Självständigt arbete på grundnivå

*Independent degree project - first cycle*

Datateknik
*Computer Engineering*

**Effect of additional compression features on h.264 surveillance video**

**Erik Comstedt**
Effect of additional compression features on h.264 surveillance video

Erik Comstedt

2017-06-12

MID SWEDEN UNIVERSITY
The Department of Information Systems and Technologies (IST)

Examiner: Ulf Jennehag, ulf.jennehag@miun.se
Supervisor: Elijs Dima, Elijs.Dima@miun.se
Author: Erik Comstedt, erco1400@miun.se
Degree programme: Civillingengör I datateknik 300 credits
Main field of study: Thesis Project DT099G
Semester, year: VT, 2017
Abstract

In video surveillance business, a recurring topic of discussion is quality versus data usage. A higher quality allows for more details to be captured at the cost of a higher bit rate, and for cameras monitoring events 24 hours a day, limiting data usage can quickly become a factor to consider. The purpose of this thesis has been to apply additional compression features to a h.264 video steam, and evaluate their effects on the videos overall quality. Using a surveillance camera, recordings of video streams were obtained. These recordings had constant GOP and frame rates. By breaking down one of these videos to an image sequence, it was possible to encode the image sequence into video streams with variable GOP/FPS using the software Ffmpeg. Additionally a user test was performed on these video streams, following the DSCQS standard from the ITU-R recommendation. The participants had to subjectively determine the quality of video streams. The results from the these tests showed that the participants did not notice any considerable difference in quality between the normal videos and the videos with variable GOP/FPS. Based of these results, the thesis has shown that that additional compression features can be applied to h.264 surveillance streams, without having a substantial effect on the video streams overall quality.

Keywords: Video stream, compression, frame rate, GOP, surveillance
Acknowledgments

I would like to thank my supervisor Elijs Dima for his guidance and advice during the course of this project.
# Table of Contents

Abstract .............................................................................................................................. iii
Acknowledgments ............................................................................................................ iv
Terminology ...................................................................................................................... vii

1 Introduction ................................................................................................................... 1
  1.1 Problem motivation ................................................................................................. 1
  1.2 Purpose .................................................................................................................... 2
  1.3 Overall aim ............................................................................................................. 2
  1.4 Concrete and verifiable goals ................................................................................. 2
  1.5 Outline .................................................................................................................... 3

2 Theory ........................................................................................................................... 4
  2.1 Video Surveillance .................................................................................................. 4
  2.2 Data bandwidth ..................................................................................................... 4
  2.3 Video streaming ..................................................................................................... 4
  2.3.1 Video streaming and data bandwidth .................................................................. 5
  2.4 Video compression ................................................................................................ 5
  2.4.1 H.264 ................................................................................................................ 6
  2.4.1.1 H.264 profiles ............................................................................................... 6
  2.4.2 Frame interpolation ............................................................................................ 8
  2.4.3 Frame blending .................................................................................................. 8
  2.4.4 Intra frame compression and inter frame compression ...................................... 9
  2.4.5 H.264 Encoders and Decoders ......................................................................... 10
  2.4.5.1 General h.264 encoder ................................................................................ 10
  2.4.5.2 General h.264 decoder ............................................................................... 11
  2.4.6 Additional compression features ....................................................................... 12
  2.4.6.1 Variable Frame rate .................................................................................... 12
  2.4.6.2 Variable GOP .............................................................................................. 13
  2.4.6.3 Dynamic ROI ............................................................................................... 14
  2.5 Video Cameras ....................................................................................................... 14
  2.5.1 IP cameras ........................................................................................................ 14
  2.5.2 Motion detection in video cameras .................................................................... 15
  2.6 Ffmpeg .................................................................................................................. 15
  2.6.1 Libavcodec ....................................................................................................... 16
  2.7 Peak signal-to-noise ratio ..................................................................................... 16
  2.8 Quality assessment methods .................................................................................. 17

3 Methodology ................................................................................................................ 18
  3.1 Literature study ...................................................................................................... 18
  3.2 Tools and equipment ............................................................................................. 18
  3.2.1 Axis M1065-LW .............................................................................................. 18
  3.2.2 Axis camera management and IPUtility ......................................................... 19
  3.3 Preliminary tests .................................................................................................... 19
  3.3.1 Model using the M1065-LW camera ............................................................... 19
## Terminology

### Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPS</td>
<td>Bits per second.</td>
</tr>
<tr>
<td>CCTV</td>
<td>Closed Circuit Television. Video surveillance.</td>
</tr>
<tr>
<td>CFR</td>
<td>Constant frame rate. A constant amount of frames is shown per time unit.</td>
</tr>
<tr>
<td>FPS</td>
<td>Frames per second. The number of image frames shown per second in a video stream.</td>
</tr>
<tr>
<td>GOP</td>
<td>Group of Pictures. A group of frames in a compressed video stream.</td>
</tr>
<tr>
<td>Hz</td>
<td>Hertz. Frequency measurement unit.</td>
</tr>
<tr>
<td>.MKV</td>
<td>Matroska. Video container format.</td>
</tr>
<tr>
<td>PSNR</td>
<td>Peak signal-to-noise ratio.</td>
</tr>
<tr>
<td>QP</td>
<td>Quantization parameter.</td>
</tr>
<tr>
<td>VFR</td>
<td>Variable frame rate. The amount of frames shown per time unit varies.</td>
</tr>
</tbody>
</table>
1 Introduction

Video surveillance is a key component in ensuring the safety and security of many organizations, cities, governments and in private properties. It provides real time monitoring of events such as people's movements, assets and the surrounding environment. This is used by organizations and companies to monitor their facilities while their employees are not present, such as during night time. It is mostly used as a way to prevent or detect a crime. In a similar way it is usually used to monitor private properties, such as houses or apartments. As safety is key responsibility to any city, many larger cities around the world have installed video surveillance solutions. It is an effective way to help the cities police identify criminals, such as muggers or vandals, and by this help guarantee their citizens safety [1]. It has also become more used in traffic situations alongside motion sensors, to notify if a vehicle is traveling faster than the allowed speed limit. One big debate in the video surveillance industry is the one between quality and data usage. But what level of quality is needed for surveillance usage, and what level of compressions can be used in order to use as little data as possible, while still maintaining the required level of quality? If we were to determine quality in terms of frames per second, different types of activities may require different frame rates. Say that one camera is monitoring a pedestrian crossing at a road. What difference would it make if this camera were to record at 5 frames per second, compared to say 25 frames per second? The recording with 25 frames per second would obviously be able to catch five times as much of what is happening at the crossing, but what purpose would that serve? Is it really needed? The camera recording at 5 frames per second would still be able to pick up the key events taking place at the pedestrian crossing, while using up less data to generate the video stream. However, let's imagine another scenario: This time there is a camera monitoring a crucial part in the production of a certain type of goods. In this case a high amount of frames per second would be helpful. As it would allow the user, be it either a human or a machine to pick up potential errors and faults in the production as soon as possible.

1.1 Problem motivation

In modern day video surveillance business, a recurring topic of discussion is video capture quality versus data usage. If quality is prioritized a good quality video stream can be generated, at the cost of a higher bit rate. This will also result in more memory usage. If instead limiting the memory usage is prioritized the generated video stream will end up with a lower bit rate, but also lower quality. With the release of the h.264 compression standard [8], a step has been taken towards achieving a sustainable trade off between quality and memory usage. The h.264 standard compresses the video stream without having a significant effect on the quality [2], and also lowers the bit rate drastically compared to other compression standards. In recent times companies and developers have begun to add additional compression features to h.264 video streams, such as
Effect of additional compression features on h.264 surveillance video
Erik Comstedt

Axis Zipstream [3]. The purpose of these additional compression features is to bring down the bit rate of the video stream, while still maintaining a high level of the video quality.

1.2 Purpose
Additional compression features often change the variable frame rate and the Group of Pictures (GOP) size on a h.264 video stream. The overall purpose of this thesis work is analyzing the effects of additional compression features on an h.264 video stream. And by this see if it is possible to add additional compression features to such video streams without affecting the videos overall quality. By achieving this it would be possible to save bandwidth and data usage, while still maintaining a good quality video, which is a recurring topic of discussion in modern day video surveillance industry.

1.3 Overall aim
The aim of the thesis is to improve visual quality whilst retaining low data usage by the use of additional compression features on a h.264 video stream.

To address this aim, the following research questions are defined:

1. Can compression using variable GOP be improved without losing image quality, and by how much?

2. Can the variable frame rate's minimum and maximum bounds be reduced without affecting the overall video quality, and by how much?

3. What types of faults can be expected from different types of media players when playing a h.264 video stream with applied compression features.

1.4 Concrete and verifiable goals
To find the answer to the thesis research questions the h.264 format will be studied, with the intent to determine the subjective difference between a normally compressed h.264 video stream, and a h.264 video stream with additional compression features added to it.

To find the answer to the thesis research questions the following concrete goals were defined.

1. Set up a test model that resembles a likely scenario in which the cameras might be used. Use this model to compare a normally compressed h.264 video stream to a h.264 video stream with additional compression features.
2. Using the set up model, try different kinds of variable GOP as well as the use of different frame rates and subjectively measure the difference the recorded video streams.

3. Analyze the results from the test model in different kinds of media players and determine the effect of playing the video stream in the media player.

4. Let non-expert test participants look at videos recorded using varying GOP size and frame rate and let them compare the subjective difference in quality.

1.5 Outline

Chapter 1 can be seen as an introduction, which describes the general problem and lists the overall aim as well as verifiable goals for the thesis work. Chapter 2 provides background theory regarding subjects relevant to the thesis work. Chapter 3 describes the methodology and the steps taken in order to obtain the results. Chapter 4 explains how the steps mentioned in chapter 3 were implemented. Chapter 5 shows the results of the thesis. Chapter 6 discusses these results and evaluates the overall aim as well as the verifiable goals.
2 Theory

In this chapter information relevant to the projects subject are being presented. This is to inform the reader about information that will be referred to in the subsequent chapters.

2.1 Video Surveillance

Video surveillance or CCTV is the monitoring of an area using one or several video cameras. The technology is commonly used to monitor public areas, such as parks, subway stations, street corners and similar. This is most of the time done with the intent to hinder or prevent crimes. One study shows that since the introduction of surveillance cameras around subway stations in Stockholm, the crime rates have fallen by 25% at those stations affected by the change [4]. Video surveillance is also commonly used by companies, such as hotels, airports, in order to monitor their facilities at all times. This is usually done with a similar purpose as when public areas are being supervised, in order to hinder and prevent crimes from happening. In certain industries it can be used as a way of monitoring the production process from another area, which can be useful if for example the area which the production is taking place is not a suitable environment for humans. One such example would be during the production of chemicals in the chemical industry.

2.2 Data bandwidth

The data bandwidth is the bit-rate capacity expressed in bps. Which is how many bits of data that can be processed during a one second time frame. The formula for digital bandwidth is shown below, where B is the bandwidth, D is the data size in bytes and time is the time in seconds.

\[ B = \frac{D}{t} \]

It should not be confused with the bandwidth used in signal processing, which is commonly referred to as analog bandwidth and is measured in hertz. While analyzing data bandwidth, it’s capacity is usually measured in throughput and good-put. Throughput is the amount of successfully delivered messages over a communication medium and good-put is the number of bits delivered from the network to it’s destination over a set period of time [5].

2.3 Video streaming

Video streaming refers to the process of delivering or receiving a video file before the entirety of the file has been recorded or downloaded. An example of this would be an end user streaming a video on an Internet website. The user has to be able to begin watching the video instantly, while the video is being
transmitted in the background. Whereas in a non-streaming alternative, the user would have to fully download the entirety of the video file before they could begin watching its contents.

2.3.1 **Video streaming and data bandwidth**

A certain amount of bandwidth is required in order to be able to stream a video. The higher the bandwidth, the more bits will be processed per second, which will ultimately lead to the video stream being processed faster. The time it takes for data to be transferred is called latency [6]. How much bandwidth that is required is also determined by which streaming protocol that is being used.

2.4 **Video compression**

The purpose of video compression is to reduce excess information in video data and by this use less network bandwidth and memory for the video. This is usually done by applying some sort of algorithm to the original video file that creates a compressed video file. Then in order to play the compressed video file a reversed algorithm is being used. The two algorithms that together compress and decompress the video file are usually called an encoder and a decoder.

During h.264 video compression there are three types of frames that are being used to compress the video into smaller size. These frames are known as I-frames, P-frames and B-frames[7]. I-frames are always included in the video compression, but the uses of P-frames and B-frames may vary depending on which video compression standard that is being used. A group of frames is called a group of pictures, or GOP for short. In a group of pictures there will always be at least one I-frame.

I-frames, which is short for inter frames, does not require any information from other frames in the group of pictures. The I-frames are also the highest quality frame out of the three kinds, but due to it’s high quality it is also the largest out of the three frames which is not ideal when it comes to video compression. Therefore it is the least used frame, see figure 1.

The second type of frame is the P-frame, which is short for predicted frame. P-frames are able to look at previous I-frames and P-frames for information. If the data stored in the previous I-frame or P-frame is similar to what is in the current frame, the data from the earlier frame will be used. A P-frame is more efficient to use than an I-frame.

The final and also most efficient frame to use in video compression is the B-frame, which is short for bi-directional frame. Similar to how P-frames can look backwards to other P-frames and I-frames, the B-frame can look in both directions for similar looking frames. Whereas P-frames can only look backwards.

The first frame of a group of pictures must be an I-frame. This is because the I-frame is the only independent frame out of the three frames. The other two types of frames will look at other frames and compare their data, but they need
something to compare with, and if they are the first frame in the group, they will not be able to do a comparison.

Figure 1. A group of pictures (GOP), with I-frames, P-frames and B-frames.

2.4.1 H.264

H.264 is an open video compression standard. Compared to other commonly used video compression standards such as MPEG-4 Part 2 and Motion JPEG the h.264 standard performs much better and are able to minimize the size of the compressed video by more than 50% compared to MPEG-4 Part 2 and more than 80% compared to Motion JPEG[8]. Having a smaller sized video allows the compressed video files to be created using less memory storage space and network bandwidth as compared to MPEG-4 Part 2 and Motion JPEG. This is achieved due to the fact that h.264 uses a different intra prediction scheme than MPEG-4 Part 2 and Motion JPEG when encoding I-frames. Using technology to reuse pixel values that have already been encoded earlier, the bit size will be cut down. This step is an important one in the h.264 video compression standards success. The h.264 standard has seven different profiles. These profiles are a set of algorithmic features which determines the complexity in the work performed by the encoder and decoder. Originally there were only two profiles, but it has since increased. Different profiles are more suitable for certain types of tasks, while other profiles are more suitable for others [9].

2.4.1.1 H.264 profiles

- **Baseline profile**: The baseline profile is the most basic profile available in the h.264 standard. It is aimed primarily towards low cost applications, such as video conferences and similar.

- **Main profile**: At the release of the h.264 standard the main profile was intended to be used as the main consumer profile, hence its name. It was at the time mostly aimed towards use in broadcast and storage applica-
tions. Since then the High profile has been released and the need for the main profile has started to dim down.

– **Extended profile**: The extended profile is intended to be used as a video streaming profile. It has a high compression capability.

– **High profile**: The high profile is the utmost profile used for disc storage applications and broadcasting. This is the profile that has been adopted into Blu-ray discs. The profile has taken over a lot of the main profiles uses since its launch.

– **High 10 Profile**: This profile is an expansion on the high profile. The difference is that the high 10 profile had support for up to 10 bits per sample picture precision. Hence the 10 in the profile's title.

– **High 4:2:2 Profile**: Once again an addition to an earlier profile. The high 4:2:2 profile builds on to the high 10 profile. Adding support for sub sampling while still retaining all the functions of high 10 profile. This profile is mostly used by professional applications which utilizes interlaced video streams.

– **High 4:4:4 Predictive Profile**: The high 4:4:4 predictable profiles builds on to the high 4:2:2 profile. It uses up to 14 bits per sample and the coding for each picture using this profile is coded as three separate color planes.

Table 1: The table shows the three most common profiles used in the h.264 standard as well as a comparison of their features and abilities[10].

<table>
<thead>
<tr>
<th>Feature</th>
<th>Baseline</th>
<th>Main</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>I and P frames</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>B frames</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Multiple reference frames</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>In-loop deblocking filter</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>CAVLC entropy coding</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>CABAC entropy coding</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Interlaced coding</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Transform adaptivity</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Quantization scaling matrices</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Separate Cb and Cr QP control</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
2.4.2 Frame interpolation

Frame interpolation, or motion interpolation as it is also known as, is a way of processing video streams. The aim of the technique is to create and generate frames based on existing frames. This is done in order to fill gaps between already existing frames, in order to make the video stream run more smoothly[11]. But also could be used when a lower frames per second video stream is being played on a higher frames per second medium. The generated frames are generated by looking at previous and next frame and uses them as a reference. The based out of the information stored in those frames, generates an additional frame to fill up the gap between the two reference frames. Thus, with these generated frames a higher frames per second video stream can be created. Over recent times, frame interpolation has become commonly seen in products such as High definition televisions and High definition video players.

![Frame Interpolation Diagram](image)

Figure 2: A comparison between two video streams. One without and one with the interpolation technology. The interpolated video stream generates additional frames, based on the previous and the next frames in the video stream. Using this technology the interpolated stream can combine the original video stream with the generated image.

2.4.3 Frame blending

In frame blending, which is a similar technique to frame interpolation, frames are blended together to create new frames. These frames will serve a similar purpose as the generated frames in frame interpolation. They will fill up the gaps between the original frames in the video stream. This generally works well. It does however cause the video stream to come out quite blurry when there is a lot of movements and actions to be captured. Because of this, the blended frames can look vastly different, causing the blended result to look odd.
Figure 3: The images above show two recordings. The left one without frame blending and the right one with frame blending enabled. Due to the high amount of movement in the recording, the frame blending recording come out blurry, and from a quality standpoint, with a worse result than the recording that was produced without it.

2.4.4 Intra frame compression and inter frame compression

*Intra frame compression* and *Inter frame compression* are two different techniques used when compressing frames in a video stream. The h.264 standard can apply both of these techniques. The intra frame compression technique is applied by looking at each individual frame in the video stream, and compress each individual frame by looking only at that frame. Inter frame compression looks at earlier and upcoming frames in the video stream and compresses the frame based on the data in those frames [12]. This yields a better compression result than Inter frame compression.
Figure 4: A comparison between intra frame compression and inter frame compression techniques[12]. The crumbled up papers symbolizes compressed frames. The intra frame technology only looks at individual frames when compressing. The inter frame technology looks at several frames in the video stream when compressing.

2.4.5 H.264 Encoders and Decoders

Encoders and decoders are a vital part in the world of video streaming. They encode and decode the raw video data from one form to another. In the case of videos it converts analog video signals to digital video signals. Additionally it is also the encoders and decoders jobs to select what data to keep and what data to discard, and also what data to compress and how that data should be compressed. The encoder and decoder are not part of the standard and does not have a standard way of being implemented. How they are to be implemented is up to the developers of the encoder and decoder. However, the resulting bit stream from decoder has to follow the standard which the encoder and decoder are suppose to be used for.

2.4.5.1 General h.264 encoder

There is no standardized implementation of an h.264 encoder. However, h.264 encoders are generally based off a certain model when being implemented. The raw video data from the video stream are broken down to frames, macro blocks and slices. This forms a 4 by 4 block. Each of these blocks are then encoded in the next step in the encoder, using either intra frame and inter frame prediction,
depending on the users choice. The predicted data received from the previous step is then subtracted from the original data, in order to obtain the remaining data. This is performed in order to remove redundant information from the encoding process. After this step is performed, the remaining data is transformed and then quantized. The quantization process in a general h.264 encoder is being controlled by a parameter known as the quantization parameter, or QP for short [13]. The quantization parameter is indicated by an integer value between 0 and 51. A lower QP value indicates higher quality, and a lower QP value indicates lower quality. The QP can also be described as the compression level controller in the h.264 encoder. The encoder has a built in decoder, which is utilized in the next step in the encoding process. By having a built in decoder, it allows the users to know that the encoder and decoder uses the same data when it comes to motion prediction [14].

![Flowchart of the components of a general h.264 encoder.](image)

**Figure 5:** Flowchart of the components of a general h.264 encoder. The raw video data is split up into blocks. Then either an Intra frame prediction or an Inter frame prediction is performed on the block. The prediction is subtracted from the original video data and the remaining data is then transformed and quantized. The data is then decoded and stored in a data pool containing data of the previous block in the video stream. Once enough video data has been processed, it is encoded along with other factors such as parameters or headers. The result of the encoding is a video file.

### 2.4.5.2 General h.264 decoder

In a general h.264 decoder the data is received and first decoded using a lossless technique for decoding. The received data from the decoder is then dequantized. Then a residual is obtained from using an inverse transform. The decoder then creates a prediction, which is based off the parameters in the data. Then the created prediction is added to our data, which leaves us with a decoded 4 by 4 block. Several of these blocks are then put together to create a video frame. Once all frames have been decoded in the decoder, the decoder leaves us with decoded video data [14]. An abstract view of the h.264 decoder model could be that it performs the tasks of the encoder, but in the reverse order.
2.4.6 Additional compression features

In recent times developers have started to look into adding further compression features to an already compressed h.264 video stream. The technologies used analyzes the situation in real time using a set of dynamic features. Using dynamic values for frame rate, group of pictures and region of interest, the technology can instantly adapt to any changes in the area which is being supervised. In short this means that the system will focus on things that are relevant and put less bandwidth and storage on redundant information [4]. For example, a car is driving by a camera using an additional compression technology, the camera would capture the car in full quality. The car is moving and therefore something is happening, which is good to have in full quality. Now lets say there is a tree at the side of the road that the car just drove by. Nothing is happening by the tree, and the tree is not going to start moving by itself, so this area is recorded with less bandwidth and memory, since it is not going to be relevant for the surveillance video anyway. Some of the features that are used for compression are Variable FPS, variable GOP and dynamic ROI. The subsequent sub-chapters, 2.4.6.1-2.4.6.3, explains these compression features in more detail.

2.4.6.1 Variable Frame rate

With a variable frame rate the camera recording the area of interest is able to first and foremost capture the video at a maximum frame rate. Then after transmitting the video stream further, the video stream is analyzed. Unnecessary and repeated video frames are then removed from the video stream. This allows scenes which has little to no action or general movement to be encoded using considerably less frame rate than it would normally require to encode it. If the scene being recorded later see changes within it, such as movement, the frame
rate of the video stream will automatically be adjusted to capture the event in full detail [15].

Figure 7: Comparison between a constant frame rate stream and a variable frame rate stream. The frames in the top stream is being shown at a constant rate. While the frames in the bottom stream is shown at a varied rate.

2.4.6.2 Variable GOP

The thought process behind variable GOP is similar to that of variable frame rate. The size of the group of pictures will be affected based on what is happening in the area which is being recorded. How this work is that the interval between the I-frames in the group of pictures is being adjusted dynamically based on the scene. In a recording, the I-frame contains the major details of the scene, in scenes with lots of movement the need for I-frames is higher due to the constant changes in what is being shown. Similar to in scenes where there is little to no movement, the need for I-frames is not nearly as large. As mentioned earlier, the I-frames are much bigger than P-frames and B-frames, and since the dynamic group of pictures technology alters the amount of I-frames used based on what is being captured, the size of the video stream will also be changed. More movement means more I-frames, which leads to more memory and bandwidth being used, and less movement meaning less I-frames, which has the opposite effect [16].

Figure 8: Example of variable GOP. The first two video streams have constant GOP sizes of 8 and 6 respectively. The third video stream has an unspecified value for its GOP size. As the distance between the I-frames is not locked to a specific value.
2.4.6.3 Dynamic ROI

Using dynamic region of interest the video stream is analyzed to find regions of interest. Regions of interest could for example be movement. The located areas of interest are then compressed less than the rest of the areas in the video stream, using up more memory and bandwidth on these areas of interest than on the other areas in the video stream. This gives the areas of interest better quality than the other areas that is being recorded [17].

Figure 9: The running woman is analyzed as a region of interest, and are therefore less compressed than the rest of the frame. The difference in quality between the different areas is noticeable.

2.5 Video Cameras

A video camera’s main purpose is to acquire electronic motion pictures and then save the pictures in a storage space. The images are joined together to form a video stream. Network IP cameras are a subgroup of video cameras. They are used by connecting the camera to the same network as a computer, and then recording settings, video and audio quality, and additional parameter settings is modified using an external program on the computer. This type of camera is commonly used in the surveillance industry.

2.5.1 IP cameras

IP cameras, or Internet protocol cameras, is a type of camera that sends and receives data over a computer network. These days they are commonly applied in the surveillance industry. The IP cameras operate over a network, using either Ethernet or Wi-Fi. The user can then access and modify the camera and it’s
many features using a certain software on either a computer or another device, such as a tablet or smart phone. Assuming that the computer or device is connected to the same network as the camera the user wants to connect to, and provided that the user has access to the correct login information, such as a user name and password. The IP cameras differ themselves from the older generation of cameras, the analog cameras. The analog cameras transmitted their video signals as analog signal, and the more modern IP cameras transmit their data digitally, using the TCP/IP protocol. Due to the fact that the camera operates digitally using the TCP/IP protocol they get access to a handful of security features that enables encryption and authentication. Examples of possible features are WPA, WPA2, TKIP and AES. A downside to the use of IP cameras is that the cameras require a stable Internet connection to be usable.

2.5.2 Motion detection in video cameras

Motion detection in video cameras is performed by the use of a built in program in the camera. The program will look for changes in the videos pixels, comparing what's being shown in the current frame with the previous frames. How many frames it compares with, and how much of a change in pixels that's required to be considered a movement can be changed by editing the motion detector's sensitivity [18]. A higher sensitivity will require less changes in pixels in order to determine it as a movement, whereas a low sensitivity level will require much more changes. The user will have to test different sensitivity levels in order to find out which level is a good fit for their particular situation. Another way to determine how much movement is taking place in the area which is being recorded is to analyze the contours of moving objects in the recording. Then the area of the contours is being calculated. The calculated area is then compared to the total area of the cameras view. If the calculated area is larger than a set value, it means that there is enough movements to trigger a motion detection [19]. In certain cases the camera wont even start recording until enough motion has been detected by the motion detection algorithm. Once the movements stop, and the pixels return to a normal and constant state, the camera will return back to it's natural position and start over again if more pixel variation is noticed.

2.6 Ffmpeg

Ffmpeg, short for fast forward mpeg, is an open source software that allows handling of multimedia data, such as video, image and audio files [20]. This is done by the use of several libraries in Ffmpeg. These libraries allows the user to perform several different actions on their multimedia data. The libraries included in Ffmpeg, as well as a brief explanation of their usage, are as following:

<table>
<thead>
<tr>
<th>Library</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Libavcodec</td>
<td>Library containing codecs for encoding and decoding video</td>
</tr>
</tbody>
</table>
Effect of additional compression features on h.264 surveillance video
Erik Comstedt 2017-06-12

<table>
<thead>
<tr>
<th>Libswresample</th>
<th>Library that performs re-sampling on audio files</th>
</tr>
</thead>
<tbody>
<tr>
<td>Libswscale</td>
<td>Library that performs rescaling and pixel format conversion on image files</td>
</tr>
<tr>
<td>Libavformat</td>
<td>Library that provides a framework for multiplexing and demultiplexing audio and video streams.</td>
</tr>
<tr>
<td>Libavutil</td>
<td>Utility library for programming</td>
</tr>
<tr>
<td>Libavfilter</td>
<td>Library which contains a filtering framework for audio and video</td>
</tr>
</tbody>
</table>

### 2.6.1 Libavcodec

*Libavcodec* is an open source collection of codecs which can be used for the encoding and decoding of video and audio streams. Due to its effectiveness and code availability it has become a vital part of many open source media applications. Example of such are the media players *VLC media player* and *Mplayer* which both uses *Libavcodec* as their main decoder for decoding video and audio files [20].

### 2.7 Peak signal-to-noise ratio

The Peak Signal-to-Noise Ratio (PSNR), is a measurement for measuring the overall quality loss during for example image compression. In such a case, the signal is referred to as the original image and the noise is the differences in the image caused by compression. When calculated using the PSNR formula, a value which approximates the quality difference between the two images is received. Below is a presentation of the PSNR formula [21].

\[
PSNR = 10 \cdot \log_{10} \left( \frac{MAX^2}{MSE} \right)
\]

The quality difference can also be described as how much does the compression affect the quality of the original image.

![Figure 10: An uncompressed image (left) and a compressed image (right).](image-url)
2.8 Quality assessment methods

In order to get a good understanding of the overall quality of either an image, or a video stream, quality assessment methods can be used. There are several different methods for performing these types of tests. Two of them, DSIS and DSCQS, are described here.

DSIS, which is short for Double-stimulus impairment scale method, is one method used for quality assessment. During a DSIS test, the subject is first presented with a reference image or video, followed by a modified version of that image or video. Once both of the images or videos has been shown to the user, the user will decide how much of a difference in terms of overall quality there were between the two images or videos. This is done based on a grading scale ranging from one to five where one is being the worst and five being the best. A good result in this case is when there is as little difference between the two images or videos as possible. This process is then repeated until all different combinations of images or videos has been tested. Once the combinations has been tested, the mean for each combination is calculated and becomes the result of the DSIS test [22].

DSCQS, short for Double-stimulus continuous quality-scale method, is the other quality assessment method. Similarly to the DSIS method, the subject is shown two images or videos after each other. These images or videos are usually shown twice, in an order such as: Image A, Image B, Image A, Image B. This is done in order for the subject to get a better view and make up a better perception about how they feel about each individual image or video sequence. Unlike the DSIS method, the subject is now given the task to determine the overall quality of both images or videos, not the qualitative difference between them, which is what's determined in the DSIS method. The test then continues until all combinations of test conditions has been shown to the test subject. After each combination has been tested, a mean value is calculated and becomes the result [22].
3 **Methodology**

In this chapter the methods used in order to finish the project are presented and described in detail. This is done with the intent to give the reader a good understanding of how the result for the thesis was achieved.

3.1 **Literature study**

In order to gain knowledge and insight about the problem and how to come up with a solution for it, a literature study would be performed. This is intended to be fulfilled by reading and studying books, reports, articles and videos relevant to the subject of video compression and video surveillance. The relevant parts of the literature study can be found in the previous chapter.

3.2 **Tools and equipment**

This sub chapter will describe the tools and equipment to be in the project. Such as cameras used to record video streams, media players used to playback the video stream and software used to make configurations and edits with.

3.2.1 **Axis M1065-LW**

The Axis M1065-LW is a wireless HD 1080p network IP camera [23]. It has a motion detection system as well as built in microphone for prerecording a message or some other type of audio. It supports both the h.264 compression standard and also the Motion JPEG compression standard. The cameras frame rate is capped at 25 fps for 50 Hz systems, and at 30fps for 50 Hz systems. It’s resolution can be modified by the user, going from 320x240 pixels all the way up to 1920x1080 pixels. For storage the camera offers the ability to store recordings on a micro SD card [23].
3.2.2 **Axis camera management and IPUtility**

In order to view and work with different parameters during the recording, the M1065-LW camera was connected to the applications Axis camera management and IPUtility. The camera management software is a program which allows the user to configure their devices and change its recording settings. This is needed in order to be able to create different recordings using different parameter settings. The IPUtility tool is a program which allows the user to set and change the IP address of a camera. This is needed in order to discover the camera on the network.

3.3 **Preliminary tests**

A model resembling a likely scenario would be set up for some preliminary tests. By using this model it is possible to analyze a compressed h.264 video stream and a h.264 video stream with additional compression features. From this it is possible to analyze the difference in video quality between the different compressions, and also compare their data usage when generating the video stream.

3.3.1 **Model using the M1065-LW camera**

A test model would be set up using an M1065-LW camera from Axis Communications. The aim of this test is to get an understanding on how varying GOP and frame rates affect the video streams, in regards to overall quality and file size. The set up test model is to be used to capture a certain area, using different compression settings.

It has to be made sure that the area that is being captured will remain constant throughout the recording. What this means is that the area being captured has to look the same for all compression settings tested, otherwise the results would be
redundant. Since for example if a road is being recorded, the amount of cars driving by is not a constant number, and will change over time. Which means that one type of compression might end up with more movements in its video stream than the other compression. Therefore a fair judgment and comparison can not be made unless a constant movement is being captured. Due to this fact, it was decided that a video should be prerecorded, using a different device to capture the video, and then use the M1065-LW camera to capture data from the already prerecorded video. Using the built in camera in a Samsung Galaxy S7 Edge, a mp4 video with 4k resolution was generated for later usage in the test model. The camera was set up to record a five minute sample of a road during a not very busy time of the day. This time of day was chosen due to the infrequent timing of vehicles driving by. If the area being recorded were mostly calm, it would be easier to spot the difference in frames per second when movements started to occur in the video.

The test model will then record the events of this prerecorded video using different parameter settings for GOP and frame rates. The GOP sizes that will be used for this experiment is 5, 32 and 200. The frame rates that will be used for this experiment is 5, 20 and 25 frames per second. The expected results, based off theory, is that the quality should increase with a decreasing GOP size, but the file size should increase. For frame rate the overall quality should increase with a higher frame rate, and so will the file size.

Figure 12: Schematics over the M1065-LW test model. Displaying how the camera was set up in order to record the events taking place on the display.

3.4 Examining playback in different media players

The next step in the process would be to analyze the effects of variable frame rate on different kinds of media players and what faults this would possibly cause. In order to play a video file some media players request the frame rate of the video as a parameter. However if the video file were to have variable frame rate, this question would be impossible to answer correctly, due to the frame rate of the video stream not being locked to a set value. In order to analyze the
events of playback in different media players, a video stream containing varied frame rate would be viewed using five different media players. The video streams is generated as a .MKV file and the media players used for this analysis were:

- VLC media player, developed by VideoLAN.
- Windows media player, developed by Microsoft.
- All player, developed by ALLPlayer Group Ltd.
- Media player classic, developed by Gabest.
- Kodi, developed by XBMC Foundation.

These media players were chosen based of accessibility reasons. The media player Windows media player is already installed on the computer that's going to be used in the project. The other four media players are open source software and therefore hold a high level of accessibility. The recording will be played in each of the five media players. To see how the different media players would react to the varied amount of frames per second in the video steam. Such as if the video stream were even playable, and if it were look for additional side effects of the varied frame rate. This would be done with the aim to answer the research question regarding media players and their faults when playing video streams with additional compression features.

3.5 **Experimental tests**

Once the preliminary tests has been finished, an additional set of experimental tests will take place, with the target to improve on the results received from the preliminary tests.

3.5.1 **Comparison between constant and variable frame rate**

In order to make out the clear differences between a constant and variable frame rate video stream, a deeper analysis on the two types of video streams is going to be performed. Two video streams, one with constant and one with variable frame rate, containing the same content were analyzed. During this process, the video streams will be examined on a frame by frame level to see how the individual frames compared to each other. The parameters that were analyzed were the length of each frame in seconds. Based off these results, it should be possible to see the compression effects of variable frame rate.

3.5.2 **Creating a customized variable frame rate video stream**

After the constant and variable frame rate video comparison has been finished, the results from these tests were analyzed. From those results it was decided that a customized version of variable frame rate should be implemented. The
desired aim with this would be to create a video stream with a variable frame rate, and by this be able to bring the file size down, while still maintaining a high level of quality. The constant 25 fps video, recorded by the test model in chapter 3.3.1, will be converted into an image sequence using Ffmpeg. By encoding this image sequence into several different videos, with different combinations for GOP sizes and frame rates, a variable GOP and a variable frame rate behavior can be imitated. This can be done by concatenating videos with different GOP sizes and frame rates into one video stream.

Once these videos has been produced, the PSNR for the videos will be calculated. For the variable video streams the average PSNR of each individual video, before being concatenated, were added up. An average PSNR score will then be calculated based out of all of the PSNR scores for the individual videos. The average PSNR value for each video will be received and this will be multiplied by a weighting system based off how long that video appears in the VFR video stream. The product of this is obtained for each video in the variable video streams. These products can then be added together and the sum of this becomes the average PSNR value for the variable video stream.

\[ Weighting_x = \frac{\text{frame count video}_x}{\text{total number of frames}} \]

\[ PSNR_x = \text{Average PSNR}_x \times Weighting_x \]

\[ PSNR_{fullvideo} = \sum_{1}^{n} PSNR_x \]

Weighting is how much out of the total amount of frames the current video is taking up. The PSNR \( x \) is then calculated, in order to get each average PSNR value proportionally to how long that video is being shown in the full video. Then the sum of all these values are added up to get an estimation of the PSNR for the full video.

In the formula above, \( x \) serves as a video ID, such as the number of frames in video \( x \), divided by the total number of frames gives the weighting for video \( x \). As well as the PSNR for video \( x \) becomes the average PSNR for video \( x \) multiplied by the weighting for video \( x \). The second variable, \( n \), is the total amount of videos that the image sequence was broken down into.

Based off the results from the PSNR tests, some of the videos will be chosen for further analysis later in the project. A group of non-expert participants will perform a DSCQS test on these videos, as described in more detail in chapter 3.5.2. The desired results would be that the participants would not notice a substantial difference in the overall quality of the videos.
3.6 User tests

Once the objective tests have been finished, some user tests will take place. Non-expert users will be asked to watch different video streams and pick out which ones they feel the highest overall quality. It would be ideal to have at least four participants for the subjective user tests.

Next, the videos chosen from the PSNR evaluation will be used for subjective user tests. The purpose of these tests will be to see if potential end users viewing experience will be effected by the additional compression features. More precisely, by variable frame rate in this case. Following the DSCQS method for subjective video tests, the participants will be shown different combinations out of the four videos chosen. Each video will be paired with the other three videos once, appearing either first or second in the pairing. Pairing each video with the other three videos will result in a total of six possible combinations. The participants will be given the linear grading scale, taken from the ITU-R recommendation [22]. The participants will be using this as a measurement and use it to rate the videos on a scale ranging from 1 to 100. The participants will be able to place their ratings anywhere on the linear grading scale. The ratings are based off how well they feel that the overall quality of the video is, and how useful they feel the video quality would be in terms of a video surveillance standpoint.

Figure 13: The linear grading scale from the ITU-R recommendation [20].

During the test, the video sequences will be shown for about ten seconds each. This time was chosen due to the fact that making the subject watch too long sequences would be damaging to the assessments results [24]. Once the video sequences has been shown twice, the users will be assigned to judge both of the video sequences. The test will then be performed for all different combinations of video sequences. Once the tests have been completed, and all possible combinations has been tested, the average rating for each of the four videos will be calculated and presented. If the participants do not notice a considerable differ-
ence in quality between the video streams, it would mean that additional compression features can be added to video streams, without affecting the quality of the recording.

Figure 14: Overview of how the presentation was performed. The subjects watch the first video (video A) for 10 seconds. This was then followed by a 3 second pause. The same pattern was then followed, and the test subjects were presented with video B, video A and video B in that order.

Figure 15: Comparison of two of the videos from one of the tests. Video A seen to the left, and video B seen to the right.
4 Implementation

This chapter contains information about how the processes described in chapter 3, methodology, is performed and what necessary steps are needed in order to receive the desired results.

4.1 Preliminary tests

In order to gain a better understanding in how different GOP sizes and frame rates affect video quality and data size, some preliminary tests were performed using the test model.

4.1.1 Testing difference in GOP size

After setting up the test model the GOP size and variable frame rate were to be tested. The camera was set up to record the prerecorded video stream three times. Each time a different GOP size were used for the recording, the GOP sizes that were used for this test were 5, 32 and 200. Theoretically a high GOP value should use less bandwidth, due to the low frequency of I-frames, but the videos quality will be lower than a video with a low GOP value. The results of the GOP size test using the test model is presented under chapter five, results.

4.1.2 Testing difference in variable frame rate

Similarly to how the GOP size of the video streams were tested, the frame rate of the video streams were to get tested. The camera was once again set up to record the already prerecorded video stream three times. For this particular test the different frame rates that were tested was 5, 20 and 25. Theoretically a higher frame rate will give higher quality, but the trade off will be more bandwidth usage. Similarly a lower frame rate will give lower quality, but use less bandwidth than a higher frame rate would require. A video stream with a lower frame rate will also appear choppy and hard to watch, due to the low amount of frames per second. Whereas a video stream with a higher frame rate will get a smoother appearance.

4.1.3 Testing Zipstream

Finally the Zipstream technology from Axis were tested using the test model. The camera was set up to record the prerecorded video stream twice. The first time using default factory settings, and the second time using the Zipstream technology. The Zipstream technology should be able to compress the bit rate further, compared to the default recording.

4.2 Experimental tests

Once the preliminary tests are finished, experimental tests will take place. This will be done in order to try and achieve better results than the ones received
from the preliminary tests. Such as a lower bit rate in the recordings, while still maintaining a high level of overall quality for the viewer.

4.2.1 Comparing constant and variable frame rate

A comparison between a constant and variable frame rate video were performed, as described in chapter 3. A detailed description of how this was achieved can be found in this sub-chapter.

4.2.1.1 Obtaining data on individual frames

The two video streams were analyzed using this ffmpeg command.

Ffprobe -show_frames -select_streams v:0 variable_fps.mp4 > output.txt

This command collects and returns information about the file variable_fps.mp4 and writes it to a text file named output.txt. After the command has finished executing the file output.txt contains information about the duration of each frame in the video stream, as well as other information such as what type of frame is being analyzed (I-, P-, B-frame). The command was then executed one more time, for the file constant_fps.mp4, which contains the same video data, but with a constant frame rate.

A C++ program was written to process and format the contents of the output.txt file. The program removed everything from the output file except for the numbers which represent the frame duration. This was performed as it allowed data to easily be transferred over into a spreadsheet for further analysis.

The contents received from the program was then pasted into a spreadsheet. Using the spreadsheet the total number of frames for both video streams were calculated. Additionally the longest, shortest and average frame duration of the streams were also calculated and analyzed.

4.2.2 Creating a customized variable frame rate video stream

A customized version of the video were then to be created, in order to get a comparison how this video would perform, in terms of quality and file size compared to a constant frame rate video. Both types of videos would later be shown to test subjects.

4.2.2.1 Obtaining the frames

In order to be able to create the customized variable frame rate video, all the frames from the recording had to be at our disposal. In order to obtain all frames, the constant frame rate video recorded at 25 frames per second during the preliminary tests were used as a reference video. This video was chosen because that video was the video with the highest frame rate from the preliminary
tests. From this video, each individual frame were extracted and saved as a .JPG image, using this ffmpeg command.

```
ffmpeg -i "input.mkv" -an -f image2 "%04d.jpg"
```

The above command takes a video file, in this case `input.mkv`, and extracts all frames from the video streams and saves them as .jpg image files. The `%04d` indicates that the images will be saved with a four decimal file name, ranging from 0001.jpg to 9999.jpg, depending on how many frames there are in the video file.

### 4.2.2.2 Determining the contents of the frames

Next each frame was looked at individually, to determine the content of the frame. If there were significant movements taking place between two frames, the frames were given a movement stamp, if there were not sufficient movement taking place between the frames, the frames were given a no movement stamp. Frames which had enough movement would then be encoded using a frame rate of 25 frames per second, and the frames without sufficient movement would be encoded with a frame rate of 5 frames per second.

**Table 3:** The table below shows the contents of the .jpg image files received from the video stream, as well as at which frame rate each group of frames were encoded at. Note that this table only shows the encoding process for the 5-25 variable frame rate video steam. Frames 1-199 were not used for this purpose.

<table>
<thead>
<tr>
<th>Frame Duration</th>
<th>Frame rate</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>200 – 718</td>
<td>5</td>
<td>No movement</td>
</tr>
<tr>
<td>719 – 785</td>
<td>25</td>
<td>Car driving from right to left</td>
</tr>
<tr>
<td>786 - 1048</td>
<td>5</td>
<td>No movement</td>
</tr>
<tr>
<td>1049 - 1262</td>
<td>25</td>
<td>Man walking from left to right</td>
</tr>
<tr>
<td>1263 - 1935</td>
<td>5</td>
<td>No movement</td>
</tr>
<tr>
<td>1936 - 2025</td>
<td>25</td>
<td>Car driving from right to left</td>
</tr>
<tr>
<td>2026 - 2245</td>
<td>5</td>
<td>No movement</td>
</tr>
<tr>
<td>2246 - 2300</td>
<td>25</td>
<td>Car driving from right to left</td>
</tr>
<tr>
<td>2301 - 2349</td>
<td>5</td>
<td>No movement</td>
</tr>
<tr>
<td>2350 - 2387</td>
<td>25</td>
<td>Car driving from left to right</td>
</tr>
<tr>
<td>2388 - 2502</td>
<td>5</td>
<td>No movement</td>
</tr>
<tr>
<td>2503 - 2557</td>
<td>25</td>
<td>Car driving from left to right</td>
</tr>
<tr>
<td>2558 - 2597</td>
<td>5</td>
<td>No movement</td>
</tr>
<tr>
<td>2598 - 3083</td>
<td>25</td>
<td>Several cars driving in both directions</td>
</tr>
<tr>
<td>3084 - 3094</td>
<td>5</td>
<td>No movement</td>
</tr>
</tbody>
</table>
### Effect of additional compression features on h.264 surveillance video

#### 4.2.2.3 Creating the video stream

After the frame content analysis had been performed, each group of .jpg images were encoded into a h.264 video stream, using either a frame rate of 5 frames per second, or a frame rate of 25 frames per second (see table 3). This was performed using the libx264 encoder in ffmpeg.

```bash
ffmpeg -framerate 25 -i %04d.jpg -c:v libx264 -psnr -crf 23 video25.mkv
```

The command above takes all images in the current folder, named after the `%04d` pattern, and encodes them with a frame rate of 25 into a video file named `video25.mkv`. The `crf` parameter controls the compression level, 23 is default compression. Libx264 tells ffmpeg to use the libx264 encoder.

For the parts that were supposed to be encoded with a frame rate of 5, frames had to be dropped. The reason for this is due to when a video stream is recoding events using a variable frame rate setting, the events happening determines how many frames is going to be captured per second. In this case since 25 frames are always being captured, individual frames had to be removed in order to replicate a variable frame rate behavior. Since 5 frames per second is equal to 20% of 25 frames per second, 80% of the frames in these groups of images had to be...
dropped. In order to do this, a simple shell script was written, which removes all .jpg files from the current folder where the filename does not end with either a 1 or a 6. Due to the .jpg images being named 01.jpg, 02.jpg, 03.jpg and so on, this was an easy way to remove 80% of the image files. Once the image files had been removed, the remaining images were used to encode a video file with a frame rate of 5 frames per second.

```bash
ffmpeg -framerate 5 -i %04d.jpg -c:v libx264 -psnr -crf 23 video5.mkv
```

The command used for creating the video file with a frame rate of 5.

### 4.2.2.4 Concatenating the created video files

Once all groups of images had been successfully encoded into video files, the video files were concatenated together into a single video file. As a comparison and reference, the entirety of the image sequence were also encoded into two additional video files. Both with constant frame rates. One with a frame rate of 5 and one with a frame rate of 25. Additionally the process listed from chapter 4.2.2.2 to 4.2.2.4 were performed for several more combinations of GOP sizes and frame rates, in order to get a selection of videos to chose from. Each combination of GOP sizes and frame rates were also encoded using different compression levels. Ranging from low to high, where low is less compressed than default settings and high compression is more compressed than during normal circumstances. The tables below lists which combinations of GOP sizes and frame rates were chosen for encoding.

#### Table 4: The different combinations of GOP sizes that were encoded from the image sequence.

<table>
<thead>
<tr>
<th>Video</th>
<th>GOP type</th>
<th>Min GOP</th>
<th>Max GOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Constant</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>B</td>
<td>Constant</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>C</td>
<td>Constant</td>
<td>250</td>
<td>250</td>
</tr>
<tr>
<td>D</td>
<td>Variable</td>
<td>30</td>
<td>250</td>
</tr>
<tr>
<td>E</td>
<td>Variable</td>
<td>100</td>
<td>250</td>
</tr>
</tbody>
</table>

#### Table 5: The different combinations of frame rates that were encoded from the image sequence.

<table>
<thead>
<tr>
<th>Video</th>
<th>Frame type</th>
<th>Min frame rate</th>
<th>Max frame rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>Constant</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>G</td>
<td>Constant</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>H</td>
<td>Constant</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>I</td>
<td>Constant</td>
<td>25</td>
<td>25</td>
</tr>
</tbody>
</table>
4.2.2.5 PSNR analysis of the produced videos

The produced videos had their peak-signal-to-noise (PSNR) calculated and recorded in a graph. By analyzing the results received from these graphs, it was possible to select which videos that should be used for further user tests later in the project.

In order to obtain the average PSNR for a video, the \texttt{-psnr} command in ffmpeg were used when encoding the images into a video stream. This command displays PSNR information on the screen during the encoding, as well as a summary once the encoding process is finished. This method was used in order to get the average PSNR for the constant GOP size and for constant frame rate video streams. For the variable video streams the average PSNR of each individual video, before being concatenated, were added up, using the formula which was introduced in chapter 3.5.2.
5 Results

This chapter contains the results received from the processes described in chapter 3 and chapter 4. Chapter 5.1 displays the results received from the preliminary videos. Chapter 5.2 contains the results from the media player evaluation. Chapter 5.3 contains the results from the process of creating the experimental videos. Chapter 5.4 contains the results from the subjective user tests.

5.1 Preliminary videos

Below are the results received from the tests performed using the test model using the M1065-LW camera, which was explained in chapter 3.3. The test model analyzed the GOP and frame rate of a video and their relation to image quality and bandwidth.

5.1.1 Preliminary tests - GOP

The same prerecorded clip was recorded three times with varying GOP size. The frame rate remained constant for these recordings at 25 frames per second. The size of the video goes down as the GOP size goes up.

Table 6: Results from the test model on GOP lengths.

<table>
<thead>
<tr>
<th>GOP size</th>
<th>Video Length (s)</th>
<th>Size (MB)</th>
<th>Bit rate (Kbit/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>156 s</td>
<td>105.8 MB</td>
<td>5425 Kbit/s</td>
</tr>
<tr>
<td>32</td>
<td>156 s</td>
<td>47.1 MB</td>
<td>2415 Kbit/s</td>
</tr>
<tr>
<td>200</td>
<td>156 s</td>
<td>25.2 MB</td>
<td>1292 Kbit/s</td>
</tr>
</tbody>
</table>
5.1.2 Preliminary tests – frames per second

The same prerecorded clip was recorded three times. For each recording the fps cap was set to a specific number. The GOP size was not modified for these recordings. The size of the video goes down as frames per second goes down.

Table 7: Results from the test model on frame rates.

<table>
<thead>
<tr>
<th>Frames per second</th>
<th>Video Length (s)</th>
<th>Size (MB)</th>
<th>Bit rate (Kbit/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>156 s</td>
<td>9.4 MB</td>
<td>482 Kbit/s</td>
</tr>
<tr>
<td>20</td>
<td>156 s</td>
<td>28.6 MB</td>
<td>1466 Kbit/s</td>
</tr>
<tr>
<td>25</td>
<td>156 s</td>
<td>30.5 MB</td>
<td>1564 Kbit/s</td>
</tr>
</tbody>
</table>

5.1.3 Preliminary tests – Zipstream

The same prerecorded clip was recorded twice. Once with the Zipstream technology enabled and once with it disabled. The Zipstream technology is able to reduce the videos file size by over 50%.

Table 8: Results from the test model on Zipstream.

<table>
<thead>
<tr>
<th>Zipstream</th>
<th>Video Length (s)</th>
<th>Size (MB)</th>
<th>Bit rate (Kbit/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disabled (cfr 25)</td>
<td>302s</td>
<td>120.0 MB</td>
<td>3178 Kbit/s</td>
</tr>
<tr>
<td>Enabled</td>
<td>302s</td>
<td>57.7 MB</td>
<td>1910 Kbit/s</td>
</tr>
</tbody>
</table>
5.2 Results from playback in different media players

In this sub-chapter the results from testing playback of a varied frames per second video steam, in a variety of different media players are presented. As mentioned in the methodology chapter, the media players that were being analyzed were VLC media player, Windows media player, Kodi, Media player classic and All player.

Table 9: Media players and how they handled the video stream.

<table>
<thead>
<tr>
<th>Media player</th>
<th>Playback possible</th>
<th>Additional side effects</th>
</tr>
</thead>
<tbody>
<tr>
<td>VLC media player</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Windows media player</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Kodi</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Media player classic</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>All player</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

As seen in table 9, all five of the media players were actually able to play the varied frames per second video steam. Additionally, they did not show any additional side effects while playing the video stream aside from a slight choppi-ness in the video.

5.3 Experimental videos

Below are the results received when comparing a constant and variable frame rate video stream, as well as the results from the customized variable GOP and variable frame rate video streams.

5.3.1 Frame comparison

Below are the results received from the comparison on individual frame level between a constant frame rate video stream and a variable frame rate video stream, see chapter 4.2.1

The data received when analyzing the constant frame rate video stream and a variable video stream are presented in the table below. As expected the constant frame rate only contains a frame of a constant length. The videos frame rate in frames per second becomes roughly 30 frames per second. The video uses a variable GOP length and therefore the GOP length varies. Unlike the constant frame rate video stream, the variable frame rate video stream's frames does not vary between two set values. The frame duration varies much more and goes both higher and lower than the set values in the constant frame rate video stream. The videos frame rate in seconds also becomes roughly 30 frames per second.
Table 10: Time information about the frames in the constant frame rate video stream and the variable frame rate video stream compared to each other.

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Constant frame rate</th>
<th>Variable frame rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video length (s)</td>
<td>302 s</td>
<td>302 s</td>
</tr>
<tr>
<td>Total number of frames</td>
<td>9079 frames</td>
<td>9079 frames</td>
</tr>
<tr>
<td>Average frame duration (s)</td>
<td>0.033 s</td>
<td>0.0332278995 s</td>
</tr>
<tr>
<td>Longest frame duration (s)</td>
<td>0.033 s</td>
<td>0.03435 s</td>
</tr>
<tr>
<td>Shortest frame duration (s)</td>
<td>0.033 s</td>
<td>0.03266 s</td>
</tr>
<tr>
<td>Average frame rate (fps)</td>
<td>30.06291391</td>
<td>30.06291391</td>
</tr>
</tbody>
</table>

Figure 17: Displays the different durations in seconds of each individual frame in the variable frame rate video stream.

5.3.2 Customized variable GOP and variable frame rate

Below are the received values from when encoding the image sequence into several different video streams with varying combinations for GOP sizes and frame rate, see chapter 4.2.2. These videos were also encoded with different compression levels. The PSNR scores did not vary much between varying GOP sizes, it did however vary a fair bit between varying frame rates. Therefore it was decided that the subjective user tests should focus around testing different compressions based around frame rates instead of GOP variation.

Table 11: The relationship between the GOP sizes and compression levels of the video streams.
Effect of additional compression features on h.264 surveillance video
Erik Comstedt
2017-06-12

<table>
<thead>
<tr>
<th>Video</th>
<th>Min/Max GOP</th>
<th>High compression PSNR</th>
<th>Medium compression PSNR</th>
<th>Low compression PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>30 constant</td>
<td>45.241</td>
<td>48.243</td>
<td>51.387</td>
</tr>
<tr>
<td>B</td>
<td>100 constant</td>
<td>44.793</td>
<td>47.571</td>
<td>50.686</td>
</tr>
<tr>
<td>C</td>
<td>250 constant</td>
<td>44.331</td>
<td>46.996</td>
<td>49.998</td>
</tr>
<tr>
<td>D</td>
<td>30-250</td>
<td>44.432</td>
<td>47.114</td>
<td>50.105</td>
</tr>
<tr>
<td>E</td>
<td>100-250</td>
<td>44.209</td>
<td>46.878</td>
<td>49.868</td>
</tr>
</tbody>
</table>

Table 12: The relationship between the frame rates of the video streams and their file size.

<table>
<thead>
<tr>
<th>Video</th>
<th>Min/Max FPS</th>
<th>High compression PSNR</th>
<th>Medium compression PSNR</th>
<th>Low compression PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>5 constant</td>
<td>47.46</td>
<td>50.307</td>
<td>53.24</td>
</tr>
<tr>
<td>G</td>
<td>10 constant</td>
<td>46.211</td>
<td>49.165</td>
<td>52.148</td>
</tr>
<tr>
<td>H</td>
<td>15 constant</td>
<td>45.14</td>
<td>48.064</td>
<td>51.047</td>
</tr>
<tr>
<td>I</td>
<td>25 constant</td>
<td>44.331</td>
<td>46.996</td>
<td>49.998</td>
</tr>
<tr>
<td>J</td>
<td>2-25 FPS</td>
<td>47.297</td>
<td>50.119</td>
<td>52.991</td>
</tr>
<tr>
<td>K</td>
<td>5-25 FPS</td>
<td>45.940</td>
<td>48.723</td>
<td>51.683</td>
</tr>
<tr>
<td>L</td>
<td>10-25 FPS</td>
<td>45.202</td>
<td>48.023</td>
<td>51.008</td>
</tr>
</tbody>
</table>

Figure 18: Visual display of how the PSNR varies for different GOP sizes. For the variable GOP size video streams, the average PSNR is shown.
Subjective user tests results

A total of four participants were used for the subjective user tests. The participants for the subjective DSCQS tests were assigned to watch all combinations of the four videos, making it 6 comparisons in total. The participants gave both videos in each comparison a score. These scores are presented in the table below.

Table 13: The table shows the score which each video received from their test subject during the user tests. The average score is shown in the results column.

<table>
<thead>
<tr>
<th>Video</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
<th>L</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFR 5 FPS</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1.0</td>
</tr>
<tr>
<td>CFR 25 FPS</td>
<td>100</td>
<td>95</td>
<td>95</td>
<td>55</td>
<td>95</td>
<td>100</td>
<td>95</td>
<td>95</td>
<td>100</td>
<td>80</td>
<td>75</td>
<td></td>
<td>90.0</td>
</tr>
<tr>
<td>VFR 5-25 FPS</td>
<td>100</td>
<td>90</td>
<td>60</td>
<td>55</td>
<td>75</td>
<td>80</td>
<td>60</td>
<td>90</td>
<td>95</td>
<td>60</td>
<td>55</td>
<td></td>
<td>73.33</td>
</tr>
<tr>
<td>VFR 10-25 FPS</td>
<td>100</td>
<td>100</td>
<td>80</td>
<td>95</td>
<td>75</td>
<td>95</td>
<td>80</td>
<td>70</td>
<td>90</td>
<td>90</td>
<td>100</td>
<td>90</td>
<td>88.75</td>
</tr>
</tbody>
</table>
6 Discussion

This chapter analyzes and discusses the results from chapter 5. Chapter 6.1 evaluates the different results from the thesis work. Chapter 6.2 analyzes the aims which were set up at the beginning of writing this thesis, and evaluates if those aims were reached or not. Chapter 6.3 summarizes potential future works that could be done regarding this subject. Chapter 6.4 is a discussion regarding the ethical issues of additional compression features and video surveillance.

6.1 Evaluation

6.1.1 Preliminary results

This sub-chapter discusses and evaluates the results received from the preliminary tests.

The test model was set up in order to record and compare different video streams, using different compression features added to them. Theoretically the data size of the recording should go down as the GOP size goes up. As seen in table 6, this was also proven to be the case from the tests made using the test model. As the GOP size went from 5 to 32 to 200 the file size containing the recording went from 105.8 MB to 47.1 MB to 25.2 MB. The quality should also increase as the GOP size decreased, which also was the case.

Similarly according to the theory the quality, in terms of how much information is being picked up in the video stream, should increase as the amount of FPS goes up. The file size of the recording should also decrease as the FPS goes up. Once again this is exactly what happened when the FPS went up, the file sizes of the different recordings decreased as frame rates went down, which can be seen in table 7.

The Zipstream technology was also evaluated during the preliminary tests. As expected, the technique is able to cut down on the bit rate. See table 8.

6.1.2 Media player tests

This sub-chapter evaluates the media player tests and the results received from those tests.

The results of the media player evaluation tests shows that the playback were possible in all five of the media players. The evaluation also showed that playback was almost identical in all of the five tested media players.

The variable frame rate video which was tested, was saved as a .mkv file. The .mkv file format has a time stamp for each frame in the video. If the vari-
able frame rate video was saved using a different container, the results of the playback would have possibly been different.

6.1.3 Experimental results

This sub-chapter discusses the steps taken in the process to create a customized variable frame rate video stream, as well as the results received during that process.

Two videos, one with CFR and one with VFR, were compared, see table 9. During this comparison both of the videos looked fairly similar on a frame level. This was a fairly small sample size though.

Additional compression features, being variable GOP and variable frame were created out of an image sequence. The image sequence itself was created out of a constant FPS/GOP video stream. With these additional compression features, it was possible to bring down the file size compared to the original video streams. This allowed the video to be compressed further, while still maintaining a high level of quality. During the encoding process, the videos were also encoded with varying QP values. To allow the compression level of the video to be adjusted to both higher and lower values. A PSNR evaluation was then performed on the videos, for the videos with varying GOP sizes, figure 18. As expected, the PSNR value goes up as the QP parameter is altered. A higher level of compression gives a lower PSNR and a lower level of compression gives a higher PSNR. For the videos with varying frame rates, figure 19, a similar result can be seen. Additionally from these tests it was noted that varying FPS affected the bit rate more than varying GOP. As seen in the graphs, the difference in bit rate between the leftmost and rightmost points is much greater in graph 4. Therefore, the subjective user tests were performed in regards to varying frame rates.

6.1.4 Subjective user tests

This sub-chapter discusses the results from the subjective user tests performed using the DSCQS method.

During the subjective user tests, the participants felt that 5FPS CFR video stream offered a bad level of quality, table 13. This is understandable, as the video the participants were asked to look at involved a bit of motion, which comes out quite choppy with such a low frame rate. Aside from that video, the other three videos received similar scores during the tests. Especially the 25FPS CFR video and the 10-25FPS VFR video, whose average rating were only separated by 1.25 points. This indicates that the participants did not notice a substantial difference between the remaining three videos, and almost no difference whatsoever between two of them. Even though the variable frame rate videos were compressed at a higher rate, and had a smaller file size than the constant frame rate video.
The results of the subjective user tests shows us that a feature like variable FPS can be applied to video steams, without having a negative impact on the videos quality. As the participants in the test could not notice a substantial difference between the video's quality. This brings us back to our research questions asked at the beginning of the report. It can now be said that additional compression features, such as variable FPS in this case, can be applied to a h.264 video stream without noticeably affecting the video streams quality.

6.2 Evaluation of overall aims

The three aims were set up at the beginning of writing this thesis will be presented and evaluated here, as well as an evaluation of the thesis overall aim.

Can compression using variable GOP be improved without losing image quality, and by how much?

This aim has been achieved. The PSNR tests shows that using variable GOP, it is possible to bring down the bit rate without affecting the image quality by a substantial amount. But it also showed that varying GOP sizes has less of an impact on quality and bit rate, compared to varying the frame rate.

Can the variable frame rate's minimum and maximum bounds be reduced without affecting the overall video quality, and by how much?

This aim has been achieved. The subjective user tests showed that the participants didn't notice a substantial difference between the videos which had variable frame rate and the one that had constant frame rate. The user tests showed us that variable frame rate can be applied without noticeably affecting the overall image quality.

What types of faults can be expected from different types of media players when playing a h.264 video steam with applied compression features.

This aim has been achieved. The media player evaluation showed that there were no issues for the media players when playing video streams with additional compression features applied to them.

The overall purpose of this thesis work is analyzing the effects of additional compression features on an h.264 video stream. And by this see if its possible to add additional compression features to such video streams without affecting the videos overall quality.

This was the overall purpose for the thesis work, and it can be said that this purpose has been fulfilled. The effects off additional compression features has been analyzed and evaluated. Additionally, it has been proved that these additional compression features can be applied to such video steams without affecting the overall quality.
6.3 Future work

There are still areas to evaluate in terms of effects of additional compression features on h.264 streams. Possibilities for future studies could be to experiment with different types of compression features, such as dynamic ROI.

Another possibility is to evaluate different media players and containers than the ones explored during this thesis work and see how they handle additional compression features. As well as test dynamic ROI on media players, if that compression feature were to be evaluated.

6.4 Ethical issues

As a fundamental part of video surveillance is capturing the events of real life people and events, questions can be asked about how the concept of video surveillance goes against human integrity and privacy. One of the main arguments against the use of video surveillance is that it is seen as a threat towards the privacy of people. Many people do not feel comfortable knowing that they are being watched, recorded and possibly analyzed by various surveillance technologies. Such as the ones that can be seen around public areas in major cites and in bigger industries.

In terms of ethical issues surrounding additional compression features, there is an issue regarding the integrity of using variable frame rate. In variable frame rate video streams there is no way to tell exactly how many frames is supposed to be played at a certain point in the video stream. If compared to a normal video capped at for example 50 frames per second, you know that for each second in the video there is going to be exactly 50 frames, and you can with this information easily analyze each frame individually. Now with variable frame rate there is no way of doing this, due to the frame rate being dynamically changed during the video stream.

This opens up the opportunities to censor the video stream, which would not have been possible with a traditional, static frames per second, video stream. With a video stream having dynamic frames per second it is impossible to determine if a frame has been removed from the video stream, due to the fact that it is not defined anywhere how many frames should appear at given times. This opens up loopholes and allows users to remove frames off events they would not want other people to see, and thereby censoring the video stream using this technique. It opens up doors to perform illegal acts, such as crimes or similar, due to the fact that the offenders know that the video streams frames can be edited to remove the recorded events without anyone taking notice. Additionally, if used as evidence material there is no way to prove that frames has not been removed from the video stream. This has even caused that variable frame rate may be prevented from being used, due to legal requirements [25].

Looking at a variable frame rate video from a safety and security standpoint this can clearly cause issues, since the whole fundamental idea of video surveillance is to provide safety and security. When a technology like this allows users to
bypass the most integral part of video surveillance as a whole, it is going to have a negative impact on the usage of the method.
References


http://wolfcrow.com/blog/intra-frame-vs-inter-frame-compression/
Retrieved 2017-03-24

https://www.vcodex.com/h264avc-4x4-transform-and-quantization/
Retrieved 2017-04-15

RWTH, 2012

with-axis-zipstream/dynamic-fps
Retrieved 2017-01-23

[16]  Axis Communications AB, “Dynamic GOP”
with-axis-zipstream/dynamic-gop
Retrieved 2017-01-23

with-axis-zipstream/roi
Retrieved 2017-01-23

https://www.videosurveillance.com/blog/technology/how_does_motion_
detection_work.asp
Retrieved 2017-02-21

[19]  Robindavid, “Motion Detection With OpenCV”
http://www.robindavid.fr/opencv-tutorial/motion-detection-with-
-opencv.html
Retrieved 2017-02-21

https://ffmpeg.org/documentation.html
Retrieved 2017-06-08

[21]  Everything about data compression, “MSU Quality Measurement Tool:
Metrics information”
Retrieved 2017-06-08

[22]  Series BT: Methodology for the subjective assessment of the quality of
television pictures, Recommendation ITU-R BT.500-13, 2012
[23] AXIS M1065-LW Network Camera Datasheet, Axis Communications
