DML AVI File Format Extensions, published by the Open DML AVI M-JPEG File Format Subcommittee, September ١٩٩٧.

الخلاصة

تقنيات ضغط الفيديو تلعب دور أساسي ومهم جداً في الخزن على أجهـزة الكومبـيوتر وفـــي النقل عبر الإنترنيت خلال حزمة ذات عرض محدد(ثابت) . لاحتواء ملفات الفـيديـو الحركية على تعاقب متكرر من الصور الثابتة المتشابهة فيما بينها مع وجود قـليل مـن الاختلاف (ضمن المشهد الواحد) حيث إن كل حركة في ملف الفــيديو تمثل بـ ٢٤ صورة ثابتة أو أكثر، لهذا فأن التعاقب المتكرر من الصور يوفر لنا كمية كبـــيرة مـن الفائضية الوقتية. بالإضافة إلى الفائضية الحيزية (الموقعية) المحـتواة في البيــانات للصورة الواحدة.

النظام المقترح ينجز نســبة ضغط تتراوح بـين (٢٠-٤٠) وهي مقاربة لنسبة الضغط التي تنجزها تقنيات ضغـــط ملفات الفيديو المعروفة MPEG.

النظام المقترح يمكن أن يلخص بأربع مراحل أساسية هي :-

المرحلة الأولى تتضمن فتح ملفات الفيديـو من نوع RIFF.AVI والحصول على التعاقب الصوري لملف الفيديو . المرحلة الثانـية تتضمن ضــغط التعاقــب الصــوري باستخدام تقنيات الضغط والممثلة بالضغط بفقدان المستخدم في ترميز الفيديو من خلال استخدام تقنيات التحويل والممثلة بالتحويل الجيبي المتقطع لتقليل الفائضية الحيزية في بيانات الصورة ذاتها وتقليل الفائضية الوقتية باستخدام تقنيات تعويض الحركة والتي تستخدم مع التحويل الجيبي المتقطع لإنتاج خطأ الصورة باستخدام تقنيات ضغط الحركة بالاعتماد على مقياس الالتواء وفي نهاية هذه المرحلة نستخدم طريقة كفؤة للترميز وخزنها في ملف الضغط ، المرحلـة الثالثة تتضمن عــكس عمليات الضغـط للحصول على التعاقب الصوري، المرحـــلة الرابعة تتضمن إعادة إنشاء ملف الفيديو من نوع RIFF.AVI .

النتائج التي تم الحصول عليها لنسبة الضغط مع مقياس الجودة للصورة الناتجة بعد فك الضغط (PSNR) لكل عنصر من عناصر الصورة الثلاثة (الأحمر، الأخضر، الأزرق) مع معدل (PSNR) لكل ملف الفيديو للمقارنة بين نوعية ملف الفيديو الناتج بعد فك الضغط مع ملف الفيديو الأصلي. كل هذه الملفات أختبرت على حاسبة من نوع Pentium ٤ وسرعة معالج ٢١٠٠GHz وذاكرة رئيسية ٢٥٦MByte اللغة المستخدمة Visual Basic ٦ النظام المقترح ينجز نسبة ضغط عالية مع فقدان للبيانات والتي توضحها قيم PSNR لملف الفيديو

ضغط ملفات الفيديو باستخدام تقنية تحويل الجيبي المتقطع

رسالة مقدمة إلى
مجلس كلية العلوم- جامعة بابل
وهي جزء من متطلبات نيل درجة الماجستير علوم
في علوم الحاسبات

من

وفاء حسن علوان المرسومي

جمادى الأول - ١٤٢٧ **أيار-٢٠٠٦**