Audio Detection for Preparation of Video Rate Controller

Olle Axelsson & Tobias Bladh

June, 2021

LUND
UNIVERSITY
Faculty of Engineering LTH
Department of Biomedical Engineering

Master's thesis work in Electrical Measurements carried out at
Axis Communications AB
Supervisors: Martin Stridh, Alexander Toresson
Examiner: Frida Sandberg

Abstract

Modern video streaming may use the protocol H.264 to encode video in high quality. The protocol takes advantage of similarities between consecutive frames to lower the bit rate. If there are differences between frames the bit rate will therefore rise. To control the fluctuations in bit rate a rate controller (RC) can be used together with the encoder. The RC manages the quantisation levels of the encoder. Rapid unforeseen changes may require the RC to drastically increase the quantisation parameter (QP) or even skip frames.

This paper evaluates if audio event detection can be used to prepare the RC for incoming changes by creating a warning signal that acts on the RC. Two methods are investigated; restricting the minimum allowed QP during detections, and allowing higher bit rates temporarily. The paper also examines a number of methods for performing audio event detection. Three types of Gaussian mixture models (GMM) are implemented and tested. Further, two envelope based algorithms are used. Finally, one GMM and one envelope algorithm are used together with the minimum QP scheme to test a combined solution.

Results show that both envelopes and GMMs can be used to perform detection useful for the purposes of RC preparation. When acted on by the warning system the RC lowers the amount of frame skips on most tests. Further, the peak quantisation needed during an event is often lower. Investigation of the resulting video quality shows that the quality score is lower in terms of VMAF and SSIM before and after the event happens. However, during the event the quality is roughly the same and sometimes slightly higher. When a real audio detection is used for preparation the frame skips are improved, but there are side effects in terms of quality related to false positives in the audio detection.

Acknowledgments

We want to thank several people for helping us write this thesis. First we want to thank Martin Stridh for his work as a supervisor at LTH helping us along the way. We also want to thank our examinator Frida Sandberg. Special thanks are sent to our supervisors at Axis, Alexander Toresson and Linus Lexfors. You have given us so much of your time, helping us understand complex systems as well as finding new and creative ideas. We are also grateful to all our colleagues at the company, who have helped us with valuable explanations and support when needed.

Olle Axelsson & Tobias Bladh

List of Acronyms and Abbreviations

1D-GMM Single Dimensional Gaussian Mixture Model

CBR Constant Bit Rate

DFT Discrete Fourier Transform

FFT Fast Fourier Transform

FPS Frames Per Second

GMM Gaussian Mixture Model

GOP Group Of Pictures

MAD Mean Absolute Difference

MB Macro-Block

MD-GMM Multi Dimensional Gaussian Mixture Model

MD-GMM-UF Multi dimensional Gaussian mixture with Updated Features

MEL Scale based on pitch comparisons

MFCC Mel-Frequency Cepstral Coefficients

MVS Minimum Volume Set

QP Quantisation Parameter

RC Rate Controller

RDO Rate Distortion Optimisation

SEA Sub-band Energy Amount

SVM Support Vector Machine

VBR Variable Bit Rate

ZCR Zero Crossing Rate

Contents

Chapter 1
Introduction

Video surveillance is an ever growing technology in terms of its usage. More places are being watched with the help of cameras and the technology involving cameras is constantly evolving [1]. As more cameras are being used and the video quality from the cameras are getting better, the technology to decrease data size of the video stream is crucial to limit the storage and network bandwidth required. Several technologies for compressing a video stream such as H.264 have been developed and are widely used. The performance of technologies such as H.264 have greatly improved compression in terms of video size while still keeping quality high. There is however still a trade-off between bandwidth and quality [2].

The H.264 protocol encodes the video stream based on partitioning of frames and groups of frames. There are independent frames, called I-frames, and frames that build upon previous frames, called P-frames. P-frames can be seen as a delta, only containing information about what has changed from the previous one. Depending on the amount of motion in the camera field of view the P-frames sizes vary in terms of bits. It is often desirable to keep a constant bit rate, and therefore a Rate Controller (RC) is usually implemented together with the encoder [3].

The RC keeps track of frame sizes over time and regulates the compression level of the encoder by changing the Quantisation Parameter (QP). The QP value controls the quantisation level during encoding. A more quantified signal generates less bits at the cost of worse quality. Thus the QP value will together with the image complexity and the amount of motion in the scene result in a frame size for each encoded frame [3]. By adjusting the quantisation, the bit rate from the camera can be controlled for scene changes. However, this regulation can only be done by estimating the required bits for the next frame based on previous frames [4].

A RC regulates the QP in order to keep a stable and limited bit rate. This works well if the scene is somewhat stable and with predictable changes over time. However, if the scene changes rapidly it might not be possible for the RC to predict the correct

QP to use for the new situation. This may result the RC having to compensate with a higher QP later to maintain its budget. Depending on the protocol and how strict the bit rate budget is, some frames may even have to be skipped in order for the budget to hold [5].

In previous research audio classification has been used in both surveillance and encoding domains. Previous work includes classifying foreground events within the audio stream, to indicate when the audio signal should be compressed with higher quality. The classification in that work was performed using a Gaussian Mixture Model (GMM) which proved to be quite accurate [6].

1.1 Purpose

The purpose of this thesis is to create an algorithm that analyses the sound environment around the camera in order to predict imminent changes in the video stream. The idea is that the microphone sensor typically has a wider capture angle compared to a cameras field of view. It should therefore be able to notice incoming changes early, given that these events generate sound. In order to do this the audio scene has to be classified into different categories depending on their relevance to the camera. Further, it is investigated if this information may be used to modify parameters of the RC to better react to incoming changes. The goal is to improve the performance of the encoded video stream in terms of the QP levels used, the number of skipped frames, and the resulting image quality.

We hope that this work can be used as a foundation for further research into RC preparation. This project will show how audio detection works in specific streaming scenarios.

1.2 Distribution of Responsibility

The responsibility of this work has been shared equally between the two authors. This includes knowledge gathering, programming and testing, data analysis, and writing.

1.3 Outline

The thesis is outlined as follows; First a chapter containing a theoretical background and relevant research. Next, there is a chapter containing the methodology, and an explanation to algorithms which are evaluated. After that the results are presented, followed by a chapter analysing the results. Finally there is a chapter where conclusions about the results are drawn.

Chapter 2
Theoretical Background

This chapter explains the theories necessary to understand the methods. It starts by discussing H.264 encoding, and then rate control. After that different types of sound based event detection are introduced.

2.1 References

References used in this thesis have been collected by searching on relevant key words. Most articles have been found by searching via Lund University libraries databases. Articles have also been found by going through reference lists of articles with similar background theory as this thesis. Every article found is also evaluated based on the title and abstract of the source. Promising sources have been further reviewed to evaluate the relevance to this thesis work.

2.2 Video Stream Coding

Digital video technology has generated several new applications for video usage, with some applications having high demand on video quality. This has created a high demand for an efficient representation of video to be able to stream high quality video with lower bit rate. H.264/AVC is a standard developed to significantly increase performance on streaming video compared to its predecessors [2].

2.2.1 H.264

A brief introduction to H.264 is given below. The main purpose of introducing H.264 is to give an understanding on where the quantisation parameter (QP) comes from and how it affects the video stream.

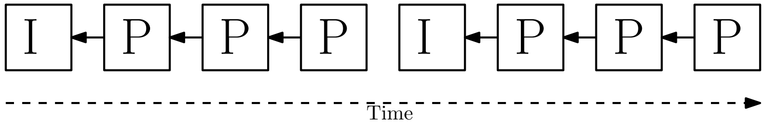


Figure 2.1: A stream with one I-frame and three P-frames per GOP. The P-frames reference back to the previous frame.

H.264 is a standard developed to handle the high demands of high quality video streams in various applications such as TV-broadcasting, videoconferencing and digital storage. It is also designed to be flexible and usable in a wide variety of network environments [7]. H.264 consists of a stream of frames in sequence. Previously coded frames may be used in coding of future frames by using inter-prediction. Frames used for prediction are called reference frames. Each coded frame is built up by Macroblocks (MB), which are grouped into slices. Slices are coded in different ways, namely *I-slices*, which only contain *intra-coded* MBs and *P-slices* which may contain *inter-coded* MBs. Intra coded MBs are predicted from previously coded samples in the same slice, and inter coded MBs are coded by using samples from previously coded pictures. Because I-slices only make predictions within the slice, they are independent and can be seen as a full representation of the slice, whereas the P-slices provide the changes that happen in the slice from frame to frame. The P-slices are not limited to carrying the differences within specific MBs. An update can also contain information about MBs that have moved to another location within the slice [3].

It is not uncommon to represent an entire frame as a single slice. In this case the frames become I-frames and P-frames. As I-frames have to be independent they contain a lot of information and are generally quite large. P-frames are smaller, especially for scenes with small motion changes. Usually frames are sent in a repeating sequence called a group of pictures, or GOP. This can for example be an I-frame followed by a number of P-frames. An example of this is shown in Figure 2.1. There are however many other setups available, including other frame types not mentioned here. For a more detailed overview of H.264 see [3].

Transformation and quantisation

H.264 *CODEC* (encoder/decoder) design is visualised in Figure 2.2. The source is sent to a predictor that may do motion prediction on MBs. After prediction, the MBs are transformed and quantified before the encoding. The encoded stream can be transmitted to a decoder that will reverse the process and reconstruct the video stream [3].

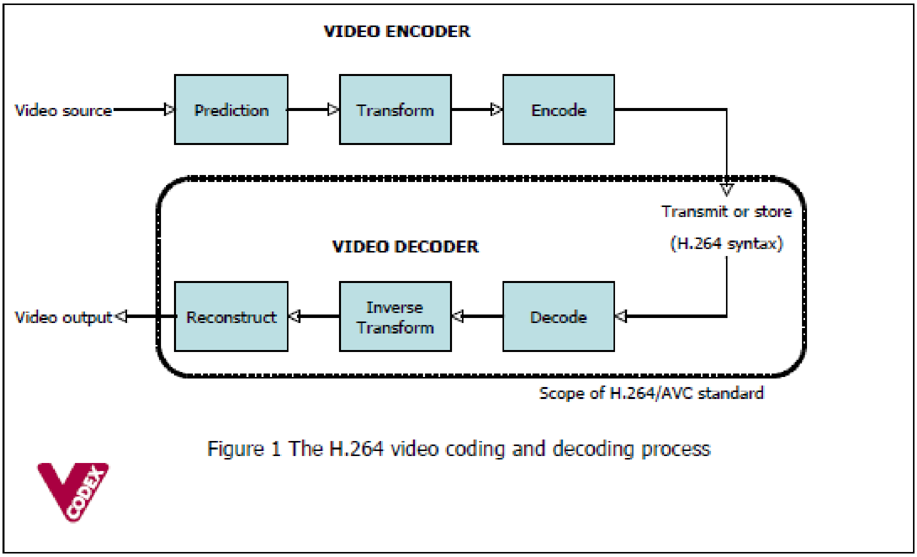


Figure 2.2: Overview of the H.264 video encoder and decoder from [8]

Transformation

Transformation is often done to images during video compression. H.264 uses three types of transforms, 4 × 4 and 2 × 2 Walsh-Hadamard Transform, and a modified version of a 4 × 4 Discrete Cosine Transform (DCT). These block-wise transforms have good memory requirements and work well with block-based motion compensation. The transforms are used to minimize the amount of data required to represent each MB, by separating data into decorrelated components. More information about the transformations can be view in [3]. The DCT is built to have similar properties as the Fourier Cosine Transform, but the DCT used in H.264 is slightly modified for performance reasons [9].

The Walsh-Hadamard transform (WHT) is often used in signal processing and consists of matrices that only contains 1 or -1 and are orthogonal. This makes the WHT fast to compute and can take advantage of redundancy. The WHT have advantages over the DFT as it is much faster to compute. The usage of the WHT is to compress the information into just a few coefficients[10]. The three transformations are applied to all MBs [3].

Quantisation

After transformation, the values are quantized according to equation (2.1), where Y_{ij} is a coefficient of the transformation described above. Here Q_{step} is the quantisation step, and Z_{ij} is the quantified coefficient. This quantisation gives the possibility to reduce the amount of bits required to represent each MB Y by instead representing it with Z and Q_{step}. This will however result in less accuracy when decoding the images. The original value can be reconstructed with some error through equation (2.2)[3].

$$Z_{ij} = \mathit{round}(Y_{ij}/Q_{step}) \tag{2.1}$$

$$Y'_{ij} = Z_{ij} \times Q_{step} \tag{2.2}$$

The quantisation step is an important value as it controls the range available to Z_{ij} and thereby the compression level. It is defined as 52 values with indexes $[0 - 51]$ which is where the QP value comes from. The quantisation step, Q_{step}, doubles every 6 QP values until it reaches 51.

The actual quantisation done in H.264 is a bit more complicated, but it achieves the same thing. The DCT transformation used is modified for efficiency. This introduces a need for post scaling to reduce the error introduced with the modified DCT, and this post scaling is incorporated into the quantisation. Also, as calculation speed is important the quantisation is further modified to remove arithmetically heavy operations such as division or floating point arithmetic. A version used in H.264 can be seen in equation (2.3) with the inverse shown in equation (2.4).

$$Z_{ij} = \mathit{round}(W_{ij}\frac{MF}{2^{qbits}}) \tag{2.3}$$

$$W'_{i,j} = Z_{i,j}V_{ij} \cdot 2^{floor(QP/6)} \tag{2.4}$$

Here W is the output of the modified transformation. In the quantisation, MF and 2^{qbits} comes from an optimisation to change division to bit shift. Also the post scaling is included. Therefore they depend on Q_{step} and the post-scaling (see [3] for more details). In the inverse quantisation V_{ij} is used to inverse the post-scaling done from MF and 2^{qbits}, while $2^{floor(QP/6)}$ causes the post-scaled output to be doubled every 6 QP [3].

2.3 Rate Control for Video Streaming

If the control parameters of a video encoder such as the QP value are kept constant, then the resulting bit rate will fluctuate depending on the content of the video stream.

An encoder with constant parameters will usually require more data transmission when there is high motion or detail in the input, and require less data when the input is static or has a simple structure. This mode of encoding is referred to as variable bit rate (VBR). If a constant bit rate (CBR) is desired, it can be advantageous to adapt the encoding parameters over time with a control system. This system is called a Rate Controller (RC) [3].

The H.264 standard does not demand the use of a RC, but it may be implemented together with the encoder. This is done in order to transmit data under constant bandwidth and ensure stable performance for the receiver. The RC allocates a certain number of bits for each frame and attempts to ensure that the frame produced by the encoder will have the correct size. It can modify the QP value of frames to ensure that the correct bit rate is achieved [4].

The RC calculates the QP value for the next frame with information about previous encoded frames as input. Thus the RC and encoder together make up a feedback loop. Usually the desired bit rate is used as the control signal. If modifying the QP is not enough to ensure sufficiently low bit rate, the RC may instead decide to skip a number of frames in order to stay within its target bit rate [11].

The RC is not bound to act on frame level. An RC may also have control over the quantisation of smaller sections of a frame, often called Basic Units. A Basic Unit can be as small as a Macro Block, or as large as an entire frame. It is also conceivable that an RC works on GOP level, assigning a QP for an entire GOP.

2.3.1 Rate Control Algorithms

There can be control systems of varying complexity, and the goals of these algorithms can be quite different. For example, the algorithm may attempt to balance the fullness of a transmission buffer [12], or follow a target bit rate [11]. Depending on the application the trade-off between image distortion, network usage, and time complexity can vary as well [3]. While some algorithms retrieve the desired QP directly from relationships within their modelling [11, 4], other algorithms use more explicit control theory such as PID controllers [12]. Some researchers have even proposed using convolutional neural networks to predict suitable quantisation parameters for the frames of a video stream [13].

Two things that many proposed controllers seem to have in common is the use of the statistic Mean Absolute Difference, and a Rate-Distortion Model that estimates Rate Distortion Optimisation (RDO). These two concepts are explained in greater detail below.

Mean Absolute Difference

The Mean Absolute Difference, or MAD, is an important statistic that is often included in rate control algorithms [11, 4]. It is a single value which simply describes the difference from one frame to the next. It is of course useful to know the MAD of the next frame relative to the previous when deciding on how to compress it. In H.264 encoding it is usually not possible to retrieve the MAD of a frame prior to performing RDO, so the MAD has to be predicted [11].

One proposed method of MAD prediction is described in equation (2.5) and is a linear prediction based of previous values.

$$\tilde{\delta}_{i,l}(j+1) = a_1 \delta_{i,l}(j) + a_2 \tag{2.5}$$

Here i and l denote the indexes of the current GOP and basic unit, respectively. $\tilde{\delta}_{i,l}(j+1)$ is the predicted MAD for frame $j+1$, $\delta_{i,l}(j)$ is the actual MAD for frame j. The parameters have initial value $a_1 = 1, a_2 = 0$ and are updated iteratively after each basic unit has been encoded [11].

Rate-Distortion Model

Rate-distortion (R-D) is a concept used in information theory that states how much compression can be achieved at the current rate, while keeping the distortion sufficiently low [14]. In rate control it provides a relationship between the statistics of the current frame, the currently available number of bits, and the quantisation parameter [11].

A quadratic R-D model used by several sources is given in equation (2.6). The relationship described is derived from the H.264 RDO scheme, and can be used to retrieve the QP_{step} value, which is related to the QP value through equation (2.7) [11, 4].

$$\tilde{b}_{l,i}(j+1) = c_1 \frac{\tilde{\delta}_{i,l}(j+1)}{Q_{step,l,i}(j+1)} + c_2 \frac{\tilde{\delta}_{i,l}(j+1)}{Q^2_{step,l,i}(j+1)} \tag{2.6}$$

$$Q_{step,l,i}(j+1) = 2^{QP_{l,i(j+1)}/6} \times d(QP_{l,i(j+1)} mod\, 6) \tag{2.7}$$

$$d(0) = 0.625,\ d(1) = 0.6875,\ d(2) = 0.8125,\ d(3) = 0.875,$$
$$d(4) = 1,\ d(5) = 1.125$$

Once again i denotes GOP and l denotes the basic unit, and $\tilde{\delta}_{i,l}(j+1)$ represents a predicted MAD. The value $\tilde{b}_{i,l}(j)$ is the number of bits that has been assigned for the basic unit, and c_1, c_2 are coefficients of the model. $Q_{step,l,i}(j+1)$ is the quantisation step for the basic unit.

2.4 Video Quality Assessment

There are several metrics that can be used to evaluate how well an encoding algorithm works in terms of the video quality. Besides minimising the file size, avoiding blurriness and minimising the amount of artifacts due to lossy compression is beneficial. Below, two metrics for assessing video quality are listed.

2.4.1 SSIM

Structural Similarity (SSIM) is a metric that tries to give a score for how well an image transformation works in this regard. Compared to a simpler metric such as mean square error (MSE), SSIM puts more significance into errors that a human would be more likely to notice, such as errors in larger structures in the image, and the colours where it matters the most [15].

2.4.2 VMAF

Another metric that has risen in popularity over the last few years is Video Multi-Method Assessment Fusion, or VMAF, developed by Netflix. The goal of VMAF is similar to SSIM in that it aims to provide a score of how good a video or image would look to a human observer, but the approach is very different. VMAF uses a Support Vector Machine (SVM) to weigh different quality metrics together, including temporal features and detail loss. The model is trained on a data set aggregated by Netflix which includes many genres of video, with a subjective Mean Opinion Score (MOS) as ground truth. The VMAF scores range from 0 to 100, and correlate well with scores obtained through subjective studies, although it has been observed to slightly over-estimate the quality of the videos [16].

Because the ground truth is based on subjective studies, the VMAF scores can be interpreted through a linear mapping to the scale used in the studies. The scale used had the steps ”bad”, ”poor”, ”fair”, ”good”, and ”excellent”. These steps can be roughly mapped to ”bad” being equivalent to a score of 20, with an increment of 20 points per step up to a score of 100 at ”excellent” quality. For example, a video sequence with a score of 70 would have a visual quality somewhere between ”good” and ”fair” [17].

2.5 Sound-based Event Detection

In video surveillance events have historically been divided into background and foreground events. Information classified as background information is typically predictable and stable, whereas the foreground information includes unexpected and dynamic events. Marco Cristani et al. extended this framework to apply it to audio based surveillance. In their research they found that Gaussian mixture models (GMM) can be used to model the statistics of the background audio, thus making it possible to separate sequences of audio into foreground if they do not fit into the background model [18].

Alternatives to GMM are Support Vector Machines (SVM) or envelope tracking. SVMs can for example be used to classify data into multiple classes, but is then often dependent on large labeled training sets. A sub-class of SVMs are the so called One Class Support Vector Machines, which as the name suggests is only trained to recognise and classify one class of data. This could be used to model the ambient background, and anything not recognised would then be labeled as a foreground event [19]. Envelope tracking is based on the envelope of the audio signal. Here the outline of the signal can be analysed to see if there is any event happening [20].

2.5.1 One Dimensional GMM

GMMs may be used to model images or audio based on the probability that a measured value is within a specific Gaussian distribution. Figure 2.3 displays how three clusters can be modeled by a Gaussian mixture model.

To separate expected sound from unexpected sound a time-adaptive mixture of Gaussians is used to model the behaviour of the signal. In equation (2.8) the relationship between measurement x_t at time t and probability is shown. As can be seen the probability is modeled as a weighted sum of K Gaussian distributions $\mathcal{N}$ where the mean μ_t and variance σ_t are updated for each time frame t.

$$P(x_t) = \sum_{k=1}^{K} w_{k,t}\mathcal{N}\left(x_t|\mu_{k,t}, \sigma_{k,t}\right) \tag{2.8}$$

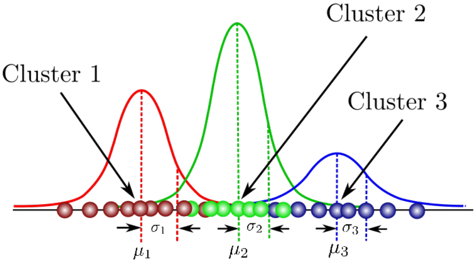


Figure 2.3: One dimensional GMM modelling three clusters [21]

For algorithmic purposes each Gaussian has a rank decided by the weight w_t and standard deviation σ_t of each model. They are therefore ordered in descending order using the ratio w_t/σ_t. The weights for each Gaussian are updated iteratively according to equation (2.9), where $M_{k,t} = 1$ for the matched Gaussian, and $M_{k,t} = 0$ otherwise. Similarly, the probability distribution parameters are updated according to equations (2.10) and (2.11) [18].

$$w_{k,t} = (1-\alpha)w_{k,t-1} + \alpha M_{k,t},\ 1 \le k \le K \tag{2.9}$$

$$\mu_{k_{hit},t} = (1-\rho)\mu_{k_{hit},t-1} + \rho x_t \tag{2.10}$$

$$\sigma^2_{k_{hit},t} = (1-\rho)\sigma^2_{k_{hit},t-1} + \rho(x_t - \mu_{k_{hit},t})^T(x_t - \mu_{k_{hit},t}) \tag{2.11}$$

$$\text{where } \rho = \alpha\mathcal{N}(x_t|\mu_{k_{hit},t}, \sigma_{k_{hit},t}) \tag{2.12}$$

A measured value x_t is said to match a distribution if the value is within n standard deviation from μ of that distribution, see equation (2.13).

$$|\mu_t - x_t| < n\sigma_t \tag{2.13}$$

Only one distribution can be counted as a match for each measurement. If a measurement matches several distributions, the one with the highest rank is chosen. When a distribution is matched, the weights of all higher ranked distributions are summed together to decide if the match should be classified as foreground or background. If the sum is greater than a chosen parameter P, the match is deemed unlikely enough for it to be regarded as foreground. The calculation is done with equation (2.14) where k_{hit} is the index of the distribution which was matched [18].

$$FG = \sum_{k=1}^{k_{hit}} w^{(t)}_{k,i} > P \tag{2.14}$$

The constant P in the equation above is along with the learning rate α and threshold n are parameters which have to be optimised for a task specific environment. P decides how unlikely events have to be to be regarded as foreground, and α determines how fast the weights and distributions can adapt to changes[18].

The one dimensional GMM is able model the distribution of one feature. Cristani et. al. used several frequency based features as input in their work. To do this multiple one dimensional GMMs are created, one fore each feature. All calculations are done for each feature, and if one feature signals foreground the entire system will signal the sound as foreground[18].

It was pointed out by Moncrieff et al. that this setup has two weaknesses. Firstly, the classification between foreground and background contains an implicit assumption that the background does not dominate the audio. They argue that if the audio that makes up the background has a greater total weight than P, sections of background can be classified as foreground. The second problem that is brought up is a problem with the adaption of the model, stemming from the updates dependence on the variance. To address these issues the background/foreground classification was reversed, and update equations were changed, see equations (2.15 - 2.18) [22].

$$FG = \sum_{k=k_{hit}}^{K} w_k < T \tag{2.15}$$

$$\text{where} \qquad T = 1 - P \tag{2.16}$$

$$w_{k,t} = (1 - \alpha M_{k,t}) w_{k,t-1} + \alpha M_{k,t} \tag{2.17}$$

$$\rho = \alpha \frac{\mathcal{N}(x_t | \mu_{k_{hit},t}, \sigma_{k_{hit},t})^{\frac{1}{d}}}{\mathcal{N}(\mu_{k_{hit}} | \mu_{k_{hit},t}, \sigma_{k_{hit},t})^{\frac{1}{d}}} \tag{2.18}$$

2.5.2 Multi Dimensional GMM

A drawback with the one dimensional approach described above is that it assumes the different features are uncorrelated. The approach also classifies events as foreground as soon as there is a foreground match in a single feature. Moncrieff et al. proposed a technique using a single multidimensional model that addresses these issues [22]. In Figure 2.4 an example of two multidimensional Gaussian distributions which models two clusters is seen.

This GMM is defined similarly to the one dimensional. The key difference is that the probability model is defined as a sum of multivariate Gaussian distributions, see equation (2.19). Like before the probability consists of a weighted sum of Gaussians, but the input X_t is a vector of multiple features, and the mean and variance is replaced by a mean vector and a co-variance matrix. See equation (2.20) [22].

$$P(X_t) = \sum_{i=1}^{K} \omega_{k,t} \mathcal{N}(X_t, \mu_{k,t}, \Sigma_{k,t}). \tag{2.19}$$

$$\mathcal{N}(X_t, \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(X_t - \mu)^T \Sigma^{-1} (X_t - \mu)} \tag{2.20}$$

As for the single dimensional version a new observation, X_t, is associated with the distributions with highest rank and where X_t is within n standard deviations from μ, see equation (2.13). Again, the rank is decided by the weight to standard deviation

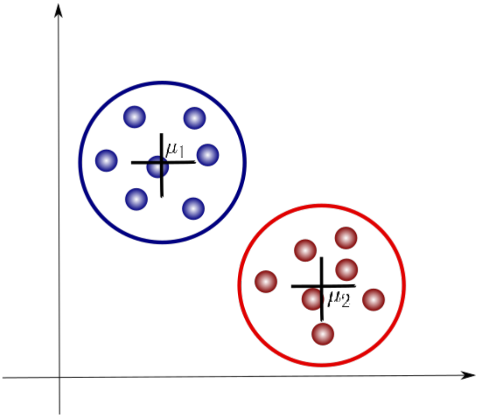


Figure 2.4: Two component Gaussian mixture model of two clusters. Each measurement point has two dimensions. [23]

ratio, w/σ. The matched Gaussian is labeled as either background or foreground depending on its rank relative to the others (see equation (2.17)[22].

The updating of internal parameters is done similarly to the one dimensional version. The sole difference is that the change is now based on the multivariate Gaussian that was matched, which takes all features of the input into account, see equations (2.21 - 2.23).

$$\omega_{k,t} = (1-\alpha)\omega_{k,t-1} + \alpha(M_{k,t}) \tag{2.21}$$

$$\mu_t = (1-\rho)\mu_{t-1} + \rho X_t \tag{2.22}$$

$$\Sigma_t = (1-\rho)\Sigma_{t-1} + \rho(X_t - \mu_t)^T(X_t - \mu_t) \tag{2.23}$$

As mentioned before a measurement X_t is said to belong to a Gaussian according to equation (2.13[22]. To calculate this distance for multivariate distributions it is possible to take advantage of the Mahalanobis distance [24]. The Mahalanobis distance is defined as equation (2.24) where X is the vector, μ is the mean vector of the distribution and, Σ is the co-variance matrix[25].

$$D_m(X) = \sqrt{(X-\mu)^T\Sigma^{-1}(X-\mu)} \tag{2.24}$$

2.5.3 GMM Initialisation

Gerard and Tedenvall suggests that its advantageous to initialise the GMM with Gaussians that already partly model the data, instead of starting the training phase with empty distributions [6]. However, other sources imply that the GMM will perform a type of clustering itself by adaptively moving the means and altering the variances of each Gaussian distribution [26].

In order to initialise the GMM, analysis may be done on the training data to figure out reasonable means and co-variances. One way to do this is to model the features of each audio frame as a point in space, and clustering the points in some way. After clusters have been created a Gaussian distribution may model each cluster [6]. Gonzales et. al. proposed an algorithm which finds an approximate solution in $O(kn)$ time, where k is the number of clusters, and n is the number of samples [27].

2.5.4 Enveloping

An envelope is a signal that approximates the outline of the original signal in some way, thus providing information about its shape over time. This information can be as important as the frequency components or other characteristics of the signal. It can be the main information carrying medium in for example speech and telecommunications. Calculating the envelope can be done in several ways, and it can be based on different features of the signal. Some approaches makes use of the energy content in the signal and filters or smooths this statistic over time to make an approximation of the signals outline. Another technique is based on the Hilbert transform, and is extremely effective on narrow-band signals but suffers greatly when the spectrum of the signal is wide and varied. In Figure 2.5 an example of an Hilbert envelope is shown, showcasing its strengths and weaknesses, but also providing a picture of what enveloping aims to achieve. Namely an alternative signal that represents the overall shape of the underlying signal of interest [20].

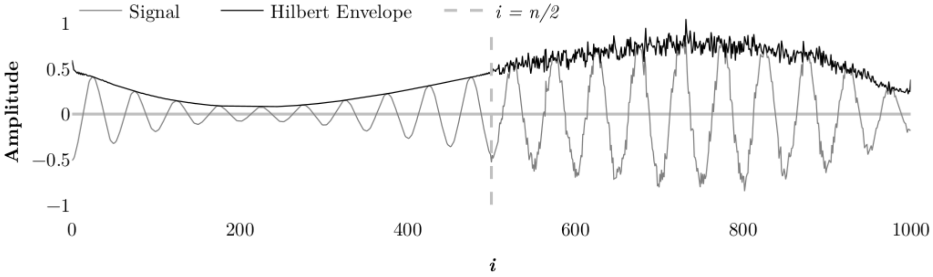


Figure 2.5: The Hilbert envelope applied to a signal with varying complexity.

Power Envelope

Marzinzik and Kollmeier test the potential of tracking power envelope dynamics for speech pause detection. The power envelope is based on frequency features, taking the power in different parts of the spectrum into account. The total energy $E(p)$ at frame p is calculated using equation (2.25). Here $E(p)$ is the total power envelope and $X(p, \omega_k)$ is the spectral component for frequency ω_k at frame p. Finally, m is the highest frequency calculated by spectral analysis [28].

$$E(p) = \sum_{k=1}^{m} |X(p, \omega_k)|^2 \tag{2.25}$$

RMS Envelope

RMS stands for Root Mean Square, and is a statistical measure related to the standard deviation, given that the signal has zero mean. The RMS of a signal $x_1, x_2, ..., x_n$ is calculated according to equation (2.26) [29]. If the calculations are done iteratively or on frames of the signal this can then be used to approximate the envelope of the signal, as the RMS will increase or decrease depending on the energy content of the signal.

$$x_{RMS} = \sqrt{\frac{1}{N}(x_1^2 + x_2^2 + ... + x_n^2)} \tag{2.26}$$

2.5.5 Audio Features

There are several features of the audio signal that could potentially be used as input to the algorithms explained above. Cristani et. al. suggests calculating the power spectral density for each audio frame, and dividing its spectrum logarithmically up to half the sampling frequency [18]. He bases this decision on article [30], which also suggests many other features such as Spectral centroid, and Spectral Flux.

Moncrieff et. al. uses different feature sets throughout different publications. In one case they makes use of the wavelet transform and uses the mean wavelet energy for 7 frequency sub-bands as features. In another they use a combination of MEL-spectrum coefficients, the zero crossing rate, and mean energy of some wavelet sub-bands [31].

Power Spectral Density

The power spectral density is strongly related to the Fourier transform of a signal, and can be calculated simply by squaring the magnitude of each Fourier bin, as in equation (2.27).

$$P(f) = |X_f|^2 \tag{2.27}$$

Spectral Flux

Spectral flux measures the change in the shape of the power spectrum over time. This is done by calculating the squared difference between the normalized magnitudes of the power spectra of successive frames [30].

MEL Spectrum

The MEL spectrum is designed to correlate more closely to the human interpretation of frequency changes than the standard linear scale. A human would hear more difference in lower frequencies than in higher frequencies [32]. The Mel-frequency cepstral coefficients (MFCC) algorithm performs a mapping between the spectrum outputted by the DFT and the MEL scale. It does this using a filter bank consisting of triangular filters. In the low frequencies the filters are spaced linearly, but in higher frequencies the width and spacing is widened logarithmically [33]. The final MFCC coefficients are then produced by decorrelating the output of the filter bank using the discrete cosine transform, DCT [34].

To get a sense of the cepstrum fluctuations over time one can calculate time derivatives of MFCC. The first derivative can be calculated by taking the difference between MFCCs from consecutive frames, or by weighting together the difference of several frames over a longer window as in equation (2.28). Here c_i is the MFCC for frame i, and Θ is the derivative window length. To get the second time derivative the process is simply repeated [34].

$$d_t = \frac{\sum_{\theta=1}^{\Theta} \theta(c_{i+\theta} - c_{i-\theta})}{2\sum_{\theta=1}^{\Theta} \theta^2} \tag{2.28}$$

Chapter 3
Method

The purpose of this project is to prepare the RC for upcoming events by observing the sound environment. This may increase the quality of the encoded video stream by having fewer dropped frames, more stable QP, and better frame quality. The problem is split into two main parts, the RC preparation and the audio event detection. This can be summarized by the following two research questions:

- May the performance of a rate controller be improved by introducing a signal that warns if motion is about to increase?
- May audio background/foreground classification be used to create a warning signal for a rate controller?

The audio classification part can be separated further into the following sub-questions:

- May foreground/background classification using a one dimensional GMM be used to improve video rate controller performance?
- May similar classification using multi-dimensional GMM be used for the same purpose, and does it yield better results?
- May envelope tracking algorithms yield similar results?
- Is it possible to separate foreground events based on the effect they have on the image encoding, allowing the system to give a more accurate warning to the rate controller?

An overview of the proposed solution can be seen in Figure 3.1. Seen in the overview are three blocks which this project focuses on. Block A includes the entire developed system. It takes an audio signal as input and updates parameters of the RC as output. The audio classification reside in block B. Lastly block C consists of the

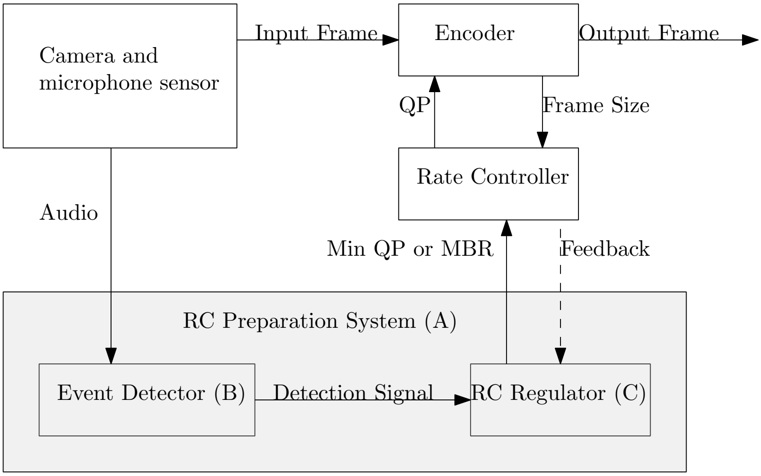


Figure 3.1: An overview of the system implemented in this project.

RC regulator, which translates a detection signal into updates for the RC. Also seen in the overview is that the encoder together with the RC creates a feedback loop.

This chapter has four sections. First, it is explained how the surveillance footage has been recorded to create a data set. Second, two implementations of part C are proposed in Section 3.2. These are evaluated using a manually created detection signal. Third, two types of algorithms for creating the detection signal from audio are presented in Section 3.3. These implements block B. Finally, in Section 3.4 block A as a whole is evaluated using the best variants of both part B and C together.

3.1 Data

In order to train, optimise, and evaluate the algorithms it is necessary to have relevant data. No large open data sets providing surveillance videos with sound could be obtained. Therefore, all training and test data had to be recorded. For the purposes of this project, clips with people or objects abruptly entering the camera's field of view are of most interest. Since there is a lot of noise present outdoors it is decided to record indoor videos. To record videos a surveillance camera with an integrated omnidirectional microphone is used.

3.1.1 Recording Settings

The RC is designed to use the lowest feasible compression and keep the image quality as high as possible, given a maximum bit rate. Therefore the RC only increases the QP or drops frames when necessary. This results in constraints on the recording and network settings needed to put the system in a situation where this behavior happens.

First of all the bit rate has to be restricted to a low but reasonable limit. Further, the length of a GOP changes the bit rate as it affects the ratio between the larger I-frames and the smaller P-frames. The frame rate (fps) also affects the bit rate. Compared to a low fps, a high fps increases the bit rate as more frames of both types have to be sent every second. However, as less time elapses between frames the scene will not change as much between P-frames, possibly reducing their sizes. The bit rate will be harder to regulate when using a low frame rate, as the size of P-frames is more unpredictable.

Drops also heavily depend on how sudden scene changes happen and therefore the person entering the frame has to walk relatively fast. The system is also sensitive to how complex the patterns on their clothes are where a more complex pattern will more likely result in dropped frames.

It is decided that all clips are recorded in 10 fps and with a GOP length of 32. This to set the camera in its most vulnerable setting were the methods tested may make a difference. Also, it is a realistic frame rate that end users are likely to use [35].

3.1.2 Recorded Videos

As the purpose of the project is to evaluate the ability to prepare the RC before the object is visible using audio information, it is important that the videos used contain reasonable scenarios. It is decided that three different activities provide a good basis for the algorithms, both in training and evaluation. The scenarios are listed below.

- One person walking
- One person running
- Two persons walking and talking to each other

In all scenarios the persons come from out of view and enter the field of view abruptly and close to the camera. The running person enters more abruptly than the walkers due to faster movement. A faster movement should generate larger spikes in bit rate. The movement in the talking scenario is about as fast as in the walking scenario. The difference is the frame complexity, which should be greater due to a larger portion of the image being affected. In all cases the audio event detectors should pick up on the footsteps or speech. An example of how a scenario can look is shown in Figure 3.2. In the figure a person is moving from left to right. Only some part of their path

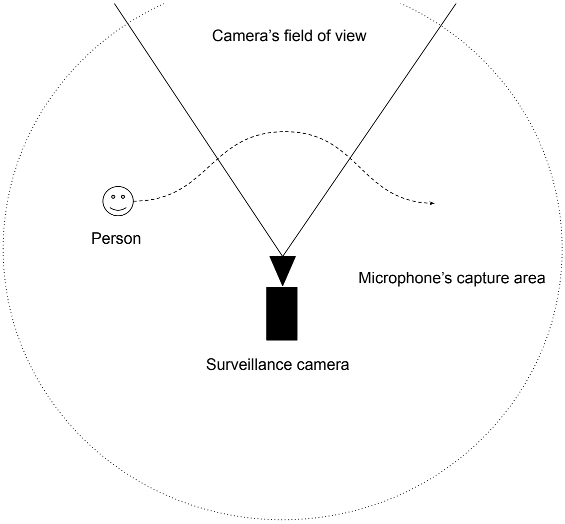


Figure 3.2: A depiction of how a person may enter and leave the field of view, while being heard the entire time.

is within the visual field of view, but the person may be heard both before and after being seen.

All recorded clips and their information are listed in Table 3.1. During the recording of the video clips a variable bit rate is used, meaning that the QP is static. The names of the clips correspond to the scenario they represent. For example, the clip *Walking1* has a person walking past the camera. Below, the different types of clips are explained.

Scenario Clips

In Table 3.1 the uppermost 15 clips are what is referred to as scenario clips. These are relatively short and contain the activities explained above. The table lists the idle bit rate for the scene, and the minimum bit rate ceiling necessary to avoid all frame drops. As can be seen in the table there are slight differences in the minimal required bit rate to remove frame drops.

From each scenario two clips are used for audio training, one for audio testing and two are saved for final evaluation of the entire system. This can be seen in the usage column. The clips marked with Audio Train are mainly used for optimisation of parameters of detection algorithms. The clips marked with Audio Test are exclusively used for testing of these algorithms. The RC Test clips are used for testing the RC performance both by itself and in combination with audio detection.

Warm-up Clips

There are also two clips marked as warm-up clips. These are only used by the audio algorithms and are run before the algorithms are used on scenarios. One is used as warm-up during training, and another is used during testing. The warm-up clips contain mostly ambient background noise, but also brief periods of activity, including walking, running and talking.

Quality Clips

Finally, at the bottom of Table 3.1 there are six clips marked as Quality Test. They are recorded with lower than normal compression, and are used for testing differences in encoding quality depending on the RC behaviour. Due to their low compression, they can be re-encoded a second time with higher compression. This gives a similar result as if they would have been encoded at higher compression from the beginning. They are also shorter as the sizes would otherwise be too large, and the re-encoding would be too time consuming. Just like the scenario clips, these contain the scenarios explained earlier.

Table 3.1: The video clips used throughout the project along with their information

Name	Length (s)	QP	Scene bit rate kbit/s	Minimal required bit rate kbit/s	Usage
walking1	120	28	220	250	Audio Train
walking2	90	28	220	300	Audio Train
walking3	90	28	220	350	Audio Test
walking4	90	28	210	350	RC Test
walking5	90	28	220	700	RC Test
running1	90	28	235	300	Audio Train
running2	90	28	220	250	Audio Train
running3	90	28	220	800	Audio Test
running4	90	28	220	800	RC Test
running5	90	28	210	700	RC Test
talking1	90	28	200	2100	Audio Train
talking2	90	28	200	1850	Audio Test
talking3	90	28	200	1150	RC Test
talking4	90	28	210	500	Audio Train
talking5	90	28	210	1400	RC Test
long4	1200	28	-	-	Test Warm-up
long6	1200	28	-	-	Train Warm-up
quality_walk1	30	11	-	-	Quality Test
quality_walk2	30	11	-	-	Quality Test
quality_run1	20	11	-	-	Quality Test
quality_run2	20	11	-	-	Quality Test
quality_talk1	30	11	-	-	Quality Test
quality_talk2	30	11	-	-	Quality Test

3.1.3 Labelling and Audio Evaluation Framework

In order to evaluate how well the audio event detection algorithms perform the video clips have are labelled based on their audio. This is done by carefully listening to the sound in each clip and writing down the start and end of the event. Event can for example refer to the sound of a walking person in the scenario. All audio samples between the start and end of the event are marked as ground truth detection.

When a clip has been labeled it is possible to measure how well the algorithms output aligns with the labels. Their overlapping can be summed up in terms of a confusion matrix, such as the one in Table 3.2. With the confusion matrix framework in place the statistics accuracy, sensitivity, and specificity can be defined. Accuracy tells us the percentage of correct predictions including both positives and negatives, see equation (3.1). Sensitivity tells us the percentage of actual positives that were correctly predicted, see equation (3.2). Finally, specificity measures how much of the label negatives that are predicted correctly, see equation (3.3).

Table 3.2: Confusion matrix showing the possible outcomes from the detection step.

	Predicted Foreground	Predicted Background
Label Foreground	True Positive, TP	False Negative, FN
Label Background	False Positive, FP	True Negative, TN

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.1}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{3.2}$$

$$Specificity = \frac{TN}{TN + FP} \tag{3.3}$$

3.2 Rate Control Regulation

The RCs job is to regulate the encoding quantisation of the encoder through the QP value. A problem with rate control in a real time system is that it is only able to adjust QP based on previous frames. It is usually not possible to know beforehand how much motion the next frame will introduce. This paper aims to solve this dilemma by analysing the sound and send a signal to notify the RC. The warning signal may give the RC time to prepare for upcoming changes. Two methods for regulating the RC are presented. The first is to restrict the minimum QP allowed by the RC, the second is to loosen up the maximum bit rate restriction.

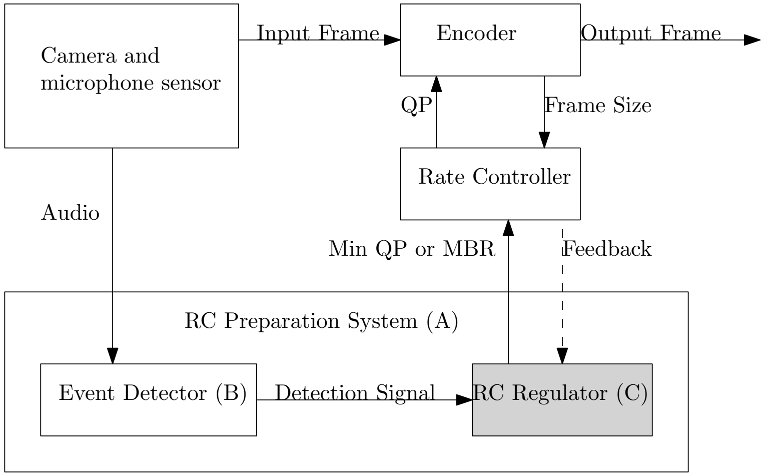


Figure 3.3: An overview of the system, with the RC regulator highlighted.

3.2.1 Used Rate Controller

The RC in this project is used for real time video streaming in a surveillance camera. The specific use case sets high timing demands on calculations. The RC takes the size of previously encoded frames as input and produces a new QP as output, resulting in a closed loop system together with the encoder. Its reference signal is a target bit rate. Some specific behaviours of this controller are explained below.

The QP value and its change is restricted in two major ways. First, the QP has a lower and upper bound. If the RC ends up with a QP outside of these bounds the value will simply saturate and rest at the limit. Secondly, the change of QP is restricted. Any change larger than the restriction is clamped, limiting the rate of change in the systems control signal between frames.

The real time environment and network restrictions may introduce forced frame drops to stay within a target bit rate. If forced to drop frames the RC outputs the number of frames to be skipped. The system that determines if frames need to be dropped is separate from the QP control system resulting in two independent systems. The first calculates a desired QP based on frame sizes. The second drops frames if the bit rate produced is too large. This is done by tracking the bit rate over the last second.

The current RC is a complex piece of software. It is difficult to introduce any internal changes to the RC without introducing unwanted changes. Therefore it is decided to let the audio detection signal only affect already existing parameters in the RC. This will however not result in an optimal RC for the job. The RC used is not optimized for having more control signals. Signals that are chosen to be manipulated are minimum QP and maximum bit rate.

3.2.2 Preparing the Rate Controller

The RC regulation schemes are designed with the assumption that a binary audio detection signal can be created. A zero represents no detection and a one represents a detection. This binary signal is sent into a rate control regulator. By analysing the input the rate control regulator decides on settings for the RC. It then manipulates the state of the RC and may therefore prevent frame drops. See Figure 3.3 for architectural overview. This design is used to not have to change functionality inside the RC, but instead only affecting parameters. In the sections that follows two strategies for the rate control regulator are presented.

Increasing the Minimum QP

A first strategy is to restrict the minimum bound of the QP value when an event is detected. This will result in more compression of the video stream before the motion enters the video. This results in a lower bit rate and more bits available for later use. When motion does enter the video stream the QP value may have to be increased further. As the bit rate for the last few frames have been lower there are some excess bits. This excess may be used by frames that takes up a larger amount of bits during the event. This in return may prevent eventual frame drops as the over all bit rate is lower. The drawback is that the image quality will suffer before the event takes place. Also depending on the severity and duration of the event the minimum QP may be increased more than necessary.

The QP can have 52 different values according to the standard. However, not all values are used in the RC. The available range can be described by equation (3.4).

$$QP_{min} \leq QP \leq QP_{max} \tag{3.4}$$

All videos are recorded using $QP = 28$, and it is decided that an increase of QP_{min} from 28 to 34 is appropriate. As the QP is restricted to be within this range, it forces the RC to increase the QP for the next frame. This results in about half the amounts of bits required for each frame.

Increasing the Maximum bit rate

This approach will increase the maximum bit rate allowed for the RC during a brief amount of time. If there is low amount of motion the QP value will be as low as possible in order to have as high quality as possible. By increasing the bit rate, the RC will have more bits for each frame. When large amounts of motion is introduced and frame sizes increase, the RC may avoid to drop frames due to the extra space available. When the event is over, the maximum bit rate can be decreased to the original size. One benefit of this is that it may give higher quality when the event takes place in the video stream. When calculating frame sizes the RC currently tries to fulfill equation (3.5) where b_{out} is the calculated bit rate and b_{max} is the maximum allowed bit rate. Equation (3.6) shows how the bit rate is calculated where $b_{frame,n}$ is the number of bits allocated for each frame, f is the frame rate, and N is the number of frames used in the calculation of bit rate.

$$b_{out} < b_{max} \tag{3.5}$$

$$b_{out} = \frac{\sum\limits_{n=1}^{N} b_{frame,n}}{\frac{N}{f}} \tag{3.6}$$

Therefore by increasing the maximum bit rate, each frame is allowed to be larger and still fit within the target bit rate. In this method it is decided that b_{max} is doubled by the regulating scheme.

3.2.3 Potential RC Feedback

A potential improvement of the preparation of the RC is to use feedback, represented by the dashed arrow in the overview, seen in Figure 3.1. From the RC it is possible to get information if there was more motion in the image. This information could be used to further improve the preparation of the RC. Either by choosing to not listen to an audio detection or to know when to stop preparing the RC. This type of analysis assumes that the audio detection is able to generate different signals depending on different types of audio events.

3.2.4 Evaluation

To evaluate the regulation schemes their performance is measured in two ways. First of all the RC behaviour under the influence of the warning signal is studied, and compared to an unaltered version. Second, the video quality when using minimum QP regulation is compared to an unaltered RC. These experiments will be conducted based on a ground truth detection which is created manually for each scenario.

Rate Control Behaviour

As the bit rate is heavily restricted it is expected that the RC increases QP and perhaps performs a few frame skips when unexpected movement happens. Ideally, the audio preparation may mitigate this behaviour. Therefore two interesting metrics when investigating the success of the rate control preparation is peak QP and mean QP, for the duration of a clip. Further, the number of drops will of course be an important statistic.

The bulk of the frame drop and QP based metrics is evaluated based on an emulator. The emulator uses a simulated encoder that calculates frame sizes based on the output of the RC and the size of the original frame in the clips referenced in Table 3.1. The simulator bases these calculations on the rule of thumb that an increase of 6 QP corresponds to halving the bits used. See Chapter Background Theory on H.264 quantisation for more details on this relationship.

Encoded Video Quality

There are other important metrics of how successful the video compression is beyond the signals within the control system. It is also of high importance that the streamed video looks good to the observer, is similar to the source video, and does not contain artifacts. In Chapter Background Theory two metrics for evaluating video quality are introduced, namely SSIM and VMAF. These metrics will be used to assess the benefits and costs of raising the minimum QP value of the rate controller.

To perform this test, short but high quality clips are re-encoded using a system consisting of the same RC and audio system used for earlier simulations, but with a real encoder. The clips are listed in Table 3.1 and marked with $QualityTest$.

In order to keep the videos synchronised, frame drops are represented by an empty P-frame. The SSIM and VMAF scores can then be computed by comparing the original high quality video with the newly encoded clips. This allows for comparison of the video qualities when regulating the RC and when not.

3.3 Audio Event Detection

To perform audio detection a definition is needed of an event. A surveillance camera can be placed in a wide variety of environments, which could have more or less audible noise. Under these circumstances an event is defined as some noise that comes from an object that may cause large motion in the video stream.

This section introduces a number of algorithms implementing block B from Figure 3.4. Two methods are used, a GMM based and a Envelope based. The GMM based is separated into three unique implementations. A single dimensional GMM (1D-GMM), a multi dimensional GMM (MD-GMM), and a multi dimensional GMM with updated features (MD-GMM-UF). Envelope tracking also has different implementations, namely the Power envelope and the RMS envelope.

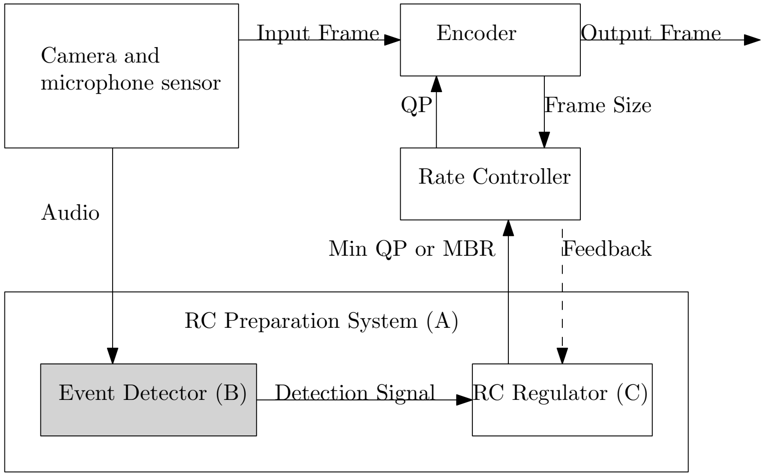


Figure 3.4: The system overview, with the audio event detector highlighted.

3.3.1 Audio Event Detection Pipeline

The event detection can be separated into a pipeline with four steps including framing, feature extraction, classifying, and post processing. The event detection block from Figure 3.4 is then expanded to the content of Figure 3.5. Each algorithm implements the four steps slightly differently, but they all follow the steps to some extent.

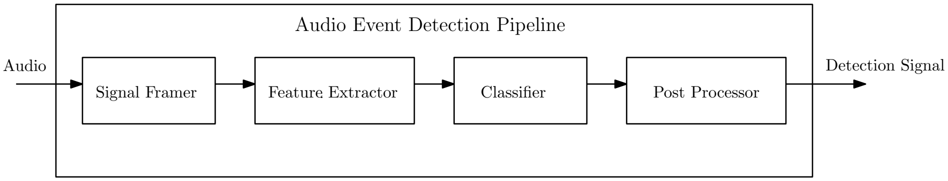


Figure 3.5: A figure showing the audio detection pipeline.

Framing Audio

The first step when analysing the audio signal is to separate it into frames. In the different methods used different frame lengths have been chosen, but they are all in the range of 512 to 4096 samples. The frequency of the microphone is 48000 Hz which means the window lengths are equivalent to around 10 to 85 ms.

For the purposes of this project the frames divide the signal sequentially without overlap. The use of overlapping windows and windowing functions was briefly tested. Windowing, with for example the Hann window, can be used to avoid spectral leakage [36]. It was however found that this method had limited benefit for this application, so the simpler alternative is used in all algorithms presented in this chapter.

Feature Extraction

The next step in the pipeline is feature extraction. Here each frame is processed in order to bring out information that the algorithms can use to classify the frame as either background or foreground. The features used defines the capabilities of the classification algorithm. Features that can differentiate sound distinct makes it possible to do categorisation with higher accuracy. A common way to extract features from an audio signal is to use the Fourier Transform calculated by the FFT. The FFT is used in all GMM versions as well as in Power Envelope. However, the implemented algorithms differs slightly in the features used.

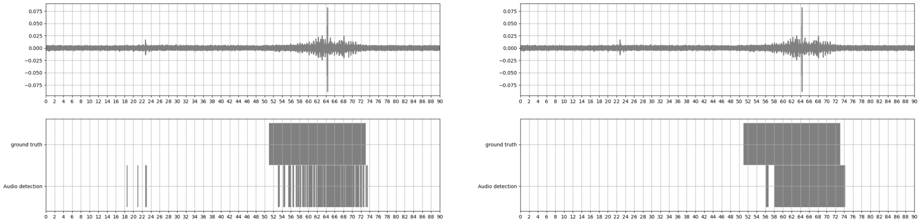
Classification

This step consists of doing the actual classification and analysis of the sound signal. The output from this step is a binary signal representing background (0) and foreground (1). The classifier differs a lot between the different algorithms. The envelope algorithms track their envelope and implement a type of threshold. The GMMs update their distributions and draws conclusions on how well the features match the model.

Post processing

For the audio detection signal to be useful as a warning it needs to be stable. A short detection of a small event, or a gap in the detection of an event may introduce undesired behavior. To counter this post processing is used. With this outlier removal can be done to remove small unwanted detections or stitch together gaps in the detection signal.

One method of post processing is k-step look-ahead (KSLA) described in detail by [37]. This algorithm iterates over the output signal of the classifier. When a one is



(a) Without post processing.

(b) With k-step look-ahead activated.

Figure 3.6: A comparison of audio detection on the clip $Walking2$ with and without using k-step look-ahead with $k = 10, p = 5$.

detected it looks ahead k steps to find the next one and converts all zeros to ones in between. The drawback of this method is that it introduces a k step delay, as it can not know the final classification of a sample until enough future samples are available [37].

The KSLA is only able to fill in gaps between detections. To also be able to remove outliers the algorithm is updated. The update adds the requirement that within the k-step window there must be at least p ones detected to return a one. If there is enough ones within the window, all samples up to the latest one are converted to ones. However, as long as there is not enough ones within the interval, each successive classification will be converted to a zero. The algorithm can be viewed in Algorithm 1. It has two parameters, where k is the window size and p is the number of ones needed for the algorithm to convert zeros samples to ones. It also internally stores the signal and *last_index* which finds the last index of the searched sample, in this case a 1.

In Figure 3.6 the effects of using KSLA with outlier removal can be observed. The example is taken from the clip $walking2$. The scenarios with a single walking person are quite subtle and therefore there are often a lot of small gaps in the detection. As can be seen in the figure the k-step algorithm removes most of the gaps, and all of the short detections. Notice that the post processing introduces a slight delay corresponding to the k-parameter.

Algorithm 1: K-step look-ahead, remove small

```
Data: Inputs a new signal measurement and returns the measured value k
      steps before
input  : $s_i$
output: $s_{i-k}$
Settings: k, p
S[i] = $s_i$;
last_converted = max(last_converted-1, $i - k$);
if $S[i - k] == 1$ then
    if sum(S) $>$ $p$ then
        last_converted = S.last_index(1);
        for j in range(i-k, last_converted) do
            S[j] = 1;
        end
    end
    else if last_converted == i-k then
        S[i-k] = 0;
    end
end
return: S[i-k];
```

3.3.2 GMM-based Background Modelling

Three versions of GMMs are implemented to evaluate the audio detection capabilities. These algorithms were chosen as one of the main focuses based on two reasons. Firstly, the GMMs do not require a large data set for training. This is a valuable property as access to large labeled data sets is lacking in this project. Second, previous research shows a potential of high accuracy when used with a similar purpose [6].

All the GMM algorithms in this project build upon the implementation proposed by Moncrieff et. al. [22]. The algorithms differ from Moncrieff et. al. by updating weights for all distributions every iteration, whether the distribution represents the measurement or not. This is done according to equation (2.9). It makes it possible for Gaussians to transfer back to foreground from background if they have been inactive for a long time. Another difference is that the ranking of Gaussians is solely based on the weight w of each Gaussian, instead of the ratio between the weight and standard deviation w/σ. If a measurement can not be represented by any existing Gaussian in the mixture, a new distribution is created. This distribution replaces the lowest ranked Gaussian, and has its mean at the same position as the measurement.

For both multidimensional variants another change is introduced. The learning rate α is separated into two parameters, α_{weight} and α_{rho}. The first represents the rate of change of the distribution weights and the second the rate of change of the mean and correlation matrix. The new updated equations can be seen in equations (3.7 - 3.10), where $M_{k,t} = 1$ if distribution k was hit at time t. Further, w, μ, and Σ represent the weight, mean, and covariance matrix of the distribution. X represents the feature vector. The changes introduced here can be compared to the original behaviour shown in equations (2.21 - 2.23) in the Background Theory chapter.

$$\omega_{k,t} = (1 - \alpha_{weight})\omega_{k,t-1} + \alpha_{weight}(M_{k,t}) \tag{3.7}$$

$$\mu_t = (1 - \rho)\mu_{t-1} + \rho X_t \tag{3.8}$$

$$\Sigma_t = (1 - \rho)\Sigma_{t-1} + \rho(X_t - \mu_t)^T(X_t - \mu_t) \tag{3.9}$$

$$\text{where} \qquad \rho = \alpha_{rho}\mathcal{N}(X_t, \mu_k, \Sigma_k) \tag{3.10}$$

As discussed in the Background Theory, the distance in terms of standard deviations is not a straight forward calculation for a multi variate distributions. Moncrieff et. al. does not state how this calculation is done. In this method the Mahalanobis distance is used to calculate the distance between the feature point and each distribution in the multivariate Gaussian mixtures.

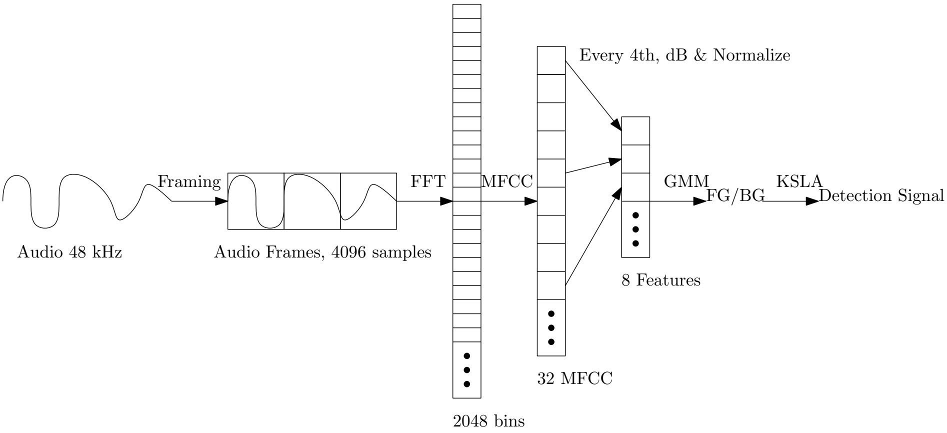


Figure 3.7: An overview of the pipeline used for the 1D-GMM.

1D-GMM

The first algorithm implemented is based on the one dimensional GMM proposed by Moncrieff et al, and with the changes explained above. An overview of the pipeline used by the 1D-GMM can be seen in Figure 3.7. This version uses a frame size of 4096 samples.

For each frame the FFT is used to calculate its frequency content. The frequency spectrum is then converted to 32 MEL cepstrum components. Out of these 32 components every fourth component is used as a feature. In the end the 1D-GMM uses eight features. The amplitudes of these eight features are converted to decibel and normalized with the largest amplitude of the frame as reference point. The reason for using every fourth feature instead of creating eight features to begin with is that the features may be less correlated this way. The 1D-GMM can not model correlations between features as it models each feature separately.

For post processing this algorithm uses KSLA with outlier removal, with ten steps and minimum five detections within those steps.

MD-GMM

The second algorithm is based on the multidimensional GMM, also proposed by Moncrieff et. al. Its pipeline can be seen summarized in Figure 3.8. For this algorithm the frame sizes are 4096 samples.

In contrast to the one dimensional GMM the multidimensional variants can model correlation between features. Therefore the MD-GMM only generates 8 MEL bands and uses all of them as features. The MEL-band magnitudes are converted

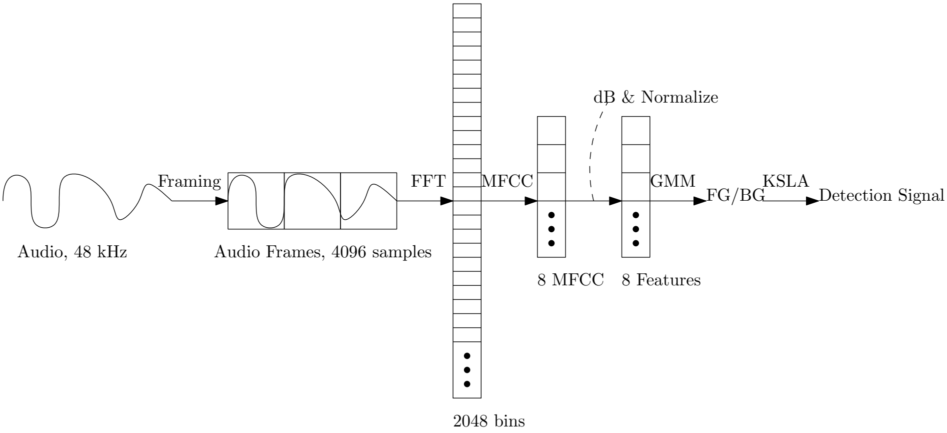


Figure 3.8: An overview of the pipeline used for the MD-GMM.

to decibel with the maximum MEL-value of the frame as reference. This is done to make the GMM model the relation between MEL-band rather than just the energy and therefore become less sensitive to volume increase.

An algorithmic change done to the MD-GMM is that when a distribution is replaced it is not based on the weight w of the Gaussian. Instead it is based on the number of iterations since the last match of that Gaussian. This is to prevent that only a few distributions are replaced over and over again when initialised with small weights. A counter is added to each Gaussian which increase each time the Gaussian is not matched and resets to 0 when it is matched. When a new Gaussian needs to be created the Gaussian with the highest number and therefore longest time since used will be replaced[6]. It is known that this may cause a heavy distribution to be replaced if long enough time has passed since it was used, but this can be prevented by having many Gaussians to choose from.

For post processing the MD-GMM uses KSLA with ten steps and minimum five detections with those steps.

MD-GMM-UF

The multidimensional GMM with updated features (MD-GMM-UF) is constructed to generate higher resolution in the detection in terms of which distributions are matched. It can be seen as an updated version of the MD-GMM, as they share much of their architecture. Despite this, it has unique alterations in all parts of the pipeline. An overview of its pipeline can be seen in Figure 3.9.

In contrast to the other two GMM variants, the MD-GMM-UF uses a frame size

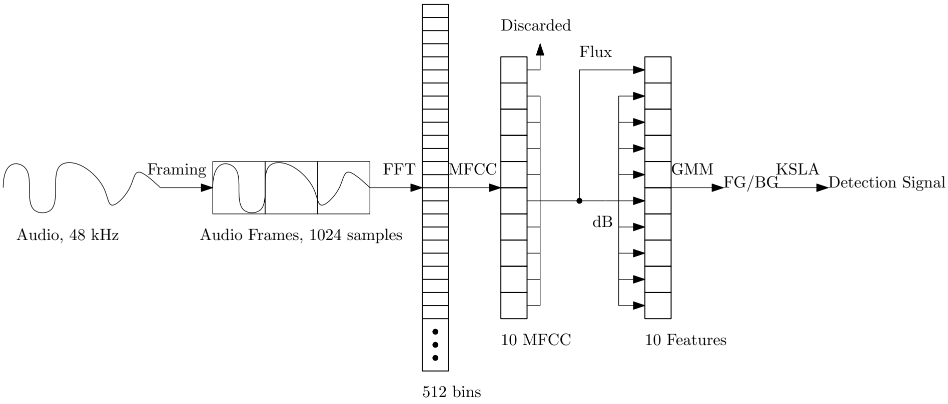


Figure 3.9: An overview of the pipeline used for the MD-GMM-UF.

of 1024 samples. It also uses an updated feature set. The updated feature set consists of the 9 MEL spectrum coefficients, and one spectral flux feature.

For each frame it generates ten MEL bands where the band containing the highest frequencies is discarded due to being noisy. The MEL bands are converted to decibel and uses zero as reference point. To complement the MEL bands, spectral flux is calculated over the nine MEL bands used. Lastly, the frame size is smaller which generates a noisy signal. To counter this the features are low-pass filtered through a first order IIR filter with attack time of 50 ms.

At the classification step this algorithm differs from the MD-GMM by using the original system of weight based replacement of distributions. For post processing this variant uses KSLA with twenty steps and minimum seven detections within those steps.

Model Initialisation and Clustering

In order to be used as classifiers the GMMs need to have a collection of well tuned distributions that together model both the background and the foreground sounds. The GMMs are adaptive which means that the distribution parameters change over time. However, there is still a question about which state they should have at initialisation. As mentioned in Background Theory it could be beneficial to initialise the GMMs in a way so that their distributions are well suited to model the input data already from the first iteration.

It is possible that a well chosen initial state increases the likelihood for good distributions adapted by the algorithm later on, or that a bad initial state makes it harder for the adaptive algorithm to find a good local optimum. To evaluate if clustering should be used as initialisation or not the clustering algorithm proposed in [27] was implemented. An alternative to clustering is to simply let the GMM algorithm run on a relatively long audio clip of $10 - 20$ minutes, which contains events similar those it should classify and let the algorithm choose the distributions itself. In this case the starting state of the Gaussian distributions is zero mean, and a high variance. For the multivariate versions the co-variances are initially zero.

After some experimentation it was concluded that both of these approaches are quite similar in performance. When the GMM warms up on the long training clip, it very quickly replaces its initial distributions with new ones with more reasonable parameters. This means that it does not take a lot of training for a GMM to catch up to the clustering through its adaptive process.

An argument for pre-clustering is that it is quite fast, and initialising the GMM this way means it is more or less ready to be used as a classifier right away. An argument against it is that in the real world this classifier has to work hours if not days or weeks after its initialisation. The pre-clustering will then be so distant that its effects will be very marginal. Because of good results without pre-clustering it is decided that the initialisation is done by warming up on a long audio clip.

3.3.3 Audio Detection Using Envelope

The other method of doing audio detection is by looking at the signal envelope as explained in Section 2.5.4. The idea is to try to find when the envelope has an unusual increase in energy which may suggest that something is creating sound near the camera. This in turn may suggest that there is something about to enter the camera's field of view. There are several ways of approximating the envelope. The two methods used in this paper are *power envelope* and *RMS Envelope*. Both are used together with an envelope tracking algorithm to get an audio detection signal. This algorithm produces less sporadic detections which is why no post processing is used.

Tracking of Envelopes

The algorithm for audio detection with envelope is inspired by Marzinizik and Kollmeier who used tracking of power envelope for speech pause detection. It has been adapted for audio event detection instead. The idea is to approximate the envelope E of the audio signal and store a maximum E_{max} and minimum E_{min} value of E. If the delta Δ calculated in equation (3.11) between E_{max} and E_{min} is greater than a threshold η there is evidence of activity. E_{max} and E_{min} are updated according to equation (3.12) and (3.13).

$$\Delta(n) = E_{max}(n) - E_{min}(n) \tag{3.11}$$

$$E_{max}(n) = max((1 - TAV) \cdot E_{max}(n-1) + TAV \cdot E(n), E(n)) \quad (3.12)$$

$$E_{min}(n) = min((1 - TAV) \cdot E_{min}(n-1) + TAV \cdot E(n), E(n)) \quad (3.13)$$

Here TAV is calculated from equation (3.14) and t is the attack time. In general t is quite high to make the filter have a slow response. The decay/raise of E_{max}/E_{min} will reset Δ when the event has passed.

$$TAV = 1 - e^{\frac{log(\frac{0.9}{0.1})/f}{t}} \quad (3.14)$$

The result will look like Figure 3.10. Each time E is greater than E_{max}, E_{max} will increase, and each time E is less than E_{min}, E_{min} will decrease. Otherwise they will converge towards E. When there is more energy in the sound it will increase E and E_{max} faster than E_{min} which is how an event can be detected.

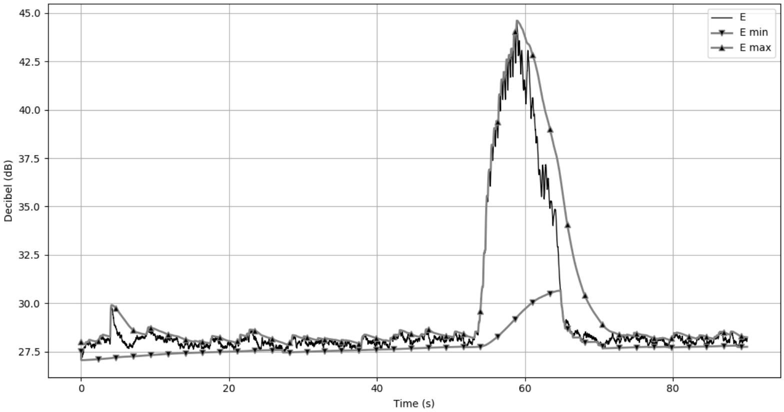


Figure 3.10: Envelope calculation of a running scenario. The delta has its peak just before the 60 second mark.

To get a more precise event detection the envelope is divided into two different frequency ranges. E_{HP} which represents frequencies over a cutoff frequency c and E_{LP} which represents frequencies below this cutoff frequency [28].

The entire algorithm is described in detail in Algorithm 2 with a short description here. This algorithm requires an approximation of E_{LP} and E_{HP} to update E_{LPmax}, E_{LPmin}, E_{HPmax}, and E_{HPmin} according to equation (3.13) and (3.12). From this, Δ_{LP} and Δ_{HP} are calculated according to equation (3.11). The requirements for an event to be detected are defined in equations (3.15) - (3.17). All statements must be fulfilled for an event to be detected. The parameters pc and η have to be optimised.

$$1.\ \Delta_{LP} > \eta \tag{3.15}$$

$$2.\ E_{LP} - E_{LPmin} > pc \cdot \Delta_{LP} \tag{3.16}$$

$$3.\ \Delta_{HP} > 0.5 \cdot \eta \tag{3.17}$$

Algorithm 2: Tracking envelope for audio detection

```
Data: Envelope tracking to determine if activity is happening
input: $eHP, eLP$
Settings: n, pc
eLPmin = min(raise(eLPmin), eLP);
eLPmax = max(decay(eLPmax), eLP);
eHPmin = min(raise(eHPmin), eHP);
eHPmax = max(decay(eHPmax), eHP);
deltaLP = eLPmax - eLPmin;
deltaHP = eHPmax - eHPmax;
if deltaLP > n AND eLP - eLPmin > pc * deltaLP AND deltaHP > 0.5 * n
 then
    return: 1;
else
    return: 0;
end
```

Power Envelope

The power envelope uses a frame size of 1024 samples, and takes advantage of the FFT to get the spectral components of the signal. The spectral components are split into an upper and lower band with the cutoff frequency c at 2000 Hz. From these bands two envelopes, E_{LP} and E_{HP}, are calculated using equation (2.25).

To reduce the noise in the signal and make it more usable a first order low pass filter is applied to the two envelopes. The filter used is defined by equation (3.18), where *TAV* has the same definition as in the tracking algorithm, and x represents the current value of the envelope.

$$x_{filt}(n) = (1 - \mathit{TAV}) \cdot x_{filt}(n-1) + \mathit{TAV} \cdot x(n) \tag{3.18}$$

The low pass filter used is found in [36] who uses it for RMS envelope calculation. Finally the two envelopes are used as input to Algorithm 2. The output from the algorithm is the detection signal. No post processing is applied for this method.

RMS Envelope

In contrast to the power envelope, the RMS can be calculated sample by sample. However it is still divided into frames to fit into the audio detection pipeline. The frame size used contains 512 samples. After framing the signal is divided into two frequency bands with the help of a Butterworth filter. The low pass band contains frequencies between 200 Hz to 2000 Hz, and the high pass band contains frequencies above 2000 Hz.

The RMS is then performed on the two frequency bands separately to calculate E_{LP} and E_{HP}. The RMS equation (2.26) is defined for an entire signal, but is for the purposes of the envelope algorithm instead calculated once per frame. First, for every frame the mean square is calculated. Second, a low-pass filter is used to smooth out the fluctuations (see equation (3.18)) [36]. Third, the root of the low-passed mean square is calculated, resulting in an estimate of the RMS. This results in a RMS envelope for E_{LP} and E_{HP} which is sent into Algorithm 2. Its output is the detection signal.

3.3.4 Evaluation

To evaluate the effectiveness of the audio detection algorithms the procedure listed below is used.

1. Run algorithm on a warm-up clip
2. Run algorithm on a scenario clip
3. Calculate performance
4. Run algorithm on the same scenario clip again
5. Calculate performance

The state of the algorithm is saved between each run, to simulate a longer continuous clip. In reality the algorithms will run for a long period of time. Therefore a warm-up clip of 20 minutes is used to get the algorithms into a realistic state. This is especially important for the GMM algorithms as they use this period to create appropriate distributions, modelling the audio scene.

Scenario clip refers to the clips with walking, running, or talking, listed in Table 3.1. As can be seen in the list each scenario is run twice in succession. The reason for this is that it should become apparent if the algorithms adapt too fast, which would result in a change in performance between the two runs.

Last an evaluation of the possibility to perform foreground categorisation is done. This finer classification may be correlated with the behaviour of the RC to allow for more accurate warnings. This evaluation is only conducted on the MD-GMM and MD-GMM-UF. It is done by giving each distribution an ID, and when a sample is matched to a specific distribution the ID of the distribution is stored. An ID is not updated when the distribution is replaced to limit the amount of IDs used. It is assumed that the GMMs are tuned to have few replacements due to being able to model different sound.

3.4 Audio Preparation for Video Rate Controller

This part aims to evaluate the performance of the whole RC preparation system together. This includes creating a detection signal from raw audio and based on this signal regulate the RC. Together this implements all of block A in the system overview shown in Figure 3.1. For this part only the most successful algorithm of each variant is used. From the audio detection part two algorithms are tested, namely the RMS envelope and the MD-GMM. On the RC regulator side the minimum QP method is used.

In practice the two modules are not connected directly. Instead the audio detection signal is calculated separately. Then all delays introduced by framing and post processing are calculated and accounted for. After this the detection signal is re-sampled to the same frequency as the frame rate and used as input to the RC regulator.

3.4.1 Evaluation

The evaluation of this part is conducted similarly to the Rate Control evaluation. The same clips are used, but at this stage an audio detection signal is of course used as input to the RC regulator instead. The audio detectors are warmed up on a long clip followed by two runs on the scenario. The detection signal of the second run is used as input to the regulator. The regulator works with the RC and a simulated encoder. The procedure can be seen in the following list:

1. Run audio detector on a warm-up clip
2. Run audio detector on a scenario clip
3. Run audio detector on the same scenario clip again and save detection

4. Simulate encoding on the clip using detection as input
5. Record number of frame drops and plot QP graph

The video quality is not evaluated on this test. The reason is that the clips used for quality evaluation have to be very detailed which requires them to be short. The audio detection step is not meaningful to perform on such short clips.

Chapter 4
Results

This chapter displays the results from all tests performed. Static parameters are given for each test together with the result. First the results from RC regulation are presented. After that the audio detection is presented. Last comes results of the combination of the two solutions as a single system.

4.1 Rate Control Regulation

The following section displays the results of each test performed for rate control regulation. All tests have a table displaying the number of dropped frames with and without audio detection. Some tests have a corresponding graph that shows the QP value changes throughout the clip. The scenarios (walking, running, or talking) requires different bit rate due to different amount of motion during their events. The maximum bit rate was decided beforehand by testing (see Table 3.1). For these tests a ground truth signal is used as audio detection in each clip.

4.1.1 Increasing the Minimum QP

The results for regulating minimum QP are displayed in Table 4.1 - 4.3. Table 4.1 shows frame drop performance where a max bit rate suitable for each scenario has been picked. The Min QP column specifies the level set when audio events are detected. Once again, the detection is based on the ground truth signal at this stage. As can be seen in the table the number of dropped frames are zero for all clips except for *Walking5*. However, *Walking5* has the greatest improvement in terms of avoided frame drops.

Table 4.2 shows when testing scenario running and talking with a limit of 260 kbit/s. *Running4* has one less drop compared to when running without a detection

Table 4.1: Frame drops of all scenarios, with bit rates chosen for them specifically.

Clip name	Min QP	Max bit rate	Dropped frames	
		Kbit/s	With audio regulation	Without audio regulation
Walking4	34	260	0	3
Walking5	34	260	2	8
Running4	34	500	0	3
Running5	34	500	0	3
Talking3	34	1000	0	1
Talking5	34	1000	0	1

signal, while *Running5* has one more drop. *Talking4* takes advantage of the audio detection signal and has one less drop. In Table 4.3 the scenarios walking and talking was tested with 500 kbit/s. No drops are recorded for the walking scenario. *Talking4* benefits from the audio regulation and has three less drops while *Talking5* instead drops one more.

Table 4.2: Frame drops of running & talking scenarios with the bit rate chosen for walking.

Clip name	Min QP	Max bit rate	Dropped frames	
		Kbit/s	With audio regulation	Without audio regulation
Running4	34	260	3	4
Running5	34	260	3	2
Talking3	34	260	8	9
Talking5	34	260	4	4

The change in the QP for *Walking4* with a bit rate of 260 kbit/s can be seen in Figure 4.1. It is possible to see that the detection signal is active well before the events enters the camera's field of view and forces a rise in QP. The grey thick line shows how the QP changes without audio detection. It increases rapidly from the minimum QP all the way to max as soon as the visual event enters. It then decreases as the motion reduces and just before the end a small spike happens. In the thin black line the regulation is seen when the line jumps to QP 34. This increase remains throughout the entire audio detection. As seen in the black line the regulated version is quicker to increase the QP to its maximum when the visual event happens. From the three vertical lines it can be seen that the frame drops comes at the beginning of the event

Table 4.3: Frame drops of walking & talking scenarios with the bit rate chosen for running.

Clip name	Min QP	Max bit rate	Dropped frames	
		Kbit/s	With audio regulation	Without audio regulation
Walking4	34	500	0	0
Walking5	34	500	0	3
Talking3	34	500	1	4
Talking5	34	500	1	0

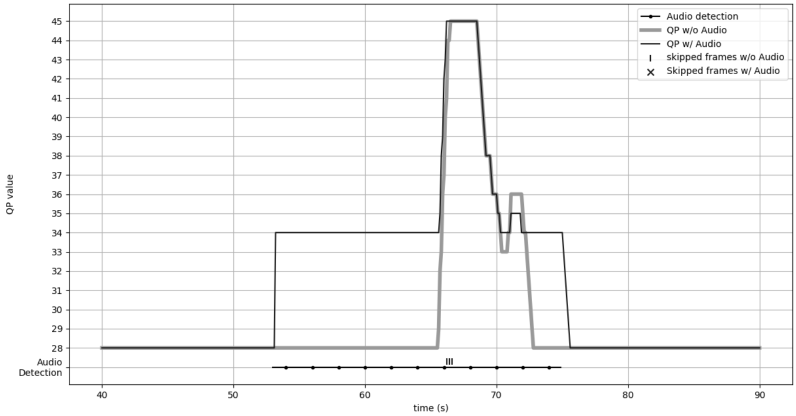


Figure 4.1: Plot showing QP, audio detection, and frame drops with a regulated and non regulated RC on *Walking4* with bit rate limit 260 kbit/s.

and that they are removed when using audio.

Figure 4.2 and 4.3 shows similar results for *Running4* with 500 kbit/s and *Talking3* with 1000 kbit/s. These figures also show when frame skips occur. Figure 4.2 is similar to the walking scenario where the regulated version is quicker to react. This time the regulated version has a slightly lower maximum QP achieved during the event. The frame drops again happens at the beginning of the event and they are removed for the regulated test. The increased minimum QP extends far beyond the event. This is seen in both the audio detection line as well as having an increased minimum QP after the event passed. Figure 4.3 it can be seen that the only dropped frame is removed by introducing audio detection. Also, this time the maximum achieved QP is 40 for the version using audio compared to 44 otherwise.

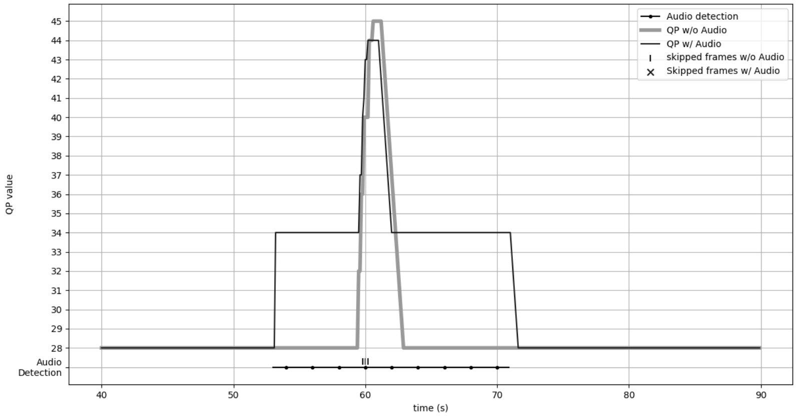


Figure 4.2: Plot showing QP, audio detection, and frame drops with a regulated and non regulated RC on *Running4* with bit rate limit 500 kbit/s.

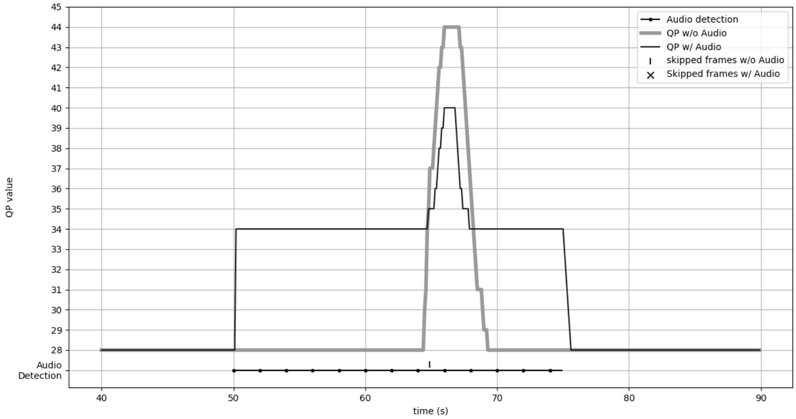


Figure 4.3: Plot showing QP, audio detection, and frame drops with a regulated and non regulated RC on *Talking3* with bit rate limit 1000 kbit/s.

4.1.2 Increasing the Maximum Bit Rate

Tests using an increased maximum bit rate during the event are presented here. A ground truth audio detection is used for optimal detection. Seen in Table 4.4 the number of dropped frames are reduced for all cases. Again all test except *Walking5* have zero dropped frames when regulating.

Table 4.4: Frame skip results when using the maximum bit rate scheme. The max bit rate is specific for each scenario.

Clip name	**Max Bit Rate multiplier**	**Max bit rate**	**Dropped frames**	
		Used for audio regulation Kbit/s	With audio regulation	Without audio regulation
Walking4	2	260	0	3
Walking5	2	260	2	8
Running4	2	500	0	3
Running5	2	500	0	3
Talking3	2	1000	0	1
Talking5	2	1000	0	1

Figure 4.4 shows that the peak QP increase is the same for both the version with and without the detection. Despite this, the one with a warning has fewer dropped frames. The main difference comes at the end of the event. Here the version having the warning signal can reduce the QP faster than the version not having the warning.

Further, looking at Figure 4.5 the main difference is that the QP does not increase as much when using a warning signal. As for *Walking4* it can reduce QP earlier. Figure 4.6 shows the same result, however here the QP increase is even smaller.

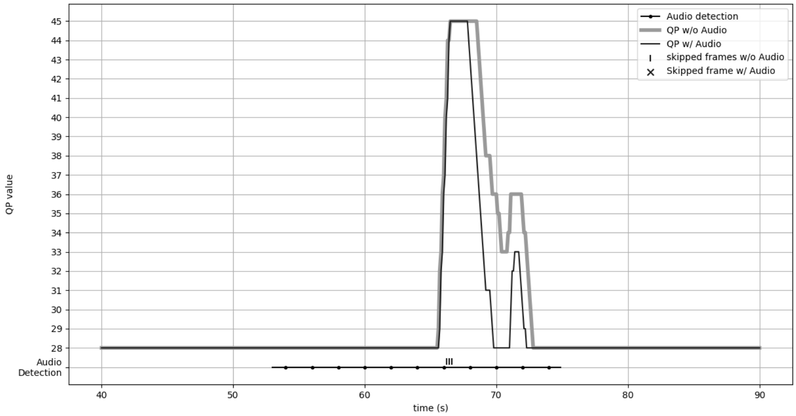


Figure 4.4: Plot showing QP, audio detection, and frame drops with a regulated and non regulated RC on *Walking4* with the bit rate limit 260 kbit/s.

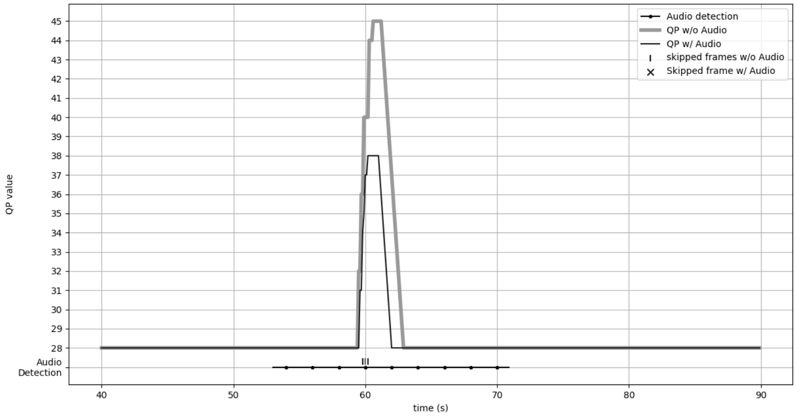


Figure 4.5: Plot showing QP, audio detection, and frame drops with a regulated and non regulated RC on *Running4* with the bit rate limit 500 kbit/s.

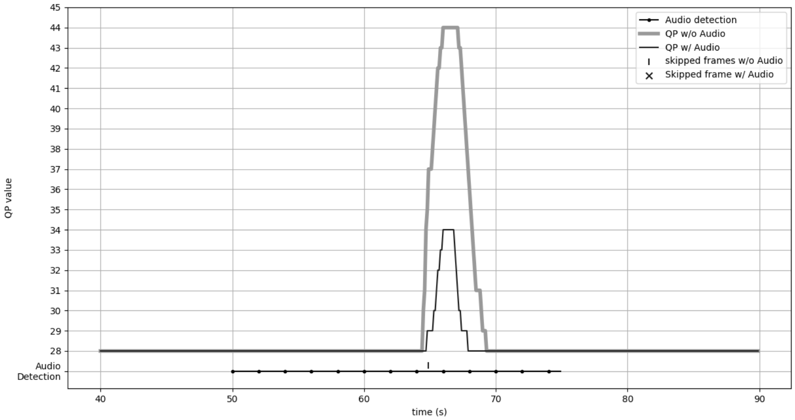


Figure 4.6: Plot showing QP, audio detection, and frame drops with a regulated and non regulated RC on *Talking3* with the bit rate limit 1000 kbit/s.

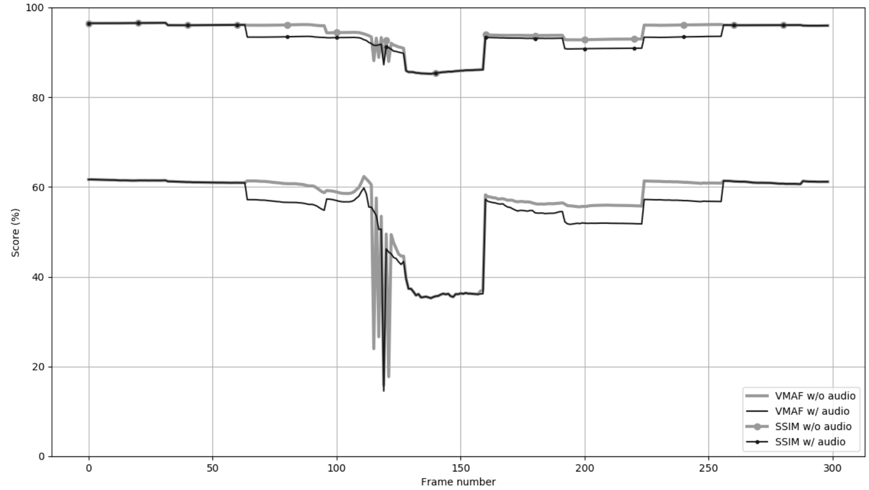
4.1.3 Encoding Quality

This section presents the results from encoding video clips using a ground truth audio detection signal, and compares this to the normal system. The minimum QP regulation scheme is used. Firstly, the mean VMAF and SSIM scores calculated on the full duration of each clip are presented in Table 4.5. In the table it can be seen that in terms of SSIM the regulated version performs slightly worse than the unregulated version. In terms of VMAF the regulated version is slightly better on the two running clips, but worse on the other four.

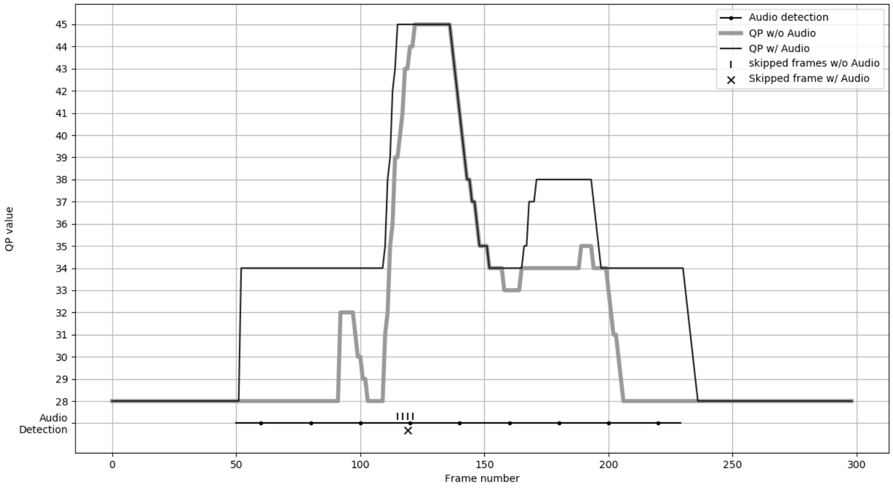
Figures 4.7 - 4.9 shows the SSIM and VMAF scores along with QP values for each frame of the re-encoded test videos. Each scenario uses the decided maximum bit rate used in earlier experiments, based on the information in Table 3.1. From the figures it can be noted that both VMAF and SSIM react strongly to frame skips. The scores are also comparably low during the majority of the detection period, with the exception of the peak of the visible event.

Table 4.5: Mean quality scores for the six test clips. Both scores are on a scale of 0 to 100, where 100 is the best.

Clip Name	**Max bit rate**	**VMAF**		**SSIM**	
	kbit/s	normal	with audio	normal	with audio
QualityWalk1	260	56.39	54.93	94.08	93.14
QualityWalk2	260	56.90	55.28	94.18	93.17
QualityRun1	500	57.21	57.50	94.44	94.23
QualityRun2	500	56.99	57.10	94.44	94.13
QualityTalk1	1000	59.43	57.52	95.24	94.04
QualityTalk2	1000	59.83	57.86	95.33	94.10

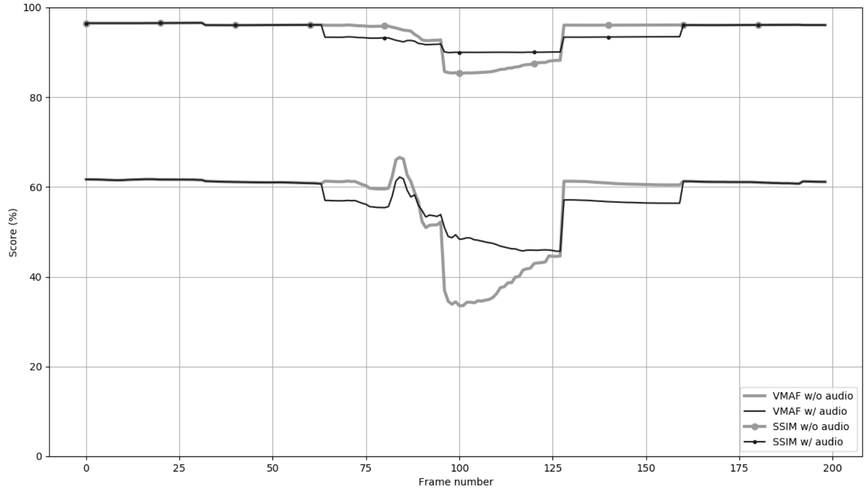


(a) SSIM and VMAF scores in percent throughout the duration of the clip. Higher score is better.

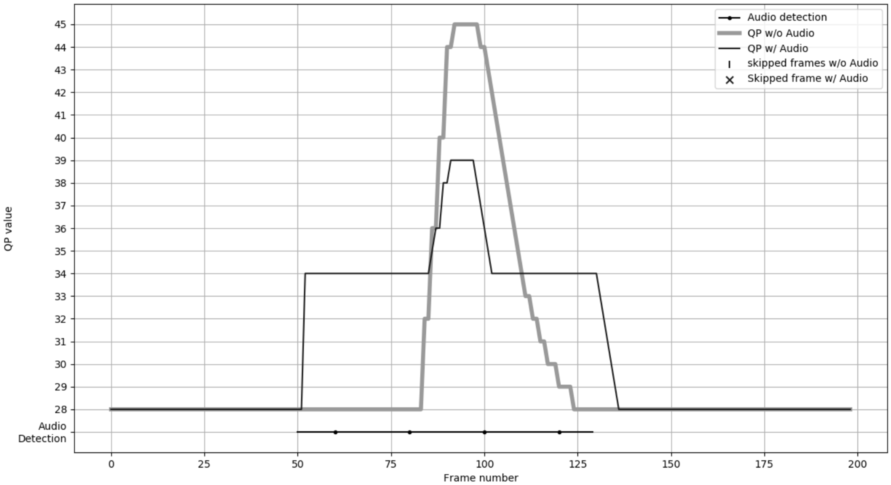


(b) QP value throughout the duration of the clip.

Figure 4.7: QP along with compression quality metrics on the video *Quality-Walk1*, with and without using audio detection.

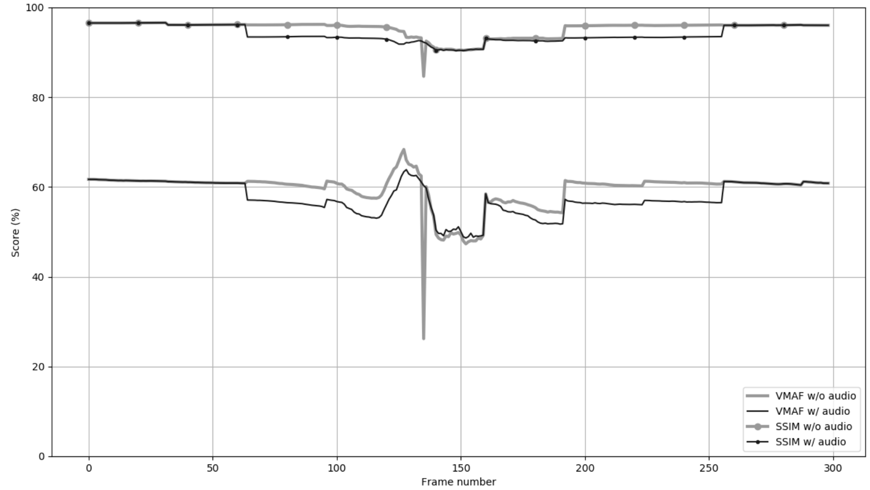


(a) SSIM and VMAF scores in percent throughout the duration of the clip. Higher score is better.

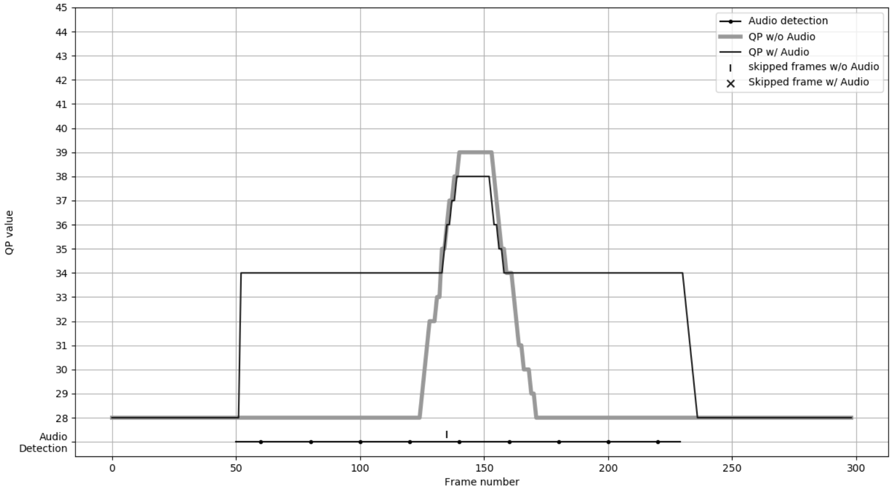


(b) QP value throughout the duration of the clip.

Figure 4.8: QP along with compression quality metrics on the video *QualityRun1*, with and without using audio detection.



(a) SSIM and VMAF scores in percent throughout the duration of the clip. Higher score is better.



(b) QP value throughout the duration of the clip.

Figure 4.9: QP along with compression quality metrics on the video *QualityTalk1*, with and without using audio detection.

4.2 Audio Detection

In this section the results of all audio event detector algorithms are presented. For each algorithm the detection performance is given in terms of accuracy, sensitivity and specificity. Some tests have a figure showing the audio signal and detection over time.

4.2.1 GMM Results

In this section the results of the GMM algorithms are presented. The first section shows the result of a 1D-GMM. The next presents the results of the MD-GMM and last comes MD-GMM-UF. In all cases the post processing algorithm KSLA has been applied after the initial detection based on the GMMs alone. A recap of parameters used and a short description can be seen in Table 4.6.

Parameter	**Usage**
α	Learning rate
α_{weight}	Learning rate for weight update
α_{rho}	Learning rate for distribution update
P	Background threshold
K	Gaussians per Mixture
mels	Number of mel bands generated
w_{init}	Initial weight of new Gaussians
σ_{init}	Initial standard deviation of new Gaussians
Σ_{init}	Initial correlation matrix for new Gaussians
d	Number of features used
n	Maximum Gaussian match distance in standard deviations

Table 4.6: Parameters used in the one-dimensional and/or the multidimensional GMM.

1D-GMM

In Table 4.7 the results of the 1D-GMM are presented. Of each scenario seen in the table, the two uppermost instances have been used to train the GMM and the lowermost has never been seen by the algorithm before. The scores presented in the table are calculated based on how well the detection of the algorithm overlaps with the labeled ground truth signal. It can be seen from the table that the average accuracy is 93%. The average sensitivity is 77% and average specificity is 97%.

Most noticeable from the table is the sensitivity on walking scenario. *Walking1* with sensitivity of 48%, *Walking2* with 53% and *Walking3* with 40%. The second

Table 4.7: Results and parameters for 1D-GMM.

Params	**Value**
α	0.0001
P	0.5
K	4
mels	32
w_{init}	0.02
σ_{init}	4
d	8
n	2.5

	Run 1			Run 2		
Testclip	**Accuracy**	**Sensitivity**	**Specificity**	**Accuracy**	**Sensitivity**	**Specificity**
Walking1	90.96%	48.36%	100.0%	91.03%	48.76%	100.0%
Walking2	87.79%	53.53%	98.87%	89.31%	59.73%	98.87%
Walking3	86.15%	40.63%	100.0%	87.19%	45.10%	100.0%
Running1	94.04%	85.33%	95.25%	96.12%	85.33%	97.62%
Running2	94.62%	95.08%	94.55%	94.62%	95.08%	94.55%
Running3	97.34%	98.61%	97.18%	97.34%	98.61%	97.18%
Talking1	96.16%	88.31%	99.34%	96.16%	88.31%	99.34%
Talking4	98.39%	95.41%	99.73%	98.39%	95.41%	99.73%
Talking2	92.85%	92.92%	92.82%	92.85%	92.92%	92.82%
Average	**93.14%**	**77.58%**	**97.53%**	**93.67%**	**78.81%**	**97.79%**

run has slightly increased sensitivity for some walking scenarios.

Looking at the other scenarios the running scenario has slightly lower specificity compared to the other scenarios. The accuracy is higher than for walking and on par with talking, however.

Figure 4.10 - 4.12 shows the detection compared to the ground truth for the three evaluation files *Walking3*, *Running3*, and *Talking2* together with the audio signal. In Figure 4.10 it can be seen that this detection happens almost 10s after the ground truth. The detection also stops earlier than the ground truth. Figure 4.11 displays *Running3*. This detection starts at the same time as the ground truth. It is however extended for a few seconds longer than the ground truth. It is also possible to see one false positive near the end of the detection. Figure 4.12 is almost equally fast as the ground truth and have more false positive detection at the end. Looking at the audio signal it can be seen that there is some form of activity at the end.

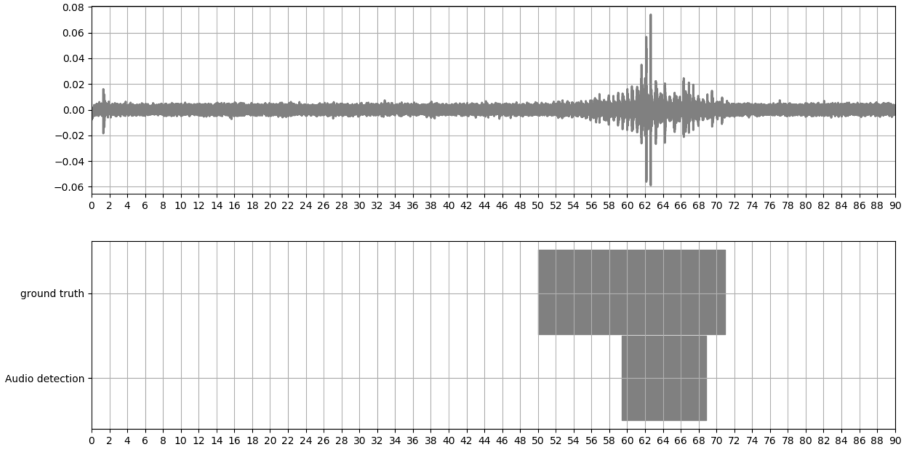


Figure 4.10: Detection for *Walking3* using the 1D-GMM.

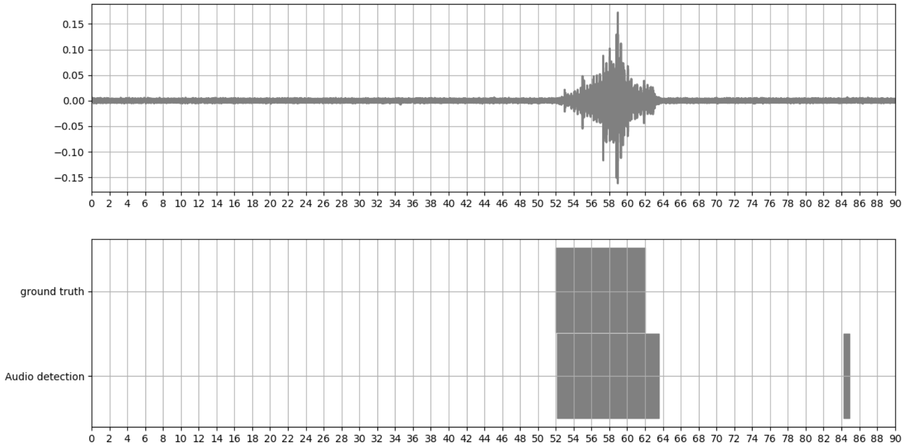


Figure 4.11: Detection for *Running3* using the 1D-GMM.

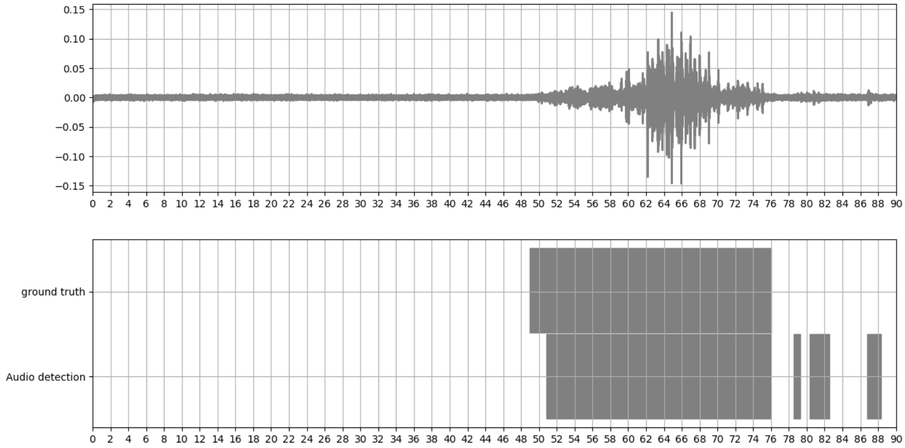


Figure 4.12: Detection for *Talking2* using the 1D-GMM.

MD-GMM

Table 4.8 displays results and parameters for the MD-GMM. The average accuracy is 96%, average sensitivity 95% and average specificity 96%. Compared to that of the 1D-GMM the results shown in this table are fairly consistent. The sensitivity for the walking scenario is lower than running and talking, but it is close to 90%. *Running2* stands out for having lower specificity than *Running1* and *Running3*. *Talking2* also has lower specificity compared to the other talking clips. Looking at the second run only *Walking1* have a greater than 5 percentage point difference compared to the first run. Sensitivity has lowered from 91% to 86%.

Looking at Figure 4.13 displaying *Walking3*, the detection is about 2 s slower

Table 4.8: Results and parameters for MD-GMM.

Params	Value
α_{weight}	0.0003
α_{rho}	0.001
P	0.66
K	15
w_{init}	0.05
Σ_{init}	$\begin{bmatrix} 17 & & 0 \\ & \ddots & \\ 0 & & 17 \end{bmatrix}$
d	8
n	5

	Run 1			Run 2		
Testclip	**Accuracy**	**Sensitivity**	**Specificity**	**Accuracy**	**Sensitivity**	**Specificity**
Walking1	97.90%	91.34%	99.29%	97.05%	86.46%	99.29%
Walking2	96.98%	89.01%	99.56%	96.98%	89.01%	99.56%
Walking3	97.34%	89.82%	99.63%	97.15%	89.00%	99.63%
Running1	99.35%	98.52%	99.46%	99.35%	98.52%	99.46%
Running2	91.21%	96.63%	90.45%	92.82%	96.63%	92.29%
Running3	97.43%	99.47%	97.18%	97.43%	99.47%	97.18%
Talking1	97.01%	97.50%	96.81%	97.01%	97.50%	96.81%
Talking4	98.20%	96.93%	98.77%	98.20%	96.93%	98.77%
Talking2	90.60%	96.91%	87.90%	90.60%	96.91%	87.90%
Average	**96.22%**	**95.12%**	**96.56%**	**96.29%**	**94.49%**	**96.77%**

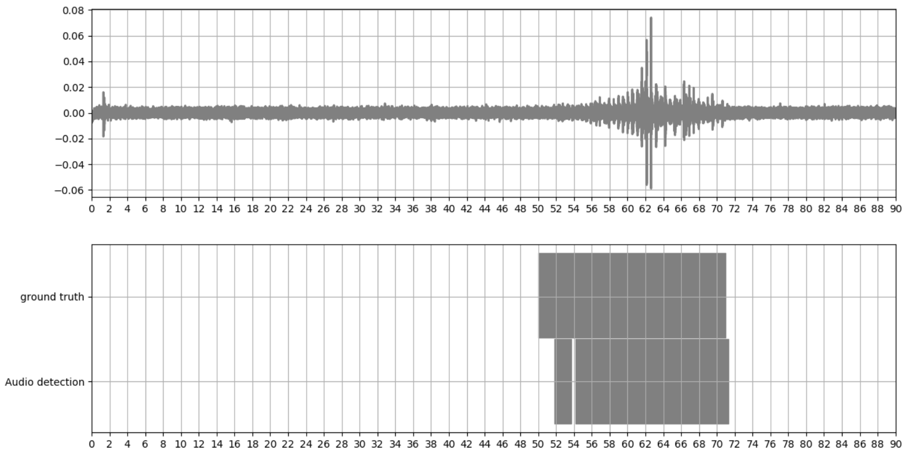


Figure 4.13: Detection for *Walking3* using the MD-GMM.

compared to the ground truth. There is also a small gap in the detection. Figure 4.14 displays *Running3* having more false positives at the end. Further, there is a gap in the detection after the event has passed. Lastly Figure 4.21 shows the detection for *Talking2*. The detection during the event is similar to the ground truth detection. However, as seen in the figure there are several false positives at the end.

Also provided with this is statistics on how the MD-GMM distributes each measured point. Figure 4.16 shows the statistic for $Walking3$. In this case distribution 0 represents background sound. This distribution is matched most of the time. During the event a second distribution is mostly matched. This distribution also have some sporadic matches throughout the clip. Also seen is that on this scenario the MD-GMM does not use more than two distributions even though there are several distributions available.

Figure 4.17 displays statistics on *Running3*. Here the two distributions are more clearly separated with fewer false positive matches. Also worth noting is that there are a few matches in a third distribution. Figure 4.18 shows *Talking2* and is similar to *Running3*. This time several more distributions get sporadic matches. At the end it is possible to see that there are several matches in the foreground distribution. These coincide with the false positives at the end of Figure 4.15.

From this metric it can be seen that for all clips there are only two dominant distributions. For the more volume intense scenarios there are sporadic matches in other distributions.

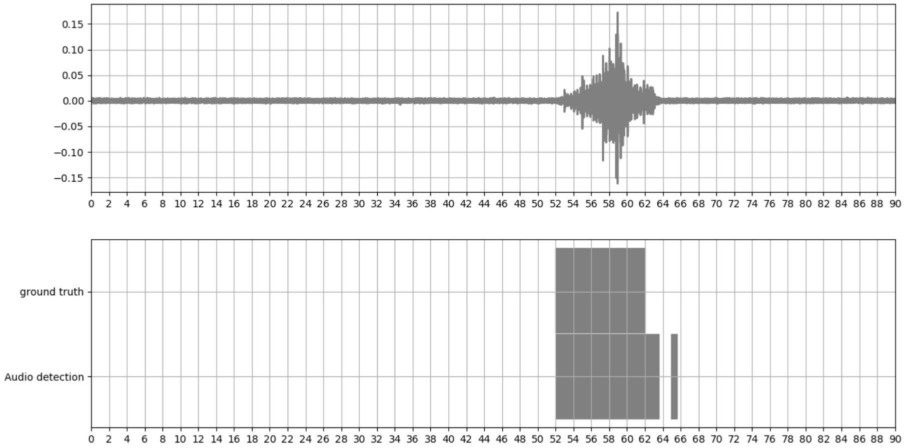


Figure 4.14: Detection for *Running3* using the MD-GMM.

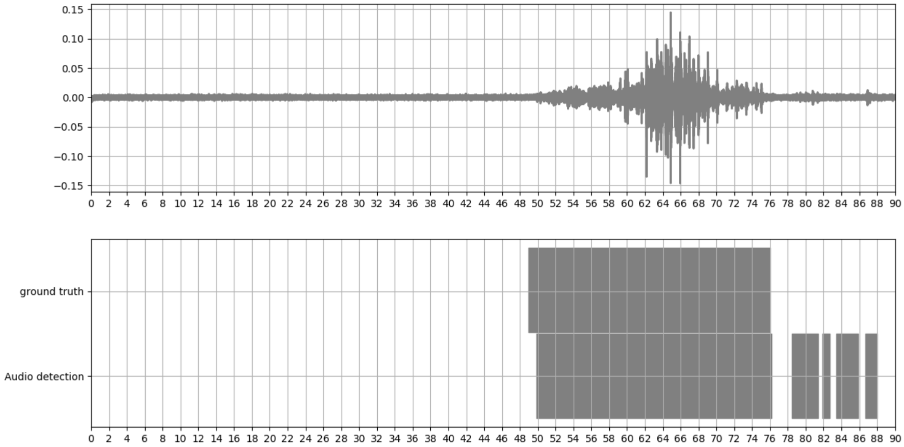


Figure 4.15: Detection for *Talking2* using the MD-GMM.

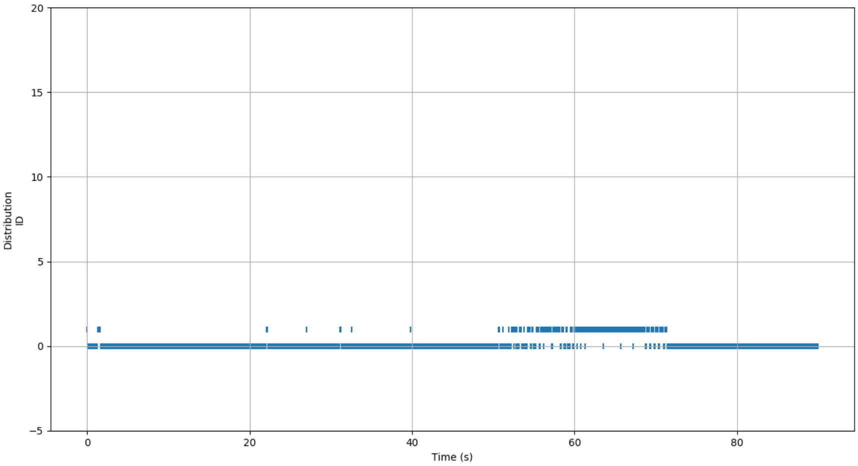


Figure 4.16: MD-GMM distribution statistics for *Walking3*.

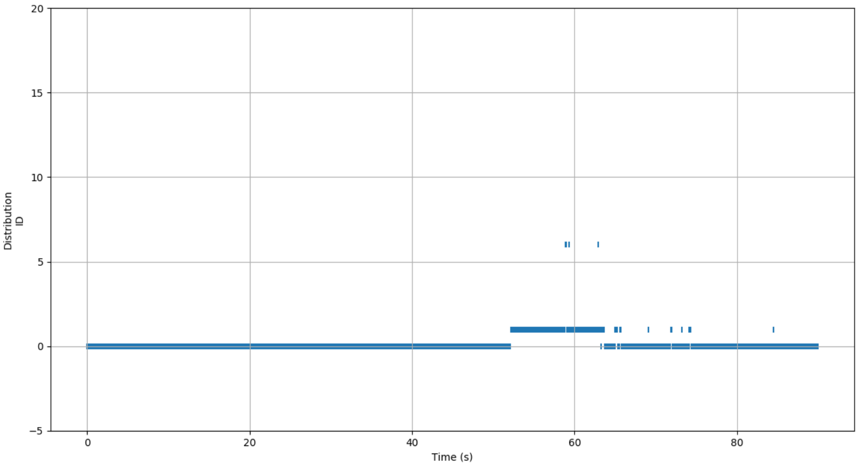


Figure 4.17: MD-GMM distribution statistics for *Running3*.

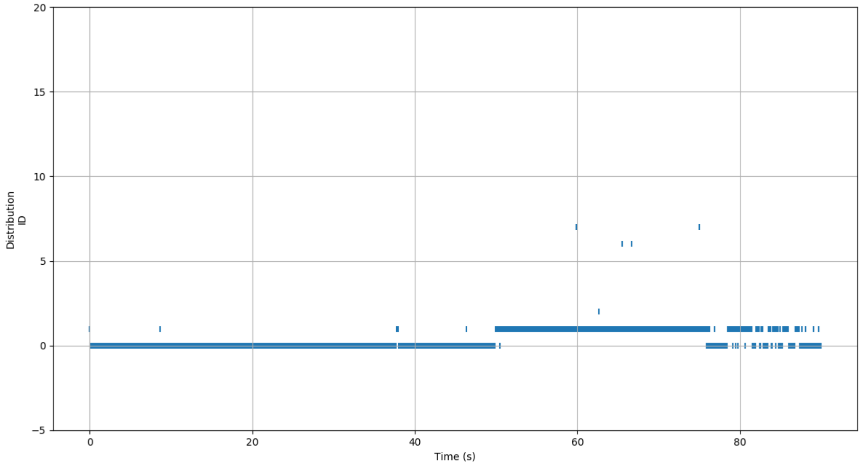


Figure 4.18: MD-GMM distribution statistics for *Talking2*.

MD-GMM-UF

Here results of the MD-GMM-UF are presented and can be viewed in Table 4.9. This algorithm has similar results to the MD-GMM, with the exception of the sensitivity. Average accuracy of 96%, average sensitivity of 88% and average specificity of 98%. The sensitivity is in general lower compared to the MD-GMM. Most noticeable are the walking scenarios having between 69% to 79% in sensitivity. The performance on the second run are similar to the first run with no major differences.

Figure 4.19 shows *Walking3*. The detection is slightly delayed and contains gaps. One small false positive detection is found at the very beginning of the clip. This corresponds to a small event at that time, visible in the audio signal plotted above.

Table 4.9: Results and parameters for MD-GMM-UF.

Params	**Value**
α_{weight}	0.00005
α_{rho}	0.0001
P	0.66
K	40
w_{init}	0.01
Σ_{init}	$\begin{bmatrix}5 & & 0\\ & \ddots & \\ 0 & & 5\end{bmatrix}$
d	10
n	6.8

	Run 1			Run 2		
Test clip	**Accuracy**	**Sensitivity**	**Specificity**	**Accuracy**	**Sensitivity**	**Specificity**
Walking1	96.27%	79.05%	99.92%	96.00%	77.52%	99.92%
Walking2	92.32%	69.24%	99.78%	92.15%	69.24%	99.56%
Walking3	93.29%	74.58%	98.98%	93.62%	76.00%	98.98%
Running1	98.52%	92.32%	99.38%	98.52%	92.32%	99.38%
Running2	94.55%	95.66%	94.39%	94.88%	95.66%	94.77%
Running3	97.77%	97.97%	97.74%	97.48%	97.97%	97.42%
Talking1	97.27%	94.79%	98.28%	97.27%	94.79%	98.28%
Talking4	97.68%	95.49%	98.67%	97.65%	95.49%	98.63%
Talking2	94.42%	96.24%	93.64%	93.87%	96.24%	92.86%
Average	**96.15%**	**88.28%**	**98.36%**	**96.13%**	**88.28%**	**98.33%**

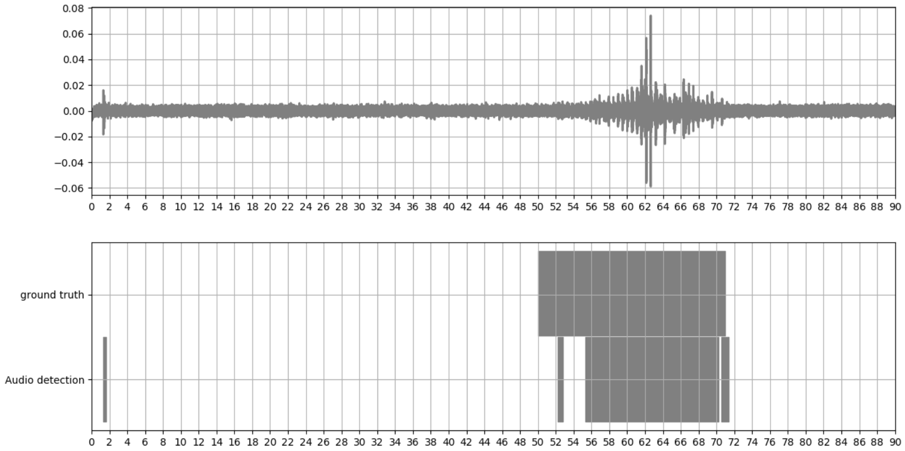


Figure 4.19: Detection for *Walking3* using MD-GMM-UF.

There are also some gaps in the detection, one large in the beginning and a smaller at the end. Figure 4.20 shows that the *Running3* detection with MD-GMM-UF is quite similar to the detection of the MD-GMM. It has two small detections after the event has passed as well as an extended detection compared to the ground truth. Lastly Figure 4.21 displays the *Talking2* scenario. This is also similar to the MD-GMM detection. It is however slightly less sensitive to the false positive detection after the event.

Here are statistics displaying how the MD-GMM-UF classifies each frame. As can be seen in Figure 4.22, during *Walking3* it uses more distributions compared to the MD-GMM. However, there is a similar pattern where there is one background distribution and one large foreground distribution. At the beginning of the event there is a lot of overlap between which one of the two distributions describes the sound best. During the peak of the event several distributions are used, which differs from the MD-GMM.

Studying Figure 4.23 it can be seen that even more distributions are used in *Running3*. For this clip there is less overlap between the background distribution and the foreground distribution, compared to *Walking3* discussed above.

Lastly, Figure 4.24 shows the distributions for *Talking2*. Almost all distributions are used, but the behaviour is not too different than *Running3* other than being longer. Also seen is that the noise at the end of the clip, seen as foreground in Figure 4.21, matches with the lowest foreground distribution.

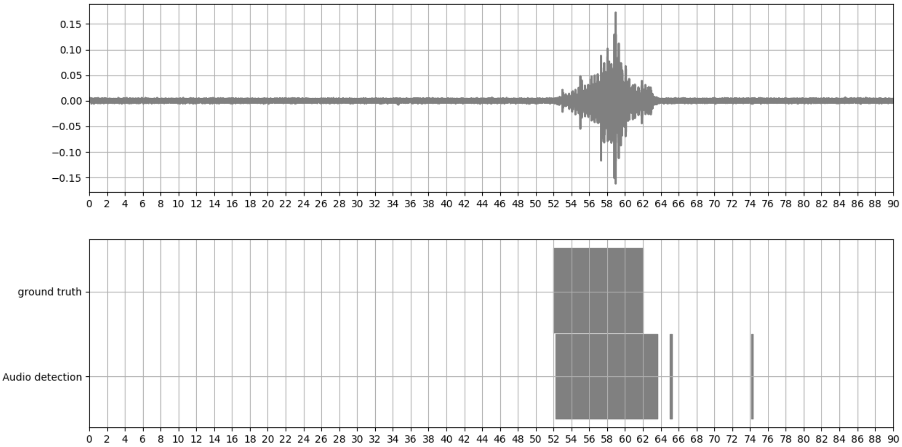


Figure 4.20: Detection for *Running3* using MD-GMM-UF.

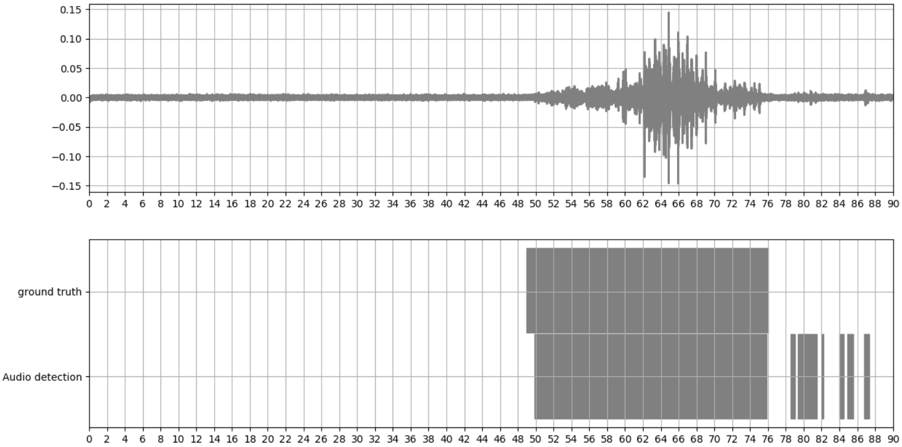


Figure 4.21: Detection for *Talking2* using MD-GMM-UF.

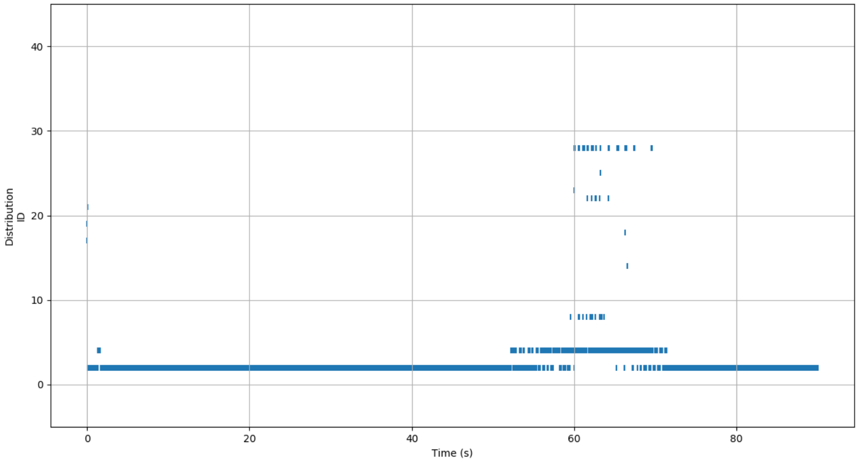


Figure 4.22: MD-GMM-UF distribution statistics for *Walking3*.

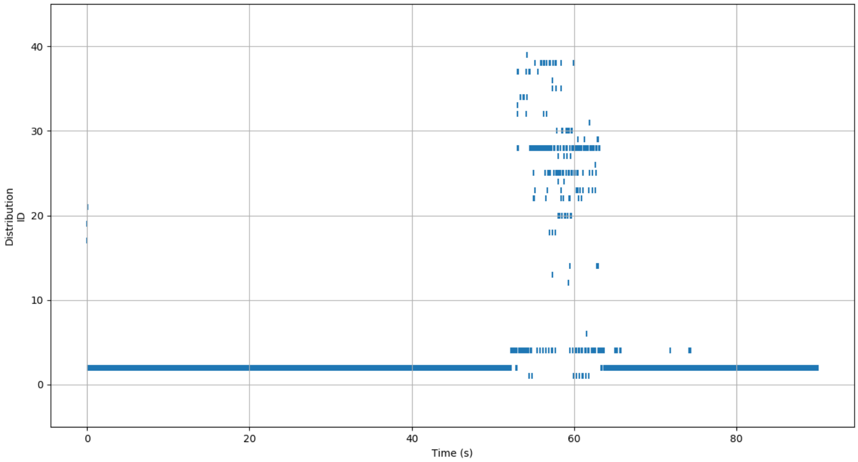


Figure 4.23: MD-GMM-UF distribution statistics for *Running3*.

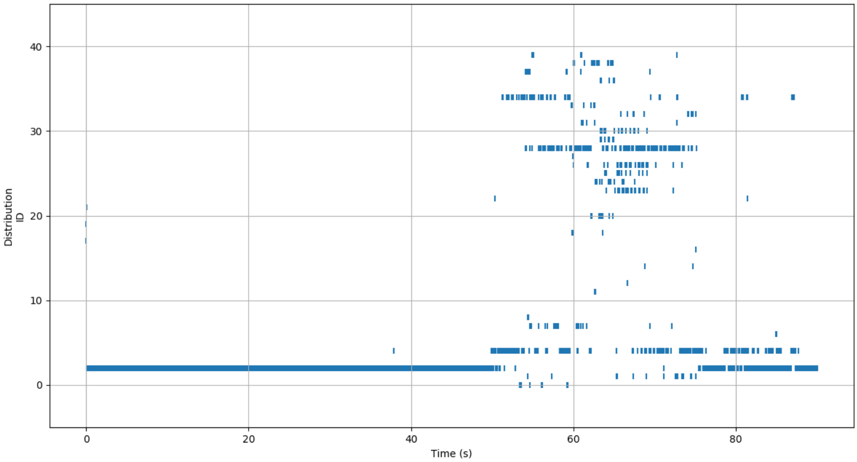


Figure 4.24: MD-GMM-UF distribution statistics for *Talking2*.

4.2.2 Envelope Audio Detection

Here results of envelope tracking as audio detection is presented. The same method for evaluating envelope tracking detection is used as for GMM based audio detection. Table 4.10 gives a short recap to the parameters used.

Parameter	**Usage**
Frame size	size of audio frames
t_{HP}	Attack time for high pass envelope filter
t_{LP}	Attack time for low pass envelope filter
t_{raise}	Attack time for E_{min}
t_{decay}	Attack time for E_{max}
f_c	cutoff frequency
f_{low}	cutoff frequency for low band high pass filter
η	Detection activation threshold
pc	Detection end threshold

Table 4.10: Parameters used in the envelope tracking algorithms for audio detection.

Power Envelope

The results of the power envelope algorithm are presented here. Table 4.11 shows the results in terms of accuracy, sensitivity, and specificity, along with the parameters used. In the table it can be seen that the averages are comparable to the results of MD-GMM-UF, but slightly behind the MD-GMM. In the table it is also seen that the results vary between the two runs.

Table 4.11: Results and parameters for envelope tracking with the power envelope.

Params	**Value**
Frame size	1024
t_{HP}	1500ms
t_{LP}	1500ms
t_{raise}	100000ms
t_{decay}	8000ms
f_c	2000Hz
η	3db
pc	0.3

	Run1			Run 2		
Test Clip	**Accuracy**	**Sensitivity**	**Specificity**	**Accuracy**	**Sensitivity**	**Specificity**
Walking1	97.31%	84.62%	100.00%	96.88%	82.18%	100.00%
Walking2	91.63%	76.41%	96.55%	93.67%	75.64%	99.50%
Walking3	96.08%	83.20%	100.00%	94.59%	76.80%	100.00%
Running1	96.34%	86.50%	97.71%	97.71%	84.95%	99.49%
Running2	96.11%	91.78%	96.71%	97.63%	91.01%	98.55%
Running3	97.13%	92.85%	97.66%	97.10%	92.64%	97.66%
Talking1	97.13%	90.44%	99.84%	97.13%	90.44%	99.84%
Talking4	95.37%	86.70%	99.28%	95.86%	86.70%	100.00%
Talking2	91.15%	90.63%	91.37%	91.12%	90.55%	91.37%
Average	**95.36%**	**87.02%**	**97.68%**	**95.74%**	**85.66%**	**98.49%**

Figures 4.25 - 4.27 shows the detection of the second run compared to the ground truth for each scenario. In Figure 4.25 the detection signal compared to the ground truth is presented for *Walking3*. Here it can be seen that the detection starts about four seconds late. There is a tiny gap at the very end of the detection. The results of *Running3* are presented in Figure 4.26. Here the algorithm has classified most of the event accurately, with a slight delay at the beginning and about 2 seconds at the end. Finally, Figure 4.27 shows the results of the Power envelope on *Talking2*. The detection covers most of the event, with the exception of small parts at the beginning

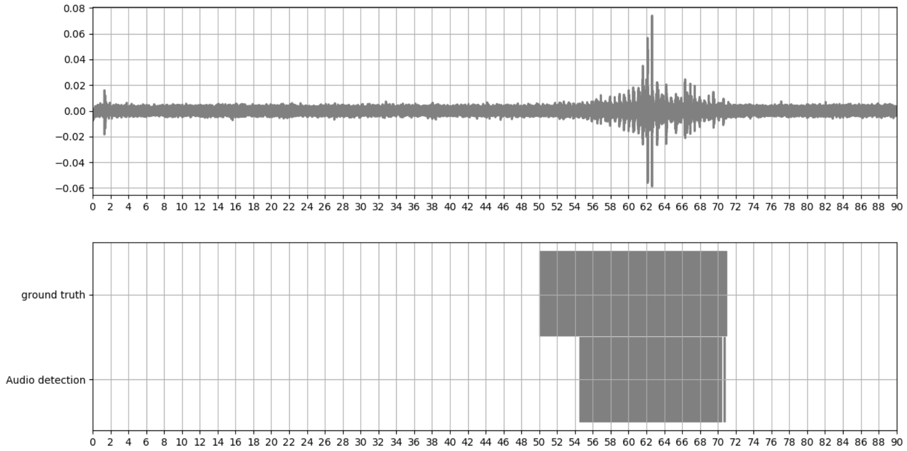


Figure 4.25: Event detection using envelope tracking with power envelope for *walking3*.

and the end. There are also some false detections towards the very end of the clip.

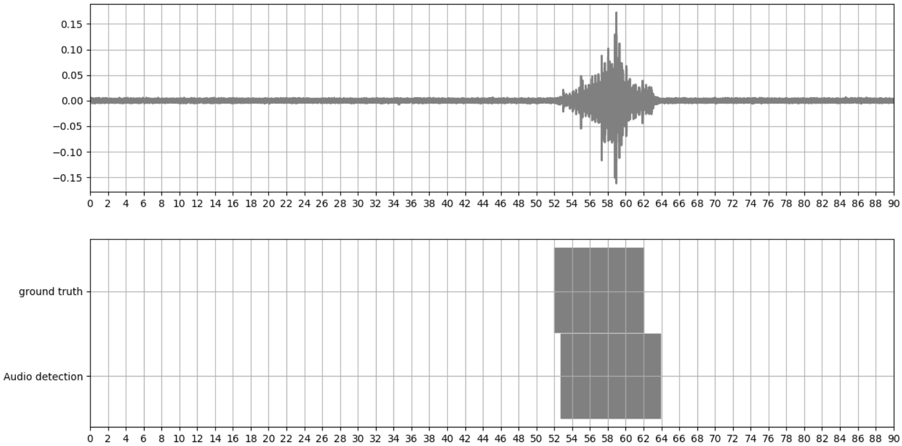


Figure 4.26: Event detection using envelope tracking with power envelope for *Running3*.

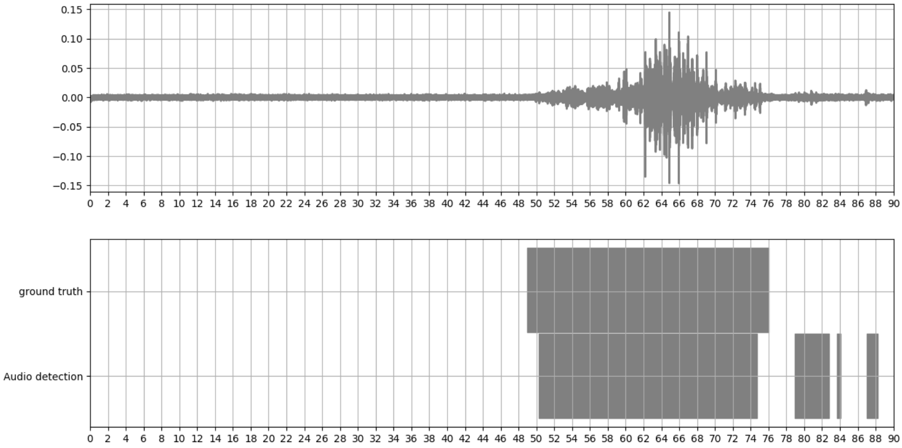


Figure 4.27: Event detection using envelope tracking with power envelope for *Talking2*.

RMS Envelope

In this section the results of approximating envelope with RMS for use in audio detection are shown. Table 4.12 shows the results in terms of accuracy, sensitivity, and specificity, along with the parameters used during the test. Just as with the Power envelope the RMS envelope has comparable performance with MD-GMM-UF. In the table it can also be seen that the performance barely changes between the two runs.

Figures 4.28 - 4.30 shows the detection of the second run compared to the ground truth for each scenario. On *Walking3* in Figure 4.28 the detection starts about two seconds late. On *Running3* in Figure 4.29 the detection starts almost as early as the ground truth, but it keeps going about two seconds too long. Finally, on *Talking2* shown in Figure 4.30 the detection is slightly shorter than the ground truth, and there are a few false detections towards the end of the clip. There are also some minor gaps in the end of the main detection.

Table 4.12: Results and parameters for envelope tracking with RMS

Params	**Value**
Frame size	512
t_{HP}	500
t_{LP}	1000ms
t_{raise}	75000ms
t_{decay}	5000ms
f_c	2000Hz
f_{low}	200Hz
η	3db
pc	0.3

	Run1			Run 2		
Test Clip	**Accuracy**	**Sensitivity**	**Specificity**	**Accuracy**	**Sensitivity**	**Specificity**
Walking1	96.87%	82.08%	100.00%	96.87%	82.08%	100.00%
Walking2	95.43%	85.14%	98.76%	94.88%	85.14%	98.02%
Walking3	96.65%	85.64%	100.00%	96.65%	85.64%	100.00%
Running1	98.10%	89.12%	99.35%	98.10%	89.12%	99.35%
Running2	94.32%	91.68%	94.69%	94.49%	91.68%	94.88%
Running3	97.60%	95.95%	97.81%	97.60%	95.95%	97.81%
Talking1	97.77%	94.05%	99.28%	97.77%	94.05%	99.28%
Talking4	96.47%	88.65%	100.00%	96.47%	88.65%	100.00%
Talking2	88.46%	86.40%	89.34%	88.43%	86.32%	89.34%
Average	**95.74%**	**88.75%**	**97.69%**	**95.69%**	**88.74%**	**97.63%**

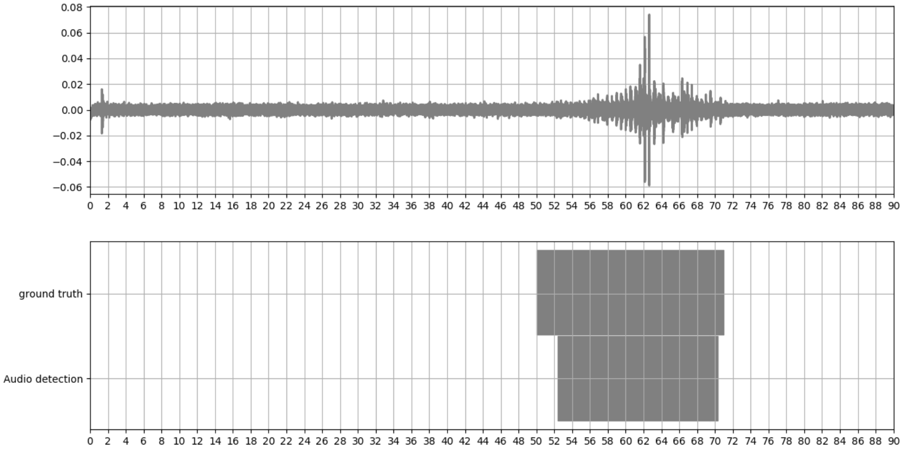


Figure 4.28: Event detection using envelope tracking with RMS for *Walking3*.

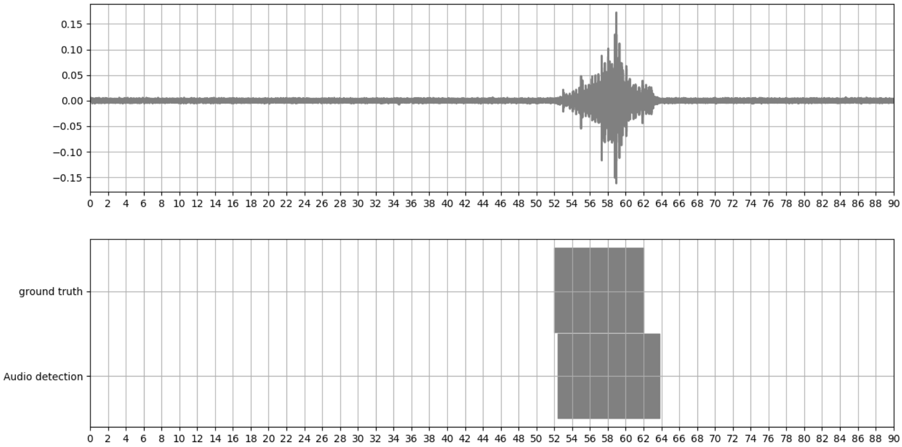


Figure 4.29: Event detection using envelope tracking with RMS for *Running3*.

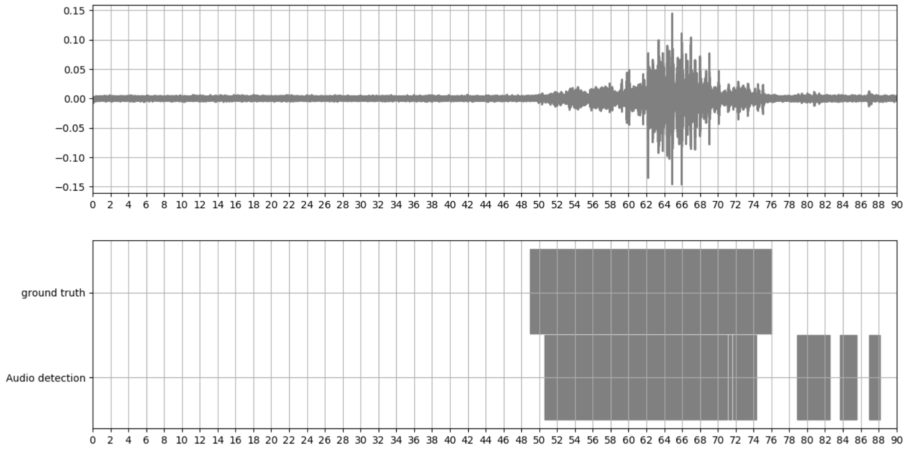


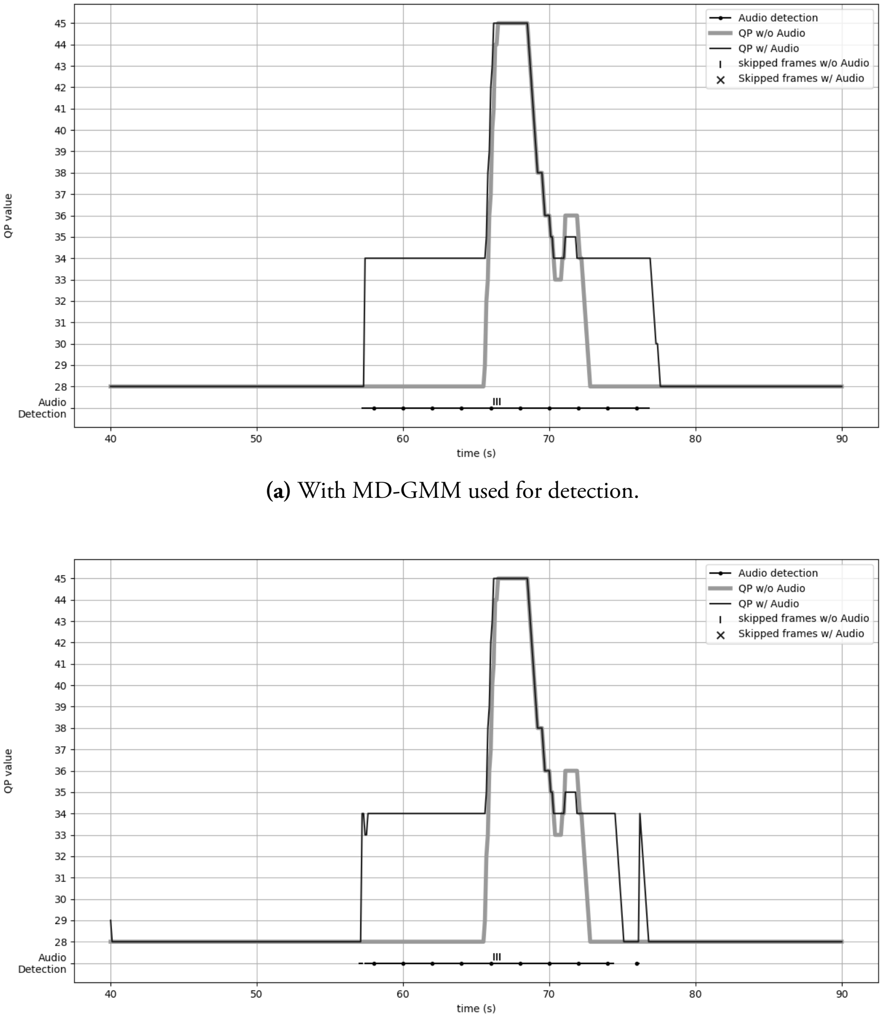
Figure 4.30: Event detection using envelope tracking with RMS for *Talking2*.

4.3 Audio Detection for Preparation of Rate Controller

In this section the results of combining the audio detection with RC regulation is presented. The audio detection is performed by two different algorithms. First by the MD-GMM and then the RMS envelope tracking algorithm. In both cases the minimum QP regulation is used.

Table 4.13 shows the number of dropped frames for different scenarios with audio detection compared to no audio detection. As can be seen the regulation lowered the number of frame skips on all tests, regardless of which of the two algorithms was used. In all cases except one the skips were lowered to zero.

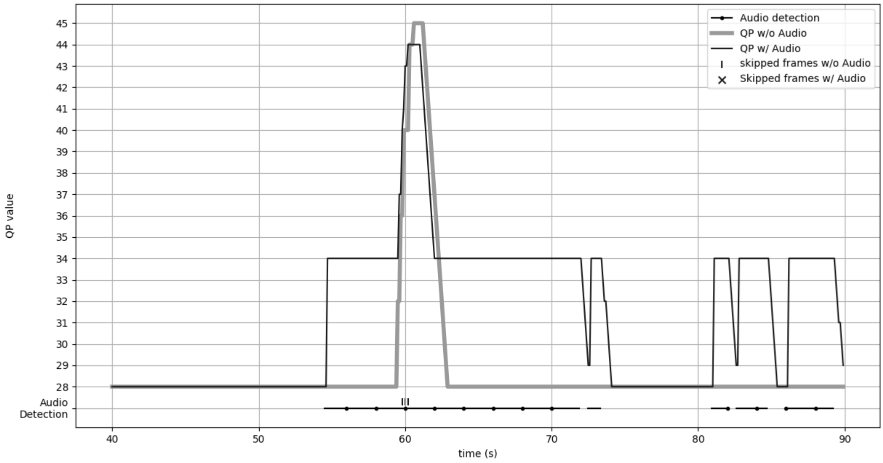
Figures 4.31a - 4.33b displays the QP value graph for each scenario when using the MD-GMM or RMS envelope. Just as in the previous sections, the minimum QP is increased to 34 when the audio regulation activates. It can be noted that the regulation behaves similar to how it did with the manually created detection signal. The main difference is that there is regulation performed outside of the event in some graphs. Most notable is the difference between Figures 4.32a and 4.32b, where the MD-GMM has many false detections which affect the RC toward the end of the clip.



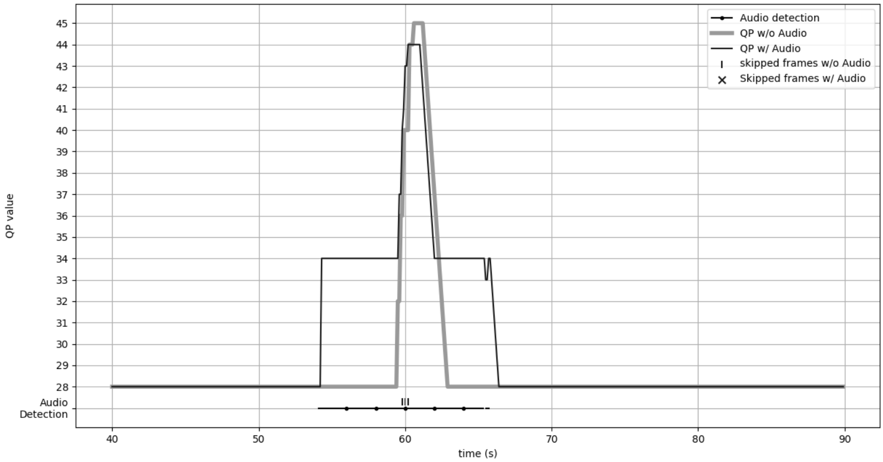
(a) With MD-GMM used for detection.

(b) With the RMS envelope used for detection.

Figure 4.31: Rate control on *Walking4* with bit rate limit 260 kbit/s. Both with and without audio detection.

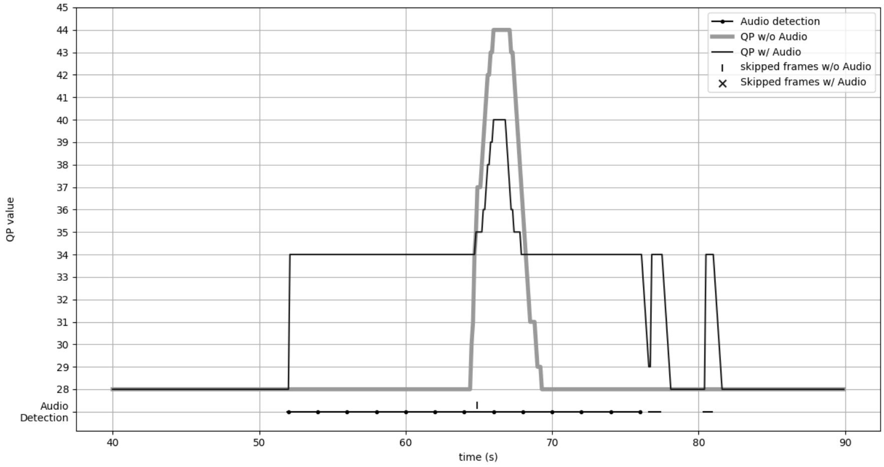


(a) With MD-GMM used for detection.

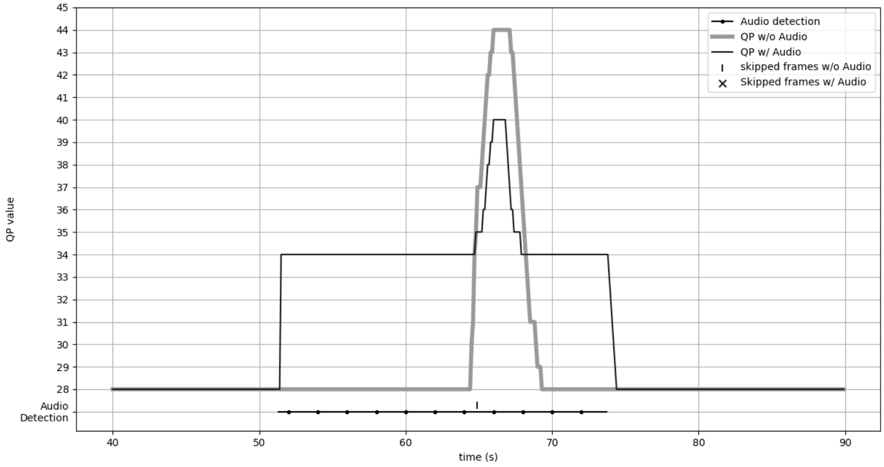


(b) With the RMS envelope used for detection.

Figure 4.32: Rate control on *Running4* with bit rate limit 500 kbit/s. Both with and without audio detection.



(a) With MD-GMM used for detection.



(b) With the RMS envelope used for detection.

Figure 4.33: Rate control on *Talking3* with bit rate limit 1000 kbit/s. Both with and without audio detection.

Table 4.13: Frame drop results when combining MD-GMM or RMS detection with minimum QP regulation.

Clip name	**Max bit rate**	**Number of dropped frames**		
	Kbit/s	Without audio regulation	Multi dim. GMM regulation	RMS Envelope regulation
Walking4	260	3	0	0
Walking5	260	8	4	4
Running4	500	3	0	0
Running5	500	3	0	0
Talking3	1000	1	0	0
Talking5	1000	1	0	0

Chapter 5
Discussion

In this chapter the results of all tests are discussed. First the results of the RC regulation is discussed. Then comes the audio event detection. Last comes the entire system results. In each section we discuss what the results tell us, the strengths and weaknesses of the algorithms and how to further improve upon these results.

5.1 Regulating with Ground Truth Detection

This section discusses results from increasing the minimum QP or increasing the maximum bit rate of the RC based on a warning signal. A ground truth detection signal was used, meaning it was as good as if a human was listening to the sound from the surveillance camera. The results which uses the ground truth should therefore be seen as proof of concept results to show if it is possible to improve the RC with a perfect detection. The two regulation methods are evaluated based on their performance as well as advantages and disadvantages they have.

5.1.1 Increasing the Minimum QP

In Table 4.4 it can be seen that for all test cases the number of dropped frames are reduced or completely removed when using minimum QP regulation. It should be noted that the bit rate budget was picked so that it would be slightly under the rate at which drops occur. Therefore these results do not show that the regulation scheme is beneficial at all bit rates. It does however imply that the scheme can be used to lower the required bit rate to avoid dropped frames in certain situations.

For most cases the regulated version reaches the maximum QP before the non-regulated version. This is due to the regulated version starting from a higher QP because of the minimum QP restriction. The head start makes a difference because the RC can only increase the QP by a few steps every iteration.

An exception to this is when the regulation lowers the peak QP reached, as seen in Figure 4.3. The regulated version reacts slower but will on the other hand catch the event with lower QP without frame drops. The reason is most likely due to the period of comparatively high quantisation before the visual event. This leads to the system having a large enough margin in its bit rate buffer.

There is however some misbehavior of the system when increasing the minimum QP. This occurs when the bit rate is far below what is required to handle the event. *Talking5* is an example of having one more dropped frame when using the warning system. Looking in Figure 5.1 it is seen that the QP is higher for the non-regulated stream when the drop occurs. Another effect seen is a small increase in QP just before the main increase. This may come from a shadow or reflection entering the view before the object itself, which affects the RC. It reacts strongly to this which will save bits at the cost of quality. The regulated RC does however not see the need to react as it is already slightly ahead. This leaves the RC with fewer bits available for the actual event compared to the original version.

A possible explanation for this behavior may be the two systems which the RC is built upon. As mentioned in the Methods the RC has two systems, one which regulates the QP value and a second that determines if frame drops are necessary. For this case it may be that the higher minimum QP will not cause any reaction in the first system. This forces the second system to have less margin due to the larger frame sizes. Then when the event occurs, the first system reacts slower than what is necessary for the second system to not drop frames. This behavior may be removed if the RC was constructed with the warning signal in mind in the first place. Therefore a continuation of this work is to incorporate an external warning signal into the RC.

All in all, the goal of having a more stable QP is achieved. But in this specific case it causes larger frame sizes which results in dropped frames. This example illustrates how difficult it is to improve the system in the general case, accounting for all small details that can differ from scene to scene, and scenario to scenario.

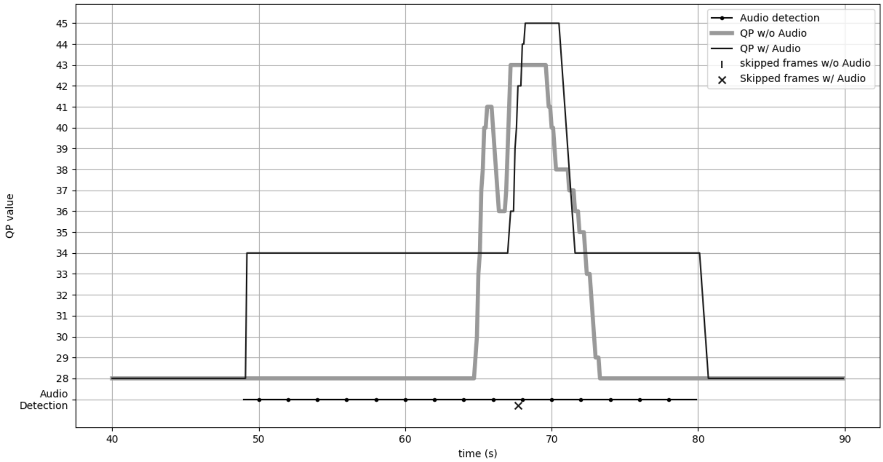


Figure 5.1: Talking5 regulated with min QP, pre-event is handled without increasing QP.

Disadvantages of Minimum QP Regulation

Minimum QP regulation may cause the image to be of lower quality before and after the actual event. This leads to worse quality when the audio detection activates. For the use case of a surveillance camera looking at a static scene this may not be too noticeable. However, simply setting a higher base QP from the beginning will achieve the same results.

Advantages of Minimum QP Regulation

One benefit of minimum QP regulation is that the base QP can be lower. That may be beneficial for capturing small motion events far away. At the same time it can handle large events which normally would lead to dropped frames by increasing the minimum QP as needed. One specific use case would be if the camera is surveying a long corridor. Close to the camera events will cause large motion, which may be detected by the audio detection and in turn warns the RC. For events far away in the corridor, the motion will be smaller, and a smaller QP may handle the motion. This makes the camera adapted for both types of events at the same time.

5.1.2 Increasing the Maximum Bit Rate

Results from regulating maximum bit rate shows that this also works well as a regulating method. The table suggests that the performance in terms of frame drops is the same as for minimum QP regulation. Just as with the minimum QP scheme all drops could be avoided except for 2 on *Walking5*. The main difference can be seen in the Figures 4.4 - 4.6. Here it is seen that the regulation will not affect the QP outside of the event. And when the event happens, it will mostly be lower than when not regulating.

An argument against this type of regulation is that it slightly cheats the problem. It is obvious that by increasing the allowed bit rate, more data can be sent. This allows for lower quantisation, and there is no longer a need for skipping frames. There is still some legitimacy to the strategy though, as ideally it would only be activated at important times.

Disadvantages of Maximum Bit Rate Regulation

This type of regulation may be sensitive to false positives by the audio event detector. Lots of detection will increase the average maximum bit rate for extended periods of time. That may result in a higher than expected bit rate which is not desirable as storage and transmission can already be crucial for surveillance systems.

Also, the maximum bit rate regulation is probably sensitive to oscillations in the detection signal. The maximum bit rate directly corresponds to the RC reference signal. This means when the bit rate is decreased, as when an audio detection event ends, there is a risk that the system adapts to the change poorly. With a high bit rate ceiling the RC will think that it has a large budget to spend. But when the detection switches it may suddenly be far over its budget, potentially causing a very high QP or frame drops.

Advantages of Maximum Bit Rate Regulation

The main benefit of increasing maximum bit rate when an event occurs is that there is more bandwidth available to capture the event. This in turn will result in that the event is captured in better quality, either by lowering the QP increase, or by reducing the number of dropped frames. This may be important for some sudden events where high quality is necessary. It may however be argued that the RC could regulate the maximum bit rate itself without any warning signal. As soon as it detects an increase in motion it could increase its maximum bit rate by itself.

There is a great benefit with this regulation if it is extended a bit. Often several surveillance cameras are connected to the same network where each camera streams it own video stream. Using audio detection may give the camera enough time to ask the network for more bandwidth ahead of an incoming event. When several cameras are installed in a large network this early warning may give them a chance to coordinate who gets network priority based on the likelihood of increased in motion. Audio detection may give time for this coordination to be done before the event is visible. This coordination of shared resources may allow for a larger amount of cameras on a shared network, or better quality for those installed.

5.1.3 Encoding Quality

The image quality after encoding with the updated QP regulation was investigated, which means that the effects of the QP fluctuations can be estimated. From Table 4.5 it can be concluded that the difference in average SSIM and VMAF is small when comparing the regulated and non-regulated video clips. Looking at VMAF, the scores are worse using the regulation system on the walking and talking clips. On the other hand the effect of the scheme is positive on the running clips.

The results in the table can be explained by observing the plots of frame specific scores seen in Figures 4.7 - 4.9. It can be observed that the quality scores decrease as soon as the audio detection activates as this increases the QP. The average is thus affected negatively by the drawn out detection of the walking and talking clips, and less affected by the relatively short detection in the running clips. Further, as seen in Figure 4.8a, the image quality is positively affected during a period of the running event, which brings a net improvement to the mean score showed in Table 4.5.

Besides the frame drops there seems to be two different phenomena that affect the image quality from frame to frame. The first is that the quality changes dramatically every time an I-frame is encoded. This happens every 32 frames, and can be seen as a sharp jump in both VMAF and SSIM score.

Secondly, during the most eventful periods of the clip the scores fluctuate in a more continuous fashion, and this has to do with how much detail each P-frame brings. During eventful periods the P-frames are more important, and the quality can then change even between I-frames. During the less eventful periods the P-frames barely bring any information, so a change of QP has little to no effect until an I-frame comes along.

Overall, the curves representing SSIM and VMAF with and without audio interference follow each other without much deviations, so it can be said that the overall quality does not differ heavily between the two regulation schemes. Keep in mind that it takes a drop of 20 VMAF points to lower a full quality grading [17]. With this in mind it can be said that before and after the events the quality grade is somewhere around ”fair”, regardless of if there is audio detection involved or not.

On the walking clip the grade goes down to ”poor” during the event with both systems. On the running clips the audio system fares much better compared to its counterpart, and on talking both systems end up somewhere between the grades ”poor” and ”fair”.

All in all, it is hard to say that the audio affected RC performs significantly better or worse than the normal RC. Rather, it is up to the end user and where their priorities lie. Perhaps it is worth the slightly lower video quality for the duration of the audio detection if this results in the possibility to catch the main event in higher detail.

5.1.4 RC Regulation Summary

To answer the research question, the performance of an RC can be improved in terms of a reduced number of dropped frames or smaller peak QP by using a warning signal. It is possible to regulate the RC by either increasing its minimum allowed QP or maximum bit rate. There may be some side effects where performance is reduced. These side effects may be mitigated by better incorporating a warning signal into the RC. The full potential of the audio detection warning is therefore not found in this project. Rather it can be seen as a proof of concept that an RC can take advantage of a warning signal to better predict motion in the upcoming frames. Looking at the end quality of the video stream after affecting the QP value it did not have any major impact in terms of VMAF and SSIM scores. The largest benefit is the reduced amount of dropped frames when streaming at low bit rates.

5.2 Audio Detection

This section discusses the different audio detection algorithms and related modules. First the strengths and weaknesses of the GMMs are discussed. After this an analysis of the envelope tracking is done.

5.2.1 Audio Detection with GMMs

Audio detection can be performed quite satisfactory with all versions of the GMMs. They have high accuracy for a foreground/background detection. Looking at the audio detection signal compared to the ground truth gives a similar picture. In most cases the detection is fast enough to be usable as a warning before the event enters the cameras field of view.

GMM Performance

The performance in terms of accuracy of all GMMs is above 90%. This should be seen as a satisfactory result where it will for most of the time do a correct detection.

Also, all GMMs have high performance in specificity, which implies a low number of false positives.

The sensitivity is especially low for the 1D-GMM in the walking scenarios. The detection is late and does not extend throughout the entire event. It stems from the nature of walking scenarios. They contain footsteps which are more silent compared to the other scenarios. Especially when happening far away from the microphone. As can be seen in Figure 4.13, the MD-GMM performs much better in this scenario. Most likely due to being able to model correlations between features. Although footsteps are quiet they are present in a wide range of frequencies and therefore in several MEL features. A small change in just one MEL band may be quite likely. But multiple small changes in several MEL bands simultaneously are less probable. The MD-GMM should be able to model that subtle difference, and this may be the reason why this version perform better in these scenarios.

The other scenarios seem to be easier to detect, probably due to them having relatively high intensities and thus standing out from the background more. Running generates loud noises for each foot step, and two persons talking is also a relatively loud event. This is seen in the detection results, as the performance on these events are high.

Regardless of algorithm used, the clips *Running2* and *Talking2* have a lower specificity compared to the others in the same scenario. This is the result of an audio detection that is not correlated to any event in the cameras field of view. This highlight that the algorithms are sensitive and will detect sound events which should ideally not be detected. A more sophisticated detection algorithm might be required to distinguish the different events.

GMM Adaptation

The GMMs are designed to be adaptive so that the distributions and weights change for each sample that is processed. This adaptation is useful as it enables the algorithms to handle a changing sound environment. However, the adaptation has to be balanced so that the GMM finds the new background fast enough, but still slow enough to not falsely categorise close successive events as background.

All GMMs more or less have the same performance on both consecutive runs. There are some exceptions though, namely *Walking2* and *Walking3* for 1D-GMM and *Running1* for MD-GMM. It may be from a change in their distributions after they have adapted more to the new clip. The post processing using KSLA may be involved by amplifying this change. An effect of the KSLA post processing is that a small change in classifications from the GMM, may change the class of a large part of the signal. This may function in both ways, either increasing or reducing the performance between runs. In relation to the other two GMM algorithms, the MD-GMM-UF has no major changes on its second runs. This was somewhat expected as it has a lower learning rate.

Foreground Categorisation

Performing any form of foreground categorisation from the GMMs looks like a difficult task. Looking at Figures 4.16 - 4.18 and 4.22 - 4.24 the difference in behavior between the MD-GMM and MD-GMM-UF can be observed. The two distributions used by the MD-GMM are not enough to do any categorisation. One distribution contains the background and the other contains the foreground. This makes it impossible to know what kind of foreground event that was detected.

For the MD-GMM-UF on the other hand has a greater usage of distributions which may be helpful. This shows a more detailed version on how the sound is constructed. If it is more dynamic such as speech there are more distributions used. This difference could possibly be used for a simple form of categorisation.

It should be noted however, that there is a big overlap in distribution usage between the events. Therefore it would not be straight forward to do the categorisation. Further, as speed is important for the RC preparation the algorithm needs to make a decision fast. From the figures it can be seen that all scenarios uses the same two distributions at the start of the event. They only differ during the peak of the event. Because of this there may not be enough time to find which type of event that is happening before it is too late. Finding and tuning features and developing an algorithm that could take advantage of the distribution information from this GMM is a big task, but it may still be possible to do.

Generalisation

A last thing to note is that the GMMs, regardless of the specific algorithm used, seem to generalize quite well. The accuracy on the unseen clips *Walking3* and *Running3* are similar to the accuracy on the training clips. Regarding *Talking2* a pattern is that all three algorithms are clearly somewhat worse compared to the other clips of the same scenario. The reason behind this is probably due to interfering noise in the audio signal. Figure 4.12, 4.15, and 4.21 all shows this. With this explanation for the slight decrease in accuracy on the test clip, it is concluded that the generalisation holds.

All clips used are however quite similar to each other. Them all having the same background scene and the same persons passing by, generating similar sound for each clip. With such a small and specific data set the risk of having an over-trained or over-optimised algorithms is high. This would be countered with more data that have varied scenes and scenarios. Due to limited time it was not possible to generate a large labeled data set with different scenes and sound environments. Therefore, to be able to clearly determine the accuracy and usefulness of these algorithms, more tests may be necessary.

5.2.2 Envelope Tracking

Both the power envelope and the RMS envelope have an high accuracy on the test clips. The sensitivity is not as good as for the MD-GMM but still quite high. While the specificity and accuracy is similar to the MD-GMM. Both algorithms have detection quite close to the ground truth. The performance is similar for both methods and both should be useful for the purposes of RC preparation. The most difficult scenario is *Talking2* which has some noise after the event, but still the actual event is captured well in both cases.

There are some benefits that comes from this kind of audio detection compared to a GMM based version. Because the detection of these algorithms are more consistent, none of the methods use any kind of post processing, such as the KSLA used by the GMMs. This is a very desirable trait as this leads to lower delay. For better and for worse, the envelope tracking is not as adaptive as the GMM algorithms. A positive effect of this is that it should result in more consistent results between runs. This is confirmed for the RMS envelope in Table 4.12. It is however not true for the power envelope, see Table 4.11, which has some small changes between the runs and especially for *Walking3*.

Looking deeper into why it is revealed that *Long4* and *Walking3* have different energy in their background sound. *Long4* has a lower volume compared to *Walking3* which causes the algorithms filters to ”fall behind” for a short while after transitioning. This makes the envelope tracker slightly more sensitive, and therefore the event is detected earlier in the first run. On the second run the algorithm has adapted to the new background volume, and the added sensitivity is lost. The different results stem more from a difference in the clips and a design choice to have a long settling time, rather than a bad adaption for the second run. It can be argued that the first run was too good and the true detection lies in the second run.

Envelope as a Feature

Based on the results of RMS envelope it seems to be a good feature of the sound. It outlines the energy content of the signal well. It could be that this feature can be incorporated into the GMM as a feature instead of being its own algorithm. The GMM could potentially combine the envelope feature with for example frequency based features to create a new and improved detection algorithm. This idea is not explored further in this project and instead left as future work.

5.2.3 Audio Detection Summary

Using a GMM as an audio classifier is a reliable way to detect events that stand out from the background sound in a surveillance context. Being able to model the background with normal distributions makes it possible to find outliers with a high accuracy. It was shown that the multidimensional variants outperform the one dimensional variant for the purposes of this project. Envelope tracking is also a possible audio detector. It removes the need of distributions and can have a similar performance. A benefit of the GMM is adaption included by updating the distributions. The envelope tracking is not adaptive in the same sense. While not fully evaluated in this work, a GMM might have potential to categorise audio based on which distributions were matched with. This information may be used for classification other than simple foreground/background detection.

5.3 Preparing the RC with Audio Detection

Results from putting the audio detection together with the RC regulation can be can be seen in Section 4.3. During these tests the RC is regulated by minimum QP due to this being a more promising method. The results are from the two best audio detection algorithms, namely the MD-GMM with original features and the RMS envelope tracking.

Despite the delays introduced by the algorithms both algorithms detect the event well before it enters the cameras field of view. This is especially important for the MD-GMM as it has a relatively long delay because of its post processing. Their detection are quite similar to the ground truth signal. Thus, the results are not too different compared to the results of using a ground truth detection signal. The number of drops remaining is mostly unchanged. The exception is *Walking5* where the audio based regulation had 4 drops, compared to the ground truth regulation with 2, and the unregulated having 8.

Running4 and *Talking3* shows some more oscillations in the QP value at the end due to some unwanted detection by the MD-GMM audio detection. This is undesirable and will affect the video stream quality in a negative way. Other than that the audio detection works well when it comes to detecting the event in time to start the regulation, and in terms of reducing the number of dropped frames.

5.4 Ethical Discussion

The technologies used in this project relate to surveillance and machine learning algorithms. Within these domains the ethical aspect is very important, as real people's lives can be affected by the resulting products. The inclusion of machine learning in video analytic creates the possibility for object detection and tracking, and further interpretation of their behavior. As the technology develops and becomes more wide spread the ethical implications have to be accounted for[38]. The use cases for surveillance are on a broad spectrum with everything from fall detection to detecting people in need of medical care[39] to employee surveillance to track productivity of workers[40]. As the potential increases the ethical discussion must follow to ensure an ethical usage with the right intentions.

Including an audio detection into a surveillance camera makes it a more potent surveillance tool. As mentioned before there is already a possibility to identify objects in a video. Combining this with audio detection to classify sounds and it may become even more effective in analysing a scene it surveys. As stated above, depending on the end user this can be used in a wide range of areas, possibly both positive and negative.

If the audio was analysed in greater detail and used to actually listen to people's conversations, we think it would be a much greater infringement to people's privacy. The specific algorithms in this project are however not aware of the exact content of the audio scene, only deviations from the background. With the GMMs it was found that there may be a possibility to differentiate between different types of audio events, but only into very broad classes. The main contribution of this work is to utilise audio information to enable greater video quality at low bit rates. This may allow for more cameras to be installed on the same network in the future. The impact of this could be positive if used responsibly.

When it comes to the development process used in this work, all data gathered is recorded by the authors. All video clips used are recordings of the authors themselves. An exception is that voices of the authors' colleagues sometimes can be heard at a distance. However, everyone involved have given their consent to be recorded.

In conclusion this project has been developed with the consent of everyone involved. The end product uses audio information to improve decision making regarding the encoding of the video stream, but it does not listen to any sensitive information. The improved encoding may enable more cameras to operate on the same network in the future. The effects of this, regarding safety and privacy, could vary depending on the goals and methods of the end users.

5.5 Future Work

There are many more topics that could be investigated in order to increase the accuracy of the results as well as increasing the performance and benefit of audio detection in a RC. These where mostly not tested due to limited time and resources.

New Implementation of Rate Controller

To better evaluate the potential of using audio to prepare the RC a new RC should be designed with the audio detection feature in mind from the beginning. In this case the RC could instead of changing external parameters such as minimum QP or bandwidth, better estimate the MAD or complexity of the next frame. Based on this estimation it could better calculate a QP that should be used to handle an upcoming event.

Another approach would be to create an outer control loop which could change the reference signal of the RC. Instead of changing the maximum bit rate, which was done in this project, the *desired* bit rate could be changed. Keeping the maximum bit rate static while changing the desired bit rate would allow the bit rate to be temporarily lowered as a way to save bits ahead of an event. The purpose is similar to the minimum QP strategy, but has the potential of being less crude, as the RC itself could decide how to achieve the lowered bit rate.

Foreground Categorisation

The current audio detection methods are primitive in the sense that the output is binary. However, it may be possible to create a better audio detection that could categorise different foreground events, and based on the category determine how likely it is to affect the camera. Feedback from the RC could be used to evaluate how relevant previously detected audio events were. After this evaluation certain types of detections may no longer require a reaction from the RC.

To accomplish the classifications necessary more sophisticated algorithms than used in this project would probably be necessary. Perhaps methods such as neural networks or SVMs. Due to limited amounts of data it was not possible to attempt these methods during this work. However, if more data is gathered it may be possible to create a better model that could very well be able to categorise foreground events in a way that is useful for rate control.

The envelope tracking detection may also take advantage of some form of categorisation by determining if the event is moving towards or away from the camera. This could be observed depending on how the envelope is changing.

Early Stopping Based on RC Feedback

In the section above we bring up the possibility of retrieving some kind of feedback from the RC that could affect future detection based regulations. A more direct usage of RC Feedback that was considered but never tested is to use changes of the QP signal as an indication of when to stop the regulation.

During an event it is usually the case that the minimum QP is increased, followed by a period of no action. Thereafter motion enters the stream and subsequently the QP increases. After some time the QP falls again and rests at the minimum QP, before the audio detection finally ends, and the minimum QP is lowered to its usual state. There could be a benefit in prematurely stopping the audio detection as soon as the QP falls to the altered minimum QP. As can be seen in for example Figure 4.9, the quality is worse for a prolonged time after the event and this could potentially be avoided with an early stopping strategy.

Better Audio Capture

This work have been based on the internal microphone of the camera and may be limited in capturing audio. A way of improving the audio detection may be to have better or external microphones. These could give a better or more accurate picture of the audio environment and in terms increase the performance and make it possible to categorise foreground.

More Extensive Data

Due to limited data available during this work, it may be necessary to further test the audio detection in environments not present in the test clips. Clips of more busy environments like outside or in a more populated place may be desirable to get a better picture on the algorithms strength and weaknesses.

Chapter 6

Conclusion

The main purpose of this work was to investigate the use of audio detection to warn an RC of upcoming events. It was found that an RC can take advantage of a detection signal to prepare for an incoming event. This can be done by either temporarily increasing the allowed bit rate, or by increasing quantisation before the event happens.

Raising the minimum QP of the RC during an audio event proved to be the most beneficial. In several cases this modification reduced the number of dropped frames, or lowered the maximum quantisation needed. However, modifying an RC to take audio into consideration this way may also introduce unwanted behavior, such as slow reaction times in certain situations.

The project also examines different methods of performing audio based foreground detection. One proposed method was a GMM based audio detection. A one dimensional GMM was implemented and had satisfactory results on the data used, with the exception of not being sensitive enough to the sound of walking foot steps.

Two variants of a multidimensional GMM have been implemented and the best variant has a 97% accuracy on the walking and running test clips, and 90% on the talking test. This variant is preferred over the one dimensional version.

As an alternative to GMMs two methods of envelope tracking were implemented, namely Power envelope and RMS envelope. Out of the two the RMS envelope achieves the best results, and is not far behind the multidimensional GMM.

The RMS, along with the MD-GMM were used together with the minimum QP scheme in a final evaluation. It was shown that both detect events fast enough to give the RC a warning and improve the performance in terms of frame drops.

Experiments also show that further differentiation between different foreground events may be possible, but are likely very hard to perform with the tools used in this project. It was therefore not possible to create as system that could separate events depending on the relevance to the RC.

References

[1] A. Martínez-Ballesté, H. Rashwan, D. Puig, and A. Solanas, "Design and implementation of a secure and trustworthy platform for privacy-aware video surveillance.," *International Journal of Information Security*, vol. 17, no. 3, pp. 279 – 290, 2018.

[2] D. Marpe, T. Wiegand, and G. J. Sullivan, "The h.264/mpeg4 advanced video coding standard and its applications," *IEEE Communications Magazine*, vol. 44, no. 8, pp. 134–143, 2006.

[3] I. E. Richardson, *H.264 and MPEG-4 video compression.* West Sussex: John Wiley & Sons Ltd., 2003.

[4] L. Wen, G. Xinbo, D. Qingeng, and W. Tisheng, "A basic-unit size based adaptive rate control algorithm.," *Fourth International Conference on Image and Graphics (ICIG 2007), Image and Graphics, 2007. ICIG 2007. Fourth International Conference on*, pp. 268 – 273, 2007.

[5] J. Minqiang, Y. Xiaoquan, and L. Nam, "Improved frame-layer rate control for h.264 using mad ratio.," *2004 IEEE International Symposium on Circuits and Systems (ISCAS), Circuits and Systems (ISCAS), 2004 IEEE International Symposium on, Circuits and systems*, vol. 3, 2004.

[6] M. Gerard and M. Tedenvall, "Adaptive content-based sound compression," 2019. Student Paper.

[7] ISO, "Information technology — Coding of audio-visual objects, Part 10: Advanced Video Coding," standard, International Organization for Standardization, Geneva, CH, 10 2004.

[8] I. E. Richardson and P. Ballantyne, “H264 video encoder,” 2021. [Online; accessed March 18, 2021].

[9] K. R. Rao, P. Yip, and P. Yip, *Discrete Cosine Transform. [Elektronisk resurs].* Academic Press, 1990.

[10] S. Hafizullah, M. S. S. V. Srikrishna Manideep, V. Sharma, P. Nath, A. Naugarhiya, and S. Verma, “An efficient hardware implementation of walsh hadamard transform for jpeg xr.,” *2018 15th IEEE India Council International Conference (INDICON), India Council International Conference (INDICON), 2018 15th IEEE*, pp. 1 – 4, 2018.

[11] Z. Li, F. Pan, K. Lim, X. Lin, and S. Rahardja, “Adaptive rate control for h.264.,” *2004 International Conference on Image Processing, 2004. ICIP ’04., Image Processing, 2004. ICIP ’04. 2004 International Conference on, Image processing*, vol. 2, p. 745, 2004.

[12] F. Wei and Z. Shanan, “A robust and adaptive rate control algorithm for real-time video communications.,” *2006 6th World Congress on Intelligent Control and Automation, Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on*, vol. 2, pp. 10220 – 10224, 2006.

[13] M. Santamaria, E. Izquierdo, S. Blasi, and M. Mrak, “Estimation of rate control parameters for video coding using cnn.,” 2020.

[14] Y. Blau and T. Michaeli, “Rethinking lossy compression: The rate-distortion-perception tradeoff.,” 2019.

[15] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[16] R. Rassool, “Vmaf reproducibility: Validating a perceptual practical video quality metric,” in *2017 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pp. 1–2, 2017.

[17] Z. Li, C. Bampis, J. Novak, A. Aaron, K. Swanson, A. Moorthy, and J. De Cock, “Vmaf: The journey continues.” `https://netflixtechblog.com/vmaf-the-journey-continues-44b51ee9ed12`, 2018. [Online; accessed 11-May-2021].

[18] M. Cristani, M. Bicego, and V. Murino, “On-line adaptive background modelling for audio surveillance,” in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol. 2, pp. 399–402 Vol.2, 2004.

[19] S. Lecomte, R. Lengellé, C. Richard, F. Capman, and B. Ravera, "Abnormal events detection using unsupervised one-class svm - application to audio surveillance and evaluation -," in *2011 8th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 124–129, 2011.

[20] C. Tarjano and V. Pereira, "Robust digital envelope estimation via geometric properties of an arbitrary real signal," *arXiv preprint arXiv:2009.02860*, 2020.

[21] O. C. Carrasco, "Gaussian mixture models 3 clusters," 2019. [Online; accessed March 18, 2021].

[22] S. Moncrieff, Svetha Venkatesh, and G. West, "Persistent audio modelling for background determination," in *2005 IEEE International Conference on Multimedia and Expo*, pp. 4 pp.–, 2005.

[23] O. C. Carrasco, "2 component gaussian mixture model," 2019. [Online; accessed March 18, 2021].

[24] D. Ververidis and C. Kotropoulos, "Gaussian mixture modeling by exploiting the mahalanobis distance," *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 2797–2811, 2008.

[25] Wikipedia contributors, "Mahalanobis distance," 2021. [Online; accessed 22-February-2021].

[26] S. Moncrieff, S. Venkatesh, and G. West, "Online audio background determination for complex audio environments," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 3, p. 8–es, May 2007.

[27] T. Gonzalez, "Clustering to minimize the maximum intercluster distance.," *Theoretical Computer Science*, vol. 38, no. 2, pp. 293 – 306, 1985.

[28] M. Marzinzik and B. Kollmeier, "Speech pause detection for noise spectrum estimation by tracking power envelope dynamics.," *IEEE Transactions on Speech and Audio Processing, Speech and Audio Processing, IEEE Transactions on, IEEE Trans. Speech Audio Process*, vol. 10, no. 2, pp. 109 – 118, 2002.

[29] Wikipedia contributors, "Root mean square," 2021. [Online; accessed 21-April-2021].

[30] V. Peltonen, J. Tuomi, A. Klapuri, J. Huopaniemi, and T. Sorsa, "Computational auditory scene recognition," in *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. II–1941–II–1944, 2002.

[31] S. Moncrieff, S. Venkatesh, and G. West, "Unifying background models over complex audio using entropy," in *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 4, pp. 249–253, 2006.

[32] Wikipedia contributors, "Hearing range," 2021. [Online; accessed 29-April-2021].

[33] L. Muda, M. Begam, and I. Elamvazuthi, "Voice recognition algorithms using mel frequency cepstral coefficient (mfcc) and dynamic time warping (dtw) techniques," *J Comput*, vol. 2, 03 2010.

[34] M. Hossan, S. Memon, and M. Gregory, "A novel approach for mfcc feature extraction.," *2010 4th International Conference on Signal Processing and Communication Systems, Signal Processing and Communication Systems (ICSPCS), 2010 4th International Conference on*, pp. 1 – 5, 2010.

[35] IPVN Team, "Frame rate guide for video surveillance," 2021.

[36] U. Zölzer, *Digital Audio Signal Processing*. Wiley, second ed., 2008.

[37] M. Thorogood, J. Fan, and P. Pasquier, "Soundscape audio signal classification and segmentation using listeners perception of background and foreground sound," *Journal of the Audio Engineering Society*, vol. 64, pp. 484–492, 08 2016.

[38] A. A. Adams and J. M. Ferryman, "The future of video analytics for surveillance and its ethical implications.," *Security Journal*, vol. 28, no. 3, pp. 272 – 289, 2015.

[39] L. Anishchenko, "Machine learning in video surveillance for fall detection.," *2018 Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT), Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT), 2018 Ural Symposium on*, pp. 99 – 102, 2018.

[40] L. Stark, A. Stanhaus, and D. L. Anthony, ""i don't want someone to watch me while i'm working": Gendered views of facial recognition technology in workplace surveillance.," *Journal of the Association for Information Science & Technology*, vol. 71, no. 9, pp. 1074 – 1088, 2020.