

Neural Video Compression Based on PVQ Algorithm

Michał Knop^{1,2(✉)}, Tomasz Kapuściński^{1,2}, and Rafał Angryk³

¹ Institute of Computational Intelligence, Czestochowa University of Technology,
Al. Armii Krajowej 36, 42-200 Czestochowa, Poland
{michal.knop,tomasz.kapuscinski}@iisi.pcz.pl

² Institute of Information Technology, Radom Academy of Economics,
Domagalskiego Street 7a, 26-600 Radom, Poland

³ Department of Computer Science, Georgia State University,
P.O. Box 5060, Atlanta, GA 30302-5060, USA
angryk@cs.gsu.edu
<http://www.iisi.pcz.pl>
<http://wsh.pl/>
<http://grid.cs.gsu.edu/~rangryk/>

Abstract. In this paper we present a video compression algorithm based on predictive vector quantization, which is a combination of vector quantization and differential pulse code modulation. We optimized the algorithm using chroma subsampling which reduces the amount of information that needs to be processed. This allowed us to combine two color channels into one and thereby reduce the number of predictors and codebooks. Furthermore, we introduced inter-frames which only store regions that changed compared to previous frames, further decreasing the size of compressed data.

Keywords: Video compression · Image compression · PVQ

1 Introduction

Video and image data compression has become an increasingly important issue in all areas of computing and communications. Various techniques for encoding of data can be used to eliminate the redundancy of the color information in images and video frames. Most video codecs and algorithms combine spatial compensation of images as well as compensation of movement in time. Currently, there are many compression standards. They can be found in a wide range of applications such as cable and land-based transmission channels, video services over the satellite, video streaming in Internet or local area network and storage formats. The most popular of these algorithms are MPEG, JPEG and H.26x. The MPEG standard describes a family of compression algorithms of audiovisual data, more details can be find in [5]. The well known members of MPEG family are MPEG-1, MPEG-2, and MPEG-4. H.261 is the first of the entire family H.26x of video compression standards. It has been designed for handling video

transmission in real time. More information about the family H.26x can be found in [1]. JPEG and JPEG2000 standards are used for image compression with an adjustable compression rate. They are also used for video compression. These methods compress each movie frames individually.

In the proposed approach we used a Predictive Vector Quantization (PVQ) algorithm to compress a sequence of video frames, which combines two methods: Vector Quantization (VQ) and Differential Pulse Code Modulation (DPCM) [2,4]. In order to reduce the amount of information needed to store the video stream we used two types of frames: key frames (intra-frame) and predictive frames (inter-frame). These frames were joined to groups containing one key frame and a series of predictive frames called GOP [14,17].

The rest of the paper is organized as follows. Section 2 describes components of our algorithm. It includes a description of neural image compression and the encoding color information. In Sect. 3 we discuss our approach to neural video compression. Next in Sect. 4 the experimental results are presented. The final section covers conclusions and the plans for future works.

2 Related Works

2.1 Predictive Vector Quantization

Predictive Vector Quantization is a neural algorithm that extends differential pulse code modulation (DPCM) scheme with vector quantization method (VQ) [6,7]. In PVQ neural predictor is responsible for vector quantization and the codebook is fulfilled by the DPCM function. The successive input vectors $V(t)$ are the macroblocks of the same dimensionality obtained from a frame, where t are indices of consecutive macroblocks. The predictor's input is a preceding macroblock, which after processing results in predicted vector $\bar{V}(t)$. The difference between current macroblock and predictor output $E(t) = V(t) - \bar{V}(t)$ is then calculated and used to select the best approximation g_j using neural quantizer from the codebook $G = [g_0, g_1, \dots, g_J]$. Approximation g_j is then added to the difference which gives reconstructed input vector $\tilde{V}(t) = \bar{V}(t) + g_j$. This vector is later used as the next input to the predictor. The codebook index j is stored in a stream. In order to decompress data, the predictor is again used to calculate predicted vectors while stored codebook indices are used to select their corresponding approximations from codebook that are then combined with predictor outputs to reconstruct the original image.

2.2 Encoding Color Information

Most of modern hardware uses RGB color model to display images and video. Unfortunately, this color model is not an efficient way of storing and processing color information. Common standards for image and video compression such as JPEG and MPEG, and color encoding systems like PAL and NTSC take advantage of the way human vision works. Human eyes are very sensitive to small changes in brightness

but they are far less sensitive to changes in chrominance [15]. For this reason we can use YC_bC_r color space. In this color space, Y is the luminance channel which describes brightness and C_b and C_r channels are chrominances which contain the remaining color information. In our method we used a modified conversion to luminance and chrominances which covers full range of values from 0 to 255 like RGB color values do. Conversion from RGB color space to YC_bC_r color space is presented in Eq. 1 while conversion from YC_bC_r space to RGB color space is presented in Eq. 2 [9, 13].

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{1}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.000 & 0.000 & 1.400 \\ 1.000 & -0.343 & -0.711 \\ 1.000 & 1.765 & 0.000 \end{bmatrix} * \begin{bmatrix} Y \\ C_b - 128 \\ C_r - 128 \end{bmatrix} \tag{2}$$

Since human eye is less sensitive to changes in chrominance, we can reduce an amount of information needed to store C_b and C_r channels without significant loss of quality. This operation is known as chroma subsampling and is used in modern lossy compression algorithms [10]. Commonly used variants, such as 4:2:2 and 4:2:0, are presented in Fig. 1.

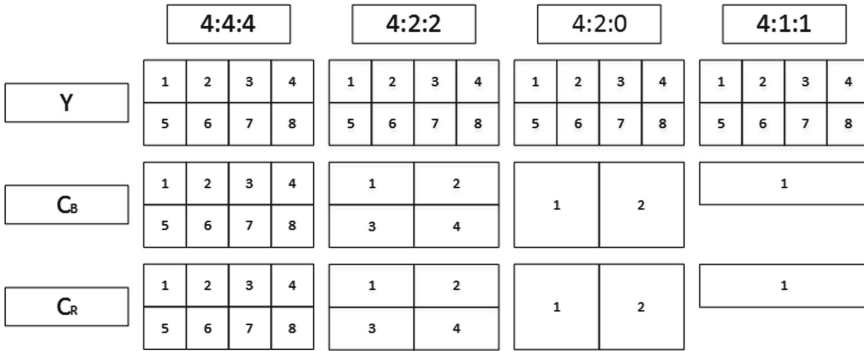


Fig. 1. Chroma subsampling

3 Proposed Method

In this paper, we propose a method of video compression based on the solutions presented in [3, 8, 11–13]. Compared to the previous methods based on the full color chrominance (4:4:4), in our method we use 4:2:0 chrominance in order to reduce the amount of data needed to record frames [15]. In this approach, only the Y channel is stored completely, while the C_b and C_r channels are reduced to half their original resolution. Moreover, C_b and C_r channels are combined into one channel before compression. This approach allowed us to reduce the number

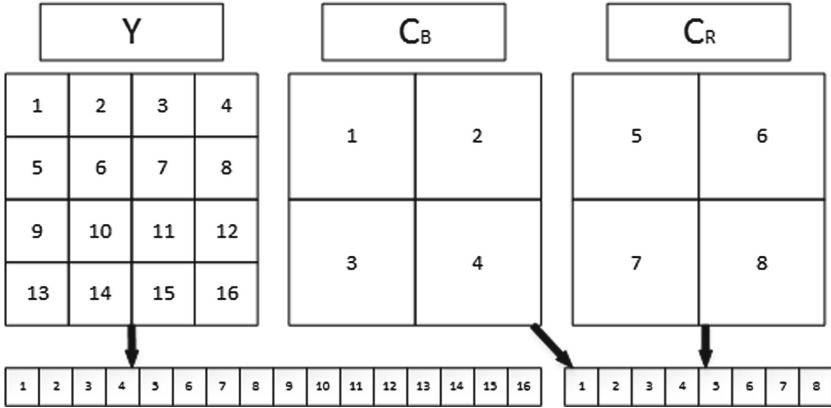


Fig. 2. Scheme encoding YC_bC_r to two channels

of sets of codebooks and predictors to two, one for the luminance channel the other one for the combined chrominance channels (Fig. 2).

Unlike the previous method that uses key frame detection algorithms, this method is based on a fixed size GOP (group of pictures) [14,17]. Each GOP consists of a key frame (intra-frame) and subsequent predictive frames (inter-frame). Each key frame is composed of complete frame data that encode entire frame, and two predictors and codebooks which are used to decode the frame. Predictive frames do not contain predictors and codebooks which are taken from the previously encountered key frame, and instead of encoding the entire frame, they can encode smaller regions within the frame, taking the remaining information from previous frame. This allows us to significantly reduce the size of each predictive frame by storing only changes between subsequent frames (Fig. 3).

4 Experimental Results

For the purpose of our research, we conducted two experiments to test the efficiency of our algorithm. We used sequences of frames from a publicly available video “*Elephants dream*” in resolution 1280 × 720 [16]. In all our tests we converted the frames into YC_bC_r color space. In 4:4:4 chroma subsampling, frames were encoded using three sets of predictors and codebooks. This corresponds to the method used in our previous works. In 4:2:0 chroma subsampling, C_b and C_r channels were first reduced by averaging 2 × 2 squares and then combined into one 8-element channel. Due to this modification algorithm only needed one standard set of predictor and codebook for Y channel and one reduced set of predictor and codebook for combined C_bC_r channel. This allowed us to reduce the number of indices stored per macroblock from 3 to 2 and reduced computational complexity of the compression.

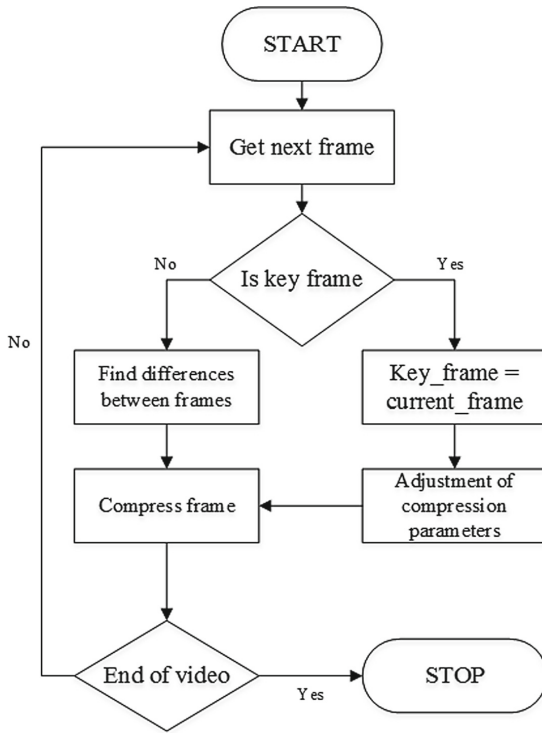


Fig. 3. Video compression algorithm

In the first experiment, we tested the efficiency of our algorithm in two types of chroma subsampling modes, 4:4:4 and 4:2:0. In both cases we used preset macroblock size 4×4 . The experiment showed that introduction of chroma subsampling did not significantly degrade the video quality which can be seen in Fig. 4.

We compared frames compressed using 4:4:4 chroma subsampling to frames compressed using 4:2:0 subsampling using Peak Signal-to-Noise Ratio (PSNR). Figure 5 shows that both methods result in very similar PSNR values.

In the second experiment we compared the video quality after compression using predictive frames. Instead of key detection algorithm we used fixed GOP with a length of 30 frames. The experiment showed that there is no significant loss of quality in the example that uses predictive frames instead of encoding full frame what can be seen in Fig. 6. The quality was again compared with the version without predictive frames using PSNR (Fig. 7). Introduction of 4:2:0 subsampling and predictive frames resulted in about 48% reduction of compressed data, which means considerably increased compression ratio.



Fig. 4. Differences between images before compression, after compression in 4:4:4 and 4:2:0 subsampling

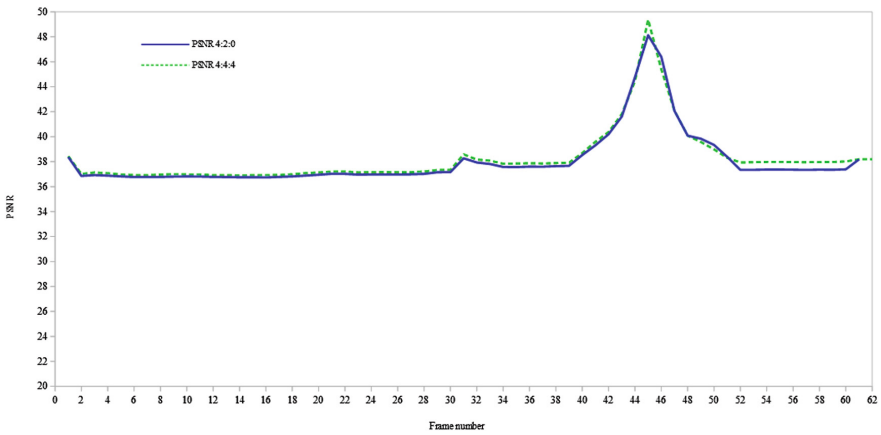


Fig. 5. PSNR of compressed video frames in 4:4:4 and 4:2:0 subsampling



Fig. 6. Differences between images before compression, after compression with and without inter-frame

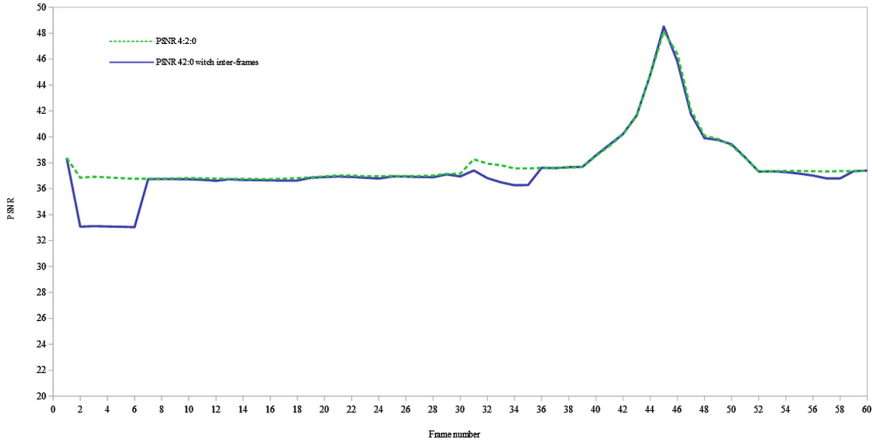


Fig. 7. PSNR of compressed video frames in 4:2:0 and 4:2:0 with inter-frame

5 Conclusions and Future Work

In this paper we showed that the presented method can be successfully used in video compression. The combination of 4:2:0 color subsampling and introduction of predictive frames allowed us to significantly reduce the size of compressed video with respect to previous approach. Thanks to encoding of YC_bC_r channels into two we were able to reduce the number of predictors and codebooks to two from original three. Predictive frames made it possible to eliminate unnecessary compression of regions that didn't change between consecutive frames, further reducing the size of compressed data. Those changes also resulted in reduced processing time.

Our future work will concentrate on improvements of the algorithm, especially optimizing the creation of predictive frames and enhancing the quality of video. In our research, we want to find and compare various schemes for encoding reduced YC_bC_r channels to see which one of them can improve quality and which one can improve compression ratio. Finally we will try to compare our results with known compression standards, such as MPEG.

References

1. CCITT: Video codec for audio visual services at px 64 kbits/s (1993)
2. Cierniak, R.: An image compression algorithm based on neural networks. In: Rutkowski, L., Siekmann, J.H., Tadeusiewicz, R., Zadeh, L.A. (eds.) ICAISC 2004. LNCS, vol. 3070, pp. 706–711. Springer, Heidelberg (2004). doi:[10.1007/978-3-540-24844-6_108](https://doi.org/10.1007/978-3-540-24844-6_108)
3. Cierniak, R., Knop, M.: Video compression algorithm based on neural networks. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2013. LNCS, vol. 7894, pp. 524–531. Springer, Heidelberg (2013). doi:[10.1007/978-3-642-38658-9_47](https://doi.org/10.1007/978-3-642-38658-9_47)

4. Cierniak, R., Rutkowski, L.: On image compression by competitive neural networks and optimal linear predictors. *Signal Process. Image Commun.* **15**(6), 559–565 (2000)
5. Clarke, R.J.: *Digital Compression of Still Images and Video*. Academic Press Inc., London (1995)
6. Gersho, A., Gray, R.M.: *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, Norwell (1991)
7. Gray, R.: Vector quantization. *IEEE ASSP Magaz.* **1**(2), 4–29 (1984)
8. Grycuk, R., Knop, M.: Neural video compression based on SURF scene change detection algorithm. In: Choraś, R.S. (ed.) *Image Processing and Communications Challenges 7*. AISC, vol. 389, pp. 105–112. Springer, Cham (2016). doi:[10.1007/978-3-319-23814-2_13](https://doi.org/10.1007/978-3-319-23814-2_13)
9. ITU-R BT.709–6: Parameter values for the HDTV standards for production and international programme exchange (2015)
10. Kerr, D.A.: Chrominance subsampling in digital images. *The Pumpkin* (3), January 2012
11. Knop, M., Cierniak, R., Shah, N.: Video compression algorithm based on neural network structures. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) *ICAISC 2014*. LNCS, vol. 8467, pp. 715–724. Springer, Cham (2014). doi:[10.1007/978-3-319-07173-2_61](https://doi.org/10.1007/978-3-319-07173-2_61)
12. Knop, M., Dobosz, P.: Neural Video Compression Algorithm. In: Choraś, R.S. (ed.) *ICAISC 2014*. AISC, vol. 313, pp. 59–66. Springer, Cham (2015). doi:[10.1007/978-3-319-10662-5_8](https://doi.org/10.1007/978-3-319-10662-5_8)
13. Knop, M., Kapuściński, T., Mleczo, W.K., Angryk, R.: Neural video compression based on RBM scene change detection algorithm. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) *ICAISC 2016*. LNCS (LNAI), vol. 9693, pp. 660–669. Springer, Cham (2016). doi:[10.1007/978-3-319-39384-1_58](https://doi.org/10.1007/978-3-319-39384-1_58)
14. Setton, E., Girod, B.: Video streaming with SP and SI frames. In: *Proceedings of Visual Communication and Image Processing* (2005)
15. Winkler, S., Kunt, M., van den Branden Lambrecht, C.J.: Vision and video: models and applications. In: van den Branden Lambrecht, C.J. (ed.) *Vision Models and Applications to Image and Video Processing*, pp. 201–229. Springer, Heidelberg (2001)
16. Xiph.org: Video test media. <https://media.xiph.org/video/derf/> Accessed 10 Mar 2016
17. Zhang, R., Regunathan, S.L., Rose, K.: Video coding with optimal inter/intra-mode switching for packet loss resilience. *IEEE J. Sel. Areas Commun.* **18**(6), 966–976 (2000)