Master Thesis
Software Engineering
Thesis no: MSE-2011:48
June 2011

Streaming Video Based on an Intelligent Frame Skipping Technique

Justas Banelis, Arunas Proscevicius

School of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona
Sweden

This thesis is submitted to the School of Engineering at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Software Engineering. The thesis is equivalent to 20 weeks of full time studies.

Contact Information:
Author(s):
Justas Banelis
Address: Taikos 95 – 75, Vilnius, Lithuania
E-mail: justas.banelis@gmail.com

Arunas Proscevicius
Address: Alyvu g. 1a, Uzubaliu k., Avizieniu sen., Vilniaus r., Lithuania
E-mail: arunas.proscevicius@gmail.com

University advisor(s):
Hussein Aziz
School of Computing

School of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona
Sweden

Internet : www.bth.se/com
Phone : +46 455 38 50 00
Fax : +46 455 38 50 57

ABSTRACT

Video streaming is an important field of global communications and data processing. It is divided into server and client sides connected via network. Video streaming is concerned with delivering video data from server to client over the network as fast and with as little loss as possible.

In this study the possibilities to minimize the amount of data transferred over the network in video streaming are investigated and a video streaming technique comprised of server and client sides is proposed.

To expand the flexibility and adaptability of the proposed video streaming technique an operational parameter system was constructed and the parameter value ranges were defined. The proposed video streaming technique was then applied to three sample videos. Before streaming the server side of the proposed technique reduced the frame count of input videos based on operational parameter values while the client side reconstructed the skipped frames. Then the quality of the resulting videos was measured and evaluated. To evaluate the reconstructed frames and videos the PSNR measurement method was used.

The study concludes that by using the proposed video streaming technique it is possible to reduce the amount of transfer data by dropping frames on the server side and reconstructing them on the client side.

Keywords: video streaming, frame skipping, frame interpolation.

Contents

1 INTRODUCTION

The steady development of IT technologies has made Internet an important part of global communications. The amount of data sent and received through the Internet rises daily along with the number of users. This in turn sparked the need for various WEB services and technologies to accommodate the needs of the users. One of the most popular and problematic WEB services is video streaming. Video streaming requires a lot of bandwidth, furthermore any delays or inconsistencies during data transfer result in reduced video quality. These limitations are even more pronounced when streaming videos over wireless networks [1]. This is because wireless networks are usually slower, interference from household devices (i.e. microwave ovens, cordless phones), radio transmitters and other network nodes affect the quality of network service [8]. This makes it difficult to sustain a steady flow of information and deliver packets by a deadline [2]. However with the popularity of mobile devices (smart phones, pocket PC's etc.) the demand for video and audio streaming over the wireless networks rises. To cope with these issues the network bandwidth of the video streaming service must be managed efficiently by adjusting the transmission rate to the capability of the wireless channel [3].

The purpose of this study is to present a video streaming technique which addresses the above mentioned limitations. The proposed technique consists of server side and client side. The server side will be responsible for dividing the video into different segments with considerable movement in them. The frame – skipping algorithm will be invoked which will drop some video frames from the video stream. The client side will be responsible for reconstructing the dropped frames from available data using frame – interpolation algorithm. The video will then be reconstructed and played on the user device.

This study is divided into seven chapters. The first chapter introduces the subject matter to the reader and presents the general outline of the work. The second chapter presents the literature study of the field. It summarizes works related to this study and points out their limitations. The third chapter states aims and objectives of the study. The fourth chapter presents the proposed video streaming technique and explains it's inner workings. The fifth chapter presents the results of the tests preformed with the proposed technique and explains their significance. The sixth chapter presents the conclusions of the work. The seventh chapter presents possible directions for future work.

2 LITERATURE STUDY

To identify the ways to counteract the shortcomings of video streaming a literature study was carried out. The literature study was focused to find the works of other researchers that used video frame skipping/dropping on the server side and reconstructed video frames on the client side by using interpolation. As such digital libraries of ACM and IEEE were searched for any studies relating to the subjects of video frame skipping/dropping, video adaptation and video frame interpolation. After refining search options, removing double entries and eliminating works dated before 2000 the titles and abstracts of the resulting works were read to identify works for further review and to find the works which contributed to this study. These works were used to familiarize ourselves with the most prominent problems of video streaming and act as a foundation upon which we built our video streaming technique.

2.1 Video Frame Skipping/Dropping

According to the subject matter the works on video frame skipping/dropping and video adaptation can be divided into two parts. The overwhelming majority of the works are concentrated on the subject of video transcoding. Nowadays video encoders and decoders are complicated and powerful pieces of software, they have a multitude of features and can accomplish many tasks. However transcoding is centered on the fact that in default settings video encoders do not operate as well as they could, examples include: transcoding of the videos over wireless networks [4], video transcoding using different encoder parameters ([5] – [6]) etc. Although all these works are different, they all are based on the use of the video encoder/decoder, furthermore they show that by manipulating various video encoder parameters, different goals can be achieved like increasing video quality or reducing bandwidth consumption. Although video transcoding is fast and efficient a lot of factors need to be considered for it to be applied: platform, encoder itself, operating environment, backwards compatibility etc. Furthermore the fundamental flaw of video transcoding is that it operates almost exclusively on GoP (Group of Pictures) coding structure [4] – [6]. While this gives more options to influence video streaming process it also ties the transcoding technique to the video encoder and the GoP coding structure. Moreover, the performance of the video transcoding techniques is largely dependent on network traffic intensity and is very hard to evaluate due to its random nature.

One of the examples of video transcoding techniques is proposed by Patil et al. [3]. They propose a method for reconstructing lost motion vectors from vector maps of adjacent frames. They state that the reconstruction of the motion vectors can be done in one of three ways: bilinear interpolation, Activity Dominant Vector Selection and Forward Dominant Vector Selection. The third method to recover the lost motion vector is by estimating the motion vector based on the vector map of the frame which has the biggest overlap with the lost frame. Although this work is focused on recovering the vector maps of the lost frames (and by extension the lost frame itself) it can also be adapted to selectively or randomly drop video frames from the video stream and for frame recovery.

Besides video transcoding other research work on the subject of video frame skipping/dropping cannot be categorized under one group. The conclusion and shortcomings of the more interesting works that are related to this field of study are presented below.

Huo et al. [2], developed a novel selective frame dropping algorithm adaptive to network bandwidth. Their work is based on GoP (group of pictures) coding structure. The algorithm itself offers dynamic frame – dropping policy creation for each GoP based on transfer results of previous GoP. Based on the experimental results it shows that the algorithm offers better real – time performance while maintaining comparatively lower computational complexity. However the results also show that the algorithm is focused on reducing the network bandwidth while the video quality is of secondary importance. Furthermore the importance of frames in different GoP is not considered (that is the difference in importance of different GoP is not addressed). Also frame recovery/reconstruction is not even considered as a way to improve the quality of transferred videos. Finally the contents of the video do not influence frame prioritization and video segmentation.

Zheng et al. [7] propose frame dropping scheme to cope with the limitations of video streaming over wireless networks. Their work is also based on GoP coding structure dividing the videos into different GoP and assigning a priority to each frame. If the network channel is free then an entire GoP is transmitted and if the channel is congested then frames are dropped according to priority. Different frames are assigned different priorities the content of the video does not influence the decision making, leading to a possible situation where important frames are considered to be less important. Furthermore behavior of the server is still defined by the condition of the network channel and frame dropping occurs only when the network channel is congested.

Zhu et al. [8] propose a video frame skipping technique based on the volume of differences between consecutive video frames. They state that video encoders/decoders are sophisticated enough to recreate video frames if the differences between them are small enough. Then an arbitrary function is introduced that judges whether the changes are small or not, based on the judgment the video frames in the stream are passed to the encoder with notes to assign them different priorities. This method offers better video stream structure than the one provided by the video encoders/decoders. However it does not propose anything to improve video frame recreation, furthermore the proposed function evaluates video frames based on the statistical differences between them and does not take into account the video contents.

Liu et al. [1] propose a frame dropping scheme in a mobile wireless network environment. This work is based on dividing the video stream in different GoP, however here the different picture groups are formed on the radio – link layer by encapsulating each frame into SDU (Service Data Unit) and sorting frames in the picture groups by priority. They propose selective frame retransmission based on frame priorities in the picture groups. This allows to shift the losses to frames of lesser importance. The proposed scheme does not offer a way to lessen the burden of the wireless network channel but only distributes the lost frames to parts of picture groups with less priority.

This concludes the summary of the more prominent works in the field of video frame skipping/dropping and video transcoding. It must be said that other proposed work exists in the field of video frame skipping/dropping like [9] – [10]. However the proposed methods offered in those works do not contribute to the aim of this study and thus are not considered. On the other hand the main shortcoming of the works we considered is that they structure the video stream based on mathematical modeling and not video content. Our proposed video streaming technique aims to structure the videos into different segments with considerable motion while dropping the less important frames according to the frame similarity. This will reduce the amount of frames that need to be sent over the network, while after receiving the video the client side will reconstruct the frames dropped in the server side and return the video to the original frame number which will be played on the client screen.

2.2 Video Frame Interpolation

Video frame interpolation is an important and very large subfield of video streaming. Video frame interpolation encompasses the tools and techniques used to rebuild the missing frames based on surrounding frames. A lot of research has been done in this field and some of them are presented below.

Asefi [11] describes the algorithms and methods used in video frame interpolation. He considers the mathematical background of each method as well as computational complexity. Furthermore the work proposes different video interpolation technique by categorizing videos according to motion estimation patterns. This work provides a useful reference list of the motion estimation methods and essentially any motion estimation algorithm described in the work Asefi [11] proposed can be used to achieve the aims of this thesis work.

In some cases algorithms are combined with other calculation and ordering methods to produce an even better result. For example, to recreate a missing picture between two input pictures Mahajan et al. [12] propose a method based on moving pixel gradients and Poisson reconstruction. They state that the proposed method addresses many of the common occlusion effects in image interpolation and allows the recreation of complex non – rigid movements.

Hiraiwa et al. [13] propose a method to automatically extract an object from video stream. This work is aimed to improve object – based browsing and content – based searching systems. However in this work also presents several tools that can be used in video frame interpolation field. The skip – labeling algorithm for feature based segmentation, occlusion – killer algorithm for accurately estimating optical flow, and shrink – merge algorithm for tracking an object. The works [12] and [13] present interesting alternatives to the motion estimation algorithm that we developed and they hold the possibility of being superior to our choice.

3 Aims and Objectives

The purpose of this thesis is to develop a method to drop/skip similar frames from video streaming sequence to reduce the amount of frames that need to be transferred over the network. Following objectives have to be met to fulfill the main aim:

1. Develop an algorithm for motion detection;
2. Implement a proof of concept prototype of the proposed video streaming technique;
3. Measure the quality of the videos by using PSNR testing method;

3.1 Research Questions

The research questions that need to be addressed in the thesis:

- How to skip frames on the sever side?
- How to reconstruct skipped frames on the client side?
- What is the relation between skipped frame count and quality of the videos?
- How does the quality of the videos change after reconstructing the video frames?

4 The Proposed Video Streaming Technique

To address the limitations of video streaming over wireless networks a video streaming technique is proposed, the technique is based on intelligent video frame skipping mechanism (IFSM) on the server side and video frame reconstruction mechanism (FRM) on the client side, as shown in Figure 1.

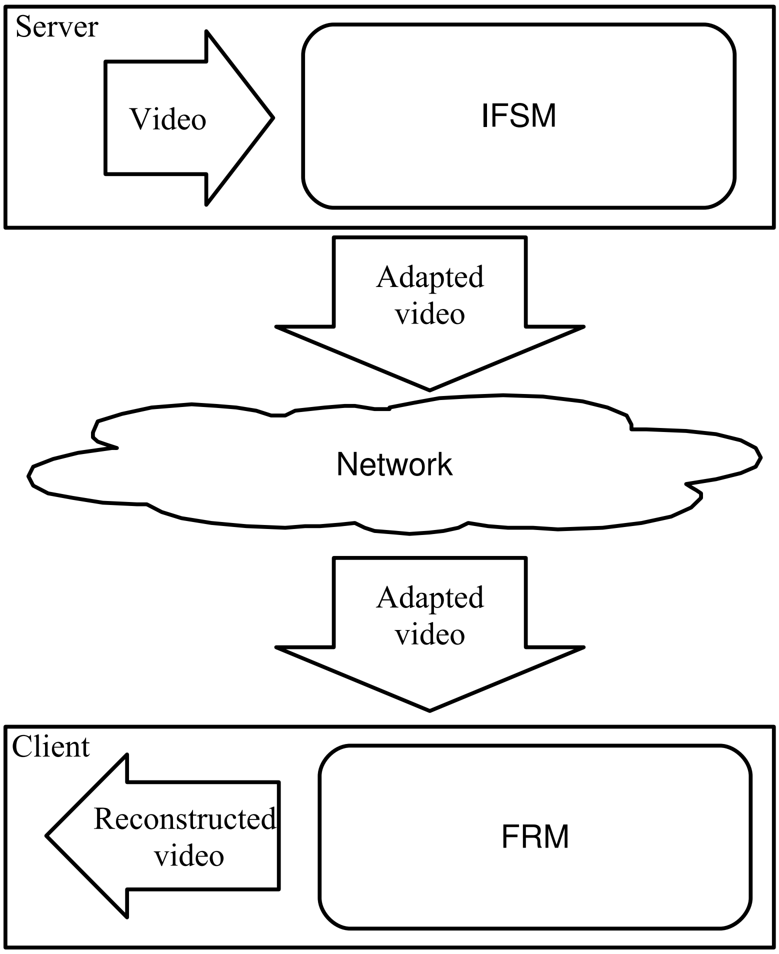

Figure 1: The Proposed Video Streaming Technique

The two main parts of the proposed technique are video streaming server and the client receiving the video. The proposed technique operates in the both, the server side and the client side:

Server side:

1. The Intelligent Frame Skipping Mechanism (IFSM) is applied to the video stream.
2. Based on estimated motions between frames IFSM 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.

Client side:

1. Client receives the adapted video stream.
2. Frame Reconstruction Mechanism (FRM) is applied to the received video frames to reconstruct the frames that were skipped on the server side.
3. Reconstructed video frames are played according to their sequence position number.

4.1 The Intelligent Frame Skipping Mechanism

IFSM is proposed to operate on the server side and it's purpose is to skip frames according to the change in the estimated motion giving the possibility to reconstruct the skipped frames on the client side using video interpolation. IFSM processes a video stream as input and returns a video stream that is lower or equal in frame count compared to the initial video stream (1).

$$COUNT(VS) \geq COUNT(IFSM(VS)) \tag{1}$$

In (1) VS is a video stream and COUNT is a number of frames in a given video stream.

IFSM skips video frames in the following way:

1. Determines motion between video frames using motion estimation.
2. Determines change in motion between video frames and skips video frames.

Motion estimation is used to identify similar frames which will be skipped on the server side. Frame skipping is described in full detail further in the coming section.

Motion Estimation

IFSM is based on a notion that it is possible to estimate motion between video frames. The motion estimation techniques are considered based on the execution speed and quality of matching. Since there is a trade off between speed of execution and quality of matching, emphasis is given to speed of execution, this decision is made due to the temporal nature of video streaming.

According to Huang et al. [14] and Cheng et al. [15] the fastest block matching algorithm is Three Step Search (TSS).

Two possible variations of TSS where implemented, which will be referred as TSS-Type1 by Jing et al. [16] and TSS-Type2 by Koga et al. [17]. Both variations use sum of absolute differences (SAD) as a difference metric to match the blocks in frames in the streaming video:

$$SAD = \sum_{n=1}^{c} \sum_{i=1}^{h} \sum_{j=1}^{w} \left| \left(f(i,j,n) - f'(i,j,n) \right) \right| \tag{2}$$

In (2) the color channel count c, the height of block in pixels h, the width of block in pixels w, f and f' are intensity functions for pixels at position i, j and color channel n.

The iterations of TSS-Type1 as shown in Figure 2 are as follows:

1. Step one:
    - Calculates SAD of initial block and stores a reference value.
    - Calculates SAD for the surrounding blocks.
    - If any of the surrounding blocks has lower SAD value chooses the block with the lowest value for the next iteration.
    - If all values are higher then the reference value stops the algorithm assuming that the best matching block was found.
2. Step two:
    - Stores the lowest SAD value of last iteration as a reference value.
    - Sets the block with the lowest SAD value as a center for next search.
    - Calculates SAD for the surrounding blocks
    - If any of the surrounding blocks has lower SAD value chooses the block with the lowest value for the next iteration.
    - If all values are higher then the reference value stops the algorithm assuming that the best matching block was found.
3. Step three is identical to the second step.

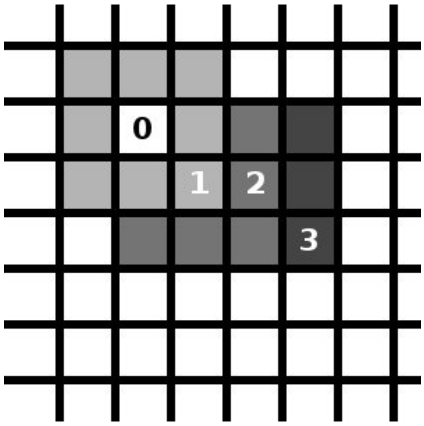

Figure 2: TSS-Type1

0 - 1st iteration matching center(matching area in lightest gray)
1- lowest SAD value for 1st iteration matching center for 2nd iteration(matching area in medium gray)
2 - lowest SAD value for 2ndt iteration matching center for 3rd iteration(matching area in darkest gray)
3 - best match

The iterations for the second type of TSS-Type2 as shown in Figure 3 are as follows:

1. SAD values of blocks at a distance of step size from the center are calculated.
2. Step size is reduced to half, the center is moved to the block with the lowest SAD value.

3. Steps 1 and 2 are repeated until step size is less than 1.

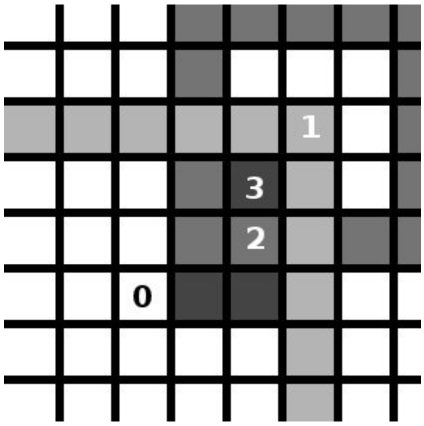

Figure 3: TSS-Type2

0 - 1^{st} iteration matching center(matching area in lightest gray)
1- lowest SAD value for 1^{st} iteration matching center for 2^{nd} iteration(matching area in medium gray)
2 - lowest SAD value for 2^{ndt} iteration matching center for 3^{rd} iteration(matching area in darkest gray)
3 - best match

Frame Skipping

The IFSM will skip frames by determining the changes in the motion between frames. IFSM uses the following data structures:

1. Vector, a structure that holds vectors beginning and end coordinates.
2. An array of vector attributes, an array of structures that hold vector speed, direction and end coordinates.
3. Vector field, a structure that holds frame dimensions, block dimensions and an array of vector structures.

To determine the changes in the motion IFSM operates in the following steps:

1. Using TSS function a vector field is created for the reference frame (f_n) and the next immediate frame (f_{n+1}). An array of vector attributes is populated.
2. Using TSS function a vector field is created for the frame (f_{n+1}) and the next immediate frame (f_{n+2}). The newly created 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.
3. If changed vector counter does not exceed the limit set by parameters the frame is marked for skipping and the IFSM goes to step 2. If changed vector counter does exceed the limit the frames that where compared last are marked for keeping, IFSM continues at stage 1.

4.2 Frame Reconstruction Mechanism

When the client receives the adapted video stream, the skipped frames need to be reconstructed. To reconstruct the skipped frames a frame reconstruction mechanism (FRM) was implemented, which takes as input an adapted video stream with reduced frame count and produces a video stream equal in frame count as the initial video stream (3). To reconstruct the skipped frames, video frame interpolation based on motion estimation is used.

$$COUNT(VS) = COUNT(FRM(VS')) \qquad (3)$$

In (3) *VS* is video stream before adaptation, *VS'* is an adapted video stream received by the client, *COUNT* is a function which returns frame count and *FRM* is frame reconstruction mechanism.

Motion Estimation

The requirements for the motion estimation technique are the same as for motion estimation technique used in IFSM: high performance and good matching quality, with emphasis on high performance. Due to the limited search scope of TSS it was decided to use a different motion estimation technique. The best search coverage is provided by full search technique, which matches all possible combinations, but such a technique is highly computationally expensive, thus does not meet the requirements. To reduce the computations done by full search technique, scope of matching was limited. Leaving us with Limited Scope Full Search (LSFS) as shown in Figure 4. The input of LSFS is two frames and the output of LSFS is a vector field that represents motion of blocks of pixels between frames. LSFS operates in the following way:

1. SAD value of the center block is calculated and saved as the lowest value.
2. Offset is increased by one SAD values for all blocks at current offset are calculated if a lower SAD value than current lowest is found it is saved.
3. Step two is repeated until offset is less than the maximum offset.

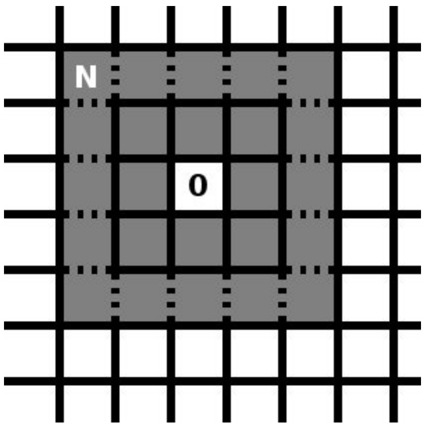

Figure 4: Limited scope full search

0 - matching center(matching area in lightest gray)
N- maximum offset from the matching center

Interpolation

After block positions have been determined in the non-skipped frames positions of blocks in the skipped frames needs to be calculated. To calculate block positions in the missing frames linear interpolation is used. Linear interpolation is a method used to calculate unknown points between two known points (4).

$$\frac{y-y_0}{x-x_0}=\frac{y_1-y_0}{x_1-x_0} \qquad (4)$$

In (4) (x_0, y_0) and (x_1, y_1) are two known points and (x, y) is a point that is calculated. In (4) it is assumed that either x or y is known which in our case is not true. Based on (4) a function that calculates x (5) and a function that calculates y (6) where created. In (5) and (6) the following assumptions are made x and y travel along a straight line between (x_0, y_0) and (x_1, y_1), and x and y have a constant speed between (x_0, y_0) and (x_1, y_1).

$$x=x_0+fN*\left(\frac{x_1-x_0}{nM+1}\right) \qquad (5)$$

In (5) x is the location of a block on the x axis in the frame being interpolated, nM is the amount of frames missing between two frames, fN is the number of the frame being interpolated, x_0 and x_1 are the points between which x moves.

$$y=y_0+fN*\left(\frac{y_1-y_0}{nM+1}\right) \qquad (6)$$

In (6) y is the location of a block on the y axis in the frame being interpolated, nM is the amount of frames missing between two frames, fN is the number of the frame being interpolated, y_0 and y_1 are the points between which y moves.

Frame Reconstruction

To reconstruct the missing 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 missing frames.

To further enhance FRM cross-fading of intensity values was introduced (7).

$$CFI=f(x_1)*\left(1-\left(\frac{fN}{nM+1}\right)\right)+f(x_2)*\left(\frac{fN}{nM+1}\right) \qquad (7)$$

In (7) CFI is a cross-faded intensity value, $f(x_1)$ is intensity value at x_1, $f(x_2)$ is intensity value at x_2, fN is the frame currently being reconstructed, nM is number of frames missing between two frames that are present.

4.3 Tools Used In The Intelligent Video Frame skipping technique

The set of video processing libraries that are used in this work are FFmpeg[1].

For selecting a video library the following criteria were set:

1. It should be open source and with a C language API.
2. It should work on Linux.
3. It should be documented.

FFmpeg was chosen to fit our criteria. The chosen version is FFmpeg 0.6.

FFmpeg is licensed under LGPL[2] 2.1, thus it's free and open source software which can be used in all free or non-free software development.

FFmpeg is written in C, thus it has a C language API and it is developed for gcc[3], therefore it compiles and runs on Linux platform.

FFmpeg also include a codec library, able to encode/decode majority of popular video codecs. Even though using other codecs falls out of the scope of this thesis it adds a possibility to test the proposed technique with variety of different codecs.

---

[1]FFmpeg - the leading audio/video codec library, available at http://ffmpeg.org
[2]LGPL - GNU Lesser General Public License, available at http://www.gnu.org/licenses/old-licenses/lgpl-2.1.html
[3]GCC - the GNU Compiler Collection, available at http://gcc.gnu.org/

5 RESULTS

To investigate the capabilities of our technique a case study was conducted. Three video clips were chosen as subjects for this: they are Akiyo; News; Foreman. All three videos are of QCIF (176x144) resolution and 300 frames each, but with different characteristics. Akiyo video mainly shows facial motion. News video shows facial, body motion as well as background motion and background scene change. Foreman video shows facial and body motion as well as camera and background motion. Furthermore the body motion in Foreman video is large, at times taking place over half of video frame. To accurately measure and calculate the results of the case study we will assume that the network (refer to Figure 2.) operates without frame loss and the adapted video stream will reach the client side intact.

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

1. Invoke the IFSM on the input video. This adapts the input video by dropping its frames based on the values of operational parameters;
2. Invoke the FRM on the adapted video for reconstructing the missing frames from the adapted video streaming sequence;
3. Compute the measures of video quality:
    - PSNR value of each frame of the reconstructed video;
    - The average PSNR value of the reconstructed video;
    - The average PSNR value of frame that were dropped;
    - The minimum PSNR value of frames that were dropped;

Such a run through was preformed on each of the video clips with wide range of operational parameter values. The operational parameters that influence the results are:

- Changedmax – maximum amount of changed motion vectors allowed. This parameter is used in IFSM to to decide whether to keep or drop the frame from the video stream. This decision is made based on the amount of motion that is different between two frames. The higher the value of this parameter the more motion it will be in skipped frames and the harder it will be for FRM to reconstruct the frame. Since the value of chagedmax is dependent on the size of the pixel block, we will use its percentile representation. So changedmax = 20 means that frames with motion that is less that 20 % different from the last reference frame will be dropped while others will be kept and streamed.
- Refframe – add reference frame every n frames. This parameter shows the minimum number of frames that will be kept from the original video. Its value range is from 2 to the number of frames of the video. Refframe = 2 means that every other frame will be kept no matter what.
- Blsize – the size of pixel block. This parameter shows the size of pixel blocks that were used in IFSM and FRM. Its value range is from 4 to the resolution of the source video.

The results of the case study were evaluated by measuring different attributes of reconstructed videos using PSNR method. This information was collected and used to judge the quality of the video clips as well as the effectiveness of the proposed video streaming technique.

This chapter will present the results of the case study and explain their significance. It will also briefly explain the relevance of each parameter and their influence on each of the test video clips. Finally alternative methods to conclude the case study will also be discussed.

5.1 Significance Of The Parameters

The properties of the operational parameters used in the case study are shown in Table 1. It was said in the definition of operational parameters that a wider range of operational parameter values can be used. However the value ranges of the parameters that are shown in Table 1 were deduced from initial run throughs of the case study and represent the configuration of the prototype of the proposed video streaming technique which produces best results. In particular if the value of the parameter chagedmax is lower than 20 then it has virtually no effect at all. On the other hand if changedmax is more than 60 then the quality of reconstructed videos is drastically reduced and in some cases the reconstructed frames are incomprehensible. That is why an interval from 20 to 60 was chosen as a value range for this parameter. Also from the initial run throughs of the case study it was deduced that refframe values higher than 10 result in increasingly poor quality as well as frozen frames, that is why 2 and 10 were chosen as border values of this operational parameter with 5 as an intermediate value. It was the hardest to define the value range for the parameter blsize. No generally accepted standards exist especially for an atypical format such as QCIF. However during the literature study it was noticed that pixel blocks of 64 and 256 were considered in other works with QCIF format [8]. Another border value of 4 was added to expand the possibilities of the proposed video streaming technique. This proved to be the right choice since with other parameters being equal the smallest pixel block produces slightly better results in almost all cases.

Figure 5 shows how the proposed video streaming technique handles video clip Akiyo with all possible parameter combinations. It can be seen from Figure 5 that the number of skipped frames (in each respective parameter combination category) hits maximum value when the value of parameter changedmax reaches 30 (approximately). The maximum number of skipped frames is as follows:

1. With Refframe = 2, the maximum number of skipped frames is 149;
2. With Refframe = 5, the maximum number of skipped frames is 239;
3. With Refframe = 10, the maximum number of skipped frames is 269.

Table 1: Significance of the parameters

Name of the parameter[4]	**Value range**	**Relation to quality**	**Impact on quality**
Changemax	From 20 to 60	Inversly proportional	High
Refframe	2, 5, 10	Inversly proportional	High
Blsize	4, 64, 256	No clear relationship[5]	High

[4]Refer to chapter 5 for specifics on each parameter.
[5]This holds true only with the specified parameter value range.

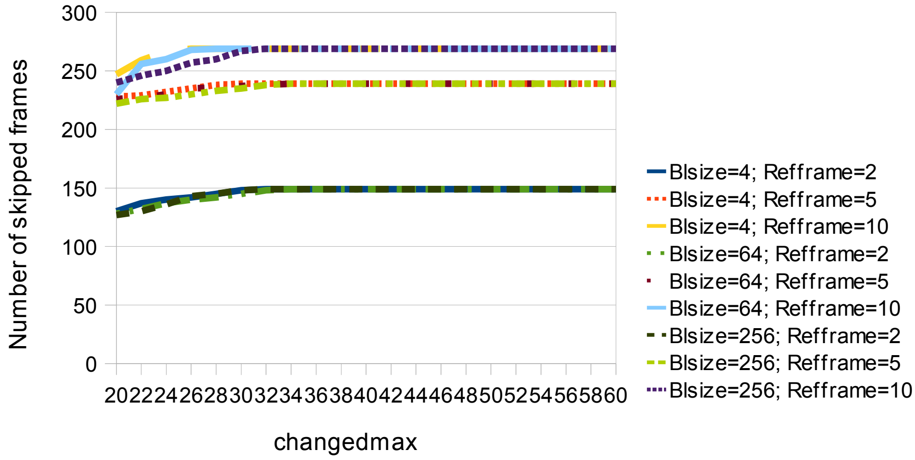

Figure 5: The relation between operational parameters and the number of dropped frames in Akiyo video clip.

The maximum number of skipped frames is reached so easily because the video clip Akiyo depicts mainly facial motion and is relatively easy to interpolate (compared to other two videos).

An argument can be made that our proposed technique does not explore all possibilities regarding this particular video clip. Furthermore it can be argued that if the maximum number of skipped frames is reached so early then even more frames can be dropped. In regards to the former statement it must be considered that the set of results of the case study depicted in Figure 5 is indeed atypical. However it is impossible to define a universal set of operational parameter values which explore all possibilities of the three test video clips. This is because the characteristics of motion shown in Akiyo, News and Foreman videos is very different (see section 5). That is why the value ranges of operational parameters were chosen such that only the best results of all three test videos would be present. In regards to latter statement it must be said that it is already hard to recreate a video with half of it's 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 5 (as well as Figure 7 and Figure 9) does not represent the quality of recreated videos in any way. We illustrate this with Figure 6 which shows the same frame that was skipped and reconstructed by using different parameters.

a:

Blsize=4
Refframe=2
changedmax=30

b:

Blsize=64
Refframe=2
changedmax=30

c:

Blsize=256
Refframe=2
changedmax=30

d:

Blsize=4
Refframe=5
changedmax=30

e:

Blsize=64
Refframe=5
changedmax=30

f:

Blsize=256
Refframe=5
changedmax=30

g:

Blsize=4
Refframe=10
changedmax=30

h:

Blsize=64
Refframe=10
changedmax=30

i:

Blsize=256
Refframe=10
changedmax=30

Figure 6: Snapshots of frame number 17 from Akiyo video which was reconstructed using different parameters.

Figure 6 shows that the increase in the parameter refframe (which in turn increases the maximum number of skipped frames) leads to increasingly worse picture quality. In fact the parameter refframe has the biggest impact on the quality of reconstructed videos. This was expected since leaving more of the original video makes frame interpolation easier. It is also shown that increasing refframe to 5 the drop in quality is barely noticeable. Increasing refframe to 10 however distorts the image, although it is still comprehend able. The parameter blsize has no noticeable effect in these samples since it is more important in handling videos with more movement in them.

Figure 7 shows the results of applying the proposed video streaming technique on video clip News with all possible operational parameter combinations. It is shown that that like with Akiyo video, eventually the maximum number of skipped frames is reached with all parameter combinations (the maximum number of skipped frames is the same for all three video clips because they all have 300 frames each). However with News video this generally requires larger values of changedmax. Overall the number of skipped frames is far more diverse than in the case of Akiyo video. In particular with smaller pixel block sizes the number of skipped frames ranges from 100 to 269. The reason behind this is because News video has a lot of small movements and with small pixel block size the IFSM has a smaller margin of error when selecting frames to skip. On the other hand with bigger pixel blocks small movements do not constitute enough changes to keep the frame. It will be shown in section 5.2 that the quality of reconstructed frames is better when using the smallest pixel block. However the decrease in quality is marginal.

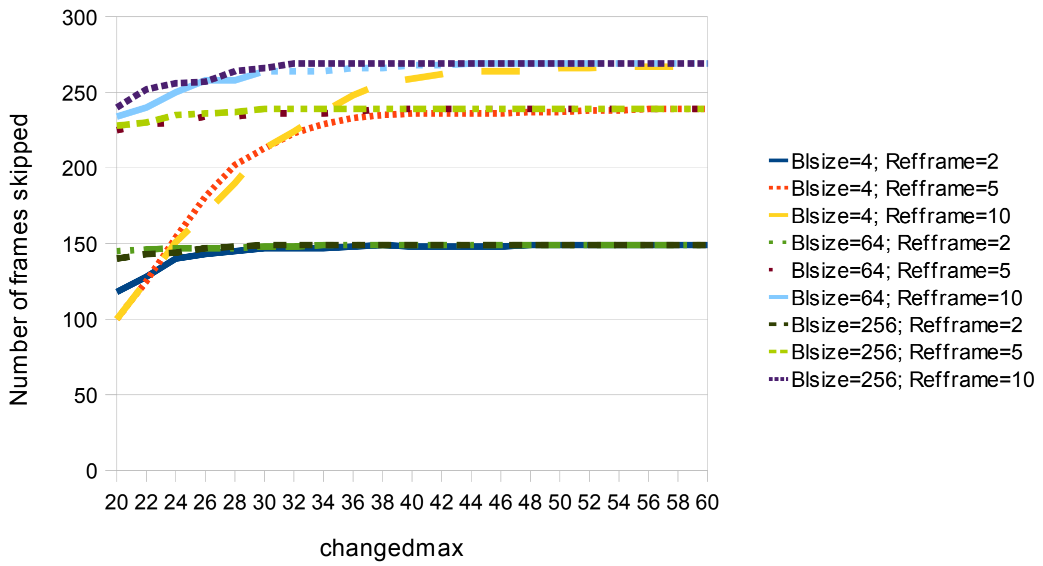

Figure 7: The relation between operational parameters and the number of dropped frames in News video clip

Figure 8 shows a frame of News video which was skipped and reconstructed using different parameter values.

a:

Blsize=4
Refframe=2
changedmax=30

b:

Blsize=64
Refframe=2
changedmax=30

c:

Blsize=256
Refframe=2
changedmax=30

d:

Bl*size=4*
*Reffra*me=5
changedmax=30

e:

Blsize=64
Refframe=5
changedmax=30

f:

Blsize=256
Refframe=5
changedmax=30

g:

Blsize=4
Refframe=10
changedmax=30

h:

Blsize=64
Refframe=10
changedmax=30

i:

Blsize=256
Refframe=10
changedmax=30

Figure 8: Snapshots of frame number 138 from News video which was reconstructed using different parameters.

From these sample frames it can be seen that the relationship between operational parameters and video quality is similar to Akiyo video. Better results are achieved with lower refframe value. Consider frame snapshots a, d and g in Figure 8. The frame gets more garbled as the parameter refframe increases from 2 to 5 and then to 10 with other parameters being equal. However the the most important point here is that bigger pixel blocks do not garble or eliminate small movements. Consider frame snapshots b and e. Take into account that according to the parameter values, as shown in Figure 7, more than 2/3 of the video was skipped in both cases. However, both the small movements (male newscaster face), and big movements (the ballerina) are recreated essentially the same as in frame snapshots a and d.

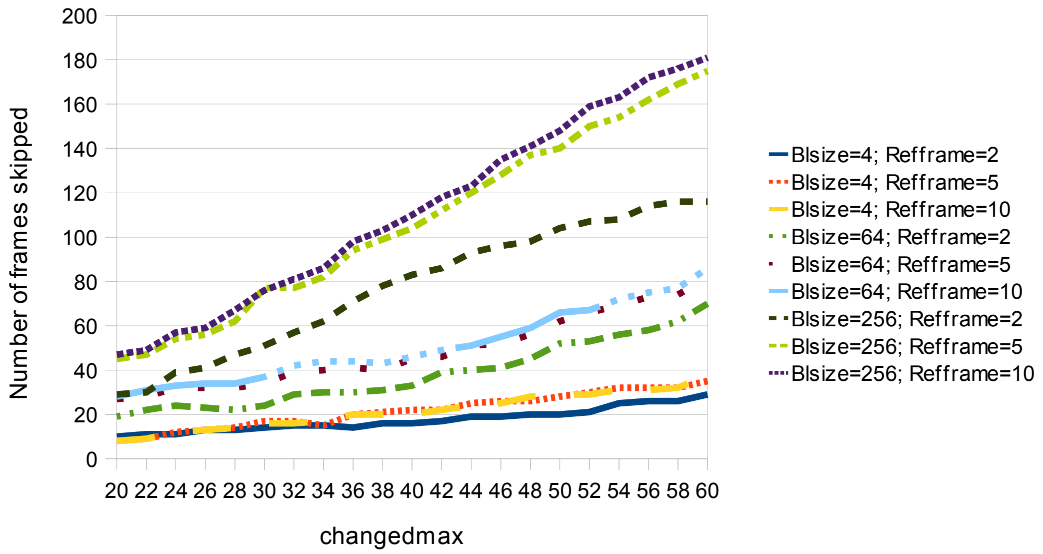

Figure 9: The relation between operational parameters and the number of dropped frames in Foreman video clip

Figure 9 shows how the proposed video streaming technique handles the video clip Foreman with all possible parameter combinations. Unlike Akiyo and News videos the number of skipped frames varies across the whole chart. Furthermore the maximum number of skipped frames is not reached with any parameter combination. This is because video clip Foreman is the most movement intensive among the three test videos. Furthermore the movements in Foreman are large. In some cases the motion constitutes more than half of video frame resolution. One would think that in this case the proposed video streaming technique would perform better with bigger pixel block sizes. It is in fact true, it is shown in Figure 9 that with larger values of blsize the IFSM allows more frames to be skipped. Particularly, consider the black line in Figure 9. With the parameter refframe = 2 the number of skipped frames is third largest out of all possibilities. It can be deduced that bigger pixel blocks handle large movements better and that the parameter refframe is less important in movement intensive videos. Like with Akiyo and News video clips a sample frame which was recreated with different parameter values is shown in Figure 10.

a:

Blsize=4
Refframe=2
changedmax=30

b:

Blsize=64
Refframe=2
changedmax=30

c:

Blsize=256
Refframe=2
changedmax=30

d:

Blsize=4
Refframe=5
changedmax=30

e:

Blsize=64
Refframe=5
changedmax=30

f:

Blsize=256
Refframe=5
changedmax=30

g:

Blsize=4
Refframe=10
changedmax=30

h:

Blsize=64
Refframe=10
changedmax=30

i:

Blsize=256
Refframe=10
changedmax=30

Figure 10: Snapshots of frame number 82 from Foreman video which was reconstructed using different parameters.

In Figure 10, frame snapshot i is garbled and the face of the man is incomprehensible. Apart from that notice that there is little to no change when increasing blsize (with refframe being the same). This once again shows that bigger pixel block sizes are more suited to handle movement intensive videos.

Changedmax

The parameter changedmax is the primary factor in IFSM. The value of this parameter is more important in handling videos with more movements in them. In such cases a wide rage of values can help achieve the desired result (weather to increase the number of skipped frames or the quality of the video). On the other hand in handling videos with little movement in them it is only important to find the value of changedmax with which the maximum number of skipped frames is achieved. In such cases increasing the value of changedmax will have no effect. As such choosing an universal value range for this parameter is quite impossible. One possible solution would be to choose the value of changedmax based on the average PSNR value of reconstructed video. However this requires that the proposed video streaming technique would be used with all possible changedmax values on the target video.

Refframe

The parameter refframe has the most impact on quality of the reconstructed videos. However a very low value of refframe prevents the proposed video streaming technique from reaching its full potential. From the sample frames of the three videos provided above it can be seen that increasing refframe value to 5 the loss of quality is acceptable. Selecting the values for refframe parameter is quite straightforward. Lower refframe values mean less frames will be skipped (in turn this increases the quality of reconstructed videos) with 2 being the lowest possible value. The maximum value of refframe is defined by the number of frames the video has. However increasing refframe past 10 makes video frame interpolation too complex since the video will have to be reconstructed from less than 90 % of it's original size.

Blsize

This parameter is a special case. The reason behind it is because motion detection and motion estimation methods used in this work were not intended for QCIF format videos. In fact motion estimation and motion detection perform worse with videos of small resolution [1] [8]. It can be argued that it was a mistake to choose such an unusual video format for testing purposes. To answer this it must be said that initially this work was aimed at mobile devices. However during the course of the work the focus shifted to build a theoretical base of video streaming technique which can handle all video resolutions. Another issue with the parameter blsize is that it has no clear relationship with video quality. All pixel block sizes that were considered produced best results on separate occasions. However no consistent patterns or dependencies were found. Although it must be noted that the smallest pixel block produced best results most of the time.

This concludes the analysis of operational parameters and how they influence results of the case study of the proposed video streaming technique. The next section will explain how quality is understood, quantified and evaluated in the scope of this work. Furthermore it will be explained what acceptable quality is and what is not.

5.2 Video Quality Evaluation

The notion of quality is an unquantifiable and subjective entity. However to properly evaluate the efficiency of the proposed video streaming technique a measurement of video quality of the reconstructed video frames is needed. In this work PSNR method is used to measure video quality.

5.2.1 Measurement Tools

The reconstructed videos will be evaluated using peak signal-to-noise ratio (PSNR). PSNR is a most commonly used objective video quality measure tool [18]. PSNR is a ratio between maximum signal power (MSP) and the power of interfering noise, represented by mean squared error (MSE):

$$PSNR = 10 * \log_{10}\left(\frac{MSP^2}{MSE}\right) \tag{8}$$

MSE is a representation of magnitude of average differences between pixels of reconstructed video and the original video:

$$MSE = \frac{\sum_{n=1}^{c}\sum_{i=1}^{h}\sum_{j=1}^{w}\left(f(i,j,n) - f^{'}(i,j,n)\right)^2}{h * w * c} \tag{9}$$

where the color channel count is c, the height of frame in pixels is h, the width of frame in pixels is w, f and f' are intensity functions for pixels at position i, j and color channel n.

MSP represents the maximum possible value of the pixel calculated by

$$MSP = (2^B - 1) \tag{10}$$

where B is number of bits per color channel.

PSNR is measured in decibels (dB) on a logarithmic scale with values from 0 to infinity, the higher value the better the quality of the video.

PSNR is an objective measurement method which is used to evaluate the digital video based on reference video.

Figure 11 shows sample frames from different videos which varying quality.

Frame 140

PSNR=30,9;

Frame 230

PSNR = 25;

Frame 6

PSNR = 29;

Frame 145

PSNR = 19,5;

Figure 11: Examples of frames with varying quality.

Frame 140 has the lowest PSNR value above 30 out of all the results. Similarly Frame 145 has the highest PSNR value below 20 (as a side note, frame snapshot i in Figure 10 has the PSNR value of 19,01 and it is the fifth worst result overall).

5.2.2 Video Frame Quality

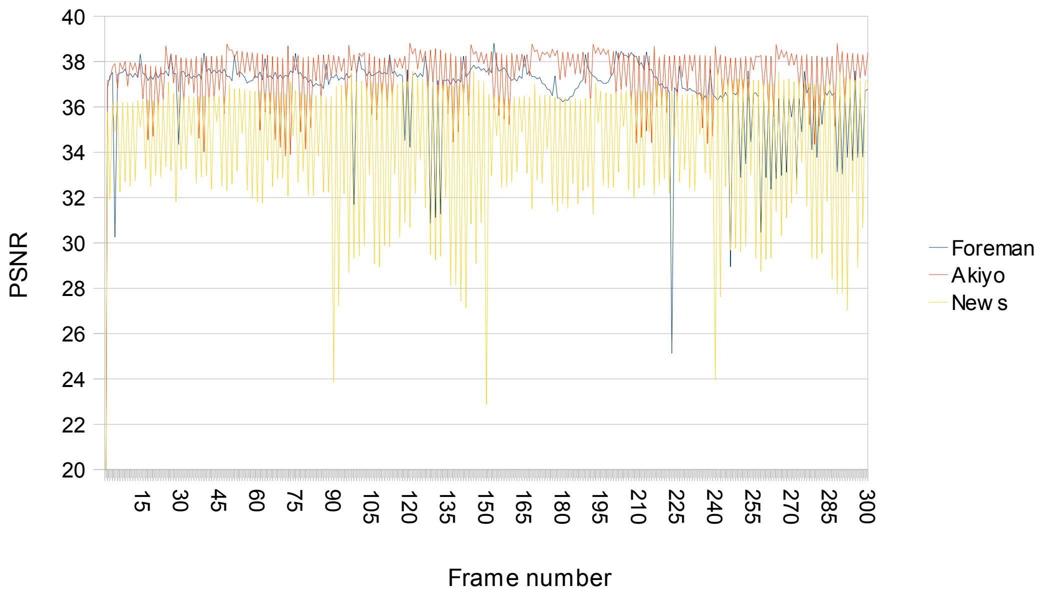

Figure 12: PSNR values of all video frames with parameters: refframe = 2; blsize = 4; changedmax = 30.

Figure 12 shows the PSNR values for all the video frames in all test videos with given parameter values. It is noted that this parameter value configuration is focused on video quality (refer to Table 1 for relation between parameter values and video frame quality). Parameters refframe and blsize have the lowest possible values and the value of the parameter changedmax is near its lower limit (its value range is from 20 to 60). It is noticed that the PSNR score never drops below 20. Furthermore, most of the frames score above 30 and only the News video has PSNR scores in the interval [20, 30] rather frequently. This happens because News is more motion intensive than other videos. In this example the frames of video Akiyo score above 30 all the time. Figure 13 shows Frame 150 and Frame 90 from the video News. These frames have lower quality than the rest of the frames shown in Figure 12.

Frame 150

Refframe=2
Blsize=4
changedmax=30
PSNR=22,87

Frame 90

Refframe=2
Blsize=4
changedmax=30
PSNR=23,87

Figure 13: Snapshots of frames with lowest PSNR score.

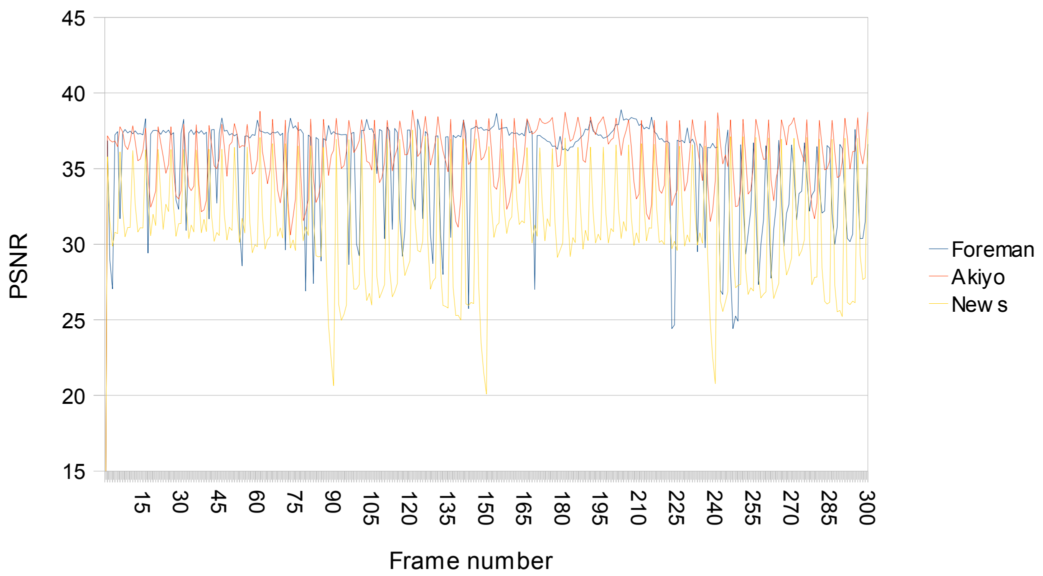

Figure 14: PSNR values of all video frames with parameters: refframe = 5; blsize = 64; changedmax = 30.

It was shown that the parameter refframe has the largest impact on the quality of the reconstructed videos. Figure 14 shows how the average quality drops after increasing refframe value to 5. 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. In this example the video Akiyo still has a PSNR score above 30 all the time, consider the fact that with these parameter values the frames were reconstructed from

1/5 th of original video size. In the case of the videos Foreman and News, the average quality drop is noticeable, however the PSNR score drops below 25 very occasionally and generally stays in the interval [25, 30]. Figure 15 shows the frames with the worst PSNR scores in this set of samples. Both of these frames include a background scene change which is one of the more difficult changes to reconstruct.

Frame 150

Refframe=5
Blsize=64
changedmax=30
PSNR=20,9

Frame 90

Refframe=5
Blsize=64
changedmax=30
PSNR=20,79

Figure 15: Snapshots of frames with lowest PSNR score.

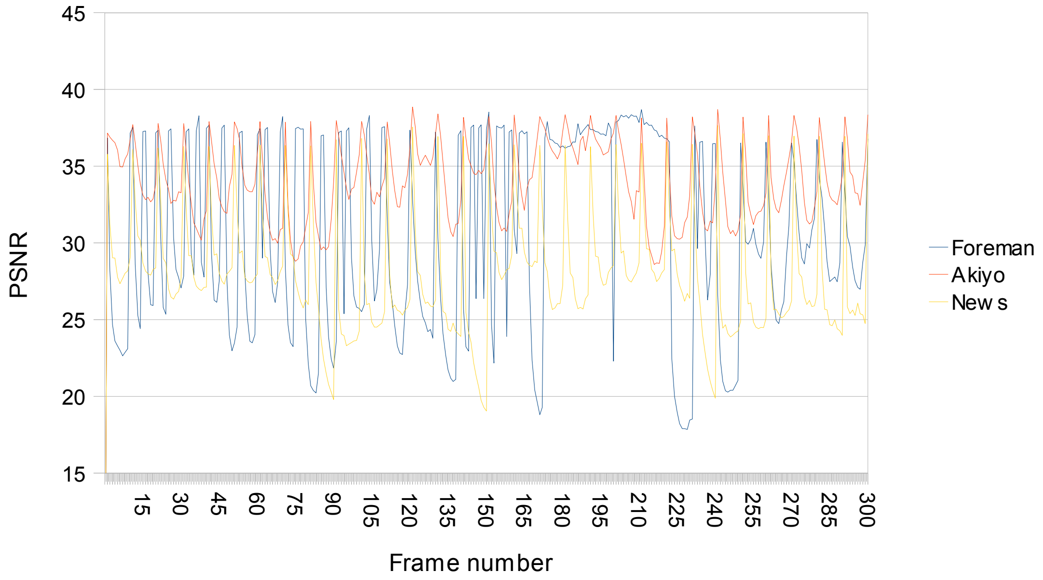

Figure 16: PSNR values of all video frames with parameters: refframe = 10; blsize = 256; changedmax = 30.

Figure 16 shows the PSNR values of the videos Akiyo, News and Foreman with the biggest parameter refframe value. Note that once more the value of the parameter blsize was increased. 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 video frames drops below 30 (although very occasionally and the average quality of this video is still above 30). Furthermore it is shown that the quality of News video frames and particularly Foreman video frames dropped dramatically. In fact most of PSNR scores of the frames of these two videos are around 25 and the difference between the PSNR scores of reconstructed frames and reference frames is around 20 dB most of the time[6]. This means that the difference between the original video frames and the recreated frames is high and there might arise coherency issues. Very occasionally does the PSNR score drop below 20, even with this parameter configuration. Another important fact is that the lowest PSNR score belongs to the frame of Foreman video although in Figures 12 and 14 it belonged to News video. Figure 17 shows the reconstructed frames with the lowest PSNR scores in this set of samples. Note that Frame 231 has the lowest PSNR score overall.

[6]The bottom points of the spikes in the Figures 12, 14 and 16 represent recreated frames and the upper points – reference frames.

Frame 172

Refframe=10
Blsize=256
changedmax=30
PSNR=19,27

Frame 231

Refframe=10
Blsize=256
changedmax=30
PSNR=17,85

Figure 17: Snapshots of frames with lowest PSNR score shown in Figure 16.

5.3 Validity Of The Results

The very nature of test videos raises some questions. The most pressing issue is how to determine the intensity of movements in the video. In the scope of this work we assigned arbitrary categories of movement intensity to each of the test videos based solely on the amount of pixels that change from frame to frame. This is the most simplistic and rudimentary way to categorize movement in videos. To better evaluate the effectiveness of the proposed video streaming technique more different and complex movement categorization systems should be used. Such a categorization system would allow, among other things, to develop a parameter selection system which is at present lacking. In regards to this issue it must be said that the three test videos have most of the movement categories that are the most troublesome in the field of video interpolation. These are: facial movements (Akiyo, News, Foreman), background/foreground movement (News), background/foreground scene change (News), camera movement (Foreman), background landscape movement (Foreman), body movement (Foreman). Another problem with the test videos is that all three of them are of QCIF format. This format is atypical in the fields of video interpolation and motion estimation. In fact motion estimation methods produce worse results when used on QCIF videos [1]. To remove this threat to validity test videos with various video formats should be examined. In fact this problem was anticipated and is discussed in chapter 7.

The second threat to validity lies in the method used to evaluate the quality of reconstructed videos. In general there are only two types of methods for quality evaluation of videos, they are objective and subjective quality evaluation methods. Two examples of these are the PSNR (Peak Signal to Noise Ratio) method [18], which is an objective quality evaluation method, and MOS (Mean Opinion Score) [19], which is a subjective quality evaluation method. Each method has its advantages and disadvantages. We chose the PSNR method because of practical reasons. Namely the number of times the run through of the case study was repeated is over 6000. This includes initial test runs, reruns for cross examining, reruns for result verification, reruns for determining the value ranges of operational parameters etc. It would have been highly impractical if not impossible to conduct a survey each time to evaluate the quality of produced videos. Because of this reason the PSNR method was chosen to evaluate the quality of reconstructed videos.

6 CONCLUSIONS

In this thesis a video streaming technique, that addresses the limitations of wireless networks by skipping part of the video frames on the sever side and reconstructing skipped frames on the client side, is presented and evaluated. The proposed technique uses intelligent frame skipping mechanism (IFSM) on the server side to skip frames based on changes in motion between frames and frame reconstruction mechanism (FRM) on the client side to reconstruct the skipped frames.

It is shown that the proposed video streaming technique is highly adaptable while accomplishing the set aim. Furthermore it was demonstrated that it is possible to skip video frames based on motion (even when using motion estimation techniques that emphasize execution speed rather than estimation of quality) and that it is possible to reconstruct the skipped frames on the client side.

The evaluation of the results of the proposed video streaming technique shows that the proposed technique is very flexible. It is shown that with the operational parameter values that allow the proposed technique to drop up to 50 % of the source video frames, the PSNR value of reconstructed frames never drops below 25 dB. By using more "aggressive" operational parameter values it is possible to reduce the number of source video frames up to 90 %; however in such cases the PSNR value of reconstructed frames stays around 19 dB. Furthermore, in cases where 90 % of the video frames was dropped the difference of PSNR values of reconstructed frames and reference frames is about 20 dB.

The provided empirical evidence are enough to conclude that the proposed video streaming technique is capable in reducing the frame rate and can be adapted for streaming videos where the frame rate needs to be reduced, to cope with the limitations in the wireless networks, however the quality of reconstructed videos is reduced.

6.1 Research Questions

In the beginning of the work four research questions were raised to map the direction the work will be taking as well as guide the project towards the thesis's main aim. Now we will examine these questions again to illustrate how the achieved results help answering them.

1. *How to skip frames on the sever side?*
   During the literature study performed in the beginning of the work a lot of informations regarding video frame skipping and video adaptation techniques was gathered. This knowledge allowed to construct the IFSM (Intelligent Frame Skipping Mechanism) as a basis of the server side of the proposed video streaming technique see section 4.1. This directly contributes to a part of the thesis aim to reduce the amount of frames that need to be transferred over the network.

2. *How to reconstruct skipped frames on the client side using interpolation techniques?*

The literature study showed that the most difficult part of video frame interpolation is which video frame interpolation tools and techniques to use. This task was one of the biggest hurdles to overcome. In the end according to the priorities of our video streaming technique the LSFS algorithm was chosen as a basis and FRM (Frame Reconstruction Mechanism) was developed which is the main part of the client side of the proposed video streaming technique.

3. *What is the relation between skipped frame count and quality of the videos?*
   To increase the adaptability and flexibility of the proposed video streaming technique three parameters were included in it's design. Results show that with various parameter combinations the skipped frame count ranges anywhere from 125 to 269 for the video Akiyo; 100 – 269 for the video News; 9 – 180 for the video Foreman. Respectively the quality ranges 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.

4. *How does the quality of the videos change after reconstructing the video frames?*
   The results show that after recreating videos even in worst cases the PSNR score rarely drops below 20. In contrast most of the time the recreated frames have a PSNR scores above 20 or even 30. The result depends on the choice of the operational parameter values. This shows the flexibility of the proposed video streaming technique and allows the user of the technique to stream the video with varying bandwidth in the network.

7 Future Work

In this thesis we have shown that the proposed video streaming technique can be effectively used for skipping video frames on the server side and reconstructing them on the client side. Nonetheless further enhancements and adaptations of the technique are possible. The most significant and interesting are described in this section.

The proposed video streaming mechanism has high computational complexity which impairs execution speed. However due to the temporal nature of video streams a high execution speed is required. One way of improving execution speed is by replacing part of computationally expensive motion estimation operations. This can be done by implementing similarity checking. Similarity checking would allow doing a frame skipping decision based on similarity of video frames, this would remove the need of motion estimation for sets of frames that are 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 impact of chosen motion estimation techniques to the results was not tested, it would be interesting to see how the proposed technique performs with different types of motion estimation techniques, when emphasis is given to quality of matching instead of execution speed. Furthermore more complex video interpolation techniques could be adapted.

In this work three testing videos considered where QCIF with 30 fps and in RBG colour format to further evaluate the proposed frame skipping technique more videos and of different formats and frame rates should be tested, this would give further understanding of the overall effectiveness of the proposed video streaming technique.

Finally, when testing the proposed technique it was assumed that we operate on a flawless network, thus effects of network errors on the proposed technique are unknown and should be studied, such study should show if the proposed technique is capable to operate in a network environment where errors(dropped frames, late frames, etc.) exist and their effects on its operation.

8 REFERENCES

[1]. Liu H., Zhang w., Yu S., Yang X. Channel – Aware Frame Dropping For Cellular Video Streaming. Institute of Image Communication and Information Processing, Shanghai Jiao Tong University, China . 2006.

[2]. Huo L., Fu Q., Zou Y., Gao W. Network Adapted Selective Frame-Dropping Algorithm for Streaming Media . IEEE. 2007.

[3]. Patil V., Kumar R. An Arbitrary Frame - Skipping Video Transcoder. Department of Computer Science and Engineering Indian Institute of Technology Kharagpur, India. 2005.

[4]. Thammineni A.,Raman A.,Chandra Vadapalli S.,Sethuraman S. Dynamic frame – rate selection for live LBR video encoders using trial frames. Ittiam Systems (Pvt.) Ltd., Bangalore, India 2008.

[5]. Balsree R., Thawani A.,Gopalan S.,Sridhar V. Inter – Frame Similarity based Video Transcoding . Proceedings of the Seventh IEEE International Symposium on Multimedia (ISM'05) , 2005.

[6]. Chia – Hung Y., Shih-Meng C., Shiunn-Jang C. Content-Aware Video Transcoding via Visual Attention Model Analysis. International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2008. IIHMSP '08.

[7]. Zheng, B., Atiquzzaman M. TSFD: two stage frame dropping for scalable video transmission over data networks. In Proceedings of IEEE Workshop on High Performance Switching and Routing, Dallas, TX, May 29-31, 2001, pp. 43 - 47.

[8]. Zhu, X. and Girod, B. Video streaming over wireless networks. In Proceedings of European Signal Processing Conference. EUSIPCO`07, Poznan, Poland, Sept. 3-7, 2007, pp. 1462-1466.

[9]. Lee I., Guan L. Reliable video communication with multi-path streaming using MDC. International Conference on Multimedia and Expo, 2005. ICME 2005. IEEE.

[10]. Robles V., Siller M., Woods J. Active Discarding Packet Mechanisms for Video Transmission. International Conference on System of Systems Engineering 2007. SoSE '07. pp.1-5, 16-18 April 2007.

[11]. Mahdi Asefi . Classification-Based Adaptive Search Algorithm for Video Motion Estimation . A thesis presented to the University of Waterloo in fulfilment of the thesis requirement for the degree of Master of Applied Science in Electrical and Computer Engineering . Waterloo, Ontario, Canada. 2006 .

[12]. Mahajan, D., Huang, F., Matusik, W., Ramamoorthi, R., and Belhumeur, P. Moving gradients: a path-based method for plausible image interpolation. ACM Transactions on Graphics, Vol. 28, No. 3, 2009, pp. 1 - 11.

[13]. Hiraiwa A., Fuse K., Komatsu N., Komiya K., Ikeda H. A Fully Automated Object Extraction from Video Stream :A useful tool for Distributed Object-based Browsing and Content-based Searching Systems. Atsugi Research Center, Telecommunications Advancement Organization of Japan . 1999.

[14]. Huang Y-W, Chen C-Y, Tsai C-H, Shen C-F, and Chen L-G. Survey on Block Matching Motion Estimation Algorithms and Architectures with New Results. *J. VLSI Signal Process. Syst.,* Vol. 42, No. 3, 2006, pp 297-320.

[15]. Cheng S-C, Hang H-M. A comparison of block-matching algorithms mapped to systolic-array implementation. IEEE Transactions on Circuits and Systems for Video Technology, Vol. 7, No. 5, 1997, pp 741 – 757.

[16]. Jing X., Chau L-P. An efficient three-step search algorithm for block motion estimation. IEEE Transactions on Multimedia, Vol 6, No. 3, 2004, pp 435 – 438.

[17]. IKoga T., Iinuma K., Hirano A. and Ishiguro Y. Motion Compensated Interframe Coding for Video Conferencing, Proc NTC81, New Orleans, LA, Nov 1981.

[18]. Huynh-Thu, Q., Ghanbari, M. Scope of validity of PSNR in image/video quality assessment. In IEEE Electronics Letters, Vol 44, No. 13, June 19, 2008, pp 800 - 801.
[19]. Itoh, Y., Tajima, K. and Kuwabara, N. Measurement of subjective communication quality for optical mobile communication systems by using mean opinion score. The 11th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Vol. 2, 2000, pp 1330.