Video Coding:

Comparison of Block Matching Techniques In Motion Estimation

Prepared by;

Nurul Najahah Abu Bakar

1974

Electrical and Electronics Engineering

Project Dissertation submitted in partial fulfilment of

the requirements for the

Bachelor of Engineering (Hons)

(Electrical & Electronics Engineering)

DECEMBER 2004

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

Video Coding: Comparison of Block Matching Techniques In Motion Estimation

by

Nurul Najahah Abu Bakar

A **Project Dissertation** submitted to the
Electrical & Electronics Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
BACHELOR OF ENGINEERING (Hons)
(ELECTRICAL & ELECTRONICS ENGINEERING)

Approved by,

(Miss Nasreen Badruddin)

UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

December 2004

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

NURUL NAJAHAH ABU BAKAR (E 1974)

ABSTRACT

This project is to investigate the advantages of using the various types of block matching methods as a type of motion estimation techniques compared to encoding each frame as a separate static image. As video is continuous media, it is important to maintain its quality and efficiency while compressing it. As the similarity of a frame and the next frame is great, it can be used as the advantage in video coding. This is because, the background of the image will usually stay the same and the only thing that will change is the moving object in that video. In order to achieve this, the author has to develop a program that can accommodate motion compensation and estimation by utilizing the computational for the motion estimation techniques, if applicable. The author will then have to run the program with some test video sequences to compare the performances of different block matching techniques for different types of video sequences. The major types of block matching techniques are Full (Exhaustive) Search, and a Fast Search (Three Step Search) has been chosen for this project. The author has also chosen to work on Quarter Common Intermediate Format (QCIF) video sequences. As the purpose of this project is to investigate the motion estimation technique, only the video frames will be considered and the sound of the actual video will be left out.

Experimental results show that MMSE has a better PSNR value than MAD but consume more time and has higher complexity of operation. Block sizes and window sizes also have a significant effect on the predicted image. The Three Step Search has experiments has shown that it has a higher speed ratio as compared to Full Search, but with reduced quality.

ACKNOWLEDGEMENTS

Alhamdulillah, with the greatest gratitude to the Almighty Allah for his gracious blessings throughout the whole period this project was undertaken. My deepest gratitude goes to my parents, who have provided me with their love and undivided attention through the sweet and sour moments. To my supervisor, Miss Nasreen Badruddin, thank you so much for your guidance and support throughout this project. Her words of wisdom have encouraged me to rise again during the times I fall.

I would also like to express my sincere appreciation to the lecturers' involved, especially Professor Venka who has given me ideas me during the first semester of this project. My gratitude also extends to all the lab technicians, especially Miss Siti Hawa whose presence is truly meaningful and valuable.

I would like to take this opportunity to thank all my friends who has contributed to this project. Your assistance is duly acknowledged and noted.

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

1.1 BACKGROUND OF STUDY

Information and technology has become a crucial part in today's society. This is more or less justified by the multimedia software that incorporates digital audio, images and videos. Thus, nearly all forms of information such as video are being increasingly generated, manipulated and transmitted in digital form. This also has resulted in advanced applications such as bigger secondary storage on personal computers from 20 Megabyte to tens of Gigabyte and the transmission bandwidth of modem from a mere 300 bps to 100Mbps (Local Area Network). Previously a single floppy disk actually can accommodate the storage, but with the fast transmission and such, the demand is higher. These caused the development of data compression like video compression techniques that can lead to more efficient use of the limited transmission and storage resources available.

1.2 PROBLEM STATEMENT

1.2.1 Problem Identification

Video to an information class called *continuous media*. It is characterized by the essentially continuous manner in which the information is presented, different from the *discrete media* such as images[3]. Thus, the temporal dimension becomes important. As an example, a video sequence compressed with a constant image quality for every frame is more desirable than the one with the image quality varies overtime. However, because the compressibility of individual frames varies over time, maintaining the constant image quality also made the coding rate varied.

1.2.2 Significance Of The Project

In a video sequence, usually the image from a frame to the next frame is very similar. The background of the image may not change except for the small change by a moving object in the frame. Thus, instead of compressing the video frame by frame, the motion estimation techniques can be used. These techniques only employ the compression of the first frame in the video and assigned it as the reference frame. The following frames will be compared to this reference frame by pixels. These will result to a set of motion vectors that describe the pixels movements. These vectors will then be used by decoder to construct the video sequences.

1.3 OBJECTIVES

The author must accomplish these objectives by the end of this project. They are as following:

1) To develop a program that can accommodate motion compensation and estimation
2) To utilize the computation for the motion estimation techniques (if applicable) as specified in the project scope.
3) To run the program with some test video sequences and compare the performances of different motion estimation for different types of video sequences.
4) To investigate different factor affecting motion estimation, for example block sizes.
5) To compare the output of compressed video sequence to the output original video sequence.

1.3.1 The Relevancy Of The Project

The process of compressing each frame in a video sequence will consume so much time and actually gave so much workload on computers' memory. Utilizing the motion estimation techniques will save time and decrease the workload of the compression, thus, increasing video compression's efficiency and effectiveness.

1.3.2 Feasibility Of The Project Within The Scope And Time Frame

This project supposedly has to be done within the twenty-five (25) weeks time frame. The author has decided to utilize the first half of the period to explore and learn more about pixels, video coding and sequences, and be able to use at least one of the techniques in image reading. The time frame may not be adequate for all techniques and video format investigation. So, the author decided to minimize the scope to one video component format and limiting block matching motion estimation techniques used to Full (Exhaustive) Search and a Fast Search Method namely the Three Step Search method for the comparison purpose.

1.4 SCOPE OF STUDY

This project will investigate the advantages of using the various types of Block Matching in motion estimation techniques compared to encoding each frame as a separate static image. The performance of the techniques will be compared by running the written program with some video sequences. The major types of block matching techniques are Full (Exhaustive) Search, Hierarchical Search, Generalized Search and the Fast Search (e.g. Three Step Search, Cross Search, etc). This method uses the Minimum Square Error (MMSE) and Minimum Absolute Difference (MAD) Maximum factors, which meets the project objectives of investigating the motion estimation techniques in video compression using these formulas. Some different factors which will also be considered in the comparison are block sizes, window sizes and the computational time for each process.

CHAPTER 2

LITERATURE REVIEW AND/OR THEORY

2.1 SUPPORTING INFORMATION (E.G., REFERENCES, ETC.)

2.1.1 Basic Terms in Video Processing

In order to investigate video compression techniques, the author have to be familiar with the basic terms in video processing. Some of the basic terms are:

- *Pixel* → the smallest discrete component of an image or picture on a CRT screen. It is usually a coloured dot and the greater the number of pixels per inch the greater the resolution.
- *Pel* →usual abbreviation used in the MPEG standard, instead of pixels.
- *Resolution* → the height and width of an image as a quantity of pixels in each direction. It can also represent the number of pixels per linear measure. In digital video, the lack of spatial resolution can result to pixellation or aliasing artefacts, while in analogue video; it will cause blurring of image in respective direction.
- *Pixellation* → the jagged edges resulting from individual pixels becoming visible. The visibility will depends on size of display and viewing distance.
- *Bitmap* → an arrangement of pixels and lines in a contiguous region of the memory. It has five (5) key parameters, namely; starting address in memory, numbers of pixels per line, pitch value, number of lines and number of bits per pixel.
- *Pitch value* → will specifies the distances in memory of the start line to the next.
- *Colour Mapping* → maps the 2^{24} distinct colours to 256 colours for 8-bits monitors without noticeable loss of colour resolution.

2.1.2 Video Component

A video sequence is made up of individual pictures occurring at fixed time increments. These individual pictures or still images are called frames. As the pictures are in colour, each picture must have its own video components. From Wikipedia.org, component video is a video signal that is transmitted as several separated channels (as opposed to a composite video signal such as NTSC or PAL, which is transmitted as single signal). Most component video signals are variation of the red, green and blue signals that make up television images. The simplest type is RGB, consists of the three discrete R, G and B signals sent three down cables. This type is commonly used in Europe through SCRAT connectors. Another superior type of video components consists of R-Y, B-Y and Y, delivered the same way. This is the most common signal type used today. It has three components, namely one *luminance* (Y) and two *chrominance* (U and V) components. U, (referred to R-Y) is the red component minus the luminance information and V, (referred to B-Y) is the blue component minus the luminance information. Luminance (or brightness) provides a monochrome picture and chrominance (colour) express the equivalent of colour hue and saturation in picture.

Component digital video signals are sometimes referred to as 4:2:2, meaning that for every 4 bits that are dedicated to the Y component, 2 bits each are dedicated to the U and V components on both even (second 2) and odd lines (third 2) of the image. The luminance or Y channel carries most of the image detail and is, therefore, assigned more bits. Another common method, 4:2:0 is used on DVDs. In this case, only the even lines have colour information; for the odd lines it is approximated by interpolation. This signal is often converted to 4:2:2 inside the player before it is sent out to other devices. The common high-end professional formats are YUV 4:2:2, used in CCIR 601 video, and the enhanced YUV 4:4:4 format, which has even higher quality. The use of YUV 4:2:2 is possible because the human eye is much better at seeing differences in brightness than in colour.

The primary advantage of YUV is that it remains compatible with black and white analogue television. The Y signal is essentially the same signal that would be broadcast from a normal black and white camera (with some subtle changes), and the U and V signals can simply be ignored. When used in a colour setting the subtraction process is reversed, resulting in the original RGB colour space. Another advantage is that the signal in YUV can be easily manipulated to deliberately discard some information in order to reduce bandwidth. The human eye actually has fairly low colour resolution, the high-resolution colour images human see actually being processed by the visual system by combining the high-resolution black and white image with the low-resolution colour image. Using this information to their advantage, standards such as NTSC reduce the amount of signal in the U and V considerably, leaving the eye to recombine them. For instance, NTSC saves only 11% of the original blue and 30% of the original red, throwing out the rest. Since the green is already encoded in the Y signal, the resulting U and V signals are substantially smaller than they would otherwise be if the original RGB or YUV signals were sent. This filtering out of the blue and red signal is trivial to accomplish once the signal is in YUV format. YUV or also called YCrCb is a versatile format which can be easily combined into other legacy video formats. For instance if the U and V signals are amplitude-modulated onto quadrature phases of a sub carrier, they will end up with a single signal called **C**, for *chroma*, which can then make the YC signal that is S-Video. If the Y and C signals are mixed, they will end up with composite video, which almost any television can handle. All of this modulating can be accomplished easily in low-cost circuitry, while the *demodulation* is often very difficult indeed. Leaving the signal in the original YUV format thus made DVDs very simple to construct, as they could easily down mix to support either S-video or composite and thus guarantee compatibility with simple circuits, while still retaining all of the original information from the source RGB signal.

There are two types of sub sequences in YUV Sequences, namely; CIF, which is also known as the Common Intermediate Format, is used to standardize the horizontal and vertical resolution in pixels of YUV sequences video. The resolution in pixels is 352 x 288. QCIF means Quarter CIF and to have fourth area as quarter implies, height and width of the frame are halved. The resolution in pixels is 176 x 144. The colour range for YUV is as shown in **Figure 2.1.2.**

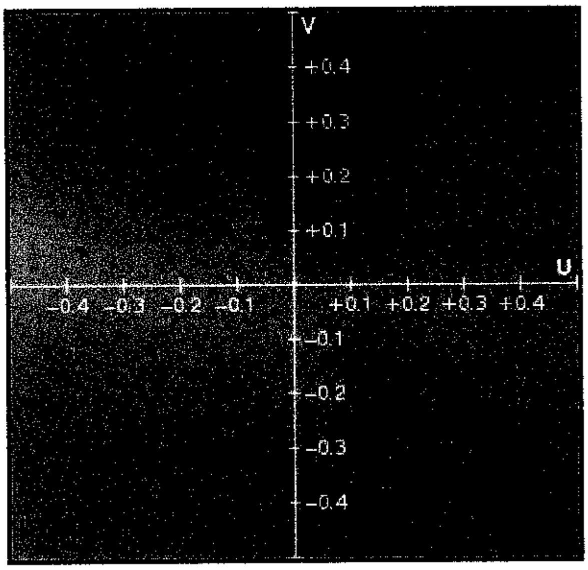


Figure 2.1.2: Range of colours for YUV

2.1.3 Compression

By definition, compression means the process by which the description of computerized information is modified so that the capacity required to store it or the bit-rate required to transmit is reduced. The need for data compression is based on the facts that voluminous multimedia data needs very high capacity of bandwidth and storage and a high processing speed. Compression makes it feasible and cost-effective to use real-time communications over data network, or to store high volume data on digital media.
Compression is subject to certain requirements and constraints, namely;

- Quality of decompressed data should be as good as possible
- Compression and decompression should be hardware independent
- The processing algorithm should not exceed certain time span.

2.1.4 Video Compression Formats

Some of the popular video compression standards are the H.261, MPEG-1, MPEG-2 and MPEG-4 standard. The H.261 standard is developed to facilitate videoconferencing and videophone services over integrated services digital network (ISDN). According to the MPEG Video Compression and Standard by Joan Mitchell, MPEG-1 is intended for intermediate data rates (1.5Mbps), MPEG-2 is for higher data rates at 10Mbps or more and MPEG-4 is for very low data rates (about 64kbps or less). MPEG-2 and MPEG-4 thus have potential uses in telecommunications.

2.1.5 Motion Estimation Technique: Block Matching Method

For this project the Full (Exhaustive) Search and a Fast Search Method namely the Three Step Search method has been chosen for the comparison purpose The Block-based estimation is one of the most popular approaches and has been adopted in the international standards for digital video compression, such as H.261 and MPEG 1-2. It is also widely used in other video applications such as motion compensated filtering for standards conversion. This technique has three different methods, namely matching criteria, search strategy and determination of block size[1]. The most popular and commonly used method is the Block Matching Criteria method simply because its practicality and minimum hardware complexity. In Block Matching method, the factors of Minimum Square Error (MMSE) and Minimum Absolute Difference (MAD) (alternatively Maximum Matching Pel Count (MPC) can be used but not evaluated in this project) are used in order to compares the frames in the video sample.

2.1.6 Window Size

For every block, a search region of (B + 2W) x (B + 2W) where W is the predetermine integer called the "search window" has been assigned (B is the block size). This estimation model assumes that the image is composed of moving blocks; with its simplest form is the translatory block, where the motion of each block is restricted to a pure translation.

2.1.7 Block Size

Considering N as the side of each block, *B*, the block size is B x B in frame k centred about the pixel **n** with the coordinate of (n_1, n_2), the block is modelled as a globally shifted version of a same-size block in frame k + l for an integer l. The simplified image formula is[1];

$$s\,(n_1, n_2, k) = s\,(n_1 + d_1, n_2 + d_2, k + l)$$

where d_1 and d_2 is the component of displacement (translation) vector for block *B*.
One of the reasons the author chose the Block-based method is because it consider the minimum mean square error (MMSE) and minimum mean absolute difference (MAD), which is the specification in this project investigation.

2.1.8 Minimum Mean Square Error (MMSE)

MMSE is defined as[1]:

$$MMSE(d_1,d_2) = \frac{1}{B1B2} \sum_{(n1,n2)\in B} [s\,(n_1, n_2, k) - s\,(n_1 + d_1, n_2 + d_2, k + l)]^2$$

MMSE criterion unfortunately, is not commonly used in actual implementation as it is hard to be realized due to its square operation. This is true especially in hardware.

2.1.9 Minimum Mean Square Error (MAD)

On the other hand the MAD criterion is defined as[1]:

$$MAD(d_1,d_2) = \frac{1}{B1B2} \sum_{(n1,n2)\in B} |s(n_1, n_2, k) - s(n_1 + d_1, n_2 + d_2, k + l)|$$

However, it is well known that the performance of the MAD criterion deteriorates as the search area becomes larger due to the presence of several local minima (complication to choose the best candidates as there are many candidates that have lowest value).

2.1.10 Peak Signal-to-Noise Ratio

The performance of the predicted image when compared to its original one will be shown by the Peak Signal-to-Noise Ration (PSNR) in decibels (dB):

$$PSNR = 20 * \log10(b/RMS)$$

where; b is the largest possible value of the signal (typically 255 for greyscale or 1 binary images), and RMS is the root mean square difference between two images. The PSNR is given in decibel units (dB), which measure the ratio of the peak signal and the difference between two images. An increase of 20 dB corresponds to a ten-fold decrease in the rms difference between two images. There are many versions of signal-to-noise ratios, but the PSNR is very common in image processing, probably because it gives better-sounding numbers than other measures[7].

2.1.11 Full (Exhaustive) Search

In this search strategy, the comparisons and evaluations are done pixels by pixels. This is done by moving the block, B by one pixel at a time, in its given search window, W. thus, it is extremely time consuming. The search window is used to limit the computational burden of the Full Search. The flowchart of this process is as shown in **Figure 2.1.11**:

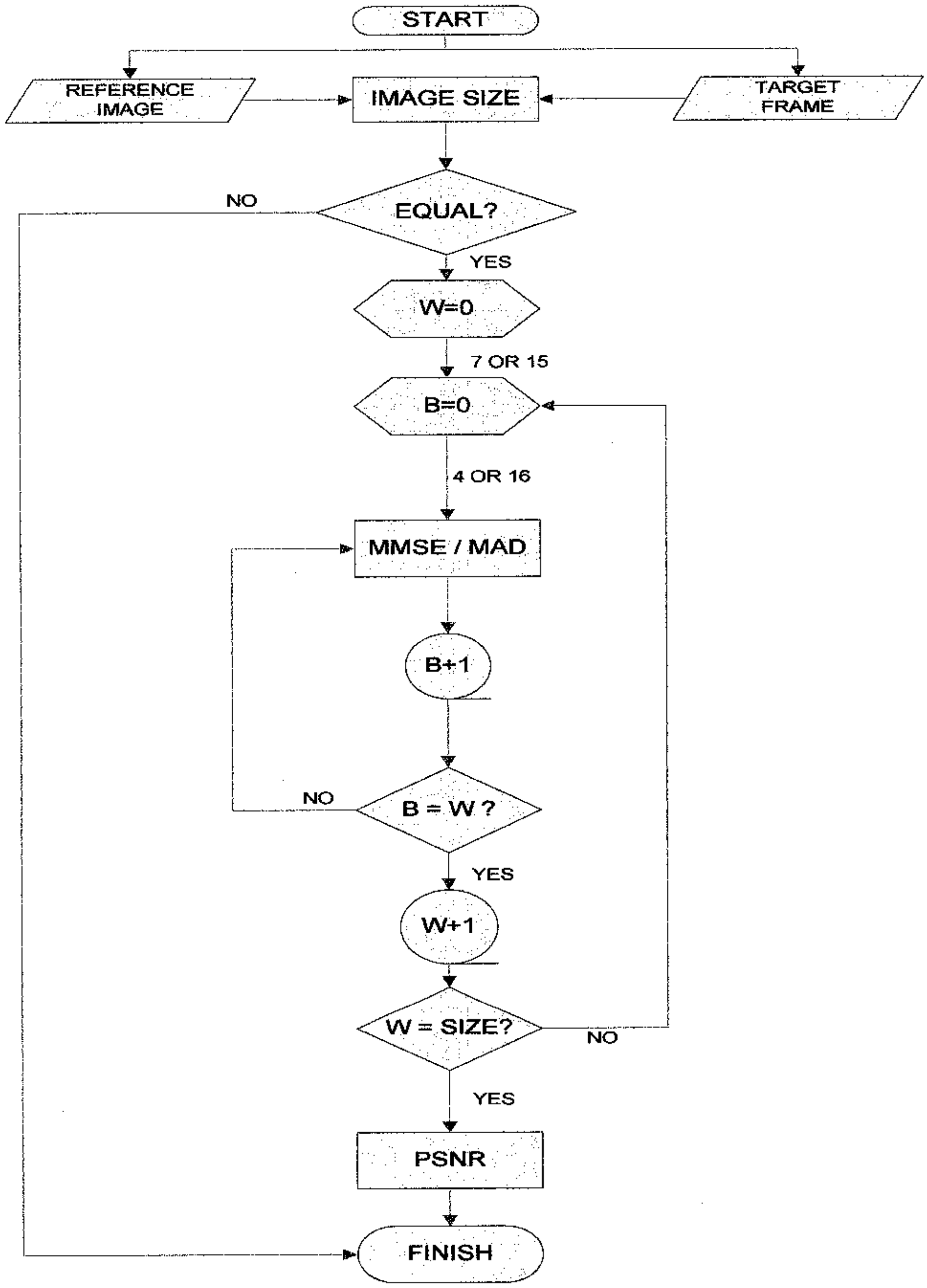


Figure 2.1.11: Flowchart of Full (Exhaustive) Search procedure

2.1.12 Fast Search: Three Step Search

As shown in **Figure 2.1.12(a)**, the search frame is depicted with the search window of W = 7 with range of -7 to 7. The coordinates (0,0) of the search region indicates the starting or middle point. In the first step, the criterion is evaluated at nine points (at coordinates (0,0) and the first step points). If the lowest MMSE or MAD is found at the coordinate (0,0), then the block have no motion. If not, the second step will be evaluated at eight points centred about the first step points that have the lowest MMSE or MAD. The third step will proceed centred about the second step points that have the lowest evaluation. Point to note that this search strategy does not necessarily have three steps. The number of steps might increase depends on the window size based on the equation: [5]

$$L = \log_2 (W + 1)$$

Thus for W = 7, L = 3 and for nth step, the step size is:

$$ss(n) = 2^{(L-n)}$$

This means that for first step, ss(1) = 4, for second step, ss(2) = 2, and for third step ss(3) = 1 as indicated in the figure.

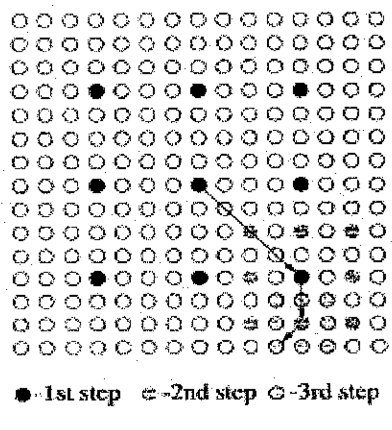


Figure 2.1.12(a): TSS procedure for W = 7

The flowchart of the Three Step Search Process is as shown in **Figure 2.1.12(b):**

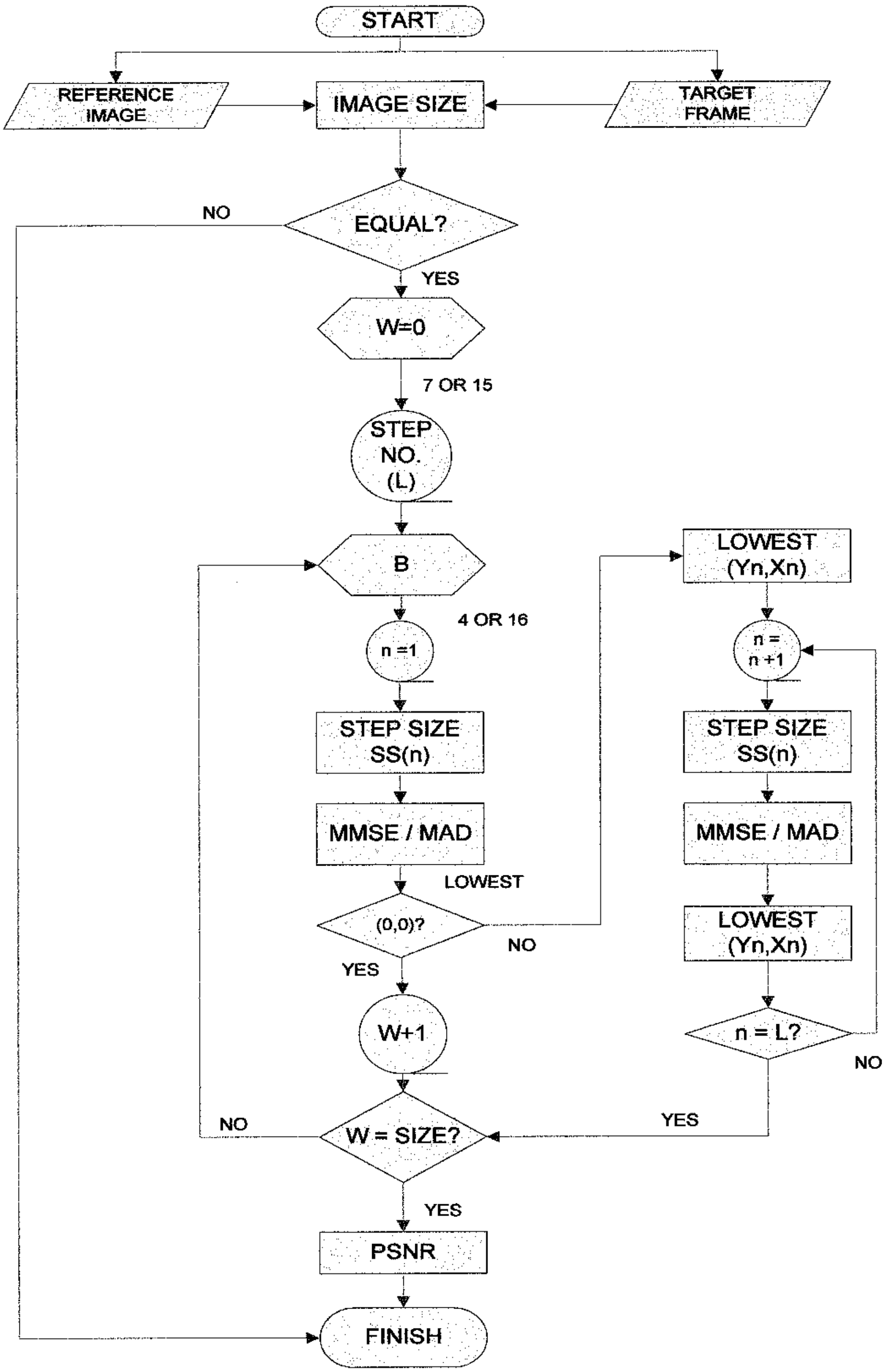


Figure 2.1.12 (b): Flowchart of Three Step Search procedure

CHAPTER 3:

METHODOLOGY/PROJECT WORK

3.1 PROCEDURE IDENTIFICATION

The procedure that has been aligned for this project is divided to the stages as shown in the flowchart in **Figure 3.1**:

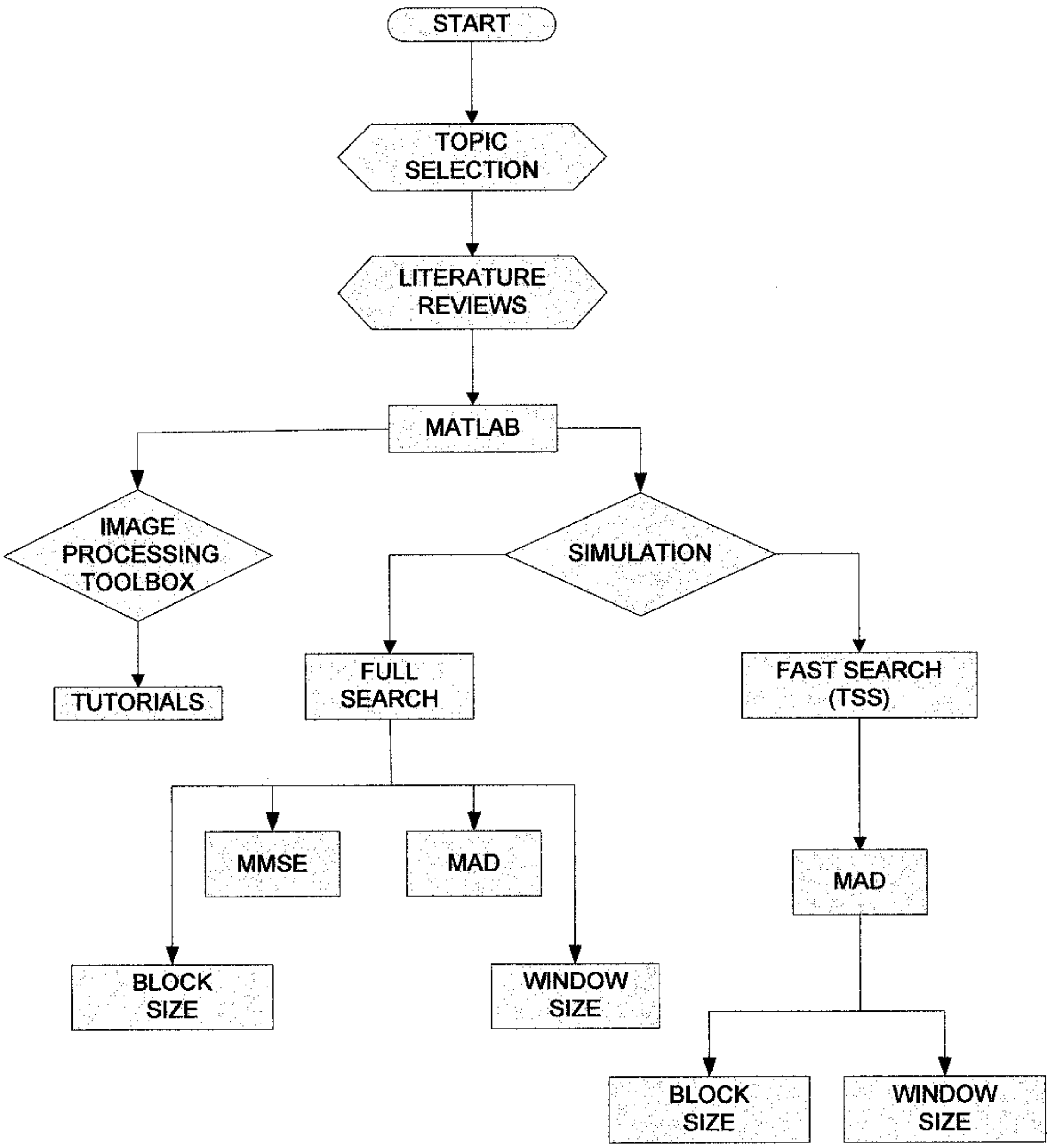


Figure 3.1: Flow chart of project procedure

For the first step of this Final Year Project, the author has selected a topic to be done. The topic selected is Video Coding: Comparison of Block Matching Techniques. Following that some literature reviews on the project background, scope and theories behind block matching algorithm have been done in order to understand the methodology and plan for further works.
The author has selected MATLAB for this project as this software can cater the purpose of this investigation and the author's need to understand further about image processing. Through its Image Processing Toolbox, tutorials have been done to familiarize with the functionality of the software and what image processing is all about. All of these stages have been done in the first semester for this project work, including the early simulation work of extracting YUV frames from some video sequences, as shown in the results.

For the second semester, further simulations have been done for the purpose of investigation. These simulations include investigation on Full Search technique with considerations on minimum mean square error (MMSE) and minimum mean absolute difference (MAD), which is the specification in this project investigation, block seizes and the search window. Some investigations on the Fast Search have also been done, although only one technique has been selected for the Fast Search due to the limited time frame. The technique selected is the Three Step Search. Taking the MAD as consideration based on several factors, the TSS will be investigated with consideration on block size and window size. The written program will be test and validate overtime along the project time frame.

3.2 TOOLS (EQUIPMENT, HARDWARE, ETC.) REQUIRED.

Some software, hardware and equipment have been used in this project. The descriptions are as shown in **Table 3.2**:

TOOLS	DESCRIPTION
Software	• MATLAB version 6.1: used for image reading and video testing and used for coding development • QCIF Play version 1.0: to play the video sequences retrieved
Hardware	• Personal computer • Server for internet (retrieving video samples and researches)
Equipment	• Video samples

Table 3.2: Tools used in the project

3.3 SIMULATION ALGORITHM

➊ Extract YUV frames from a given video sequences:

Clear all parameters and close all previous results

% Set parameters:

Set qcif video sequences to filename

Set amount of frames to frames

Set width of image to col

Set height of image to row

Set U and V frame ratio to UV_ratio

Set U and V width to col divide with UV_ratio;

Set U and V height to row divide with UV_ratio;

Set size of frame Y toY=zeros(row,col,frames);

% To open file:

Set Y size to col times by row;

Set U size to UV_col times by UV_row

Set V size to UV_col times by UV_row

Set buffer size to Y size + U size + V size;

Set file to be read

Set graphic handler for images

% To read and save each YUV frame

Loop:

For ith_frame equals to 1 until specified frame number (step size equals to 1)

Set read file with specified sizes and image class

Extract Y frame and reshape

Plot Y frame to the first row, first column

Show Y frame with colour mapped of 0 to 255, titled 'Y'

Write image to file and set filename with '<frame number>Y<video sequence>'

Pause

Extract U frame and reshape

Plot U frame to the first row, second column

Show U frame with colour mapped of 0 to 255, titled 'U'

Write image to file and set filename with '<frame number>U<video sequence>'

Pause

Extract V frame and reshape

Plot V frame to the first row, third column

Show V frame with colour mapped of 0 to 255, titled 'V'

Write image to file and set filename with '<frame number>V<video sequence>'

Set label to the subplots

Press any key for next frame numbers

End loop

❷ **Minimum Mean Square Error (MMSE)**

Clear all parameters and close all previous results

% Set parameters:

Loop: For loop equals to 2 increments by 1 to specified frame numbers

Read the anchor image of frame Y

Read the specified target image (numbered from loop)

Set block size to B

Set dimension of image block size to sz

Set search window to R

Find the anchor image dimension

Pad anchor image with zero to both height and width based on the search window

Start timer

%Set distinct block size on image

Loop1: For i equal to 1 increment by block size to height minus block size minus 1

Loop2: For j equal to 1 increment by block size to width minus block size minus

%j,k= column, i,l = row

Set every parameters to zero

%Set start pixel of every sliding block for row

Set m equal to 1

%Set search window (2R+1)

Loop3: For l equals to negative R, increment by 1, to R

%Set start pixel of every sliding block for column

Set n equals to1

Loop4: For k equals to negative R, increment by 1, to R

% Calculate MSE

Set displacement k to dx(n)

Set displacement l to dy(m)

Subtract anchor and target frame, with target slide one pixel in specified search window

Change class uint8 to double for square operation

Do square operation for MSE

Sum all MSE for one block

Store MSE

Increment n by 1

End loop4

Increment m by 1

End loop3

Find the lowest MSE value

%Write predicted image for MSE

Loop5: For p equals to 1, increment by 1, to 2 times R plus 1

Loop6: For q equals to 1, increment by 1, to 2 times R plus 1

If MSE in (p,q) is equal to B

Write block with lowest MSE value in anchor image to imp matrix

Set coordinates for distinct block

Record the estimated MV

End if

End loop6

End loop5

End loop2

End loop1

Set elapsed time to t(loop)

%Draw motion vector

Set motion vector grid for every block

Show predicted image, colour mapped from 1 to 255, entitled 'imp'

Hold on image

Set motion vector mapping

Hold off image

Write imp to file and set filename

❸ **Minimum Mean Absolute Difference (MAD)**

Clear all parameters and close all previous results

% Set parameters:

Loop: For loop equals to 2 increments by 1 to specified frame numbers

Read the anchor image of frame Y

Read the specified target image (numbered from loop)

Set block size to B

Set dimension of image block size to sz

Set search window to R

Find the anchor image dimension

Pad anchor image with zero to both height and width based on the search window

Start timer

%Set distinct block size on image

Loop1: For i equal to 1 increment by block size to height minus block size minus 1

Loop2: For j equal to 1 increment by block size to width minus block size minus

%j,k= column, i,l = row

Set every parameters to zero

%Set start pixel of every sliding block for row

Set m equal to 1

%Set search window (2R+1)

Loop3: For l equals to negative R, increment by 1, to R

%Set start pixel of every sliding block for column

Set n equals to1

Loop4: For k equals to negative R, increment by 1, to R

% Calculate MAD

Set displacement k to dx(n)

Set displacement l to dy(m)

Subtract anchor and target frame, with target slide one pixel in specified search window

Do absolute operation for MAD

Sum all MAD for one block

Store MAD

Increment n by 1

End loop4

Increment m by 1

End loop3

Find the lowest MAD value

%Write predicted image for MAD

Loop5: For p equals to 1, increment by 1, to 2 times R plus 1

Loop6: For q equals to 1, increment by 1, to 2 times R plus 1

If MAD in (p,q) is equal to B

Write block with lowest MAD value in anchor image to imp matrix

Set coordinates for distinct block

Record the estimated MV

End if

End loop6

End loop5

End loop2

End loop1

Set elapsed time to t(loop)

%Draw motion vector

Set motion vector grid for every block

Show predicted image, colour mapped from 1 to 255, entitled 'imp'

Hold on image

Set motion vector mapping

Hold off image

Write imp to file and set filename

❹ Three Step Search

Clear all parameters and close all previous results

% Set parameters:

Loop: For loop equals to 2 increments by 1 to specified frame numbers

Read the anchor image of frame Y

Read the specified target image (numbered from loop)

Set block size to B

Set dimension of image block size to sz

Set search window to R

Find the anchor image dimension

Set the window size to W

Set centre pixel of window to M

Pad anchor image with zero to both height and width based on the search window

Set number of steps to L

Start timer

%Set distinct block size on image

Loop1: For i equal to 1 increment by block size to height minus block size minus 1

Loop2: For j equal to 1 increment by block size to width minus block size minus

%j,k= column, i,l = row

Set every parameters to zero

%%%Start First Step%%%

Set nth to 1

Set step size for nth step to S

%Set start pixel of every sliding block for row

Set m equal to 1

%Set search window (2R+1)

Loop3: For l equals to negative R plus(R minus S), increment by S1, to R minus (R minus S)

%Set start pixel of every sliding block for column

Set n equals to1

Loop4: For k equals to negative R plus(R minus S), increment by S1, to R minus (R minus S)

% Calculate MAD

Set displacement k to dx(n)

Set displacement l to dy(m)

Subtract anchor and target frame, with target slide one pixel in specified search window

Do absolute operation for MAD

Sum all MAD for one block

Store MAD

Increment n by 1

End loop4

Increment m by 1

End loop3

Find the lowest MAD value

%Write predicted image for first step

Loop5: For p equals to 1, increment by 1, to 3

Loop6: For q equals to 1, increment by 1, to 3

If MAD in (p,q) is equal to B

Set a equals to dy(p);

Set b equals to dx(q);

End if

End loop6

End loop5

If (a equals to M) and (b equal to M)

Set MADblock equasl to MADfin(p,q);

Write predicted image for first step into file

Set coordinates for distinct block
Record the estimated MV
%%%End First Step%%%

Else
Set coordinates of the lowest MAD

%%%Start Second & Third Step%%%%
Loop 7: For nth equals to 2, increment by 1 to specified number of steps, L
Set step size for nth step to S
%Set start pixel of every sliding block for row
Set Pm equal to 1
%Set search window (2R+1)

Loop8: For Pl equals to negative R plus(R minus S), increment by S1, to R minus (R minus S)
%Set start pixel of every sliding block for column
Set Pn equals to1
Loop9: For Pk equals to negative R plus(R minus S), increment by S1, to R minus (R minus S)
% Calculate MAD
Set displacement Pk to Pdx(n)
Set displacement Pl to Pdy(m)
Subtract anchor and target frame, with target slide one pixel in specified search window
Do absolute operation for MAD
Sum all MAD for one block
Store MAD
Increment Pn by 1
End loop9
Increment Pm by 1
End loop8

Find the lowest MAD value

%Write predicted image for second and third step
Loop10: For Pp equals to P1, increment by 1, to 3
Loop11: For Pq equals to P1, increment by 1, to 3
If MAD in (Pp,Pq) is equal to B
Set MADblock equas to MADfin(Pp,Pq);
Write predicted image for third step into file
Set coordinates for distinct block
Record the estimated MV
End if
End loop11
End loop10

Reset value for nth step
End loop7
%%%End Second and Third Step%%%
End if
End loop1
End loop2
Stop timer
End loop frames

❸ Peak Signal-to-Noise Ratio

If target frame equals to predicted frame

Set error message 'Images are identical: PSNR has infinite value'

End if

Subtract target and predicted frame and set to diff

Square operation on diff and store to C

Calculate PSNR

Display PSNR value

CHAPTER 4

RESULTS AND DISCUSSION

4.1 RESULTS

4.1.1 Y, U and V Frame

From the simulation done using MATLAB, the author managed to extract the still images frames in the sample QCIF video and save them in JPEG format for analysis. The simulation is done for one hundred frames and can be increased according to requirements. Samples are taken from three video sequences with three different criterions;

I. Foreground movements, small motion sequences → *claire.qcif*

II. Background movements, moderate motion sequences → *news.qcif*

III. Foreground and background movements, large motion sequences → *carphone.qcif*

The sizes of U and V are actually halves of Y components. However, for the purpose of visual inspection in this report, the U and V images has been resized. The samples taken for Y, U and V frame are as shown in the figures for each video sequences. Factors considered to select each video sequence to their own criterion can be seen in **Section 4.2.2** and will be discussed in its discussion.

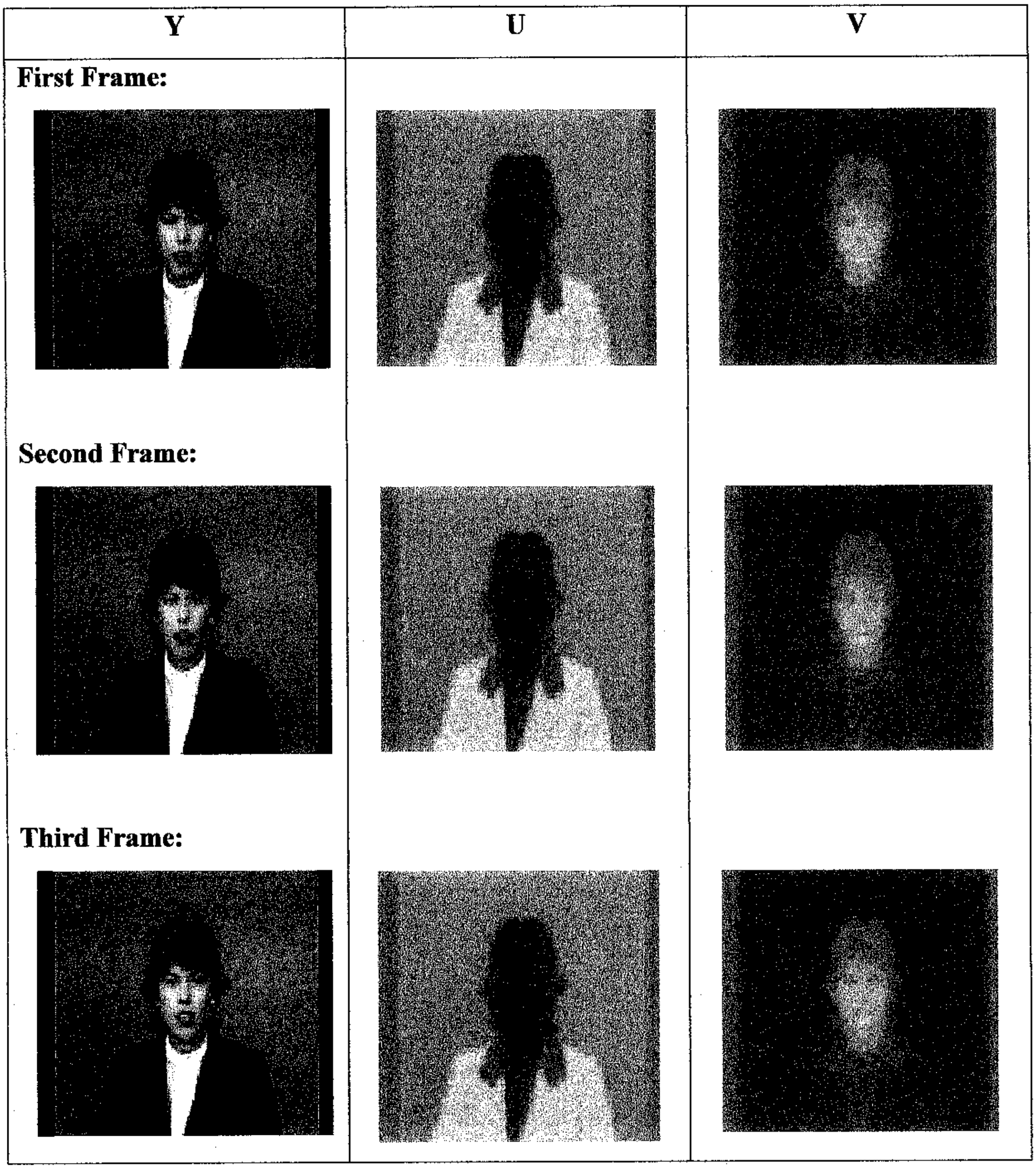


Figure 4.1.1(a): Video sequence for *claire.qcif*

The motion in this video only includes the anchorwoman's head and the expression of her face, which are small movement variations.

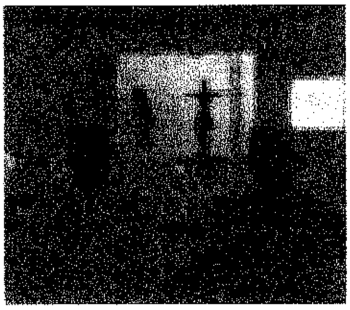
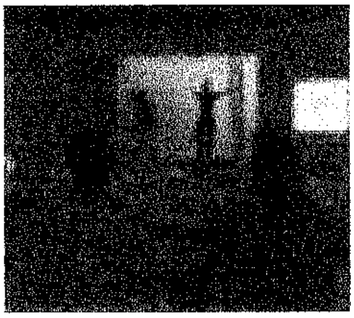
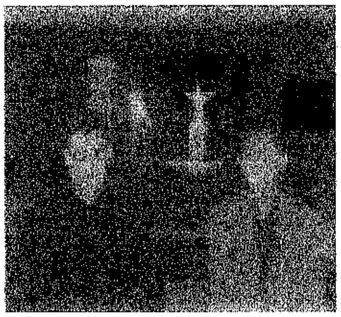
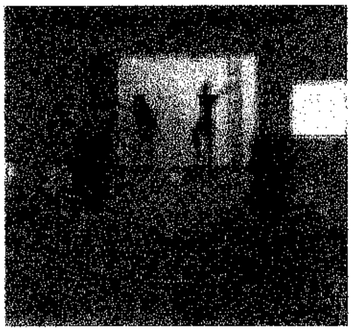
Y	U	V
First Frame: 		
Second Frame: 		
Third Frame: 		

Figure 4.1.1(b): Video sequence for *news.qcif*

The differences in these images are appears very small in the beginning sequences as the motion of the characters in the video did not vary much for each frame. However, as the frame number increase the motion can be seen more clearly. Although there are some movements from the newsreaders, the clearest difference can be seen at the background of the anchorman and anchorwoman, which is the image of a girl doing ballet.

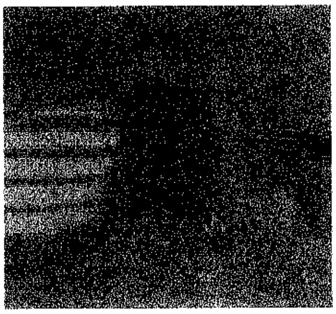
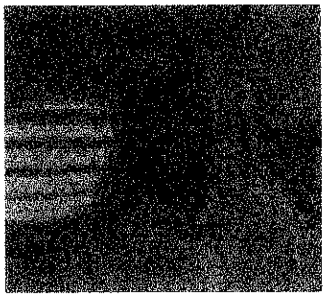
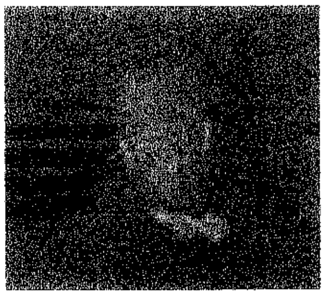
Y	U	V
First Frame: 		
Second Frame: 		
Third Frame: 		

Figure 4.1.1(c): Video sequence for *carphone.qcif*

Like *claire.qcif* video sequence, not much movement can be seen at the beginning frames of this video and more will come as the frame number increases. However, this video has a lot of movements, such as facial expression, body movements and the moving background as the car moves.

4.1.2 Comparison: Minimum Mean Square Error (MMSE) and Minimum Mean Absolute Difference (MAD)

The first comparison done for this project is between the MMSE and MAD images. Luminance images are used as they give a better visual quality compared to chrominance. The results are taken from two types of block sizes with the same search window size, W = 7.

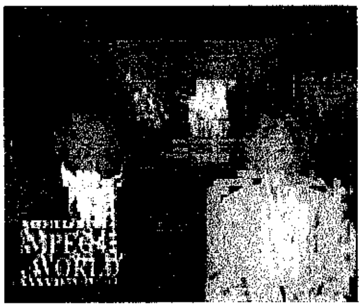
For *news.qcif*, the parameters are as shown in **Table 4.1** for Frame 1 and 8 from its video sequence. Frame 1 is chosen as the reference frame and Frame 8 is the current frame that wants to be predicted. The Peak Signal-to Noise Ratio (PSNR) is the result from the comparison of Frame 8 and its predicted image:

The sample images of one of the video sequences (*news.qcif*) are as shown below:

Frame 1 of *news.qcif*

Frame 8 of *news.qcif*

Figure 4.1.2(a): Original images



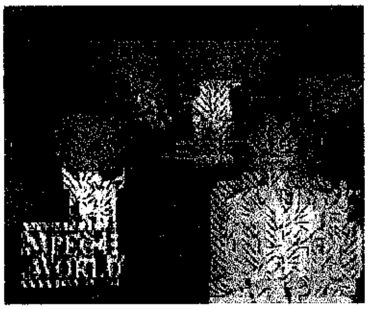
MMSE (B = 4, W = 7)

MAD (B = 4, W = 7)

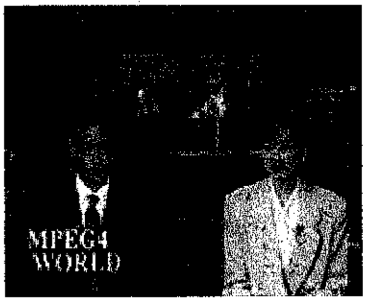
MMSE (B = 16, W = 7)

MAD (B = 16, W = 7)

Figure 4.1.2(b): Predicted images for MMSE and MAD



MMSE (B = 4, W = 7)

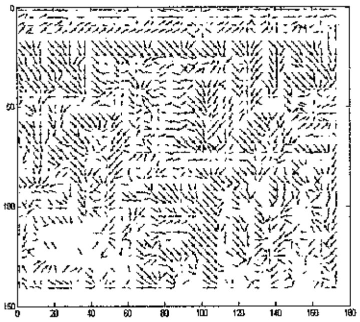


MAD (B = 4, W = 7)

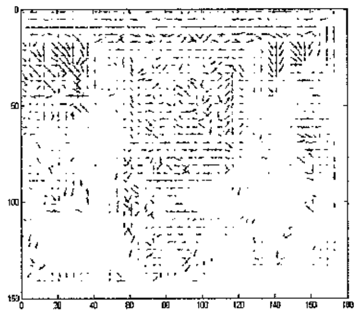
MMSE (B = 16, W = 7)

MAD (B = 16, W = 7)

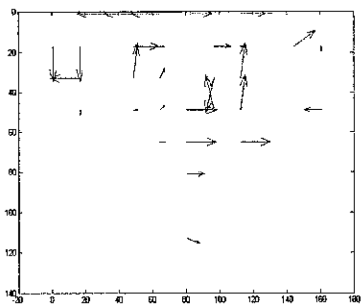
Figure 4.1.2(c): Predicted images for MMSE and MAD with motion vector overlay



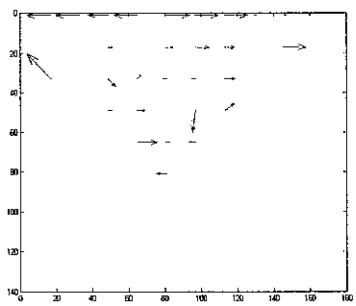
MMSE (B = 4, W = 7)



MAD (B = 4, W = 7)



MMSE (B = 16, W = 7)



MAD (B = 16, W = 7)

Figure 4.1.2(d): Motion vector for predicted images for MMSE and MAD

Graphs shown in figures are the result for the comparison of MMSE and MAD for each selected video sequence:

For *claire.qcif*:

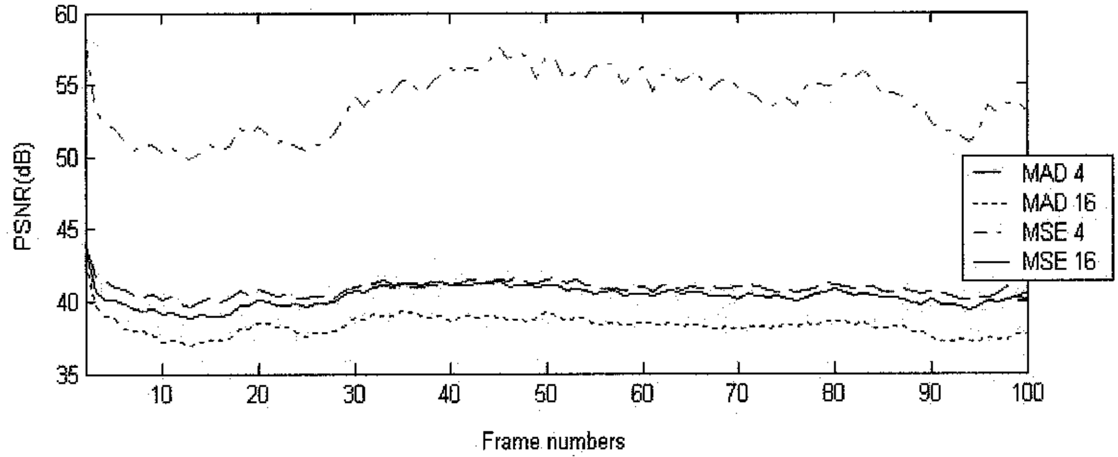


Figure 4.1.2(e): PSNR for *claire.qcif*

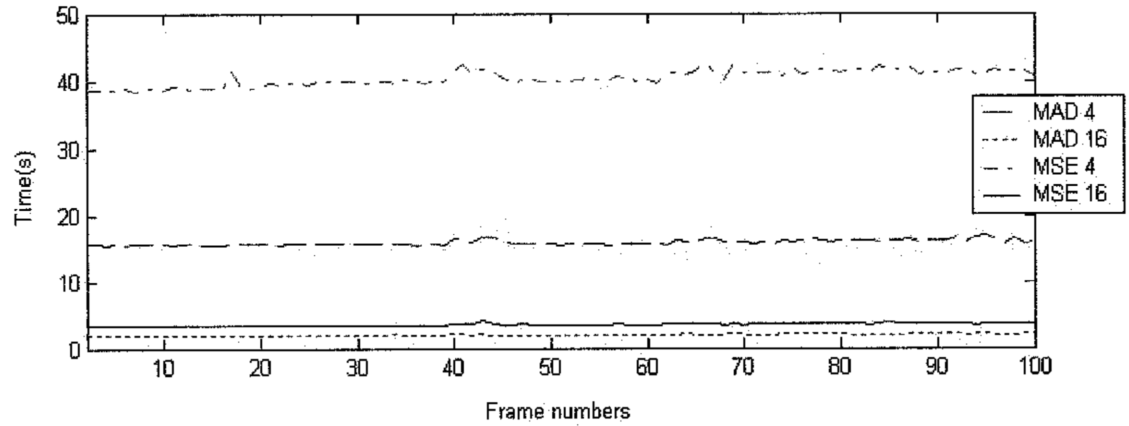


Figure 4.1.2(f): Elapsed time for claire.*qcif*

For *news.qcif*:

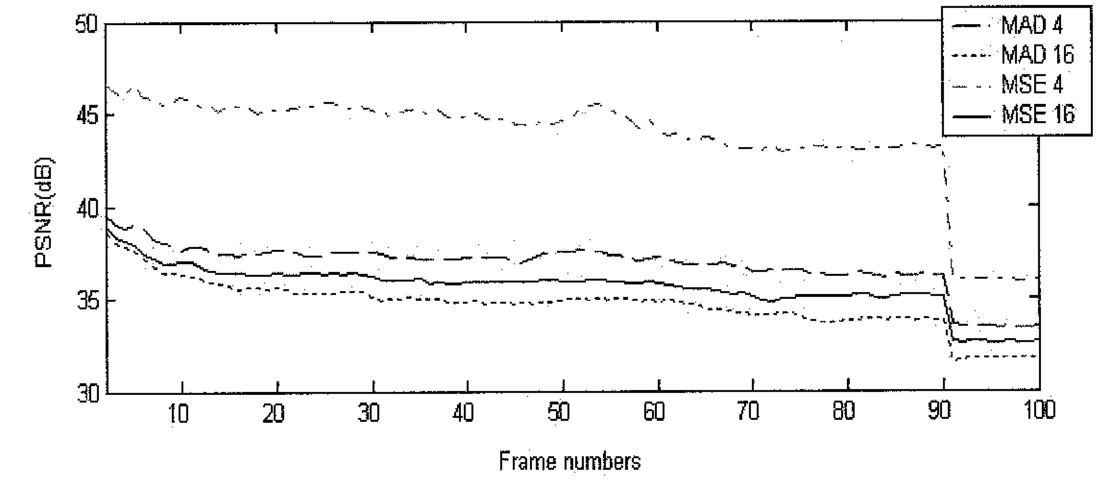


Figure 4.1.2(g): PSNR for *news.qcif*

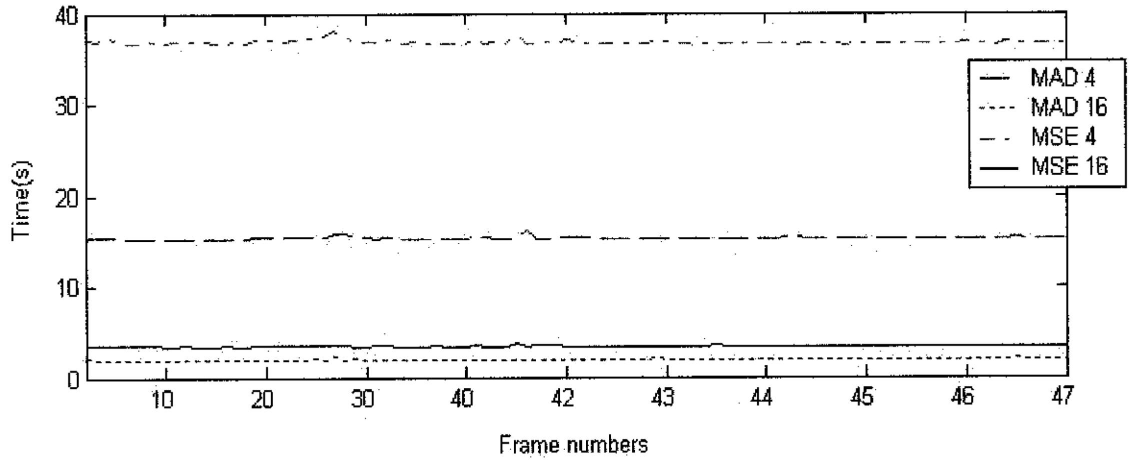


Figure 4.1.2(h): Elapsed time for *news.qcif*

For *carphone.qcif*:

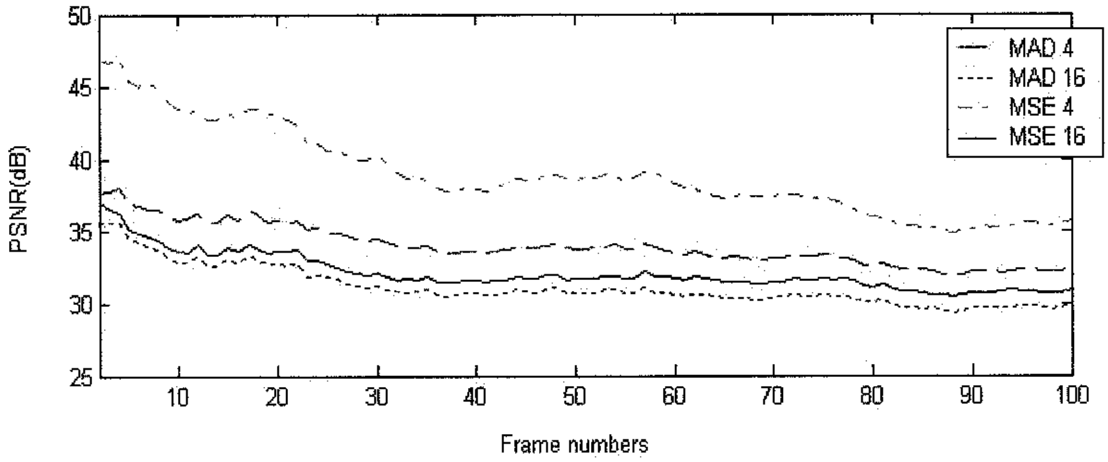


Figure 4.1.2(i): PSNR for *carphone.qcif*

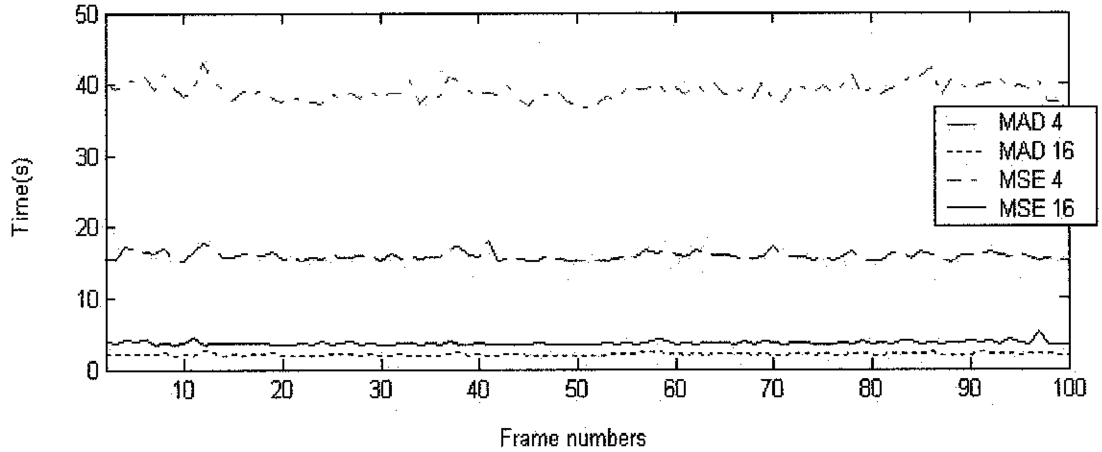


Figure 4.1.2(j): Elapsed time for *carphone.qcif*

4.1.3 Comparison: Effect of Block Sizes and Window Sizes in Full (Exhaustive) Search

The parameters considered for this section is the MAD only, with the reasoning based on discussion in **Section 4.2.1**. Point to be noted is that the block size and window sizes have same length on each side. The effect of the block size (B) and the search window size (W) are as shown in the tables and charts:

For *claire.qcif*:

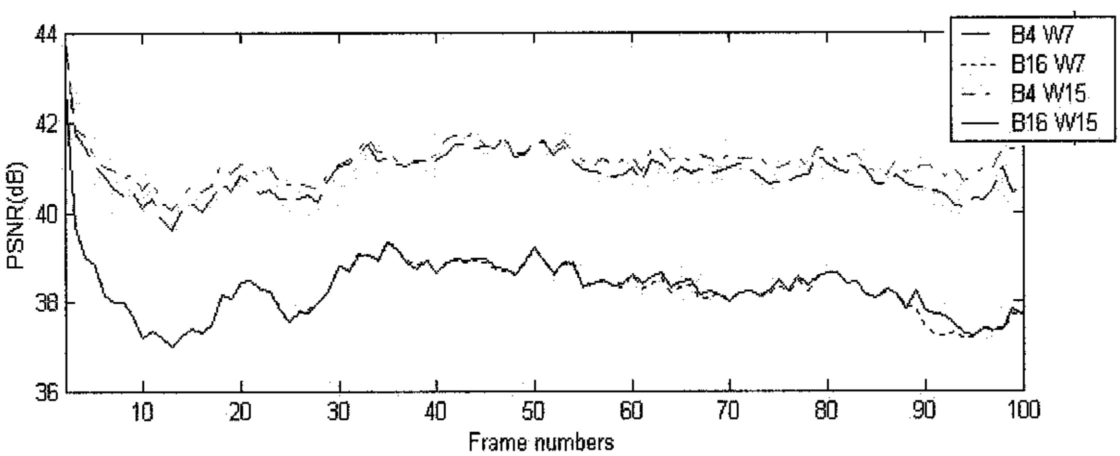


Figure 4.1.3(a): PSNR (dB) for *claire.qcif*

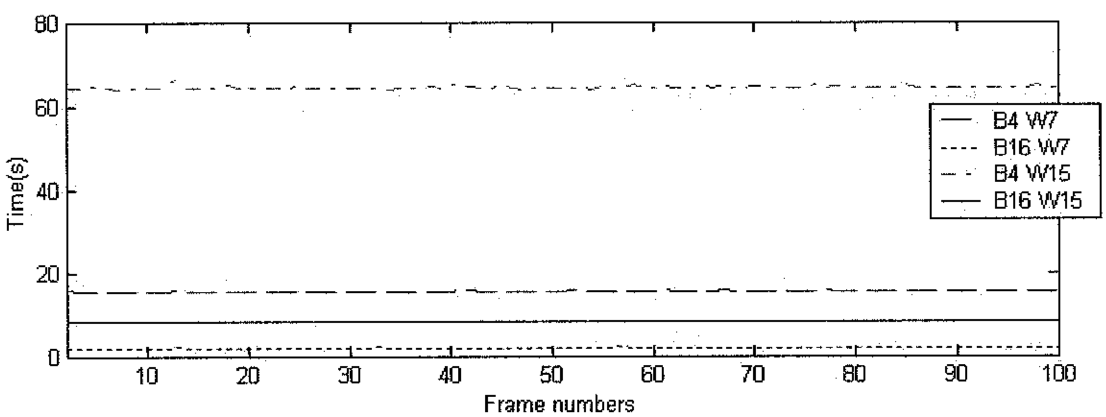


Figure 4.1.3(b): Elapsed Time (s) for *claire.qcif*

For news.qcif:

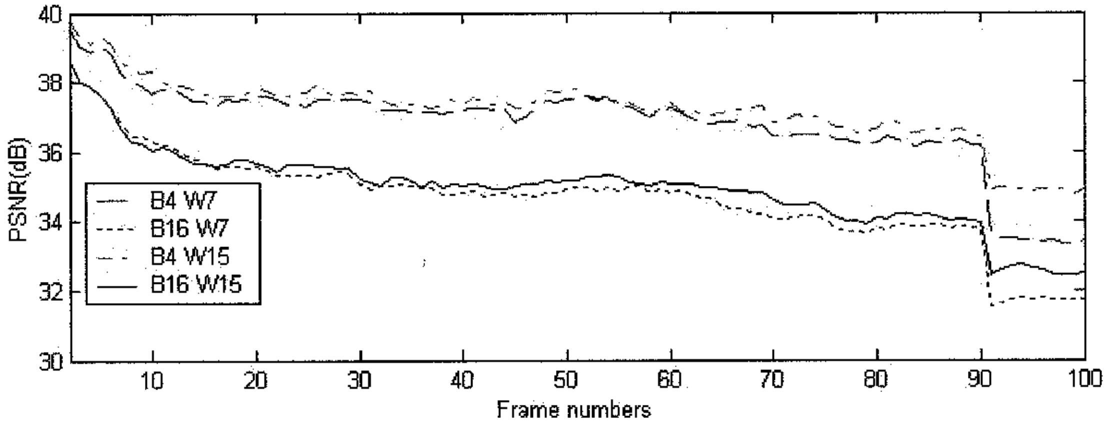


Figure 4.1.3(c): PSNR (dB) for *news.qcif*

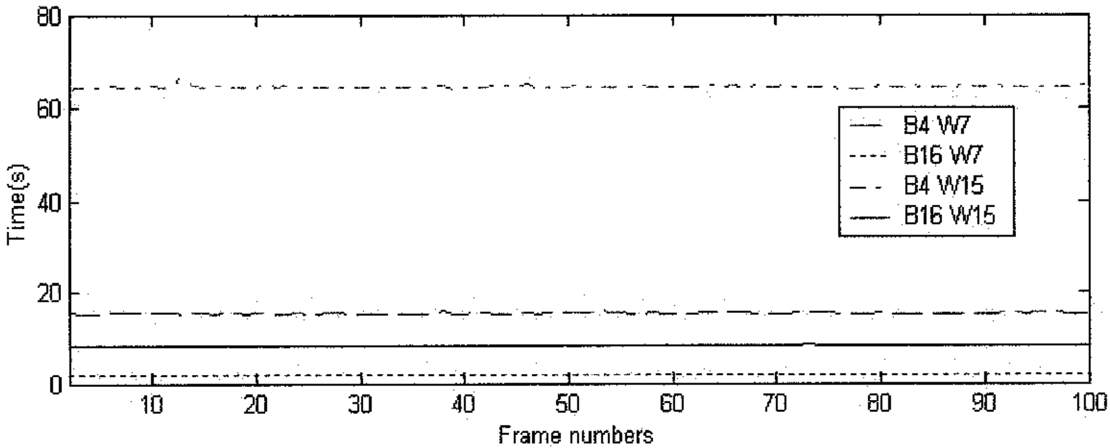


Figure 4.1.3(d): Elapsed Time (s) for *news.qcif*

For *carphone.qcif*:

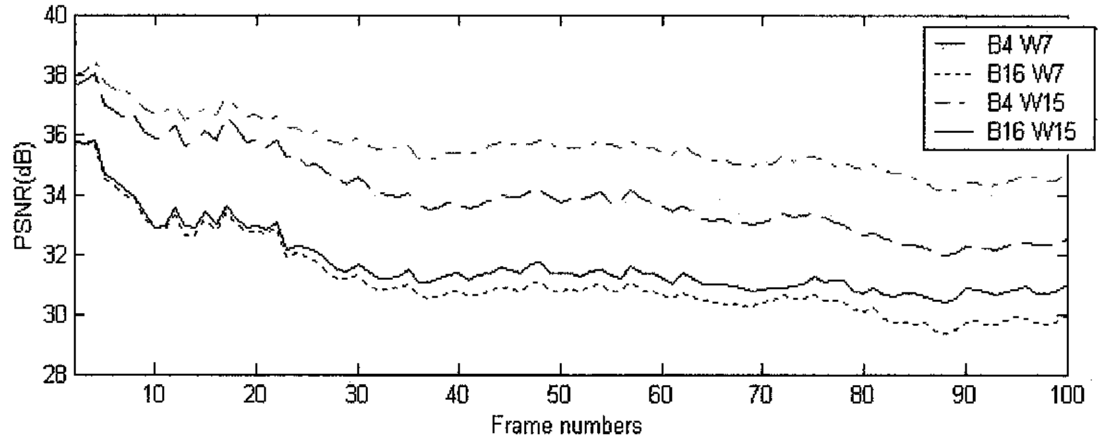


Figure 4.1.3(e): PSNR (dB) for *carphone.qcif*

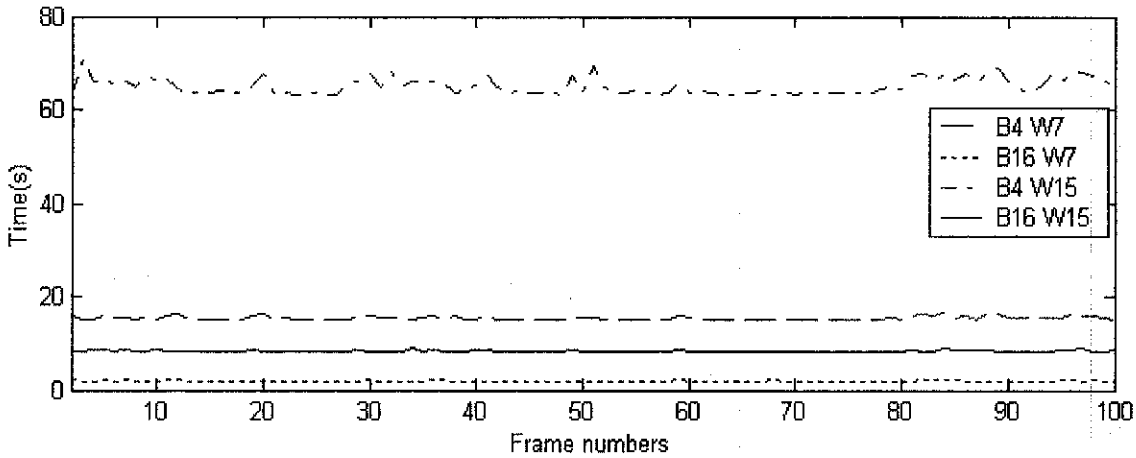


Figure 4.1.3(f): Elapsed Time (s) for *carphone.qcif*

4.1.4 Comparison: Fast Search (Three Step Search) on Block Sizes and Window Sizes

The sample image results of the simulation for TSS are as shown in the figures below. The samples are taken from *news.qcif* video sequence.

Figure 4.1.4(a): Original frame 10 image *news.qcif*

N = 4, W = 7

N = 16, W = 7

N = 4, W = 15

N = 16, W = 15

Figure 4.1.4(b): Predicted images with TSS *news.qcif*

The effect of the block size (N) and the search window size are as shown in the tables and charts:

For *claire.qcif*:

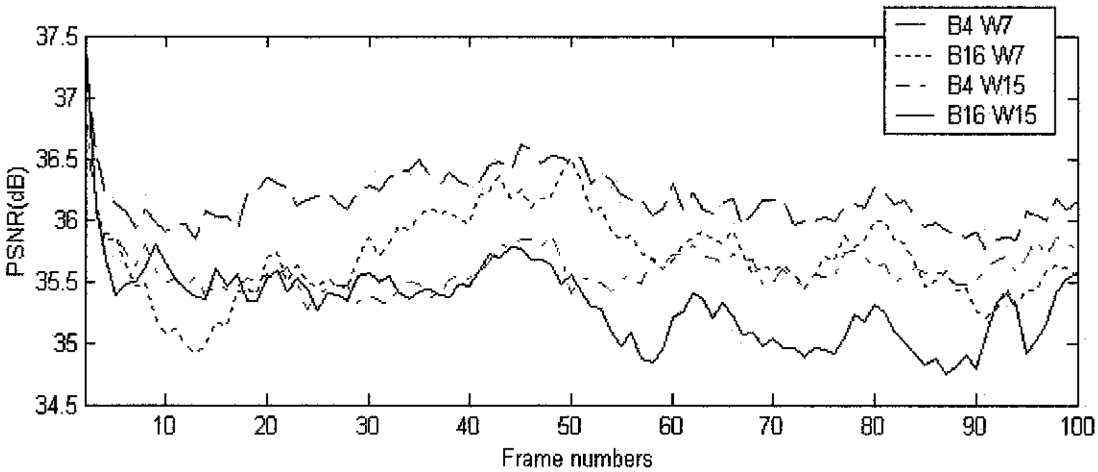


Figure 4.1.4(c): PSNR (dB) for *claire.qcif* (TSS)

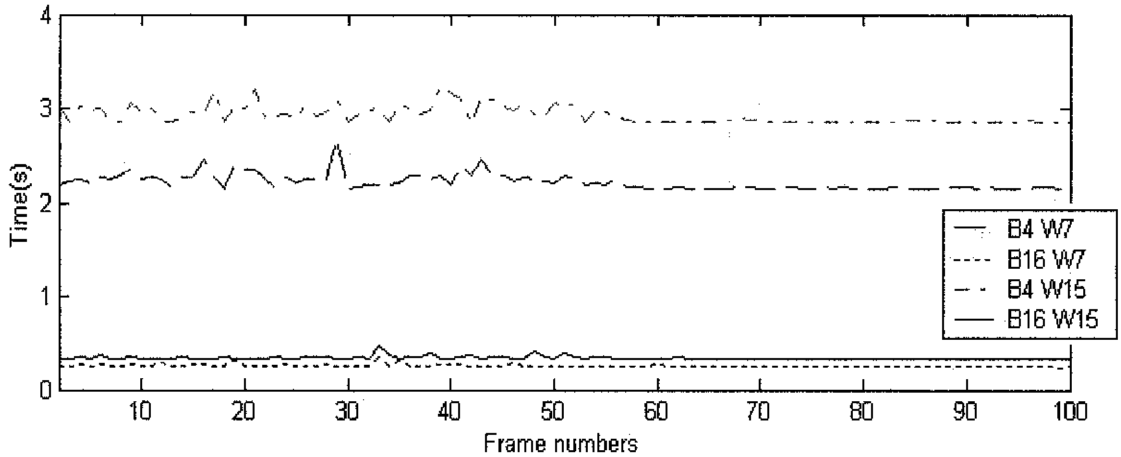


Figure 4.1.4(d): Elapsed Time (s) for *claire.qcif* (TSS)

For *news.qcif*:

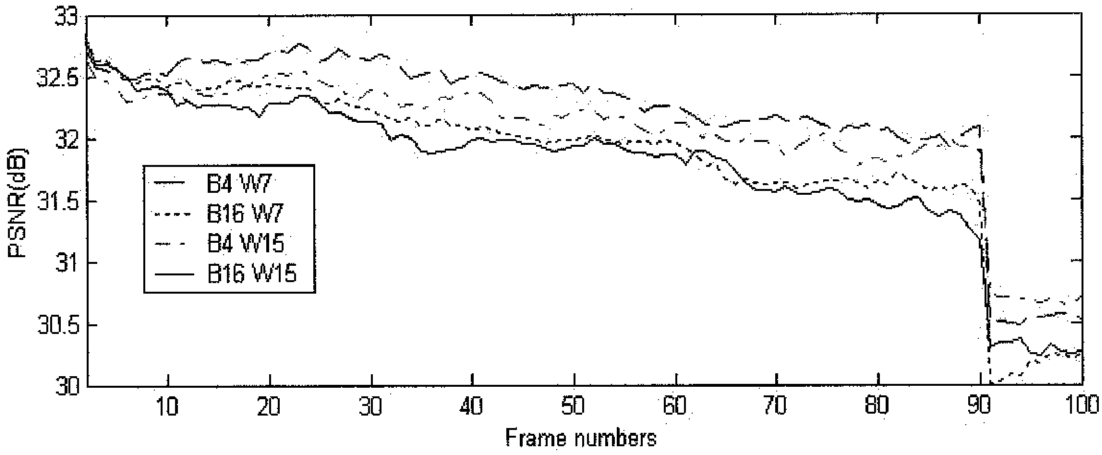


Figure 4.1.4(e): PSNR (dB) for *news.qcif* (TSS)

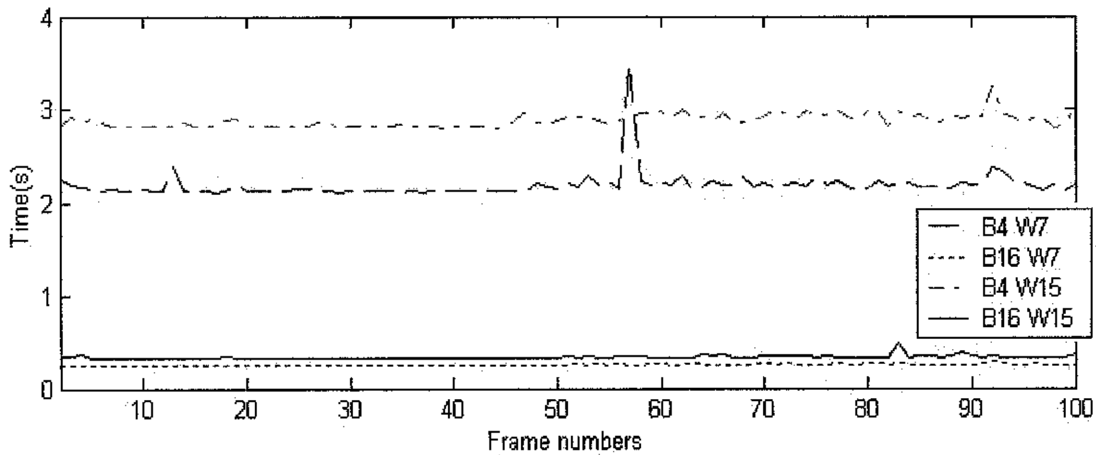


Figure 4.1.4(f): Elapsed Time (s) for *news.qcif* (TSS)

For *carphone.qcif*:

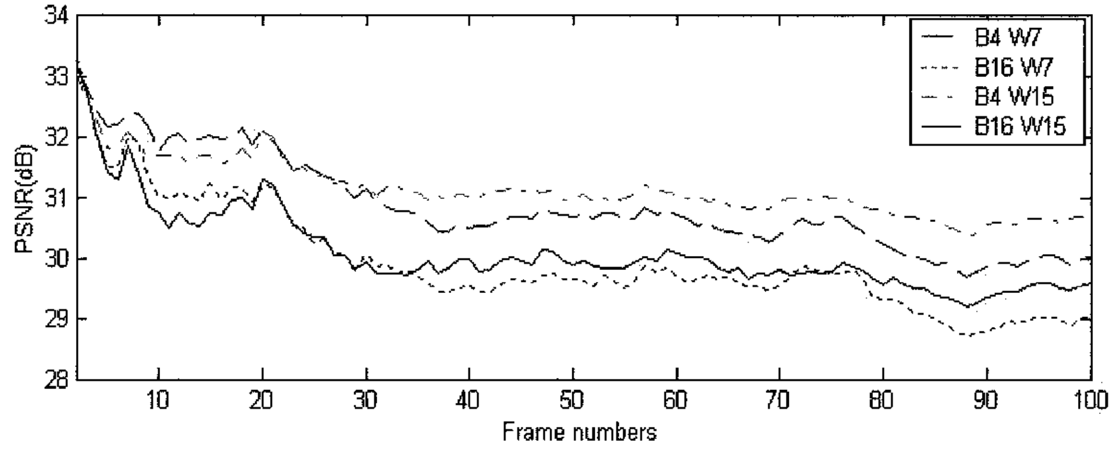


Figure 4.1.4(g): PSNR (dB) for *carphone.qcif* (TSS)

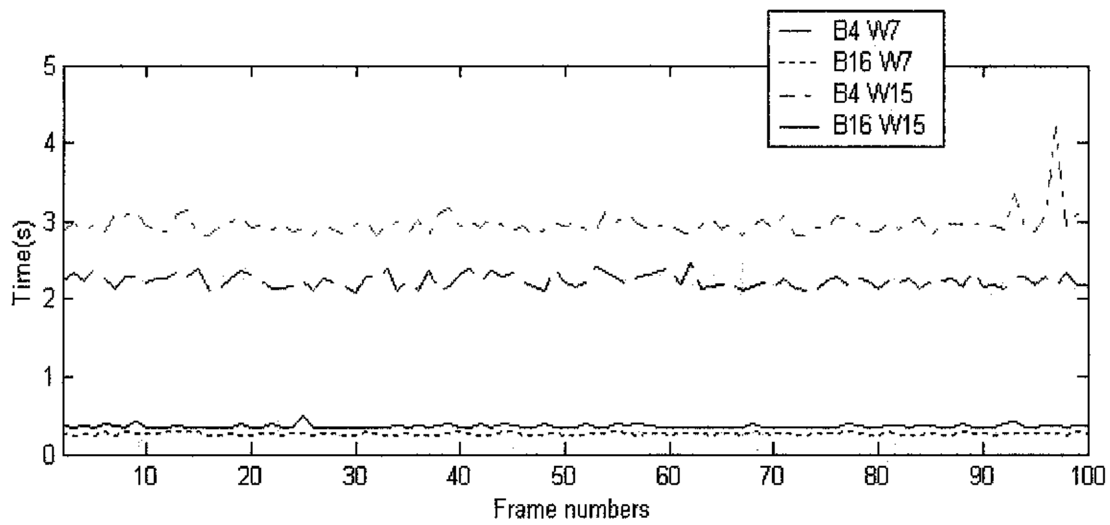


Figure 4.1.4(h): Elapsed Time (s) for *carphone.qcif* (TSS)

Note: the image results for *claire.qcif* and *carphone.qcif* can be found in **APPENDIX J** and **APPENDIX K** respectively.

4.1.5 Comparison: Fast Search (Three Step Search) and Full (Exhaustive) Search

The comparisons were done on the best PSNR values of MAD for each algorithm as they gave the clearest PSNR values and easier to compare:

Parameters	**claire.qcif**		**news.qcif**		**carphone.qcif**	
Algorithm	**FS**	**TSS**	**FS**	**TSS**	**FS**	**TSS**
Block	B = 4	B = 4	B = 4	B = 4	B = 4	B = 4
Window	W = 15	W = 7	W = 15	W = 7	W = 15	W = 15
PSNR (dB)	40.0 ↓ 44.0	35.5 ↓ 37.5	34.3 ↓ 40.0	30.5 ↓ 33.0	36.0 ↓ 38.2	30.7 ↓ 33.2
Time (s)	60.2 ↓ 60.3	2.0 ↓ 2.5	60.2 ↓ 60.3	2.0 ↓ 3.5	60.2 ↓ 60.5	2.0 ↓ 4.2

Table 4.1.5: Comparison of FS and TSS for the best PSNR values

The graphs for PSNR and elapsed time for all video sequences are as show in these figures:

For *claire.qcif*:

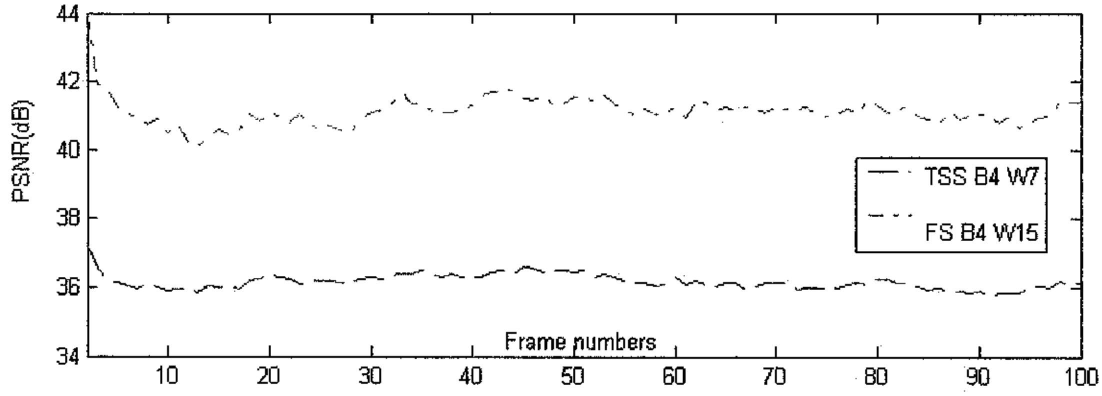


Figure 4.1.5(a): PSNR (dB) for *claire.qcif* (FS andTSS)

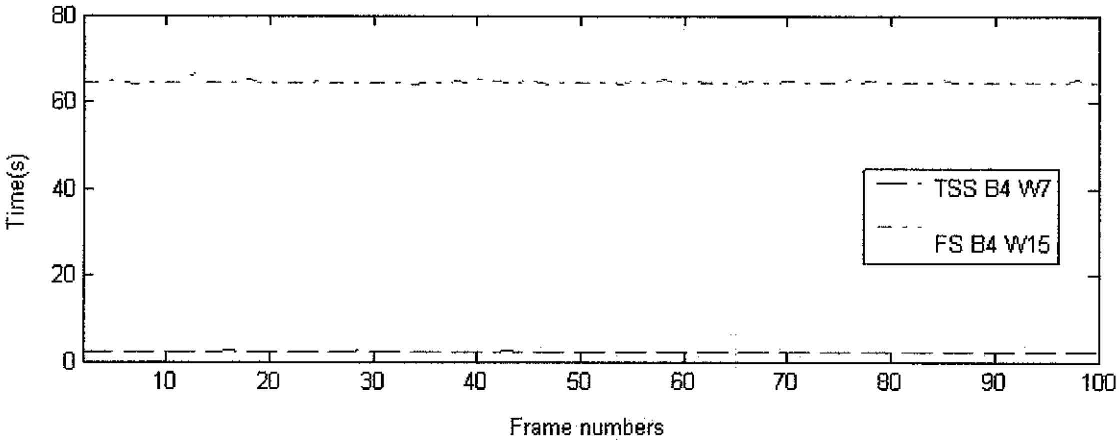


Figure 4.1.5(b): Elapsed Time (s) for *claire.qcif* (FS andTSS)

For *news.qcif*:

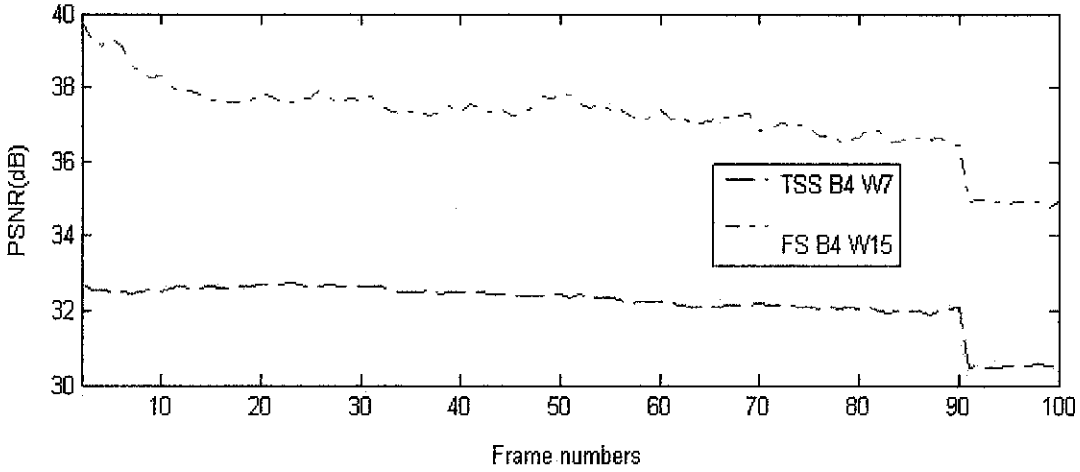


Figure 4.1.5(c): PSNR (dB) for *news.qcif* (FS andTSS)

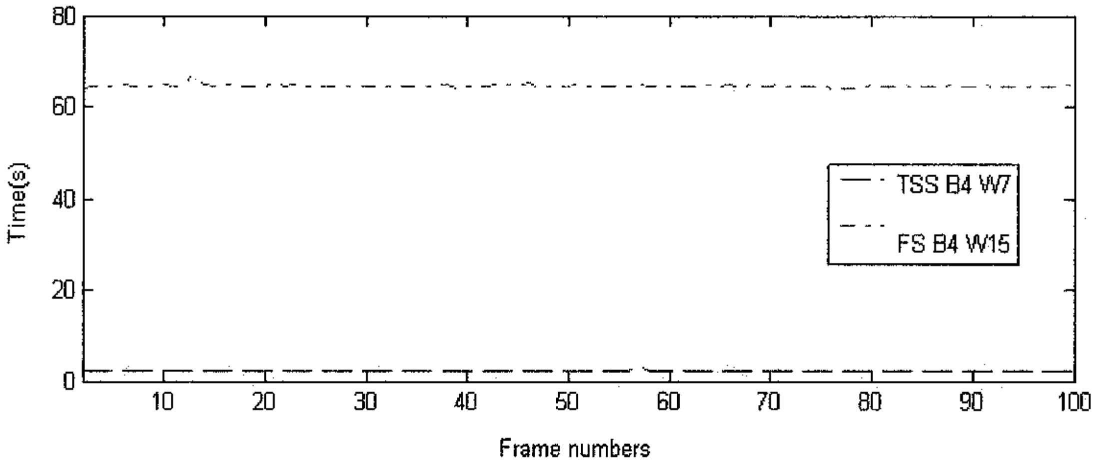


Figure 4.1.5(d): Elapsed Time (s) for *news.qcif* (FS andTSS)

For *carphone.qcif*:

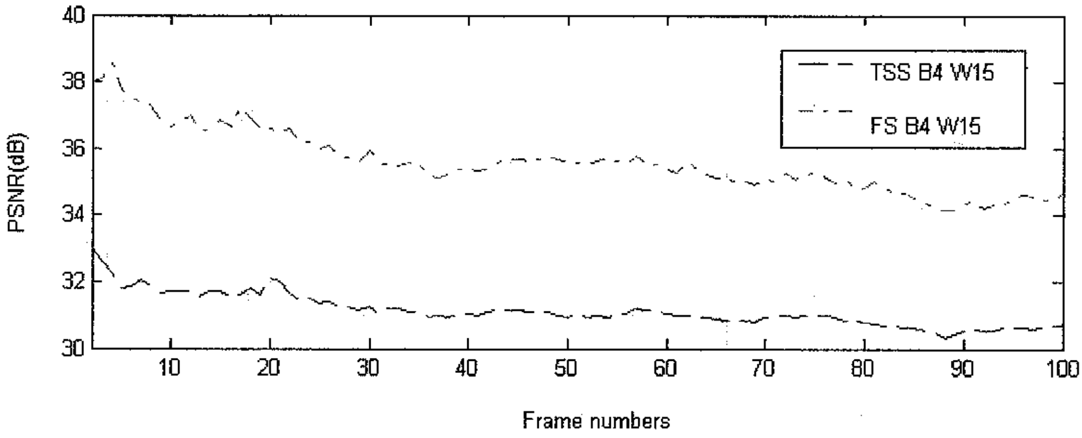


Figure 4.1.5(e): PSNR (dB) for *carphone.qcif* (FS andTSS)

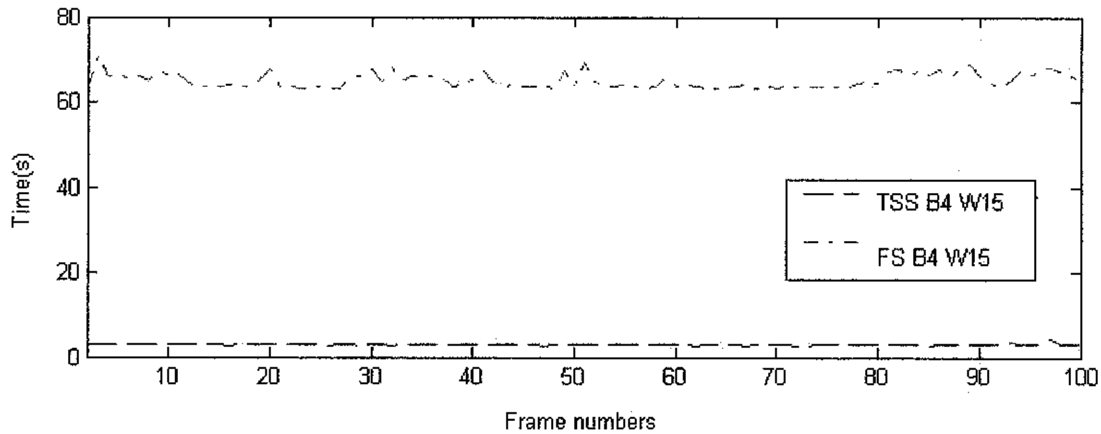


Figure 4.1.5(f): Elapsed Time (s) for *carphone.qcif* (FS andTSS)

4.2 DISCUSSIONS

4.2.1 Y, U and V Frame

As the images have been colour mapped from 0 to 255 colour values in the simulation, the images appear as greyscales images. For the comparisons in becoming investigations, only Y frame will be used as it contains the luminance or the perceived intensity of light. Human eyes have greater sensitivity to luminance rather than chrominance factors of U and V frames. This will ease the visual inspection of the predicted frames results.

4.2.2 Comparison: Minimum Mean Square Error (MMSE) and Minimum Mean Absolute Difference (MAD)

For this comparison purpose, the window size of 7 and block size of 4 and 16 are chosen and the comparison is done for both MAD and MMSE. From the graphs, it can be seen that the PSNR values are high for MMSE compared to MAD. This is suppose to indicate that the quality of predictions of MMSE are better than MAD. However, MMSE is not commonly used in actual implementation due to its square operation that adds to inefficiency of the process.

As shown in the graphs for elapsed time, MAD shows a better response time, nearly two times or less than MMSE. Based on these reasoning, the author decided to take MAD as the benchmark for further comparisons.

The decision to characterize the video sequences to three different criterions is based on the factors shown in the PSNR graphs. For this purpose, we can see at MMSE that has the block size of 4 at it shows the clearest visualization.

For *claire.qcif*, although there's much variability in the motion, the PSNR values are really big, ranging from 50 dB to 60 dB, with only 10 dB displacements. This has puts it in small motion sequences criterion.

For *news.qcif* on the other hand, although at the first the range of motion seems to be small, at the last frame numbers it shows a great displacement, from over 45 dB to over 35 dB, making it range from 35 dB and 50 dB. And considering that the maximum possible value of its PSNR is nearly 50 dB, the PSNR values of this video sequences are lower than *claire.qcif* and the author puts it in the moderate motion sequences.
Finally, for *carphone.qcif*, the PSNR shows a great displacement, from 50 dB to 35 dB throughout the frame numbers. The overall PSNR values, like *news.qcif*, are lower than *claire.qcif*. This has made it the suitable candidate for large motion sequences criterion.

4.2.3 Comparison: Effect of Block Sizes and Window Sizes in Full (Exhaustive) Search

The parameters are taken for MAD only, as discussed in **Section 4.2.2**. Block sizes and window sizes have significant effect on processing the predicted image. Point to be noted is that the block size and window sizes have same length on each side. From the investigation and figures in **Section 4.1.3**, we can see the effects on the three different criterions of video sequences:

◈ **Small Motion Sequence → *claire.qcif***

» **Block Size (B = 4 and B = 16)**

For block size equals to 4, it holds the highest PSNR values for both windows. This is because the smaller the block size, more accuracy in comparing the pixels will be obtained.
However, smaller block size also means that the higher search points will have to be done for comparison. Thus time taken for block size of 4 is higher than block size 16.

» **Window Size (W = 7 and W = 15)**

Window size has more significant effect on B = 4 rather than B = 16 for the small motion video sequence. For B = 4 with W = 15, the PSNR values are higher than B = 4 with W = 7, probably because the larger search window means higher amount of search points can be done to search for the best possible search candidate. The PSNR values for B = 16 for both search windows are very much the same, because the motion sequences are very small, they hardly can be detected in large block sizes. The whole moving image might be contained in one block.

The time taken for B = 4 with W = 15 holds the highest value. This is because the smaller the block size means higher search counts and larger search region for each block size.

Ranges of the best PSNR (B = 4, W = 15):

- PSNR = 40 dB to 44 dB
- Time = 60.2 s to 60.3 s

◈ **Moderate Motion Sequence → *news.qcif***

» **Block Size (B = 4 and B = 16)**

For block size equals to 4, it holds the highest PSNR values for both windows. This is because the smaller the block size, more accuracy in comparing the pixels will be obtained. This also can be seen from the visual inspection on **Figure 4.1.2(b)**. This is because from the motion vector overlay that can be seen from **Figure 4.1.2(c)**, we can see that the B = 16 block actually holds the whole image of the ballet woman in the background. As the comparison is done between pixels in each block, there are no great differences when it is processed compare to when they are process with B = 4.

However, smaller block size also means that the higher search points will have to be done for comparison. Thus time taken for block size of 4 is higher than block size 16.

» **Window Size (W = 7 and W = 15)**

For this motion sequence, the W has more significant effect as compared to small motion sequence video. This can be deducted as it has effect block size of 16. B = 16 with W = 15 is seen to perform better in the increasing frame number than the one with W = 7. this is also true for B = 4 because the higher value the frame number is, the larger motion sequence can be found compared to lower frame number in *news.qcif.*
Time taken for B = 4 with W = 15 still holds the highest value while B = 16 with W = 7 is still the lowest value.

Ranges of the best PSNR (B = 4, W = 15):

- PSNR = 34.3 dB to 40 dB
- Time = 60.2 s to 60.3 s

◈ **Large Motion Sequence → *carphone.qcif***

» **Block Size (B = 4 and B = 16)**

The impact of block size can be seen clearly in large motion video sequence. However, its highest PSNR value is lower than *claire.qcif* and *news.qcif* as it has much more motions than those two video sequences. B = 4 still have higher PSNR values than B= 16.
However, smaller block size also means that the higher search points will have to be done for comparison. Thus time taken for block size of 4 is higher than block size 16.

» **Window Size (W = 7 and W = 15)**

Apparently, B = 4 with W = 15 gives higher value than B = 4 with W = 7. For this sequence, the differences between the search windows of B = 16 can seen more clearly. This means that the larger the motion sequences, the higher impact of search window on the results.

The time taken for B = 4 with W = 15 is still the highest like other motion sequences and B = 16 with W = 7 still holds the lowest time consumption.

Ranges of the best PSNR (B = 4, W = 15):

- PSNR = 36 dB to 38.2 dB
- Time = 60.2 s to 60.5 s

4.2.4 Comparison: Fast Search (Three Step Search) on Block Sizes and Window Sizes

◈ **Small Motion Sequence → *claire.qcif***

For this motion sequences, the window size gave a very clear impact on its PSNR performance, rather than the block size only. This can be seen on the performance of B = 4 with W = 7 and B = 16 with W = 7 compared to their respective block sizes, with different search windows. B = 4 with W = 15 gave a very poor performance on this simulation as it has lower values for most of the frames even when compared to B = 16 with W = 7, which has always holds the lowest values in the Full search algorithm.

B = 4 with W = 15 still holds the highest time consumption than other parameters, followed closely by B = 4 with W = 7. B =16 block sizes still maintain the lower ones.

Ranges of the best PSNR (B = 4, W = 7):

- PSNR = 35.5 dB to 37.5 dB
- Time = 2 s to 2.5 s

◈ **Moderate Motion Sequence → *news.qcif***

B = 4 with W = 7 still holds the highest PSNR values and followed this time is followed closely by B = 4 with W = 15. B = 16 with W = 7 is still higher than B = 16 with W = 15 for most of the frames.

B = 4 with W = 15 still holds the highest time consumption than other parameters, followed closely by B = 4 with W = 7. B =16 block sizes still maintain the lower ones.

Ranges of the best PSNR (B = 4, W = 7):

- PSNR = 30.5 dB to 33 dB
- Time = 2 s to 3.5 s

◈ **Large Motion Sequence → *carphone.qcif***

For this large motion video sequence, block size with larger window holds the higher value compared to smaller windows, for most of the frames.

B = 4 with W = 15 still holds the highest time consumption than other parameters, followed closely by B = 4 with W = 7. B =16 block sizes still maintain the lower ones.

Ranges of the best PSNR (B = 4, W = 15):

- PSNR = 30.7 dB to 33.2 dB
- Time = 3 s to 4.2 s

4.2.5 Comparison: Fast Search (Three Step Search) and Full Search (FS)

As expected, as a Fast Search technique, the time consumption for TSS is many times better than FS. This has, however, cost its performance. The checking points for Full Search is 225 form 15 x 15, for W = 7 and 961 for 31 x 31 for W = 15. The checking points for TSS is 25 points for W = 7 and 33 from W = 15. This results from the number of step taken that's increase as the window increase.

For the first step, the checking points are nine and the other step after it has 8 check points. For W = 7, the number of steps taken are 3 steps while for W = 15, the number of steps taken are 4 steps. These actually decrease the time taken for the process, with the speed up ratio of 9 from 225 / 25 for W = 7 and 29.1 for W = 15.

The PSNR value for the TSS is lower than the Full Search results. This is foreseen as the Full Search compared each pixels for the block in the search region where as TSS only compare for lower amount of points. However, for the applications that needs speed as the main factor rather than quality, TSS is not a bad compromise, as the values are not so far from FS, compared to reduction of time it can provide.

FS is still the better option if higher quality compression is needed.

CHAPTER 5
CONCLUSION AND RECOMMENDATIONS

5.1 CONCLUSION

From the first investigation, the result shows that MMSE gives higher PSNR values than MAD but have higher time consumption. Realistically, MAD gives a better performance with its lower computation time and acceptable quality.

In the second investigation, the result shows that for MAD, with B = 4 and W = 15 gave the best options if quality is needed most. If the speed is the most important factor, B = 16 with W = 15 and W = 7 holds nearly the same computation time. However, the best solutions for the Full Search (FS) algorithm is B = 16 with W = 7 as its PSNR values is near to B = 4 with W = 15.

The third investigation shows that in Three Step Search (TSS), a higher window size for small block size will result to reduced performance. This is completely reverse to FS. This is because for a small block size in a large window size, there will be more displacement between search points. This will result to reduced comparative accuracy, especially to the smaller motion sequences video, as their image displacements are small compared to the search point distance.

This project also investigates the significant effect of FS and the TSS as one of the Fast Search mechanism. The results shown that FS has effectively obtained higher PSNR values but also very time consuming, compared to a very large speed up ratio of 9 by TSS.

5.2 RECOMMENDATIONS

Although Full (Exhaustive) Search gives the best PSNR values and highest quality for block matching techniques, it has its own time constraints. This is not a good prospect as one of the objectives of video compression is to cope with the transmission rate of data in the fast and advance technology.

Thus, Fast Search seems like a better option. Although in this project only the Three Step Search (TSS) is considered, there are many more techniques that have been evolved from this technique and have some improved criterion. Some of them that can be employed are:

❶ **Fast Three Step Search (FTSS)[4]**

- → Similar to TSS but have refined results.
- → Performs well with low complexity video, which is commonly used in video conferencing
- → Faster than TSS and does not have the worst penalty on search cases
- → Works well with "head and shoulder" type of video

❷ **Simple and Efficient Search Algorithm (SES)[5]**

- → Speed up TSS by factor of two but still maintain regularity and parallelism
- → Provides good performance comparable to TSS in terms of PSNR and MMSE

❸ **Efficient Three Step Search (E3SS)[6]**

- → Employs a small diamond checking pattern
- → Have less search points than TSS, means higher speed
- → Performs better than TSS in MMSE for moderate to large motion (for both small and large window)
- → Suitable for wide range of video application

REFERENCES

❁ **Books and Journals:**

[1] Murat Tekalp; 1995; Digital Video Processing; Prentice Hall Signal Processing Series

[2] Dzung Tien Hoang, Jeffery Scott Witter; 2002; Efficient Algorithms for MPEG Video Compression; Wiley Series in Telecommunications and Signal Processing

[3] Joan L. Mitchell, William B. Pennebaker, Chad E. Fogg, Didier J. LeGall; 1996; MPEG Video Compression Standard; Kluwer Academic Publishers

[4] Krit Panusope and Xuemin Chen; 1999; A Fast Three Step Search Motion Estimation Algorithm for Video Conferencing; IEEE; General Instrument Corporation, Advanced Technology Department, US

[5] Jianhua Lu and Ming L. Liou; 1997; A Simple and Efficient Search algorithm for Block-matching Motion Estimation; IEEE Transactions on Circuits and Systems for Video Technology

[6] Xuan Jing and Lap-Pui Chau; Senior Member; 2004; An Efficient Three Step Search Algorithm for Block Motion Estimation; IEEE Transactions on Multimedia

[7] Fractal Image Compression by Yuval Fisher et al., (Springer Verlag, 1995), section 2.4, "Pixelized Data".

❁ Websites:

[8] General information on digital image processing:

- www.thefreedictionary.com
- www.Wikipedia.org
- www.mpeg.org/MPEG/video.html

[9] IEEE Homepage:

- www.ieeexplore.ieee.org

[10] MATLAB Homepage:

- www.mathworks.com

[11] YUV Video Sequences Homepage:

- http://trace.eas.asu.edu/yuv/qcif.html

[12] QCIF Play Software Homepage:

- http://www.ifp.uiuc.edu/~lqian/Research/Software

❁ Documentation and Tools:

[13] MATLAB Image Processing Toolbox

[14] QCIF Play for 32-bit Microsoft Windows version 1.0 Copyright© 1996; Multimedia Communications and Networking Lab; Texas A&M University

APPENDICES

APPENDIX A1

Milestone for the First Semester of 2 Semester Final Year Project

No.	Detail/Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1.	**Selection of Project Topic**														
	-Propose Topic														
	-Topic being assigned														
2.	**Preliminary Research Work**														
	Video Compression														
	** Logbook-Week 3 (9/2)														
	** Preliminary Report (13/2)														
	Video Format														
	** Logbook-Week 4 (16/2)														
	Video Components and Standards														
	** Logbook-Week 5 (23/2)														
3.	**Project Work**														
	Image Processing Toolbox														
	** Logbook-Week 6 (1/3)														
4.	**Project Work Continue**														
	- Project Simulation & Corrections-MATLAB														
	** Logbook-Week 7 (15/3)														
	** Progress Report (19/3)														
5.	**Analysis of Project Output**														
	** Logbook-Week 8 (22/3)														
	** Logbook-Week 9 (29/3)														
	** Logbook-Week 10 (5/4)														
6.	**Reports**														
	** Draft of Interim Report (12/4)														
	** Interim Report (28/4)														
7.	**Oral Presentation (5/7 through 7/7)**														

Process

APPENDIX A2

Milestone for the Second Semester of 2 Semester Final Year Project

No.	Detail/ Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	**Project Work Continue**														
	Project Simulation & Corrections-MATLAB														
	Build Source Code for MAD														
	** Logbook-Week 2 (26/7)														
	** Logbook-Week 3 (2/8)														
2	**Submission of Progress Report 1**														
3	**Project Work Continue**														
	Build Source Code for MSE														
	Simulating Predicted Image														
	** Logbook-Week 4 (9/8) - Week 7 (30/8)														
4	**Submission of Progress Report 2**														
	** Logbook-Week 8 (13/9)														
	** Logbook-Week 9 (20/9)														
5	**Project work continue**														
	-Program alterations for Full Search														
	-Build source Code for Fast Search														
6	**Pre-EDX**														
7	**Submission of Dissertation Final Draft**														
8	**Oral Presentation**														
9	**Submission of Project Dissertation**														

Process

APPENDIX B

```
% extQCIF.m                                          *%
% open qcif file and extract YUV                     *%
% by Nurul Najahah Abu Bakar UTP (E1974)             *%
%*******************************************%
clear all;
close all;

% parameters
filename='news.qcif';
frames=100;
col=176;
row=144;
UV_ratio=2;
UV_col=col/UV_ratio;
UV_row=row/UV_ratio;
Y=zeros(row,col,frames);

% open file
Y_size=col*row;
U_size=UV_col*UV_row;
V_size=UV_col*UV_row;
buffer_size=Y_size+U_size+V_size;

[fid,message]=fopen(filename,'r');

h = findobj('Type','line','Color','r','LineStyle',':'); %graphic handler

%read and save each YUV frame
for ith_frame=1:1:frames
 [raw_YUV,count]=fread(fid,buffer_size,'uint8');

  temp_Y_frame=reshape(uint8(raw_YUV(1:Y_size)),[col row])';
  subplot(1,1,1);  imshow(temp_Y_frame,[0 255]); title('Y');
  filenameY = strcat(num2str(ith_frame),'Ynews');
  imwrite(temp_Y_frame,filenameY,'jpeg')
  pause

  temp_U_frame=reshape(uint8(raw_YUV(Y_size+1:Y_size+U_size)),[UV_col UV_row])';
  subplot(1,1,1);  imshow(temp_U_frame,[0 255]); title('U');
  filenameU = strcat(num2str(ith_frame),'Unews');
  imwrite(temp_U_frame,filenameU,'jpeg')
  pause

  temp_V_frame=reshape(uint8(raw_YUV(Y_size+U_size+1:Y_size+U_size+V_size)),[UV_col
  UV_row])';
  subplot(1,1,1);  imshow(temp_V_frame,[0 255]); title('V');
  filenameV = strcat(num2str(ith_frame),'Vnews');
  imwrite(temp_V_frame,filenameV,'jpeg')


  xlabel(strcat(num2str(ith_frame),'th frame. Press any key to show next'));


end
```

APPENDIX C

Source Code for Minimum Mean Square Error (MMSE)

```
% MSEcalc.m                                    *%
% Calculate MSE and write predicted image      *%
% Obtain Elapsed Time(s)                       *%
% by Nurul Najahah Abu Bakar UTP (E1974)       *%
%*******************************************%

%Parameters:
%i1: anchor frame; im2: target frame, imp: predicted image;
%B: block size; R: search window range

%********************** MSE ******************************%
%for Y frame:
clear all;
close all;

for loop = 2:100 %start loop frames
i1 = imread ('1Ynews');
im2 = imread (strcat(num2str(loop),'Ynews'));
B = 16;%block size
sz = B*B;
R = 7; %search window
[r,c,d] = size(i1); %obtain size for image i1

im1 = padarray(i1,[(R) (R)],0,'both');%pad both x and y sides with '0'
tic %start timer

for i=1:B:r-(B-1)
for j=1:B:c-(B-1)  %for every block in the reference and predicted image frame (dinstinct)


%j,k= column, i,l = row
MSEsumfin = 0;
MSEabsdiff =0;
MSEsum = 0;
MSEfin=0;
m=1; %start pixel of every sliding block = row
for l= -R:1:R %Search window (2R+1)
   n=1;%start pixel of every sliding block = col
for k= -R:1:R
% calculate MSE for this candidate
dx(n)= k; dy(m) = l;
MSEdiff = imsubtract(im2(i:i+B-1,j:j+B-1),im1(i+R+l:i+R+l+B-1,j+R+k:j+R+k+B-1));% im1 will
slide its block by one pixel at a time
MSEdiffconv = double(MSEdiff); %change class uint8 to double for square operation
MSEsqrdiff = immultiply(MSEdiffconv,MSEdiffconv);%square operation
MSEsum = sum(MSEsqrdiff);
MSEsumfin = sum(MSEsum);%sum all MSE avg for one block
MSEfin(m,n) = MSEsumfin/sz; %store MSEfin
n = n+1;
end
m=m+1;
end
```

```
C = min(MSEfin);
N = min(C); %find the lowest MSE value


%_________________writing predicted image MSE_________________________%

for p = 1:(2*R)+1
for q = 1:(2*R)+1
if MSEfin(p,q) == B
   imp(i:i+B-1,j:j+B-1)= im1(dy(p)+i+R:i+R+dy(p)+B-1,dx(q)+j+R:j+R+dx(q)+B-1);
   iblk = floor ((i-1)/B+1);jblk = floor ((j-1)/B+1);
   mvx(iblk,jblk) = dx(q);
   mvy(iblk,jblk) = dy(p); %record the estimated MV
end
end
end
%____________________________________________________________________%

end
end
t(loop) = toc;

[X,Y] = meshgrid(1:B:c-(B-1),1:B:r-(B-1)); %set motion vector coordinates

figure, imshow(imp,[0 255]); title ('imp')
hold on
quiver(X,Y,mvx,mvy)
hold off

imwrite(imp,'18Y_16newsMSE15.jpg')%write predicted image to file

%____________________________________________________________________%
end %end loop frames
```

APPENDIX D

Source Code for Mean Absolute Difference (MAD)

```
% MADcalc.m                                      *%
% Calculate MAD and write predicted image        *%
% Obtain Elapsed Time(s)                         *%
% by Nurul Najahah Abu Bakar UTP (E1974)         *%
%*******************************************%

%Parameters:
%i1: anchor frame; im2: target frame, imp: predicted image;
%B: block size; R: search window range

%********************** MAD ******************************%
%for Y frame:
clear all;
close all;

for loop = 2:100 %start loop frames
i1 = imread ('1Ynews');
im2 = imread (strcat(num2str(loop),'Ynews'));
B = 16;%block size
sz = B*B;
R = 7; %search window
[r,c,d] = size(i1);
im1 = padarray(i1,[(R) (R)],0,'both');%pad both x and y sides with '0'

tic %start timer

for i=1:B:r-(B-1)
for j=1:B:c-(B-1)  %for every block in the reference and predicted image frame (dinstinct)

%j,k= column, i,l = row
MADsumfin = 0;
MADabsdiff =0;
MADsum = 0;
MADfin=0;
m=1; %start pixel of every sliding block = row
for l= -R:1:R %Search window (2R+1)
   n=1;%start pixel of every sliding block = col
for k= -R:1:R
% calculate MAD for this candidate
dx(n)= k; dy(m) = l;
MADabsdiff = imabsdiff(im2(i:i+B-1,j:j+B-1),im1(i+R+l:i+R+l+B-1,j+R+k:j+R+k+B-1));% im1 will
slide its block by one pixel at a time
MADsum =  sum(MADabsdiff);%sum all MAD avg for one row
MADsumfin = sum(MADsum);%sum all MAD avg for one block
MADfin(m,n) = MADsumfin/sz; %store MAD (array)
n = n+1;
end
m=m+1;
end

C = min(MADfin);
N = min(C); %find the lowest MAD value

%____writing predicted image_____________%
```

```
for p = 1:(2*R)+1
for q = 1:(2*R)+1
if MADfin(p,q) == B
   a = dy(p);
   b = dx(q);
   MADblock = MADfin(p,q);
   imp(i:i+B-1,j:j+B-1)= im1(dy(p)+i+R:i+R+dy(p)+B-1,dx(q)+j+R:j+R+dx(q)+B-1);
   iblk = floor ((i-1)/B+1);jblk = floor ((j-1)/B+1);
   mvx(iblk,jblk) = dx(q);
   mvy(iblk,jblk) = dy(p); %record the estimated MV
end
end
end
%_______________________________________%

end
end
t(loop) = toc; %stop timer

[X,Y] = meshgrid(1:B:c-(B-1),1:B:r-(B-1));   %Set motion vector coordinates

figure, imshow(imp,[0 255]); title ('imp')
hold on
quiver(X,Y,mvx,mvy)
hold off

imwrite(imp,'18Y_16newsMAD15.jpg') %write predicted image to file

%______________________________________________________________________%
end %end loop frames
```

APPENDIX E

Source Code for Three Step Search (TSS)

```
% TSS.m                                            *%
% Calculate TSS and write predicted image          *%
% Obtain Elapsed Time(s)                           *%
% by Nurul Najahah Abu Bakar UTP (E1974)           *%
%*******************************************%

%Parameters:
%i1: anchor frame; im2: target frame, imp: predicted image;
%B: block size; R: search window range

%********************** MAD_TSS ********************************%
%for Y frame:
clear all;
close all;

for loop = 2:100 %start loop frames
i1 = imread ('1Ynews');
im2 = imread (strcat(num2str(loop),'Ynews'));
B = 16;%block size
sz = B*B;
R = 15; %search window
[r,c,d] = size(i1);
W = 2*R + B; %Search window range
M=R+1; % center pixel (point(0,0)
im1 = padarray(i1,[(R) (R)],0,'both');

L = log2(R+1); % number of steps

tic %start timer

for i=1:B:r-(B-1)
for j=1:B:c-(B-1)  %for every block in the reference and predicted image frame (dinstinct)

%j,k= column, i,l = row
MADsumfin = 0;
MADabsdiff =0;
MADsum = 0;
MADfin=0;

%%%%%%%%%%%%%%%%%%%%%%%%%%START FIRST
STEP%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%
nth = 1;
S = 2^(L-nth); %step-size for nth step
m=1; %start pixel of every sliding block = row
for l= -R+(R-S):S:R %Search window
   n=1;%start pixel of every sliding block = col
for k= -R+(R-S):S:R
% calculate MAD for this candidate
dx(n)= k; dy(m) = l;
MADabsdiff = imabsdiff(im2(i:i+B-1,j:j+B-1),im1(i+B+l:i+R+l+B-1,j+B+k:j+R+k+B-1));% im1 will
slide its block by one pixel at a time
MADsum =  sum(MADabsdiff);%sum all MAD avg for one row
```

```
MADsumfin = sum(MADsum);%sum all MAD avg for one block
MADfin(m,n) = MADsumfin/sz; %store MAD (array)
n = n+1;
end
m=m+1;
end

C = min(MADfin);
N = min(C);

%_________________writing predicted image for first step_________________________%

for p = 1:3 %%%for 3 point for each axis
for q = 1:3
  if MADfin(p,q) == B
  a = dy(p);
  b = dx(q);
  end
end
end
%_____________________________________________________________________________
____%

if (a == M) & (b == M)
  MADblock = MADfin(p,q);
  imp(i:i+B-1,j:j+B-1)= im1(dy(p)+i+R:i+R+dy(p)+B-1,dx(q)+j+R:j+R+dx(q)+B-1);
  iblk = floor ((i-1)/B+1);jblk = floor ((j-1)/B+1);
  mvx(iblk,jblk) = dx(q);
  mvy(iblk,jblk) = dy(p); %record the estimated MV
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%END
FIRST
STEP%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%

else
  Py = i+R+a; Px = j+R+b; %set coordinates of the lowest MAD

%%%%%%%%%%START SECOND & THIRD
STEP%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%
 for nth = 2:L
  S = 2^(L-nth) ;
  Pm=1;
 for Pl= -R+(R-S):S:R-(R-S) %Search window
  Pn=1;%start pixel of every sliding block = col
 for Pk= -R+(R-S):S:R-(R-S)
% calculate MAD for this candidate
 Pdx(Pn)= Pk; Pdy(Pm) = Pl;
 MADabsdiff = imabsdiff(im2(i:i+B-1,j:j+B-1),im1(Py+Pl:Py+Pl+B-1,Px+Pk:Px+Pk+B-1));% im1
will slide its block by one pixel at a time
 MADsum =  sum(MADabsdiff);%sum all MAD avg for one row
 MADsumfin = sum(MADsum);%sum all MAD avg for one block
 MADfin(Pm,Pn) = MADsumfin/sz; %store MAD (array)
 Pn = Pn+1;
 end
 Pm=Pm+1;
 end

 PC = min(MADfin);
 PB = min(C);
```

```
%______________________writing predicted image after second and third
step_______________________%

for Pp = 1:3 %%%for 3 point for each axis
for Pq = 1:3
  if MADfin(Pp,Pq) == PB
  Pa = Pdy(Pp);
  Pb = Pdx(Pq);
  MADblock = MADfin(Pp,Pq);
  imp(i:i+B-1,j:j+B-1)= im1(Py+Pdy(p):Py+Pdy(p)+B-1,Px+Pdx(Pq):Px+Pdx(Pq)+BN-1);
  iblk = floor ((i-1)/B+1);jblk = floor ((j-1)/B+1);
  mvx(iblk,jblk) = Pdx(Pq);
  mvy(iblk,jblk) = Pdy(Pp); %record the estimated MV
  end
end
end
Px = Px+Pa; Py = Py+Pb; % reset value for nth step

%______________________________________________________________
____________________%
end

end
end
end
toc(loop)%stop timer

[X,Y] = meshgrid(1:B:c-(N-1),1:B:r-(B-1));  %set motion vector coordinates

figure, imshow(imp,[0 255]); title ('imp')
hold on
quiver(X,Y,mvx,mvy)
hold off

imwrite(imp,'110Y_16carphoneTSS15.jpg')
%______________________________________________________________
_____________________%
end %end loop frames
```

APPENDIX F

Source Code for PSNR

```
% PSNR255.m                                        *%
% Calculate PSNR value (dB)                        *%
% by Nurul Najahah Abu Bakar UTP (E1974)           *%
%*******************************************%

clear all;
close all;

A = imread ('10Ynews');%original image
B = imread ('18Y_4newsMSE7.jpg');%predicted image


if A == B
  error('Images are identical: PSNR has infinite value')
end

diff = imsubtract (A,B);
C = immultiply (diff,diff);
indecibels = 20*log10(255/(sqrt(mean(mean(C)))));
sprintf('PSNR = +%5.2f dB',indecibels)
```

APPENDIX G

For *claire.qcif*:

Peak Signal-to-Noise Ratio (PSNR)										
W	**7**				**15**				**7**	
Frame	**MMSE 4**	**MMSE 16**	**MAD 4**	**MAD 16**	**MAD 4**	**MAD 16**	**TTS 4**	**TSS 16**	**TSS 4**	**TSS 16**
2	57.85	43.51	43.90	42.79	43.91	42.78	36.84	37.43	37.21	37.43
3	53.18	40.76	41.79	39.68	41.88	39.67	36.06	36.11	36.54	36.11
4	52.37	40.13	41.49	39.01	41.72	39.01	35.88	35.66	36.19	35.81
5	51.95	40.07	41.05	38.85	41.25	38.85	35.86	35.39	36.12	35.86
6	51.17	39.85	40.84	38.14	41.02	38.14	35.73	35.47	36.06	35.64
7	50.47	39.51	40.52	37.98	40.93	37.98	35.61	35.49	35.93	35.46
8	50.73	39.31	40.39	37.99	40.74	37.99	35.79	35.60	36.08	35.48
9	50.84	39.43	40.45	37.65	40.87	37.65	35.58	35.79	35.97	35.20
10	50.31	39.11	40.09	37.20	40.50	37.20	35.47	35.63	35.89	35.08
11	50.71	39.21	40.31	37.33	40.73	37.33	35.53	35.51	35.95	35.12
12	50.13	39.03	39.86	37.19	40.23	37.19	35.43	35.42	35.95	34.99
13	49.86	38.84	39.61	37.00	40.07	37.00	35.53	35.37	35.83	34.89
14	50.26	39.08	40.01	37.21	40.36	37.21	35.39	35.34	36.06	34.98
15	50.85	39.01	40.23	37.38	40.60	37.37	35.46	35.60	36.01	35.15
16	50.50	38.97	40.01	37.29	40.44	37.29	35.42	35.46	36.02	35.14
17	50.69	39.14	40.28	37.50	40.60	37.50	35.45	35.56	35.92	35.41
18	52.08	39.79	40.68	38.15	41.05	38.14	35.51	35.33	36.20	35.43
19	51.46	39.79	40.46	38.10	40.85	38.09	35.48	35.34	36.23	35.40
20	52.16	40.07	40.83	38.45	41.05	38.45	35.55	35.51	36.34	35.68
21	51.56	39.93	40.64	38.47	41.00	38.46	35.54	35.58	36.29	35.73
22	51.01	39.77	40.40	38.27	40.80	38.27	35.61	35.42	36.27	35.49
23	51.18	39.70	40.52	38.25	41.02	38.24	35.49	35.52	36.11	35.62
24	50.76	39.83	40.29	37.85	40.68	37.84	35.26	35.42	36.16	35.48
25	50.50	39.63	40.29	37.56	40.63	37.55	35.38	35.26	36.18	35.44
26	50.89	39.80	40.32	37.77	40.68	37.76	35.43	35.40	36.21	35.52
27	50.89	39.92	40.36	37.82	40.58	37.77	35.45	35.39	36.13	35.45
28	51.62	40.00	40.21	38.02	40.49	37.98	35.46	35.34	36.08	35.44
29	52.63	40.46	40.70	38.24	40.83	38.23	35.33	35.53	36.22	35.67
30	54.28	40.76	41.04	38.83	41.07	38.82	35.38	35.56	36.27	35.85
31	53.46	40.65	41.07	38.68	41.15	38.66	35.33	35.48	36.23	35.70
32	54.33	40.97	41.27	39.07	41.33	39.03	35.31	35.53	36.34	35.80
33	54.62	41.16	41.53	39.03	41.69	39.06	35.44	35.40	36.39	35.95
34	54.46	40.99	41.16	38.95	41.36	38.94	35.49	35.35	36.39	35.91
35	55.26	41.19	41.19	39.35	41.33	39.33	35.48	35.42	36.48	36.01
36	55.32	41.26	41.06	39.19	41.14	39.16	35.36	35.42	36.37	36.10
37	54.30	41.06	41.03	38.91	41.08	38.90	35.38	35.38	36.28	36.05
38	55.03	41.35	41.15	38.81	41.11	38.74	35.50	35.36	36.38	36.06
39	55.65	41.42	41.15	38.97	41.18	38.92	35.55	35.47	36.31	35.98
40	56.18	41.06	41.17	38.66	41.28	38.65	35.51	35.46	36.28	35.97
41	55.76	41.18	41.26	38.87	41.59	38.86	35.57	35.58	36.30	36.19
42	56.12	41.14	41.50	38.94	41.70	38.96	35.66	35.71	36.44	36.23
43	56.02	41.20	41.50	38.88	41.72	38.91	35.79	35.68	36.48	36.35
44	56.59	41.19	41.42	38.93	41.73	38.97	35.77	35.77	36.39	36.17
45	57.71	41.40	41.46	38.90	41.49	38.96	35.83	35.76	36.60	36.24
46	56.81	40.96	41.42	38.71	41.43	38.78	35.84	35.67	36.56	36.10

47	57.22	41.06	41.70	38.68	41.59	38.73	35.77	35.68	36.45	36.16
48	56.87	41.11	41.23	38.59	41.30	38.61	35.88	35.62	36.52	36.18
49	55.41	40.98	41.25	38.85	41.32	38.87	35.68	35.48	36.50	36.38
50	56.93	41.15	41.52	39.17	41.56	39.21	35.40	35.55	36.46	36.50
51	56.66	41.10	41.59	38.92	41.54	38.91	35.52	35.42	36.50	36.31
52	55.25	40.74	41.27	38.61	41.37	38.65	35.52	35.29	36.29	36.06
53	55.80	40.81	41.49	38.86	41.58	38.89	35.49	35.28	36.38	36.09
54	55.38	40.75	41.14	38.87	41.36	38.86	35.41	35.10	36.32	35.90
55	56.22	40.47	40.95	38.31	41.16	38.37	35.52	34.98	36.19	35.84
56	56.44	40.62	40.91	38.38	41.03	38.42	35.54	35.08	36.15	35.86
57	56.34	40.69	40.88	38.45	41.20	38.49	35.46	34.87	36.15	35.72
58	55.03	40.39	40.78	38.35	41.06	38.37	35.68	34.84	36.04	35.67
59	55.71	40.48	40.83	38.30	41.14	38.42	35.62	34.94	36.08	35.60
60	56.18	40.47	40.98	38.60	41.20	38.62	35.71	35.20	36.29	35.67
61	54.52	40.36	40.65	38.30	40.91	38.38	35.71	35.24	36.07	35.79
62	55.68	40.61	41.14	38.39	41.38	38.56	35.79	35.39	36.21	35.86
63	55.52	40.69	41.05	38.53	41.38	38.63	35.71	35.35	36.08	35.89
64	55.15	40.50	40.78	38.21	41.14	38.31	35.69	35.19	36.04	35.83
65	55.56	40.67	41.00	38.33	41.26	38.44	35.64	35.33	36.13	35.81
66	55.51	40.65	40.85	38.35	41.12	38.47	35.71	35.21	36.17	35.91
67	54.46	40.42	40.94	38.05	41.25	38.17	35.67	35.06	35.93	35.73
68	55.19	40.33	40.81	38.11	41.18	38.23	35.67	35.08	36.05	35.55
69	55.34	40.39	40.90	38.17	41.10	38.17	35.48	34.98	36.15	35.63
70	54.73	40.12	40.99	38.02	41.26	38.00	35.49	35.03	36.15	35.58
71	54.53	40.51	41.08	38.19	41.21	38.20	35.60	34.95	36.15	35.63
72	53.97	40.41	40.94	38.25	41.20	38.24	35.50	34.96	35.95	35.55
73	53.19	40.19	40.80	38.12	41.23	38.12	35.46	34.88	35.98	35.44
74	53.62	40.38	40.63	38.16	40.99	38.23	35.55	34.96	35.99	35.55
75	54.00	40.13	40.68	38.37	40.97	38.47	35.53	34.95	36.01	35.69
76	53.55	40.04	40.67	38.18	41.05	38.23	35.54	34.91	35.99	35.66
77	54.70	40.21	40.77	38.48	41.18	38.56	35.73	35.04	36.09	35.86
78	55.10	40.51	40.81	38.23	41.04	38.38	35.75	35.24	36.12	35.78
79	54.86	40.68	41.27	38.53	41.45	38.58	35.70	35.17	36.10	35.84
80	54.93	40.86	41.08	38.65	41.30	38.66	35.59	35.31	36.25	35.99
81	55.58	40.64	40.95	38.62	41.11	38.64	35.65	35.25	36.22	35.98
82	55.26	40.40	40.87	38.39	41.21	38.41	35.48	35.09	36.12	35.84
83	55.84	40.48	41.05	38.45	41.26	38.45	35.53	35.01	36.15	35.81
84	55.44	40.41	40.95	38.14	40.97	38.16	35.61	34.91	36.03	35.66
85	54.24	40.21	40.62	38.06	40.91	38.10	35.57	34.82	35.94	35.46
86	54.36	40.25	40.61	38.22	40.80	38.26	35.57	34.87	35.98	35.50
87	54.07	40.12	40.82	38.19	41.17	38.14	35.55	34.75	35.91	35.58
88	53.38	39.83	40.63	37.88	40.88	37.82	35.47	34.81	35.89	35.43
89	53.55	39.76	40.53	37.85	40.98	38.23	35.47	34.90	35.85	35.44
90	52.40	40.08	40.55	37.43	41.03	37.79	35.51	34.79	35.89	35.26
91	51.72	39.68	40.45	37.25	40.98	37.70	35.61	35.06	35.75	35.20
92	51.75	39.70	40.37	37.24	40.78	37.68	35.67	35.32	35.83	35.28
93	51.29	39.61	40.13	37.31	40.97	37.45	35.70	35.39	35.85	35.44
94	50.92	39.34	40.10	37.15	40.65	37.25	35.57	35.29	35.84	35.30
95	51.53	39.64	40.28	37.18	40.81	37.25	35.70	34.92	36.06	35.43
96	53.45	39.94	40.28	37.40	40.94	37.37	35.84	35.01	36.03	35.43
97	52.94	39.91	40.48	37.31	41.00	37.33	35.71	35.15	35.99	35.58
98	53.58	40.14	41.00	37.37	41.41	37.38	35.85	35.41	36.17	35.63
99	53.80	40.19	40.41	37.72	41.37	37.85	35.80	35.51	36.07	35.61
100	53.06	40.13	40.55	37.76	41.40	37.66	35.76	35.57	36.15	35.53

Elapsed Time (s)										
W	7				15				7	
Frame	MMSE 4	MMSE 16	MAD 4	MAD 16	MAD 4	MAD 16	TTS 4	TSS 16	TSS 4	TSS 16
2	38.8450	3.4650	16.0540	2.0630	66.5260	8.7120	3.0140	0.3300	2.1930	0.2610
3	38.7960	3.4550	17.9060	1.9630	67.1260	8.6030	2.8540	0.3410	2.2240	0.2500
4	38.5050	3.4450	16.7340	2.1230	67.4370	8.4520	3.0250	0.3500	2.2530	0.2700
5	38.8060	3.4550	16.5540	2.1630	68.1580	8.6930	2.9840	0.3410	2.2130	0.2600
6	38.9060	3.4750	16.4730	2.1230	68.3990	9.0830	3.0050	0.3700	2.2730	0.2800
7	38.5050	3.4650	16.3030	2.0430	67.6180	8.5220	2.8540	0.3310	2.2440	0.2600
8	38.7960	3.4550	16.2230	2.0530	69.4600	8.2720	2.8650	0.3300	2.2830	0.2500
9	39.0060	3.5350	16.0130	2.0130	64.4630	8.2220	3.0540	0.3610	2.3630	0.2710
10	38.6460	3.4650	15.7820	1.9530	64.1730	8.7020	2.9640	0.3400	2.2430	0.2510
11	39.2170	3.4850	16.2340	1.9420	65.7650	8.7420	2.9940	0.3410	2.2640	0.2600
12	39.1370	3.4750	16.6230	2.1740	69.0490	8.9330	2.8640	0.3400	2.2430	0.2910
13	38.6460	3.4450	17.1040	2.2340	67.5370	8.2820	2.8540	0.3300	2.1530	0.2510
14	38.9860	3.4550	16.6540	2.0630	71.0820	8.4220	2.9040	0.3610	2.2740	0.2500
15	38.9860	3.4350	16.9450	1.9420	69.8910	8.8330	2.9640	0.3300	2.2730	0.2710
16	38.7160	3.4750	15.9530	1.9520	68.2090	9.1130	2.9550	0.3400	2.4640	0.2700
17	41.5700	3.4550	16.7540	2.0330	65.5140	8.6820	3.1740	0.3310	2.2630	0.2510
18	39.3760	3.4550	16.0630	2.0730	67.9370	8.4530	2.8540	0.3510	2.1530	0.2500
19	39.0060	3.4550	16.1630	2.0430	66.5050	8.7530	3.0140	0.3310	2.3840	0.3100
20	39.3870	3.4650	16.4640	1.9830	66.0850	8.4020	3.0040	0.3410	2.3530	0.2510
21	39.6470	3.4650	16.1830	2.0130	66.8460	8.4820	3.2050	0.3400	2.3530	0.2600
22	39.4060	3.4750	16.2030	1.9330	64.1120	8.2820	2.8940	0.3310	2.2430	0.2600
23	39.7470	3.5050	16.0130	2.0130	64.6330	8.2110	2.9150	0.3500	2.1530	0.2500
24	39.8170	3.4850	15.6820	1.9430	64.5930	8.2120	2.9340	0.3310	2.2640	0.2700
25	39.5870	3.4750	15.9530	2.0430	64.6530	8.3820	2.8840	0.3310	2.2130	0.2500
26	40.1370	3.4950	16.0130	1.9430	64.4420	8.2320	3.0340	0.3610	2.2530	0.2610
27	40.0480	3.5150	15.6030	1.9320	64.3830	8.2620	2.9250	0.3500	2.2530	0.2500
28	39.6570	3.4950	15.6420	1.9420	64.7730	8.2220	2.9640	0.3510	2.2430	0.2710
29	39.9570	3.4950	15.6830	1.9430	64.6230	8.2420	3.1040	0.3310	2.6730	0.2510
30	39.9370	3.5050	15.7120	1.9930	64.2930	8.2110	2.8650	0.3300	2.1530	0.2500
31	39.7870	3.5050	15.6330	1.9330	64.1420	8.2120	2.9440	0.3500	2.1630	0.2510
32	39.9680	3.4950	15.6020	1.9430	64.3130	8.2420	2.9740	0.3410	2.1940	0.2500
33	40.0270	3.5050	15.6120	1.9430	64.1720	8.1920	2.9950	0.4800	2.1830	0.3500
34	39.6570	3.4950	15.6430	1.9320	64.1130	8.1910	2.8640	0.3710	2.1830	0.2500
35	40.2680	3.5150	15.6220	1.9430	64.0120	8.2120	3.0350	0.3400	2.2030	0.3200
36	40.1180	3.5050	15.6420	1.9430	64.6630	8.2320	2.9940	0.3510	2.2940	0.2500
37	39.7770	3.4960	15.6230	1.9320	64.0730	8.2010	2.9440	0.3600	2.2930	0.2510
38	40.0780	3.5050	15.6420	1.9430	64.1220	8.2120	2.9840	0.3910	2.2430	0.2500
39	40.0880	3.5050	15.5920	1.9630	65.7450	8.3620	3.2440	0.3410	2.2830	0.2710
40	41.5600	3.6350	15.6420	1.9330	64.1520	8.2020	3.1640	0.3310	2.1830	0.2710
41	42.4310	3.5760	15.5930	1.9430	64.2020	8.2020	3.0940	0.3500	2.3930	0.2710
42	41.1590	3.5850	15.6320	1.9430	64.7030	8.2420	2.8440	0.3710	2.3030	0.2600
43	41.7500	4.1060	15.7730	1.9330	64.7830	8.4120	3.1240	0.3410	2.4640	0.2500
44	41.5490	3.6060	16.0130	1.9430	68.2780	8.5620	3.0940	0.3610	2.3030	0.2510
45	40.5780	3.5150	16.6940	2.1730	69.2390	8.7230	3.0150	0.3500	2.2830	0.2500
46	39.8770	3.5150	16.7240	2.0630	69.1190	9.0030	2.9740	0.3410	2.2340	0.3000
47	40.3380	3.5360	16.5030	1.9430	66.9760	8.5320	3.0540	0.3400	2.2730	0.2510
48	40.2580	3.5250	16.1230	2.0130	65.6550	8.5620	2.9440	0.4210	2.2530	0.2600
49	39.9070	3.5050	16.2430	2.0130	69.5300	8.7030	2.9240	0.3600	2.2330	0.2610

50	40.1280	3.5150	16.5940	1.9430	65.4140	8.5420	3.0640	0.3410	2.2140	0.2500
51	40.0970	3.5050	15.9830	1.9430	66.6660	8.8330	3.0140	0.3910	2.2930	0.2510
52	39.9170	3.5050	16.7640	2.1830	68.1980	8.5830	3.0450	0.3500	2.2430	0.2600
53	40.1780	3.5050	16.2130	1.9830	65.5540	8.4530	2.8640	0.3300	2.1930	0.2510
54	40.4380	3.5150	16.5630	1.9820	65.0740	8.2920	2.9740	0.3610	2.2140	0.2600
55	40.3080	3.5150	15.6530	1.9420	64.1030	8.2410	2.9550	0.3600	2.1830	0.2600
56	40.0980	3.5050	15.6120	1.9730	64.5530	8.5220	2.9140	0.3410	2.2540	0.2500
57	40.8290	3.5650	15.8230	1.9730	65.8450	8.2020	2.8840	0.3410	2.1630	0.2500
58	40.2280	3.5150	15.5820	1.9430	64.2020	8.2420	2.8640	0.3410	2.1630	0.2500
59	40.1370	3.5160	15.5730	1.9320	66.0650	8.6630	2.8640	0.3400	2.1530	0.2500
60	40.1980	3.5250	16.0430	2.0230	66.7760	8.8030	2.8640	0.3400	2.1530	0.2710
61	39.8380	3.4950	16.2230	1.9730	64.4930	8.2120	2.8540	0.3410	2.1540	0.2500
62	41.1190	3.5660	15.6320	1.9530	64.1520	8.2320	2.8640	0.3510	2.1730	0.2600
63	41.1790	3.6660	15.7130	1.9830	68.6490	8.4720	2.8640	0.3410	2.1530	0.2500
64	41.1490	3.6150	16.3930	1.9330	65.8550	8.2720	2.8550	0.3300	2.1530	0.2500
65	41.4900	3.6450	15.6730	1.9430	64.2120	8.2420	2.8640	0.3400	2.1530	0.2510
66	42.8620	3.6350	15.5720	1.9430	64.5430	9.1430	2.8640	0.3300	2.1530	0.2510
67	41.3090	3.6050	16.3340	2.0430	69.0990	8.9730	2.8740	0.3310	2.1530	0.2500
68	40.1880	3.5050	16.3840	1.9830	69.0890	8.7130	2.9040	0.3410	2.1630	0.2500
69	42.5010	3.5950	16.5340	1.9930	66.9060	8.3920	2.8740	0.3300	2.1730	0.2510
70	41.2790	3.5050	17.0140	2.0630	65.6550	8.7930	2.8740	0.3400	2.1430	0.2610
71	41.3690	3.7250	16.5830	2.0930	66.9760	8.5320	2.8640	0.3300	2.1430	0.2510
72	41.3090	3.5650	16.0730	1.9730	65.0640	8.5820	2.8640	0.3310	2.1530	0.2500
73	41.2790	3.5960	15.8430	2.1530	68.1780	9.4340	2.8540	0.3310	2.1630	0.2500
74	40.7390	3.6550	17.2850	1.9430	69.6500	9.4830	2.8640	0.3310	2.1530	0.2500
75	42.0210	3.6850	15.9320	1.9320	66.0850	8.2220	2.8550	0.3300	2.1530	0.2500
76	40.9290	3.5550	15.9430	2.0130	70.5720	9.2230	2.8540	0.3300	2.1630	0.2510
77	41.2490	3.5860	16.8340	2.2530	67.9870	8.2220	2.8640	0.3300	2.1530	0.2510
78	41.8000	3.6660	15.5830	1.9420	64.0320	8.2210	2.8640	0.3400	2.1630	0.2510
79	41.7000	3.6950	16.0030	2.0830	71.2220	8.6420	2.8640	0.3310	2.1530	0.2500
80	41.1990	3.6850	16.7240	1.9620	65.6950	9.5940	2.8640	0.3310	2.1530	0.2500
81	41.8200	3.6150	15.5520	1.9530	70.6820	9.1030	2.8650	0.3300	2.1530	0.2500
82	41.8100	3.6350	17.2350	2.1930	67.3570	8.5820	2.8640	0.3300	2.1530	0.2510
83	41.2900	3.5250	17.2850	1.9430	67.9180	8.7420	2.8640	0.3300	2.1730	0.2510
84	42.2410	3.7550	17.0650	2.4030	72.1440	8.3920	2.8740	0.3410	2.1540	0.2500
85	41.7100	3.7960	17.6860	2.2530	68.0180	8.3220	2.8740	0.3310	2.1430	0.2500
86	41.8200	3.5650	15.7430	1.9420	64.2930	8.2120	2.8740	0.3310	2.1530	0.2500
87	42.2310	3.6050	15.5620	1.9430	64.1520	8.3920	2.8740	0.3300	2.1530	0.2510
88	40.9990	3.6150	15.5920	1.9330	64.4630	8.3120	2.8540	0.3400	2.1630	0.2510
89	41.4000	3.6750	15.6020	1.9430	64.4430	8.2120	2.8540	0.3300	2.1630	0.2510
90	41.3400	3.6350	15.6120	1.9430	64.1820	8.2520	2.8540	0.3310	2.1630	0.2500
91	41.3990	3.6550	15.6730	1.9530	64.2620	8.2120	2.8540	0.3310	2.1530	0.2500
92	41.7700	3.7750	15.6020	1.9420	64.3530	8.2320	2.8540	0.3410	2.1630	0.2500
93	41.6500	3.7060	15.6620	1.9430	64.4930	8.2120	2.8550	0.3300	2.1530	0.2500
94	40.9190	3.5250	15.6820	1.9330	64.5330	8.2520	2.8640	0.3300	2.1530	0.2510
95	41.4100	3.6750	15.6430	1.9420	64.0730	8.4720	2.8640	0.3400	2.1530	0.2510
96	41.7500	3.7850	15.5320	1.9330	64.3930	8.2110	2.8740	0.3400	2.1530	0.2510
97	41.3090	3.5750	15.5920	1.9330	64.1120	8.2120	2.8540	0.3310	2.1630	0.2500
98	41.5700	3.5650	15.5820	1.9430	64.1020	8.2520	2.8540	0.3310	2.1630	0.2500
99	41.4900	3.6250	15.5830	1.9420	64.1120	8.2120	2.8640	0.3310	2.1530	0.2400
100	40.6890	3.7550	15.5720	1.9330	64.2620	8.2420	2.8550	0.3300	2.1730	0.2500

APPENDIX H

For *news.qcif*:

Peak Signal-to-Noise Ratio (PSNR)										
W	**7**				**15**				**7**	
Frame	**MMSE 4**	**MMSE 16**	**MAD 4**	**MAD 16**	**MAD 4**	**MAD 16**	**TTS 4**	**TSS 16**	**TSS 4**	**TSS 16**
2	46.60	38.98	39.51	38.62	39.75	38.61	32.62	32.77	32.69	32.80
3	46.17	38.36	39.03	38.02	39.28	38.01	32.47	32.59	32.57	32.62
4	45.95	38.18	38.86	37.84	39.10	37.84	32.46	32.59	32.55	32.62
5	46.71	37.97	39.08	37.61	39.31	37.60	32.39	32.54	32.49	32.57
6	45.94	37.41	38.83	37.26	39.09	37.25	32.29	32.49	32.48	32.49
7	45.85	37.20	38.28	36.82	38.58	36.62	32.28	32.38	32.47	32.44
8	45.49	36.95	38.00	36.45	38.34	36.27	32.33	32.40	32.51	32.47
9	45.58	36.95	37.91	36.41	38.24	36.24	32.36	32.40	32.53	32.47
10	45.93	36.99	37.66	36.28	38.31	36.04	32.35	32.37	32.50	32.41
11	45.67	37.02	37.79	36.20	38.01	36.14	32.33	32.27	32.59	32.44
12	45.40	36.75	37.79	36.13	37.91	36.05	32.31	32.29	32.64	32.46
13	45.50	36.49	37.64	35.94	37.90	35.82	32.34	32.25	32.62	32.38
14	45.13	36.41	37.45	35.80	37.77	35.69	32.34	32.25	32.62	32.39
15	45.40	36.36	37.47	35.75	37.64	35.66	32.34	32.27	32.63	32.40
16	45.39	36.37	37.27	35.55	37.61	35.59	32.34	32.25	32.60	32.46
17	45.31	36.30	37.48	35.57	37.61	35.64	32.38	32.24	32.59	32.46
18	45.03	36.34	37.44	35.55	37.60	35.76	32.42	32.24	32.62	32.41
19	45.20	36.35	37.52	35.55	37.67	35.76	32.46	32.17	32.62	32.42
20	45.24	36.38	37.60	35.60	37.85	35.69	32.51	32.28	32.69	32.43
21	45.21	36.35	37.62	35.49	37.73	35.55	32.52	32.28	32.71	32.41
22	45.30	36.31	37.52	35.32	37.59	35.44	32.51	32.27	32.69	32.39
23	45.45	36.37	37.36	35.30	37.60	35.62	32.55	32.34	32.77	32.39
24	45.45	36.39	37.35	35.33	37.66	35.65	32.52	32.33	32.72	32.40
25	45.62	36.36	37.29	35.33	37.70	35.61	32.46	32.28	32.62	32.30
26	45.59	36.41	37.49	35.25	37.95	35.58	32.41	32.19	32.64	32.28
27	45.36	36.32	37.49	35.36	37.69	35.58	32.40	32.19	32.71	32.30
28	45.28	36.36	37.50	35.43	37.62	35.50	32.33	32.16	32.65	32.27
29	45.36	36.28	37.51	35.36	37.68	35.54	32.31	32.12	32.64	32.24
30	45.16	36.18	37.49	35.07	37.66	35.24	32.37	32.12	32.63	32.24
31	44.99	35.99	37.40	34.94	37.73	35.12	32.38	32.11	32.67	32.16
32	44.89	36.01	37.21	35.04	37.51	35.03	32.30	31.97	32.63	32.16
33	45.08	36.01	37.17	35.04	37.34	35.25	32.25	32.01	32.56	32.10
34	45.15	36.02	37.19	35.06	37.30	35.27	32.26	31.99	32.47	32.17
35	45.21	36.07	37.15	35.00	37.35	35.19	32.23	31.87	32.51	32.09
36	44.91	35.82	37.16	34.95	37.29	34.96	32.25	31.87	32.51	32.09
37	45.21	35.88	37.12	34.99	37.26	35.15	32.32	31.88	32.53	32.16
38	44.77	35.76	37.08	34.79	37.36	34.99	32.30	31.90	32.43	32.08
39	44.84	35.82	37.12	34.76	37.48	35.05	32.34	31.91	32.47	32.09
40	44.77	35.87	37.21	34.83	37.31	35.04	32.35	31.98	32.49	32.05
41	45.00	35.94	37.24	34.89	37.52	35.06	32.35	31.98	32.51	32.09
42	44.67	35.88	37.24	34.72	37.47	35.00	32.30	31.96	32.49	32.06
43	44.66	35.90	37.17	34.78	37.33	34.91	32.16	31.97	32.45	32.02
44	44.66	35.94	37.21	34.80	37.40	34.92	32.16	31.98	32.43	32.02
45	44.35	35.93	36.85	34.73	37.22	35.01	32.15	31.93	32.38	31.99
46	44.43	35.86	37.00	34.75	37.24	35.05	32.11	31.93	32.41	31.99
47	44.40	35.92	37.22	34.71	37.43	35.09	32.09	31.91	32.39	31.94

48	44.50	35.98	37.44	34.84	37.76	35.16	32.16	31.89	32.40	31.96
49	44.41	35.97	37.37	34.80	37.71	35.11	32.15	31.89	32.42	31.97
50	44.60	35.91	37.50	34.93	37.76	35.18	32.21	31.93	32.43	31.98
51	44.84	35.93	37.48	34.97	37.78	35.17	32.21	31.93	32.35	31.98
52	45.16	35.95	37.59	34.95	37.56	35.30	32.21	31.99	32.43	32.00
53	45.41	36.02	37.50	34.89	37.47	35.27	32.14	31.94	32.351	31.98
54	45.50	36.01	37.57	34.96	37.52	35.33	32.16	31.94	32.36	31.95
55	45.12	35.90	37.30	34.89	37.41	35.29	32.16	31.90	32.33	31.94
56	45.01	35.95	37.30	34.95	37.41	35.14	32.10	31.87	32.28	31.96
57	44.70	35.85	37.15	35.00	37.23	35.06	32.05	31.87	32.20	31.96
58	44.22	35.77	37.01	34.81	37.07	35.03	32.00	31.84	32.25	31.94
59	44.64	35.86	37.13	34.92	37.21	35.13	32.05	31.85	32.24	31.97
60	44.11	35.81	37.27	34.84	37.37	35.09	32.11	31.86	32.27	31.94
61	43.79	35.72	37.11	34.85	37.15	35.06	32.10	31.77	32.21	31.89
62	43.91	35.56	36.95	34.75	37.16	35.05	32.03	31.88	32.16	31.85
63	43.57	35.53	36.86	34.66	37.10	34.97	31.98	31.88	32.09	31.76
64	43.48	35.51	36.78	34.67	37.03	34.99	32.00	31.82	32.08	31.77
65	43.64	35.45	36.85	34.40	37.11	34.90	31.96	31.80	32.07	31.65
66	43.59	35.38	36.85	34.38	37.17	34.88	31.94	31.72	32.10	31.61
67	43.25	35.28	36.90	34.36	37.19	34.86	31.97	31.63	32.13	31.66
68	43.07	35.21	36.69	34.23	37.26	34.82	31.95	31.57	32.14	31.63
69	43.10	35.32	36.74	34.23	37.30	34.81	31.95	31.56	32.14	31.62
70	43.08	35.14	36.42	34.05	36.84	34.62	31.91	31.54	32.17	31.63
71	42.98	34.88	36.37	34.07	36.86	34.45	31.84	31.59	32.12	31.62
72	43.16	34.83	36.47	34.09	37.03	34.45	31.91	31.53	32.13	31.58
73	42.82	34.98	36.47	34.18	36.97	34.48	31.95	31.53	32.15	31.62
74	42.97	35.04	36.50	34.14	36.98	34.52	32.00	31.55	32.10	31.63
75	43.04	35.09	36.48	33.99	36.90	34.30	32.02	31.58	32.08	31.64
76	43.07	35.11	36.36	33.83	36.70	34.10	31.89	31.56	32.02	31.63
77	43.17	35.10	36.33	33.70	36.70	34.01	31.85	31.49	32.08	31.65
78	43.09	35.11	36.23	33.70	36.52	34.03	31.77	31.48	32.06	31.63
79	43.12	35.07	36.24	33.64	36.57	33.92	31.81	31.49	32.10	31.57
80	43.17	35.13	36.23	33.79	36.65	34.12	31.82	31.45	32.03	31.64
81	42.93	35.24	36.43	33.74	36.87	34.04	31.77	31.41	32.07	31.61
82	42.94	35.15	36.31	33.93	36.82	34.23	31.84	31.42	31.99	31.71
83	43.03	35.01	36.16	33.92	36.53	34.19	31.82	31.48	31.94	31.64
84	43.02	35.02	36.11	33.81	36.54	34.14	31.88	31.51	31.97	31.62
85	43.09	35.12	36.26	33.89	36.62	34.21	31.94	31.40	31.98	31.57
86	43.15	35.16	36.23	33.90	36.63	34.13	31.91	31.36	32.00	31.58
87	43.27	35.16	36.23	33.82	36.60	34.02	31.97	31.39	31.91	31.57
88	43.26	35.18	36.32	33.85	36.65	34.04	31.90	31.36	31.98	31.60
89	43.05	35.08	36.18	33.81	36.50	34.03	31.92	31.25	32.04	31.60
90	43.31	35.06	36.18	33.77	36.42	33.96	31.88	31.16	32.08	31.46
91	36.02	32.66	33.60	31.56	34.94	32.43	30.71	30.29	30.49	30.00
92	36.02	32.61	33.52	31.66	34.99	32.59	30.71	30.34	30.51	30.01
93	35.99	32.63	33.49	31.74	34.92	32.70	30.70	30.33	30.49	30.08
94	35.99	32.66	33.45	31.77	34.91	32.72	30.69	30.36	30.48	30.07
95	36.01	32.61	33.44	31.75	34.86	32.66	30.67	30.24	30.55	30.18
96	35.97	32.58	33.41	31.78	34.93	32.59	30.66	30.23	30.54	30.17
97	35.98	32.63	33.48	31.75	34.88	32.45	30.68	30.33	30.55	30.25
98	35.89	32.61	33.37	31.73	34.90	32.42	30.65	30.26	30.57	30.22
99	35.88	32.59	33.37	31.74	34.79	32.45	30.62	30.24	30.55	30.22
100	36.05	32.70	33.47	31.75	34.94	32.48	30.72	30.27	30.52	30.23

Elapsed Time (s)										
W	7				15				7	
Frame	MMSE 4	MMSE 16	MAD 4	MAD 16	MAD 4	MAD 16	TTS 4	TSS 16	TSS 4	TSS 16
2	37.3040	3.5050	15.8830	1.9930	64.1420	8.2320	2.8140	0.3500	2.2530	0.2510
3	37.1630	3.5050	15.3420	1.9730	64.2520	8.2320	2.9240	0.3510	2.1930	0.2600
4	37.6550	3.4750	15.5330	2.0120	64.4830	8.2520	2.8540	0.3700	2.1730	0.2610
5	37.3030	3.5050	15.3620	1.9730	64.5630	8.2920	2.8940	0.3400	2.1530	0.2510
6	37.5240	3.4550	15.5120	1.9930	64.8030	8.4120	2.8640	0.3310	2.1340	0.2500
7	37.0830	3.4450	15.3020	1.9830	64.4430	8.2410	2.8240	0.3410	2.1430	0.2500
8	37.0130	3.4450	15.2720	1.9730	64.5720	8.2520	2.8240	0.3410	2.1230	0.2600
9	37.2130	3.4640	15.3020	1.9830	64.6130	8.2420	2.8140	0.3310	2.1330	0.2500
10	37.0630	3.4550	15.4420	1.9730	64.7030	8.2320	2.8240	0.3410	2.1430	0.2600
11	37.1230	3.4550	15.2920	1.9930	64.5230	8.2220	2.8140	0.3300	2.1330	0.2510
12	37.0140	3.4440	15.3120	1.9730	64.4830	8.2410	2.8240	0.3300	2.1230	0.2510
13	36.9830	3.4450	15.3720	1.9830	67.1970	8.4020	2.8240	0.3410	2.3840	0.2600
14	37.0230	3.4450	15.2620	1.9730	64.6930	8.2420	2.8640	0.3410	2.1330	0.2500
15	37.0430	3.4650	15.2720	1.9730	64.2920	8.2520	2.8140	0.3410	2.1330	0.2500
16	37.2440	3.4550	15.3720	1.9730	64.7030	8.2220	2.8140	0.3400	2.1230	0.2600
17	37.0030	3.4350	15.2820	1.9730	64.3020	8.2220	2.8240	0.3400	2.1130	0.2610
18	36.9330	3.4550	15.2620	1.9830	64.3230	8.2320	2.8740	0.3500	2.1230	0.2610
19	37.5840	3.5050	15.3120	1.9830	64.2520	8.2320	2.9040	0.3310	2.1940	0.2500
20	37.6640	3.4960	15.5320	1.9930	64.4730	8.2410	2.8240	0.3410	2.1230	0.2500
21	37.5640	3.4750	15.4920	2.0030	65.0640	8.2520	2.8240	0.3410	2.1330	0.2500
22	37.5140	3.4950	15.4920	1.9830	64.3920	8.2520	2.8240	0.3310	2.1230	0.2500
23	37.5140	3.4950	15.4520	2.0030	64.6530	8.2420	2.8140	0.3410	2.1230	0.2500
24	37.6140	3.4950	15.5030	2.0030	64.3830	8.2320	2.8250	0.3300	2.1230	0.2500
25	37.6740	3.4950	15.5120	2.0030	64.3020	8.2820	2.8240	0.3400	2.1430	0.2510
26	37.4030	3.4750	15.4630	2.0130	64.3420	8.2420	2.8140	0.3400	2.1430	0.2510
27	36.9830	3.4450	15.4320	1.9830	64.2730	8.2320	2.8540	0.3400	2.1230	0.2510
28	37.1230	3.4550	15.2720	1.9830	64.3820	8.2120	2.8340	0.3310	2.1330	0.2600
29	37.2640	3.5150	15.2820	1.9730	64.1520	8.2020	2.8140	0.3410	2.1130	0.2500
30	37.2140	3.4650	15.5720	1.9930	64.4930	8.2120	2.8140	0.3310	2.1230	0.2500
31	37.5140	3.4950	15.4020	1.9930	64.3130	8.2220	2.8240	0.3310	2.1330	0.2500
32	37.4640	3.5050	15.6220	2.0030	64.2520	8.2020	2.8240	0.3310	2.1230	0.2500
33	37.6840	3.5150	15.6330	2.0030	64.3520	8.2320	2.8140	0.3410	2.1230	0.2500
34	37.6240	3.4950	15.5420	2.0030	64.3730	8.2420	2.8140	0.3410	2.1330	0.2500
35	37.8840	3.4750	15.5520	2.0030	64.3920	8.2320	2.8350	0.3300	2.1130	0.2500
36	37.1340	3.4650	15.3720	1.9830	64.3730	8.2220	2.8140	0.3300	2.1230	0.2610
37	37.1730	3.5050	15.3820	1.9930	64.3930	8.2120	2.8140	0.3300	2.1130	0.2610
38	37.3930	3.4550	15.4620	1.9730	64.6930	8.2220	2.8240	0.3300	2.1330	0.2510
39	37.0540	3.4550	15.3630	1.9720	64.2030	8.2220	2.8040	0.3400	2.1330	0.2510
40	37.1230	3.4550	15.2920	1.9830	64.2920	8.2220	2.8140	0.3300	2.1230	0.2610
41	37.0630	3.4550	15.2920	1.9830	64.5730	8.2420	2.8240	0.3310	2.1330	0.2500
42	37.1240	3.4550	15.3420	1.9730	64.3020	8.2220	2.8240	0.3410	2.1230	0.2500
43	37.1230	3.4550	15.2920	1.9830	64.6630	8.2320	2.8140	0.3310	2.1330	0.2600
44	37.1630	3.4550	15.3920	1.9730	64.7530	8.2520	2.8040	0.3310	2.1230	0.2500
45	37.2330	3.4650	15.3420	1.9730	64.5730	8.2510	2.8040	0.3410	2.1230	0.2600
46	37.0330	3.4450	15.3620	1.9830	65.0840	8.2520	2.8950	0.3400	2.1330	0.2500
47	37.0940	3.4550	15.3420	1.9720	64.2420	8.2720	2.9740	0.3410	2.1230	0.2610
48	37.1330	3.4450	15.3220	1.9730	64.5130	8.2320	2.8640	0.3310	2.2130	0.2500
49	37.1530	3.4550	15.2920	1.9830	64.2930	8.2920	2.8450	0.3400	2.1630	0.2500

50	37.2030	3.4650	15.4430	1.9830	64.4320	8.2520	2.8840	0.3400	2.1530	0.2510
51	37.0530	3.4550	15.3220	1.9830	65.0540	8.2420	2.9240	0.3510	2.2030	0.2500
52	36.9430	3.4350	15.3120	1.9730	64.5330	8.2310	2.9250	0.3400	2.1630	0.2700
53	36.9430	3.4350	15.3020	1.9730	64.4130	8.2320	2.9240	0.3510	2.2840	0.2600
54	37.0140	3.4350	15.2920	1.9720	64.4420	8.2510	2.8850	0.3300	2.1930	0.2700
55	37.1230	3.4650	15.4120	1.9730	64.4420	8.2820	2.8440	0.3400	2.2330	0.2610
56	37.0630	3.4550	15.3920	1.9730	64.4330	8.2320	2.8840	0.3510	2.1440	0.2700
57	36.8930	3.4450	15.2720	1.9720	64.9330	8.2820	2.9140	0.3510	3.4650	0.2500
58	36.9230	3.4450	15.3020	1.9720	64.3230	8.2320	2.9640	0.3500	2.2330	0.2610
59	37.0340	3.5150	15.2910	1.9820	64.1720	8.2220	2.9540	0.3410	2.1930	0.2700
60	37.2530	3.4750	15.4120	1.9830	64.2920	8.2220	2.9740	0.3400	2.2130	0.2810
61	37.4540	3.4650	15.5220	2.0030	64.6130	8.2420	2.9040	0.3410	2.1930	0.2600
62	37.2230	3.4850	15.3620	1.9830	64.3830	8.2620	3.0040	0.3400	2.3030	0.2510
63	36.9730	3.4550	15.3220	1.9830	64.4930	8.2420	2.9140	0.3410	2.1430	0.2900
64	37.0140	3.4650	15.3320	1.9820	64.7130	8.2720	2.9150	0.3800	2.1730	0.2600
65	37.2230	3.4650	15.3620	1.9730	64.3930	8.2610	2.8640	0.3510	2.2330	0.2600
66	37.0030	3.4450	15.3120	1.9830	64.6630	8.2410	2.9540	0.3810	2.1930	0.2500
67	37.1340	3.4550	15.3730	1.9820	64.2320	8.2820	2.8740	0.3300	2.1830	0.2510
68	37.2330	3.4550	15.3520	1.9830	64.7030	8.2320	2.8640	0.3410	2.2740	0.2600
69	36.9530	3.4540	15.2710	1.9720	64.4320	8.2120	2.8840	0.3410	2.1630	0.2500
70	36.9740	3.4450	15.3420	1.9720	64.5730	8.2520	2.9140	0.3600	2.2130	0.2910
71	37.1130	3.4550	15.3420	1.9730	64.8130	8.2320	2.9850	0.3600	2.1630	0.2500
72	37.0330	3.4550	15.4120	1.9830	64.5430	8.2520	2.9640	0.3510	2.1930	0.2710
73	37.2840	3.4550	15.3730	1.9820	64.5430	8.5020	2.9040	0.3510	2.2030	0.2600
74	36.9930	3.4550	15.3020	1.9730	64.4430	8.2420	2.9750	0.3600	2.1730	0.2500
75	37.1440	3.4550	15.2920	1.9730	64.7130	8.2320	2.9140	0.3410	2.2640	0.2500
76	37.1030	3.4650	15.3020	1.9730	64.2120	8.2220	2.9350	0.3500	2.1930	0.2600
77	37.0840	3.4550	15.4930	2.0520	64.2120	8.2320	2.9740	0.3410	2.2330	0.2810
78	37.1830	3.4650	15.4320	1.9730	64.1730	8.2410	2.9140	0.3310	2.1730	0.2600
79	37.0240	3.4650	15.3120	1.9820	64.2130	8.2220	2.8640	0.3400	2.1230	0.2510
80	37.3530	3.4650	15.3120	1.9830	64.9730	8.2520	2.9740	0.3310	2.1540	0.2700
81	37.0630	3.4750	15.3830	1.9820	64.3620	8.2410	2.9850	0.3400	2.2330	0.2700
82	37.4040	3.4850	15.3720	1.9830	64.4130	8.2320	2.8240	0.3300	2.1730	0.2500
83	37.0330	3.4450	15.3220	1.9730	64.7330	8.2120	2.9840	0.4910	2.2130	0.2700
84	37.0030	3.4550	15.3920	1.9730	64.5830	8.2410	2.9040	0.3410	2.2140	0.2600
85	37.0740	3.4550	15.2920	1.9730	64.3630	8.2220	2.9350	0.3500	2.1630	0.2500
86	37.0930	3.4550	15.3620	1.9830	64.3020	8.2220	2.8840	0.3500	2.1730	0.2510
87	37.0130	3.4550	15.2620	1.9730	64.5530	8.2520	2.8540	0.3410	2.1530	0.2600
88	37.1640	3.4450	15.2920	2.0430	64.8430	8.2220	2.9340	0.3610	2.1430	0.2500
89	37.0630	3.4550	15.3720	1.9830	64.4630	8.2220	2.8940	0.3900	2.2130	0.2510
90	37.2030	3.4550	15.3820	1.9730	64.6020	8.2310	2.9240	0.3510	2.1940	0.2500
91	37.0840	3.4650	15.3330	1.9720	64.9730	8.2720	2.9250	0.3400	2.1930	0.2700
92	37.0430	3.4650	15.3720	1.9830	64.6430	8.2520	3.2350	0.3500	2.3640	0.3000
93	37.0030	3.4650	15.3220	1.9830	64.5330	8.2420	2.9440	0.3310	2.3240	0.2800
94	37.0130	3.4750	15.2720	1.9730	64.5830	8.2420	2.9140	0.3400	2.2330	0.2500
95	37.1230	3.4550	15.5420	1.9830	64.5930	8.2520	2.8440	0.3300	2.2130	0.2500
96	37.0030	3.4650	15.3520	1.9630	64.4620	8.2320	2.9040	0.3410	2.1630	0.2700
97	37.1030	3.4550	15.3620	1.9830	64.3630	8.2520	2.9040	0.3410	2.1230	0.2600
98	37.2230	3.4550	15.4020	1.9830	64.5230	8.2210	2.8040	0.3400	2.1830	0.2500
99	37.1630	3.4650	15.3520	1.9720	64.6730	8.2320	2.8140	0.3400	2.1330	0.2510
100	37.1140	3.4450	15.5620	1.9830	64.7530	8.2320	3.0340	0.3810	2.1830	0.2510

APPENDIX I

For *carphone.qcif*:

Peak Signal-to-Noise Ratio (PSNR)										
W	**7**				**15**				**7**	
Frame	**MMSE 4**	**MMSE 16**	**MAD 4**	**MAD 16**	**MAD 4**	**MAD 16**	**TTS 4**	**TSS 16**	**TSS 4**	**TSS 16**
2	46.90	37.03	37.62	35.72	37.98	35.74	33.00	33.13	33.13	33.28
3	46.61	36.51	37.77	35.65	38.09	35.69	32.59	32.71	32.81	32.71
4	47.00	36.29	38.03	35.61	38.53	35.76	32.25	31.97	32.41	31.92
5	45.39	35.11	36.95	34.52	37.72	34.65	31.80	31.40	32.16	31.49
6	44.97	34.95	36.76	34.29	37.46	34.46	31.85	31.28	32.21	31.49
7	45.42	34.70	36.51	33.95	37.39	34.18	32.06	31.82	32.33	31.97
8	44.76	34.45	36.57	33.82	37.26	33.93	31.85	31.37	32.38	31.93
9	43.95	33.93	36.00	33.15	36.84	33.34	31.63	30.82	32.17	31.31
10	43.44	33.70	35.85	32.85	36.65	32.89	31.66	30.73	31.73	31.00
11	43.26	33.53	35.90	32.83	36.75	32.94	31.69	30.50	31.97	30.99
12	43.36	34.11	36.30	33.27	36.97	33.54	31.68	30.75	32.04	31.05
13	42.77	33.42	35.59	32.65	36.47	32.92	31.55	30.54	31.87	31.00
14	42.78	33.40	35.70	32.62	36.60	32.85	31.68	30.54	31.95	30.92
15	43.13	33.89	36.12	33.15	36.83	33.41	31.69	30.73	32.00	31.21
16	42.95	33.59	35.77	32.77	36.64	33.02	31.55	30.71	31.96	30.94
17	43.54	34.20	36.56	33.41	37.23	33.60	31.60	30.94	31.98	31.16
18	43.44	33.71	36.23	33.01	36.90	33.15	31.80	30.98	32.13	31.13
19	43.10	33.51	35.71	32.74	36.57	32.88	31.60	30.80	31.84	30.93
20	42.96	33.65	35.77	32.77	36.61	32.91	32.07	31.29	32.10	31.19
21	42.71	33.65	35.54	32.69	36.42	32.83	31.99	31.19	31.93	31.15
22	42.32	33.78	35.81	32.85	36.60	33.05	31.65	30.86	31.72	30.86
23	41.28	33.01	35.22	31.93	36.20	32.16	31.42	30.55	31.45	30.52
24	41.18	33.07	35.29	32.00	36.20	32.27	31.50	30.41	31.57	30.49
25	40.56	32.78	34.94	31.89	35.98	32.20	31.31	30.34	31.42	30.26
26	40.57	32.55	34.97	31.73	36.06	32.06	31.37	30.34	31.35	30.31
27	40.31	32.25	34.72	31.43	35.78	31.78	31.22	30.04	31.29	30.08
28	39.95	32.10	34.53	31.21	35.70	31.45	31.24	30.04	31.16	30.07
29	39.92	31.98	34.34	31.18	35.60	31.40	31.13	29.84	30.98	29.80
30	40.29	32.17	34.58	31.30	35.93	31.64	31.23	29.91	31.12	30.08
31	39.65	31.82	34.30	30.94	35.59	31.41	31.02	29.73	30.92	29.85
32	39.01	31.63	34.01	30.83	35.48	31.16	31.17	29.74	30.82	29.88
33	38.68	31.71	33.96	30.80	35.47	31.17	31.17	29.74	30.77	29.75
34	38.84	31.69	33.93	30.86	35.56	31.24	31.10	29.71	30.77	29.80
35	38.57	31.84	34.01	30.98	35.57	31.47	31.08	29.79	30.71	29.69
36	38.10	31.53	33.67	30.63	35.15	31.07	30.95	29.94	30.58	29.59
37	37.86	31.49	33.47	30.50	35.15	31.05	31.00	29.74	30.44	29.45
38	37.78	31.44	33.52	30.56	35.23	31.17	30.94	29.82	30.44	29.43
39	38.03	31.61	33.71	30.72	35.39	31.29	31.05	29.97	30.57	29.48
40	38.00	31.60	33.66	30.76	35.38	31.34	31.01	29.97	30.51	29.51
41	37.84	31.52	33.55	30.56	35.35	31.11	31.00	29.77	30.51	29.43
42	37.99	31.60	33.59	30.63	35.35	31.27	31.06	29.80	30.52	29.42
43	38.38	31.69	33.79	30.78	35.58	31.39	31.11	29.96	30.63	29.56
44	38.68	31.85	33.94	30.95	35.70	31.56	31.13	30.01	30.69	29.64
45	38.57	31.73	33.90	30.84	35.68	31.46	31.11	29.96	30.69	29.61
46	38.69	31.74	33.92	30.77	35.60	31.37	31.09	29.90	30.66	29.59
47	38.78	31.98	34.06	31.00	35.67	31.67	31.09	30.12	30.79	29.71

48	38.94	32.00	34.10	31.11	35.79	31.75	31.10	30.11	30.73	29.75
49	38.75	31.67	33.93	30.78	35.65	31.33	30.96	29.95	30.71	29.66
50	38.55	31.75	33.74	30.75	35.56	31.34	30.95	29.90	30.69	29.64
51	38.64	31.75	33.82	30.80	35.56	31.37	31.03	29.95	30.71	29.63
52	38.66	31.72	33.84	30.78	35.53	31.24	30.93	29.84	30.64	29.53
53	38.89	31.82	33.92	30.93	35.65	31.41	30.98	29.84	30.74	29.69
54	38.81	31.92	34.09	30.98	35.71	31.50	30.91	29.83	30.73	29.62
55	38.46	31.77	33.71	30.72	35.63	31.29	30.99	29.83	30.61	29.49
56	38.64	31.81	33.76	30.73	35.52	31.20	31.05	29.92	30.63	29.63
57	39.08	32.22	34.18	31.10	35.80	31.61	31.19	30.02	30.83	29.87
58	39.02	31.89	33.86	30.76	35.52	31.34	31.12	29.94	30.72	29.76
59	38.58	31.89	33.70	30.74	35.54	31.39	31.08	30.14	30.77	29.84
60	38.37	31.79	33.57	30.62	35.38	31.18	31.04	30.06	30.70	29.67
61	38.06	31.61	33.44	30.50	35.29	31.00	30.99	30.01	30.65	29.60
62	37.96	31.83	33.61	30.67	35.65	31.34	31.01	30.01	30.54	29.64
63	37.63	31.72	33.39	30.59	35.38	31.17	30.93	29.98	30.59	29.71
64	37.46	31.62	33.17	30.46	35.23	31.02	30.94	29.84	30.46	29.65
65	37.32	31.51	33.09	30.37	35.12	31.01	30.86	29.75	30.43	29.68
66	37.49	31.52	33.20	30.35	35.17	31.02	30.84	29.82	30.40	29.58
67	37.60	31.44	33.08	30.30	34.97	30.94	30.79	29.66	30.39	29.48
68	37.47	31.37	33.06	30.30	35.00	30.87	30.82	29.72	30.36	29.54
69	37.37	31.39	33.00	30.26	34.93	30.78	30.80	29.72	30.26	29.43
70	37.45	31.48	33.08	30.37	35.07	30.80	30.91	29.80	30.36	29.53
71	37.58	31.53	33.17	30.47	35.01	30.88	30.99	29.72	30.49	29.67
72	37.55	31.74	33.35	30.61	35.28	30.88	30.97	29.74	30.64	29.85
73	37.49	31.65	33.22	30.54	35.05	30.96	30.95	29.77	30.56	29.82
74	37.37	31.67	33.30	30.48	35.24	30.99	31.00	29.75	30.56	29.75
75	37.33	31.79	33.37	30.66	35.27	31.25	30.98	29.76	30.67	29.72
76	37.16	31.64	33.23	30.45	35.13	31.07	30.98	29.91	30.66	29.73
77	36.91	31.69	33.08	30.46	34.95	31.09	30.91	29.84	30.52	29.80
78	36.55	31.65	32.98	30.36	35.08	31.11	30.83	29.80	30.43	29.66
79	36.17	31.27	32.69	30.12	34.88	30.75	30.81	29.65	30.30	29.35
80	36.03	31.16	32.64	30.07	34.83	30.69	30.77	29.57	30.23	29.32
81	35.91	31.31	32.72	30.19	35.00	30.88	30.71	29.67	30.15	29.32
82	35.67	30.97	32.49	29.86	34.74	30.61	30.69	29.54	30.05	29.21
83	35.34	30.85	32.43	29.73	34.71	30.56	30.67	29.51	30.02	29.09
84	35.38	30.85	32.36	29.70	34.66	30.71	30.61	29.48	29.98	29.06
85	35.31	30.71	32.33	29.66	34.51	30.68	30.61	29.37	29.89	28.97
86	35.27	30.63	32.22	29.69	34.41	30.58	30.54	29.34	29.91	28.88
87	35.00	30.62	32.05	29.45	34.17	30.47	30.47	29.29	29.79	28.77
88	34.87	30.54	31.96	29.37	34.14	30.39	30.34	29.20	29.68	28.70
89	34.99	30.59	32.01	29.50	34.14	30.50	30.41	29.26	29.76	28.74
90	35.20	30.78	32.28	29.74	34.38	30.87	30.54	29.35	29.85	28.79
91	35.36	30.75	32.25	29.75	34.40	30.79	30.59	29.40	29.91	28.84
92	35.32	30.70	32.21	29.68	34.23	30.72	30.55	29.42	29.92	28.95
93	35.28	30.81	32.14	29.67	34.31	30.65	30.55	29.44	29.87	28.93
94	35.40	30.95	32.26	29.77	34.38	30.77	30.63	29.54	29.96	28.99
95	35.65	30.90	32.38	29.89	34.48	30.80	30.64	29.58	30.00	29.01
96	35.63	30.84	32.41	29.85	34.60	30.91	30.64	29.57	30.04	29.02
97	35.53	30.78	32.30	29.73	34.54	30.71	30.58	29.51	30.01	29.02
98	35.45	30.78	32.31	29.67	34.46	30.67	30.65	29.46	29.90	28.87
99	35.50	30.76	32.32	29.72	34.42	30.75	30.66	29.54	29.99	28.97
100	35.75	31.00	32.50	29.94	34.72	30.99	30.66	29.60	30.00	29.05

Elapsed Time (s)										
W	7				15				7	
Frame	MMSE 4	MMSE 16	MAD 4	MAD 16	MAD 4	MAD 16	TTS 4	TSS 16	TSS 4	TSS 16
2	40.2180	3.8850	15.6520	2.0830	63.7720	8.2510	2.8740	0.3610	2.2440	0.2600
3	39.1860	3.6050	15.5730	2.1130	70.9720	8.2420	2.9940	0.3500	2.3430	0.2510
4	40.1580	4.1060	17.2050	2.0830	66.2350	8.6130	2.8740	0.3610	2.2230	0.2700
5	40.5880	3.8260	16.6940	2.0830	65.9350	8.8620	2.9640	0.3510	2.3830	0.2510
6	41.0490	4.0860	16.5240	2.0830	65.5940	8.2720	2.8540	0.3810	2.2530	0.2800
7	39.2860	3.4250	16.4130	2.2240	66.1060	8.5420	3.0850	0.3600	2.1230	0.2500
8	41.5600	3.6350	16.9540	2.3330	65.0230	8.4830	3.0650	0.3400	2.2730	0.3000
9	39.6270	3.4350	15.3620	1.9730	66.9760	8.2220	3.1440	0.4110	2.2840	0.2700
10	38.1450	3.6550	15.3820	2.1330	66.2660	8.6020	2.9440	0.3510	2.2130	0.2710
11	39.3570	4.4160	16.5240	1.9830	66.4760	8.2920	2.9040	0.3310	2.2530	0.2600
12	42.9810	3.4950	17.8460	2.6140	64.4630	8.2520	2.8640	0.3400	2.2630	0.2810
13	40.8490	3.6850	16.9440	2.2930	63.5410	8.2520	3.0850	0.3600	2.3240	0.2900
14	39.2060	3.5960	15.8130	1.9830	63.4910	8.2120	3.1340	0.3400	2.2840	0.2900
15	37.8650	3.5550	15.7420	2.0530	63.4910	8.2220	2.8440	0.3310	2.3940	0.2900
16	39.0360	3.6550	16.1630	1.9730	64.0420	8.2220	2.8240	0.3410	2.1130	0.2500
17	38.4850	3.6360	16.0830	2.1830	63.4810	8.2220	2.9340	0.3510	2.1330	0.2500
18	39.3470	3.6050	16.0930	2.2130	63.6020	8.2910	3.0040	0.3410	2.2530	0.2610
19	38.0250	3.5150	16.6440	2.2930	65.4240	8.6130	3.0350	0.3800	2.3630	0.2500
20	37.4840	3.4850	15.5530	1.9920	67.4870	8.2220	2.9140	0.3410	2.3130	0.2500
21	38.1750	3.4450	15.8130	2.0030	63.5210	8.2120	2.9340	0.3400	2.2430	0.2610
22	37.7440	3.5850	15.1920	1.9630	63.5820	8.2620	2.8940	0.3810	2.1440	0.2600
23	37.5440	3.4550	15.7230	1.9730	63.4110	8.2020	2.9340	0.3410	2.1430	0.2500
24	37.2840	3.9050	15.4320	1.9730	63.2810	8.1920	2.8540	0.3400	2.1530	0.2610
25	38.4450	3.4450	16.0130	2.1130	63.4910	8.1820	2.8840	0.5010	2.2230	0.2710
26	38.3450	3.6350	15.8330	2.1430	63.3910	8.1820	2.8040	0.3310	2.1130	0.2500
27	38.6960	3.8150	15.8930	2.2330	63.3810	8.2020	2.9340	0.3500	2.2530	0.2610
28	38.0440	3.4750	15.9430	2.0030	65.5440	8.4120	2.8940	0.3410	2.2040	0.2600
29	38.9560	3.5850	16.1540	2.1130	65.9850	8.7830	2.8140	0.3310	2.1530	0.2500
30	38.1850	3.4650	15.3620	1.9630	67.5270	8.4720	2.9750	0.3400	2.0930	0.2500
31	38.8060	3.4650	16.2730	1.9930	65.0030	8.3020	2.9940	0.3310	2.2840	0.2800
32	38.6460	3.6550	15.5430	2.0620	68.0080	8.4220	2.8040	0.3410	2.2630	0.2600
33	40.4480	3.5760	15.9030	1.9830	64.7430	8.2020	2.9340	0.3310	2.3840	0.2500
34	37.2940	3.4450	15.6320	2.0030	66.0250	9.0630	2.9940	0.3610	2.1130	0.2500
35	38.8460	3.6950	15.6820	1.9930	65.8850	8.4620	2.8340	0.3400	2.2330	0.2610
36	38.1550	3.5050	15.8230	1.9730	66.1950	8.6130	3.0240	0.3610	2.1030	0.2510
37	40.9590	3.9250	17.0650	2.2030	65.4740	8.3920	2.8250	0.3300	2.3530	0.2500
38	40.6080	3.5150	17.4160	2.2730	63.7310	8.1920	3.0940	0.3610	2.1030	0.2510
39	39.0870	3.4550	16.1140	1.9820	63.8420	8.3720	3.1750	0.4010	2.1530	0.2600
40	38.6160	3.6950	15.7520	2.0230	65.1440	8.8030	2.9450	0.3400	2.3130	0.2800
41	38.8860	3.4350	17.9860	1.9830	67.5070	8.4920	2.9840	0.3410	2.3940	0.2700
42	38.3850	3.4750	15.1720	1.9830	64.8830	8.2220	2.9140	0.3800	2.2130	0.2500
43	39.7370	3.4960	15.5220	2.0830	63.4910	8.2120	3.0540	0.3310	2.3640	0.2500
44	37.6340	3.4350	15.4920	2.1830	64.1620	8.1920	2.9440	0.3810	2.2830	0.2810
45	36.9630	3.4450	15.2620	1.9730	63.4810	8.1820	2.8750	0.3600	2.3430	0.2900
46	38.5060	3.7650	15.3420	2.0030	63.4510	8.2020	2.9440	0.3510	2.2030	0.2510
47	38.5750	3.4650	15.9830	1.9630	63.4410	8.1920	2.7940	0.3310	2.1630	0.2500
48	38.8960	3.4150	15.6220	2.0030	63.4110	8.2620	2.8940	0.3810	2.1130	0.2500
49	37.5240	3.4350	15.2820	2.0030	67.3870	8.5420	2.9640	0.3400	2.3830	0.2610

50	36.9230	3.4350	15.3020	1.9730	63.7920	8.2420	2.9340	0.3310	2.2130	0.2500
51	36.7530	3.4450	15.2620	1.9620	69.1190	8.2420	2.9550	0.3500	2.1630	0.2500
52	37.0930	3.4650	15.1820	1.9730	64.8530	8.2120	2.8840	0.4010	2.2130	0.2500
53	38.3550	3.6160	15.2720	1.9630	63.5620	8.2110	2.9040	0.3500	2.4130	0.2610
54	37.8350	3.5150	15.5620	2.0730	63.7420	8.2920	3.1450	0.3400	2.3440	0.2700
55	39.3660	3.6160	16.0430	2.1030	64.0620	8.2110	2.9940	0.3810	2.2640	0.2800
56	39.2660	3.5350	15.8320	2.1040	63.6820	8.1920	3.1040	0.3610	2.1730	0.2600
57	39.3060	3.5550	16.8740	2.3040	63.2110	8.2020	2.9550	0.3800	2.2730	0.2500
58	40.0980	4.0460	16.3040	2.3530	63.6210	8.4120	2.8840	0.3710	2.3030	0.2700
59	38.7860	3.8150	16.8240	2.0430	66.2560	8.5920	2.9350	0.3500	2.3630	0.2700
60	39.3960	3.5150	16.0440	2.1630	63.8010	8.2320	2.9440	0.3510	2.3940	0.2400
61	38.5650	3.7260	15.7930	1.9630	63.9620	8.2620	2.8540	0.3400	2.1730	0.2610
62	39.9480	3.4150	16.9140	2.1730	63.6310	8.2020	2.8140	0.3410	2.4640	0.2600
63	38.6450	3.5860	16.0030	1.9730	63.2710	8.2110	2.9450	0.3500	2.1430	0.2600
64	40.0370	3.5450	15.9130	2.1630	63.3910	8.2320	3.0440	0.3410	2.1530	0.2510
65	38.6060	3.5550	15.9230	2.0530	63.6210	8.2220	2.8940	0.3410	2.1930	0.2700
66	38.6150	3.4850	15.8830	2.0330	63.5210	8.2420	2.9750	0.3300	2.1730	0.2600
67	39.2560	3.8360	15.3730	2.0630	63.8520	8.2020	2.8140	0.3400	2.1130	0.2510
68	37.9340	3.4250	15.5920	1.9730	63.3710	8.1810	2.9440	0.3810	2.1530	0.2810
69	40.3280	4.0360	15.8820	2.0730	63.9720	8.2120	3.0250	0.3500	2.2030	0.2700
70	37.8140	3.6060	17.3350	1.9830	63.4210	8.1920	2.8940	0.3400	2.1430	0.2510
71	37.3830	3.5250	15.9030	1.9830	63.4410	8.2110	3.0640	0.3510	2.2540	0.2800
72	39.7770	3.6060	15.7130	2.0930	63.6010	8.2310	2.8050	0.3300	2.1230	0.2500
73	39.5270	3.5750	15.8020	2.0630	63.7720	8.2820	2.8050	0.3300	2.1030	0.2500
74	38.4560	3.5250	15.2820	1.9730	63.5910	8.2020	2.9050	0.3400	2.1030	0.2600
75	39.8080	3.5050	15.6720	2.1130	63.4420	8.2010	2.8840	0.3510	2.2230	0.2610
76	38.7860	3.4250	15.2020	1.9820	63.7420	8.2120	3.0750	0.3300	2.2730	0.2700
77	39.6770	3.7550	15.6630	1.9830	63.2810	8.2120	3.0240	0.3910	2.1730	0.2910
78	41.5500	3.4950	16.4540	2.0330	64.3120	8.1920	2.9750	0.3700	2.2630	0.2700
79	38.2550	3.9050	15.3120	2.0930	64.7930	8.4130	2.8540	0.3310	2.2240	0.2500
80	39.4470	3.4350	15.2920	1.9930	64.4130	8.3520	2.8640	0.3310	2.1230	0.2500
81	38.5550	3.5360	15.3320	1.9830	67.4570	8.7030	2.9550	0.3700	2.2330	0.2900
82	39.3360	3.5250	16.1730	2.0230	67.7070	8.4120	2.8940	0.3710	2.2130	0.2510
83	39.8370	3.8160	16.2130	2.0730	66.4060	8.3920	2.9640	0.3510	2.2530	0.2500
84	40.9590	4.0160	15.4120	2.1130	67.1460	8.9130	3.0550	0.3600	2.1430	0.2500
85	40.9880	3.5260	16.4430	2.1130	66.3960	8.7820	2.8040	0.3710	2.2130	0.2700
86	42.2310	3.5750	16.0030	2.2830	67.8580	8.5720	2.9450	0.3500	2.2330	0.2500
87	38.3250	3.4250	15.6120	2.0130	65.9050	8.7020	2.9740	0.3310	2.2530	0.2510
88	40.4580	3.5750	15.1220	1.9730	67.6570	8.6730	2.9740	0.3910	2.1530	0.2800
89	39.6570	3.6560	15.9730	1.9920	68.8790	8.3620	2.9640	0.3310	2.3730	0.2710
90	39.1260	3.8050	15.9330	1.9830	65.9150	8.4620	2.9340	0.3410	2.1530	0.2600
91	39.6470	3.7450	16.0440	2.2930	64.5830	8.4320	2.9350	0.3400	2.1830	0.2500
92	40.0380	3.8550	16.5340	2.1530	63.7520	8.4020	2.9040	0.4000	2.1230	0.2710
93	40.7190	3.4850	15.9530	2.0630	64.7630	8.3020	3.3350	0.4210	2.2830	0.2600
94	39.2760	4.2360	15.7830	2.0530	67.2160	8.3720	2.9140	0.3500	2.2930	0.2710
95	39.8470	3.6350	16.2230	2.0130	65.9650	8.8430	2.8540	0.3400	2.1930	0.2610
96	39.2170	3.5150	15.8030	2.1430	67.6380	8.7420	3.0750	0.3600	2.2930	0.2700
97	40.4680	5.2470	15.2520	2.0230	68.1980	8.9030	4.1960	0.3710	2.1540	0.2700
98	37.4140	3.4350	15.5420	2.0630	67.1370	8.3420	2.9250	0.3500	2.3330	0.2500
99	37.4340	3.4250	15.1820	1.9720	66.2950	8.3220	3.0940	0.3610	2.1930	0.2810
100	39.1960	3.4650	15.1720	1.9630	64.9640	8.8820	3.0750	0.3600	2.1730	0.2700

APPENDIX J

TSS Image Results for *claire.qcif*

Figure 4.1.4(i): Original frame 10 image *claire.qcif*

N = 4, W = 7

N = 16, W = 7

N = 4, W = 15

N = 16, W = 15

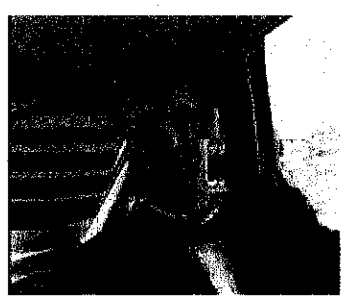
Figure 4.1.4(j): Predicted images with TSS *claire.qcif*

APPENDIX K

TSS Image Results for *carphone.qcif*

Figure 4.1.4(k): Original frame 10 image *carphone.qcif*

N = 4, W = 7



N = 16, W = 7

N = 4, W = 15

N = 16, W = 15

Figure 4.1.4(l): Predicted images with TSS *carphone.qcif*