A NEW APPROACH TO COLOR VIDEO CODING USING SPATIO-TEMPORAL CORRELATION OF PRIMARY COLORS

Rudi Primorac and Moshe Porat

Department of Electrical Engineering, Technion-Israel Institute of Technology
Haifa 32000, Israel
phone: +972-4-8294684, fax: +972-4-8295757, rudiprim@tx.technion.ac.il, mp@ee.technion.ac.il
web: vision.technion.ac.il/mp

ABSTRACT

Most video compression techniques do not operate directly in the highly-correlated RGB space. Instead, they transform the primary colors into a decorrelated color space, such as YCbCr or YIQ, where the compression takes place. In this paper we propose a different approach. The video stream is sub-divided into 3D blocks (x, y, t), of high correlation between the color components. In each block, we encode two of the color components as polynomials of the third component (defined as 'base color'). The base color is then compressed using conventional tools, while the two remaining colors are transmitted to the decoder by their polynomial coefficients. Experimental results show that the proposed approach outperforms presently available methods. Our conclusion is that the high correlation between primary RGB colors could be helpful for video coding and that the new spatio-temporal approach to video compression is more efficient than conventional decorrelation-based techniques.

1. INTRODUCTION

Natural images and video sequences, given in the RGB color space, are characterized by high correlation between their primary colors (R, G and B). In order to reduce this redundancy, most compression algorithms, like JPEG and MPEG [1], decorrelate the color components by transforming the input image or video from the highly correlated RGB color space into a less correlated space, such as YCbCr. Since the human visual system is less sensitive to changes in chrominance (Cb, Cr) than to changes in luminance (Y), the chrominance components are then down-sampled and processed/encoded at lower rates.

Unlike this common approach, several algorithms for still image compression were proposed recently [2], [3], [4]. These algorithms utilize the correlation between the primary colors instead of decorrelating them. To some extent, the correlation between the primary colors was also used in high-quality video coding [5]-[9], however, limited to the use of high correlation between residual signals.

In this paper, we directly extend the correlation-based approach to still image compression [2], [3], [4] into video coding by generalizing the spatial cases (*x,y*) into a spatio-temporal (*x, y, t*) approach.

2. THE PROPOSED ALGORITHM

Naturally, a video signal can be considered as a 3D signal, with two spatial dimensions (*x, y*) and one temporal (*t*). According to our approach, since the RGB components are likely to be highly correlated in a small 3D block, two subordinate colors can be approximated as a linear function of the third color (base-color), i.e., in the case of 1st order polynomials, $s_k=a_1b_k+a_0$, where s denotes a subordinate color and b denotes the base color. Optimal approximation coefficients a_1 and a_0 can be calculated using basic numerical analysis,

$$a_1=\frac{Cov(s,b)}{Var(b)}$$
$$a_0 = E(s)-\frac{Cov(s,b)}{Var(b)}E(b)\,. \tag{1}$$

The minimum mean-square approximation error (MSE) could be then calculated,

$$\mathrm{MSE} = (1-\rho^2_{bs})\mathrm{Var}(s), \tag{2}$$

where ρ_{bs} is the Pearson correlation coefficient between the base color and the subordinate color.

From (2) it can be seen that the mean approximation error decreases as Pearson correlation coefficient between the base and the subordinate colors increases and the variance of the subordinate colors decreases. Higher correlation between the colors is generally observed in smaller blocks. Naturally, the colors variance decreases as the block size decreases. Hence, as expected, the MSE is lower in smaller blocks.

In our proposed algorithm, the base color in each block is chosen using the following method [4]:

1. Set one component as a base color, for example, R.
2. Approximate the remaining two colors as linear functions of the chosen base color, and calculate approximation errors using (2). Calculate the total error as the sum of the approximation errors for each subordinate color

3. Repeat Steps 1 and 2 for the G and the B components
4. Choose the component with the lowest total error as the base color for the block

Using blocks of fixed size is not optimal though. There are areas in the video stream where no motion occurs (mainly the background). In these areas, the size of the block along the temporal axis can be larger. On the other hand, in intervals of significant motion or changes in the spatial domain, the size of the block should be smaller. This is in accordance with (2), since it lowers the color's variance, which, in turn, decreases the approximation error.

Figure 1 shows schematically the process of block processing. The video stream is initially divided into blocks of fixed size, *N*x*N*x*K*. In each block, we choose the base color as described. The blocks are then processed: for each block and each subordinate color, a decision is made if further subdivision of the block is necessary. The decision is based on the estimation how accurately the subordinate colors in the block can be approximated as a function of the base color. Accurate approximation can be obtained by calculating optimal approximation coefficients using (2), and then calculating a mean approximation error. If the error is smaller than a defined threshold, the approximation coefficients are transmitted and the next block is processed. Otherwise, we divide the block into two sub-blocks of the same size (3 possible cases – division in *x, y* and *t* axis). If any of these sub-divisions yields an approximation error lower than the threshold, the best division is taken and the next block is processed. Otherwise the original block is subdivided into 8 sub-blocks of equal sizes, and the process is repeated for each sub-block.

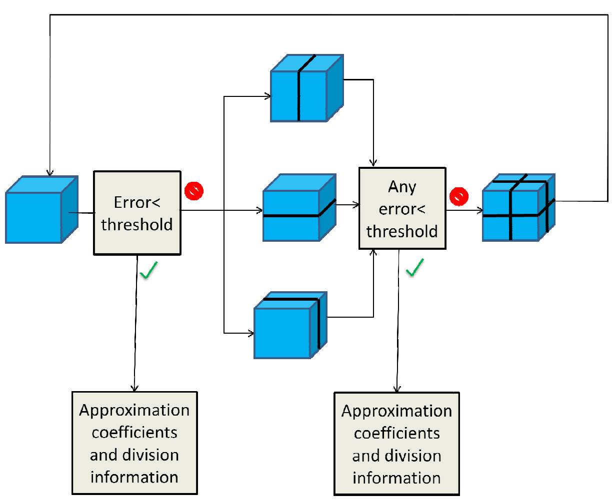


Figure 1- Steps of processing a single 3D block

To guarantee correct decoding, side information is needed, as follows:

- 0 indicates that a block is not subdivided
- 1 indicates that a block is divided into 8 equal sub-blocks
- 2 indicates that a block is cut in half in *x* axis
- 3 indicates that a block is cut in half in *y* axis
- 4 indicates that a block is cut in half in *t* axis

This side information, along with the information about the initial size of the blocks, is sufficient for the decoder to reconstruct the block sequence.

The steps performed by the encoder are summarized schematically in Figure 2, as follows.

- The video stream is divided into *N*x*N*x*K* blocks
- In each block, a base color is chosen
- The base color is compressed using a conventional compression algorithm, such as MPEG
- For each block and each subordinate color:
  1. Calculate optimal coefficients for the linear approximation of the subordinate color by Equation (1) and approximation error using Equation (2)
  2. If the approximation error is above the threshold, sub-divide the block into 2 or 8 sub-blocks of equal sizes, providing sub-division information (i.e., 1, 2, 3 or 4). For each sub-block, repeat the process. Otherwise, transmit the approximation coefficients with a division index of 0 and proceed to the next *N*x*N*x*K* block
- Quantize the approximation coefficients, using uniform quantization
- Entropy code (Huffman) the approximation coefficients
- Entropy code (Huffman) the side information regarding block subdivision (an array of 0,1,2,3,4)

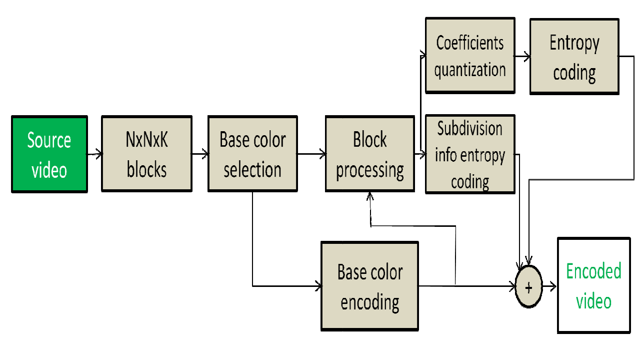


Figure 2 – Block diagram of the encoder

The decoder uses a reverse scheme, shown schematically in Figure 3, as follows.

- Reconstruct the base color component using a conventional decoder, as used in the encoding stage.
- Entropy decode the approximation coefficients
- Reconstruct the approximation coefficients
- Entropy decode the blocks subdividing information
- Using previously decoded subdividing information, reconstruct each block accordingly
- For each block, reconstruct subordinate colors as a linear function of the base color, using previously decoded approximation coefficients

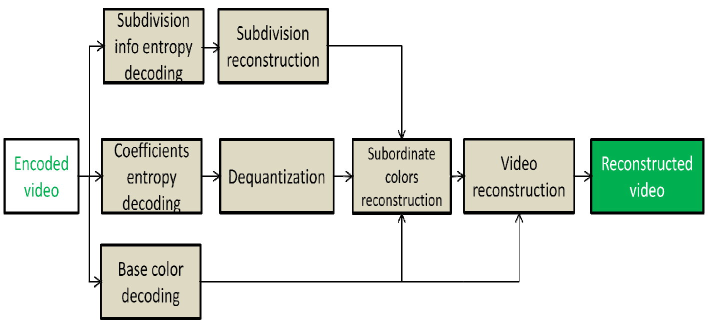
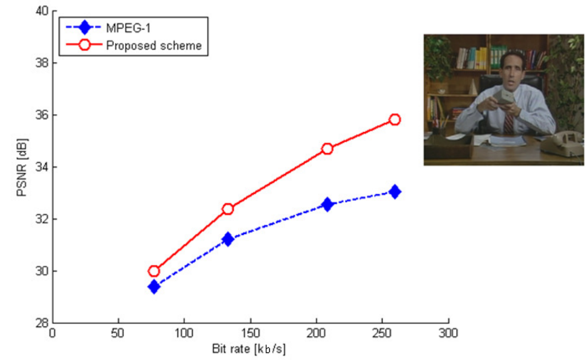


Figure 3 – Block diagram of the decoder

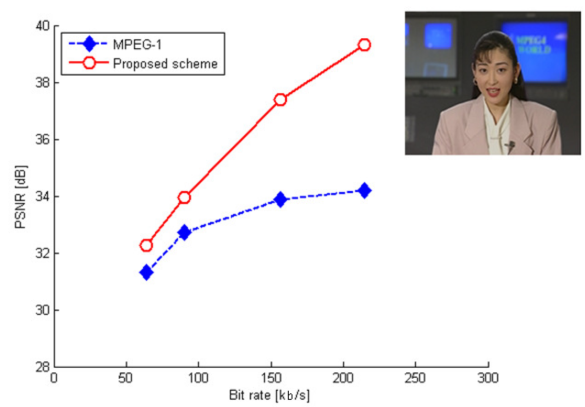
3. EXPERIMENTAL RESULTS

Several sequences of color video were tested, with the base color component compressed using MPEG-1. The two subordinate colors were encoded using the proposed approach. We then compared the performance of the proposed algorithm for the entire color sequence with that obtained by MPEG-1 applied to the complete color sequence. As mentioned, however, our algorithm is general, though, and the base-color compression can be based on any other video compression method for color video, such as H.264/AVC [10] etc. We used the UC Berkeley MPEG encoder to obtain MPEG-1 files. The GOP (group of pictures) used was with N=15 and M=3, i.e., IBBPBBPBBPBBPBB. All the sequences were QCIF-sized (176x144 pels), 30 frames/sec. The initial block size in all the experiments was 8x8x64.

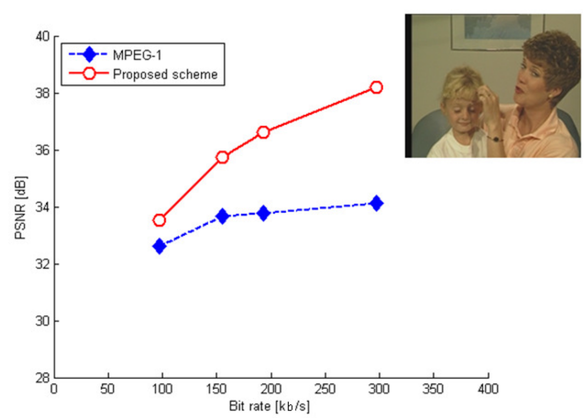
Figure 4 (a), (b), (c), shows PSNR results (in dB) as a function of bit-rate (in kB/s) for the 'Salesman', 'Akiyo' and the 'Mother & daughter' sequences, respectively. Figure 5 (a), (b), (c) shows a frame from the original sequence, the frame reconstructed by MPEG, and by our approach. Figure 6 (a), (b), (c) shows PSNR (in dB) as a function of the frame number for each of the tested sequences reconstructed by both MPEG and by our approach.



(a) Salesman

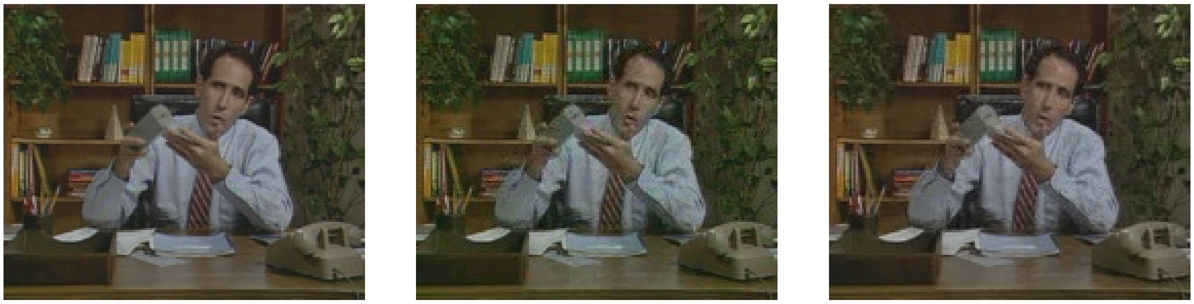


(b) Akiyo



(c) Mother & daughter

Figure 4 - Coding performance. Shown are the PSNR results vs. the bit-rate for three typical video sequences: Salesman, Akiyo and Mother & daughter. As can be seen, the proposed algorithm (red) outperforms MPEG (blue).



(a) Salesman

(b) Akiyo

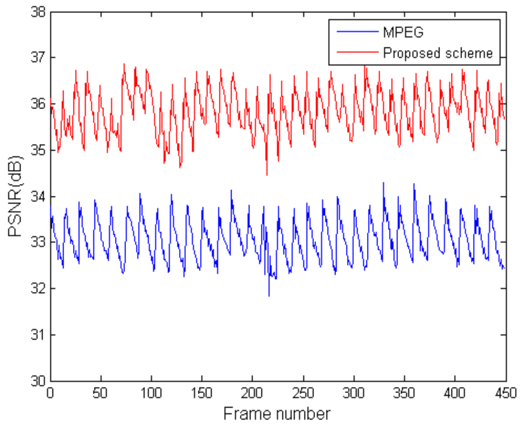
(c) Mother & daughter

Figure 5 – Visual comparison of the new algorithm with MPEG. Shown are the original frame (left), the frame reconstructed by MPEG (center) and by the new method (right). Although the differences between the images are hardly visible *spatially*, the changes in color could be more easily observed, e.g., the color of the telephone and the vegetation above it in 'Salesman', the face and jacket color in 'Akiyo', and the hair color in 'Mother & daughter'.

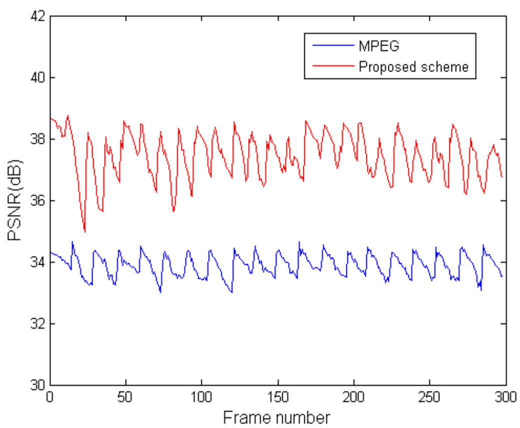
4. SUMMARY AND CONCLUSIONS

We have presented a new approach to color video coding. Instead of transforming the input video from RGB into a decorrelated space such as YCbCr or YIQ, our algorithm subdivides the video sequence into 3D blocks, whereas in each block the correlation is actually used for efficient coding [11], [12]. Only one base-color is video coded, while the additional two colors are represented as polynomials of the base color. Our experiments show a quantitative advantage of our method compared to MPEG in terms of PSNR, and in most cases in visual appearance as well, where the emphasis is on preserving the color information better than MPEG. As for the complexity, the proposed algorithm could be more complex than MPEG in case of many small 3D blocks, however, since the number of possible subdivisions of a 3D block is limited, the overall computational complexity of subordinate color's coding is only a linear function of the number of pixels. Our conclusion is that using the high correlation between primary colors in RGB color spaces could be more efficient than conventional decorrelation-based compression methods.

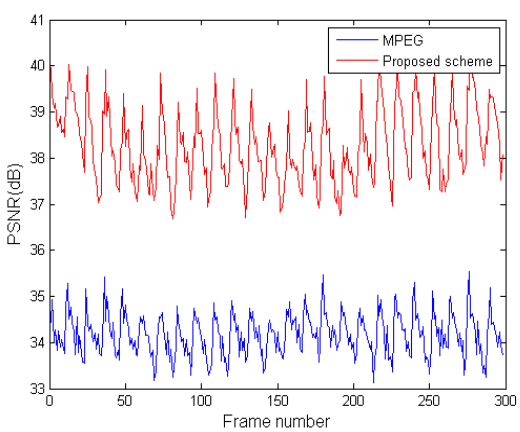
ACKNOWLEDGMENTS This work was supported in part by the Ollendorff Minerva Centre. Minerva is funded through the BMBF.



(a) Salesman



(b) Akiyo



(c) Mother & daughter

Figure 6 – PSNR as a function of frame number for MPEG (blue) and the new algorithm (red).

5. REFERENCES

[1] ISO/IEC 11172-2, "Information technology — Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1,5 Mbit/s – Part 2: Video," 1993.

[2] H. Kotera, and H. Kanamori, "A Novel Coding Algorithm for Representing Full color image by a Single Color Image", J. Imaging Technology, Vol. 16, Aug. 1990, pp. 142-152.

[3] L. Goffman-Vinopal, and M. Porat, "Color image compression using inter-color correlation," in Proc. IEEE International Conference on Image Processing, ICIP, vol. 2, Sep. 2002, pp. II-353–II-356.

[4] E. Gershikov, E. Lavi-Burlak, and M. Porat, "Correlation-based approach to color image compression", Signal Processing Image Communication., vol. 22, no. 9, Oct. 2007, pp. 719–733.

[5] W. Kim, D. Cho and H. Kim, "Inter-plane prediction for RGB video coding", in Proc. IEEE International Conference on Image Processing, ICIP, Vol. 2, Oct. 2004, pp. 785–788.

[6] D. Marpe, H. Kirchhoffer, V. George, P. Kauff, and T. Wiegand, "Macroblock-adaptive residual color space transforms for 4:4:4 video coding," in Proc. IEEE International Conference on Image Processing, ICIP, pp. 3157–3160, Oct. 2006.

[7] Y.-H. Kim, J.-H. Park, J.-W. Kim, B. Choi, and J. Paik, "Efficient RGB video coding using adaptive inter-plane residual prediction," in Proc. IEEE ICCE, Jan. 2008, pp. 1–2.

[8] Yong-Hwan Kim, Byeongho Choi, and Joonki Paik, "High-Fidelity RGB Video Coding Using Adaptive Inter-Plane Weighted Prediction" IEEE Trans. on Circuits and Systems for Video Technology, Vol. 19, No. 7, July 2009, pp. 1051-1056.

[9] Kim YS., Jeong J., Choe Y., "RGB video coding using the matched neighborhood block in the inter-color plane" IEICE Electronics Express, Vol. 6, Dec. 2009, pp. 1715-1720.

[10] ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC, "Advanced Video Coding for Generic Audiovisual Services," 2003.

[11] S. Nemirovsky and M. Porat, "On Texture and Image Interpolation using Markov Models", Signal Processing: Image Communication, Vol. 24, pp. 139-157, 2009.

[12] M. Porat and Y.Y. Zeevi, "The generalized Gabor scheme of image representation in biological and machine vision", IEEE Trans.on Pattern Analysis and Machine Intelligence (PAMI) 10:44, pp. 452-468, 1988.