JACOBS
UNIVERSITY

Screen Content Coding with VP9

Petru Lupascu

Bachelor Project for the Thesis in Electrical and Computer Engineering

Prof. Dr.-Ing Werner Henkel
Steffen Schulze(LMI)
Name and title of the supervisors

Date of Submission:tbd

Contents

List of Figures

1 Introduction

From the early days of digital television, video compression techniques have gained increased attention, mainly due to bandwidth always being an expensive asset, fact which is still relevant. Throughout the years, video coding techniques played a vital role in reducing the size of video sequences without significant alteration of its quality. In parallel with advancement in computer performance, video coding allowed for services such as video telephony and digital television to be more accessible, which in turn increased the demand. As a consequence, the development of video coding techniques was incentivised. Straight Forward Pulse Code Modulation (PCM) was one of the first attempts in coding video signals at around 140 Mbits/s. Since then the coding techniques has seen massive development in order to catch up with the demands. Modern codecs are able to code Video Signals as low as 9 Mbits/s for HDTV format. Newer generation codecs target the same performance as the previous generation at half the bit rate, however at the expense of increasing complexity. Most coding schemes require hardware implementations for optimized performance, making standardization essential in order to ensure compatibility with as large amount of devices as possible [1] .

Most of the standards developed by two groups: the Video Coding Experts Group (VCEG), known for H26x family of codecs and Moving Picture Experts Group (MPEG) known for MPEG-X family of codecs. After being approved, a codec is licensed to software developers and hardware manufacturers for a fee. Since the middle of the last decade, open source and royalty free codecs are being developed and have gained traction, effectively competing with the standard families mentioned above. Some of the most popular ones that were developed such as the VPx family.

Due to the good performance and open source nature of the VPx codec family, companies, such as LogMeIn started incorporating them in their Video Conferencing software, by implementing software-based codecs. Currently, LogMeIn uses a software-based codec to encode and decode screen content of online conferences. The codec performs compression of screen content video stream prior to sending the stream over the Internet and decompresses it at the receiver side. Moving forward, the current software-based video codec should be replaced by a modern real-time video codec, such as VP9. VP9 was designed around ordinary video use cases, webcam and movie video content. Libvpx, VP9s software implementation, focuses on these use cases with some enhancements regarding animations. Thus, in this project we are investigating the capabilities of VP9 to encode screen content, by evaluating the codec in terms of Bitrate, PSNR and CPU-requirement, with screen and regular content, uncover its weak areas and propose solutions to optimize them.

The next chapter is focused on providing background information on Image and Video codecs. We will as well define the Screen content and introduce the VP9

2 Background and literature review

2.1 Digital Image and Video Signals

An Image is defined as a projection of a 3-D scene, characterized by depth, texture, and illumination, onto a 2-D plane characterized by texture and illumination without depth information [2, pp. 5], or in case of colour Images additionally chrominance. It may also be defined as a 2 dimensional signal $f(x, y)$, where x and y are spatial coordinates and f is the intensity at that point. When x, y and f are finite we call this image a Digital Image [3, ppp. 1]. Following this definition, a Video represents a sequence of images over a period of time and can be defined as $f(x, y, t)$, where x, y, f are spatial and intensity values and t is the time. For the sake of simplicity we will call the 2-D point a pixel and its intensity, a pixel value and each image in a video sequence frame. Furthermore, an image can be characterized in terms of its resolution and colour format, for the video, additionally there is duration. The resolution commonly describes the amount of pixels present in the image, for example: 740x480. The colour format represents a typical arrangement of colours in an image such as gray scale, where the pixels value represents the light intensity (luminance) information, commonly 0 to 255 for an 8 bit image. Important colour formats are YUV and RGB, where the image is divided into three sub planes containing luminance, red chrominance and blue chrominance values for YUV and Red, Green and Blue colour intensity values for RGB. Usually, 8 bits values per sub plane pixel are used. Generally, all the parameters mentioned depend on the particular application. However, in most the cases, the amount of data required to store or transmit a video or an image tends to be very large. A two-hour Standard Definition (SD) 720x480x24 bits per frame movie, displayed at 30 frames per second must be accessed at $31,104,000$ bytes/sec and would require roughly 224 GB of storage. The rate and the storage get much larger High Definition (HD) videos where the resolution is 1920x1080x24 [3, pp. 525-526], which is a widespread expectation now days.

2.2 Image Codec

Storing video data in it's raw form is extremely inefficient, deeming a compression scheme necessary. Such a compression scheme is commonly referred to as a codec. Commonly, a video tends to have both high spatial redundancy across a frame and temporal redundancy across multiple frames. A group of neighbouring pixels tends to have the same or similar pixel intensity values and can be present in multiple frames across a video sequence. This allows the compression scheme to be optimized beyond typical source coding schemes such as Arithmetic Coding. Typically, a video codec will efficiently decorelate a video in attempt to remove spatial and temporal redundancies and then perform entropy coding.

Since a video is a sequence of frames, one might want to compress each frame individually decorrelating the image by applying a 2-D transform such as Karhunen-Loeve transform (KLT) to sub blocks of the image. KLT has the usefull property that its coefficients are decorrelated and the energy of the block is packed into the minimal amount of coefficients. However, it is computationally inefficient since the functions required to compute the transform must be calculated in advance and transmitted to the decoder, rendering its use impractical. Another transform that performs nearly as well as the KLT,

but is much more efficient, is the Discrete Cosine Transform (DCT). The DCT, is a transform similar to the DFT but with real coefficients, representing a discrete signal in terms of a sum of cosines of different frequencies with the energy concentrated at the few top left coefficients representing the low frequencies while the higher frequencies components are sparse, with most of the values being close to 0 as its illustrated in the Figure 1.

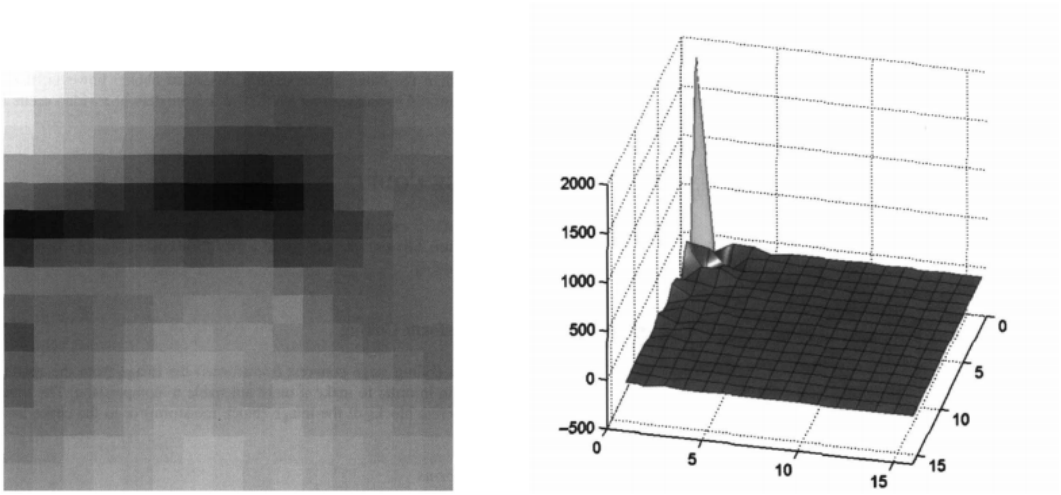

Figure 1: A 16x16 sample(a) image and it's DCT coefficients(b) [2, pp. 35]

Due to the orthonormality of the transformation, the energy in both, the image and DCT domains is the same, hence no information has been reduced. At the same time the energy being concentrated at the low frequency allows for quantization of the image without a significant loss in quality. Furthermore, the human visual cortex is less sensitive to distortions at higher frequencies. Therefore, applying a coarser quntization step, would pass unnoticed by the human eye while improving the compression rate.

The non-zero quantized coefficients are being grouped together scanning through the block in a zigzag sequence [Figure 2], since non-zero coefficients are concentrated at the top left. Such scanning would represent the image as short runs of non-zero values followed by a long runs of zero-valued coefficients and might be efficiently represented as pairs by performing run-length encoding. Furthermore, to represent the frequently occurring runs with shorter codes, a type of entropy encoding, such as Huffman or Arithmetic coding, is applied. Ignoring the quantization step would allow a decoder to perfectly recover the original image at the expense of a lower compression, called lossless-compression, while with quantization, some of the high frequency information would be irreversibly lost, then accordingly named lossy-compression, but allowing higher compression rates to be achieved.

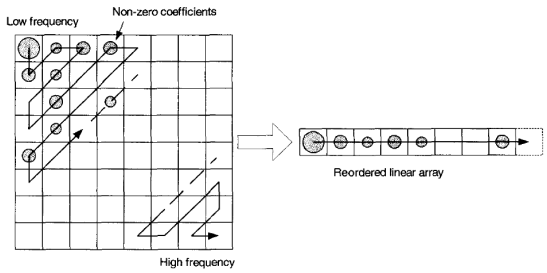

Figure 2: Zigzag scan of Quantized DCT coefficients [2, pp.40]

The sequence of transforming, quantizing, run-length, and entropy encoding represents a typical image compression scheme, which would attempt to remove the spatial

redundancies in a digital image. Using the DCT as the transform of choice and with additional pre-processing steps such as color space conversions and down-sampling, this image codec is well known under the name JPEG which was standardized in 1992 and could achieve a 10:1 compression ratio [4].

2.3 Video Codec

Decent compression ratios can be achieved by encoding each video frame with an image codec. However, better compression results can be achieved by exploiting the temporal redundancy alongside the spatial Redundancy. This is commonly done by means of frame prediction from previous sample frames and transmitting the prediction error to the output. Due to the relative similarity of the neighbouring frames, the frame difference will contain much less information. Modern video compression systems involve more complex ways of predicting a frame by making use of motion estimation and compensation techniques.

Motion estimation is a way of changing a frame that has already been decoded and stored as a reference frame, in order to match the current frame as close as possible. This can be achieved by matching blocks from the reference frame to blocks of the current frame, done by measuring the Mean Square Error (MSE) between the blocks. After the best match is found, it is subtracted from the current frame to produce the residual blocks and alongside the motion vectors are encoded and transmitted to the decoder. Motion Vectors represent the 2-D offset of the matched region relative to the block position in the current frame. Transmitting the residual frame alongside motion vectors is commonly called inter-frame coding.

Due to the close similarity of frames in a sequence, the best match is usually in the

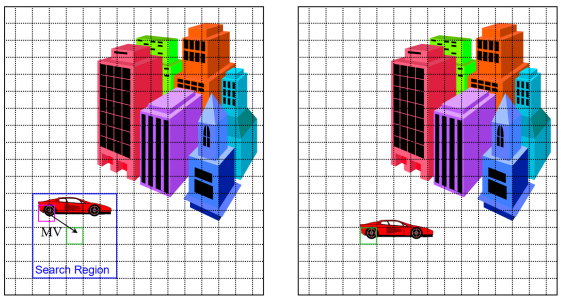

Figure 3: Block Matching in two consecutive frames. [5, week 14]

proximity of the reference block position, limiting the search region in the current frame. Searching the whole current frame for the best match is computationally expensive and usually avoided. Such an algorithm is called Full Search Motion Estimation. However, if a Full Search is needed, other metrics are used, such as Mean Absolute Error (MAE) or the Sum of Absolute Errors (SAE) are used. Both, MAE and SAE are computationally cheaper than MSE and provide a reasonable approximation [2, pp. 99]. Alternatively, fast search algorithms have been developed as an alternative which compare a subset of SAE values providing a significant boost but having the possibility to converge at a local minimum. Some known fast search algorithms are Logarithmic and Hierarchical Search, both which narrow the search region after each iteration [6]. Newer Codecs use sub-pixel motion estimation by performing block-matching with sub-pixel accuracy, resulting

in smaller residuals after the compensation. Sub-pixel motion estimation interpolates between pixels of the search area, up-sampling it respectively, searches at full and sub-pixel locations in the region and compensates at the full or sub-pixel resolution. For example VP9 can achieve up do 1/8 sub-pixel accuracy [7, pp. 29].

Aside from inter-frame coding, some blocks in a frame can be reconstructed from other blocks, by predicting the pixel values, called intra-frame prediction. Once the top and left blocks in a frame have been recovered it is possible to extrapolate their values and predict the rest of the frame. An example is shown in Figure 4. A codec that performs both type of frame prediction is called a Hybrid Motion Compensated Video Codec.

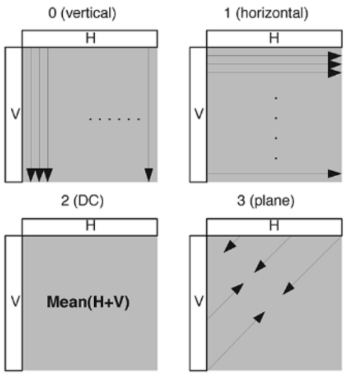

Figure 4: Intra-prediction modes [8, pp.381]

Often reconstructed frames contain blocking artifacts generated by the block based prediction and 2-D transforms. They can be easily spotted at lower bit-rater with the naked eye. In order to reduce the impact of those errors, a loop filter is used to adaptive smooth the sudden discontinuities which cause the artifacts while attempting to preserve the natural edges of the frame.

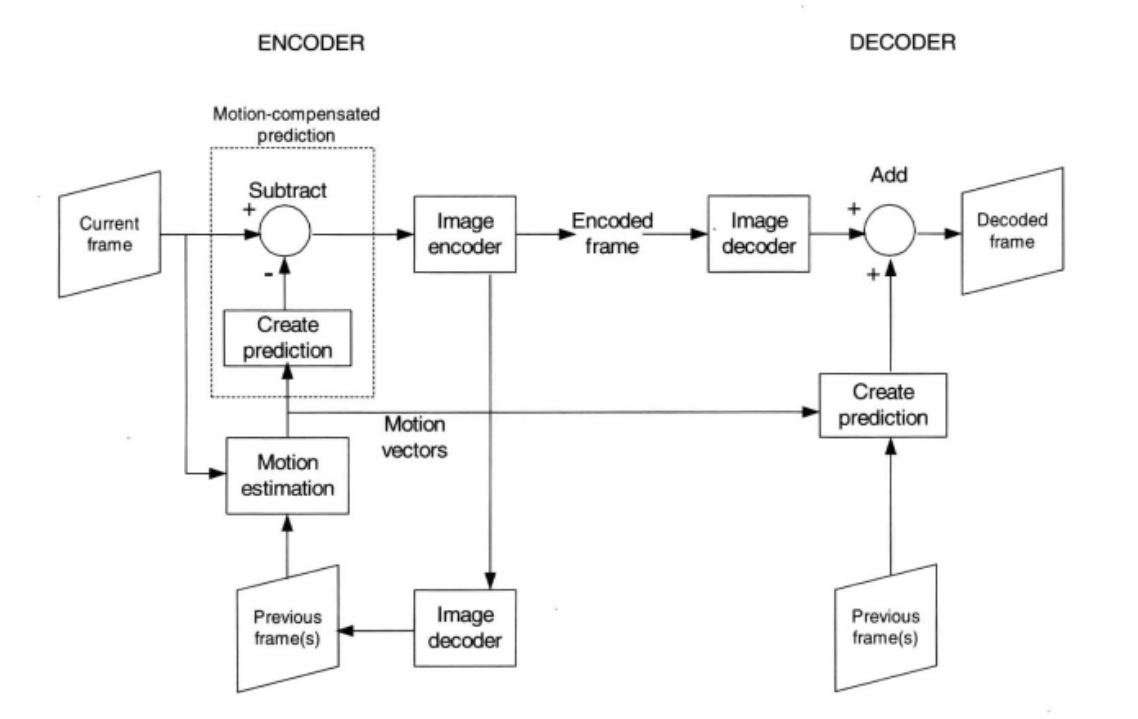

Figure 5: Typical video codec block diagram[2, pp. 44]

A video frame must be as well preprocessed. Generally, choosing a coding profile, which represents the quality of the encoded video, specifying the color format. Due to the fact that the Human Visual Cortex is more sensible to changes in the luminance than colour, the red and green colours spaces of the YUV format can be sub-sampled, for example, for each 4 luminance pixels only 2 chrominance pixels(one red and one green) are taken into account. Aside from choosing a coding profile, the frames have to be partitioned in tiles of macro-blocks which are then processed by the encoder.

A video codec usually encodes the first frame in intra-mode. After the intra-frame, the encoder predicts the further frames and transmits only the residual to the decoder. A frame might be predicted from the future frame as well as from the past ones, the process being called bidirectional prediction. The choice of which frames are intra and which frames are inter coded is up to the particular standard and coding scheme. Since an intra-coded frame take more information than an inter-coded one, the trade-off between compression efficiency and . For example, MPEG-2, encodes the 1st frame as intra-frame, from which it predicts the 4th frame, both of which are used to predict the 2nd and third frames as it's displayed in Figure 6.

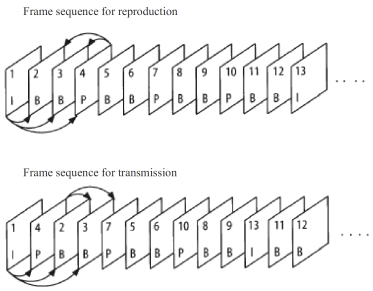

Figure 6: MPEG2 frame sequence [9, pp.98]

At the decoder side, the first intra-frame is entropy decoded, inverse quantized, transformed, intra motion compensated if that is the case, and stored as a reference for for the motion compensation. Further frames arrive as residuals and motion vectors and must be reconstructed by means of inter-prediction. One can clearly observe that this process replicates the image decoding and prediction at the decoder with exception of motion vectors, which don't need to be found, consequently showing that the decoder is a part of the encoder as it can be seen in Figure 5. In a standardized Video Codec, only the bit-stream and the decoding process in order to ensure compatibility across a broad range of devices and applications. The actual encoder and decoder are not fixed and can be implemented process is not fixed and can be changed as long as the output bit-stream complies with the standard [6].

2.4 VP9 coding standard

VP9 is a modern bandwidth-efficient video coding standard which was initially released by Google on the 17th of June 2013. It is a Hybrid Motion Compensated Video codec following the structure mentioned above.

The standard provides 4 coding profiles, numbered 0 to 3. Profiles 0 and 1 allows only 8 bits per color are allowed while 2 and 3 allow 10-12 bits per sample. Profiles 0 and 2 require the chroma pixels sub-sampled in the 4:2:0 format while the other two allow for

for other formats[7, pp. 23].

The frame is partitioned in tiles which contain 64x64 superb-blocks which can be partitioned in sub-blocks of power 2 size down to 4x4 as shown in Figure 7.

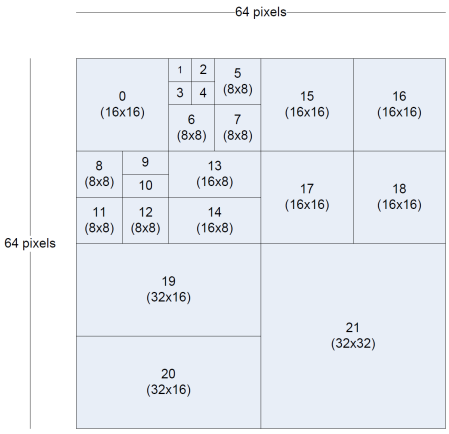

Figure 7: VP9 block partitioning [7, pp. 15]

As entropy coder, VP9 uses an arithmetic coder with the probability model stored for the whole frame.

Due to the varying sub-block sizes, vp9 specifies different 2-D transform of sizes 64x64, 32x32, 16x16, 8x8 and 4x4. The DCT and the Asymmetric Discrete Sine Transforms (ADST) are the used transforms. For the intra-mode, the pixel values near the top and left edges are predicted better than the ones further form away, the errors being concentrated at one side, ADST maximizes the energy at the unknown boundaries making it more fit for those kind of shapes. The DCT respectively is applied to the residuals in inter-mode [7, pp. 16]. The Inverse DCT (IDCT) process the data as 1-D by a series of stages of butterfly operations. The structure is of a 2^n point IDCT which contains the $2^(n-1)$ point IDCT within it. The ADST as well is performed on a 1-D with a butterfly structure [7, pp. 18-19].

2.5 Screen Content Coding

Bibliography

References

[1] M. Ghanbari, *Standard codecs : image compression to advanced video coding*. Institution of Engineering and Technology, 2011, ISBN: 9780863419645.

[2] I. Richardson, *Video codec design : developing image and video compression systems*. Wiley, 2002, ISBN: 9780471485537.

[3] R. Gonzalez, *Digital image processing*. Prentice Hall, 2008, ISBN: 9780131687288.

[4] S. L. Haines Richard F.; Chuang, " The effects of video compression on acceptability of images for monitoring life sciences experiments," NASA, Tech. Rep. NASA-TP-3239, A-92040, NAS 1.60:3239, 1992.

[5] Y. Wang, *Image and video processing lecture notes*, 2015.

[6] J. C. P. Aggelos K. Katsaggelos, *Fundamentals of image and video processing lecture notes*, 2019.

[7] A. G. from Google; Jonathan Hunt from Argon Design; Peter de Rivaz from Argon Design, *Vp9 bit-stream and decoding specification*, Open Standard Video Codec, 2016.

[8] Y.-W. Huang, B.-Y. Hsieh, T.-C. Chen, and L.-G. Chen, "Analysis fast algorithm and vlsi architecture design for h. 264/avc intra frame coder [c]," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 15, pp. 378–401, Apr. 2005. DOI: 10.1109/TCSVT.2004.842620.

[9] P. D.-I. W. Henkel, *Digital signal processing lecture notes*, Feb. 2018.