Istanbul Technical University
Faculty of Computer and Informatics

BLG477E Multimedia Computing
Term Project Report

Cem Yusuf Aydoğdu
150120251

Mehmet Taner Ünal
150130702

Abstract

A video compression standard H.264, has become very popular and useful today. In this work, background and features of H.264 are presented. Also, x264 and terms like frame, field and slice are defined. Besides, there are many use areas of the H.264 standard, and usually different areas require different approaches. Solutions of this need, profiles like baseline, main, extended, and high profiles are explained in detail. On x264 codec in the ffmpeg software, H.264/AVC standard is tested for some video samples. Results showed that PSNR (Peak signal noise ratio) values change significantly between different videos and, larger PSNR and SSIM (Structural Similarity Metric) values correspond to better quality.

Introduction

Background and History

A video compression standard H.264, in the other words a codec, was cooperatively advanced by the International Telecommunications Union and International Organization for Standardization/International Electrotechnical Commission Moving Picture Experts Group. It was known as "H.264" by ITU and "MPEG-4 Part 10, Advanced Video Coding (AVC)" by ISO/IEC. So they both are basically the same thing [14]. H.264/AVC standardisation's main objectives are to increase compression performance and to boost provision of a network-friendly video representation addressing conversational (video telephony) and non-conversational (storage, streaming, or broadcasting) applications [15].

To talk about the history of H.264, it is good to have some background information and to understand the reason that made it a need. MPEG-2 which is also known as ITU-T H.262, was an extension of previous MPEG-1 video capability with support of interlaced video coding and it was a facility to worldwide digital television systems. But as the number of services and popularity grow, the need for a higher coding efficiency is increased. Besides, current transmission speed was not sufficient for media. With the development of the ITU-T H.261, H.262 (MPEG-2) and H.263, video coding for telecommunication applications started getting more powerful. Continuous efforts were on maximising coding efficiency while dealing with the diversification of network types and their characteristic formatting and loss/error sturdiness prerequisite. In early 1998, the Video Coding Experts Group made proposals on a project named H.26L. Their goal was to double the coding efficiency over the other existing video coding standards for a wide variety of applications. The first draft design was completed in October of 1999. About December of 2001, a Joint Video Team is formed. They came up with a settlement to conclude the draft new video coding standard for formal approval submission as H.264/AVC in March 2003 [15]. The scope of the standardisation is illustrated in Fig. 1.

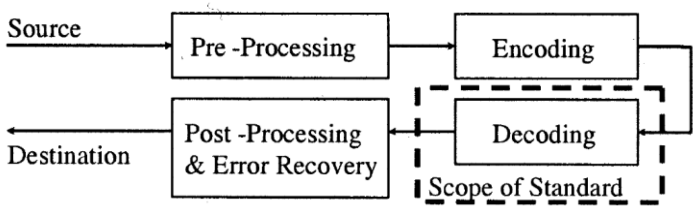

Fig 1: Scope of video coding standardisation [15]

Improvements and Methods

By improvements in technology today such as high-speed chips, H.264 can convey MPEG-4 quality with a frame size up to four times greater. Additionally, it grants MPEG-2 quality at a downsized data rate [16]. Using motion estimation, which lessens temporal redundancies; intra estimation, which lessens spatial redundancies; transformation of motion estimation and intra estimation into the frequency domain; reduction of compression artifacts; and entropy coding, which assigns a smaller number of bits to frequently encountered symbols and a larger number of bits to infrequently encountered symbols better compression and perceptual quality of H.264 are achieved [17].

Some Information about H.264 and Some Comparison with MPEG-4

H.264 is promising to be the next standard for format convergence in the digital video industry disregarding of the platform. H.264 is backed by big internet players like Google/YouTube, Adobe, and Apple iTunes [18]. Also, since H.264 is cooperatively maintained with MPEG, H.264 and MPEG-4 have identical technical content [18]. To obtain very exact portrayal of the displacements of moving areas, H.264 grants quarter-pixel precision for motion compensation. The resolution is halved both vertically and horizontally for chroma. So, one-eight chroma pixel grid units is used by the motion compensation of chroma [18]. For video network delivery and for delivery of high definition video, H.264 is more charming than MPEG-4 [18].

H.264 is very open to new optimisations. For example, use of software-hardware-co-design for the entire entropy coding system, more optimisation can be achieved. In the given example, arithmetic coding, context managing, and the most demanding tasks in the data binarisation are taken over by a hardware accelerator [19].

X264 Definition

"x264 is a free software library and application for encoding video streams into the H.264/MPEG-4 AVC compression format, and is released under the terms of the GNU GPL [20]."

Softwares like Avidemux, ELDER, ffdshow, ffmpeg, GordianKnot, Handbrake, LiVES, MeGUI, MEncoder, Bencos (formerly RealAnime), StaxRip, VLC Media Player use x264 [20].

System

Introduction

In order to examine the system in detail, one should study the terminology first. According to Glossary of Telecommunication terms published by National Communication System, the term *frame* refers to a full image consist of several images; the term *field* refers to a partial frame which consists n_{th} line of a frame [5]. In an interlaced video system, a frame consists of two fields, top and bottom fields for even and odd numbered lines respectively [1] [2].

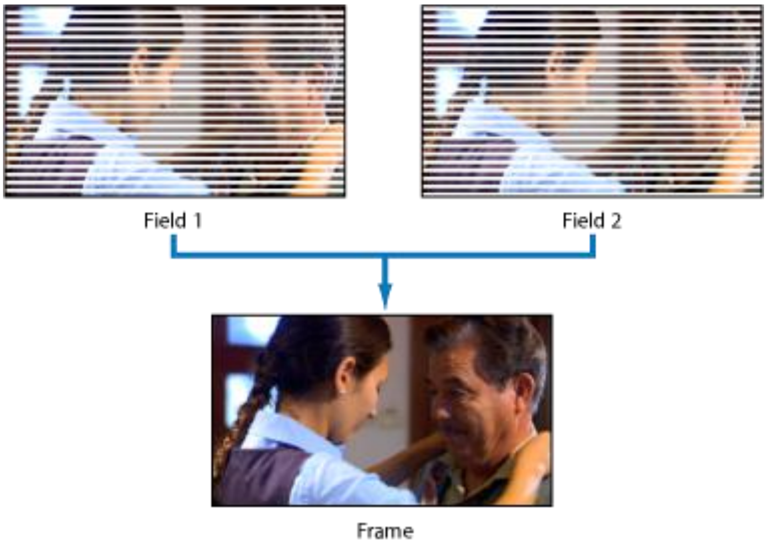

Fig 2: relation between a frame and its fields [6].

Coded picture denotes a compressed video frame, m*acroblock* refers to a fixed sized region in the frame or coded picture, with 16x16 luma, 8x8 Cr and 8x8 Cb components, and *slice* refers to “a set of macroblocks”[1]. There are five slice types in the H.264/AVC, *I*, *P*, *B*, *SI* and *SP* [3]. *Intra I* slices are produced using current frame, whereas “predictive” *P* and “bi-predictive” *B* slices are produced using previously referenced frames [3]. Furthermore, “switching P” *SP* and “switching I” *SI* slices refer to changing bit rates of encoding [3].

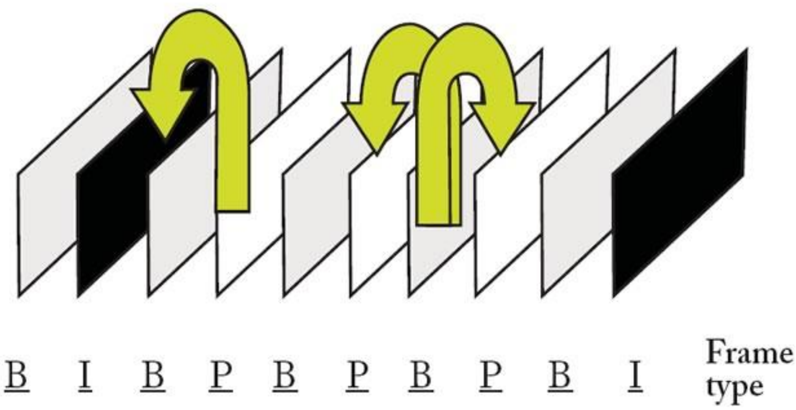

Fig 3: Different frame types based on *I*, *P*, *B* slices [7].

Overview of Encoder and Decoder

Encoder and decoder block diagrams are shown below. In order to create coded picture, the system uses previous reference frames with the current frame. In the diagram of encoder below, ME stands for Motion Estimation, MC stands for Motion Compensation before inter-intra options, T stands for transform, Q stands for quantization and NAL stands for Network Abstraction Layer [1]. In the Inter prediction, spatial samples from previously coded frames are used, but in Intra prediction, spatially nearby blocks are referred in the current field [3]. H.264 standard composes both approaches in the encoding phase [8].

In the upper path through encoding to network abstraction layer, a prediction is calculated using inter or intra modes for each macroblock in the current frame initially [1]. After, that prediction is extracted from the current block, which results a residual (difference) block D_n [1]. Then, that residual block is sent to transform and quantization phases. In the transform phase, Integer Transform method is used rather than Discrete Cosine Transform (DCT) which was used in older standards, because Discrete Cosine Transform (DCT) introduces prediction shift due to quantization error in floating point calculations [8]. In the quantization phase, uniform quantization parameters are used with the zig-zag scanning method [2] [3]. After transform and quantization phases, resulting quantization coefficients (denoted with X) are encoded using various entropy coding techniques such as Context Adaptive Variable Length Coding (CAVLC) and Context Adaptive Binary Entropy Coding (CABAC) [1] [2]. Finally, encoded data are arranged for the network abstraction layer (NAL) which consists of stream of bytes with necessary headers [2].

In the reconstruction path below of the block diagram, reconstructed blocks are formed by using prediction blocks with inverse quantizied and inverse transformed blocks [1]. These blocks are also included in intra prediction phase, alongside with the filtering [1].

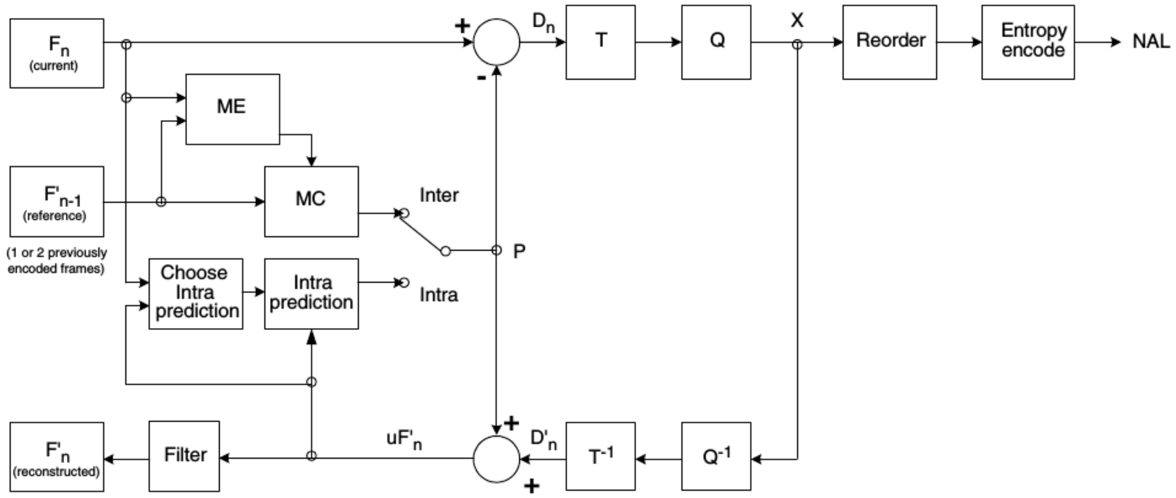

Fig 4: System blocks of H.264 encoder [1].

Similarly, operations done in the encoder are reversed in the decoder. First, compressed data taken from the Network Abstraction Layer are re-ordered. Then, the data which consists of quantization coefficients are inverse quantizied and inverse transformed using same principles as in the encoding stage. After that, output blocks from inverse transformation are merged with prediction blocks which are estimated from reference frames and finally filtered to produce reconstructed blocks.

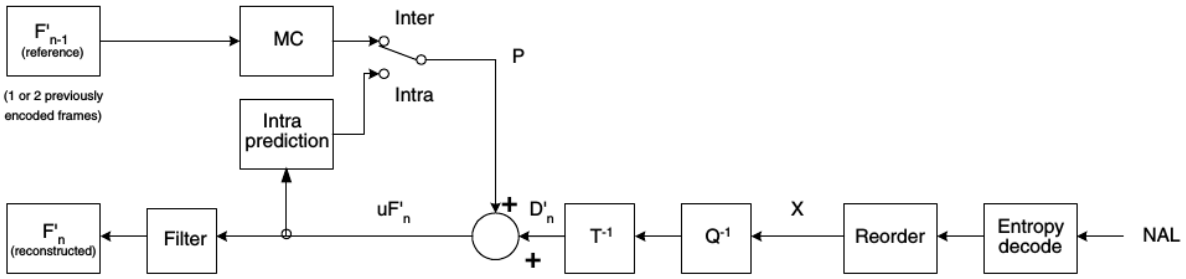

Fig 5: System blocks of the H.264 decoder [1].

Profiles and Levels in the H.264 Standard

There are many usage areas of the H.264 standard, and usually different areas require different technical requirements. Profiles specify algorithms and coding instruments which are implemented in the encoding or decoding of video frames, and levels refer to constraints which are applied in the process of producing output bitstream [2]. H.264 standard introduces three profiles, Baseline Profile, Main Profile and Extended profile. However, new "High" profiles introduced with the Fidelity Range Extensions Amendment in 2004, for better quality requirements [4].

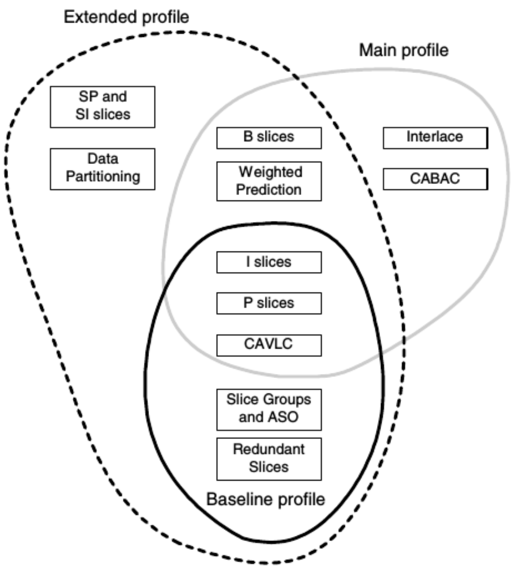

Fig 6: Overview of three profiles in the H.264 standard [1].

1. Baseline Profile

This profile was introduced to support basic functionalities for real-time applications like teleconferencing or mobile videos [9]. This profile provides sequences with I slices which consist of intra-coded blocks and P slices which consist intra-coded, inter-coded or skipped blocks [1]. After transformation, quantization and re-ordering of difference blocks, resulting coefficients are coded using Context Adaptive Variable Length Coding (CAVLC) [1]. The baseline profile consists several tools in addition to fundamental tools, such as Arbitrary Slice Order (ASO), Flexible Macroblock Order (FMO) and Redundant Slices [8].

2. Main Profile

The main profile was designed for storing digital video and video broadcasting implementations like digital television [1]. This profile shares some of additional features with the baseline profile such as I slices, P slices and Context Adaptive Variable Length Coding (CAVLC) entropy coding technique. However, it has other features, which are B slices, Weighted Prediction, Interlace and Context Adaptive Binary Arithmetic Coding (CABAC) [1].

3. Extended Profile

The Extended Profile which is also known as X Profile, was intended for utilization of video streaming [8]. It consists of all features of Baseline profile, with additional features such as B slices and Weighted Prediction like Main Profile, and also SI slices, SP slices and Data Partitioning [1].

4. High Profiles

Four new high profile types were introduced into the standard with Fidelity Range Extension Amendment in 2004 [4]. These profiles are High, High 10 (Hi10P), High 4:2:2 (Hi422P), High 4:4:4 (Hi444P) [4]. Latest features in these profiles are adaptive block switching between 8x8 - 4x4, perceptual based quantization scaling matrices, separate control of quantization parameters for each chroma component [4]. A comparison table of these high profiles is given below.

Coding Tools	High	High 10	High 4:2:2	High 4:4:4
Main Profile Tools	X	X	X	X
4:2:0 Chroma Format	X	X	X	X
8 Bit Sample Bit Depth	X	X	X	X
8x8 vs. 4x4 Transform Adaptivity	X	X	X	X
Quantization Scaling Matrices	X	X	X	X
Separate Cb and Cr QP control	X	X	X	X
Monochrome video format	X	X	X	X
9 and 10 Bit Sample Bit Depth		X	X	X
4:2:2 Chroma Format			X	X
11 and 12 Bit Sample Bit Depth				X
4:4:4 Chroma Format				X
Residual Color Transform				X
Predictive Lossless Coding				X

Fig 7: Comparison of high profiles [4].

Experimental Results

In this section, H.264/AVC standard is tested using command line interface of the x264 codec in the FFmpeg software. Uncompressed video samples “coastguard”, “container”, “news”, “riverbed” from “Derf’s Test Media Collection” are used for experiments [10]. Information about these raw videos is given below.

Video Name	Resolution	Frame Per Second	Chroma Format
coastguard	352 x 288	29.97	4 : 2 : 0
container	352 x 288	29.97	4 : 2 : 0
news	352 x 288	29.97	4 : 2 : 0
riverbed	1920 x 1080	25	4 : 2 : 0

In the testing, input videos are encoded with x264 encoder using baseline, main and extended profiles, then PSNR and SSIM values are calculated by comparing encoded outputs and uncompressed raw inputs. PSNR refers to “Peak Signal to Noise Ratio” and SSIM refers to “Structural Similarity Metric” [11]. PSNR is calculated by the formula below.

In this formula, is average of the maximum values in each image component and is mean square error [11]. Average PSNR values of each different type of encoded video are shown in the below chart. Despite the fact that same profile settings are used in every video, PSNR values are varying significantly between different videos. Larger values correspond to better quality.

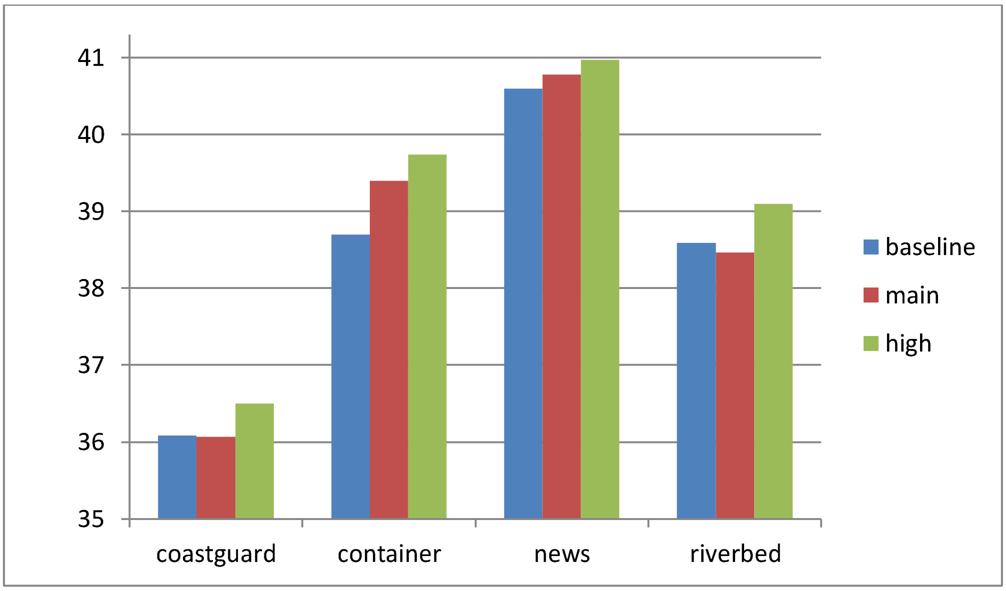

Chart 1: Average PSNR values of encoded raw videos

Structural Similarity Metric combines luminance, contrast and structure comparison of two input signals in order to create a better performance measurement tuned to human visual system [12]. Calculation diagram and formula of the SSIM are given below:

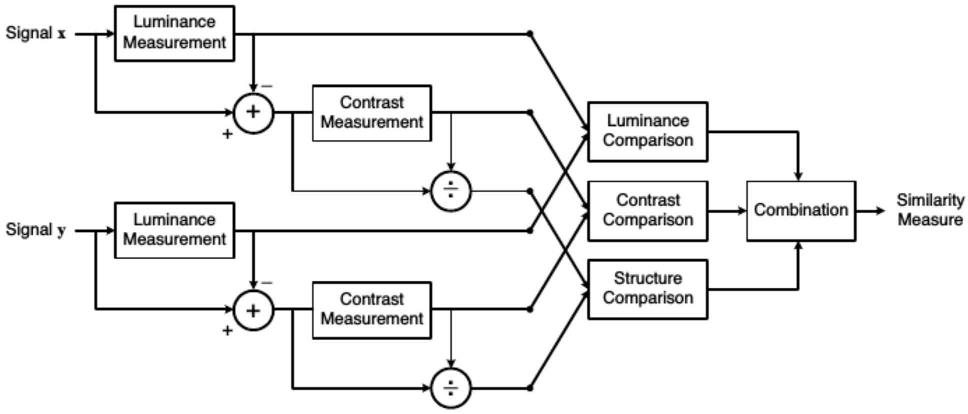

Fig 8: Calculation flowchart diagram of the SSIM measurement procesure [12].

$$SSIM(x, y) = [l(x, y)]^{\alpha} \cdot [c(x, y)]^{\beta} \cdot [s(x, y)]^{\gamma}$$

where

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1},$$

$$c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2},$$

$$s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}$$

Fig ** : Formula for calculating SSIM index. μ values denote local means and σ values denote standard deviations [13].

SSIM values of encoded videos are shown below. Larger values also correspond to better quality. Results are similar with the PSNR values.

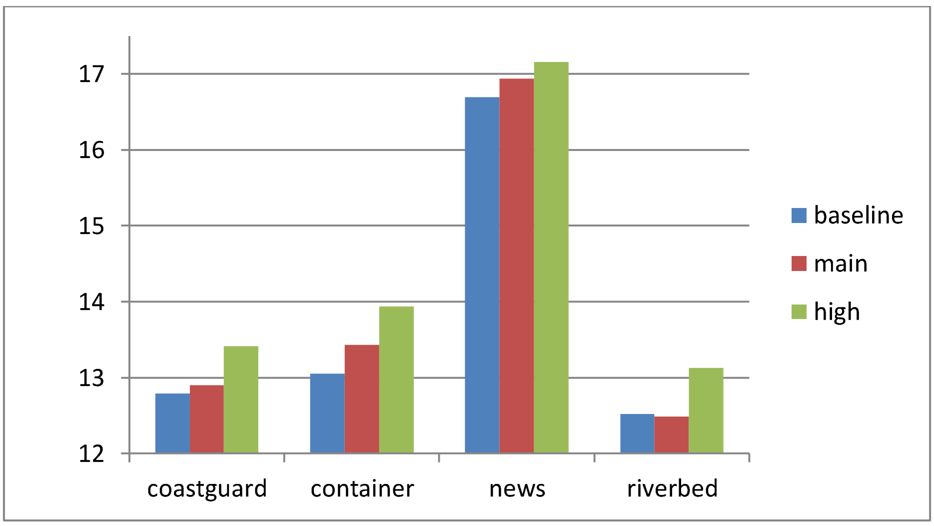

Chart 2: SSIM values of videos which are encoded in different profiles

Bash script used in the test is given below. Code is utilized for encoding videos, calculating PSNR and SSIM values and merging videos.

```
#! /bin/bash
RAW_VIDEO="raw.y4m"

echo "Input video name:" $RAW_VIDEO
echo ""
echo "1: Encode video in baseline profile"
echo "2: Encode video in main profile"
echo "3: Encode video in high profile"
echo "4: Calculate PSNR and SSIM"
echo "5: Combine two videos"
echo ""
echo "Select option:"
read option

if [ $option -eq 1 ]; then
    eval "ffmpeg -i $RAW_VIDEO -c:v libx264 -report -ssim 1 -psnr -profile:v baseline out_baseline.mp4";
fi

if [ $option -eq 2 ]; then
    eval "ffmpeg -i $RAW_VIDEO -c:v libx264 -report -ssim 1 -psnr -profile:v main out_main.mp4";
fi

if [ $option -eq 3 ]; then
    eval "ffmpeg -i $RAW_VIDEO -c:v libx264 -report -ssim 1 -psnr -profile:v high out_high.mp4";
fi

if [ $option -eq 4 ]; then
    echo "Encoded video:"
    read ENCODED

    echo "Reference video:"
    read REF
    eval "ffmpeg -i $ENCODED -i $REF -lavfi  \"ssim;[0:v][1:v]psnr\" -f null -"
fi

if [ $option -eq 5 ]; then
    echo "First video (will be on the left):"
    read FIRST

    echo "Second video (will be on the right):"
    read SECOND

    echo "Output name"
    read OUTPUT

    eval "ffmpeg -i $FIRST -i $SECOND -filter_complex \"[0:v:0]pad=iw*2:ih[bg]; [bg][1:v:0]overlay=w\"
    $OUTPUT"
fi
```

Resources

[1] I. Richardson, *H.264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia*. Wiley, 2003.

[2] T. Wiegand, G. J. Sullivan, G. Bjontegaard and A. Luthra, "Overview of the H.264/AVC video coding standard," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560-576, July 2003.

[3] D. Marpe, T. Wiegand and G. J. Sullivan, "The H.264/MPEG4 advanced video coding standard and its applications," in *IEEE Communications Magazine*, vol. 44, no. 8, pp. 134-143, Aug. 2006.

[4] Sullivan et. al, “The H. 264/AVC advanced video coding standard: Overview and introduction to the fidelity range extensions,” presented at the Optical Science and Technology, the SPIE 49th Annual Meeting, 2004.

[5] “Federal Standard 1037C: Glossary of Telecommunications Terms.” [Online]. Available: http://www.its.bldrdoc.gov/fs-1037/fs-1037c.htm. [Accessed: 08-May-2016].

[6] “Final Cut Pro 7 User Manual: Scanning Method.” [Online]. Available: https://documentation.apple.com/en/finalcutpro/usermanual/index.html#chapter=C%26section=9%26tasks=true. [Accessed: 08-May-2016].

[7] J. Ozer, “Encoding options for H.264 video | Adobe Developer Connection.” [Online]. Available: https://www.adobe.com/devnet/adobe-media-server/articles/h264_encoding.html. [Accessed: 08-May-2016].

[8] Z. Li, M. Drew, and J. Liu, *Fundamentals of Multimedia*. Springer, 2014.

[9] B. Juurling et. al., “Understanding the Application: An Overview of the H.264 Standard,” in *Scalable Parallel Programming Applied to H.264/AVC Decoding*, Springer, 2012.

[10] “Xiph.org :: Derf’s Test Media Collection.” [Online]. Available: https://media.xiph.org/video/derf/. [Accessed: 08-May-2016].

[11] “FFmpeg Filters Documentation.” [Online]. Available: https://ffmpeg.org/ffmpeg-filters.html. [Accessed: 08-May-2016].

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

[13] “Structural Similarity Index (SSIM) for measuring image quality - MATLAB ssim.” [Online]. Available: https://www.mathworks.com/help/images/ref/ssim.html. [Accessed: 08-May-2016].

[14] J. Ozer, "What is H.264? - Streaming Media Magazine", *Streaming Media Magazine*, 2011. [Online]. Available: http://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-is-H.264-74735.aspx. [Accessed: 09- May- 2016].

[15] T. Wiegand, G. J. Sullivan, G. Bjontegaard and A. Luthra, "Overview of the H.264/AVC video coding standard," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560-576, July 2003. doi: 10.1109/TCSVT.2003.815165

[16] "H.264 Definition from PC Magazine Encyclopedia", *Pcmag.com*, 2016. [Online]. Available: http://www.pcmag.com/encyclopedia/term/44034/h-264. [Accessed: 09- May-2016].

[17] "What is H.264 (MPEG-4 AVC)? - Definition from WhatIs.com", *WhatIs.com*, 2016. [Online]. Available: http://whatis.techtarget.com/definition/H264-MPEG-4-AVC. [Accessed: 09- May- 2016].

[18] "H.264 versus MPEG-4 - Video Encoding Formats Compared", *ReelSEO*, 2007. [Online]. Available: http://www.reelseo.com/encoding-formats-mpeg4-vs-h264/. [Accessed: 09- May- 2016].

[19] R. R. Osorio and J. D. Bruguera, "A new architecture for fast arithmetic coding in H.264 advanced video coder," *8th Euromicro Conference on Digital System Design (DSD'05)*, 2005, pp. 298-305.doi: 10.1109/DSD.2005.9

[20] "VideoLAN - x264, the best H.264/AVC encoder", *Videolan.org*, 2016. [Online]. Available: http://www.videolan.org/developers/x264.html. [Accessed: 09- May- 2016].