Image compression with xvc

Jonatan Samuelsson
Divideon
jonatan.samuelsson@divideon.com
https://www.divideon.com

Per Hermansson
Divideon
per.hermansson@divideon.com
https://www.divideon.com

Abstract

This paper describes xvc – a format for efficient compression of visual data – originally developed for compression of video sequences, but in the context of this paper applied to still images. The xvc codec is a block based lossy codec using a traditional approach for prediction, residual representation and entropy coding. There are no elements of Machine Learning or Artificial Neural Networks in the xvc encoder or decoder. The xvc codec offers support for tuning towards PSNR or perceptual quality. The images submitted for the CLIC challenge and the descriptions included in this paper are based on the perceptually tuned setting.

1. Introduction

The xvc codec is a compression format for video and images that was first released in Septemeber 2017 [1]. The source code of the xvc project is publicly available, and can be accessed through xvc.io [2] or directly from GitHub [3]. The technology included in xvc is inspired by the technology in AVC [4], HEVC [5] and JEM [6] but there is no level of interoperability between xvc and any of these codecs or exploration models, and xvc includes technology that is not present in any of the other codecs. The xvc codec is designed to offer better compression performance than other video codecs and image codecs that exists in the market, without introducing increased computational complexity for the decoder.

2. Technology

The compression process in the xvc codec can be divided into several parts, and each of these parts will be described in this paper. It should be noted that several of these parts have corresponding processing steps in both the encoder and the decoder, but there are some steps that are only performed in the encoder. Section 2.1 presents the parts which are common or corresponding in both the encoder and the decoder, while section 2.2 presents the steps that are only performed in the encoder.

2.1. Processing parts common in encoder and decoder

2.1.1 Color conversion

The input sample data used in the CLIC challenge [7] is represented as RGB data with 8-bit sample depth. In order to decorrelate the color information, the RGB data is converted to YUV data (one luma channel and two chroma channels, sometimes denoted Y'CbCr) using the color matrix defined in BT.601 [8]. Internally in the xvc encoder and decoder, the YUV data is stored in a 10-bit representation in 4:4:4 chroma format, i.e. no supbsampling is applied to the chroma channels. After decoding the compressed images, the xvc decoder converts the YUV data back to RGB. For the purpose of the CLIC challenge, the lodePNG library [9] is used to store the images as PNG files.

2.1.2 Block partitioning

In xvc, the data is processed in blocks of size 64x64 samples called Coding Tree Units (CTUs). Each CTU can be split with a horizontal split, a vertical split, or a quad-split, resulting in two or four Coding Units (CUs). A CU can be split further, recursively forming a tree of coding units and it is at the leafs of this tree that prediction and transform is applied. Each CTU consists of one coding unit tree for luma and another coding unit tree shared by the two chroma channels.

2.1.3 Prediction

Sample data in a CU is predicted from previously coded CUs above and/or to the left of the current CU. The xvc codec supports DC prediction, 65 different angular intra prediction modes and a specific mode called “planar mode” which uses a bi-linear weighting function to predict each sample in the current CU, based on the neighboring samples in adjacent CUs.

2.1.4 Cross component prediction

Even though a transformation from RGB to YUV reduces correlation between the different color channels, it is still quite common that some level of correlation exists between the luma channel and the two chroma channels, for example at an object boundary. In xvc, this correlation is exploited in a specific mode that constructs a linear model for predicting chroma information from the luma information [10].

2.1.5 Transforms and quantization

The difference between the original sample values and the predicted sample values is called the residual. Residual data is transformed using one of several two-dimensional separable transforms in xvc. In total there are five different DCT-based and DST-based transforms available in xvc. The transform coefficients are quantized and signaled in reverse order in subblocks consisting of 4x4 coefficients and with a scanning pattern that depends on the prediction direction. In the decoder, the inverse process is applied, where the coefficients are first parsed, then dequantized and then transformed with the inverse transform. A quantization parameter (QP) is used to determine how the transform coefficients are interpreted. A higher QP means larger steps between the coded values which leads to lower quality and lower compressed image size. When xvc is run with perceptual tuning, different QP values are used for different CTUs as described in section 2.2.2.

2.1.6 Sign data hiding

The concept of Sign Data Hiding was proposed during the HEVC standardization [11] and included in the HEVC specification. The concept exploits the fact that among a set of non-zero coefficients it is typically possible to change the rounding of on of the coefficients with very minor impact on the resulting residual. By calculating the combined parity (odd or even) of the absolute value of all coefficients in a 4x4 subblock, the sign of one of these coefficients can be hidden. If the parity is "wrong", one of the coefficients will be changed and rounded differently to produce the right parity, and thereby the right sign.

2.1.7 Entropy coding

The xvc codec uses a context adaptive binary arithmetic coder (CABAC) which is the same entropy coder as is used in AVC [4] and HEVC [5]. In CABAC, all symbols are translated to binary representations for which the probabilities are adjusted throughout the coding of a picture.

2.1.8 Deblocking

Since the sample data is processed in blocks, there is a risk that edges will become visible, especially at high QP levels. By analyzing the edges between different CUs, different amount of edge filtering can be applied based on the sample values so as to remove visual blocking artifacts without significantly blurring or distorting image information.

2.2. Processing parts only present in encoder

There are several processing steps in xvc that are performed in the encoder for which there is no corresponding processing steps in the decoder. This causes the encoder to be significantly more computational complex compared to the decoder.

2.2.1 Rate-Distortion Optimization

One of the most important concepts in modern video encoders is Rate-Distortion Optimization [12]. The central idea in RDO is to perform evaluation of several different options during the encoding process and select the option that minimizes the cost function:

$$J = D + \lambda R \tag{1}$$

where J is the cost, D is distortion, λ is a variable derived from the QP and R is the rate (estimated or actual). When rate-distortion optimization is applied, the encoding process will result in the best possible quality at as low rate as possible, but it is typically not possible to determine in advance what the level of the quality or what the rate will be. However, the QP parameter (and thereby the lambda) can be changed during encoding or in an outer loop to reach a specific target quality or target rate. In the CLIC challenge [7] there is a requirement to not exceed a total size for all pictures corresponding to 0.15 bits per pixel. For xvc, this is achieved by encoding all pictures at several different QP levels and selecting a set that gives as high quality as possible without exceeding the limit. For tuning towards PSNR this process can be completely automated, but for perceptual tuning, visual inspection was also performed to ensure a consistent quality level among the pictures, since the adaptive QP method described in the next section doesn't include any normalization step.

2.2.2 Adaptive QP selection

Block based coding formats such as JPEG, AVC, HEVC and xvc gives rise to compression errors that become more visible at lower rates. These compression artifacts are typically identified as "blockiness", "ringing", "bluriness" etc. and they are typically more apparent in areas with low textural information (flat areas) even though the distortion, when measured with mean square error (MSE), might be of similar magnitude in all areas of the picture. When perceptual tuning is applied in xvc, the QP for each CTU is selected based on the median variance of the 16x16 subblocks of the CTU. If the variance is low, the QP is decreased since low variance corresponds to low level of textural information. Conversely, if the variance is high, the QP is increased, since blocks with high variance will have a high level of textural information which can mask compression errors. The QP is allowed to change between -3 and +7 relative to the nominal QP.

2.2.3 Perceptual distortion function

A well known problem when tuning towards PSNR is that the encoded images may look to soft and blurry. This is mainly due to that the distortion function only looks at squared errors one sample at a time and there is no account for other characteristics of the sample in combination with its neighboring samples. When perceptual tuning is applied in xvc, a different distortion function is used to reduce the blurring and better match the perceived quality.

The distortion function consists of a weighting of MSE and structural similarity (SSIM) [13], with weights that depend on the QP used to encode the block. By including an SSIM term in the distortion function, the encoder is more likely to make RDO decisions that maintain a similar level of structural information as the original image.

3. Subjective quality

The xvc encoder includes support for two different tuning options; perceptual tuning and PSNR tuning. The default setting is perceptual tuning, and that is also the setting that is used for the CLIC challenge [7]. Annex A. shows a visual example of the difference between PSNR tuning (Figure 2) and perceptual tuning (Figure 3) at the same compressed image size. An image encoded with JPEG at the same compressed image size is also included for reference (Figure 1). The JPEG encoding was created with FFmpeg [14] using "-q:v 26" to attain the right compression level.

References

[1] Divideon, "Divideon Introduces xvc – A World-Class Video Codec with a Revolutionary Licensing Model" http://www.releasewire.com/press-releases/divideon/release-863489.htm

[2] Divideon, "xvc – a world class video codec – with indemnificaiton" https://xvc.io/

[3] Divideon, "xvc GitHub repository" https://github.com/divideon/xvc

[4] ITU-T/ISO/IEC "H.264/14496-10 Advanced Video Coding" https://www.itu.int/rec/T-REC-H.264

[5] ITU-T/ISO/IEC "H.265/23008-2 High Efficiency Video Coding" https://www.itu.int/rec/T-REC-H.265

[6] JVET "JEM Software" https://jvet.hhi.fraunhofer.de/

[7] CLIC "Workshop and Challenge on Learned Image Compression" http://www.compression.cc/

[8] ITU-R "BT.601 Studio encoding parameters of digital television for standard 4:3 and wide screen 16:9 aspect ratios" https://www.itu.int/rec/R-REC-BT.601/en

[9] L. Vandevenne "LodePNG" http://lodev.org/lodepng/

[10] J. Chen, E. Alshina, G.-J. Sullivan, J.-R. Ohm, J. Boyce "Algorithm Description of Joint Exploration Test Model 1" Feb. 2016. http://phenix.int-evry.fr/jvet/doc_end_user/current_document.php?id=2610

[11] G. Clare, F. Henry, J. Jung (Orange Labs) "Sign Data Hiding" Nov. 2011, http://phenix.int-evry.fr/jct/doc_end_user/current_document.php?id=3528

[12] G.J. Sullivan, T. Wiegand "Rate-distortion optimization for video compression" IEEE Signal Processing Magazine 15 (6), 74-90. Nov. 1998

[13] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli "Image quality assessment: from error visibility to structural similarity" IEEE Transactions on Image Processing, Volume: 13, Issue: 4, April 2004

[14] FFmpeg "A complete, cross-platform solution to record, convert and stream audio and video." http://ffmpeg.org/

Annex A. Example images for visual comparison

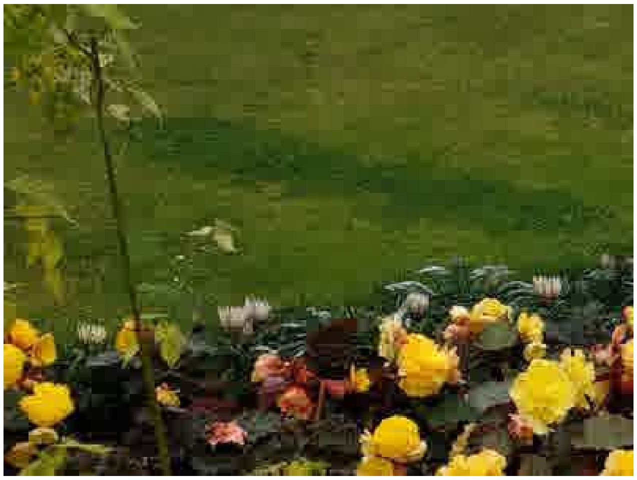

Figure 1. JPEG at 0.65 bpp

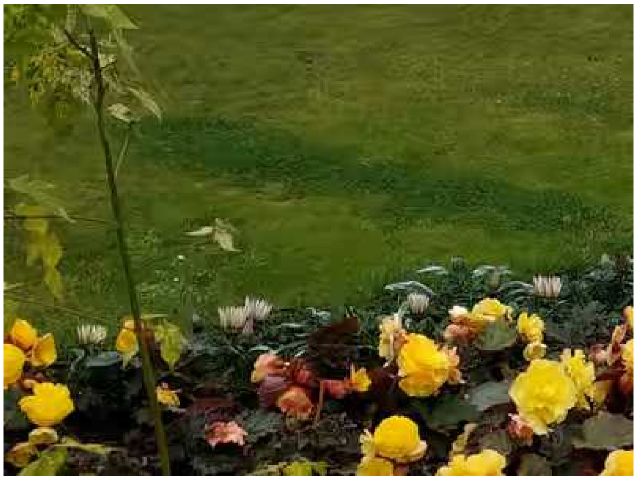

Figure 2. xvc with PSNR tuning at 0.65 bpp

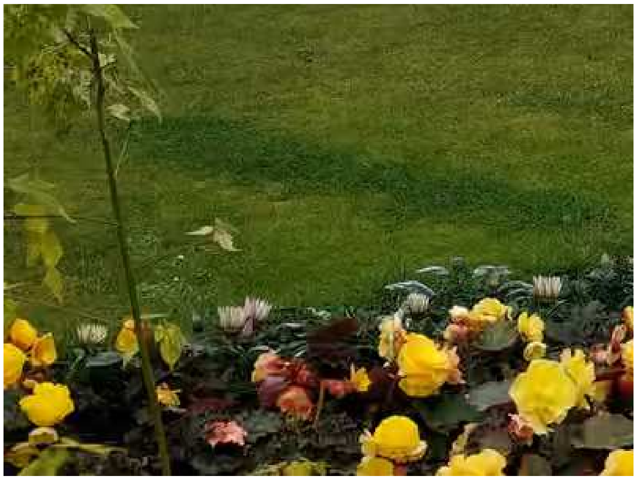

Figure 3. xvc with perceptual tuning at 0.65 bpp