Sudan University of Science & Technology

College of Graduate Studies

بسم الله الرحمن الرحيم

كلية الدراسات العليا

جامعة السودان للعلوم والتكنولوجيا

كلية الدراسات العليا

Approval Page

Name of Candidate: Fadlallah Ali Fadlalla Rabih

Thesis Title: Development of Video Streaming based on Frame Skipping Mechanism

تطوير طريقة لسريان وتوقف الفيديو باستخدام آلية محسنة لتخطي الأطر

Approved by:

1. External Examiner

Name: Iman AbuelMaaly Abdelrahman

Signature: [signature] Date: 30/9/2017

2. Internal Examiner

Name: Rashid A. Saeed

Signature: [signature] Date: 30/9/2017

3. Supervisor

Name: Osman Khalifa

Signature: [signature] Date: 30/9/2017

SUDAN UNIVERSITY OF SCIENCE AND TECHNOLOGYCOLLEGE OF GRADUATE STUDIES

Development an Approach for Video Streaming based on Enhanced Frame Skipping Mechanism

تطوير طريقة لسريان ودفق الفيديو بناء علي آلية محسنة لتخطي الأطر.

A thesis submitted in fulfillment of the Requirements
for the degree of Doctor of Philosophy in Computer Science

By:

Fadlallah Ali Fadlallah Rabih

Supervisor:

Prof. Dr. Othman O. Khalifa

September 2017

بسم الله الرحمن الرحيم

قال الله تعالى:

كَمَا أَرْسَلْنَا فِيكُمْ رَسُولاً مِّنكُمْ يَتْلُو عَلَيْكُمْ آيَاتِنَا وَيُزَكِّيكُمْ وَيُعَلِّمُكُمُ الْكِتَابَ وَالْحِكْمَةَ وَيُعَلِّمُكُم مَّا لَمْ تَكُونُواْ تَعْلَمُونَ(151) فَاذْكُرُونِي أَذْكُرْكُمْ وَاشْكُرُواْ لِي وَلاَ تَكْفُرُونِ(152).

سورة البقرة

DEDICATION

I hereby declare that this Thesis is the result of my own investigation, except where otherwise stated. I also declare that it has not been previously or concurrently submitted as a whole for any other degrees at Sudan University of Science and Technology or other institutions.

Fadlallah Ali Fadlallah Rabih

Signature_________________ Date____________________

ACKNOWLEDGEMENT

First of all, I would like to thank Allah (Subhanahu Wa Ta'ala) who supplies all my needs: of the patience, the perseverance, and the hope you offer all who would place their trust in you, and also, I would like to express my gratitude to the Sudan University Science and Technology, specifically to the Department of "Computer Science and Information Technology." for giving me the chance to do this thesis;

I would like also to express my appreciate to thank my supervisor, Prof. Dr. Othman O. Khalifa, for sharing his knowledge, guidance, patience and support me throughout this Thesis. I truthfully present my appreciation to anyone who supporting me to do this Thesis.

Lastly, I *would* like to thank my family for their continued support of my education.

I would like to thank the International Cooperation and Exchange Office (ICEO) and Kulliyyah of Engineering in International Islamic University Malaysia (IIUM) for giving me a change to complete my Thesis.

Fadlallah Ali Fadlallah

ABSTRACT

Video streaming (Vs) is one of the most important field in the global communication system, and data processing. It is always divided into server and receiver connected via network. The video streaming is a process that concerned with delivering video data from transmitter to one or more receivers, over unreliable network as quickly as possible, and with the least amount of data losses. In this work the possibilities to reduce the amount of video data that are transferred to end users over the network in video streaming are investigated and a video streaming technique comprised of a transmitter and receiver sides is proposed. To expand the flexibility and adaptability of proposal video streaming technique an operational parameters system was constructed, and their values range was defined. The proposed video streaming technique was then applied to three sample videos.

Before streaming the server side of the proposed technique reduced the amount of video data by identifying the less-motion-similarity between video frame sequences, which are considered as the Frame of Interest (FOI), and drop the frames of higher-motion similarity, which are considered as the non-Frame-Of Interest (non-FOI), based on operational parameter values, while the receiver-side will perform linear interpolation to re-construct non-FOI (Dropped frames) from reference. Then the quality of the resulting videos was measured and evaluated. To evaluate the quality level of the re-constructed videos that is obtained from the proposed techniques. The Peak-signal- to- noise- ratio-(PSNR) metric is used to compare the similarity between the original frames and the re-constructed videos frames.

The study concludes that by using the proposed video streaming technique it is possible to reduce the amount of transfer data by skipping frames on the sever side and re-constructing them on the receiver side.

المستخلص

عرض وسريان الفيديو هو واحد من المجالات الهامه والحيوية جداً في نظام الاتصالات العالمية ومعالجة البيانات. وينقسم عرض الفيديو إلى قسمين: الخادم والعميل وهما متصلين عبر شبكة. تهتم تقنية عرض الفيديو بسريان ودفق بيانات الفيديو (الصورة والصوت) من الخادم إلى العميل عبر الشبكة بالسرعة المطلوبه مع أقل قدرممكن من فقدان البيانات.

هذه الدراسة تقترح تقنية لدفق وسريان الفيديو في جانبي الارسال بين المرسل والمستقبل وذلك بتقليل كمية بيانات الفيديو المنقولة في جانب المرسل(المخدم) مع إمكانية إعادة استرجاعها في جانب المستقبل، وذلك بهدف تحسين جودة صورة الفيديو عند المستقبل النهائي. ولزيادة المرونة وقدرة النظام المقترح على تكيف الفيديو مع قيود الشبكة (مثل عرض النطاق الترددي)، تم تشًيد نظام المتغيرات التشغيلية و تحديد مدي لقيم هذه المتغيرات.

تم تطبيق تقنية دفق الفيديو المقترحة علي ثلاثه عينات من الفيديو. يقوم الخادم(المُرسل) قبل ارسال دفق الفيديو بتقليل عدد الأطارات المكونة لشريط الفيديو من خلال تطبيق قيم المتغيرات التشغيلية مجتمعه علي شريط الفيديو المدخل، وفي الجانب الآخر عند المستلم يتم استرجاع تلك اللإطارات التي تم اسقاطها عند المخدم (المرسل).

تم قياس جودة الفيديو الناتج أو الأطارات التي أعيد بناؤها عند طريق وحدة قياس قياسيه تسمي: نسبة الاشارة الي الضجيج القصوى، وخلصت هذه الدراسة إلى أنه باستخدام هذه التقنية المقترحة لسريان ودفق الفيديو، أنه من الممكن تقلل كمية بيانات الفيديو المنقوله من خلال إسقاط بعض اللأطرمن شريط الفيديو المدخل عند جانب المرسل وإعادة بناؤها عند جانب المستقبل.

Table of Contents

Contents

LIST OF TABLES

LIST OF FIGURES

List of ABBREVIATIONS

TERM	**ABBREVIATION**
16CIF	Sixteen Common Intermediate Format
3-D	Three Dimension (video)
4CIF	Four Common Intermediate Format
4GB	Four Giga Bytes
ACM	Association for Computing Machinery
AVC	Advanced Video Coding
AVI	Audio Video Interleave
B FRAME	Bidirectional Prediction Frame
BAD	Boundary of Absolute Difference
BDM	Block Distortion Measure
BM	Block Matching
B-SIZE	Block Size
CABAC	Context-based Adaptive Binary Arithmetic Coding
CB	Converted Background
CCIR	Commander's Critical Information Requirements
CCITT	CCITT, now known as the ITU-T (for Telecommunication Standardization Sector of the International Telecommunications Union)
CD-ROM	Compact Desk Read Only Memory
CIF	Common Intermediate Format
CPDT	Cascaded Pixel-Domain Transcoders
CPU	Central Processing Unit
DCT	Discrete Cosine Transformation
DDT	Discrete Cosine Transformation Domain Transcoders
DVD	digital versatile disc
DVS	Dynamic Voltage Scaling
EDT	Delayed decoding time
FOI	Frame Of Interest
FRM	Frame Reconstruction Method
FSC	Cumulative Frame Skipping
FSM	Frame Skipping Method
GHz	Gigahertz
GOP	Group of Pictures

HDTV	High Definition Television
HMC	Huff-Man Coding
I FRAME	Intra Frame Prediction
ICEO	International Cooperation and Exchange Office
IDCT	Inverse Discrete Cosine Transformation
IEEE	Institute of Electrical and Electronics Engineers
IIUM	International Islamic University Malaysia
ISDN	Integrated Services for Digital Network
ISO	International Standard for organization
ITU	International Telecommunication Union
MAE	Mean Absolute Error
MB	Micro-Block
MBPS	Magi Bit Per Second
M-CHANGE	Maximum change allowed
MCI	Motion compensation Interpolation
MO	Moving Object
MOS	Mean Opinion Value
MPEG	Moving Picture Expert Group
MCFI	Motion Compensation Frame Interpolation
MSE	Mean square Error
MSP	Maximum Signal Power
MV	Motion vector
NF	Number of Frames
NON-FOI	Non-Frame Of Interest
OLT	Open loop Transcoding
P FRAME	Inter Frame Prediction
PC	Personal Computer
PMV	Pixel-level Motion Vector
PSNR	Peak Signal to Noise Ratio
PSTN	Public Switched Telecommunications Network
QCIF	Quarter Common Intermediate Format
QP	Quantization Parameter
QSIF	Quarter input Source Format
RFM	Reconstruction Frame Mechanism
R-FRAME	Reference Frame

RGB	Red-Green-Blue colors
RLE	Run Length Encoding
RND	Randomly Frame Dropping
RPCA	Robust principle component analysis
RQs	Research Questions
SAD	Sum Absolute Different
SAE	Sum of Absolute Error
SB	Static Background
SDMC	The Square Different of Motion Changed
SDTA	The spatial Domain Transcoding Architecture
SFM	Skipping Frame Mechanism
SIF	The Input Source Format
SQCIF	Sub Quarter Common Intermediate Format
TSS	Three Steps Search
TV	Television
UB	Unconverted Background
VCEG	Video Coding Experts Group
VFI	Video Frame Interpolation
VLC	Variable length Coding
VS	Video Stream
WEB	Web
WI-FI	Wireless Fidelity
WI-MAX	Worldwide Interoperability for Microwave Access

GLOSSARY

Bitstream

It's a sequence (or time series)of bits which represents a stream of data. In video compression this sequence is produced by an encoder and is based on the codec used for compression.

Chrominance

Chrominance is the information that represents colour of an image without the brightness (Luminance).

Codec

Codec is a computer program (software) that is able to do the job of both encoding and decoding.

Decoder

Decoder is a software or a device that is operated to recover back the original information representation or row data from a previously encoded data.

Encoder

Encoder is a software or a device that is performed to convert information from one representation to another. It is also employed to obtain a good and compressed representation of a row data.

Granularity

Granularity describes the extent a system is broken down into smaller parts.

Luminance

Luminance is the information that represents the brightness in an image.

Chapter I: Introduction

1.1 Introduction

Recently, advanced types of interactive multimedia services and applications are proposed, such as video conferencing, distance learning, video telephony, and many others, which are used as a means of communication in a daily life, for work or special purposes [1]. Further, the amount of data sent and received through wireless networks rises daily along with the number of users. This in turn sparked the need for different services and technologies to accommodate the needs of the users. One of the most popular and problematic services over network is video streaming[2]. Video streaming over wireless-networks has become very popular nowadays due to the wide spread use of computer laptops, and mobile devices. The transmission rate of wireless channels varies from time to time and it depends on the available-bandwidth[3]. Wireless-channels are unable to guarantee the number of video frames that are transmitted to the user devices[4]. The video frames could be lost, delayed, or affected by errors and become unreadable by the decoder.

Therefore, this study proposed a technique to reduce the amount video data. The size of the video data can be reduced by identifying and transmitting the most important frames, which are considered as the Frame-Of-Interest (FOI) and drop the less-important frames, which are considered as the non-Frame-Of-Interest (non-FOI). The client side will performed linear interpolation to re-construct non-FOI from reference frames. The technique is designed in a way that can improve the end-user's perceived quality.

The streaming environment study is based on three entities; the streaming server, communication and the receiver device. In the streaming server, we implement our proposed video streaming technique for the chosen videos. The chosen videos are considered as professional test videos with different characteristics and different important regions.

This work contributes to domain of video streaming, in one hand it proposes a technique for streaming video over limited bandwidth network, on other the hand, it proposes a framework for video server, which will be constructed to reducing the amout of video data by skipping some frames on the server-side, and re-constructing them on the receiver side. The quality of re-constructed frames will evaluate by $PSNR$ metric.

The results show that, our technique significantly reduces the amount of data to be streamed over networks with acceptable video quality is provided to the user viewers.

1.2 Problem Statement

Video streaming over networks has become very popular nowadays due to the wide spread use of computer laptops, and mobile devices. The transmission rate of wireless channels varies from time to time and it depends on the available-bandwidth. Therefore, network channels are unable to guarantee the number of video frames that are transmitted to the user devices. So streaming video over network may lead to frame losses, delays, or affects by errors and become unreadable by the decoded. Our research goal is to provide a satisfactory video quality for real-time video streaming over networks with limited bandwidth. Real-time video streaming over network requires special technique that can overcome the loss of video frames by reduce the amount of data that are transmitted to the end user. This work, proposes a technique to cope with the bandwidth limited-network and to provide a smooth video playout with a satisfactory quality to the end users.

As a result of the aforementioned difficulties, it is needed to focus on this area of research in order to develop an efficient technique for reducing the amount of video data that needed to be transmitted over limited bandwidth network. This technique, search for the best effort to deliver high or better content viewing in order to satisfy the customers' needs in different requirements, with lower communications overhead.

1.3 Research Questions

This work can outline the problem statement in four main questions to map the work direction, and also using as guide the study towards to fulfill the main aim. This research will try to answer the following Research Questions (**RQ**):

RQ1: How can we use skip/drop frame concept to reducing the amount of video data on the server side?

RQ2: How can we re-construct the frames dropped on the receiver side?

RQ3: What is relationship between the number of frame skipped and videos quality?

RQ4: How does the quality of the videos change after reconstructing the video frames?

1.4 Research Significant/Motivation

Video streaming over wireless-network often suffers from bandwidth limitations, which may lead to frame losses. Video frame losses have a significant effect on the user-perceived quality. The main goal of this research is to provide a satisfactory video quality for real-time video streaming over wireless-networks with limited bandwidth.

Video streaming over wireless-network requires special technique that can overcome the loss of video frames by reduce the amount ofdata that are transmitted to the end user. In this work, we propose a technique to cope with the bandwidth limited network and to provide a smooth video playout with a satisfactory quality to the end users.

Furthermore, the proposed video streaming technique will realize multimedia services, and help the society in different fields such as learning systems, conferences, sports events, and many others applications. However, only a few works are found in the field of frames skipping from the video sequence and video adaptation.

1.5 Research Objectives

The main objective of this study is to improve the end user's perceived quality of video streaming over unreliable networks with limited bandwidth. This can be achieved by reducing the number of video frames that are streamed to the end user. Table.1.1 shows the sub-objectives to direct the study to achieve main goal:

Table 1.1: The Objectives of the study

Objective	Statements
Obj1	To investigate the current video frame skipping, frame interpolation, and video adaptation techniques that utilizes the network resources.
Obj2	To develop a scheme that could maintain a good stream over limited bandwidth network.
Obj3	To simulate the proposed scheme which efficiently utilizes the network limited bandwidth.
Obj4	To evaluate the performance of the proposed scheme and benchmark to the existing approaches.

1.6 Research Methodology and Tools

The proposed technique uses a mixture of both the qualitative and quantitative research methodologies in order to investigate the aforementioned research questions by means of effective literature review and also by developing a simulator to study the performance of the video quality after the frames lost. The literature review forms the qualitative research methodology, and the development of the simulator forms the quantitative part of research methodology. At the initial stage, a detailed literature study is made to understand the concept of adaptation video streaming over limited bandwidth networks and the issues concerning the mobile video streaming. A simulator is developed to study the performance smoothness of the video frames on the user device and to address the key issues with the streaming mechanism. The creation of the simulator is to validate our initial assumption of before streaming the server can reduce the account of frames and streamed to client, while receiving video the client will re-construct the dropped frames. This could improve the smoothness of the video frames on the user device. Tests are carried out with the aid of different test videos by using different data transmission rates and streaming the test videos in both colour and gray-scale formats.

The environment software that used to simulate the proposed video streaming technique is MATLAB version 2014a. MATLAB is interactive software that has become the most widely used software in academia and industry for modeling and simulation. MATLAB provides features that allow us to stream the video in real-time

In ordered to achieve the above objectives, the following steps ($approaches$) must be applied as shown in figure 1.1.

1. Studies the related researches works on the video streaming adaptation, and interpolation frame researches.
2. Propose enhanced approach method for streaming video over limited bandwidth networks.
3. Design and Develop a new scheme to solve these problems.
4. Implement the proposed video streaming management scheme.
5. Evaluate the proposed scheme by using analyses ($Matlab$) simulation.
6. This proposed scheme is benchmarked with standard-Benchmark $PSNR$.

7. Publication of the research results.

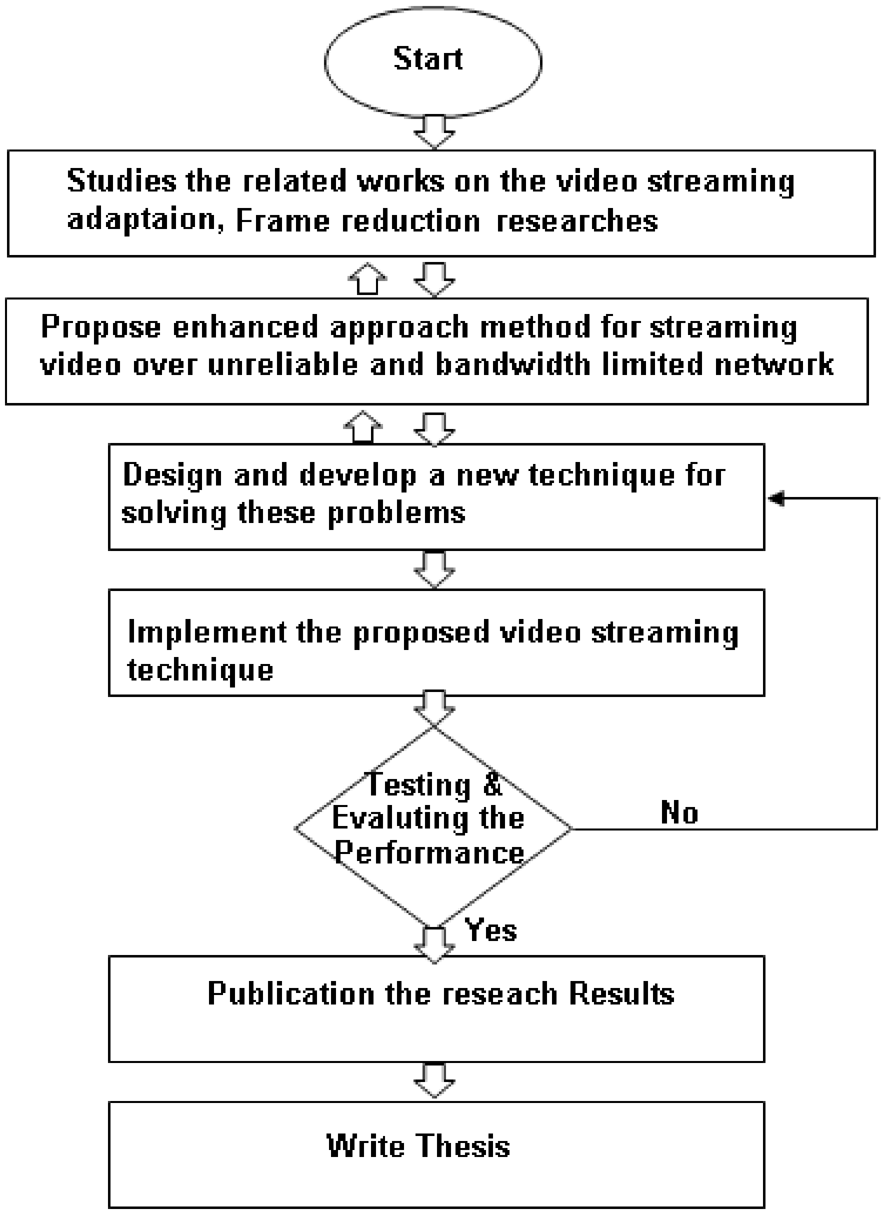


Figure 1.1: Research Methodology Diagram.

1.7 Scope of this Study

The field of video streaming is very wide. Therefore, this study is proposing a video streaming technique based on the concept of frame reduction. The size of video data can be reduced by identifying and transmitting the important frames, which is considered as the Frame-Of-Interest (FOI), and drop the less important frames, which is considered as the non-Frame-Of-Interest (non-FOI). The proposed technique is designed in a way that can improve the end-user's perceived quality. The proposed technique is concerned on delivering video data efficiently with acceptable video-quality that is provided to the user viewers. Some concepts related to video streaming modeling not in our scope such as voice management, and network transmission protocols. In the following section, the thesis will describe the common features of the proposed video streaming technique.

1.8 Research Hypothesis

This thesis addressed the problem of delivering video data efficiently to end user over limited bandwidth network. This technique will be considerable to reduce the size of video data that are transmitted to the user over limited channel-bandwidth, by:

1. Using motion estimation to determine the change in motion between consecutive video frames for reducing the frame counts in server side (i.e., skip/drop some video frames).
2. Using Linear Interpolation technique in the client side to re-construct video frames that were dropped in server side.

1.9 Research Validity

The results that are obtained from the system design that has been evaluated by an objective metric (PSNR) are quite encouraging, however, there are some limitations that need to be highlighted.

Regarding the simulation study, we design our technique in a way that allows us to stream the videos sequence from the server node to the client node without considering the communication setup time between them, which we assume to be constant in all our studies. Further, there are different types of failures that could occur during the streaming video, e.g., error transmission and intermediate node failure, which could have different effects on the streaming video. Therefore, only the frame losses are considered in the simulation environment, as it is one of the main focuses in this work.

The chosen videos for the experimental studies had different characteristics and motion levels. The number of videos are used in each experimental with different senior to proposed technique, as three videos are used in chapters four. If more videos are considered with different variety of objects and scenes, then we could have a better vision of the technique that is proposed in this work.

During the objective experimental study, the lengths of the test videos are extended by repeating the same video several times to achieve the purpose and the requirements of the scenarios that are proposed in chapters 3 and 4.

The videos that are obtained from each chapter could be measured by using objective PSNR metric to get an accurate evaluation. The PSNR index use a full-reference frames (original frames) metric, which should be available to evaluate the similarity between the full-reference frames and the test videos frames.

1.10 Research Outlines

This thesis is organized into five chapters.

The first **chapter** introduces the subject matter to the reader and presents the general outline of the work, which provides an overview of this thesis cover the problem statement of the study, motivation of the study, the objectives of the study, the research questions, the research methodology and the scope of the study. The rest of the thesis is organized as follows:

Chapter two outlines the background and discusses some issues of the video processing and the importance of video codec. In addition, this chapter reviews related studies of skipping frames, video transcoding, video adaptation, and frame interpolation. Lastly, the challenges and open issues were discussed.

In **chapter three**, the research design and method are described. The description of the research method involves the proposed structure of frame skipping Method (FSM), frame re-construct Method (FRM). The FSM and FRM the operation and diagrams, and flowchart.

Chapter four covers the results analysis of the proposed video streaming technique compared to $PSNR$ Standard, using $MATLAB\ R2014a$.

Lastly, **chapter five** presents the conclusion and recommendations of the study.
Finally, the thesis ends up with Chapter 5, which shows the conclusions and future work.

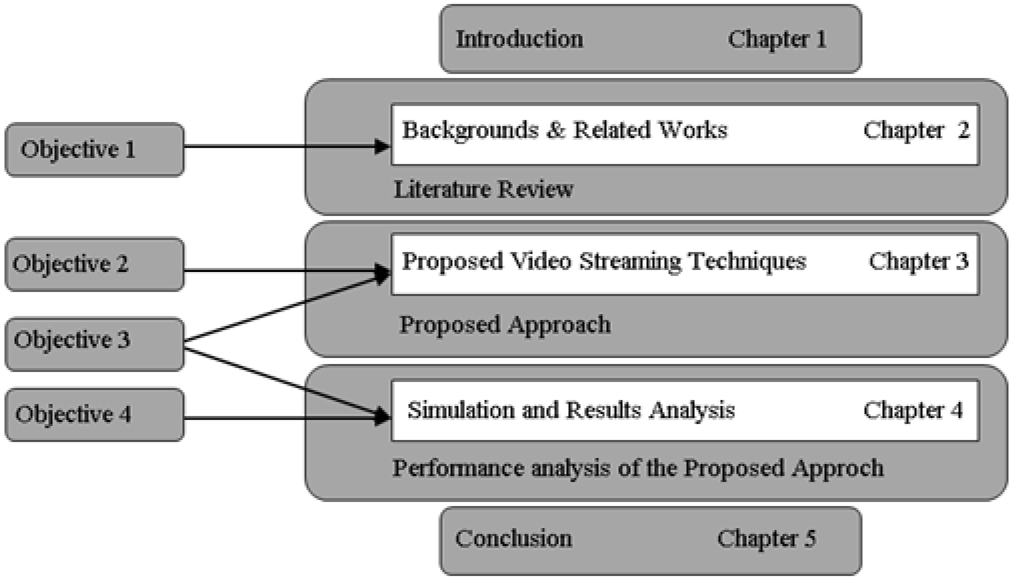


Figure 1.2: Organization of the thesis

Chapter II: Background and Related works

2.1 Introduction

High-quality digital video has been the goal of companies, researches and standard bodies over the last two decades[5][6]. Recent development of multimedia services and applications is due to improved digital video technologies. Highly efficient and scalable video compression formats enable many advanced types of interactive and distribution services, such as Digital TV broadcasting, Distance Learning, Video on Demand, Video Telephony and multipoint Video Conferencing. Digital video is an increasing technology which will continue to pervade business, networks and homes[7]. Getting digital video from its source (a camera or a stored clip) to its destination (a display) involves a chain of processes and components. The key processes in this chain are compression (encoding) and decompression (decoding), in which bandwidth intensive digital video is first reduced to a manageable size for transmission or storage, and then re-constructed for display[8]. Even with constant advances in storage and transmission capacity, compression is an essential component of multimedia services [9]. Video compression makes possible to use digital video in transmission and storage environments that would not support uncompressed video [10]. For example, current Internet throughput rates are not sufficient to handle uncompressed video in real-time (even at low frame rates and/or small frame size). Digital Versatile Disk (DVD) video storage would not be practical without video and audio compression. In addition, video compression enables a more efficient use of transmission and storage resources [10]. If a high bitrate transmission channel is available, video compression allows sending high-resolution compressed video or multiple compressed video streams instead of sending a single, low-resolution, uncompressed stream [4]. The objective is to provide a better image quality, more reliable and flexible solutions.

2.2 Video Coding

In the recent years, High-Quality 3D (or video) has become a goal for many companies, standard bodies, and researchers[6]. This is due to a development of multimedia services and applications, especially, within the development of digital video compression technologies [11]. These technologies have opened up avenues for the developing several applications of multimedia systems, such as video on demand, distance education, and broadcast digital TV and video multi-conferencing and video telephony, etc. [12]. further, many video coding standards are proposed for several applications, for example, MPEG-1 for storage media applications, H.263 typically used for low bitrate video communications, MPEG-2 outlined for high-quality video application, and Television broadcasting, MPEG-4 used for mobile applications, streaming 3-D (or video), and interactive multimedia applications, H.264 designed for high-compression applications [13][14]. Every standard principally describes a coded representation that defines a compressed form of visual data, and a technique of decoding the syntax to recreate the visual data, as shown in table 2.1.

Table: 2.1: An overview of video compression standards[15]**.**

Standards	**Applications**	**Bit-rate**
H.261	Video teleconferencing over $ISDN$	64-kbs
MPEG-1	Video on digital storage media ($CD\text{-}ROM$)	1.5-mbs
MPEG-2	Digital TV	2-20mbs
H.263	Video telephony over $PSTN$	>33.6kbs
MPEG-4	Multimedia over internet, object based coding	Variable
H.264/MPEG4	Improved video Compression	10-100 kbs
MPEG-7	Content description multimedia database searches	Variable
MPEG-21	Multimedia terminal specification	Variable

The Advance Video Coding (AVC) for H.264 has been developed jointly by ITU-T's Video Coding Experts Group (VCEG) and ISO/IEC's Moving Picture Experts

Group (MPEG). The video coding standard MPEG-4/H.264 (AVC)[16] removes redundancy or similarities between the neighboring frames in the video sequence [17]. There are two main types of redundancies present in video frame sequences: spatial redundancies and temporal redundancy[18] [4].

The spatial characteristics of the video scene, such as the one shown in Figure 2.1, that are relevant for video processing and compression are, e.g., texture variation within the scene, the number, colour and shape of objects. Temporal characteristics are, e. g., object motion, changes in illumination, and camera movement [19].

Temporal-quality means to the number of frames per second, where the motion in a scene appears smoother if more frames per second are played [20].The transmitted frames should be received and played according to their deadlines, otherwise the video is frozen. Spatial-quality can be expressed as the number of pixels in the video frame, where the texture frame provides information about the spatial segmentation or selected area of interest in the video frame [10]. The size of the video frame resolution can be changed to fit into different display screens. The resolution can be changed for both reduction and expansion, by removing and adding pixels in different parts of the video frame as in[10]. The video frame is blurred or juddered if a pixels or blocks of pixels are affected by artifacts. Both temporal and spatial quality, as shown in Figure 2.2, are usually determined before video encoding, preventing dynamic trade-offs during the encoding process[21] [22].

H.264 (AVC) is the most popular standard for video coding, and it can provide an excellent compression ratio [23]. Video transmission over error prone environments and limited bandwidth channels will affect the video quality; therefore there is a need for video scalability [13]. Video scalability refers to the removal of parts in the video stream in order to adapt it to different needs according to the end-user's specification and the terminal capabilities, as well as the network's condition. Scalable Video Coding (SVC) is an extension of the H.264 (AVC) standard that supports temporal and spatial scalability. Temporal scalability refers to the number of frames that can be removed from the streaming video sequence. Spatial scalability refers to scalability with respect to the resolutions of the video frames [23].

Figure 2.1: Still image from a video scene.

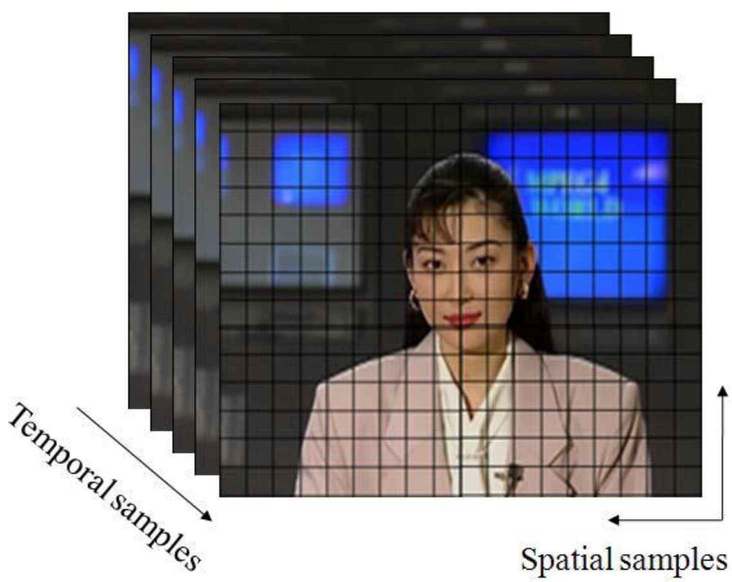


Figure 2.2: Spatial and temporal sampling of a video sequence.

The open source ffmpeg codec software is used in this thesis to encode/decode the test videos, as in [24]. The reason of using the ffmpeg codec is to study the effect of the proposed technique on the video size. The compressed video sizes are compared between the original videos and to the videos that are obtained from the proposed technique [24].

2.3 Video Compression

A compression is a software process generally is employed for converting (encoding) the amount of data into a fewer bit as shown in figure 2.3. The data might contain some elements that do not provide any relevant information: so-called *data redundancy*. Data redundancy is a central issue in compression process. As shown in figure 2.4, the compression process enables storage and transmission of the data efficiently[25] [26].

The inverse process is called decompression (decoding). Compressions normally consist of a complementary pair system encoder and decoder [27][28]. Spite of only encoding separate frames individually can first predict a frame based on the preceding frame and code only the change in this prediction. Therefore to reduce the error and be able to become a better compression needs an efficient method of prediction[23][29].

In current video encoding systems predictions are made through compensating for relative motion between two video frames[30][31]. This process is known as motion-compensated prediction and is practiced through isolating the present frame into blocks, finding the best matching block from the preceding (and/or future) frame and calculating an associated motion vector [32] [33]. This fact then enables us to predict current frames using previous frames and a set of associated motion vectors (MVs) and avoids storage and transmission of all the bits from the original video. However such strong dependency also makes partitioning and scheduling of distributed video processing more difficult [34][35].

The main challenge of video transmission is the large amount of information that would be needed to store and transmit without any coding. For example, let us consider a frame size of 176x144 pixels with three colour components quantized with 8 bits per colour. We get 24 bits per pixel and 608 k-bits per frame. Then, a one-minute video sequence recorded at 30 frames per second requires 1,094 G-bits for storage and a bandwidth of 18, 25 Mbps for transmission. Therefore, it is necessary to find a representation of the video frames that allow us to store and transmit the video sequence more efficiently[36].

Generally, there are two main types of compression systems: lossless and *lossy* compression[37].

i. Lossless compression can be compressed and restored without any loss of information a % 100 recovery. It is typically used for text or executable files, where a loss of information is a major damage. This compression may use statistical procedures to reduce data redundancies, such as Huffman-Coding (HMC) and Run Length Encoding (RLE) that are enabling high-compression ratios based on the data. On the other hand,

ii. The lossy compression doesn't need to recover an exact of the original data since it allows some acceptable degradation. This type can be used for data which is less sensitive of the losses and redundancies, such as images, video or audio.

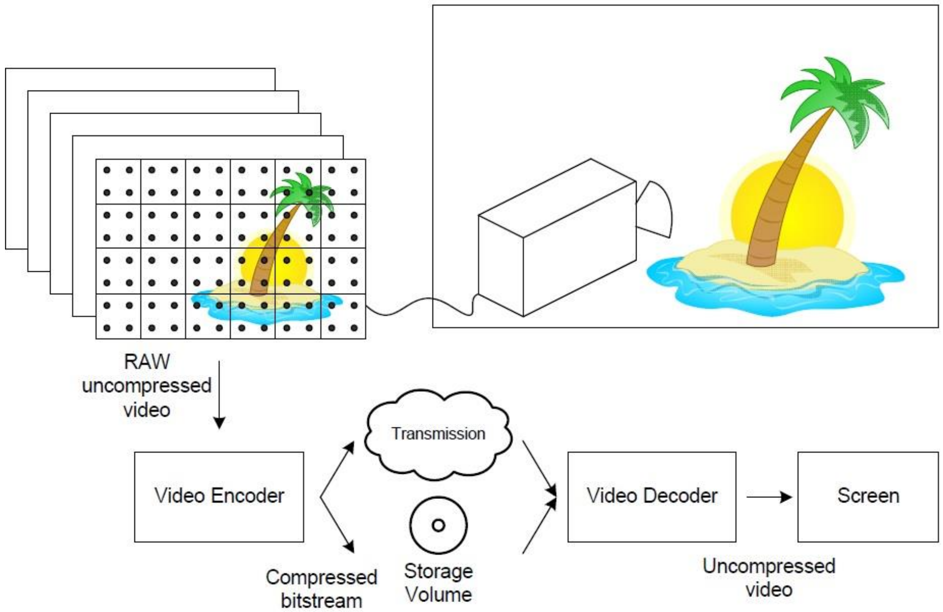


Figure 2.3: Video Compression System[10]

2.3.1 Video Frame Format

Table 2.2 presents a different video frame resolution. Digital video content exchange between various industries, applications, networks, hardware platforms needs a standard digital video formats. There are several digital video formats accepted for various services, they are typically associated to the Commander's Critical Information Requirements ($CCIR$)-601 sampling standard. The CCIR 601 was outlined for broadcast quality services. Input Source format (SIF) was defined for lower regulation storage. A lower resolution version of SIF is the quarter-SIF ($QSIF$) format [38].

The H.261 from $CCITT$, now known as the ITU-T (for Telecommunication Standardization Sector of the International Telecommunications Union), $CCITT$ defines another source format, called common intermediate format (CIF) as shown in figure 2.6. Then CIF followed by Quarter-CIF ($QCIF$). A set of popular frame resolutions is based on the CIF, in which each frame has a resolution of N x M pixels.

The resolutions of these formats are listed in Table 2.2. Other video formats consisting $CCIR$ 601, SIF, and $HDTV$ are also summarized in Table 2.2. The Common Intermediate Format (CIF) is that the basic one for general set of video frame formats. The select of frame resolution based on the application and on the available storage or transmission capability. E.g., $4CIF$ is appropriate for TV and DVD-video [23]. CIF and $QCIF$ are appropriate for video-conferencing applications. $QCIF$ and $SQCIF$ are usually used for mobile multimedia applications [38].

Table 2.2: Digital Video formats

Standards	Formats	Luminance resolution
The Commander's Critical Information Requirements ($CCIR$)-601	CCIR-601	720 x 486
	SIF	352 x 240
	QSIF	176 x 120
	Sub-QCIF	128 x 96
Telecommunication Standardization Sector of the International Telecommunications Union ITU-$CCITT$	Quarter CIF (QCIF)	176 x 144
	CIF	352 x 288
	4CIF	704 x 576
	16CIF	1408 x 1152
	HDTV	1920 x 1080

2.4 Video Codec

In this section we are going to discuss the basic building of blocks and how to exploit the temporal redundancy. The encoder-decoder process is commonly defined as $CODEC$. The major aim of a video codec is to perform highly efficient compression on video with higher-quality. These two goals are typically conflicting because a lower compressed bitrate usually produces a worse picture quality at the decoder [10].

The video encoder involves three main functional items: the temporal, spatial, and an entropy compression model, as presented in Figure 2.4.

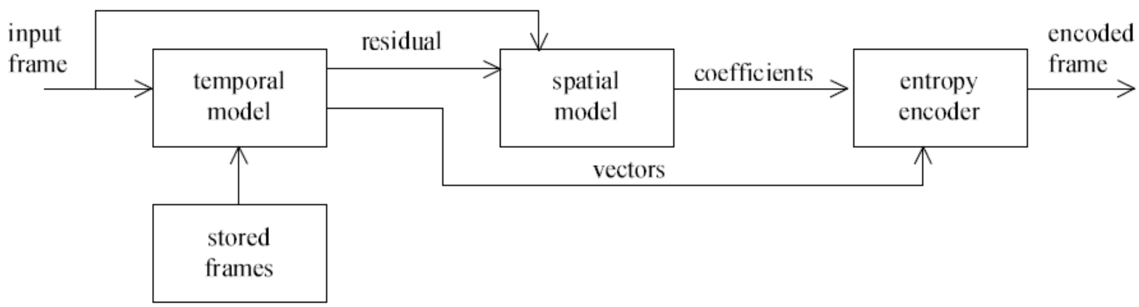


Figure 2.4: Video encoder diagram[14]**.**

Temporal compression model decreases the similarities between adjacent video frames sequence by applying the prediction process of the present frame with respect to one or more preceding/future frames: so-called a reference-frame. The output is a set of MVs, and a residual frame which is produced by subtracting the reference from the present frame [14][39].

The spatial compression model decreases spatial redundancy between adjacent samples in residual frame by performing a conversion from one to another domain (such as using a DCT one the most general transformation), and then quantizing the results to eliminate unimportant values. The result's a collection of quantized transform coefficients. These coefficients, together with the MVs, are compressed by the entropy encoder to eliminate statistical redundancy (usually occurring vectors and coefficients are denoted by short binary codes)[23]. In some codecs, the spatial compression procedure will be applied directly to the present frame while not applying the temporal compression procedure [40]. The video decoder rebuilds a video frame from the compressed bitstream. It applies the entropy decoder to coefficients and MVs, and uses these MVs together with one or more previously decoded frames, to make a prediction of the present frame (so-called Motion Compensation method). By adding the residual frame to this prediction, it gets the present frame [10].

2.4.1 Motion Estimation

The video can be understood as a sequence of frames successively transmitted and showed so as to offer a continuing of actions. This is occurred by adjusting the frequency between every two successive frames to the properties of the visual human system [41]. The aim of the temporal process is to decrease redundancy between transmitted frames, by creating a predicted frame, and subtracting this from the present frame, to find a residual frame. The predicted frame is generated from one or more preceding or future frames (reference frames). The simplest way of temporal prediction is to use the past frame as the predictor of the current frame. The more accurate is the prediction process, the less information is contained in the residual frame [10]. Figure 2.7 illustrates the dependencies generated among sequence of video frames that allows video compression. As can be noted from the figure MPEG has three types of frames namely *I, P* and *B*.

I frame, or intra-coded frames are frames which are coded individually of any other frames. These frames will be decoded by themselves while not requiring any additional information from neighboring frames [40]. They are used to give a random access (seek point) to the video stream and as a key frame from which other types of frames can start to be calculated and predicted. *P frames*, or predictively coded are frames which are coded based on previously coded frame. These means a reference to a previously coded frame is required to decode these frames *B frames*, or bidirectional predicted frames which are coded using both previous and future coded frames [10], [18], [42], [43]. In current video encoding systems predictions are made through compensating for relative motion among two video frames. This procedure is called *motion-compensated prediction* and is practiced through dividing the current frame into blocks, finding the best matching block from the preceding (and/or future) frame and calculating an associated *MV*. This fact then enables us to predict current frames using previous frames and a set of associated motion vectors and avoids the storage and transmission of all the bits from the original video [10]. However, such strong dependency also makes partitioning and scheduling of distributed video processing more difficult. The following figure 2.5 shows the prediction dependencies that exist in a sequence of video frames[23]. The coding structure can be presented using the different types of frames. We can define a loop in the coding scheme between every I-frame; a loop example could be IBPBPB, (see figure 2.8). The encoders carry out the coding algorithm of every loop in an iterative way by breaking the loop in shorter serials of frames [44]. Each serial uses the last frame of the previous one as reference, and all the frame types are represented in case that the compression mode uses all the types. An example of these serials is the frames between two I or P frames, IB or PB in this case.

Due to the different type of frames, and the different necessities owing to the references, the encoding order is not the same as the order of the frames in the sequence [18]. The order depends on the number of references; the first frame transmitted is the I-frame, I_0, and then the P-frame, P_1. The B frames are transmitted when both their references have been transmitted; B_2 and B_4. [18] the resulting transmission order is:

$$I_0\ P_1\ B_2\ P_3\ B_4\ P_5\ B_6\ I_7\ B_8\ P_9 \ldots$$

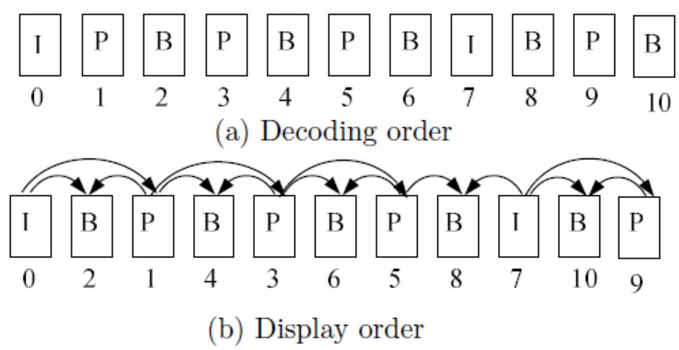


Figure 2.5: MPEG-4 video frames structure[14]**.**

2.4.2 Motion Compensation

The chosen predicted area by the motion estimation process in the reference frame is subtracted from the M x N present block to create a residual frame (luminance and chrominance) [2]. This process, achieved in the video encoder, is called motion compensation (MC). The residual frame is coded and transmitted together with the MV. The decoder employs the received MV to recreate the predicted area, decodes the residual block, and adds this one to the predicted area, rebuilding a version of the original block [10]. In addition, in the encoder, the residual frame is decoded, and added to the predicted area, to form a recreated block which is stored as a reference for further motion-compensated prediction [14]. The encoder uses the recreated block to achieve the prediction of next block so that it can have the same reference for MC that the decoder uses to recreate the original block [45].

Several coding standards, such as MPEG2, MPEG4, H.263, and H.264 (see table 2.1), the basic unit for motion compensated prediction is the macroblock, corresponding to a 16 x 16-pixel region. Smaller motion compensated block sizes (8 x 8 or 4 x 4 pixel regions) can produce better MC results, decreasing the residual data. However, such smaller block sizes increase the complexity of the MC process, in terms of search processes and bits to encode an increased number of MVs. H.264 codec adapts the MC block size due to the frame characteristics, so that large block sizes are adopted in homogeneous regions of a frame, while small block sizes are chosen around areas with large motion and great details [23]. This technique of dividing a macroblock into motion-compensated sub-blocks of varying size (two 16 x 8 partitions or two 8 x 16 partitions or four 8 x 8 partitions, each one of these 8 x 8 partitions may be split in a further four ways, or as one 8 x 8 sub-macroblock, two 8 x 4 partitions, two 4 x 8 partitions or four 4 x 4 partitions) is known as Tree Structured *MC*. The MC, as well as the ME process, can be achieved with a sub-pixel resolution, 1 by searching sub-sample interpolated positions, as well as integer-sample positions, choosing the position that provides the minimum residual information, and using the integer or sub-sample values at this position for motion compensated prediction [38] [10].

2.4.3 Quantization

Quantization is a method that leads a signal with a range of values to a quantized signal with a reduced range. The quantization-parameter (QP) has a great effect on the encoder performance, because it standardizes on how much spatial information can be saved[4]. As the increases of the QP, some of the details are combined, so that the bitrate drops with some increases in distortion and some losses of the video quality. The frame size can be reduced to eliminate the artifacts at low bitrate setting[4]. However, the reduction of the size doesn't guarantee a good-quality, as the original video frames in high-resolution, where the quality of the videos will be reduced when the bitrate is low.

This allows representing the quantized signal with fewer bits than the original one, since the range of possible values is smaller. Most video coding standards assume a scalar quantization; mentioned in [13] as the following :

$$Z_{ij} = round(\frac{Y_{ij}}{Q_{step}}) \qquad (2.1)$$

Where Y_{ij} is a coefficient, Q_{step} is a quantization step size, and Z_{ij} is a quantized coefficient. Round is the rounding process. The values of Q_{step} are indexed by a Quantization Parameter (QP) and are correlated to the position (i, j) of the coefficient in the image. In video compression codecs, the quantization operation (Q) is made in the encoder. A higher range of quantization steps makes it possible for an encoder to have a more accurate control of the trade-off between bit rate and quality [10]. If the step size is big, the range of quantized values is small, the image is really compressed, but the re-quantized values are very different from the original ones. On the contrary, if the step size is small, the re-quantized values match more closely the original ones, but the larger range of quantized values decreases the compression efficiency [10], [18], [42], [46]

2.4.4 Entropy Coding

The entropy coding transforms a set of symbols in a compressed bitstream. The "$Huffman$" variable length coding, and arithmetic coding are most common entropy encoding techniques. The first method transits input symbols to a series of code words (variable length codes or $VLCs$), thus, that frequently-occurring symbols are represented with short codes, while less common symbols are represented with long codes [14]. After that, Huffman coding assigns a variable length code to each symbol based on the probability of occurrence of different symbols. Arithmetic coding is a practical alternative to Huffman coding. It is a form of variable length entropy encoding, where the entire message is coded into a single fractional number between zero and one. An efficient arithmetic coding system is Context-based Adaptive Binary Arithmetic Coding ($CABAC$), used $in\ H.264$ codec [10]. It achieves good compression performance by choosing probability models for each syntax element from a selection of available models based on the statistics of recently-coded data symbols[10]. It uses the arithmetic coding, updating the selected context-based probability model according to the actual coded value [10] [38].

2.5 Peak Signal to Noise Ratio (PSNR)

Currently, multimedia data are everywhere; and its evaluation plays an essential role in the development and validity of image and video applications, like video compression and enhancing. There are two main types of quality assessments, subjective and objective quality assessment [47]. The subjective quality assessment is evaluated subjectively by a human viewer. However, it's time-consuming and expensive, and cannot be employed in real-time applications. To account for these drawbacks, a number of objective test methods have been developed, in which the quality rating is automatically calculated [14]. Therefore, the expert must choose the objective metric very carefully in order to get appropriate results. On the other hand, the objective one can measure between un-distortion differences and distorted image quality signal. There are several aspects which are effect on multimedia quality such as blur/sharpness, noise, color fidelity, resolution, geometric distortions, frame dropping and freezing, and so on. There are three categories of Objective quality metrics according to their input data [10]. The first representative of objective metrics is mostly used ($PSNR$) Peak Signal to Noise Ratio. This metric requires both original and distorted image in order to be computed [48]. Therefore, it is a full-reference method (FR). Other FR metrics are based on Structural Similarity or Image Evaluation based on Segmentation. Then the second category is reduced-reference (RR) metrics, it used when we have just little information about the original image. There are many examples of reduced reference metrics can be found e.g. in [49]and [50]. The final one is no-reference metrics. This it used when we do not need any information about the original image or video. Thus, they can be used in cases, where the FR and RR metrics cannot be.

One of the most popular objective tools performed to measure the video sequence quality, is a stand for Peak Signal to Noise Ratio (PSNR). A quality measurement is a very necessary demand for creators and developers of the digital video communication systems. The quality measure is a very important for compressed video, subsequently most video compression processes are lossy, hence that compression is done at the expense of video quality [10], [18], [42]. However, the advanced nature of human sighted makes hard accurately modeling the response of an individual's viewer. The most essential measures adapted to calculate the residual data are Mean Squared Error (MSE), Mean Absolute Error (MAE), and Sum of Absolute Errors (SAE), as shown in the following [51]:

$$MSE = \frac{1}{N\ x\ M} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (C_{ij} - R_{ij})^2 \qquad (2.2)$$

The most widely used measure in literature, is Peak Signal to Noise Ratio (PSNR). It is an objective measure, calculated on a logarithmic scale was defined as follows:

$$PSNR_{dB} = 10log_{10}\frac{(n^2-1)^2}{MSE} \qquad (2.3)$$

Where $(n^2 - 1)^2$ the square of the possible signal value in the frame, where n is the number of bits per image sample. The mean square error (MSE) is calculated between the original and the recreated frame. In this study, PSNR used to evaluate the quality of the proposed technique[49].

2.6 Video Bitstream Structure

Several standards are set to specify the bitstream structure of compressed video content so as to create interoperability of video stream among various devices. This bitstream is what is transmitted and stored among different devices [7]. The specific structure of this bitstream is different for different compression techniques but has a general structure that follows the fundamental concepts that make compression possible. The following figure shows the bitstream structure in an abstract way (see figure 2.7).

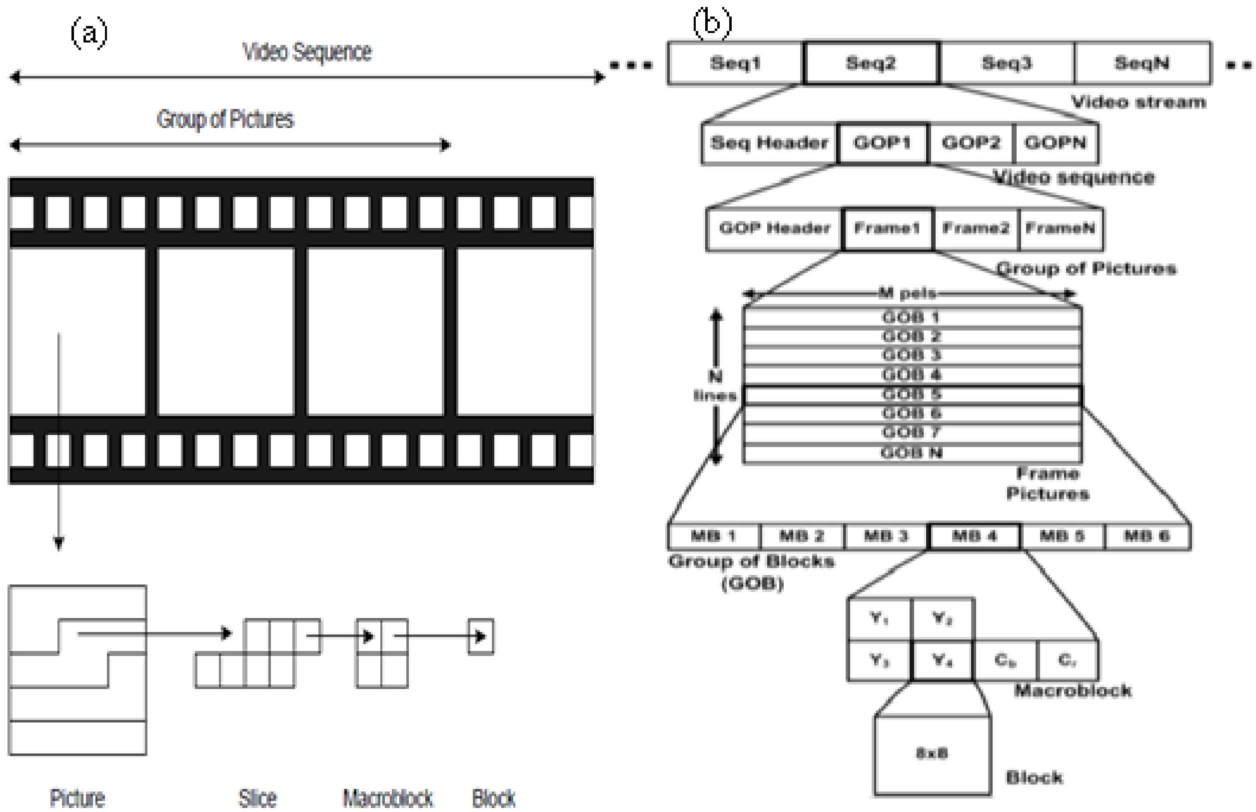


Figure 2.7: Video codecs in a hierarchy of layers for two standards bodies: (a) ITU-I and (b) ISO[45]**.**

Figure 2.7 illustrate the bitstream structure that result from video compression algorithms. From the perspective of distributed transcoding we are more interested in this structure for the main reason that we require a suitable way to split the bitstream among different processing parts. Selecting the right layer for this propose is necessary as different granularity results in different performance in terms of system latency, total transcoding time and implementation difficulty[17].

Employment of layers with smaller granularity such as macro-block results in higher communication and synchronization costs as well as higher implementation complexity. This is due to tighter data dependency down the hierarchy as seen in figure 2.7. Granularity such as video sequence (see figure 2.8) will result in a higher latency of the system affecting the streaming behavior of the video being processed. This is because it takes more time to process a video sequence than a GOP or a frame [45].

The distributed transcoder implementation in this work utilizes the group of pictures as atomic units of distribution. However, the main reasons for this choice are mainly implementation complexity and the side effect of fine granularity in terms of communication over heads [10].

2.7 Video Transmission

Video transmission over wireless-networks use two modes, namely downloading and streaming. In download mode, an entire video file will be downloaded from the video server and then the video file will be played[52][53]. In streaming mode, all video content does not need to be downloaded before viewing. Instead the video is played when a sufficient number of video frames have been received by the user device [40].

Video transmission requires a steady flow of information and delivery of packets by a deadline. However, wireless radio networks have difficulties to provide a reliable service. Video transmission over a dynamic channel, like mobile wireless-networks, is more difficult than over a static channel, since the bandwidth, delay, and packets loss are not known in advance[53]. Therefore, wireless-networks need some effort in order to use channel resources efficiently such as bitrate control [54].

Rate Control: If the control-parameters (motion-estimation search area and quantization-step size) of a video encoder are kept constant, the number of bits created for each frame will change based on the content of the frame (more bits when there is high-motion, and more details, fewer bits when there is low-motion and not much details)[5]. This causes a variable bitrate (measured in bits per second) of the encoder output. This variable bitrate cannot be supported, for example, by a constant bitrate channel [10].

Moreover, it is needed to adapt the bitrate generated by a video encoder to match the available bitrate of the transmission mechanism. A typical technique used by the encoder to smooth variable bitrates is buffering the encoded data before to transmission. High bitrate variations can determine over-or-under flowing of the buffer, and great decoding delay. Rate control is a mechanism able to control the encoder output bitrate, preventing buffer overflow and underflow. Rate control modifies the quantization parameters (QP), since increasing QP reduces bitrate with a lower-quality of the decoded frames [2]. The motivation of rate control is to optimize the tradeoff between the encoder output bitrate and the quality of the decoded video sequence [17]. Many rate control algorithms for video coding have been proposed, according to different video applications and coding standards (see section 2.1)[10] [2].

2.8 Quality of Service

Quality of Service (QoS) refers to the ability to provide a satisfactory service during a communication session[47][55]. Consistently, anticipating and meeting users' QoS needs are what distinguishes successful communication services and product providers from their competition[56]. Several parameters could affect the quality of video [4] transmission over wireless networks, e. g., Compression parameters. The main issue that makes video streaming difficult is the large number of video frames that are transmitted over wireless networks. However, video streaming is compressed in a lossy manner by H.264 codec, leading to smaller representations of video data than those that are available with lossless data compression [4]. Compression plays an important role in video streaming. The nature of the video scene, like the amount of motion, colour, contrast, frame size, and the number of frames that are transmitted per seconds, can also have an impact on the human perception of the video quality.

[4] The main issue with real-time video streaming is that it is difficult to guarantee an end-to-end QoS during the entire streaming process [4].

The current best-effort networks do not offer any QoS guarantees for video transmission over wireless networks [4]. Wireless network performance is defined as the requirements that must be guaranteed, such as bandwidth, end-to-end delay, and jitter. The network services depend on the traffic behavior and perform due to the traffic parameters such as the peak data rate. Further, frame losses during the transmission could have a negative effect on the quality of the video[4].

2.9 Video Transcoding

Rapid development in information and communications technology lead to the emergence of many versions of interactive services, which are used daily, for business or private purposes[57]. The most challenge is to deliver these services to multiple users, over different networks. For that, multimedia information needs to be adapted dynamically due to the user situation. Hence, the transcoding technology is required to meet these tasks [58].

Video transcoding is a process that converts of a compressed video stream into different one with various features, without performing the full decoding and re-encoding methods [42], [59]. To allow different devices with various capabilities (processing, memories, decoding, etc.) interconnect with each other over heterogeneous networks with different characteristics, different kinds of transcoding are needed.

As illustrated in Figure 2(8 and 9), Video transcoding can play several roles (or tasks) and functions, such as format conversion, bitrate conversion (quality transcoding), resolution scaling (spatial transcoding), bitrate conversion (quality transcoding), and frame rate conversion (temporal transcoding)[60], [61][10].

i. Format conversion: can change a syntax form of video coding standard from one to another.

ii. Spatial transcoding: decreases the spatial resolution of the compressed video, to faces the problem of limited size in many access terminals.

iii. Quality transcoding: works on the bit allocation for each frame, by enhancing the quantization parameters, according to the target bitrate.

iv. Temporal transcoding: is a process, which skips some frames in order to adjustment the frame rate of the video sequence, without reducing the video quality of not skipped frames.

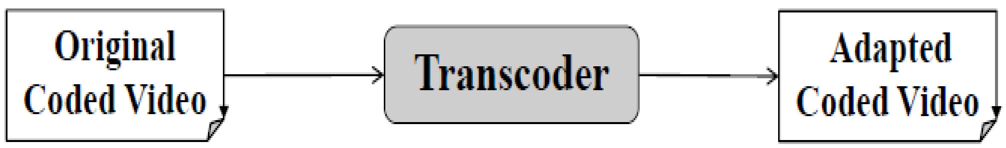


Figure 2.8: Video Transcoding scheme.

Previously, transcoding technique is used to adjust bitrate of the compressed video stream to adapt with the channel bandwidth [6]. For that, source data is compressed at high bitrate but when transferred over a channel, it needs to be at a lower bitrate.

Generally, transcoding is divided into main two categories and additional functions: *homogeneous* and *heterogeneous* transcoding[62]. The homogeneous transcoding converts bitstreams within the same standard with frame-rate adjustments, an adaptive quantization and a resolution conversion. The heterogeneous transcoding perfors conversion between various video coding standards [6][62].

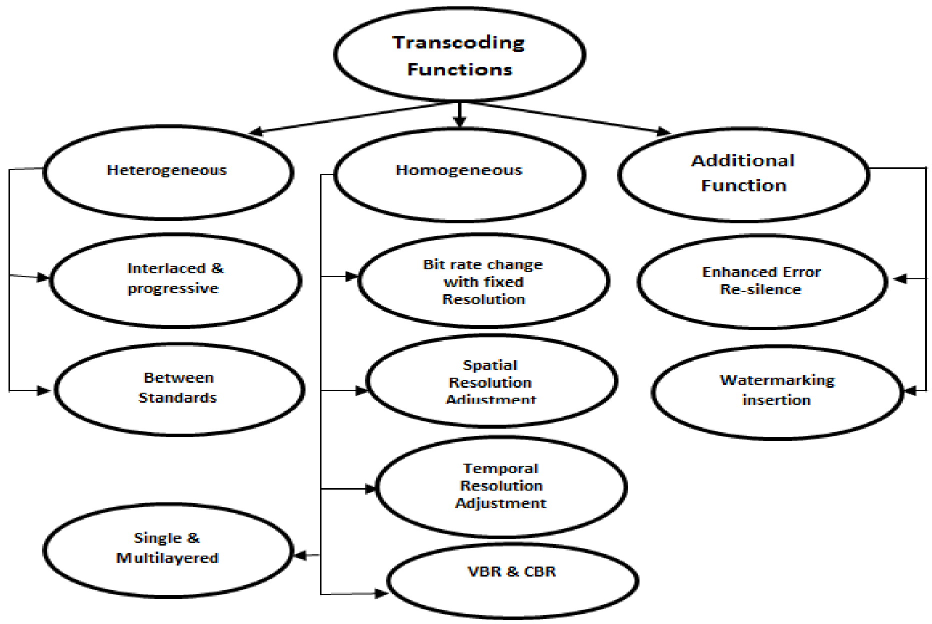


Figure 2.9: Classification of Transcoding functions.

2.10 Related Works

Exchange digital video between various industries, applications, networks, hardware platforms requires a standard digital video codec[63]. The coded video sequence should be transformed into particular bitrates for every outgoing channel. When the bandwidth in a network channel is very limited, the video quality degradation will be high, if the frame rate is constant. So they need a good special technique to adjust frames rate of video sequence without reducing the quality of remaining video frames.

A literature review was carried out in order to find out a way that to face the weakness points of streaming video over unreliable network. The literature study had been concentrated to find out the related works that reduced frames of video sequence by using skipping techniques on the transmitter side, and the interpolation scheme in order to reconstruct the video frames back in the receiver side. The study searched in digital libraries, focusing a lot of in IEEE and ACM for each research relating to topics of dropping videos frame counts, digital video adaptation, and video frame interpolation. This study was used to know the most video streaming problems and act as a basis upon to constructed our proposed technique.

2.10.1 The Frame Skipping Techniques

The selection of frames to be dropped is a vital and really very sensitive issue, because it greatly affects the volume (or quality) of video sequences. The processes the on video frame rate, the research relied on video transcoding. Temporal transcoding is a method employed to drops some frames in order to adjustment frame rate of a video sequence, while not reducing the quality of the remaining video frames. Also, video transcoding using different encoder parameters to achieved different aims like incrementing quality of the video or decreasing bandwidth consuming [49].

Furthermore, there are another research works on the subject of video frame dropping which are not classified under video transcoding. The more studies which are related to this research are offered in the following.

Most video frame skipping techniques are relying on motion vectors *MVs* to skip unnecessary frames. In [64][65] proposed a new variable skipping-frame approach. Their work is based on motion detection to perform skipping-frame on the real-time encoding. Also, selecting frames to be encoded depends on the associations of the consecutive frames in an adaptive length sliding window; the length is depending on the target bitrate in order to choose the number of dropping frames. And, to assess the amount of the motion, temporal differences between corresponding components of two consecutive video frames are employed for motion-detection. However, the results of this work provide a very low computational complexity, and are appropriate for real-time services.

Ref [66] proposed two kinds of frame dropping algorithms to sustain the energy efficiency of Dynamic Voltage Scaling (DVS) in MPEG decoding. When MPEG is a decoding frame, the processor maximizes frequency to speed works in order to avoid violation of the deadline. Once the decoder finished the processor waiting until the deadline of the frame must be played out. This work is trying to exploit this period of processor idle to avoid wasting energy and maximize the throughput. Moreover, this work offers better methods to avoid consecutive frame skipping to preserve the quality of video playing. The Late-skipping algorithm choices a frame to skip when the delayed decoding time (EDT) of a frame is larger than one EDT. And the Early-skipping algorithm choices a frame to skip when the delayed decoding time of a frame is larger than one-half of an EDT. The algorithm is focused on saving energy while the video quality is secondary importance.

Authors [67] proposed a method for adaptive network bandwidth by selective frame skipping. This method depends on a (GoP) coding structure. The method provides a dynamic frame–skipping mechanism design for each(GoP) built on transmission results of proceeding GoP. From the experiment outcomes it displays that the method provides better real-time performance and less computational complexity. But, the outcomes present that, the technique is concentrated on decreasing the congestion of network bandwidth, when the quality of the video is a lower significance. Finally, the frame rebuilding is not taken into account as a method to enhanced or improved the of transmitted videos quality.

The [52] proposed a new bitrate transcoding method, which is appropriately modified for the H.264 encoded orders. This algorithm employs directly in the compressed field. Besides that, this work is based on the macroblok(MB) types, to predict the next frame reference. Depends on that each frame can be dropped does not reference to another frame to avoid the problem of drift error (frames mismatch). This method is a very efficient and provides the best real-time performance, and lesser computational complexity. From an experimental, the algorithm concentrated on avoiding the problem of drift error, while the video quality is a secondary issue.

Authors [68] proposed a new method to select frames to drop from a video sequence. The method used both the perceived audio and video play out quality in consideration of the selection technique. This work uses three types of polices to implement skipping frames, RND – used to dropping frames randomly, D (2), and D (5) - both do not dropping a consecutive frame. Furthermore, from their experimental outcomes, it showed that the method offered a better performance of perceiving audio-video play out quality is very similar if audio information is used in selecting video frames to drop.

The authors [69] proposed a new frame-skipping approach to deal with the restrictions of wireless networks channel bandwidth. The study is also depends on a group of picture coding structure(GoP). The algorithm is segmenting the videos into various GoP and each frame assign a priority. The method is performed when the network is congested then frames are skipped due to priority, otherwise the entire GoP is transferred. Different frames are set different priorities for video content so as not to affect the decision-making process, if any possibility to make necessary frame as a less important frame leads to a problem. Finally, this algorithm doesn't perform on the transmitter side unless the network channel congested.

The authors [34] proposed a method for video frame-dropping depends on the size of changes between successive frames of the videos. This method presents the best video stream structure than the one offered by the video CODEC(encoders/ decoders). But, the method doesn't introduce nothing to enhance the video frame reconstruction. In addition, the method tracks the frames of video based on the statistically differences between frames, and don't considerable the contents of the video.

Ref. [70] proposed a method for video frame-dropping in a mobile network system. The approach is also depends on GoP, by separating the video into various $GoP,$ and set priority to each GoP. This makes easer specify frame with lesser importance to skip. The proposed algorithm doesn't improve the solution of the wireless network channel problem, but adding another burden by distributing the missing frames only on the parts of the image sets with lower priority.

This study concludes with a summary of the most pronounced in the area of video frames dropping, and transcoding video. Our research goal is to propose a technique to manage and structure the video contents into various parts, with considerable movements (motions), while dropping the unnecessarily frames due to similarity of the video frames. This to decrease the frames number that requires to be sent over the network, while after receiving the video, the client re-construct the skipped frames that will displayed on the screen of the receiver-end.

2.10.2 The Reconstruction Video frames

Video Frame Interpolation (VFI) is defined as a process which using known data (frame) values to estimate unknown data (frame) values[71]. The video frame interpolation (VFI) is an essential and very wide area in digital video contents process. The VFI has a lot of methods and functions that are used to re-constructing the lost frames based on neighboring (or reference) frames. In this area, many studies have been carried out, and some of them are presented below.

2.10.2.1 Classical Interpolation Techniques

Earlier years some very simple Classic Interpolation Techniques (CIT) have been employed in order to increase the video frame-rate. These methods are frame repetition (FR) and frame averaging(FA). The success is not due to the quality obtained, but because of its simplicity [44]. The CIT do not achieve a high success in video sequences with high motion. However, regardless of not very good results, they are operated in video with high frame rates, (e.g. 25 or 30 frames per second). So, these rates have very low motion between successive frames due to the large number of frames per second. In high motion environments these techniques present annoying jerky or blurry artifacts very sensitive to the human eye.

The frame Repetition contains on repeating the previous frame as an interpolated frame, so its implementation is really simple computationally. In spite of it is employed very often to refresh the screen, the results obtained applying this method are not good due to the motion jerkiness is not reduced[44].
One difference of the typical frame repetition employed in the television broadcasting contains in interpolating a frame repeating both the previous and the following frames. It is operated interlacing the even and the odd lines of the frame.

On the other hand, the frame averaging is depends on estimating the lost frames by computing the mean value of the previous and the following frames pixel by pixel. If the motion of the image is slow, the changes between decoded frames are easier. However, if the motion is high it produces trace effects[72].

2.10.2.2 Motion Compensated Frame Interpolation

The Motion Compensation Frame Interpolation ($MCFI$) was established to improve the quality of the interpolation approaches and offer better perception quality of the video at the receiving end [44]. These methods are depending on the motion vectors between two successive transmitted frames of a video. Then they are exploiting the correlation of the temporal and spatial of the sequence of video. There are two main motion compensated interpolation methods: 1) the pixel based-$MCFI$, and 2) the block based-$MCFI$. The pixel-based did the analysis on a pixel by pixel basis. At the beginning it categorizes each pixel in the frame into one of four classes: moving object (MO), static background (SB), covered background (CB), and uncovered background (UB) [44].

The pixel-based MCFI uses various interpolation techniques to each one of the above groups. So, it is difficult to apply this technique to real time services. Because the pixel-based MCFI is that the computational cost of motion estimation-pixel by pixel is very high[73].

Hence, the computational problem of the pixel-based MCFI is solved by the block-based interpolation techniques[74]. The video frames are divided into squares, typically with a size of 8x8 pixels or 16x16-pixels. The interpolation is done from the motion vectors transmitted of the video codecs.

The main two problems of the interpolation methods are to get accurate motion vectors that describe the motion of the elements in the image, and how to use them to get the interpolated frame. The major constraint of these processes for real-time services is the computational *cost*. In real-time services it has to catch a lot of block motion estimations in a short period of time to interpolate the dropping frames. When these techniques are implemented, it is necessary to avoid the blocking artifacts generated from the block partitioning. As the human eye is more sensitive to the luminance changes of the images than to the real-intensity, these blocking artifacts are annoying. Therefore one of the most important goals is to avoid the blocking effect[71][44].

Motion compensation frame interpolation is a widely studied method, and there are a lot of implementations that seek to improve the output quality. Therefore, it is important to have an overview of recent algorithms implementations. At this point some of them will be briefly introduced[44].

2.10.2.3 Non-Linear Motion- Compensated Interpolation

This method uses a new form of motion compensation; the main difference is that it doesn't suppose a linear interpolation. Until now, all the methods work with the motion vectors embedded in the bit-stream or calculated in the decoder, and then they apply linear estimation between the previous and the current frame, but now we are changing this lineal model to the 6-parameter affine motion model 2, which admits rotation and translation[71]. This new model has better results than the lineal model, especially when the number of skipped frames is higher. However, the computational cost is very high.

The authors [75] proposed a novel method for frame-interpolation using a phase-based technique. This method is based on optical-flow motion estimation scheme to finding pixel correspondences across images are depending on optical flow. The method estimates and adjusts the phase shift information using a coarse-to-fine approach, if that a high frequency content moves in the same way to lower-frequency content. The algorithm also offered lower-computational complexity, and provided a better solution for the most intuitive parameters that mostly is fixed, as well as a graceful degradation one. Thus, this research also offerings different functions that can be used in video frame interpolation topic[65].

The [76] presented a new model for video frame interpolation. The model is providing a video frame-interpolation method of locally adaptive robust principal component analysis (RPCA) used weight priors. Besides that, the method performs a high-resolution reference image and resulting preserves much of a high-resolution detail in the sequence. The interpolated frame is initialized by a simple (Motion-Compensation-Interpolation) MCI-based method which calculates the intensity value at every pixel together with a weight of how confident that pixel has plausibly resulted from true motion estimation[77]. Finally, from experimental results showed that the algorithm provides better performance if it compared with the conventional algorithms.

The authors [78] presented a novel multi-level frame interpolation technique by using the interactions among different stages. The model has employed three major stages that work at a block level, pixel level, and sequence level. The block-level motion estimation with dropping unreliable motion vectors (MV), pixel-level motion vector (PMV) depending on feature transform flow matching, and sequence-level total variation standardized completion[79][25]. Compared to old methods that attention at only one single level, this method offers a management and interactivity among three levels based on their distinct characteristics and intertwined relationships.

The author [80] introduced a concatenated model consisting of frame interpolation, warping and post-processing stages. The model is also integrated several spatial interpolation methods into warping algorithm as well as one spatial domain great resolution method. The experimental results show that the warping algorithm improves the objective and subjective quality significantly on sequences with low motion, regardless of which spatial interpolation methods are used.

Table 2.3: Related works strength and limitations

Author	Approach	Strength	Limitation
Huo et al. [48]	Method for selective frame dropping to adapting the network bandwidth. The method itself offers dynamic frame – dropping policy creation for each GoP based on transfer results of previous GoP.	From their experiment results shows that, the method provides a better real-time performance and less- computational complexity.	However, the results present that, the method is concentrated on decreasing the bandwidth network, while the quality of video is less significance.
Zheng et al. [49]	Technique for frame dropping to deal with video streaming limitations over wireless networks.	The method is performed when the network is congested then frames are skipped due to priority, otherwise the entire GoP is transferred.	Different frames are set different priorities for video content which not to affect the decision-making process, if any possibility to make necessary frame as a less important frame leads to a problem.
Zhu et al. [50]	A method for dropping video	This algorithm presents a good video	However, it doesn't introduce

Author	Approach	Strength	Limitation
	frame depends on the size of changes between successive frames of the videos.	stream framework than the one offered by the video CODEC.	nothing to enhance the video frame reconstruction.
Liu et al. [51]	Proposed frame dropping approach in a mobile device wireless network system.	The approach is also depends on GoP, by separating the video into various GoP, and set priority to each GoP.	The approach doesn't improve the solution of the wireless network channel problem, but adding another burden by distributing the missing frames only on the parts of the image groups with lower priority.
Tien-ying et al., [52]	These techniques are used in a video sequences with low motions.	Simple implementation techniques.	These techniques are presented jerkiness with high motion video sequences. And, produced high computational cost.

Author	Approach	Strength	Limitation
Motion compensation frame interpolation	This method is pased on the motion vectors between two successive transmitted frames of a video.	This techniques were established to improve the quality of the interpolation methods and offer better perception quality of the video at the receiving end.	The main two problems of the interpolation methods are to get accurate motion vectors that describe the motion of the elements in the image, and how to use them to get the interpolated frame.
Simone et al., [53]	A novel method for frame-interpolation using a phase-based technique. This method is based on optical-flow motion estimation scheme to finding pixel correspondences across images.	The algorithm also offered lower-computational complexity, and provided a better solution for the most intuitive parameters that mostly is fixed, as well as a graceful degradation one.	However method doesn't make any improvements to the video-frame reconstruction or video quality.
Dao et al., [54]	Providing a video frame interpolation method by locally-adaptive robust principal component	The algorithm performs a high-resolution reference image and resulting preserves much of a	The method is concentrated on preserving the frames priors, while the video

Author	Approach	Strength	Limitation
	analysis (RPCA) with weight priors.	high-resolution detail in the sequence.	quality is less significance

2.11 Summary

This chapter discussed the basic concepts of the video coding, and then presented the video compression techniques; explore the motion estimation and frame types (I-frames, P-frames, and B-frames). Also, the DCT transformation when it used and how is described. The quantization is processed that maps a signal with range value to a signal with reduced range. Then, we present Peak Signal to Noise Ratio PSNR, one of the most popular tools that used to evaluate the video quality. Video transcoding is a technology for providing multimedia access by the internet users with different access links and devices. This section studied several existing video transcoding systems. The cascade transcoding is simple, flexible and this technique can be used as a transcoder benchmark for comparison of the performance of other architectures.

Finally, the chapter conferred an outline of a related work to discover a way to face the weakness points of streaming video. This chapter was also focused to find the related works that decreased video frames sequence by using dropping techniques on the transmitter-side, and then rebuilt the dropped frames on the receiver-end by performing interpolation technique. The study searched in digital libraries, focusing more on IEEE and ACM to any research relating to the topics of dropping frame of video sequence, video-adaptation, and video frame-interpolation.

Chapter III: The Proposed Video Streaming techniques

3.1 Introduction

Currently, mobile-networks are able to support different type of services, such as video streaming that creates a great demand on wireless networks-bandwidth. The Bandwidth is the most important resource in mobile-networks; therefore, it is important to service adaption mechanism for efficient use of the available bandwidth. Network adaptation means how many network-resources should be utilized for video content, resulting in developing an adaptive streaming scheme for video transmission[30]. The main aim of H.264/SVC is to offer bandwidth-optimized transmission for real-time video streaming by observing the current network conditions[81][82]. The H.264 covers a rate-control method that is automatically adjusts the encoder-parameters to accomplish a target bitrate by assigns a budget of bits to the video frames sequence. The main aim of H.264/SVC [83]is to provide bandwidth-optimized transmission for video streaming by observing current network conditions. The H.264/SVC provides three kinds of improvements for optimized the bandwidth-transmission: 1) it can assist spatial improvements of quality via a signal-noise-ratio. 2) it can assist temporal improvements by adjusting the frame rate, and 3) it can support spatial improvements via resolution[84].

The user attention is ability to determine the interested regions for a given scene, which is call region-Of-Interest. divide the video frames into area of interest, and non-area of interest; therefore to adjusting the resolution of the video frames. The areas of interest have more details and it has the most important motion in the video frames. The area of interest will achieve coding on the original- resolution, whereas the non-area of interest will employ coding on low-resolution. Mavlankar et al. [65] and [85]study how to determine the slice-size for streaming the areas of interest. The server side will adapt the streaming frames due to the areas size of the video data that are preferred by the client[79]. The optimal slice-size performs the best trade off to minimize the predictable number of bits that transmitted to the client per frame, and it will be based on the user screens resolution. [82]adaptive the video frames by determining the areas of interest and skipping the background (non-area of interest) for (SVC) scalable video coding[86][87]. The adaptive dropping decision is based on the motion levels of the non-area of interest, whereas the kept bits by dropping non-area of interest are used to improve the area of interest quality.

This study proposed a technique to reduce the amount video data. The size of the video contents can be reduced by determining and transmitting the important frames, which is considered as the Frame-Of-Interest (FOI), and drop the less important frames, which is considered as the non-Frame-Of-Interest (non-FOI). The proposed-technique is designed in a way that can improve the end-user's perceived quality.

3.2 The Proposed Technique

Real-time video streaming over wireless- network sometime suffers from bandwidth channel limitations which are unable to handle the high-amount of video data. To guarantee a good user-experience to the videos that are transferred to the client side, it is important to adapt the streaming rates of videos over limited channel-bandwidth. Therefore, this work proposes a technique to minimize the video data by applying Frame-Skipping-Mechanism (FSM) in server side to identify the (lees/ high) motion similarity between adjacent video frames. The technique that used to defining the motion estimation is (SAD) Sum of Absolute-difference. The SAD is computed to the consecutive video frames to identify the motion similarity. The frames of less-motion similarity are considered as Frame-of-Interest (FOI), and drop frames that are of the higher-motion similarity, which is considered as non-frames of-interest (non-FOI). The FOIs and the reference frames (RF) are streamed to the client. The client will use a re-construction mechanism to re-construct the skipped frames (non-FOIs) that are skipped on the server side, which is called this Re-construction-Frame-Mechanism (RFM), as shown in figure 3.1:

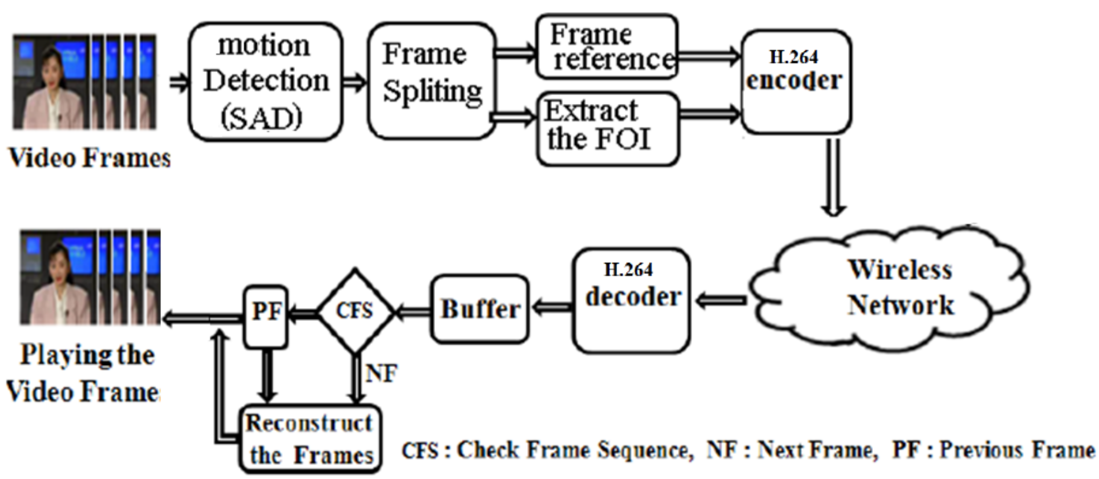


Figure 3.1: Block Diagram of Video Streaming Techniques.

The proposed technique consist of two main parts, as illustrated in figure 3.1 above, one for reducing the video frames count, so-called streaming server, and the other receiving and re-constructing frames of video back, called client. The technique operates in the both parts as follows in figure 3.2.

In the streaming server:

1. The Skipping-Frame-Mechanism(FSM) is applied to the incoming video stream (vs).
2. Based on estimated motions between frames FSM determines if there are frames to skip and marks them for skipping.
3. The video stream is adapted by removing frames marked for skipping.
4. The adapted video stream is streamed to the network

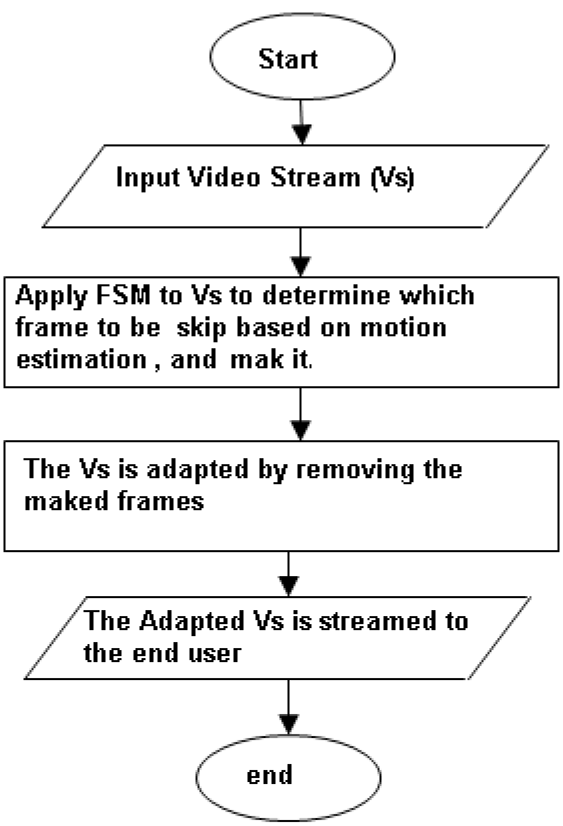


Figure 3.2: Video Frame Skipping in Server side.

In the Receiver side:

1. Receiver receives the adapted video stream Vs'.
2. FRM (Frame Reconstruction Method) is applied to the received video frames to reconstruct the frames that were dropped in transmitter side.
3. Then, Re-constructed video frames are played according to their sequenceposition number.

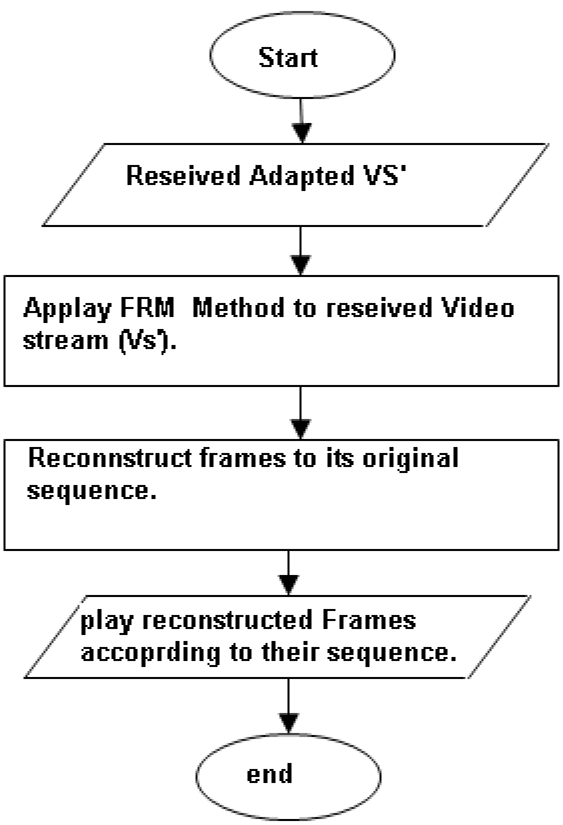


Figure 3.3: Reconstruction Frames in client side.

3.3 The Frames Skipping Method(FSM):

The main objective of the proposed technique is to provide smooth video playout in the client side. This technique will divide the video stream into two different segments with considerable movement in them. The first part is considered as the reference frame (RF), and the second part is the Frame of Interest (FOI), that is extracted from the frames that are between RF, and drop the non-Frame of Interest (non-FOI). The transmitter part applies the Frame Skipping Method (FSM), which is designed to determine frames to-be-skipped (non-FOI) based on the changes in the motion estimations and giving possibility to re-construct the dropped frames on the user side. After that, the FOI and reference frames are streamed to client side. By the other words, the FSM procedures a video stream Vs as input and return it with less or equal to the original video stream as the following:

$$Fn(V_S) \geq Fn\big(FSM(V_S)\big) \qquad (3.1)$$

Where Fn is video frame numbers, FSM is Frame Skipping Method, and Vs is the original video frames sequence, $Fn\ (FSM(Vs))$ is the remaining video sequence (only the key-frames), in the other words, the adaptive video in Vs.

3.3.1 Detecting the non-FOI

The streaming server will establish the connection according to the client request. The server will apply the FSM to identify the less motion frames similarity (FOI) between the consecutive video frames and drop frames of the high-motion similarity (non-FOI). The technique is used to calculate a motion value is Sum-of Absolute-difference (SAD). The SAD is a commonly used technique for motion estimation in various video encoding standards like H.264 [44]. The idea is to take the absolute differences between consecutive video frames. The SAD value will be zero-except for the changes induced by the objects moving between the video frames. If there is a lot of motion in one-region of the frames, the SAD value in this frame will be relatively high, and if there is no motion then the SAD value in this frame will be zero. The SAD is calculated to determine the position of the motion in video frames. The idea is to keep the frames of less motion similarity in the video sequence and drop the frames of higher-motion similarity (non-FOI). The test videos are used in this work were the samples of video sequences, with a resolution of 176 x 144 pixels. The selected videos are well known as professional test videos that have different characteristics and different motion-levels, as shown in Table 4.1.

The adjacent video frames are scanned from top-left to bottom-right corner by calculating the SAD to determine the motion differences between the video frames:

The SAD(i, j) is calculated to the consecutive video frames and from top-left to bottom-right corner to skip frames according to the change in the estimated motion giving the possibility to re-construct the skipped frames on the client side using video interpolation, as shown in Figure 3.4 and according to (3.2).

$$SAD(i,j) = \sum_{z=1}^{L-1}\sum_{i=0}^{X-1}\sum_{j=0}^{Y-1}|F_z(i,j) - F_{z-1}(i,j)| \quad (3.2)$$

Where L is the length of the frame sequences, X is the width, Y is the height.

Figure 3.4: Scanning **the consecutive video** frames based on SAD(**i**, **j**).

The SAD is computed to the consecutive video frames to find the differences values to the test videos. The differences are used to identify the most differences between the video frames.

FSM is operated to drops frames of incoming video as follows:

- Motion estimation technique is used to identify motion between video frames.
- Skipping video frames is based on the change in motion between consecutive video frames.

Motion estimation (ME) is technique performed to identify the similarity between video frames that need to skip on the transmitter side. In the coming sections, we will explain in details the frame skipping.

Motion Estimation

The main objective of the block-matching (BM), motion estimation (ME) is a process operated to compare pictures taken two different time frames, and estimate the direction of motion place between the two frames (i.e., it called motion vectors (MV)), as seen in figure 3.5. Hence, the challenge is to obtain best motion vectors (MVs) by using a pixel-domain or block-domain search method, and appropriate process of BM parameters.

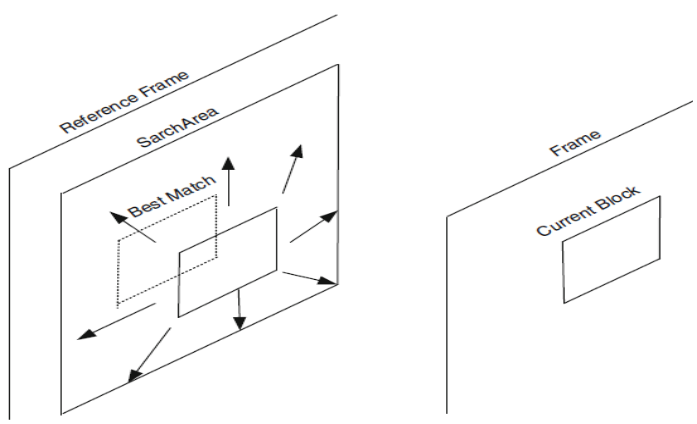


Figure 3.5: Motion Estimation diagram.

The concept of the FSM is depends on the understanding that it could estimate the motion between video frames. And also, the motion estimation(ME) tasks are usually based on quality of matching and the execution speed. Therefore, the study here is select execution speed instead of quality-of-matching, because this choice is made due to the temporal nature of video streaming. The authors [88], [89] are proposed the faster block matching technique is called $Three\ Step\ Search\ (TSS)$.

The two types are presented TSS-TYPE-1 by [90] and TSS-TYPE-2 by [89] respectively. The Both techniques are used Sum of Absolute Differences (SAD) as a difference metric to match the blocks in frames of the videos, as the following:

The operation of TSS process will be seen in Figure 3.6 as follows:

Step one:

1 Calculate SAD of initial block, and stores it as a reference value.
2 Calculate SAD for the surrounding blocks.
3 If any value of the surrounding blocks has a lowest SAD value, choose the block of the lowest value for the next iteration.
4 If all values are higher than the SAD value then stops the algorithm, assuming that better block matching is found.

Step two:

1 Stores the lowest SAD value of last iteration as a reference value.
2 Sets the block with the lowest SAD value as a center for next search.
3 Calculate SAD for the surrounding blocks.
4 If any value of the surrounding blocks has a lowest SAD value, choose the block of the lowest value for the next iteration.
5 If all values are higher than the SAD value then stops the algorithm, assuming that better block matching is found.

Step three

- Is identical to the second step.

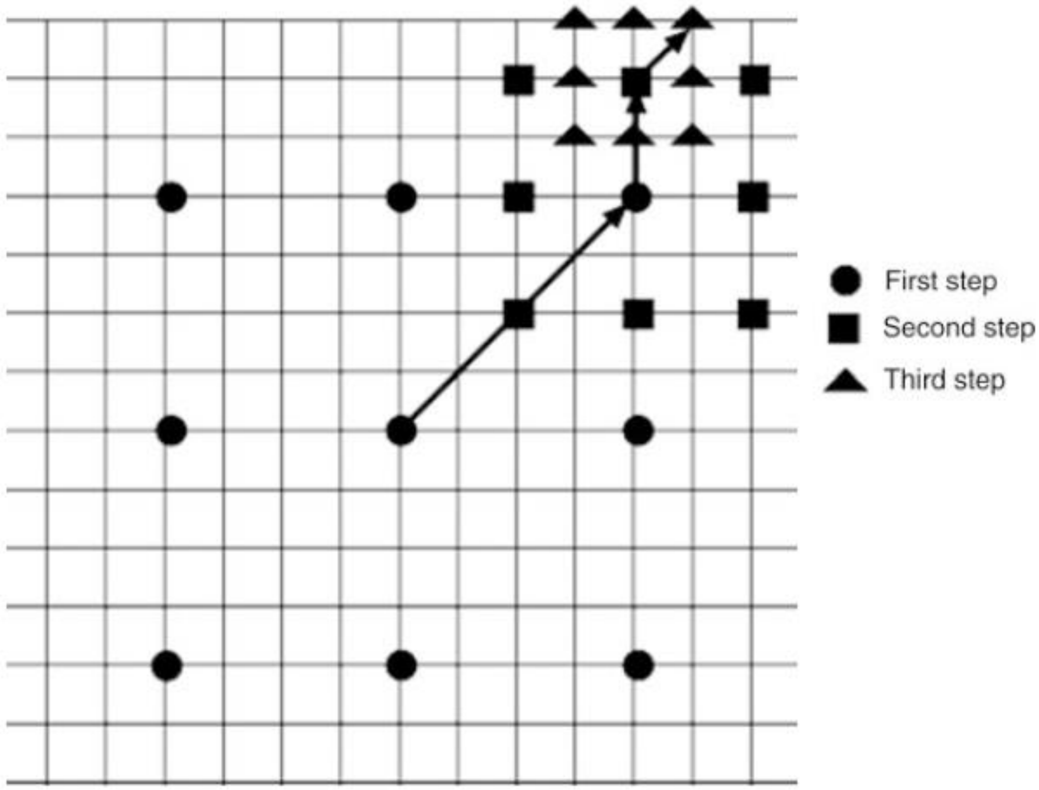


Figure 3.6: Search Patten of (SST)[14]**.**

3.3.3 Frame Skipping

The following table 3.1 shows the data structures that are used by FSM to skip frames based on the changes in motion between video frames sequence:

Table 3.1: The structures of skipping video frames

Structure	Attributes
Vector	a structure that holds vectors beginning and end coordinates
Array of vectors attributes	Array of structures that hold vector speed, direction and end coordinates.
Vector field (VF)	Structure that holds frame dimensions, block dimensions, and an array of vector structures.

The following steps are operated by FSM to identifying the changes in the motion between video frames:

i. Using TSS method, a VF is generated for a reference frame (F_n) and the following next frame (F_{n+1}). So, array of vector attributes is created.

ii. Using TSS method, a VF is generated for frame (F_{n+1}) and the following close frame (F_{n+2}). A recently generated vector field is compared to an array of vector attributes, for every set where the difference in vector speed or direction exceeds the set limits; the changed vector-counter is increased.

iii. Therefore, if vector counter (VC) doesn't go above the limit established by the proposed parameters, mark this frame for dropping, and then the FSM goes to *ii*. Otherwise, if VC go above the limit, the frames that where compared last are marked for keeping, the FSM goes to the step *i*.

3.4 Frame Reconstruction Method (FRM)

The client will receive and decode the adapted stream, and are resident in the buffers waiting to be displayed. The chick frame sequence will be applying to incoming bitstream to re-construct the skipped frames. Then, frame interpolation method uses the output of the video decoder, to re-construct the dropped frames. The resulting video sequence can be represented in the following equation:

$$Fn(V_S{}') = Fn(RFM(V_S{}'')) \qquad (3.3)$$

Where Fn is a video frame number, $V_S{}'$ is original video stream; $V_S{}''$ *is* an adapted video stream, $Fn(RFM(V_S{}''))$ is a method that returns the frame number and RFM is Re-constructed Frame Method.

3.4.1 Check Frame Sequence (CFS)

The CFS and the re-construction mechanism are used to identify the missing frames and build the video to it is original sequence. This is done according to the following checking procedure, as shown in figure 3.7:

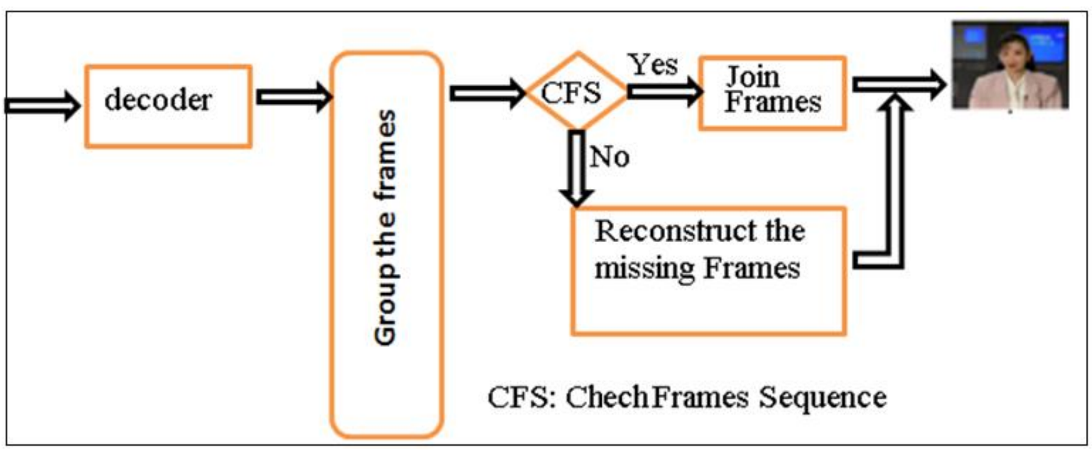
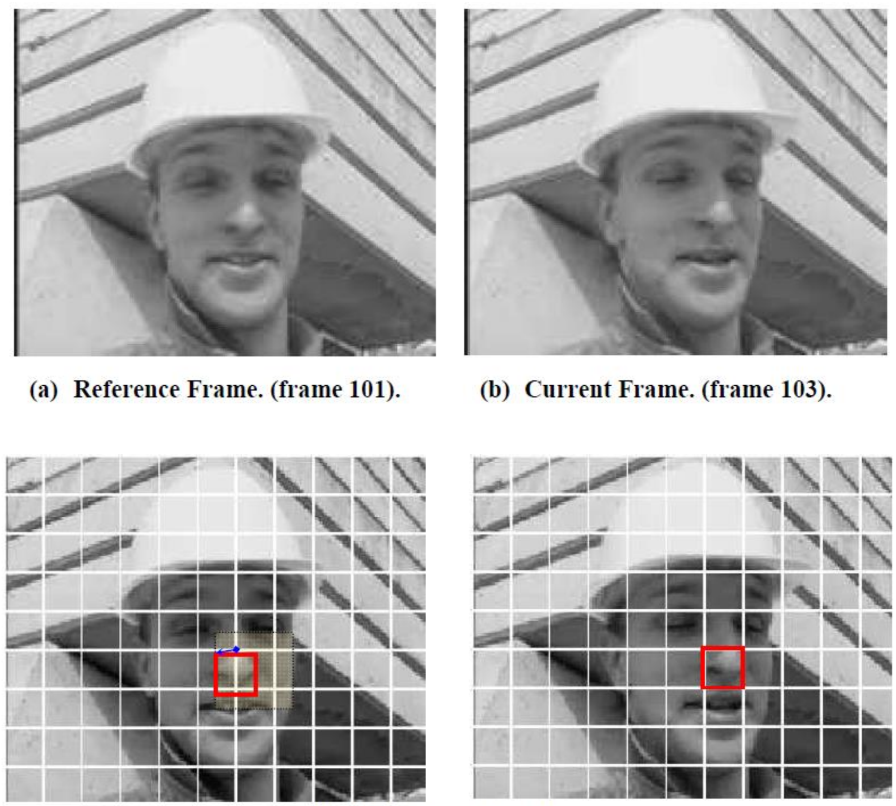


Figure 3.7: check frame sequence diagram.

3.4.2 Reconstructing the Video Frames

Assume frames f and (f+k) have been received and decoded, and are resident in the buffers waiting to be displayed, and that frames(f + 1), (f+2)... (f + k-1) have all been lost, as shown in figure 3.8.

The current frame (f + k) is divided into a matrix of macro blocks (16x16), which are then compared with corresponding macro block and its adjacent neighbors in the previous frame (frame f) to create a vector that specifies the movement of a block from one location to another in the previous frame. This movement calculated for all the blocks of a frame constitutes the estimated motion in the current frame, as shown in figure 3.8.



(a) Reference Frame. (frame 101). **(b) Current Frame. (frame 103).**

Block Matching Algorithm.

Figure 3.8: Motion tracking for frame estimation.

For each macro block of the current frame, current macro block in Figure 3.6, we will find matching block in the previous frame. Since in the previous frame we will not search for the best match in the whole image instead we will search in area constituted and up to p pixels on all its four sides.

The motion estimation technique is the same as operated in transmitter (FSM), good matching quality and high-performance. The full search technique of TSS is provided a best search matching for all possible combinations. However, such this meaner is provided a highly computationally costly, which doesn't desirable. Thus, to minimize the computations raised by full search technique, the scope of matching was limited; departure us with Limited Scope Full Search, see Figure 3.9. The input of the Limited Scope Full Search (LSFS) is two frames and it is output of a vector field that represents motion of blocks of pixels between frames. Limited Scope Full Search (LSFS) works in the following:

The parameter p is called as the search parameter. Larger motion larger p, and the larger the search parameter computationally expensive the process of estimation becomes. Usually, the macro as a square of side 16 pixels, and the search parameter p is 7 pixels. The idea is represented in Figure 3.7.

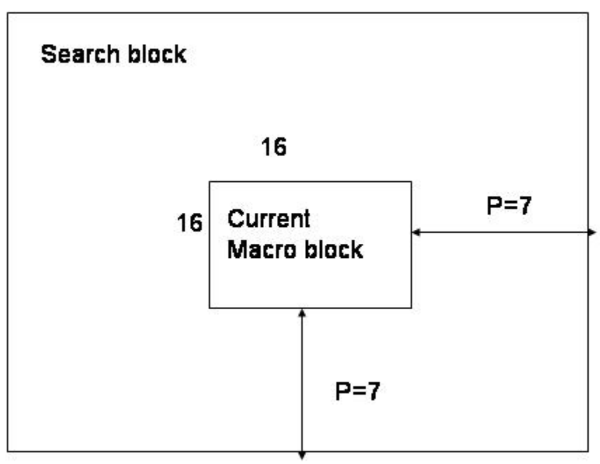


Figure 3.9: Block matching a macro block of side 16 pixels and a search parameter **p** of size 7 pixels.

The matching of one macro-block with another is based on the output of a cost function. The macro-block that results in the least cost is the one that match the closest to the current-block and its displacement (motion vectors) is recorded (see section 3.2.2). There are various cost functions, the most popular and less computationally is Mean Square Error (MSE) given by:

$$MSE = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (C_{ij} - R_{ij})^2 \qquad (3.3)$$

Where N is the side of the macro block, C_{ij} and R_{ij} are pixels being compared in current macro block and reference frame respectively.

After creating the block motion-vectors between frames f and (f + k), the vectors are then used to estimate the missing frames in the middle. The position of each block in the missing frames between frames f and (f + k) is created as linear displacement from the corresponding position originating from frame f to (f + k).

To reconstruct the dropping frames FRM uses vector field generated by LSFS to determine positions of corresponding blocks between frames and linear interpolation to calculate the positions of blocks in the dropping frames.

To additional enhance FRM padding algorithm values was presented in Figure (3.10). After Block-based frame interpolation, some crack- a region that is not filled may remain around the area of high motion activity. Figure 3. 10-(a) shows the screen shot of Foreman sequence before padding, which has a large number of cracks. Pixel-based frame padding fills a crack (i.e., unfilled pixels) using the adjacent interpolated pixels as shown in Figure 3. 10-(b). The pixels located at the edge of crack are filled first; then the pixels neighboring to the first-filled pixels are filled using the first-filled pixels, and so on. The adjacency between pixels is determined based on four-connectivity (left, right, top, and bottom). Using the available adjacent pixels, the current pixel ($P_{i,j}$) can be padded as defined in [91] as follows:

$$P_{i,j} = \frac{1}{np}\sum_{k=0}^{3} nbu(p_{i,j}, k) \quad (3.4)$$

Where nP is the number of the available neighboring pixels, and $nbu(P_{i,j}, k)$ the function to return the neighboring pixel values of $P_{i,j}$ according to direction k (i.e., left:0, right:1, top:2, and bottom:3). If no neighboring pixel is available, $nbu(P_{i,j}, k)$ returns zero. Cracks usually represent background regions and padding using the neighboring pixels yields a reasonable compromise between visual quality and computational complexity.

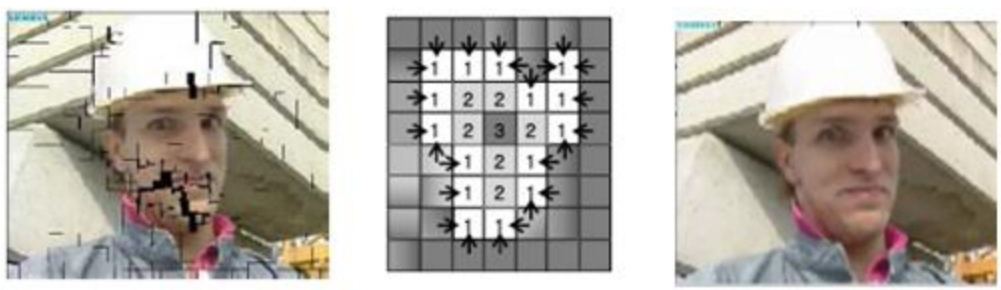


Figure 3.10: (a) Image before padding; (b) after padding algorithm.

3.5 Summary

This chapter discussed a proposed video streaming technique that addresses the shortcomings of streaming video over the unreliable networks by decreasing the amount of the video frames on the transmitter side and re-constructing them on the receiver side. The proposed technique uses the (FSM) on the transmitter to drop some frames from input video, relying on frames similarity, (i.e., depends on changes in motion between frames), and also employs the (FRM) on the receiver to re-constructed the dropped frames, (i.e., depends on interpolation technique).

It was found that the proposed video streaming technique is very adaptable while performing the set goals. Moreover, it was appeared that it is possible to bring down (or skip) some frames of input video depending on similarity between video frames (i.e., the motion estimation technique is used to emphasize speed of execution rather than quality of matching) and also, possibility to re-constructing the frames were dropped on the receiver-end.

Chapter IV: Simulation and Results Analysis

4.1 Introduction

To investigate the capabilities of the proposed technique a case study was conducted, as shown in table 4.1. The professional test videos with different characteristics and different importance region were selected as topics of this study are: 1) The Akiyo. 2) News, and 3) Foreman. All these samples are of QCIF (176 x 144) resolution, and 300 frames of each, but with different properties (see table 4.1). The first test sample (Akiyo) is generally displays facial movement. The second one (news) displays facial movement, and body, as well as the background movement, and the background scene change. The third sample (Forman), the video displays the face and body movement, as well as background and camera movement. Besides the body motion in Foreman video is great, at times taking place over half of video frame. To precisely compute and evaluate the results of the case study we will assume that the network runs without frame loss and the adapted stream will reach the receiver side as it.

The testing was conducted on a laptop with Intel core™ i3, 2.4 GHz CPU and 4 GB RAM running Microsoft Windows™ 7.

Table 4.1: The video clips types and characteristics

Video	Format	Size	Frame #	Characteristics
Akiyo.avi	QCIF	176x144	300	Low
News.avi	QCIF	176x144	300	Moderate
Foreman.avi	QCIF	176x144	300	Fast
Stefan.avi	QCIF	176x144	300	Fast

The outline of one run through of the case study is:

i. Apply FSM on the incoming video, this adapts the incoming video by skipping some frames of input video, according to the values of operational parameters;
ii. Apply RFM on the received Adapted Video (AV) for re-constructing the dropping frames from the adapted video stream;
iii. Calculate the measurements of video quality:
   - Calculate the PSNR -value of each frame of the re-constructed video;
   - Calculate the average PSNR -value of the re-constructed video;
   - Calculate the average PSNR -value of frame that are skipped;
   - Calculate the smallest PSNR -value of frames that are skipped;

4.2 MATLAB 2014a

In order to investigate the aforementioned research questions by means of effective by developing a simulator to study the performance of the video quality after the frames lost and re-constructed. The tool uses to study and evaluate the proposed performance is (vary famous software in academia and manufactories,) MATLAB version 2014a [92]. The name MATLAB stands for MATrix LABoratory [93]. MATLAB was written originally to provide easy access to matrix software developed by the LINPACK (linear system package) and EISPACK (Eigen system package) projects.

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming environment. Furthermore, MATLAB is a modern programming language environment: it has sophisticated data structures, contains built-in-editing and debugging tools, and supports object-oriented programming. These factors make MATLAB an excellent tool for teaching and research.

MATLAB has many advantages compared to conventional computer languages (e.g., C, FORTRAN) for solving technical problems[94]. MATLAB is interactive software whose basic data element is an array that doesn't require dimensioning. The software package has been commercially available since 1984 and is now considered as a standard tool at most universities and industries worldwide.

It has powerful built-in routines that enable a very wide variety of computations. It also has easy to use graphics commands that make the visualization of results immediately available. Specific applications are collected in packages referred to as toolbox. There are toolboxes for signal processing, symbolic computation, control theory, simulation, optimization, and several other fields of applied science and engineering.

4.3 The Operational Parameters

The operational parameters that are used to influence a number of frames and a quality of test videos that needed to send over wireless network, with the range of parameters combinations are: the Maximum-Change (***M-change*)**, the Reference-Frame **(*R-Frame*)**, and the Block-Size **(*B-size*).**

1. **M-change:** M-change describes the maximum value of changed motion vectors (MVs) allowed. It is employed in FSM (Frame Skipping Mechanism) to choose either to $keep\ or\ drop$ the frame(s) from the input videos. Thus, this choice is decided according to the degree of similarity between video frames (i.e., to the value of motion estimation between two frames). The higher value of $M{-}change$ give FSM more skipping frames, but it would be more difficult for RFM to re-construct the frame. Since the value of $M{-}change$ based on the size of the pixel block, we will use percentage representation. Therefore, if the $M{-}change = 20$ that means, frames with motion that is lower than 20 % differ from the last reference and hence, the frame will be skipped, and the others frames will be kept and streamed.
2. **R-Frame: -** this parameter adds reference-frame every n frames. It is presents the smallest number of frames that will be kept from the input video. The R-Frame range value is from 3 to the end of video frames. The R-frame is designed to place the reference-frames (full original frames) in different sequence positions in the streaming video. The first one is placed every 3^{rd} frame, e.g., 0, 3, 6, 9..., in the second one the reference frames are every 3^{th} frame, e.g., 0, 4, 8, 12..., in the third one the reference frames is every 5^{th} frame, e.g., 0, 5, 10, 15..., as shown in figure 4.1.

3. **B-Size: B-Size** the size of the pixel-block. Its presents the size of pixel blocks that are operated in FSM and RFM. The $B\text{-}Size$ value range is from 4 to the resolution of the original video.

The results of the case study were calculated by measuring different aspects of the re-constructed video using $PSNR$ metric. This result is used to judge the quality of the video samples, and also of the proposed technique efficiency.

This chapter will discuss the results of the case study and describes its importance. And also it will explain briefly the importance of each parameter and its impact on each of all test videos samples.

Figure 4.1: The proposed seniors

4.3 The Significance of Operational Parameters

Table 4.2 shows the characteristics of the operational parameters that are employed in the case study. The wide value ranges of the operational parameters were defined. Hence, the value ranges of the parameters that are selected from the first run on of the case study and represent the building of the prototype of the proposed technique that are achieved better results. Specifically if the value of $M\text{-}change$ is less than 20, it means not clearly affect at all. On the other hand, if $M\text{-}change$ is greater than 60, the quality of the re-constructed videos will be reduced significantly, and in some cases frames that reconstructing be incomprehensible. For this reason, the range has been chosen of this parameter is between $20 - 60$.

Moreover, through the first run of the case study, it found that the values of $R\text{-}Frame$ greater than 5, and less than 3 have a negative effect on the quality of the re-constructed frames, as well as frozen frames, and for this reason the values 3 and 4 were chosen as board values for parameter $R\text{-}Frame$, and 4 as an intermediate value. That was very hard to describe the range value for $B\text{-}Size$ parameter. Because there are no usually one unified standards exist for video frame format for example $QCIF$ and many others. Nevertheless, in the course of the previous work, it was saw that $QCIF$ format is considered with 64 and 256 of pixel blocks [34][60]. Another value of 4 was added to the proposed technique. This proved to be the right choice since with other parameters being equal the smallest pixel block creates somewhat better results in almost all cases.

Figure 4.2 presents that how the proposed technique deals with Akiyo test video, with all parameter combinations. Besides, it can find in Figure 4.2, the skipped frames number hits or near to the maximum value allowed, when the $M\text{-}change$ value touches 30. The maximum number of frames that were skipped as follows:

- $when\ R\text{-}Frame = 3$, the maximum number of skipped frames ≈ 60;
- $when\ R\text{-}Frame = 4$, the maximum number of skipped frames ≈ 90;
- $when\ R\text{-}Frame = 5$, the maximum number of skipped frames ≈ 108.

Table 4.2: the Importance of the **parameters**

Parameters	Values Range	Relation To Quality	Impact On Quality
M-change	From 20-60	Intensively proportional	High
R-Frame	3,4 and 5	Intensively proportional	High
B-Size	4, 64 and 256	Not-clear relationship	High

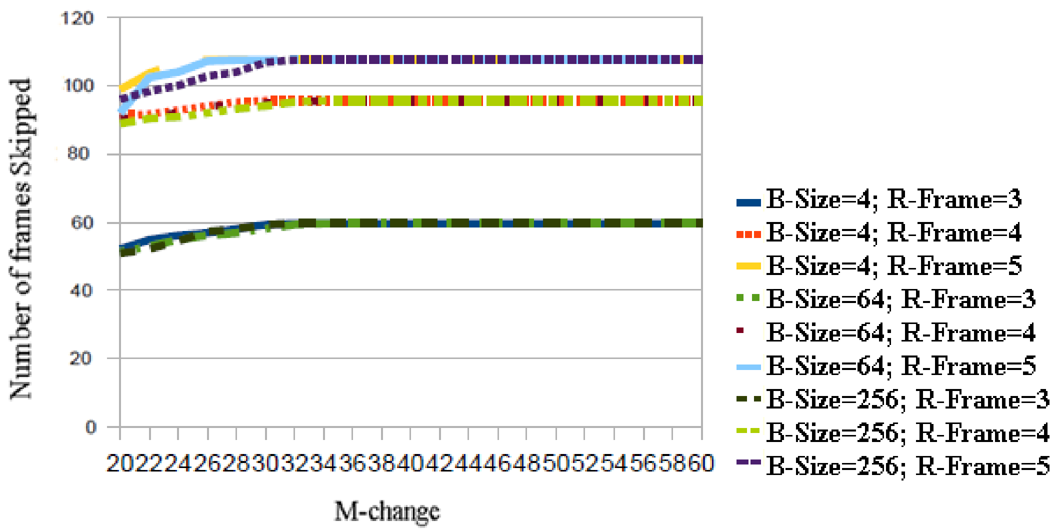
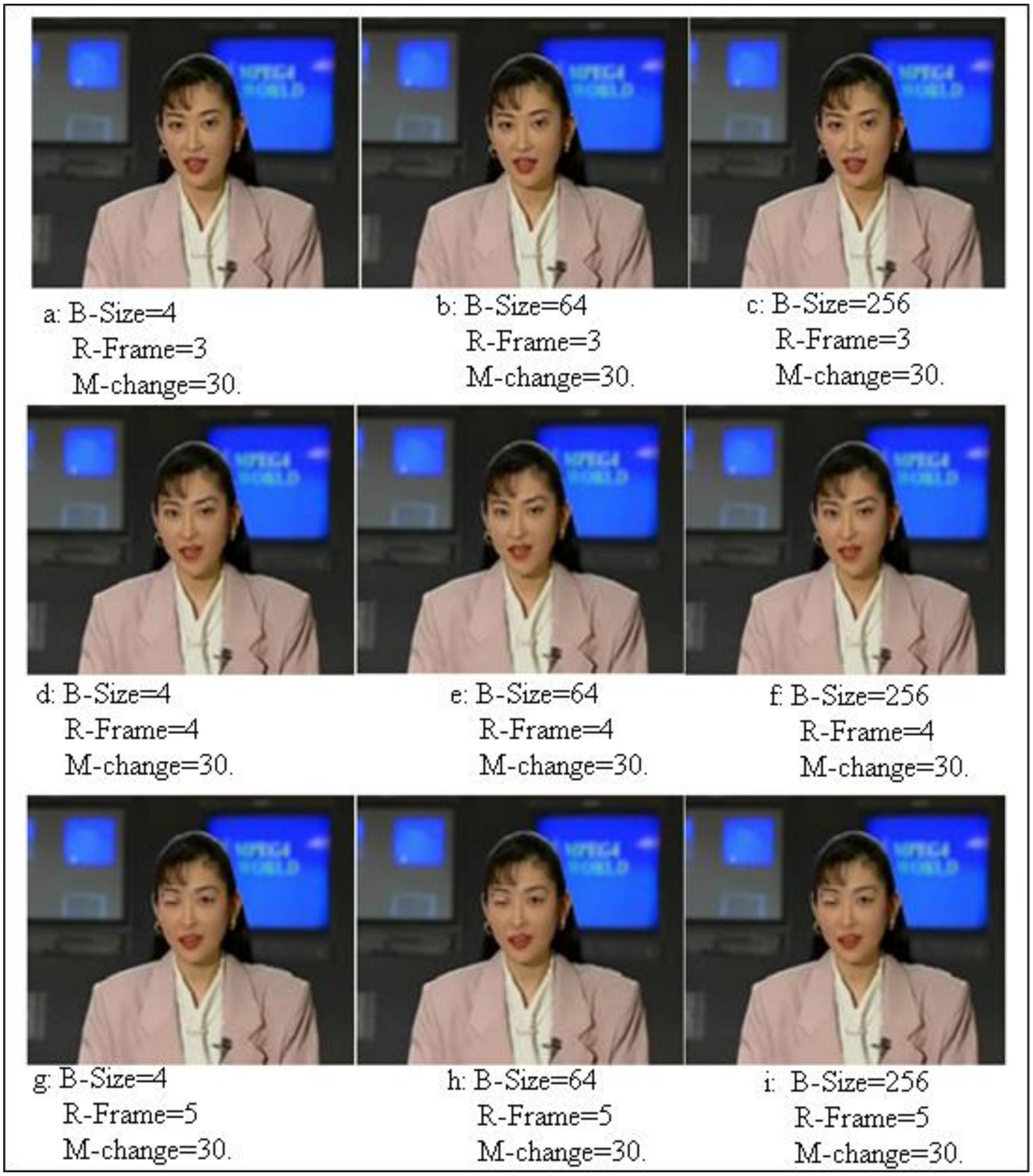


Figure 4.2: Akiyo sample: the operational parameters V.S the number of frames skipped.

In case of *Akiyo* sample, to reach the maximum number of skipped frames is very easy, since the video is mostly displays facial motion, and is comparatively easy to interpolate (instead of other samples).

It can be said that our proposed technique doesn't explore all possibilities regarding this particular test videos. Figure 4.3 is shown a set of results of the case study, but is certainly atypical. However, it can be said that it is difficult to define a universal set of operational parameter values which are explore all possibilities of the three test video clips. This is because the characteristic of motion shown in test samples (Akiyo, News, and Foreman videos) is different. In regards to latter statement it must be said that it is already hard to recreate a video with half of its frames missing (note that all three test videos have 300 frames each) and the complexity keeps increasing as the ratio of skipped frames goes up. It also must be said that Figure 4.2 (as well as Figure 4.4 and Figure 4.10) doesn't represent the quality of recreated videos in any way. Figure 4.3 shows the same frame (FRAME NUMBER 13) that was skipped and re-constructed by using different parameters.



a: B-Size=4 R-Frame=3 M-change=30.

b: B-Size=64 R-Frame=3 M-change=30.

c: B-Size=256 R-Frame=3 M-change=30.

d: B-Size=4 R-Frame=4 M-change=30.

e: B-Size=64 R-Frame=4 M-change=30.

f: B-Size=256 R-Frame=4 M-change=30.

g: B-Size=4 R-Frame=5 M-change=30.

h: B-Size=64 R-Frame=5 M-change=30.

i: B-Size=256 R-Frame=5 M-change=30.

Figure 4.3: Frame 13 snapshots from $\boldsymbol{Akiyo}$ vide after re-constructed, using different operational parameters values.

Figure 4.3 presents that when the parameter R-Frame value increases that is leads to increasingly worse image quality (i.e. increases the maximum number of skipped frames). In fact that, the $R\text{-}Frame$ values have a greatest effect on the re-constructed videos quality. The study also showed that when the $R\text{-}\ Frame$ value = 4, drops in quality is very noticeable. More increasing value of $R\text{-}Frame$ leads to more dropping in the quality, when $R\text{-}Frame$ value=5 garbles the image quality, despite that it is still readable. The values of the $B\text{-}Size$ parameter have no clearly effect in the test samples, but it is more important when dealing with videos have more movement.

The figure 4.4 presents the results of the $News$ test video with all combination of the operational parameters. It appears that, with the Akiyo test video, it was easy to reach the maximum number of skipped frames with all parameter combinations. But, with the $News$ test video usually it needs higher value of M-change. General from chart, the number of skipped frames is more different than from the case of $Akiyo$ test video. In particular, with smaller block-size of pixel the number of skipped frames is between, 60 to 107. The $News$ video has smaller movements and with smaller block-size of pixel $the\ FSM\ has$ a small error margin when choosing frames to be drop. And also, with a big pixel of the blocks size, small movements don't organize changes enough to keep the frame. It will be presented in part 4.3 that the re-constructed frames quality is better with small pixel of block size; but the reduction in quality is negligible.

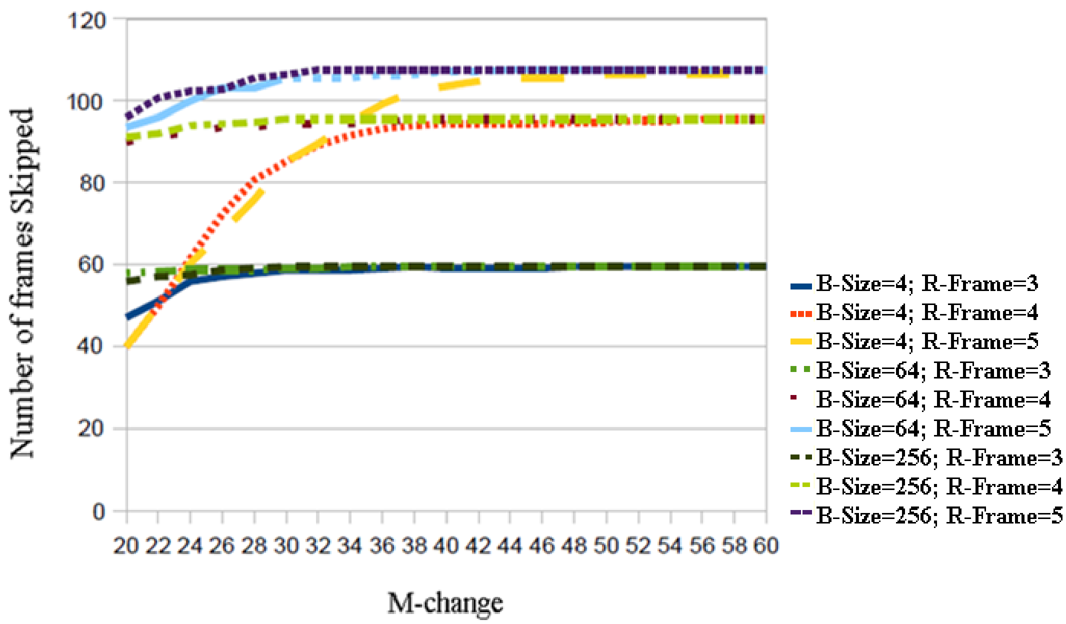


Figure 4.4: News sample: the operational parameters V.S the number of frames skipped

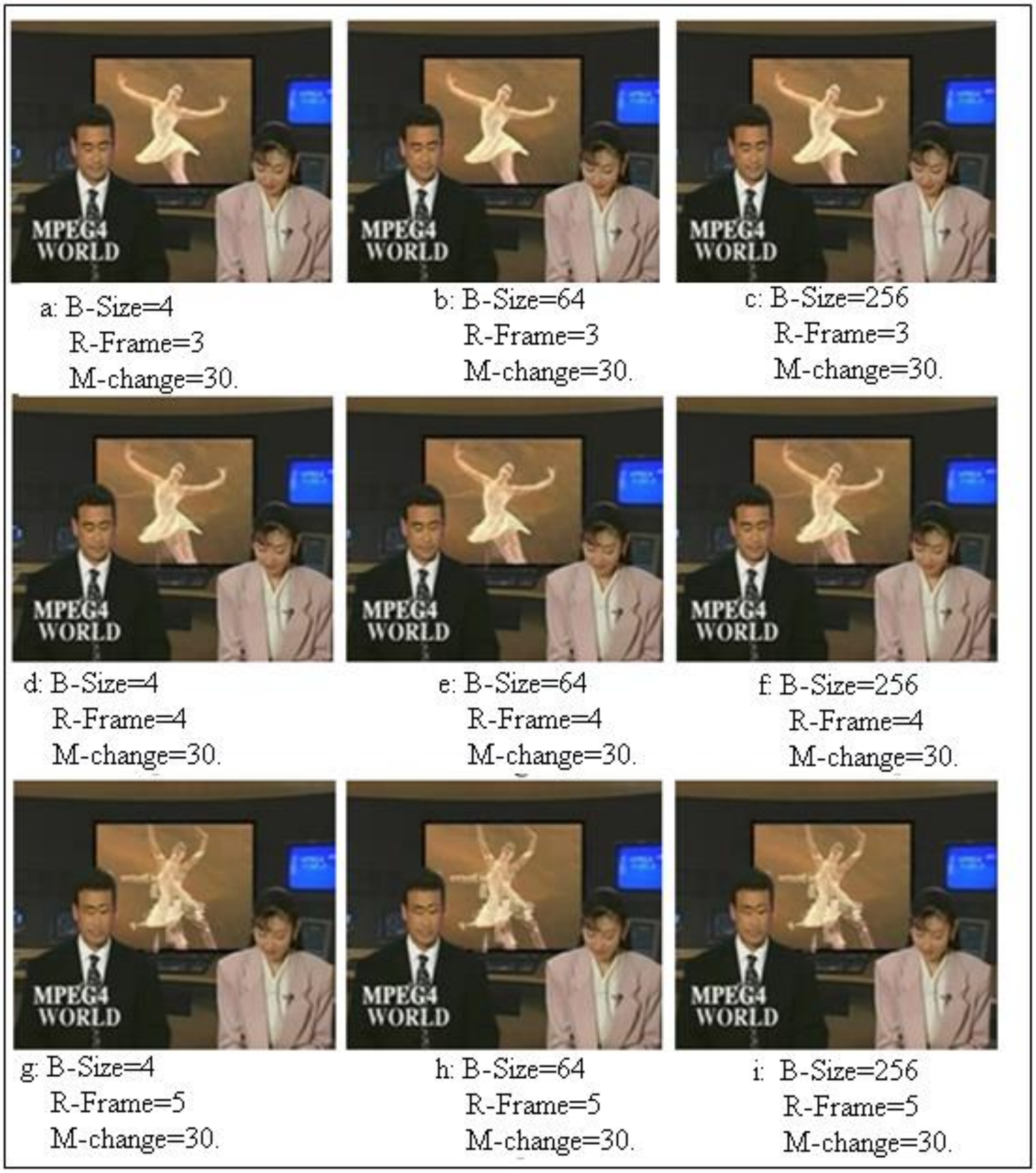


Figure 4.5: Frame 136 snapshots from **News** vide after re-constructed, using different operational parameters values.

The figure 4.5 presents the relationship between $operational\ parameters$ and $video\ quality$ of the reconstructed frame number 136 from $News$ test video, with all different parameter values. It is alike to $Akiyo$ test video. However, lower values of $R\text{-}Frame$ achieved better results. It is noticed that in figure 4.4, frame snapshots $(a, d, \text{and} g)$. When $R\text{-}Frame$ increases from 3 to 4 and, then to 5 the frame gets more distorted, while the other parameters are same. However, it can say that, the bigger pixel of blocks ($B\text{-}Size$) don't effect to the movements. In figure 4.3, take into account that according to the parameter values shown in frame snapshots b and e. You can found that more than $2/3$ of the video frames were dropped in b and e (when $R\text{-}Frame = 4,5$). Hence, in the $News$ video clip, small movements ($male\ newscaster\ face$), and big movements (ballerina) are reconstructed basically in the same as $frame\text{-}snapshots\ a$, and d (refer to figure 4.5).

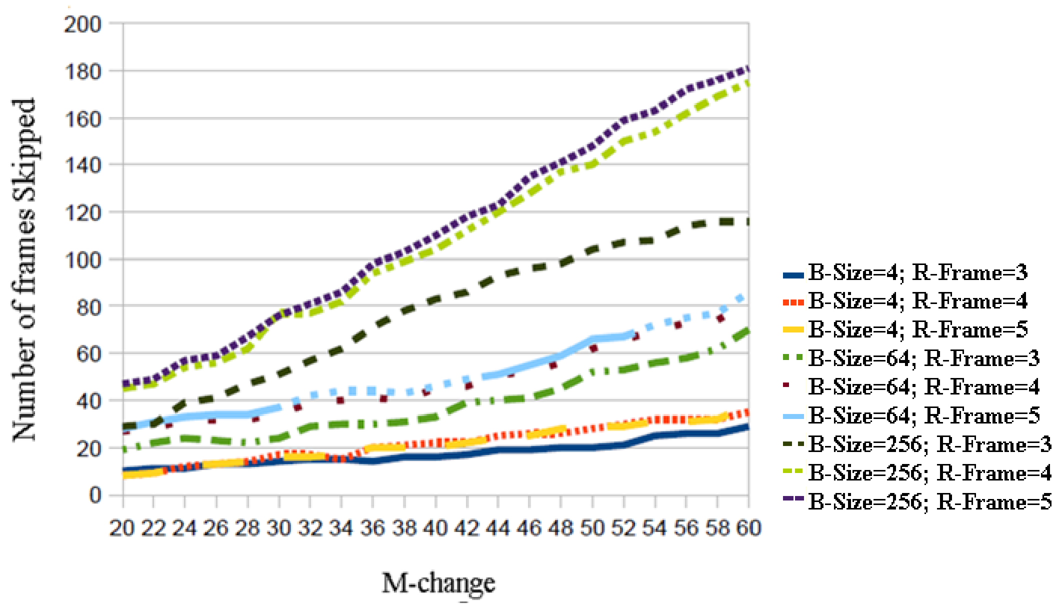


Figure 4.6: Forman sample: the operational parameters V.S the number of frames skipped

Figure 4.6 presents the video streaming technique and how it is copes with the video clip $Foreman$ with all possible combinations of operational parameters. It's different from the two test samples ($Akiyo\ and\ News$), from the chart above; the skipped frame number is differs from both test samples chart. Moreover, the target of skipped frames (Maximum) is not got with any combinations of operational parameter. This is due to the video $Foreman$ contains more intensive movements of all test video samples (i.e. $Akiyo$, $News$ and $foreman$). Additionally, the video $Foreman$ has large movements. Furthermore, in some parts of Foreman video the movement constitutes more than half of frame resolution. Thus, in this case the video streaming technique would achieve well witha bigger pixel block-$size$. This can be seen in Figure 4.6 that with bigger $B\text{-}Size$ values enables FSM to skip more frames. Specifically, if given careful consideration in the black-line in the Figure 4.6. With $R\text{-}Frame = 3$, the skipped frames number is three times largest than all possibilities. Hence, it could be conclude, the bigger pixel-blocks size deals with bigger movements better, and here $R\text{-}Frame$ is less significant in videos with intensive movement.

Figure 4.7 Frame 82 snapshots from **Foreman** video after re-constructed, using different operational parameters values.

Figure 4.7 presents that frame 82 snapshots i is distorted, and the face of the man is not clear. A part from that it is notice there, and that little to no change when increasing $B\text{-}Size$ as well as with $R\text{-}Frame$. However, we conclude that the larger pixels of blocks are more appropriated to deal with videos of intensive movements. We can summarize the significance of the operational parameters and their impact on the proposed video streaming technique in the following:

M-change:

The parameter $M\text{-}change$ is the main element in the FSM method. The $M\text{-}change$ values are even more significant in dealing with the videos of much movement. In one hand, in such cases a big range of values can help to achieve the desired result, either reduces, increases the skipped frames number, or quality of the videos. On the other hand, dealing with the videos with small movements, it is important only to find the value of $M\text{-}\ change$ that will be achieved the maximum skipped frames number. In such cases, if values of $M\text{-}\ change$ increase, it will not have any effect. As such choose a universal value for this parameter is quite impossible. One of the possible solutions to choose the value of $M\text{-}\ change$ based on the $PSNR$ average value of re-constructed videos. Hence, this needs the proposed technique would be used with all possible values of $M\text{-}change$ on the target videos.

R-Frame

The $R\text{-}Frame$ parameter has a greatest effect on the re-constructed videos quality. Nevertheless, the parameter $R\text{-}Frame$ with a lower value prevents the proposed technique to reach its full potential. Its notice through the sample frames of the three videos obtainable above, it could be seeing that when $R\text{-}Frame$ value increase to 4 the loss of frame-quality is acceptable. Select values for $R\text{-}Frame$ isq uite clear. It means, lower $R\text{-}Frame$ values will skip fewer frames (and this will increase the quality of the video, which was rebuilt) with $R\text{-}Frame = 3$ frames being less possible value. The higher value of $R\text{-}Frame$ is identified by a number of video frames have. Though, when increasing $R\text{-}Frame$ to 5 the interpolation video frame is become very complex because the frames of videos will rebuilt from less than 60% of their original sizes.

B-Size

The B-Size is a special parameter case. This is because the motion detection and the motion estimation techniques used in this study was not intended for $QCIF$ video format. Furthermore, these techniques operate worst with videos of a small frame resolution[70] and [34]. It could be argued that it was a mistake in the selection of such a video format unusual for testing issues. The reason for this is that, at the beginning of this study was aimed at mobile devices. But during the course of the study the emphasis has shifted to building a theoretical base for video streaming technique, which can handle with all video resolutions. And also found that there is no clear relationship between $the\ B\text{-}Size$ has and video-quality. Because, all parameter values were considered has created good results on isolated cases. However, there are no steady patterns or dependencies were found. It is noted that, with the smallest $B\text{-}Size$ values created best frames results most of the time.

It can see that, the operational parameters analysis and their effect on the frames re-constructed quality of the case study of the proposed video streaming technique. The next section will describe how quality is understood, measured and calculated in the scope of this study. Moreover, it will be described what suitable quality is and what is not.

4.4 Video Quality Evaluation

Currently, multimedia data are everywhere; and multimedia evaluation plays an essential role in the development and validity of image and video applications, like video compression and enhancing. There are two main types of quality assessments,

$subjective$ and $objective$ quality assessment [26], [95]. The subjective quality assessment is evaluated subjectively by a human viewer (eyes). However, it's time-consuming and expensive, and cannot be employed in real-time applications. To account for these drawbacks, a number of objective test methods have been developed, in which the quality rating is automatically calculated. Therefore, the expert must choose the objective metric very carefully in order to get appropriate results. On the other hand, the objective one can measure between un-distortion differences and distorted image quality signal. There are several aspects which are effect on multimedia quality such as blur/sharpness, noise, color fidelity, resolution, geometric distortions, frame dropping and freezing, and so on. There are three categories of Objective quality metrics according to their input data. The first representative of objective metrics is mostly used ($PSNR$) Peak Signal to Noise Ratio. This metric requires both original and distorted image in order to be computed (Kim et al., 2014). Therefore, it is a full-reference method (FR). Other FR metrics are based on Structural Similarity or Image Evaluation based on Segmentation. Then the second category is reduced-reference (RR) metrics, it used when we have just little information about the original image. There are many examples of reduced reference metrics can be found e.g. in [1][49][60]. The final one is no-reference metrics. This it used when we do not need any information about the original image or video. Thus, they can be used in cases, where $theFR\ and\ RR$ metrics cannot be. However, to correctly calculate the efficiency of the proposed technique a measurement (objective method) of video quality of the re-constructed video frames is needed. In this work $PSNR$ method is used to measure the video quality.

4.4.1 Measurement Tools

After videos reconstructed the peak signal-to-noise ratio ($PSNR$) is used to evaluate the quality of the video. $PSNR$ is a most commonly an objective method that employed to measure the quality of videos[40]. $PSNR\ is$ a ratio between maximum signal power (MSP), and the power of interfering noise, represented by mean squared error (MSE) is defined in [49]:

$$PSNR = 10 * \log_{10}\left(\frac{MSP^2}{MSE}\right) \qquad (4.1)$$

MSE *is* a representation of amount of average differences between pixels of re-constructed video and the original video:

$$MSE = \frac{1}{N^2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}(C_{ij} - R_{ij})^2 \qquad (4.2)$$

where N is the side of the macro block, C_{ij} and Rij are pixels being compared in current macro block and reference frame respectively.

MSP, represents the maximum possible value of the pixel computed by

$$MSP = (B^2 - 1) \qquad (4.3)$$

B is number of bits per-color channel.

The PSNR is measured in decibels (dB) on a logarithmic scale with values from zero to infinity, the higher value is the better results. This tool is used to compare original video with reconstructed one.

Frame 140
PSNR=30.9

Frame 230
PSNR=25.0

Frame 6
PSNR=29.0

Frame 145
PSNR=19.5

Figure 4.8: Examples of frames with varying quality.

As shown in figure 4.8, the Frame number 140 from $Akiyo$ test video that has the lower value of $PSNR$ above 30 out of all the results. Likewise Frame 145 from $news$ video clip has the higher value of $PSNR$ below 20. We also note that, frame i from $Foreman$ video clip in Figure 4.7 has the worse value of $PSNR$ 19. 01 of all results overall.

4.4.2 Video Frame Quality

A panel of users evaluates the case study according to Peak Signal Noise Ratio (PSNR) measurement, as shown in Figures 4(9, 10, and 11). The original videos are decoded with a bit-rate of 64 kbps, and the observers evaluate the videos after the frames lost and re-constructed.

It is been noticed from that, the PSNR score is high for the videos with statics background, instead to the videos with dynamic background e.g., Akiyo. The PSNR score for the dynamic background its show acceptable rate, e.g., news, while the videos that are shaking, e.g., Foreman video shows the worst evaluation from the user's panel, as the re-construction mechanism had high-effect on the video, as shown in the following figures.

The figures show the PSNR scores for all frames in all test videos with given parameter values (when M-change =30, R-Frame=3, and B-Size=4).

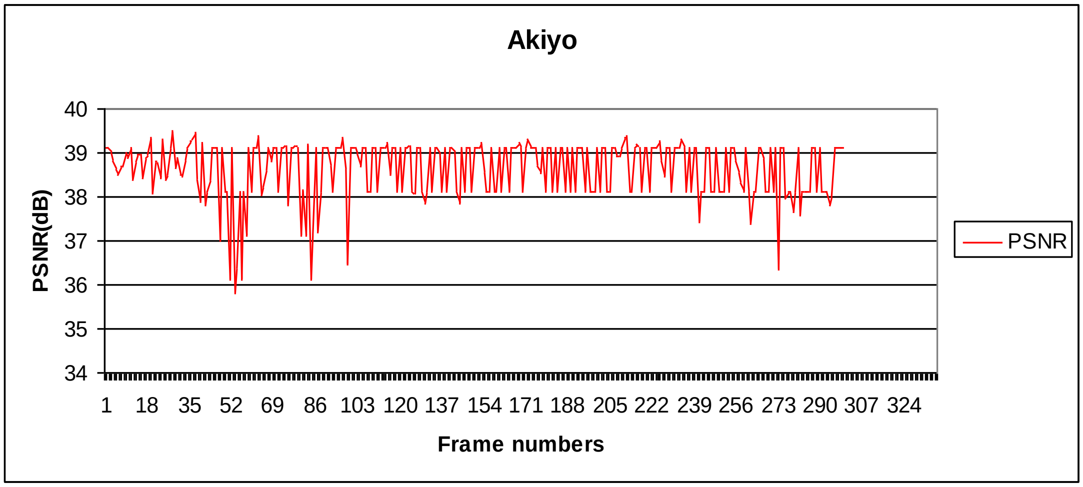


Figure 4.9: the PSNR score for all Frames in the Akiyo test video with a given set parameters.

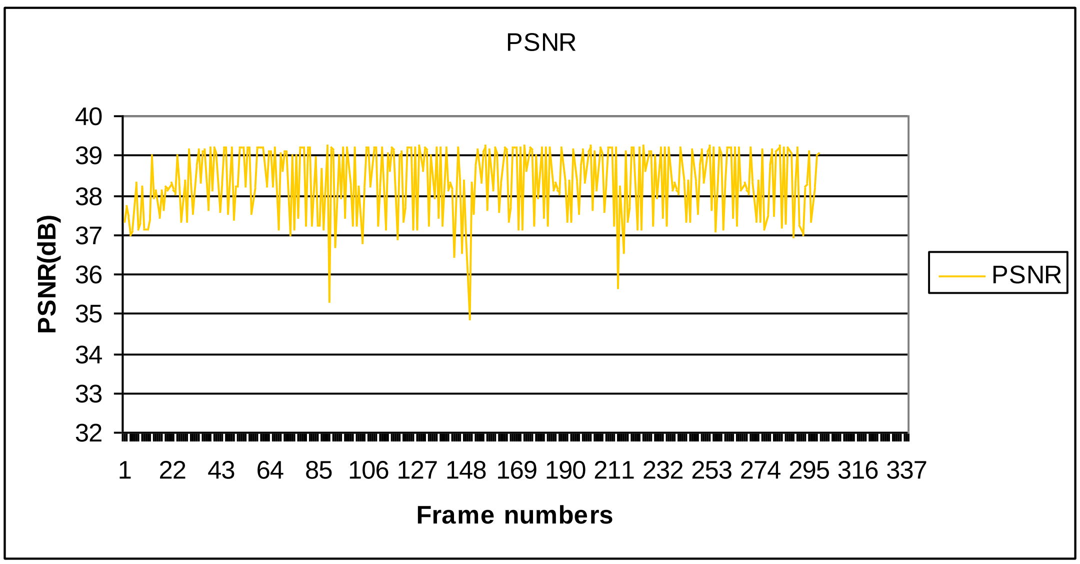


Figure 4.10: the PSNR score for all Frames in the News test video with a given set parameters.

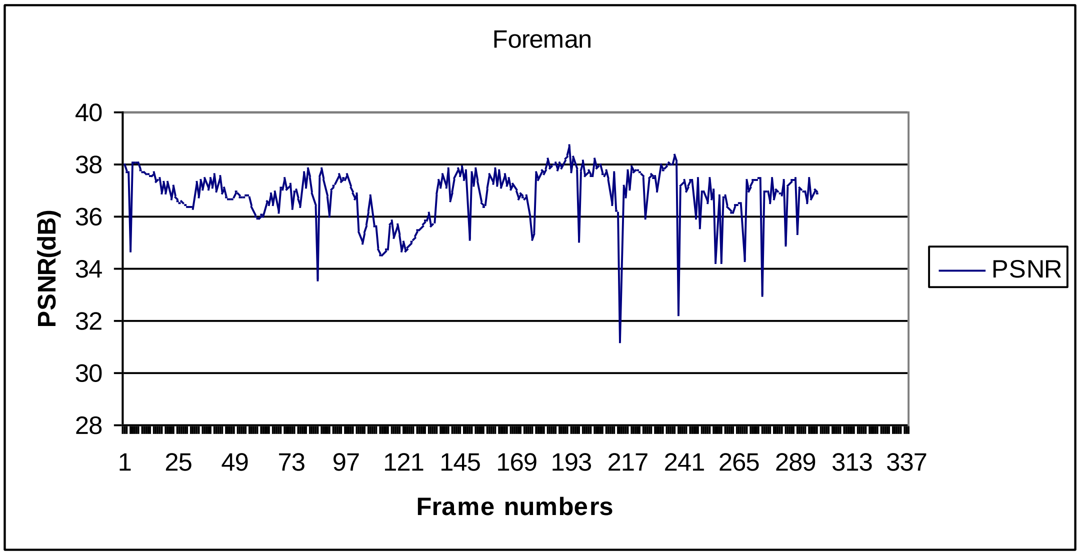


Figure 4.11: the PSNR score for all Frames in the Foreman test video with a given set parameter.

The figure 4.9 presents the values of $PSNR$ for each frame of all test videos ($Akiyo, News\ and\ Foreman$) with given parameter values. These parameter values setup is concentrated on the quality of the video (see Table 4.1). The parameters R-$Frame$, and B-$Size$ have the lowest values and the value of M-$change$ equal 30. From the chart of figure 4.9, the values of PSNR score do not drop down below 35. Furthermore, the most of the video frames have PSNR score above 38, and only the $News$ test video has values of $PSNR$ in between [34-38] as shown figure 4.10. This happened because the $News$ video has intensive movements of all test videos. Further, it is noticed that, from figure 4.11 the Foreman video frames have a $PSNR$ values all time above30, and has PSNR scores in the interval [31, 39] rather frequently. The lowest PSNR scores from this set are depicted in Figure 4.10 from news test video: Frame150 and Frame90 . Based on $PSNR$ value, these frames have lowest quality of all frames displayed in given set of operational parameters.

Figure 4.12: Snapshots of frames with lowest PSNR value
When: R-Frame=3; B-Size=4; M-change=30.

Figures below show the PSNR values of the videos Akiyo, News and Foreman with increases the R-Frame to 4, B-Size to 64, and M-change to 30. The quality of each frame as follows:

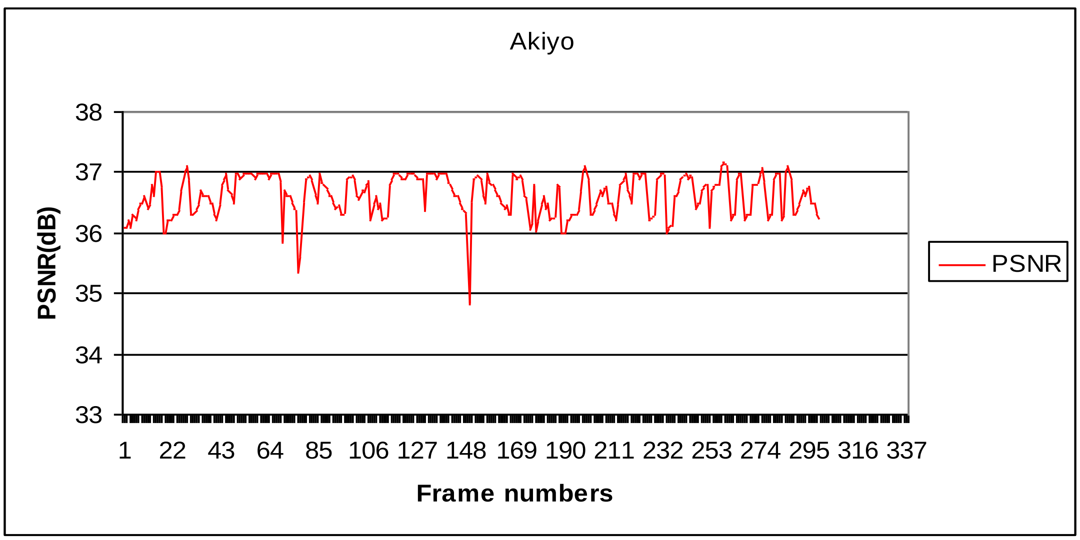


Figure 4.13: the PSNR score for all Frames in the Akiyo test video with a given set parameters.

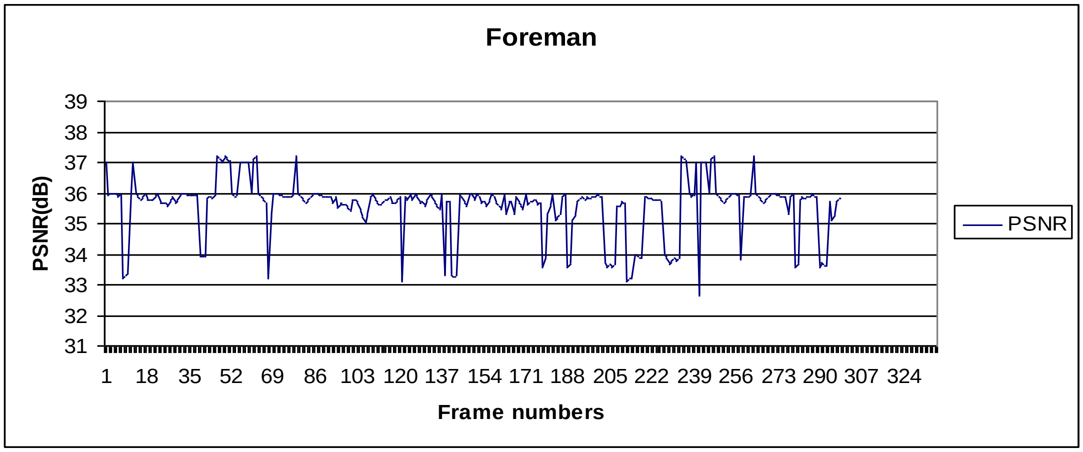


Figure 4.14: the PSNR score for all Frames in the Foreman test video with a given set parameters.

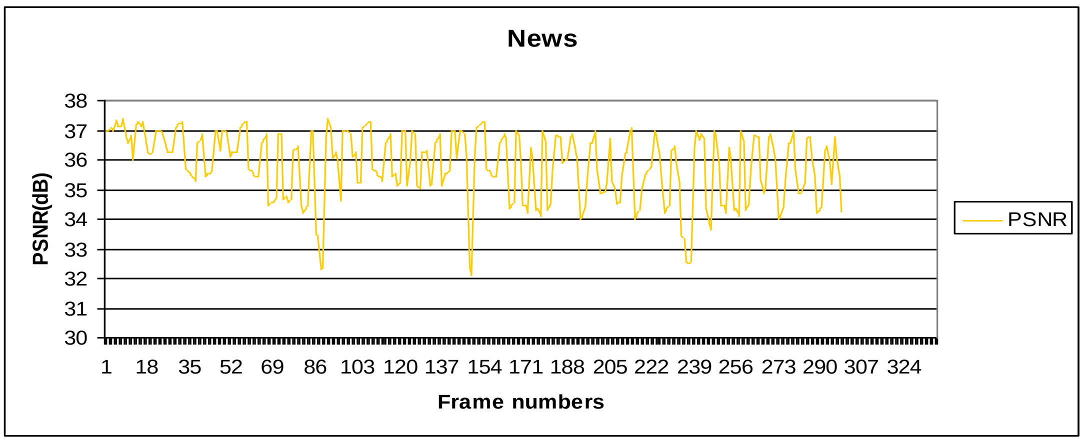


Figure 4.15: the PSNR score for all Frames in the News test video with a given set parameter.

It was shown that the parameter R-Frame has the largest effect on the quality of the re-constructed videos. Figures 4(13, 14 and 15) show how the average quality drops after increasing R-Frame value to 4. It is noted that for this set of samples a block of 64 pixels was used since in this case it shows slightly better results than other pixel block-sizes (i.e., when B-Size increases from 4 to 64).

Figure 4.13 shows the Akiyo test video still has a PSNR score above 34 dB all the time, consider the fact that, with these parameter values the frames were re-constructed from $1/5^{th}$ of original video size. In the case of figure 4.14 and 4.15, the videos Foreman and News, the average quality drop is noticeable; however, the PSNR score drops below 32 very occasionally and generally stays in the interval [32, 39].

Figure 4.16 shows the frames with the worst quality PSNR scores in this set of samples from News test video (see figure 4.14). Both of these frames include a background scene change, which is one of the more difficult changes to re-construct.

Figure 4.16: Snapshots of frames with lowest PSNR value.

The figures below show the PSNR values of all test videos: Akiyo, News and Foreman when increases the parameter R-Frame to 5, B-Size to 256, and M-change to 30. The quality of each frame as follows:

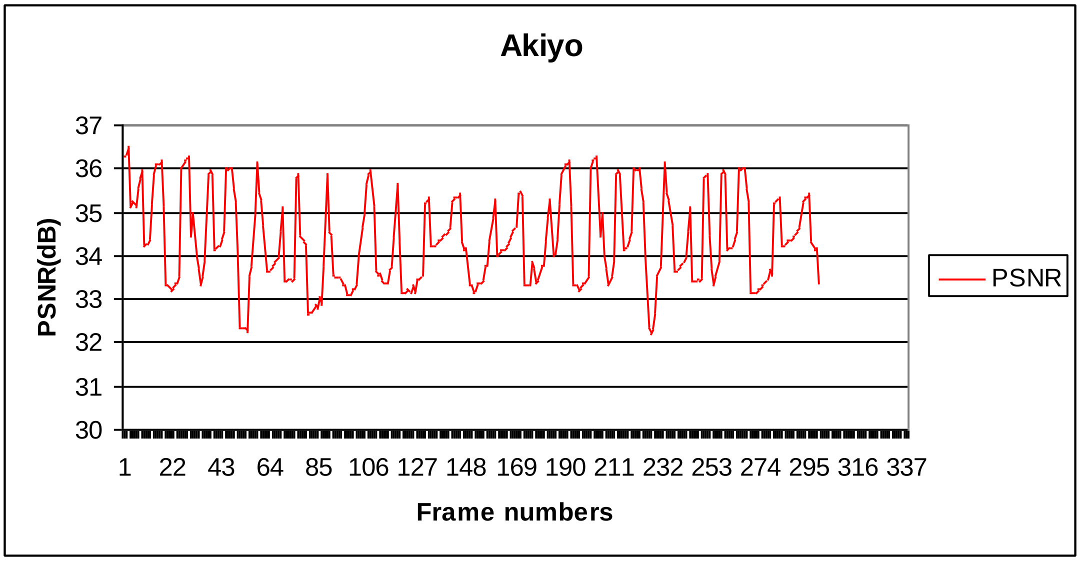


Figure 4.17: the PSNR score for all Frames in the Akiyo test video with a given set parameters.

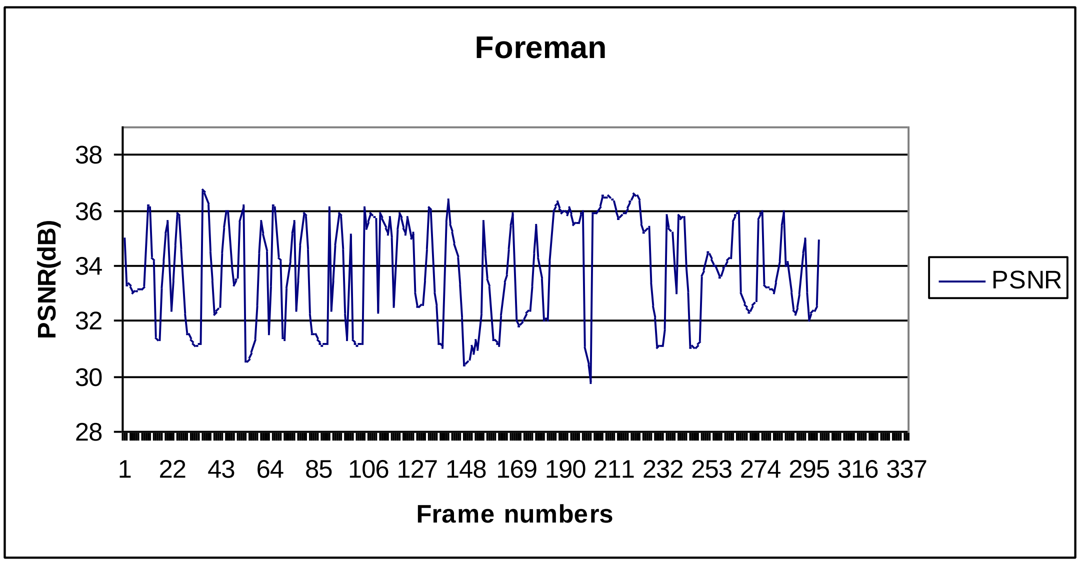


Figure 4.18: the PSNR score for all Frames in the Akiyo test video with a given set parameters.

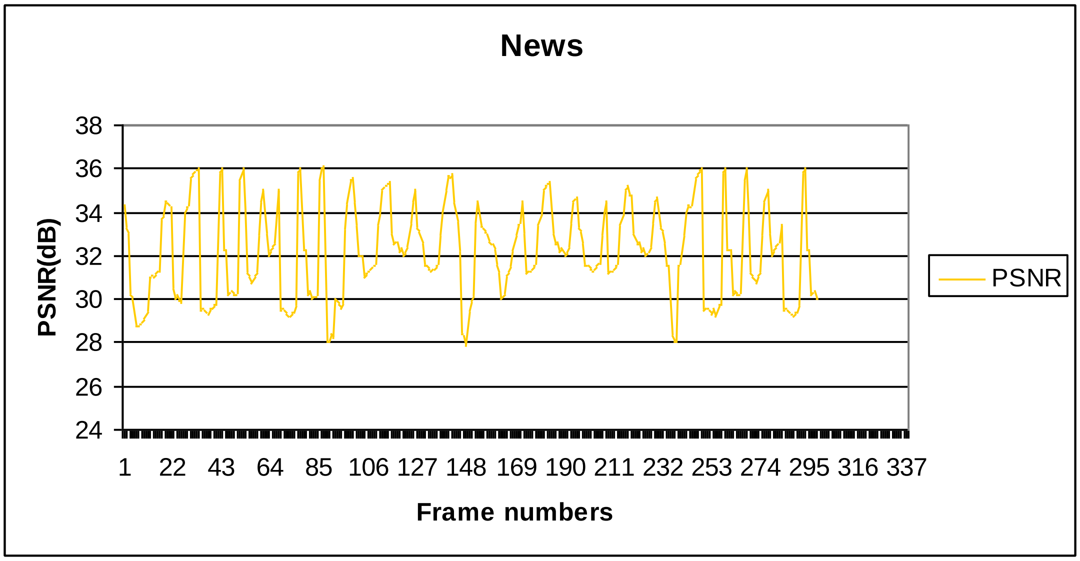


Figure 4.19: the PSNR score for all Frames in the Akiyo test video with a given set parameters.

Figure 4(17, 18 and 19) show the PSNR scores of all test videos: Akiyo, News and Foreman with the biggest parameter values of R-Frame and B-Size (5 and 256) respectively. This was done because it is one of the parameter configurations with which the lowest PSNR values are reached. In this case even the quality of Akiyo test video frames quality drops below 30 (although very occasionally and the average quality of this video is still above 30) (refers to figure 4.17). Further it is shown that the frames quality of News test video (figure 4.19), and particularly Foreman test video frames quality dropped dramatically (see Figure 4.18). In fact the most of PSNR scores of the frames of these two videos (News and Foreman) are around 32 and the difference between the PSNR scores of re-constructed frames and reference frames is around 28 dB most of the time. This means that the difference between the original video frames and the re-created frames is high and there might arise coherency issues. Very occasionally does the PSNR score drop below 28, even with this aggressive parameter configuration. Another important fact is that the lowest PSNR score belongs to the frame of Foreman video although in Figures 4.20. Figure 4.18 shows the re-constructed frames with the lowest PSNR scores in this set of samples.

Figure 4.20: Foreman frames snapshot with lower values of PSNR.

4.5 Validity of the Results

The main question in this study that needs to answer is how to determine the movement intensity in the videos. In this study different types of movement intensity were examined (i.e., simple, moderated, and complex movements). To evaluate and measure the usefulness of our proposed technique, different types of movements should be employed. Hence, the most of the movement types have been found in the three test videos samples that are upsetting in the topic of video interpolation. The movements are: 1) facial movement can be found in all test videos (Akiyo, $News\ and\ Foreman$); 2) background and foreground movement can be found in $News$ video; 3) background and foreground scene change can be found in $News\ video$, 4) camera movement can be found in $Foreman$ video; 5) background land-scape movement can be found in $Foreman$ video; 6) the body movement can be found in $Foreman$ video.

In one hand, the problem faced by the proposed technique is that all three test samples (videos) were all format, such format is unfamiliar in the subjects of video interpolation and motion estimation. When using the $QCIF$ format motion estimation technique generate the worst result [70]. The elimination of this type of threats on the validity test of the videos, different video formats should be used and tested. The fact is that this problem will discuss in the last chapter (in future work).

On the other hand, threatening the validity based on the method that used to evaluate the re-constructed videos quality. Generally, there are two types of methods to measuring the quality of videos; either objective or subjective quality assessment method. The objective quality assessment method is the ($PSNR$) Peak Signal to Noise Ratio method (Lee et al., 2008), and (MOS) Mean Opinion Value (Itoh et al., 2000) that is a subjective quality assessment method.

This study selected the $PSNR$ method, to the practical reasons. Specifically, the number of times run of the case study was repeated more than 2000 times. These include the first test runs, repeats for more investigative, repeats for more result confirmation, repeats for more identifying the operational parameters ranges values etc., it was impractical, if not impossible, to conduct a survey in each time to measure the video quality that was re-constructed. For this reason $PSNR$ scheme was selected to calculate the re-constructed videos quality.

4.6 The PSNR Results for interpolated frames

Table 4.3 shows the PSNR score improvement from Frame Rate Up-conversion with Interpolation (FRUI) (López et al., 2008) to our interpolation technique. Despite the average result values are quite good in all cases, the improvement from FRUI change a lot depending on the test videos sequence. Obviously, every possible implementations of our technique with frame re-construction will have better results according to the following table 4.3.

Table 4.3: The PSNR Results for interpolated frames

Test sequence	Frame Rate-Up Conversion with frame interpolation (FRUI) (dB)	Our method (dB)
Foreman	34,69567	36,45561
Akiyo	40,36142	39,36122
News	34,99790	35,09150

4.7 Summary

This chapter discussed the results of the case study and describes its importance. And also it explained briefly the operational parameters and an importance of each parameter and its impact on each of all test videos samples.

The results of the case study were calculated by measuring different aspects of the re-constructed video using *PSNR* metric. This result is used to judge the quality of the video samples, and also of the proposed technique efficiency.

This chapter presented the results of the case study and described the importance of each test video. And, also explains the importance of the operational parameters and their impact on all the test video. To reduce the amount of data that are streamed over wireless networks, a video streaming technique is proposed in this study. The technique is based on identifying and extracting the motion FOI by computing the Sum of Absolute Differences (SAD) for video streaming.

Four different FOI cases for three different scenarios are proposed to evaluated each video, corresponding to send the full frame information (reference frames) every 3rd, 4th, and 5th frame, respectively.

The proposed adaption scheme is compressed by H.264 codec to study the effects on the video size, by sending the reference frames more often, e.g., every 3rd frame instead of every 4th and 5th frame; it will increase both the PSNR and the file size.

Chapter V: Conclusions

5.1 Introduction

This study proposes the video streaming technique, which to deals with the problematic of wireless networks by reducing the video frames number to be sent in transmitter, and re-constructing the dropped frames on the receiver-end. $Frame\ Skipping\ Method(FSM)$, of the proposed technique is employs on the transmitter to skips videos frames, based on motion changes between video frames, and $Frame\ Reconstruction\ Method(FRM)$ uses on the receiver-end to re-construct the videos dropped frames. The proposed technique has shown high adaptable during achieve the set aims. Moreover it was found that, it is possible to reduce the video frames depends on motion changes between video frames, and that it is possible to re-construct the dropped frames using interpolation frame technique on the receiver side.

From the results evaluated, it was found that, the proposed video streaming technique is very flexible technique. Because, it was demonstrated that, the operational parameter values enable the proposed technique to drop up to half (50%) of the original video frames; the re-constructed video frames values of $PSNR$ is never drops below $25dB$. Through operating aggressive values of the operational parameter, it is possible to reduce the original video frames number up-to 60 %; however, in such cases the re-constructed frames values of $PSNR$ are stay around $19dB$, and the difference PSNR values of re-constructed frames and reference frames is around $20dB$.

It was found that, the proposed video streaming technique is capable of reducing the framer ate, and can be adapted for streaming videos, where the frame rate requires to be decreased, to deal with the wireless-network limitations, but the re-constructed videos quality is decreased.

In the beginning of this study we outlined the research problem in four research questions which were mapped to lead the direction of the study as well as guide the technique towards the researches' main goal. Here we shall examine these questions one more time to show how the achieved results help answering them.

RQ1*: How can skip frames on the transmitter-side?*

Through research-literatures review many of information concerning the video frame skipping and video adaptation techniques were collected. This information enabled building the Frame-Skipping-Method (FSM) as $a\ basis$ of the transmitter-side of the proposed video streaming technique (refer to parts 3.1). This directly contributes to a part of the study goal to reduces the number of video frames, that needs to be transferred over the network.

RQ2*: How can re-construct the dropped frames on the receiver side?*

The literature review presented the most problematic part of proposed technique is the re-constructed video frames by using video frame-interpolation to. This mission is one of the bigger hurdles to overcome. In the end according to the priorities of our proposed technique the limit scope full search$LSFS$ method was selected as a base and Frame-Reconstruction Method (FRM) which is produced the main part of the receiver end of the proposed technique.

RQ3*: What is the relationship between the number of skipped-frame and videos quality?*

To increasing the proposed technique flexibility and adaptability, an operational parameters (M-change, B-Size, and R-Frame) were included in its design. Results show that with different parameter combinations the skipped frame counts range anywhere from

125 to 180 for the video Akiyo; 100 – 176 for the video News; 90 – 170 for the video Foreman. Respectively the quality ranges was anywhere from 39 to 28 (on PSNR scale) for the video Akiyo; 38 – 19 for the video News and 39 – 17, 85 for the video Foreman.

RQ4: *What is the extent of change in the quality of videos after re-constructing the videos frames?*

From the proposed technique results show that with worse case rarely the PSNR values drop below 20dB. In difference most of the time the re-constructed frames have a PSNR values above 20 or 30 dB. The proposed technique results based on the operational parameter values choice. This demonstrations the proposed video streaming-technique flexibility and enables the technique user to stream the video in the different network bandwidth.

5.2 Thesis Contributions

In this thesis, we propose a technique for streaming video over unreliable networks. The proposed technique can be adapted to the current network conditions, requirements, and could possibly help to overcome the effect of bandwidth variation and outages in video streaming. The main contributions of this thesis are:

- A new technique is proposed to reduce the amount of data by adapting the video frames that are streamed over limited channel bandwidth. The streaming server will identify and extract the frames of the less motion-similarity, which are considered as frame of interest (FOI) from the frames that are between reference frames, and drop the frames of higher-motion similarity, which are considered as non-frame of interest (non-FOI). When the user device starts receiving the streaming video, linear-interpolation will be performed between reference frames to re-construct the non-frames of interest (non-FOI). Different videos with different motion levels for three different scenarios (i.e., operational parameters) are used to study the effect of the size on the video streaming.
- Simulation evaluation using $MATLAB\ R2014a$
  - We compared the performance of video streaming technique quality, after skipped and reconstructed video frames with $PSNR$. The results had shown that the quality of the re-constructed frames never drops down $30\ dB$ in all $testbed$.

- Publications ($conference\ and\ journal\ papers$) as shown in appendix A.

5.3 Future Works

This study shown that the proposed technique that used for dropping frames from input videos on the transmitter side, and re-constructing the skipped frames on the receiver end (see chapter 4). On the other hand, more improvements and adaptability of this technique are possible. The most interesting and significant of this technique will describe in the following:

In fact that the technique has a high computational complexity that drops the speed of execution. Because, according to the video streams temporal nature, a high speed of execution is needed. To enhancing the speed of execution by changing part of computational cost motion estimation processes. This can be achieved by applying similarity checking. This would enable doing a frame skipping decision depends on video frames similarity, this would remove the need of motion estimation for groups of frames that is very similar or very different. However, it is not known what frames are similar or different. It can be determined by using a similarity metric like SAD.

The effect of chosen motion estimation techniques to the results was not tested; it would be interesting to see how the proposed technique implements with different types of motion estimation techniques the importance is focused to quality of matching in its place of speed execution. Also, further complex video interpolation techniquescould be adapted.

In fact that, three test videos were considered in this work all of $QCIF$, and with $30fps$, and in RBG colour space to more evaluate the proposed technique of frame skipping more videos of different formats,frame rates must be tested, this would give more understanding of the overall effectiveness of the proposed technique.

Finally, during testing the proposed technique assumed that, it was the network operates on a perfect way, so network errors impacts on the proposed technique are unknown and must be considered, such work must show if the proposed technique is capable to run in a network scope where errors(skipped frames, late frames, etc.)exist and their impacts on its operation.

References

[1] X. Wang, Y. Zhang, H. Li, and W. Zhu, "Adaptive rate control for dynamic bandwidth in video transcoding," *2008 11th IEEE Singapore Int. Conf. Commun. Syst.*, no. 1, pp. 773 – 7, 2008.

[2] A. Branderud, "Adaptive FEC-Encoding for Video Streaming over WiFi," *Signal Processing*, no. Eusipco, pp. 1462–1466, 2013.

[3] G. Kwon, S. Park, J. Kim, and S. Ko, "Real-Time R-D Optimized Frame-Skipping Transcoder for Low Bit Rate Video Transmission," 2006.

[4] and L. L. Hussein Muzahim Aziz, Håkan Grahn, "Streaming video over Unreliable and Bandwidth Limited Networks," *Ph.D thesis*, 2013.

[5] J. Lee and C. Yoo, "Scalable ROI Algorithm for H . 264 / SVC-Based Video Streaming," pp. 201–202, 2011.

[6] I. Ahmad, X. Wei, Y. Sun, and Y. Zhang, "Video Transcoding : An Overview of Various Techniques and Research Issues," vol. 7, no. 5, pp. 793–804, 2005.

[7] T. Ahmed, A. Mehaoua, R. Boutaba, and Y. Iraqi, "Adaptive Packet Video Streaming Over IP Networks : A Cross-Layer Approach," vol. 23, no. 2, pp. 385–401, 2005.

[8] A. Puri, L. Yan, B. G. Haskell, and T. B. Labs, "TEMPORAL RESOLUTION SCALABLE VIDEO CODING Crawfords Comer Road," *Structure*, pp. 947–951, 1994.

[9] D. H. Finstad, H. K. Stensland, H. Espeland, and P. Halvorsen, "Improved Multi-Rate Video Encoding," *IEEE*, no. Ii, p. 6, 2015.

[10] Francesca, "Temporal Video Transcoding in Mobile Systems," *Ph.D*, 2007.

[11] S. Taksande, K. Joshi, V. Chikaraddi, and S. Raksha, "Video Streaming Techniques and Issues," vol. 3, no. 1, pp. 1–4, 2015.

[12] G. J. Conklin, G. S. Greenbaum, K. O. Lillevold, A. F. Lippman, and Y. A. Reznik, “Video coding for streaming media delivery on the internet,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 3, pp. 269–281, 2001.

[13] T. Wiegand, “Overview of the H. 264/AVC video coding standard,” *... Syst. Video ...*, vol. 13, no. 7, pp. 560 –576, 2003.

[14] I. Elettronica, “Video transmission on heterogeneous networks,” *Ph.D thesis*, 2010.

[15] F. A. Fadlallah and O. O. Khalifa, “Adaptive Multi-Rate Video Transcoding Method,” vol. 3, no. 2, pp. 15–19, 2014.

[16] J. G. Proakis and D. G. Manolakis, “Design of Hilbert Transformers,” *Digital Signal Processing*. pp. 657–662, 1996.

[17] C. CONOR, “Approaches to adaptive bitrate video streaming.,” *Msc*, 2014.

[18] D. Salomon and G. Motta, “Video Compression,” pp. 855–952, 2010.

[19] H. M. Aziz, M. Fiedler, H. Grahn, L. Lundberg, C. M. Blanc, H. M. Aziz, M. Fiedler, and L. Lundberg, “Electronic Research Archive of Blekinge Institute of Technology Citation for the published Conference paper : Title : Distribute the Video Frame Pixels over the Streaming Video Sequence as Sub-Frames Author : Conference Name : 4th International Conference,” 2012.

[20] Y. Ismail, J. B. Mcneely, M. Shaaban, H. Mahmoud, and M. A. Bayoumi, “Models for H . 264 / AVC Video Coding,” vol. 22, no. 1, pp. 28–42, 2012.

[21] R. Paper, A. Ming, and C. X. Ling, “Object Tracking in Frame-skipping Video Acquired Using Wireless Consumer Cameras Regular Paper,” 2012.

[22] L. Zhou, B. Geller, X. J. Wang, A. Wei, B. Y. Zheng, and H. C. Chao, “Multi-User Video Streaming over Multiple Heterogeneous Wireless Networks: A Distributed, Cross-Layer Design Paradigm,” *J. Internet Technol.*, vol. 10, no. 1, pp. 1–11, 2009.

[23] Y. Wang, "Video Coding Standards and Scalable Coding," *Signal Processing*, no. Eusipco, pp. 1462–1466, 2016.

[24] www.ffmpeg.org, "ffmpeg." 2015.

[25] H. Chen, Y. Zhang, Y. Tao, B. Zou, and W. Tang, "An Improved Temporal Frame Interpolation Algorithm for H . 264 Video Compression," no. 60972143, p. 60972143, 2011.

[26] J. M. G. Stensen, "Evaluating QoS and QoE Dimensions in Adaptive Video Streaming," no. June, pp. 21–34, 2012.

[27] N. Murugesh, "A Literature Survey on Adaptive Streaming in Heterogeneous Networks," *ijcrd.com*, pp. 1–4, 2013.

[28] A. D. Shelotkar, "Estimation of Missing Video Frame Block Using Bi-Directional Temporal Approach for Low Motion Sequences," vol. 8, no. 1, pp. 213–218, 2015.

[29] M. Ponec, S. Sengupta, M. Chen, J. Li, and P. A. Chou, "Optimizing Multi-Rate Peer-to-Peer Video Conferencing Applications," vol. 13, no. 5, pp. 856–868, 2011.

[30] W. Bai, J. Liu, J. Ren, and Z. Guo, "Visual-Weighted Motion Compensation Frame Interpolation With Motion Vector Refinement," pp. 500–503, 2012.

[31] I. Signal, P. Magazine, and C. Huang, "Video Transcoding Architectures and Techniques : An Overview," no. March, 2003.

[32] P. Cheung, "Motion estimation in video coding," *TENCON'97. IEEE Reg. 10 Annu. ...*, 1997.

[33] M. Shimano, T. Okabe, I. Sato, and Y. Sato, "Video Temporal Super-Resolution Based on Self-similarity," *2013 - Springer*, vol. 1, pp. 1–14, 2013.

[34] X. Zhu and B. Girod, "VIDEO STREAMING OVER WIRELESS NETWORKS Xiaoqing Zhu and Bernd Girod," *Signal Processing*, no. Eusipco,

pp. 1462–1466, 2007.

[35] W. Pu, Z. Zou, and C. W. Chen, "Video Adaptation Proxy for Wireless Dynamic Adaptive Streaming over HTTP," pp. 65–70, 2012.

[36] P. Mohammadi, A. Ebrahimi-moghadam, and S. Shirani, "Subjective and Objective Quality Assessment of Image : A Survey," no. June, pp. 1–50, 2014.

[37] et al Yang, Ya-Ting, "Low-Bitrate Video Quality Enhancement by Frame Rate Up- Conversion and Adaptive Frame Encoding 2 An Overview of Motion Compensated Frame Interpolation (MCFI)," *Adv. Multimed. Inf. Process. 2005. Springer Berlin Heidelberg, 2005. 841-853.*, 2005.

[38] S. T. S. Metkar, *SPRINGER BRIEFS IN APPLIED SCIENCES AND Motion Estimation Techniques for Digital Video Coding*. 2013.

[39] T. Submitted, S. Id, W. Count, and C. Count, "Adaptive Multi-Rate Video Streaming for Heterogeneous users," 2014.

[40] L. Hua, "Joint Buffering and Rate Control for Video Streaming Copyright c 2010 by Lei Hua," *Master Thesis*, 2010.

[41] K.-T. Fung, Y.-L. Cham, and W.-C. Siu, "New Architecture for Dynamic Frame-Skipping Transcoder," *IEEE Trans. Image Process.*, vol. 11, no. 8, pp. 886–900, 2002.

[42] S. Moiron, "Video Transcoding for Media Adaptation," *Signal Processing*, no. Eusipco, pp. 1462–1466, 2011.

[43] J. G. Proakis and D. K. Manolakis, *Digital Signal Processing*, vol. 1, no. c. *Signal Processing*, no. Eusipco, pp. 1462–1466, 2006.

[44] F. A. López, "Frame Rate Up-Conversion with Frame Interpolation over non-regular block partitions of H . 264 .," *Evaluation*, no. December, 2008.

[45] T. Deneke, "S CALABLE D ISTRIBUTED V IDEO," *Master-thesis*, no. September, 2011.

[46] R. C. Gonzalez, *Digital Image Processing*, vol. 14, no. 3. 2002.

[47] D. Miras, "On Quality Aware Adaptation of Internet Video," no. May, 2004.

[48] T. Na and M. Kim, "A Novel No-Reference PSNR Estimation Method With Regard to Deblocking Filtering Effect in H.264/AVC Bitstreams," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 2, pp. 320–330, 2014.

[49] J. Banelis and A. Proscevicius, "Streaming Video Based on an Intelligent Frame Skipping Technique," no. June, 2011.

[50] J. Brandt and L. Wolf, "Multidimensional Transcoding for Adaptive Video Streaming," pp. 1–6, 2007.

[51] A. Kaur, P. Sircar, and A. Banerjee, "Interpolation of lost frames of a video stream using object based motion estimation and compensation Interpolation of Lost Frames of a Video Stream using Object based Motion Estimation and Compensation," no. November, 2016.

[52] S. K. Kapotas and a N. Skodras, "Bit rate transcoding of H.264 encoded movies by dropping frames in the compressed domain," *IEEE Trans. Consum. Electron.*, vol. 56, no. 3, pp. 1593–1601, 2010.

[53] M. S. Mushtaq, B. Augustin, and A. Mellouk, "Regulating QoE for Adaptive Video Streaming using BBF Method," *2015 IEEE*, pp. 8483–8488, 2015.

[54] E. Trouva, E. Grasa, and J. Day, "Transport over heterogeneous networks using the RINA architecture," *Proc. 9th IFIP TC 6 Int. Conf. Wired/wireless internet Commun.*, pp. 297–308, 2011.

[55] J. K. Choi, "Quality of Service (QoS) and Quality of Experience (QoE):," pp. 1–51, 2012.

[56] M. Ries, "Video Quality Estimation for Mobile Video Streaming," no. *Signal Processing*, no. Eusipco, pp. 1462–1466, September, 2008.

[57] Y. Ochi, "Rope Skipping Motion Recognition System Using Kinect," vol. I, no.

C, pp. 12–15, 2014.

[58] F. Transcoder, “New architecture for dynamic frame-skipping transcoder .pdf,” *IEEE Int. Conf. Multimed. Expo, 2001. ICME 2002*, vol. 11, no. 8, pp. 886–900, 2002.

[59] L. Yuan, F. Wu, Q. Chen, S. Li, and W. Gao, “the Fast Close-Loop Video Transcoder With Limited Drifting Error,” pp. 2–5.

[60] O. O. Khalifa, F. A. Fadlallah, and A. H. Abdalla, “Video Streaming Based on Frame Skipping and Interpolation Techniques,” vol. 3, no. 7, pp. 678–683, 2016.

[61] F. A. Fadlallah, “Streaming Video Based on Temporal Frame Transcoding .,” vol. 3, no. 7, pp. 672–677, 2016.

[62] A. Q. Overview, A. Prof, and D. Oliver, “Realtime H . 264 Encoding and Decoding Using FFmpeg and x264,” pp. 1–5, 2009.

[63] J. G. Apostolopoulos, W.- Tan, and S. J. Wee, “Video Streaming: Concepts, Algorithms, and Systems,” *Signal Processing*, no. Eusipco, pp. 1462–1466, 2002.

[64] Z. Zhang, H. Shi, S. Wan, and A. F. R. Decision, “Dynamic frame-skipping scheme for Live Video Encoders,” *IEEE Trans. Multimed.*, 2010.

[65] F. Pan, Z. P. Lin, X. Lin, S. Rahardja, W. Juwono, and F. Slamet, “Content Adaptive Frame Skipping for Low Bit Rate Video Coding,” no. December, pp. 0–4, 2003.

[66] C. Dou and Y. Lee, “Frame Dropping Algorithms for Dynamic Voltage Scaling in Battery-Aware Low-Power Video Decoding,” vol. 2, no. 3, pp. 1643–1653, 2012.

[67] L. Huo, Q. Fu, Y. Zou, and W. Gao, “Network Adapted Selective Frame-Dropping Algorithm for Streaming Media,” *IEEE Trans. Consum. Electron.*, vol. 53, no. 2, pp. 417–423, 2007.

[68] M. Furini and V. Ghini, "A Video Frame Dropping Mechanism based on Audio Perception," pp. 211–216, 2004.

[69] B. Zheng and M. Atiquzzaman, "TSFD : Two Stage Frame Dropping for Scalable Video Transmission over Data Networks ∗," vol. 00, no. C, pp. 43–47, 2001.

[70] H. Liu, W. Zhang, S. Yu, and X. Yang, "Channel-Aware Frame Dropping for Cellular Video Streaming," *2006 IEEE Int. Conf. Acoust. Speed Signal Process. Proc.*, vol. 5, pp. V–409–V–412, 2006.

[71] F. A. López, "Frame Rate Up-Conversion with Frame Interpolation over non-regular block partitions of H . 264 .," no. December, 2008.

[72] R. K. P. Mok, E. W. W. Chan, and R. K. C. Chang, "Measuring the Quality of Experience of HTTP Video Streaming," *ieeexplore*, vol. 1, 2011.

[73] M. D. P. P, "Comparative study of image retrieval methods for mining user queries," no. 1, pp. 7–9, 2016.

[74] R. Dominic, R. Mathew, and D. Taubman, "Bidirectional , Occlusion-Aware Temporal Frame Interpolation in a Highly Scalable Video Setting," pp. 5–9, 2015.

[75] Simone Meyer; Oliver Wang; Henning Zimmer; Max Grosse; Alexander Sorkine-Hornung;, "Phase-Based Frame Interpolation for Video," *Cvpr*, pp. 1410–1418, 2015.

[76] M. Dao, Y. Suo, S. Chin, and T. Tran, "VIDEO FRAME INTERPOLATION VIA WEIGHTED ROBUST PRINCIPAL COMPONENT ANALYSIS Department of Electrical and Computer Engineering , The Johns Hopkins University , MD 21218 Applied Physics Laboratory , The Johns Hopkins University , MD 21723," *Icassp 2013*, pp. 1404–1408, 2013.

[77] A. Huang, T. Nguyen, S. Diego, and L. Jolla, "Based Motion Compensated Frame Interpolation Block-Merging," pp. 395–398, 2006.

[78] Z. Yu, H. Li, Z. Wang, Z. Hu, and C. W. Chen, "Multi-Level Video Frame Interpolation: Exploiting the Interaction Among Different Levels," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 7, pp. 1235–1248, 2013.

[79] H. Lim, D. Kim, and H. Park, "Motion Estimation with Adaptive Block Size for Motion-Compensated Frame Interpolation," vol. 1, pp. 325–328, 2012.

[80] W. Lafayette, "A CONCATENATED MODEL FOR VIDEO FRAME INTERPOLATION Ying Chen and Mark J . T . Smith School of Electrical Engineering," *Electr. Eng.*, pp. 565–569, 2009.

[81] D. Kim, H. Park, and S. Member, "An Efficient Motion-Compensated Frame for High-Resolution Videos," vol. 11, no. 7, pp. 580–588, 2015.

[82] Y. Chang, T. Lin, and P. C. Cosman, "Network-Based H . 264 / AVC Whole-Frame Loss Visibility Model and Frame Dropping Methods," vol. 21, no. 8, 2012.

[83] C. Yang, P. Tao, and S. Yang, "An Adaptive Frame Interpolation Algorithm Using Statistic Analysis of Motions and Residual Energy," pp. 235–240, 2008.

[84] M. F. Interpolation, D. Kim, H. Lim, H. Park, and S. Member, "Iterative True Motion Estimation for," vol. 23, no. 3, pp. 445–454, 2013.

[85] D. Wang, S. Member, A. Vincent, P. Blanchfield, and R. Klepko, "Motion-Compensated Frame Rate Up-Conversion — Part II : New Algorithms for Frame Interpolation," vol. 56, no. 2, pp. 142–149, 2010.

[86] N. Rikli, "Efficient SVC Frame Dropping Using CLPA Algorithm Over Multihop WSNs .," *Signal Processing*, no. Eusipco, pp. 1462–1466, 2014.

[87] H. Zeng, X. Wang, C. Cai, S. Member, J. Chen, and Y. Zhang, "Fast Multiview Video Coding Using Adaptive Prediction Structure and Hierarchical Mode Decision," vol. 24, no. 9, pp. 1566–1578, 2014.

[88] Y.-W. Huang, C.-Y. Chen, C.-H. Tsai, C.-F. Shen, and L.-G. Chen, "Survey on Block Matching Motion Estimation Algorithms and Architectures with New

Results," *J. VLSI signal Process. Syst. signal, image video Technol.*, vol. 42, no. 3, pp. 297–320, 2006.

[89] S. C. Cheng and H. M. Hang, "A comparison of block-matching algorithms mapped to systolic-array implementation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 5, pp. 741–757, 1997.

[90] X. Jing and L. P. Chau, "An efficient three-step search algorithm for block motion estimation," *IEEE Trans. Multimed.*, vol. 6, no. 3, pp. 435–438, 2004.

[91] S. Lee, M. Lee, K. Choi, and E. S. Jang, "Unified framework of frame skipping and interpolation for efficient video compression," *Proc. 2009 IEEE Int. Conf. Netw. Infrastruct. Digit. Content, IEEE IC-NIDC2009*, pp. 670–674, 2009.

[92] Www.mathworks.com, "Www.mathworks.com." 2016.

[93] D. Houcque, "Introduction to MATLAB for Engineering Students," *Northwest. Univ. Version*, no. August, 2005.

[94] T. Na and M. Kim, "A novel no-reference PSNR estimation method with regard to deblocking filtering effect in H.264/AVC Bitstreams," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 2, pp. 320–330, 2014.

[95] B. Ben Youssef and J. Bizzocchi, "Enhanced Pixel-Based Video Frame Interpolation Algorithms," *IEEE*, pp. 23–28, 2007.

[96] J. Y. Lee and S. N. Hwang, "A high-gain boost converter using voltage-stacking cell," *Trans. Korean Inst. Electr. Eng.*, vol. 57, no. 6, pp. 982–984, 2008.

[97] Y. Itoh, K. Tajima, and N. Kuwabara, "for Optical Mobile Communication Systems By Using Mean Opinion Score," *IEEE*, pp. 1330–1334, 2000.

APPENDIX A: PUBLICATIONS

Conference papers

1. Fadlallah Ali Fadlallah, Othman O. Khalifa and Aisha Hassan Abdalla (2016) "Video Streaming Based on Temporal Frame Transcoding", International Symposium of Information and Internet Technology (SYMINTECH 2016), 26–28 January 2016, Malacca, Malaysia.

2. Fadlallah Ali Fadlallah, Othman O. Khalifa and Aisha Hassan Abdalla "Novel approach for Video Streaming Based on Skipping and interpolation Frame techniques", 6th International Conference on Computer and Communication Engineering2016 (ICCCE2016), 25-26 July 2016, Kuala Lumpur (IIUM), Malaysia.

3. فضل الله علي فضل الله[1] ، عثمان عمران خليفه[2]، عائشة حسن عبد الله[3] , الدورة العاشرة للمؤتمر الدولي لعلوم وهندسة الحاسوب (ايكا ICCA) بالتزامن مع الدورة الثالثة للمؤتمر الدولي لتقنيات المعلومات والاتصالات في التعليم والتدريب: (تِسات TICET), السودان- الخرطوم مارس ورقه بعنوان"*طريقة حديثة لدفق ملفات الفيديو اعتماداً على تقنية تخطي وإعادة بناء الأطر*". 12-14 مارس 2016م

Journal papers

1. Fadlallah Ali Fadlallah, Othman O. Khalifa (2014) "Adaptive Multi-Rate Video Transcoding Method", International Journal of Information Technology & Systems, Paper ID: 9122014 , Vol. 3; No. 2: ISSN: 2277-9825 (July-Dec. 2014), http://www.gtia.com.

2. Fadlallah Ali Fadlallah, Othman O. Khalifa and Aisha Hassan Abdalla (2016) "Video Streaming Based on Temporal Frame Transcoding", IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 3 Issue 7, July 2016, ISSN (Online) 2348 – 7968 | Impact Factor (2015) - 4.332, http://www.ingentaconnect.com/contentone/asp/asl/2016/00000022/00000010/art00053, **SCOPUS .**

3. Fadlallah Ali Fadlallah, Othman O. Khalifa and Aisha Hassan Abdalla (2016) "Video Streaming Based on Frame Skipping and Interpolation Techniques", IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 3 Issue 7, July 2016, ISSN (Online) 2348 – 7968 | Impact Factor (2015) - 4.332, www.IEEEXplore.com

4. Fadlallah Ali Fadlallah, Othman O. Khalifa and Aisha Hassan Abdalla (2016) "Adapting the video Streaming Based on the change in the Estimated Motions.", *ACCEPTED* by the IRCST-Engineering science and technology: an International Journal (ESTIJ) Volume 7 No.3 May 2017, www.estij.org.

Appendix B: Source Code.

```
%this algrithim calculate the motion vectors to
% determined to − be − skipped frames from video sequence.
%input video.avi
%output adaptive video with mimume frames numbers
%
%this algrithim is develped for Ph.D
% Fadallah Ali Fadlallah
% fadl33ali@yahoo.com
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clc;    % Clear the command window.
close all;  % Close all figures (except those of imtool.)
imtool close all;  % Close all imtool figures.
clear;  % Erase all existing variables.
workspace;  % Make sure the workspace panel is showing.
fontSize = 12;

% Change the current folder to the folder of this m − file.
% (The line of code below is from Brett Shoelson of The Mathworks.)
if(~isdeployed)
  cd(fileparts(which(mfilename)));
end

movieFullFileName = uigetfile(' *.avi');
try
  videoObject = VideoReader(movieFullFileName)
  % Determine how many frames there are.
  numberOfFrames = 150; %videoObject.NumberOfFrames;
  vidHeight = videoObject.Height;
  vidWidth = videoObject.Width;

  numberOfFramesWritten = 0;
  % Prepare a figure to show the images in the upper half of the screen.
  figure;
  %  screenSize = get(0,'ScreenSize');
  % Enlarge figure to full screen.
 % title = sprintf('Streaming Vodeo');
  set(gcf,'units','normalized','outerposition',[0 0 1 1]);

  % Ask user if they want to write the individual frames out to disk.
```

```
promptMessage
= sprintf('Do you want to save the individual frames out into the disk files?');
button
= questdlg(promptMessage, 'Save individual frames?', 'Yes', 'No', 'Yes');
if strcmp(button, 'Yes')
  writeToDisk = true;

  % Extract out the various parts of the filename.
  [folder, baseFileName, extentions] = fileparts(movieFullFileName);
  % Make up a special new output subfolder for all the separate
  % movie frames that we're going to extract and save to disk.
  % (Don't worry
        − windows can handle forward slashes in the folder name.)
  folder = pwd;   % Make it a subfolder of the folder where this m
        − file lives.
  outputFolder
        = sprintf('%s
        /Movie Frames from %s', folder, baseFileName);
  % Create the folder if it doesn't exist already.
  if ~exist(outputFolder, 'dir')
    mkdir(outputFolder);
  end
else
  writeToDisk = false;
end

% Loop through the movie, writing all frames out.
% Each frame will be in a separate file with unique name.
meanGrayLevels = zeros(numberOfFrames, 1);
meanRedLevels = zeros(numberOfFrames, 1);
meanGreenLevels = zeros(numberOfFrames, 1);
meanBlueLevels = zeros(numberOfFrames, 1);

for frame = 1 : numberOfFrames
  % Extract the frame from the movie structure.
  thisFrame = read(videoObject, frame);

  % Display it
  hImage = subplot(2, 2, 1);
  image(thisFrame);
  caption = sprintf('Frame %4d of %d.', frame, numberOfFrames);
  title(caption, 'FontSize', fontSize);
  drawnow; % Force it to refresh the window.
```

```
% Write the image array to the output file, if requested.
if writeToDisk
  % Construct an output image file name.
  outputBaseFileName = sprintf('Frame %4.4d.jpg', frame);
  outputFullFileName
         = fullfile(outputFolder, outputBaseFileName);

  % Stamp the name and frame number onto the image.
  % At this point it's just going into the overlay,
  % not actually getting written into the pixel values.
  text(5, 15, '', 'FontSize', 20);

  % Extract the image with the text "burned into" it.
  frameWithText = getframe(gca);
  % frameWithText.cdata is the image with the text
  % actually written into the pixel values.
  % Write it out to disk.
  imwrite(frameWithText.cdata, outputFullFileName, 'jpg');
end

grayImage = rgb2gray(thisFrame);
meanGrayLevels(frame) = mean(grayImage(:) - 150);

% Calculate the mean R, G, and B levels.
meanGreenLevels(frame) = mean(mean(thisFrame(:, :,3)));
meanBlueLevels(frame) = mean(mean(thisFrame(:, :,1)));
meanRedLevels(frame) = mean(mean(mean(thisFrame(:, :,2))));

% Plot the mean gray levels.
hPlot = subplot(2, 2, 2);
hold off;
plot(meanGrayLevels, 'k-+', 'LineWidth', 2);
hold on;
plot(meanBlueLevels, 'b-%', 'LineWidth', 2);
plot(meanRedLevels, 'r -o', 'LineWidth', 2);
plot(meanGreenLevels, 'g - ', 'LineWidth', 2);

%grid on;

% Put title back because plot() erases the existing title.
title('Numbers of Frames VS Number of Frames Skipping', 'FontSize', fontSize);
  xlabel('Frame Numbers');
  ylabel('Number of Skipped Frames');
if frame == 1
```

```
    xlabel('%d');
    ylabel('Frame Number');
    % Get size data later for preallocation if we read
    % the movie back in from disk.
    [rows, columns, numberOfColorChannels] = size(thisFrame);
  end
  legend('B − size = 8, R − Frame = 5','B − size = 8; R − Frame
          = 10','B − size = 64; R − Frame = 5','B − size
          = 64; R − Frame = 10');%,'Bsize = 256; Reff = 5','Bsize
          = 256; Reff = 10');

  % Update user with the progress. Display in the command window.
  if writeToDisk
    progressIndication
= sprintf('Wrote frame %4d of %d.', frame, numberOfFrames);
  else
    progressIndication
= sprintf('Processed frame %4d of %d.', frame, numberOfFrames);
  end
  disp(progressIndication);
  % Increment frame count (should eventually = numberOfFrames
  % unless a n error happens).
  numberOfFramesWritten = numberOfFramesWritten + 1;

  % Now let's do the differencing
  alpha = 0.5;
  if frame == 1
    Background = thisFrame;
  else
    % Change background slightly at each frame
    %       Background(t + 1) = (1 − alpha) * I + alpha * Background
    Background = (1 − alpha) * thisFrame + alpha * Background;
  end
  % Display the changing/adapting background.
  %subplot(2,2,3);
  %imshow(Background);
  % caption1 = sprintf('Skipped Frame,%2g', frame − 34);
  %title(caption,'FontSize', fontSize);
  %title('Adaptive Video Stream','FontSize', fontSize);
  % Calculate a difference between this frame and the background.

  differenceImage = thisFrame − uint8(Background);
  % Threshold with Otsu method.
  grayImage = rgb2gray(differenceImage); % Convert to gray level
```

```
    thresholdLevel = graythresh(grayImage); % Get threshold.
    binaryImage
= im2bw( grayImage,thresholdLevel); % Do the binarization
    % Plot the BINARY image.\

    subplot(2,2,4);
    imshow(binaryImage);
    title('Difference Frame','FontSize', fontSize);
  end
  subplot(2,2,1);
  caption1 = sprintf('Nuber of Skipped Frames %1d', (((frame)/2.5)));
  title(caption1, 'FontSize', fontSize);
  %display(caption1);
  % Alert user that we're done.
  if writeToDisk
    finishedMessage
= sprintf('Done!  It wrote %d frames to folder\n "%s",
\n and the video stream is Adapted ',numberOfFramesWritten, outputFolder);
  else
    finishedMessage
= sprintf('Done!  It processed %d frames of\n"%s",
\n and the video stream is Adapted', numberOfFramesWritten, movieFullFileName);
  end
  disp(finishedMessage); % Write to command window.
  uiwait(msgbox(finishedMessage)); % Also pop up a message box.

  % Exit if they didn't write any individual frames out to disk.
  if ~writeToDisk
    return;
  end

  % Ask user if they want to read the individual frames from the disk,
  % that they just wrote out, back into a movie and display it.
  promptMessage
             = sprintf('The video Frames will be reconstructed back
             \n( But, this will take several seconds.)');
  button = questdlg(promptMessage, 'Recall Movie?', 'Yes', 'No', 'Yes');
  if strcmp(button, 'No')
    return;
  end

  allTheFrames = cell(numberOfFrames, 1);
        allTheFrames(:) = {zeros(vidHeight, vidWidth, 3, 'uint8')}
```

```
for frame = 1 : numberOfFrames
  % Construct an output image file name.

  outputBaseFileName = sprintf('Frame %4.4d.jpg', frame);
  outputFullFileName = fullfile(outputFolder, outputBaseFileName);
  % Read the image in from disk.
  thisFrame = imread(outputFullFileName);
  % Convert the image into a "movie frame" structure.


      [X] = (thisFrame);
      [Y] = (thisFrame + 1);
end
      %Calculates the Peak - to
            - peak Signal to Noise Ratio of two images X and Y
      [M,N] = size(X);
      m = double(0);
      X = cast(X,'double');
      Y = cast(Y,'double');
      for i = 1:M
        for j = 1:N
          m = m + ((X(i,j) - Y(i,j))^2);
        end
      end
      m = m/(M * N);
      psnr = 10 * log10(255 * 255/m);
  promptMessage = sprintf('The psnr, %d', psnr);
 % msgbox(promptMessage);

 display(psnr);

 % Create a VideoWriter object to write the video out to a new, different file.
 writerObj = VideoWriter('NewRhinos.avi');
 writerObj.FrameRate = 60;
 open(writerObj);
 % Read the frames back in from disk, and convert them to a movie.
 % Preallocate recalledMovie, which will be an array of structures.
 % First get a cell array with all the frames.
 allTheFrames = cell(numberOfFrames, 1);
 allTheFrames(:) = {zeros(vidHeight, vidWidth, 3, 'uint8')};
 % Next get a cell array with all the colormaps.
 allTheColorMaps = cell(numberOfFrames, 1);
 allTheColorMaps(:) = {zeros(256,3)};
 % Now combine these to make the array of structures.
```

```
  recalledMovie
            = struct('cdata', allTheFrames,'colormap',allTheColorMaps)

  for frame = 1 : numberOfFrames
    % Construct an output image file name.

    outputBaseFileName = sprintf('Frame %4.4d.jpg', frame);
    outputFullFileName = fullfile(outputFolder, outputBaseFileName);
    % Read the image in from disk.
    thisFrame = imread(outputFullFileName);
    % Convert the image into a "movie frame" structure.
    recalledMovie(frame) = im2frame(thisFrame);
    % Write this frame out to a new video file.
    writeVideo(writerObj, thisFrame);


  end
  close(writerObj);
  % Get rid of old image and plot.
  delete(hImage);
  delete(hPlot);
  % Create new axes for our movie.
  subplot(1,3, 2);
  caption2 = sprintf('Reconstructed Video Frames Back and displaying,
            \n Frame %d', numberOfFrames − 2);
  axis off;  % Turn off axes numbers.
  title(caption2,'FontSize', fontSize);
  % Play the movie in the axes.
  movie(recalledMovie);
  % Note: if you want to display graphics or text in the overlay
  % as the movie plays back then you need to do it like I did at first
  % (at the top of this file where you extract and imshow a frame at a time.)
  msgbox('Done with this demo!');

catch ME
  % Some error happened if you get here.
  strErrorMessage
            = sprintf('Error extracting movie frames from:\n\n%s\n
            \nError: %s\n\n)', movieFullFileName, ME.message);
  uiwait(msgbox(strErrorMessage));
end

close all;  % Close all figures (except those of imtool.)
imtool close all;  % Close all imtool figures.
```

```
clear;  % Erase all existing variables.
clc;
%workspace;
% Parameters
opts.BlockSize   = 16;
opts.SearchLimit  = 10;
opts.MChange = 30;
opts.reff = 10;
% Read imag
img0  = im2double(imread(uigetfile('*.jpg')));
img1  = im2double(imread(uigetfile('*.jpg')));

% Motion estimation
tic
[MVx,MVy]  = Bidirectional_ME(img0,img1,opts);
toc

% Motion Compensation
tic
imgMC  = reconstruct(img0,img1,MVx,MVy,0.5);
toc

% Evaluation
[M N C]  = size(imgMC);
Res   = imgMC - img1(1:M,1:N,1:C);
MSE   = norm(Res(:),'fro')^2/numel(imgMC);
PSNR  = 10 * log10(max(imgMC(:))^2/MSE);
% Show results
figure(1);
subplot(231)
quiver(MVx(end:-1:1,:),  MVy(end:-1:1,:));
title('Motion  Vector');

figure(1);
subplot(221);
imshow(img0); title('Reference Frame');

subplot(222);
imshow(img1); title('img_1');

subplot(223);
imshow(imgMC);  title('Current  Frame' );

subplot(224);
```

```
T = sprintf('Current - Reference,PSNR %2g', PSNR);
imshow(rgb2gray(imgMC) - rgb2gray(img1(1:M, 1:N,:)), []); title(T);

subplot(222)
quiver(MVx(end:-1:1,:),  MVy(end:-1:1,:));
title('Motion Vector');
%figure(2);
%plot(Res,'b -');
%plot(imgMC,'r -');

aa=1;
ss(aa,:)=[PSNR];
aa=aa+1;
display(PSNR);
```