H.264 Video Frame Size Estimation

Edpalm, Viktor; Martins, Alexandre; Maggio, Martina; Årzén, Karl-Erik

2018

Document Version:
Publisher's PDF, also known as Version of record

H.264 Video Frame Size estimation

Viktor Edpalm[1]
Alexandre Martins[1,2]
Karl-Erik Årzén[2]
Martina Maggio[2]

[1]Axis Communications, Lund, Sweden

[2]Department of Automatic Control, Lund University, Sweden

LUND
UNIVERSITY

Department of Automatic Control

Authors' emails and affiliations:
Viktor Edpalm, viktor.edpalm@axis.com and Alexandre Martins, alexandre.martins@axis.com, are with Axis Communications, Lund, Sweden. Karl-Erik Årzén, karlerik@control.lth.se, and Martina Maggio, martina.maggio@control.lth.se, are with Automatic Control, Lund University, Sweden.

Technical Report TFRT-7654
ISSN 0280–5316

Department of Automatic Control
Lund University
Box 118
SE-221 00 LUND
Sweden

Printed in Sweden by Dept. Automatic Control, Lund University.
Lund 2018

Abstract

This report describes a method to estimate the video bandwidth for IP cameras using the H.264 standard. The precise determination of bandwidth allows us to model the network access as a scheduling problem and/or estimate the amount of data that would traverse it during different periods. The paper is written to be as didactic as possible and presents a set of experiments, conducted in an industrial testbed, that validate the estimation. We believe that a more precise estimation will lead to savings for network infrastructure and to better network utilization.

Contents

1. Introduction 7
2. Nomenclature 7
3. Brief overview of the H.264 standard 8
4. Metrics measurements 9
5. Real-Time Network Scheduling 13
6. Video frame sizes prediction 14
7. Experimental results 21
8. Conclusion 29

References 29

A. Metrics detail 31

1. Introduction

Video surveillance systems are more and more prevalent in today's society, they are used at different levels and at different scales (cities, public places, companies, homes...). These systems become more and more heterogeneous, complex and interconnected. Previously only big institutions were equipped with such surveillance systems and had dedicated infrastructures (network and storage units) to handle the expected load of information produced. However, today's systems are usually sharing the same network infrastructure as other network users and devices, even sometimes running over the cloud or on the internet with disseminated recording installations that could be in another continent. In light of these changes, predicting with accuracy how much data such systems will produce becomes increasingly important.

A typical video surveillance system comprises multiple different cameras (which can be fixed or Pan-Tilt-Zoom cameras) disseminated over an area and recording 24/7 a specific scene (office space, parking lot, road, etc). The scene can differ but its characteristics will not evolve significantly over time. A common challenge in the video surveillance industry is to be able to tailor a system in order to achieve a certain level a quality while keeping the cost as small as possible. This is often done when designing the system and choosing the cameras and then rarely, if not never adjusted during run time. Today the video industry is mainly focused on using IP video cameras which stream video compressed using the H.264 standard. In order to tailor a system one must be able to anticipate how much data each device in the system is expected to produce given its unique set of internal characteristics and settings, position, placement, surrounding environment etc. In this paper we propose a set of simple measurable parameters which combined allow to predict the expected H.264 frame sizes and by extension the video bandwidth. It proposes a method to tailor the storage and/or network bandwidth required for a surveillance system in advance using information about which cameras will be used and under which conditions. This paper aims to be as didactic as possible, we will start by briefly introducing the H.264 standard and then continue by introducing different variables and how to measure them. We then illustrate how to combine measured and estimated variables for intra-frames, inter-frames and within a video and conclude with a comparison of our proposed model with a set of measurements done with cameras in different situations as well as comparison with some other known state of the art techniques. For simplicity reasons we added a part in the appendix that enumerates the contributing metrics in the paper and could be used as a reference when implementing the algorithm and measuring variables.

2. Nomenclature

In this part we will list the metrics used in this paper. You can find a more in depth description in Appendix A. If not stated otherwise, all the metrics listed are in the $\mathbb{N}$ space.

Bandwidth (B): Video bandwidth (amount of bits per second generated by the video).

Camera detail properties (D_C): Constant reflecting the camera capacity to retain scene details.

Camera motion cost (MC): This is a temporary variable modeling the camera encoder capacity to encode motion.

Camera noise (N_C): Constant indicating the amount of noise in the camera.

Compression: Temporary variable indicating how much information the video was configured to lose.

Dynamic range factor (DR): Indicates if the video is using high dynamic range (HDR) or similar technology.

Frame rate (FPS): Number of frames per second of the video.

Frame size (F): Average frame size. This value is provided in kilobits per second (kb/s).

Group of pictures (GOP): Number of frames between two consecutive I-frames.

Height (h): Number of pixels in the shortest image plane (usually y axis).

I-frame size (I): Size of I-frames. This value is provided in kilobits per second (kb/s).

I-frame size constant (I_c): Temporary constant for I-frame size (only used for didactic explanation).

I-frame details size ($I_{details}$): Temporary constant for I-frame size (only used for didactic explanation).

I-frame noise size (I_{noise}): Temporary constant for I-frame size (only used for didactic explanation).

Motion encoder efficiency (M_{EC}): Encoder related constant which reflects the ability of the encoder to efficiently encode moving objects.

Motion level (ML): Part of the image that is expected to be moving.

Nature factor (N_F): Amount of nature (trees, bushes, grass, etc.) present in the scene.

P-frame size (P): Size of P-frames. This value is provided in kilobits per second (kb/s).

P-frame size constant (P_c): Temporary constant for P-frame size (only used for didactic explanation).

P-frame details size ($P_{details}$): Temporary constant for P-frame size (only used for didactic explanation).

P-frame noise size (P_{noise}): Temporary constant for P-frame size (only used for didactic explanation).

QP: Compression parameter defined in the H.264 standard.

ΔQP: Difference between the QP used and the reference QP used during measurement of some needed constants.

Reference frame rate (FPS_{ref}): Number of frames per second used as reference during measurement of some needed constants.

Reference QP: Compression parameter used during measurement of some needed constants (see QP).

Resolution: We consider resolution as the number of points (pixels) in the image.

Scene detail level (D_S): Constant indicating the amount of detail in the scene.

Scene illumination (L): Constant indicating the luminance (amount of light) in the scene.

Size of Average Object (SAO): This metric reflects the expected distance of an object in the image.

Width (w): Number of pixels in the longest image plane (usually x axis).

3. Brief overview of the H.264 standard

H.264 or MPEG-4 Part 10, Advanced Video Coding (MPEG-4 AVC) is a block-oriented motion-compensation based video compression standard. The standard only defines the decoder and stream, the way the encoder is implemented is left to the manufacturer's discretion. Its first official version was approved in March 2003 [ISO/IEC MPEG & ITU-T VCEG, 2003] and evolved over time by adding more features and modes, the latest version being approved in April 2017 [ITU-T, 2017]. The MPEG LA organisation administers most of the licenses for patents applying to this standard.

3.1 Frame types

An H.264 stream contains a sequence of frames. The display order does not necessarily correspond to the encode order, i.e. the order the frames are used is not necessarily the order which they are shown in the video. An H.264 video can contain three types of frames:

- I-frames. Self contained, they are encoded using information contained within the I-frames themselves. The I-frames contain the full images and do not require any additional information to reconstruct them (if it is an Instantaneous Decoding Refresh frame).
- P-frames. Encoded using information from the frames and previous (older) frames. In a P-frame, part of an older image can be referenced and used for encoding.
- B-frames. Encoded using information from the frames, previous (older) and future frames. In a B-frame, part of an older and future (in display order) image can be referenced and used for encoding.

Note: In this document we do not highlight the contribution of B-frames. We consider B-frame size prediction to be close to P-frame size prediction and as such a similar model can be used.

3.2 Group of pictures

The H.264 standard defines the group of pictures (GOP) as the number of frames between two consecutive I-frames. A GOP consists of an I-frame followed by P and/or B-frames. If the I-frame is marked as IDR (Instantaneous Decoder Refresh) it means that the coming frames do not contain any information about frames prior to the IDR frame. If all I-frames are IDR frames then each GOP decoding is independent. An example of GOP structure is illustrated in Figure 1

3.3 Macroblocks

All H.264 frames are split into 16x16 pixel squares named macroblocks. Each macroblock is encoded separately. Macroblocks can be split into sub-blocks down to 4x4 pixels, see Figure 2.

Macroblocks can be of the same types as the frame types enunciated in Section 3.1. An I-frame can only contain I-blocks, a P-frame I- and P-blocks and a B-frame I-, P- and B-blocks. Macroblocks encoding is briefly illustrated in Figure 3.

4. Metrics measurements

As briefly listed in the nomenclature (Section 2) and in more details in Appendix A, different metrics need to be measured at different level/interval in order to be able to use the paper's equation. We provide here a summary of these metrics as well as a short list in Table 2.

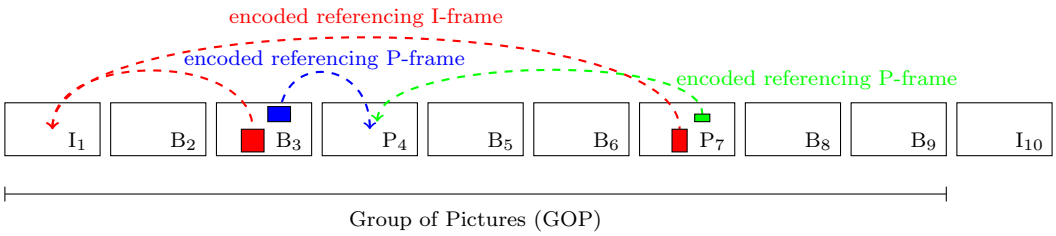


Figure 1 H.264 frame sequence: I-frames, P-frames, B-frames, and Group of Pictures.

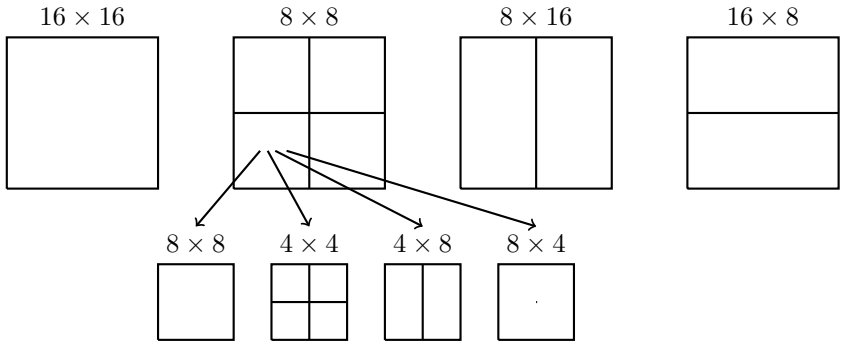


Figure 2 H.264/AVC macroblock division.

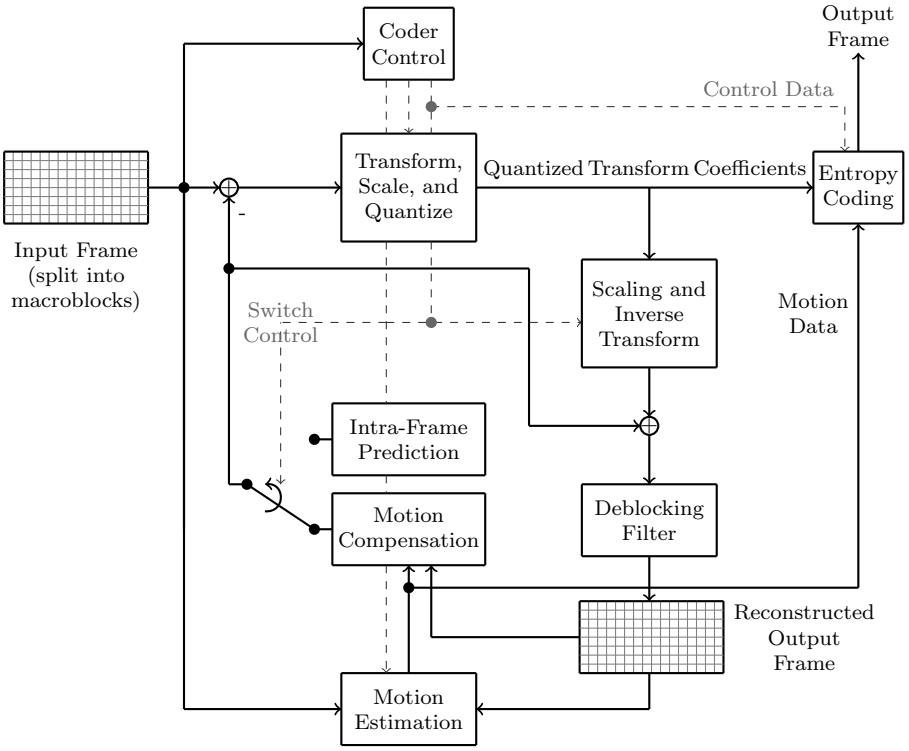


Figure 3 Basic coding structure of a H.264/AVC macroblock.

4.1 Platform specific measurements

For each different encoder generations, brand, etc (what we call platform) there are two parameters that need to be estimated: Camera motion cost (MC) and Motion encoder efficiency (M_{EC}). This is done by isolating the encoder, or an equivalent encoder model, with a series of video sequences which are encoded using varying compression. The Motion encoder efficiency reflects how good the encoder can detect and encode motion. In order to determine this value we need to feed the encoder with videos with known motion of different length/speed and see how many motion misses there are.

4.2 Laboratory environment

Measurements need to be done in a reproducible environment, this could be a dedicated lab or simply a box with controllable parameters. The main idea is to be able to reproduce certain scene conditions. The environment should contain different levels of details (areas with few/no details as well as some with details). The scene should also have the possibility

Table 2 Characteristics summary.

	Platform	Camera model	Scene/video	Frame
Camera detail properties (D_C)		✓		
Camera motion cost (MC)		✓		
Camera noise (N_C)		✓		
Compression/QP				✓
Dynamic range factor (DR)			✓	
Frame rate (FPS)			✓	✓
Group of pictures (GOP)			✓	✓
h, w, resolution			✓	
Motion encoder efficiency (M_{EC})	✓			
Motion level (ML)				✓
Nature factor (N_F)			✓	
Scene detail level (D_S)			✓	
Scene illumination (L)			✓	✓
Size of Average Object (SAO)			✓	✓

to control luminosity (high, medium, low) and have some reproducible motion (using a fan, toy train, etc). The position of the camera relative to the test scene should also be fixed. An example of laboratory is illustrated in Figure 4.

4.3 Reference camera and parameters

Most of the measurements needed are used to compare a specific camera/platform to a known reference camera/platform. The reference camera should be a camera that you know/use the most and that is easy to access in the case of having a change of laboratory environment. The reference camera is also the one we should have the most data points from.

4.4 Camera model specific measurements

Measurement scenarios Three parameters need to be measured for each camera model: Camera detail properties, Camera noise and Camera motion cost. These are determined using data retrieved from the laboratory setup by recording scenes with no motion, motion, details, no details and at three different light levels (night light, daylight and high light). You should do that with the chosen reference compression value chosen (see Section 4.3). Remember that the scene must be repeatable (see Section 4.2). Then run a software to extract frame sizes for all I and P frames in each video. When you have the statistics for the videos, plot the average I-frames and P-frame sizes for different conditions.

Camera detail properties Get the average frame size value (I and P-frames included) for the chosen compression level and compare it with the one you have for the reference camera. This ratio gives you the Camera detail properties (D_C).

Camera noise In order to find the Camera noise parameter (N_C), simply do the same as for D_C but taking in account only low light videos in your data. N_C is still the ratio between the reference camera's N_C and your current model.

Note: The D_C and N_C parameters of the reference camera are considered to be 1.

Camera motion cost To find the Camera motion cost (MC) value, simply multiply the M_{EC} found in Section 4.4 and D_C measured previously.

Figure 4 Example of an image laboratory.

4.5 Scene specific measurements

Some parameters to be measured at scene level, such as frame rate, group of pictures length, resolution are self-explanatory, we will not describe them here.

Dynamic range factor The dynamic range factor (DR) represents the amount of data added by the dynamic range feature. To measure this parameter, record the scene type your camera should be used in with HDR activated and deactivated. Then compare the average I-frame sizes with or without HDR, i.e.,

$$DR = \frac{average(Iframesize_{HDRon})}{average(Iframesize_{HDRoff})} \tag{1}$$

Motion level The Motion Level (ML) represents roughly how much of the image is moving. A scene with no movement would then have a motion level of 0.0, a scene where the camera is rotating would have a motion level of 1.0. This can be estimated or measured in real time with motion detection algorithms for example.

Nature factor The Nature Factor (N_F) represents roughly how much of the image is from nature scenery (trees, grass, bushes, etc) and susceptible to present wind. This parameter is considered because of the specific behavior of such objects in a video. An office space would have a nature factor of 0.0 (the beautiful orchid on your colleague's desk does not count) while a forest scene would have a nature factor of 1.0. As an example, an outdoor parking lot surveillance scene presents usually a nature factor between 0.1 and 0.5 (there are usually some trees and grass in the video).

Scene detail level The reference camera (see Section 4.3) is used to measure the scene detail level (D_S). There are two possibilities: go where the target camera (the one we want to predict the size from) will be used and measure directly on site or, record similar type of scenes (parking lot, office space, train station, etc) multiple times (at least 5) and reuse the average measured parameter for future similar scene types. The scene detail level is computed by taking the average I-frame sizes (in millibits/px) and using it directly. If you have a good knowledge of the specific camera parameters, you can extrapolate this parameter from the other parameters. This is not covered in this document.

Scene illumination The scene illumination (L) is measured by comparing the lab results we took in Section 4.4. We simply calculate the I-frame average for different scenes illuminations. We then consider the high illumination scene to be 1.0 and the scene illumination (L) is simply the ratio between the average I-frame sizes in high illumination and the illumination we consider.

Size of average object To find the size of average object (SAO) we can compare the average I-frame sizes for different zoom levels. The easiest way to find it is to put the camera on site and record videos with different zoom levels. We pose:

$$SAO = \frac{average(Iframesize_{zoom50\%})}{average(Iframesize_{consideredzoom})} \tag{2}$$

The SAO levels are for simplicity divided into three levels, large, medium and small. As a general rule of thumb, one can determine the SAO level for a 1080p video as:

- Large SAO. Objects taking up more than 1% of the pixels. An example is a licence plate camera, commonly setup to capture mainly a car with sufficient margin around it.
- Medium SAO. Objects are between 1% and 0.01% of the pixels. This is the normal case and usually sufficient for identification purposes.
- Small SAO. Objects are very small, less than 0.01% of the pixels. This is sufficient only for scene awareness, i.e. knowing what happened in the scene.

Note that the SAO effect is linked to the resolution, but the assumption here is that one will use a higher resolution camera on a larger view, rather than increasing the resolution of the current scene (in which case one either was using a too low resolution camera in the first place or one is now using an unnecessarily high resolution camera).

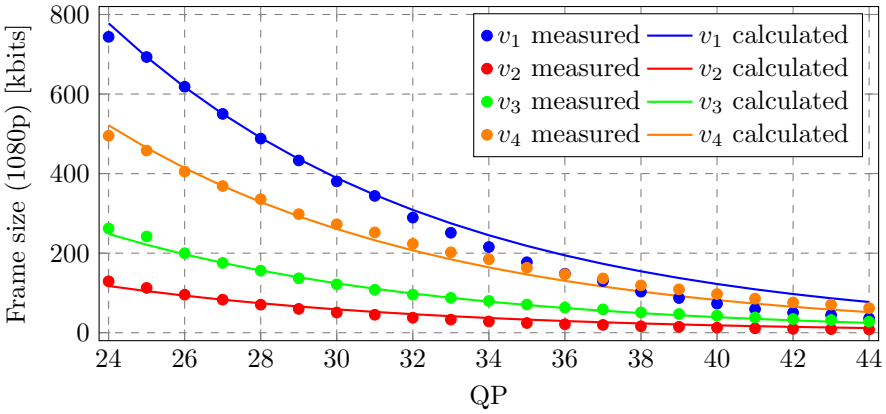
5. Real-Time Network Scheduling

Assume it is possible to compute an upper bound for the size of I-frames, denoted with I^* and an upper bound for the size of P- and B-frames, denoted with P^*. Knowing the network speed $\mathcal{N}$, e.g., 100 Mbps, one can then then translate these bounds into knowledge of the Worst Case Transmission Time (WCTT) for the two types of frames in the network. The GOP parameter specifies how many "dynamic" (P- and B-) frames there are in between two "static" (I-) frames.

In fact, when a set $\mathcal{C} = \{c_1, \ldots, c_p\}$ of p surveillance cameras share the same network, one can say that the i-th camera behaves according to the multiframe task model [Mok and Chen, 1997]. The camera has a vector of execution times $[E^0, E^1, \ldots E^{\mathrm{GOP}-1}]$ and a single period and deadline, equal to the inverse of the frame rate $1/f_i$. C^0 is then equal to the upper bound on the transmission time of the I-frame $I^*/\mathcal{N}$ and all the other execution times $[E^1, \ldots E^{\mathrm{GOP}-1}]$ are equal to the upper bound on the transmission time of the P-frame, i.e., $P^*/\mathcal{N}$. This allows us to reuse theoretical results developed for the specific model [Zuhily and Burns, 2009; Han, 1998; Baruah et al., 1999b; Lu et al., 2007] or for its generalizations [Baruah et al., 1999a; Peng and Fisher, 2016; Stigge et al., 2011; Li et al., 2014; Zeng and Di Natale, 2013; Ekberg et al., 2015; Chakraborty and Thiele, 2005]. In particular, once we have determined the WCTTs for the different frame types, we can use the analysis on non-preemptive scheduling of multiframe tasks [Andersson et al., 2012; Baruah and Chakraborty, 2006] to determine schedulability properties for a set of video-surveillance cameras communicating over switched Ethernet [Andersson, 2008].

As video encoders are very complex software elements, we cannot really compute an upper bound with static analysis or formal methods, that would guarantee that the size will never exceed the one predicted. However, we can compute an approximation of such upper bound, that is proven conservative in most cases. We believe that the very few circumstances in which the size of frames exceeds the computed values are due to problems and bugs of the execution of video-surveillance software. We then proceed in the discussion by finding reasonable estimations for the frame sizes.

Figure 5 Measured I-frame sizes and calculated ones for different videos, varying QP.



6. Video frame sizes prediction

6.1 Intra frame model

General assumptions A naive approach to approximate the Intra-frame (I-frame) size I is to consider it to be constant such that

$$I = I_c. \tag{3}$$

Adding the contribution of the frame pixel resolution to the size of the Intra-frame introduces a proportional relationship to Eq. 3

$$I \propto resolution. \tag{4}$$

The resolution is defined as the number of pixels in the image, which is expressed as the product of the video width and height (in pixels) such that

$$resolution = w \times h. \tag{5}$$

The video bandwidth is controlled by a compression level, denoted by the Quantization Parameter (QP) in H.264. By extension the I-frame sizes will also be influenced by this parameter.
Note: this parameter is assumed to be constant within the frame, while advanced H.264 encoders can spatially vary its value within a frame.

$$I \propto compression \tag{6}$$

From the H.264 standard, we can infer that "an increase of 1 in QP corresponds to an increase of the quantization step size by approximately 12% (an increase of 6 means an increase of the quantization step size by a factor of 2) [Wiegand et al., 2003].

$$compression = 2^{-\Delta QP/6}. \tag{7}$$

The ΔQP factor is used to scale the frame sizes between two compression levels. We select $QP = 28$ as the base QP from which all frame sizes are scaled. The choice of this base QP is arbitrary, but it is nice from a practical point of view to have it at a reasonable and commonly used QP value.

Including Equations 3-7 the I-frame size can then be defined as

$$I = I_c \times w \times h \times 2^{-\Delta QP/6}. \tag{8}$$

The quantization parameter (QP) could be different within the same frame in order to optimize spatial compression, this can be another parameter to consider but this possibility will not be considered here. A comparison of the measured I-frame sizes and the calculated one is shown in Figure 5

Table 3 Measured illumintation factor as I-frame size relative I-frame size at high illumination. Values are averaged over a QP range of 14-46.

Camera	L medium illumination	L low illumination
Cam A	0.806	0.542
Cam B	0.879	0.587
Cam C	0.781	0.281
Cam D	0.773	0.607
Average	**0.8**	**0.5**

Scene and camera related factors Now that a general encoder-related model has been determined, one can expand it by including camera and scene parameters. We will then refine the proposed I_c constant into noise and details terms such as

$$I_c = I_{details} + I_{noise}. \tag{9}$$

The detail part is proportional to the level of detail in the image, which is greatly influenced by the illumination of the surroundings, more light will indeed allow for more parts of the scene to be visible and vice verse complete darkness will hide details in the scene.

$$I_{details} \propto \text{scene detail level } (D_S). \tag{10}$$

$$D_S \propto \text{scene illumination } (L) \tag{11}$$

The scene illumination factor will be a function with a decreasing value as the illumination level goes down. For user simplicity, this is reduced into three discrete steps representing

- High Illumination. Daylight scenario or a well lit indoor environment such as an office or a store.
- Medium Illumination. Nighttime scenario with some light source illuminating the scene.
- Low Illumination. Nighttime without any major light sources.

Table 3 contains a set of measurements forming our scene illumination factor. Since the high illumination is used as a basis for most other measurements and L will be a relative scaling between the illumination levels, we furthermore define

$$L_{highillumination} \equiv 1. \tag{12}$$

Note that the D_S is a measurable quantity describing the amount of details in a scene. Experimentally used values for some scenes are provided in Table 4

The scene detail level is also highly correlated with the amount of nature in the scene (lawns, bushes, trees, etc). This factor is included since a high level description of a scene (e.g. a road) can leave out the amount of nature, which must be indicated for a good I-frame size prediction. Examples are shown in Table 5.

$$D_S \propto \text{nature factor } (N_F). \tag{13}$$

Another important factor affecting the I-frame size is the size of typical objects and details in the scene. This is parameter simplification based on a combination of the distance to the scene, the zoom level and the field of view. The effect of this is to reduce the I-frame size for scenes where the objects are large, since the amount of details in a typical

Table 4 Measured scene detail level for a collection of scenes in mBits/pixel.

Scene	D_S
Highway	1200 mbits/pixel
Office	820 mbits/pixel
Parking Lot	780 mbits/pixel
Retail	1800 mbits/pixel
Intersection	1050 mbits/pixel
Onboard	920 mbits/pixel
Reception	810 mbits/pixel
ATM	850 mbits/pixel
Street Corner	990 mbits/pixel
Pedestrian zone	1300 mbits/pixel
Perimeter	660 mbits/pixel
Busy Station	1500 mbits/pixel
Emergency Exit	710 mbits/pixel
Checkout Line	1280 mbits/pixel
Mall	1400 mbits/pixel

Table 5 Measured scene bitrates for similar scenarios (Parking, Highway, etc.) with and without nature (compensated for QP).

Scene	D_S
Without Nature	1130 mbits/pixel
With Nature	1408 mbits/pixel
Relative Nature Factor	125%

Table 6 Measured average I-frame sizes from 42 parking scenes in mbits/pixel.

SAO	**I-frame size**	**Ratio**
Large	351 mbits/pixel	0.45
Medium	773 mbits/pixel	1.00
Small	848 mbits/pixel	1.10

surveillance video object usually do not scale with resolution. Example values are provided in Table 6.1.

$$D_S \propto SAO. \tag{14}$$

The camera properties are included into the equation by taking into account the sensor type, lenses properties, etc. These are gathered into a single constant D_C. This constant will represent how well the camera captures the details of the scene and sharpens them.

$$I_{details} \propto \text{camera detail properties } (D_C). \tag{15}$$

Note that D_C is a measurable quantity describing the amount of details captured by each camera model/type. Both D_S and D_C are measurable quantities, D_S is measured for a chosen standard camera receiving a $D_C \equiv 1$ while the other cameras will be have their D_C measured relative the standard camera. See some example cameras in Table 7.

The dynamic range of the scene, together with the camera's ability of capturing it through various HDR techniques is modelled using the dynamic range factor, DR. If one assumes

Table 7 Measured camera sharpness, D_C using a standardized test scene.

Camera	Camera Sharpness
A	1.00
B	0.98
C	1.23
D	0.54
E	0.81
F	1.03

that the different light ranges have the same bitrate characteristics and that the camera auto exposure will select the range filling the most pixels then $DR \in [1, 2]$. The edge cases are a scene with no additional dynamic range to capture such as an indoor scene or a foggy day scene, which will have a $DR = 1$ and a scene where half the scene is low dynamic and half the scene is high dynamic such as an indoor scene with large windows, which will have $DR = 2$. An average value for all real world scenarios is something in between our edge cases and the tested HDR cameras had on average a 35% larger I-frame size. Note that the test data has some bias for higher dynamic range scenes since more of those are likely to trigger a user to turn on HDR, however that bias is likely cancelled out by the fact that a user is more likely to be turning HDR on in a higher dynamic range scene.

$$D_C \propto \text{dynamic range factor } (DR). \tag{16}$$

Note that many HDR cameras also have a negative impact on the video noise level since a multiexposure HDR solution will incur more sensor noise. This effect is not modelled in this report.

For the noise part of I_c, we consider that the camera generated noise (sensor, line, etc) is the sole contributor.

$$I_{noise} \propto \text{camera noise } (N_C). \tag{17}$$

We consider that the total noise amount is in direct relation to the scene noise level, the more light there is (before sensor saturation), the more photons the sensor receives and the less noticeable the camera noise becomes (as a general parameter incorporating different noise sources in a camera). The noise level is heavily camera dependent, depending on both hardware (e.g. optics and sensor) and software (exposure strategies, noise filtering technologies and image settings). The noise level cost will increase continuously with the noise level until there is no information left in the image and the video becomes black. However, from a user point of view it is desirable to have a simplified configuration, thus a set of a low, medium and high light level were mapped to the measurements. Some example noise levels are presented in Table 8. For the step wise noise levels, the light levels correspond to the same as the scene illumination, L.

$$N_C \propto \text{Noise level.} \tag{18}$$

$$I_{noise} = N_{C,L}, \text{where L = the light level.} \tag{19}$$

We combine Equations 9-19 together to obtain the I_c constant.

$$I_C = D_S \times L \times D_C \times (1 + N_F) \times DR \times SAO + N_{C,L}, \tag{20}$$

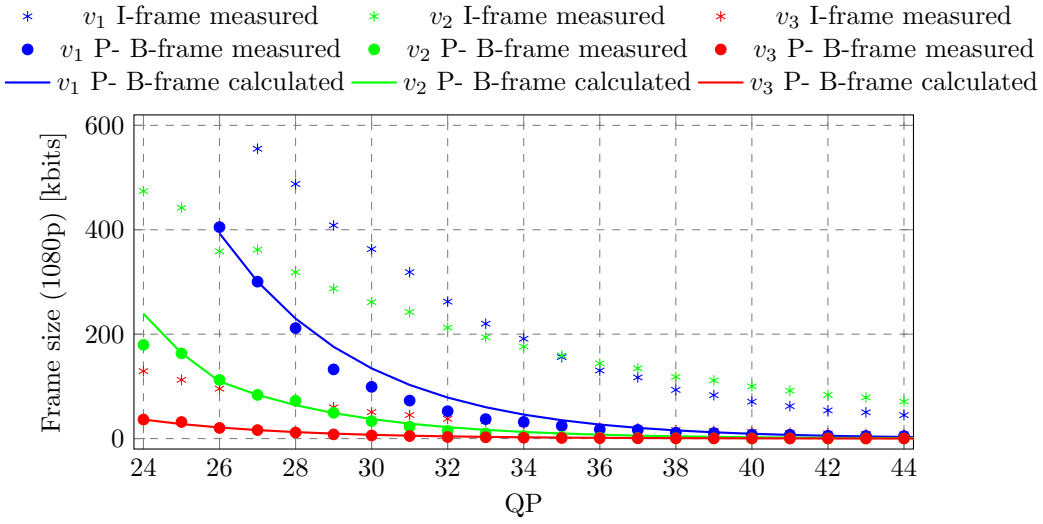
which then completes the final I-frame size prediction:

$$I = \left(D_S \times L \times D_C \times (1 + N_F) \times DR \times SAO + N_{C,L}\right) \times w \times h \times 2^{-\Delta QP/6}. \tag{21}$$

Table 8 Measured camera noise levels, $N_{C,L}$ using a standardized test scene.

Camera	Good light	Medium light	Low light
A	2.50	2.75	22.2
B	0.25	2.75	230
C	0.35	1.10	102
D	0.75	4.05	5.60
E	1.25	12.00	35.0
F	2.25	2.70	119

Figure 6 Measured P- and B-frame sizes and calculated ones for different videos, varying QP.



6.2 Inter frame model

Note: In this document we do not highlight the contribution of B-frames. We consider B-frame size prediction to be close to P-frame size prediction and as such a similar model can be used.

General assumptions Using the same logic as for I-frames, one can re-use Equation 8 while changing the compression part to predict the size of P-frames.

$$P = P_c \times w \times h \times compression, \tag{22}$$

with P being the P-frame size (in millibits), P_c a constant depending on level of details and noise (as I_c is), and w, h being the pixel width and height of the image.

P-frames are highly correlated with neighboring frames due to the way they are compressed. The relation between the compression parameter (QP) and frame size that we used for I-frames does not apply for P-frames due to this correlation. We introduce such correlation by changing the compression term to $5^{-\Delta QP/6}$, with the base 5 experimentally achieved through curve fitting. The difference between measured and calculated P-frame sizes (as well as a comparison with I-frame sizes for the same sequences) can be viewed in Figure 6

$$P = P_c \times w \times h \times 5^{-\Delta QP/6}. \tag{23}$$

Scene and camera related additions Now that a general encoder related model has been determined, and one can expand it with camera and scene parameters. We will refine the proposed P_c constant into a motion and noise term.

$$P_c = P_{motion} + P_{noise}. \tag{24}$$

Table 9 Measured scene motion level for a collection of scenes as % of the image.

Scene	Low Motion	Medium Motion	High Motion
Highway	2.50%	6.50%	15.00%
Office	0.50%	1.50%	5.00%
Parking Lot	0.50%	3.00%	10.00%
Retail	0.75%	3.00%	10.00%

The motion cost of the P-frame size is determined by the scene motion level and what we describe as a camera motion cost which reflects how well the H.264 encoder handles moving object encoding.

$$P_{motion} \propto \text{motion level } (ML). \tag{25}$$

$$P_{motion} \propto \text{camera motion cost } (MC). \tag{26}$$

The motion level constant ML is directly linked to the frame rate of the video, the lower the frame rate of the video, the more difference there will be between consecutive frames and the larger the motion "step" will be. This larger gap will translate into higher chances of a motion miss by the encoder and leads to higher bandwidth. There will also be time for more motion to happen in the scene. The model has been empirically tested and the motion level is modelled as being proportional to the inverse square of the video frame rate.

$$ML \propto \frac{1}{\sqrt{FPS}}. \tag{27}$$

Equation 27 is motivated using empirical evidence. Note that since the model will increase the motion level for a low fps stream the effect of this will be small for a low motion video, and larger for a medium or high motion video.

The ML_S is a measurable quantity for each scene at a certain reference frame rate, FPS_{ref}. To simplify the user interface a generic set of possible motion levels (e.g. High, Medium, Low) are measured for each scene. The motion level, if accurately known, can be uniquely used and varied per frame in a rate control prediction step use case. However, since the primary use case of this frame size prediction is to estimate the required storage during the system design phase there is a strong added benefit to simplify for the user and only pose the question, "Compared to a typical retail scene, do you anticipate that this scene will contain more or less motion?". Examples are provided in Table 9

$$ML \propto ML_S. \tag{28}$$

As indicated in Equation 26, the motion cost of P-frames is related to the efficiency of the encoder to detect motion and encode it.

$$MC \propto \text{motion encoder efficiency}(M_{EC}). \tag{29}$$

Note that M_{EC} is a measurable quantity per camera, but can likely be simplified because the camera encoding capabilities are often dependent on the encoder capabilities and efficiency.

Also, when the encoder cannot detect a motion (because of a too big motion "step" or complex motion for example) it will encode the part of the image as it would for an I-frame, this means that we can re-use part of the I-frame cost previously defined in Equation 20

$$MC \propto I_C. \tag{30}$$

Since Equation 20 is an estimation of the detail cost it is also a good estimate of the base cost for encoding objects in motion.

$$P_{motion} = ML_S \times clamp(\sqrt{\frac{FPS_{ref}}{FPS}}) \times I_C \times M_{EC}, \tag{31}$$

with clamp being the simple equation of limiting the value between a minimum and maximum value, which is is similar to $clamp(x) = max(min(x))$.

Clamp is here to make sure that we obtain a reasonable value, FPS and FPS_{ref} cannot be negative and should not be unrealistically high. Keep in mind that a FPS approaching infinity will cause the motion level to approach zero (there will not be any visible changes between consecutive frames) and vice-verse. Reasonable clamping limits are between 0.5 and 2, restricting the fps effect on the motion level to within double or half of the uncompensated motion level. The effect is likely present outside of this range but we have not verified this.

The same reasoning as Equation 30 can be applied to the noise part of the P-frame, we then define part of the noise as being equivalent to an I-frame encoding from Equation 19.

$$P_{noise} \equiv I_{noise} = N_{C,L} \tag{32}$$

Using the same simple relation between the previously described parameters, we combine Equations 24-32 together to obtain the P_c constant.

$$P_c = ML_S \times clamp(\sqrt{\frac{FPS_{ref}}{FPS}}) \times I_C \times M_{EC} + N_{C,L}, \tag{33}$$

which then completes the final P-frame size prediction:

$$P = \left(ML_S \times clamp(\sqrt{\frac{FPS_{ref}}{FPS}}) \times I_C \times M_{EC} + N_{C,L}\right) \times w \times h \times 5^{-\Delta QP/6}. \tag{34}$$

6.3 Video bandwidth model

One important contributor in a H.264 video bandwidth is the existence and size of the Group Of Pictures (GOP). The longer the GOP, the more P-frames there will be. Knowing that I-frames are a major contributor to the video bandwidth, the frequency of I-frames will then directly impact our prediction. We define a simple constant F to account for this.

$$F = \frac{I + (GOP - 1) \times P}{GOP}, \tag{35}$$

where I and P are the I-frame and P-frame sizes as described in Equation 8 using Equation 20 as well as Equation 22 using Equation 33, GOP is the Group of Pictures size of the video.

Calculating the video bandwidth is then simply multiplying our predicted average frame size F in Equation 35 with the video frame rate FPS.

$$bandwidth = \text{frame sizes} \times \text{video frame rate}. \tag{36}$$

$$B = F \times FPS. \tag{37}$$

This gives us the desired prediction of the video bandwidth B.

6.4 Simplified bandwidth model

For a simplified use case one may simplify Equation 21 and Equation 34 in such a way that all cameras are assumed to be the reference camera (camera A in the results) and no fps motion level scaling is performed. The nature factor, HDR, SAO and scene detail level are merged to an average value across all measured scenes, yielding the constant 1250 mbit/pixels used in Equation 38.

$$I = \left(1250 \times L + N_{camA,L}\right) \times w \times h \times 2^{-\Delta QP/6}. \tag{38}$$

$$P = \left(ML_S \times I_C \times 0.45 + N_{camA,L}\right) \times w \times h \times 5^{-\Delta QP/6}. \tag{39}$$

7. Experimental results

This paper's model was tested using different IP cameras from Axis Communications®.

The experimental part is decomposed into three major parts:

- First, we present different models present in the academic literature,
- Then, we ran a frame by frame test to see how good the model performs in a "real-time" scenario
- Last, we compared long term videos in real life scenarios to the predictions and benchmarked against some available industry prediction tools available.

7.1 State of the art bandwidth estimation model

To the best of our knowledge, there are two known alternative methods to estimate the frame size, and in turn the expected video bandwidth needed for the video transmission. These methods are based on other encoding methods (respectively MJPEG and MPEG-4) and aim to provide an estimate of the expected frame sizes. To the best of our knowledge, we propose the first open source frame size estimation for MPEG-4 part 10 AVC (H.264).

We denote the MJPEG method with `LIN`. This method only considers the compression parameter (QP for H.264 videos), and scales the frame size linearly according to such a parameter that we name q_l. Given a maximum size, identified with the term s_{max}, the frame size $s(q_l)$ is computed as $s(q_l) = q_l \cdot s_{max}$. The parameter q_l indicates the quality of the encoding, and relates, as indicated previously, to the Quantization Parameter QP. The scale and logic used are different and in MJPEG $q_l \in [0.01, 1.0]$, 1 being the lowest compression and 0.1 the highest, therefore $q_l = 1.01 - (\mathrm{QP}/51)$. In the case of a 1080p YCbCr color video with 8 bits per pixel, $s_{max} = 1920 \cdot 1080 \cdot 8 \cdot 3 = 49766400$ [bits per frame]. This model is used for example in [Seetanadi et al., 2017] to devise a control strategy to determine the quality to be applied given a target bandwidth consumption.

This model unfortunately produced really poor results, on average overestimating the bitrates with a factor of 200! Hence we implemented a slightly better interpretation of q using the theoretical H.264 QP exponential quality scaling [Wiegand et al., 2003] such that

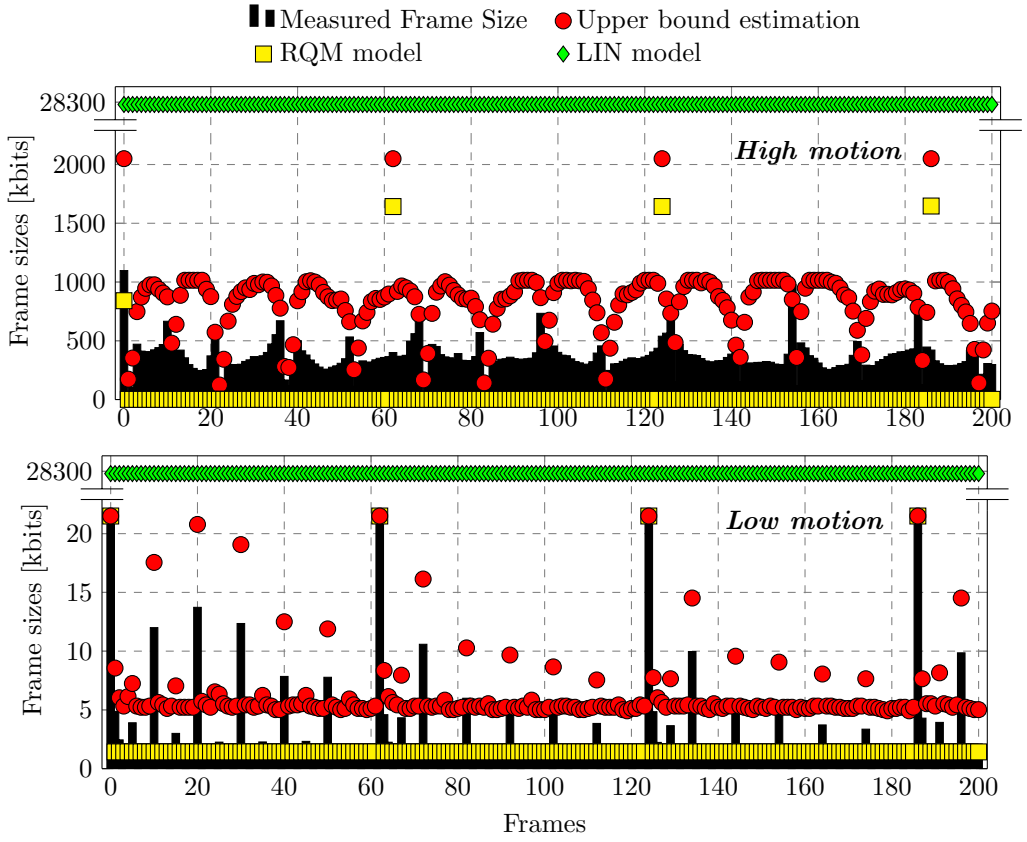
$$q = 2^{-QP/6} \times 99 + 1. \tag{40}$$

The model is designed such as $QP_{max} = 51$, $q(0) = 100$ and $q(QP_{max}) = 1$. Finally we scale the frame size to the video bandwidth using Equation 41.

$$B = s \times w \times h \times FPS. \tag{41}$$

There is also another model based on MPEG-4 (which also predicates H.264), we name it `RQM`. This model is used in [Almeida et al., 2007] and described in [Ding and Liu, 1996]. It uses curve fitting to determine the parameters of a rate-distortion curve, modeled with a Gaussian random variable. Denoting with α a constant accounting for overhead bits, with β a constant that varies with the resolution and amount of motion in the video, with q_r the compression level for MPEG-4 ($q_r \in [1, 31]$), and with γ a constant that varies depending on the frame type ([Ding and Liu, 1996] providing recommended bounds of $\gamma \in [0.5, 1]$ for I-frames and $\gamma \in [0.5, 1.5]$ for P-frames), the size of the frame can be written as $s(q_r) = \alpha + \beta \cdot 1/{q_r}^{\gamma}$. Due to its a-posteriori nature and the time it takes to calibrate for this model we did not do a long-term comparison for this report.

Figure 7 Results of the comparison experiment with the high- and low-motion video.



7.2 Frame-by-Frame Evaluation

We present here a first validation experiment done with our reference Camera A. We recorded two videos of the same scene in an office. The scene has a lot of details. Our aim is to show a frame-by-frame comparison between our frame size estimation and the state-of-the-art techniques.

The two videos differ in the amount of motion that is introduced[1]. A toy, present in the scene, allows us to introduce very limited but non-zero motion in both cases. In the first video, we also sharply changed the position of the camera. This simulates a fast movement for a video-surveillance camera. In the second video we kept the camera still, thus the only movement comes from the toy. The first video is characterized by a large amount of motion μ_s, while the second video has a very low μ_s.

The Camera A parameters for the two videos are: camera level detail $d_c = 1$, enhancement factor $e = 1.35$ (HDR), width $w = 1920$ [pixels], height $h = 1080$ [pixels], frame rate $f_s = 25$ [frames per second], QP $= 29$, noise level $n_{c,\ell} = 2.5$, motion encoder efficiency $\mu_x = 0.45$, GOP $= 64$. The scene parameters are: no nature, $n = 0$, very good illumination, $\ell = 1$, scene detail $d_s = 780$ [millibit per pixel], and size of the average object SAO $= 1$.

Figure 7 shows the results we obtained for the two videos. Each plot represents 200 frames of one video, the top one being the high-motion one and the bottom one being the low-motion case. The black bars represent the real frame sizes measured after the encoding. The circles represent the estimated upper bound on the frame sizes provided by the algorithm presented in this paper. The squares show the estimate produced by the `LIN` model, which does not take into account the difference between I, P, and B frames. Finally, the squares represent the estimate produced by the `RQM` model.

[1] The two videos are available online: `https://www.youtube.com/watch?v=614BbbhD56M` (high-motion), and `https://www.youtube.com/watch?v=q4j3LlVrOls` (low-motion). We have manipulated them to also visually show the motion vectors detected for both the original videos: `https://www.youtube.com/watch?v=5YrxlGhadsY` (high-motion), and `https://www.youtube.com/watch?v=cfrO8CZQa-E` (low-motion)

For the RQM model, we used the low-motion video to tune the parameters α, β, and γ, as recommended in [Almeida et al., 2007]. The tuning resulted in $\alpha = 0.55$ and $\beta = 1.7$. As γ changes depending on the frame type, we fit $\gamma_I = 0.5$ and $\gamma_P = 4$ separately. The RQM tuning resulted in average errors on I-frames and P-frames respectively of 1.80% and 1.38%, which indicate very good performance for the low motion video. The square points in the lower plot of Figure 7 are therefore *a posteriori* estimations, and are clearly a very good fit for the video, despite the presence of a few outliers. The RQM model neglects motion — i.e., the β parameter is not sufficient to take motion into account. In fact, when the parameters determined with the low-motion video are used for *a priori* estimating the size of the frames in the high-motion video, the estimate frame size greatly underestimates the real value. The RQM approximation is therefore not a good fit to upper bound the size of the frames.

On the contrary, the LIN model gives very conservative results for both the high- and low-motion video, as its only parameter is a translation of the encoding quality QP. These are too conservative to be used in any practical setting, since the estimates are roughly 30 times as large as the real values. The LIN approximation is therefore also not a good upper bound for the size of the frames.

In the case of our upper bound I^* and P^*, the circles represent for both plots *a priori* estimates based on the parameters that we have selected and on a standard computation of the motion level μ_s based on the percentage of pixels that differ from one image to the next (which could be determined before the encoding step). Roughly, the computed upper bounds are twice as large as the real values. While this could be reduced with a more conservative setup of parameters, we believe that there could be a risk of cases in which the real frame size exceeds the upper bound. In the full length of the two videos (low-motion 751 frames, high-motion 376 frames) this never happens for the low-motion case, and happens five times for the high-motion case. Inspecting these five occurrences prompted us to suspect some capturing error or some encoding miss, possibly due to the sharp movement.

7.3 State of the art bandwidth comparison

This section compares the bandwidth prediction from the proposed model (MODEL), the proposed simplified model (SIMP) against the state of the art linear model (SOTALIN) and exponential model (SOTAEXP). All bitrates are in unit [kbit/s].

The videos were sorted using measurable parameters:

- Video resolution
- Video frame rate
- Video GOP size
- Camera filter parameters (sharpness, noise reduction, etc.)
- Camera model specifics (optics type, sensor type, filters used, etc.),

and then annotated manually to add extra parameters:

- Scene type (indoor, outdoor)
- Scene nature level (low, medium, high)
- Scene light level (low, medium, high)
- Scene motion (low, medium, high)

The results were then gathered, the average bandwidth (B) measured for part or full footage and compared with the model output. We will here go through some relevant examples to illustrate the correlation with the paper's model and real scene measurements obtained. Please note that due to company secrecy reasons some parameters have been omitted intentionally (i.e. camera model, weights used, etc.)

The motion levels have been estimated visually as no motion ($\approx 0\%$), low motion ($\approx 2\%$), medium motion ($\approx 5\%$), high motion ($\approx 10\%$) and very high motion ($\geq 20\%$).

Note that the formula parameters were obtained with different videos than the ones used in the result section.

Table 10 Parking lot scenarios parameters.

Scenario	FPS	QP	GOP	Motion Level
1	25	28	62	No motion ($\approx 0\%$)
2	25	28	62	Low motion ($\approx 2\%$)
3	25	28	62	High motion ($\approx 10\%$)
4	12	32	32	Low motion ($\approx 0\%$)
5	12	32	32	Medium motion ($\approx 2\%$)
6	12	32	32	High motion ($\approx 10\%$)

Table 11 Parking lot scenario measured bitrates and predicted bitrates (in kbits/s).

Scenario	Real	MODEL	SIMP	SOTALIN	SOTAEXP
1	1040	1010	1175	189307	20312
2	1600	1541	1750	189307	20312
3	3200	3664	4049	189307	20312
4	544	538	634	75411	6879
5	720	661	727	7511	6879
6	1200	1157	1099	75411	6879

Parking lot scenario Videos recorded for a parking lot scene using camera A, 1920x1080 resolution, HDR, medium SAO and without nature during day time with good illumination. Six scenarios were recorded at slightly different times, described in Table 10, yielding the bitrates and predictions in Table 11.

Traffic scenario Videos recorded for a traffic scenario for a highway scene using camera A, 1920x1080 resolution, 25 fps, QP 28, GOP 32, HDR, medium SAO and without nature during day time with good illumination. The scenario motion levels are as described in Table 12, yielding the bitrates and predictions in Table 13.

Perimeter defense scenario Videos recorded for a perimeter defense scenario using camera C, 30 fps, QP 18, GOP 32, without HDR, medium SAO and with nature. The scenario scenes, illumination, resolutions and motion levels are as described in Table 14, yielding the bitrates and predictions in Table 15.

4k scenario Videos recorded for a city street scenario in a street corner scene using camera D, 3840*2160, 25 fps, QP 24, GOP 4, without HDR, medium SAO, without nature, with low illumination and a low motion level (2%). The scenario, **11**, yields the bitrates and predictions in Table 16.

Industry scenario Videos recorded for an industry scenario in a perimeter scene using camera E, 3072*1728, 25 fps, QP 32, GOP 32, with HDR, medium SAO, without nature, with low illumination and a low motion level (2%). The scenario, **12**, yields the bitrates and predictions in Table 17.

City scenario Videos recorded for a city scenario in an intersection scene using camera A, 1280*720, 15 fps, QP 36, GOP 30, with HDR, medium SAO, with nature, with good illumination and a high motion level (10%). The scenario, **13**, yields the bitrates and predictions in Table 18.

Estimations and errors A summary of the SOTA comparison is presented in Tables 19-21. Details about each scenario parameters are summarized in Table 25. Note that due to its high bitrate, Scenario 11 will disproportionately affect the MAE and MSE scores. Hence a score is also presented without them.

Table 12 Traffic scenarios parameters.

Scenario	Motion Level
7	Very high motion ($\approx 20\%$)
8	Low motion ($\approx 2\%$)

Table 13 Traffic scenario measured bitrates and predicted bitrates (in kbits/s).

Scenario	Real	MODEL	SIMP	SOTALIN	SOTAEXP
7	10000	9643	6924	189307	20312
8	2800	2300	1750	189307	20312

Table 14 Perimeter defense scenarios.

Scenario	Scene	Illumination	Resolution	Motion Level
9	Perimeter	High	640*480	Medium motion ($\approx 5\%$)
10	Parking lot	Low	384*288	Low motion ($\approx 2\%$)

Table 15 Perimeter defense scenario measured bitrates and predicted bitrates.

Scenario	Real	MODEL	SIMP	SOTALIN	SOTAEXP
9	4215	3955	5150	47967	9861
10	4966	5321	1530	17268	3550

Table 16 4k scenario measured bitrates and predicted bitrates (in kbits/s).

Scenario	Real	MODEL	SIMP	SOTALIN	SOTAEXP
11	42500	46529	109332	886037	119232

Table 17 Industry scenario measured bitrates and predicted bitrates (in kbits/s).

Scenario	Real	MODEL	SIMP	SOTALIN	SOTAEXP
12	2837	2726	2923	402191	36687

7.4 Commercial bandwidth comparison

External partner field test at a hotel complex, together with five commercial bitrate estimations (Ext 1-5) as well as the model predictions (MODEL and SIMP) for eleven scenarios. The videos are filmed over 5 days using 15 fps, QP 28 and GOP 62 for all of them. The resulting errors are summarized in Tables 22-23 and summarized in Table 24.

The model average relative error is 29%**, compared to the simplified model average relative error of** 105%**, the best SOTA model relative error** 2100% **and the best external model relative error** 336%**.**

Note that a good part of this error is a general overestimation caused by the videos using Axis Zipstream technology at a low setting which will produce a lower bitrate than if not activated. As indicated previously this technology is omitted purposely in this report.

Table 18 Industry scenario measured bitrates and predicted bitrates (in kbits/s).

Scenario	Real	MODEL	SIMP	SOTALIN	SOTAEXP
13	620	505	321	33308	2817

Table 19 SOTA comparison absolute errors.

Scenario	MODEL	SIMP	SOTALIN	SOTAEXP
1	30	135	188267	19272
2	59	150	187707	18712
3	464	849	186107	17112
4	6	90	74867	6335
5	59	7	74691	6159
6	43	101	74211	5679
7	357	3076	179307	10312
8	500	1050	186507	17512
9	260	935	43752	5646
10	355	3436	12302	1416
11	4029	66832	843537	76732
12	111	86	399354	33850
13	115	299	32688	2197

Table 20 SOTA comparison relative errors.

Scenario	MODEL	SIMP	SOTALIN	SOTAEXP
1	2.88%	12.96%	18102.64%	1853.12%
2	3.70%	9.36%	11731.72%	1169.53%
3	14.49%	26.55%	5815.86%	534.77%
4	1.17%	16.57%	13762.27%	1164.47%
5	8.13%	0.98%	10373.71%	855.38%
6	3.60%	8.43%	6184.23%	473.23%
7	3.57%	30.76%	1793.07%	103.12%
8	17.86%	37.51%	6660.98%	625.45%
9	6.17%	22.18%	1038.00%	133.95%
10	7.14%	69.19%	247.72%	28.51%
11	9.48%	157.25%	1984.79%	180.55%
12	3.93%	3.03%	14076.62%	1193.15%
13	18.57%	48.22%	5272.21%	354.30%

Table 21 SOTA comparison summary.

Scenario	MODEL	SIMP	SOTALIN	SOTAEXP
MAE	491	5927	191023	16995
MAE - 11	197	851	136647	12017
RMSE	1145	18586	285969	25735
RMSE - 11	262	1418	171163	15062
MRE	7.74%	34.08%	7464.91%	666.89%

Table 22 External comparison absolute errors.

Scenario	MODEL	SIMP	EXT 1	EXT 2	EXT 3	EXT 4	EXT 5
14	13	63	490	366	1198	397	1519
15	130	627	1305	1181	2013	1212	2335
16	116	621	1140	1017	1848	1048	2170
17	73	547	715	591	1423	622	1744
18	386	537	870	746	1578	777	1900
19	171	30	281	226	596	240	739
20	47	287	205	150	520	164	663
21	94	338	569	514	884	528	1027
22	69	111	368	507	599	606	601
23	24	69	86	66	202	71	254
24	34	30	205	256	290	292	291

Table 23 External comparison absolute errors.

Scenario	MODEL	SIMP	EXT 1	EXT 2	EXT 3	EXT 4	EXT 5
14	1.48%	7.06%	54.78%	40.92%	134.02%	44.39%	170.03%
15	167.48%	806.24%	1678.17%	1518.85%	2588.49%	1558.73%	3002.18%
16	47.67%	255.90%	470.10%	419.02%	761.96%	431.81%	894.59%
17	10.91%	81.79%	106.88%	88.34%	212.79%	92.98%	260.92%
18	75.24%	104.66%	169.61%	145.46%	307.64%	151.50%	370.37%
19	51.40%	9.14%	84.40%	67.88%	178.81%	72.02%	221.71%
20	11.37%	70.20%	50.19%	36.74%	127.08%	40.10%	162.02%
21	207.49%	745.77%	1254.80%	1133.41%	1948.37%	1163.79%	2263.57%
22	9.56%	15.40%	51.00%	70.22%	82.89%	83.87%	83.17%
23	17.33%	49.91%	62.27%	47.73%	145.35%	51.37%	183.10%
24	17.43%	15.22%	105.64%	131.82%	149.07%	150.42%	149.46%

Table 24 External comparison summary.

Scenario	MODEL	SIMP	EXT 1	EXT 2	EXT 3	EXT 4	EXT 5
MAE	105	296	567	511	1014	542	1204
RMSE	145	380	684	613	1180	642	1404
MRE	56.12%	196.48%	371.62%	336.40%	603.32%	349.18%	705.56%

Table 25 External test scenarios.

Scenario	Scene	Camera	Illumination	Motion Level	Resolution	HDR	SAO	Nature
14 day	Reception	B	High	Low motion ($\approx 2\%$)	1920x1080	On	Medium	No
14 night	Reception	B	Medium	Low motion ($\approx 2\%$)	1920x1080	On	Medium	No
15	Emergency Exit	C	High	No motion ($\approx 0\%$)	1920x1080	Off	Large	No
16	Office	C	Medium	Low motion ($\approx 2\%$)	1920x1080	Off	Large	No
17 day	Street Corner	A	High	Medium motion ($\approx 5\%$)	1920x1080	On	Large	No
17 night	Street Corner	A	Medium	Low motion ($\approx 2\%$)	1920x1080	On	Large	No
18	Reception	C	High	Low motion ($\approx 2\%$)	1920x1080	Off	Medium	No
19 day	Mall	C	High	No motion ($\approx 0\%$)	1280x720	Off	Medium	No
19 night	Mall	C	Medium	No motion ($\approx 0\%$)	1280x720	Off	Medium	No
20	Elevator/On-board	C	High	Medium motion ($\approx 5\%$)	1280x720	On	Large	No
21	Emergency Exit	F	Medium	Low motion ($\approx 2\%$)	1280x720	On	Large	No
22 day	Parking lot	A	High	Medium motion ($\approx 5\%$)	1280x720	On	Small	Yes
22 night	Parking lot	A	Low	Low motion ($\approx 2\%$)	704x480	Off	Small	Yes
23	Parking lot	F	High	Medium motion ($\approx 5\%$)	704x480	On	Medium	No
24 day	Parking lot	A	High	Medium motion ($\approx 5\%$)	704x480	On	Medium	No
24 night	Parking lot	A	Medium	Low motion ($\approx 2\%$)	704x480	On	Medium	No

8. Conclusion

In this report we presented a practical contribution on how to derive upper bounds for the size of video frames in a streaming system. We have discussed which characteristics influence the bandwidth requirements of different cameras, derived models for the upper bound of the size of I-, P-, and B-frames. We have also systematized the knowledge on the involved quantities and parameters. We divided such quantities into parameters that are known, characteristics that are measurable, and values that are computable. We have then taken the measurable characteristics and discussed how to conduct field tests to obtain reasonable values for them, and — when possible — how to guess based on the environmental conditions.

The derivation of reasonable upper bounds for the video bandwidth requirements allows us to precisely formulate the problem of allocating network bandwidth to a set of cameras in a switched Ethernet network environment and to reuse well-known scheduling results. We have shown with a thorough experimental campaign that our upper bounds are more reliable, and closer to the real frame sizes than state-of-the-art estimation techniques.

A proper estimation of the frame sizes is the key to properly dimension network infrastructures for real-time video-surveillance systems. Our results demonstrated that we can dimension the network infrastructure, being able to accurately predict the bitrate consumption of video streams. Our findings have a significant industrial relevance, as they permit to reduce the infrastructure cost and allows us to reuse known scheduling results.

Acknowledgments

This work was partially supported by the Wallenberg Artificial Intelligence, Autonomous Systems and Software Program (WASP) funded by Knut and Alice Wallenberg Foundation. Alexandre Martins, Karl-Erik Årzén, and Martina Maggio are members of the LCCC Linnaeus Center and the ELLIIT Excellence Center at Lund University.

References

Almeida, L., P. Pedreiras, J. Ferreira, M. Calha, J. A. Fonseca, R. Marau, V. Silva, and E. Martins (2007). "Online QoS adaptation with the flexible time-triggered (FTT) communication paradigm". In: *Handbook of Real-Time and Embedded Systems*.

Andersson, B. (2008). "Schedulability analysis of generalized multiframe traffic on multihop-networks comprising software-implemented ethernet-switches". In: *IEEE International Symposium on Parallel and Distributed Processing*, pp. 1–8. DOI: 10.1109/IPDPS.2008.4536565.

Andersson, B., S. Chaki, D. de Niz, B. Dougherty, R. Kegley, and J. White (2012). "Non-preemptive scheduling with history-dependent execution time". In: *24th Euromicro Conference on Real-Time Systems*, pp. 363–372. DOI: 10.1109/ECRTS.2012.38.

Baruah, S. K. and S. Chakraborty (2006). "Schedulability analysis of non-preemptive recurring real-time tasks". In: *Proceedings 20th IEEE International Parallel Distributed Processing Symposium*. DOI: 10.1109/IPDPS.2006.1639406.

Baruah, S., D. Chen, S. Gorinsky, and A. Mok (1999a). "Generalized multiframe tasks". *Real-Time Systems* **17**:1, pp. 5–22. ISSN: 0922-6443. DOI: 10.1023/A:1008030427220. URL: https://doi.org/10.1023/A:1008030427220.

Baruah, S., D. Chen, and A. Mok (1999b). "Static-priority scheduling of multiframe tasks". In: *Real-Time Systems, 1999. Proceedings of the 11th Euromicro Conference on*, pp. 38–45. DOI: 10.1109/EMRTS.1999.777448.

Chakraborty, S. and L. Thiele (2005). "A new task model for streaming applications and its schedulability analysis". In: *Proceedings of the Conference on Design, Automation and Test in Europe - Volume 1*. DATE '05. IEEE Computer Society, Washington, DC, USA, pp. 486–491. ISBN: 0-7695-2288-2. DOI: 10.1109/DATE.2005.26.

Ding, W. and B. Liu (1996). "Rate control of mpeg video coding and recording by rate-quantization modeling". *IEEE transactions on circuits and systems for video technology* **6**:1, pp. 12–20.

Ekberg, P., N. Guan, M. Stigge, and W. Yi (2015). "An optimal resource sharing protocol for generalized multiframe tasks". *Journal of Logical and Algebraic Methods in Programming* **84**:1, pp. 92–105. ISSN: 2352-2208. DOI: https://doi.org/10.1016/j.jlamp.2014.10.001.

Han, C.-C. J. (1998). "A better polynomial-time schedulability test for real-time multiframe tasks". In: *Proceedings 19th IEEE Real-Time Systems Symposium*, pp. 104–113. DOI: 10.1109/REAL.1998.739735.

ISO/IEC MPEG & ITU-T VCEG, J. V. T. (of (2003). *"Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264/ISO/IEC 14496-10 AVC)"*.

ITU-T (2017). *Advanced video coding for generic audiovisual services*. https://www.itu.int/rec/T-REC-H.264-201704-I/en. (Visited on 2017-08-23).

Li, S., S. Rubini, F. Singhoff, and M. Bourdelles (2014). "A task model for tdma communications". In: *Proceedings of the 9th IEEE International Symposium on Industrial Embedded Systems (SIES 2014)*, pp. 1–4. DOI: 10.1109/SIES.2014.7087455.

Lu, W.-C., K.-J. Lin, H.-W. Wei, and W.-K. Shih (2007). "New schedulability conditions for real-time multiframe tasks". In: *19th Euromicro Conference on Real-Time Systems (ECRTS'07)*, pp. 39–50. DOI: 10.1109/ECRTS.2007.20.

Mok, A. K. and D. Chen (1997). "A multiframe model for real-time tasks". *IEEE Transactions on Software Engineering* **23**:10, pp. 635–645. ISSN: 0098-5589. DOI: 10.1109/32.637146.

Peng, B. and N. Fisher (2016). "Parameter adaption for generalized multiframe tasks and applications to self-suspending tasks". In: *2016 IEEE 22nd International Conference on Embedded and Real-Time Computing Systems and Applications*, pp. 49–58. DOI: 10.1109/RTCSA.2016.15.

Seetanadi, G. N., L. Oliveira, L. Almeida, K.-E. Arzen, and M. Maggio (2017). "Game-theoretic network bandwidth distribution for self-adaptive cameras". In: *15th International Workshop on Real-Time Networks*.

Stigge, M., P. Ekberg, N. Guan, and W. Yi (2011). "The digraph real-time task model". In: *Proceedings of the 2011 17th IEEE Real-Time and Embedded Technology and Applications Symposium*. RTAS '11. IEEE Computer Society, Washington, DC, USA, pp. 71–80. ISBN: 978-0-7695-4344-4. DOI: 10.1109/RTAS.2011.15. URL: http://dx.doi.org/10.1109/RTAS.2011.15.

Wiegand, T., G. J. Sullivan, G. Bjontegaard, and A. Luthra (2003). "Overview of the h.264/avc video coding standard". *IEEE Transactions on Circuits and Systems for Video Technology* **13**:7, pp. 560–576. ISSN: 1051-8215. DOI: 10.1109/TCSVT.2003.815165.

Zeng, H. and M. Di Natale (2013). "Outstanding paper award: using max-plus algebra to improve the analysis of non-cyclic task models". In: *2013 25th Euromicro Conference on Real-Time Systems*, pp. 205–214. DOI: 10.1109/ECRTS.2013.30.

Zuhily, A. and A. Burns (2009). "Exact scheduling analysis of non-accumulatively monotonic multiframe tasks". *Real-Time Systems* **43**:2, pp. 119–146. ISSN: 0922-6443. DOI: 10.1007/s11241-009-9085-6. URL: http://dx.doi.org/10.1007/s11241-009-9085-6.

A. Metrics detail

In this part we will gather the metrics used in this paper.

Note: If not stated otherwise, all the metrics listed are in the $\mathbb{N}$ space.

Bandwidth (B): Video bandwidth (amount of bits per second generated by the video). This value is provided in kilobits per second (kb/s).

Camera detail properties (D_C): Constant reflecting the camera capacity to retain scene details. This constant is camera specific and a simplified reflection of many camera characteristics such as sensor size, sensor type, lenses properties, etc. It reflects the difference between a reference camera and the measured camera. This parameter needs to be measured for a specific camera model and lens combination. This value has no unit (it is a relative value) and is typically in [0.1;10.0].

Camera motion cost (MC): This is a temporary variable modeling the camera encoder capacity to encode motion.

Camera noise (N_C): Constant indicating the amount of noise in the camera. This constant is camera specific and a simplified reflection of many camera characteristics such as sensor size, sensor type, camera image tuning, camera electronics design, etc. It is indicated in millibits per pixel. This parameter needs to be measured for a specific camera model. $N_C \in \mathbb{R}^+$, a typical experimental value is within [1.0 ; 500.0], a lower value indicating an indoor high light environment and the higher one indicating a low-light environment.

Compression: Temporary variable indicating how much information the video was configured to lose. A high compression indicates a big data loss, with worse image/video quality. A low compression indicates that the loss of information should be moderate, thus leading to a better image/video quality but also an increased bit rate. This parameter is modeled in the H.264 standard by the quantization parameter (see QP).

Dynamic range factor (DR): Indicates if the video is using high dynamic range (HDR) or similar technology. High dynamic range images usually have more details and are sharper than non HDR images. $DR \in \mathbb{R}^+$ and an experimental value is typically within [1.0 ; 1.35], a lower value indicating a video without HDR and higher one a video with HDR.

Frame rate (FPS): Number of frames per second of the video.

Frame size (F): Average frame size. This value is an average of I-frame, P-frame (and B-frame) sizes taking in account the GOP length (see GOP). This value is provided in kilobits per second (kb/s).

Group of pictures (GOP): Number of frames between two consecutive I-frames. A Group of Picture is a sequence of frames that starts with an I-frame followed by P and B-frames (see [Wiegand et al., 2003])

Height (h): Number of pixels in the shortest image plane (usually y axis).

I-frame size (I): Size of I-frames. This value is provided in kilobits per second (kb/s).

I-frame size constant (I_c): Temporary constant for I-frame size (only used for didactic explanation).

I-frame details size ($I_{details}$): Temporary constant for I-frame size (only used for didactic explanation).

I-frame noise size (I_{noise}): Temporary constant for I-frame size (only used for didactic explanation).

Motion encoder efficiency (M_{EC}): Encoder related constant which reflects the ability of the encoder to efficiently encode moving objects. The more efficient the encoder is to encode motion, the lower the cost. An encoder with a large motion search window for example will have a low motion cost. $M_{EC} \in \mathbb{R}^+$ and an experimental value is typically within [0.1;1.0].

Motion level (ML): Part of the image that is expected to be moving. It is indicated as the portion of the image that is moving. $ML \in \mathbb{R}^+$ and in the range [0.0; 1.0].

Nature factor (N_F): Amount of nature (trees, bushes, grass, etc.) present in the scene. Nature scenery has specific type of motion and noise, these generate extra bandwidth that an office scene will not have for example. $N_F \in \mathbb{R}^+$ a typical experimental value is within [0.0 ; 1.0], a lower value indicating a low level of vegetation and higher one a high part of vegetation on the image.

P-frame size (P): Size of P-frames. This value is provided in kilobits per second (kb/s).

P-frame size constant (P_c): Temporary constant for P-frame size (only used for didactic explanation).

P-frame details size ($P_{details}$): Temporary constant for P-frame size (only used for didactic explanation).

P-frame noise size (P_{noise}): Temporary constant for P-frame size (only used for didactic explanation).

QP: Compression parameter defined in the H.264 standard [Wiegand et al., 2003]. The value is from 1 to 51, 1 indicating a low compression and 51 a high compression (see Compression)

ΔQP: Difference between the QP used and the reference QP used during measurement of some needed constants (see QP and Reference QP).

Reference frame rate (FPS_{ref}): Number of frames per second used as reference during measurement of some needed constants. We use a typical value of 30 frames per second.

Reference QP: Compression parameter used during measurement of some needed constants (see QP).

Resolution: We consider resolution as the number of points (pixels) in the image. We consider the image size and not the number of bits that compose each pixel. Resolution is simply $Width \times Height$ pixels.

Scene detail level (D_S): Constant indicating the amount of detail in the scene. $D_S \in \mathbb{R}^+$, value is in [0.0; 10.0].

Scene illumination (L): Constant indicating the luminance (amount of light) in the scene. $L \in \mathbb{R}^+$, a typical experimental value is within [0.25 ; 1.0], 0.25 indicating a low light condition and 1.0 a high light level.

Size of Average Object (SAO): This metric reflects the expected distance of an object in the image. It is determined by different factors such as zoom level, field of view, lens type (panoramic, fisheye, etc.) and placement of the camera (far from objects or really close). It is unitless. $SAO \in \mathbb{R}^{*+}$, a typical experimental value is within [0.5; 1.5].

Width (w): Number of pixels in the longest image plane (usually x axis).