A Very Low Bit-Rate Minimalist Video Encoder Based on Matching Pursuits

Vitor de Lima and Helio Pedrini

Institute of Computing - University of Campinas
Campinas, SP, Brazil, 13084-971

Abstract. This work proposes and implements a simple and efficient video encoder based on the compression of consecutive frame differences using sparse decomposition through matching pursuits. Despite its minimalist design, the proposed video codec has performance compatible to H.263 video standard and, unlike other encoders based on similar techniques, is capable of encoding videos in real time. Average PSNR and image quality consistency are compared to H.263 using a set of video sequences.

1 Introduction

Video compression at very low bit-rates is needed for applications that operate using low bandwidth communication channels, for instance, video transmission in mobile equipments. Some techniques that have been suggested for such applications include hybrid-DCT coding [6], wavelet-based coding [20], model-based coding [2], and fractal coding [11].

Extreme compression rates demanded by low bit-rate video applications require unusual video encoding techniques. One possible approach is the matching-pursuit video coding, however, it involves a very time-consuming encoding process [15] due to its exhaustive image scan in order to find patterns that can be represented efficiently.

The approach proposed in this paper is extremely simple and capable of compressing video sequences in real time. The video encoder compresses only the difference between two consecutive frames through matching pursuits. No motion compensation algorithm [9] is used in the process and the quantization is performed by rounding the coefficients to the nearest integer. An innovation of the proposed method is the subdivision of the frame into blocks and application of matching pursuit to each block instead of scanning the entire image looking for regions that have relevant characteristics that can be compressed and then applying matching pursuits to those regions.

A dictionary generated by K-SVD algorithm [1] is used to create sparse decompositions of the processed frame sub-blocks, which are compressed by a context-adaptive arithmetic encoder [18].

Compared to H.263 video codec [10], which has a motion compensation algorithm, more sophisticated quantizers and mechanisms for rate-distortion

I. Bloch and R.M. Cesar, Jr. (Eds.): CIARP 2010, LNCS 6419, pp. 176–183, 2010.
© Springer-Verlag Berlin Heidelberg 2010

optimization, the proposed method achieves compatible PSNR values, as demonstrated in the experiments using well-known benchmark video sequences at several average bit rates per second.

The text is organized as follows. Section 2 describes the main algorithms used in the proposed solution, as well as reviews of some relevant encoders based on matching pursuits found in literature. Details of the proposed methodology are presented and discussed in Section 3. Experimental results obtained with our video codec are shown in Section 4. Finally, conclusions of the work and future directions are presented in Section 5.

2 Related Work

This section briefly describes some relevant concepts and techniques related to the proposed video encoder.

2.1 Matching Pursuits

Transforms, such as DCT [5], decompose signals as a linear combination of mutually orthogonal elements belonging to a predetermined basis. This basis contains a minimum number of elements sufficient to express any vector belonging to a particular vector space.

A possible generalization for such type of transform involves using more than the minimum required number of elements within the basis, thus forming an overcomplete dictionary, In this case, a single vector has several possible decompositions and, for data compression purpose, the most interesting decompositions are those that have the largest possible number of linear coefficients equal to zero.

Finding such decompositions is a NP-hard problem [7], so that matching pursuits [12] is a greedy heuristic for finding a very sparse decomposition of a signal using low processing time. Given an overcomplete dictionary $D = \{g_\gamma\}_{\gamma \in \Gamma}$, a signal f to be decomposed and a threshold of the decomposition error ϵ, Algorithm 1 determines which elements of D and linear coefficients are used in a sparse decomposition of f. Term R^k is the signal residue not yet represented by the chosen bases until step k.

Algorithm 1. Matching pursuit algorithm.

$R^0 f = f$
$n = 0$
repeat
  $i = \arg\max_{k \in \Gamma} \langle R^n f, g_k \rangle$
  $R^{n+1} = R^n f - \langle R^n f, g_i \rangle g_i$
  $n = n + 1$
until $n < n_{max}$ OR $|R^{n+1} f| < \epsilon$

2.2 Optimized Orthogonal Matching Pursuits

A more powerful heuristic for searching for sparse signal representations using overcomplete dictionaries was employed in the proposed video codec, known as optimized orthogonal matching pursuit [17].

At each step of the encoding process, after choosing an element g_i of the dictionary by the same criterion of the conventional matching pursuit, such search heuristic orthogonalizes the entire dictionary with respect to g_i. Therefore, the chosen element in the following step is orthogonal to all elements used previously. The heuristic ensures more sparse representations at a higher computational cost.

2.3 K-SVD

A well generated overcomplete dictionary ensures more sparse decompositions, provides a higher convergence speed in matching pursuits, is capable of representing only psychovisually significant features and ignores minor irrelevant details. It is possible to develop such dictionaries through machine learning algorithms [19], among them the K-SVD, which is a generalization of the algorithm for solving the K-means problem.

Two alternating steps are performed during its execution. In the first step, data from the training set is decomposed according to the initial overcomplete dictionary to be optimized using any algorithm capable of doing it. In the second step, each element of the dictionary is replaced by a new one, calculated to minimize the error of each data from the training set that used it in its sparse decomposition, as described in Algorithm 2.

Algorithm 2. K-SVD algorithm.

Input: initial set $Y = \{y_i\}_{i=1}^{N}$ of training signals, an initial dictionary D with normalized columns, a target sparsity T and the total number of iterations k.
Output: an approximate solution to $min_{D,X}\{||Y - DX||_F^2\}$ subject to $\forall i, ||x_i||_0 \leq T$ and $\forall j, ||D_j||_2 = 1$.
for $n = 1$ to k **do**
  $\forall i, x_i = \arg\min_\gamma\{||y_i - D\gamma||_2^2\}$ subject to $||\gamma||_0 \leq T$
  for each column j in D **do**
    $D_j = 0$
    $I = \{$indices of the signals in Y whose decompositions use $D_j\}$
    $E = Y_I - DX_I$
    $\{d, g\} = \arg\min_{d,g} ||E - dg^T||_F^2$ subject to $||d||_2 = 1$
    $D_j = d$
    $X_{j,I} = g^T$
  end for
end for

2.4 Matching Pursuit Video Coding

The absolute majority of video codecs based on matching pursuits [3,14,21,22] have their origins in [15]. The method uses an inner-product search to decompose motion residual signals over an overcomplete dictionary of 2D separable Gabor functions.

Despite the high computational cost of such search, the approach avoids artificial block edges and presents both better perceptual image quality and higher PSNR than DCT-based methods for low bit rates video coding. However, the dictionary must be efficiently built to allow fast inner-product computation between its elements and various regions of the residue.

The proposed encoder avoids performing costly searches working similarly to DCT-based coders, where the difference between two consecutive frames is partitioned into non-overlapping blocks that are independently coded using matching pursuits. This allows encoding parallelization of the sub-blocks, however, it does not prevent artifact appearance at the intra-block edges.

3 Proposed Video Codec

Initially, the encoder calculates the difference between the frame to be processed and the previous uncompressed frame. If the norm of this subtraction is greater than a certain threshold, the entire frame is used in the next step, otherwise only the difference between these two frames is used.

The image generated in the previous step is then subdivided into blocks of 8×8 pixels without overlapping. Each block is decomposed as a sparse linear combination of the dictionary elements through the Optimized Orthogonal Matching Pursuit algorithm [17]. The average bit rate is controlled by manually varying the error threshold ϵ used in the algorithm.

The overcomplete dictionary used in our encoding method is the same used by Elad and Aharon [8] for image denoising. The learned dictionary contains 256 elements and was trained using K-SVD algorithm using a number of several photographs as a training set.

In the final step, a flag is coded to indicate whether what is being transmitted is only the difference between two consecutive frames or an entire frame.

For each block of the current frame decomposed in the previous step, its sparse representations are transmitted through an arithmetic encoder using four distinct symbols, each one containing its proper adaptive context. The first symbol indicates the number of elements of the dictionary used in the decomposition of that block. For each used element, sign and magnitude of the linear coefficient associated with that element and its index are transmitted in different symbols.

4 Experimental Results

The proposed video codec was implemented on a graphics processing unit (GPU) with CUDA [16]. Our codec was compared to the implementation of the H.263 video standard present in the open-source `libavcodec` library [4].

Several video sequences were used in the experiments [13]. Results for three video samples are reported in this work. The videos have resolution of 176×144 pixels and 10 frames per second with subsampled chrominance (format 4:2:2).

All videos were compressed both with our codec and H.263 at different average rates of kilobits per second. The comparison was based on peak signal-to-noise ratio (PSNR) value, expressed by

$$\mathrm{PSNR} = 10 \log_{10} \left(\frac{255^2}{\mathrm{MSE}} \right) \tag{1}$$

where MSE is the mean squared error between the resulting image after compression and uncompression steps and the original image.

Average PSNR values for all frames and three color channels of the tested video sequences are shown in Table 1.

Table 1. Average PSNR (in decibels) obtained by using the proposed codec (MP) and H.263

	Akiyo		Salesman		Hall Monitor	
kbps	H.263	MP	H.263	MP	H.263	MP
15	30.70	30.71	29.41	28.74	29.54	29.07
20	31.67	31.73	30.01	29.38	30.22	30.18
30	33.60	33.38	31.21	30.55	31.60	32.11
40	35.20	34.66	32.29	31.55	32.88	33.56
50	36.54	35.77	33.18	32.30	34.06	34.80

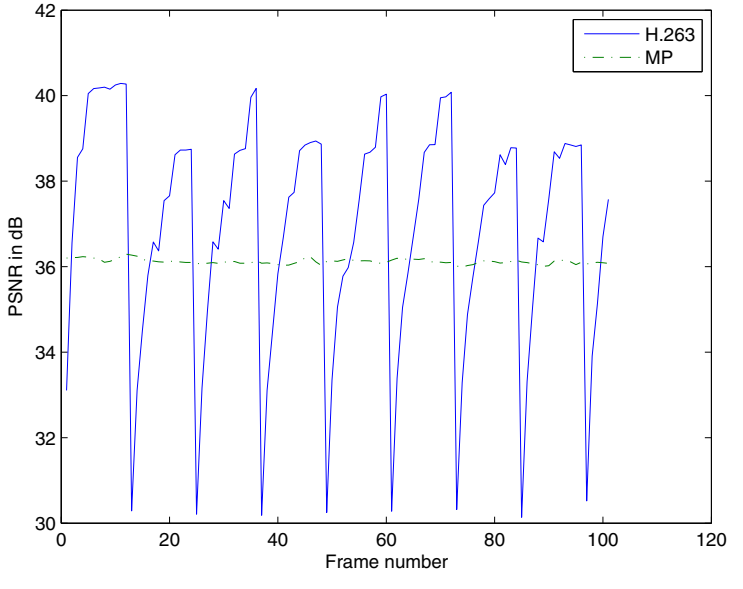

Fig. 1. Comparison of per-frame PSNR values between the proposed encoder and H.263 for Akiyo sequence at 50 kbps

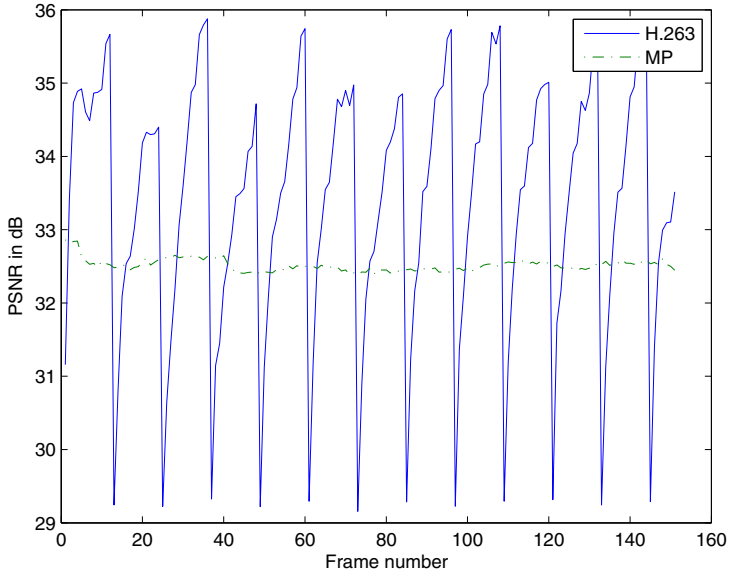

Fig. 2. Comparison of per-frame PSNR values between the proposed encoder and H.263 for Salesman sequence at 50 kbps

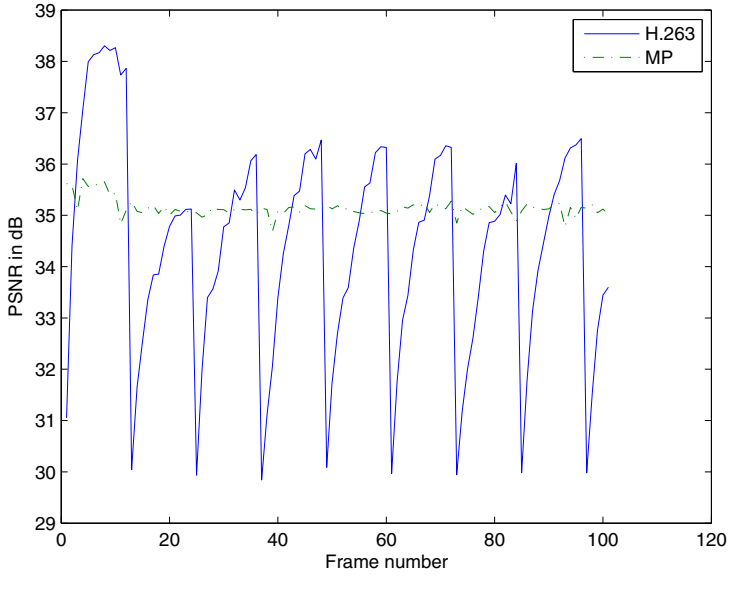

Fig. 3. Comparison of per-frame PSNR values between the proposed encoder and H.263 for Hall monitor sequence at 50 kbps

Despite the extreme simplicity of the proposed approach, its performance is very similar to H.263 video standard. The lack of a motion compensation algorithm prevented effective use of statistical redundancy present in the consecutive video frames.

Another important characteristic of the presented approach is its consistency in the video frame quality. As can be seen in Figures 1, 2 and 3, PSNR value of each frame changed abruptly when compressed by H.263, however, it is kept almost constant by the proposed algorithm. This is mainly due to the rate control mechanism of H.263.

5 Conclusions and Future Work

A video encoder is proposed to compress the difference between two consecutive frames through the matching pursuit approach using a dictionary previously trained by K-SVD method.

Unlike other video codecs based on matching pursuits, the proposed approach is able to encode video in real time and has performance compatible to H.263 when tested for some video sequences used in standard benchmarks.

Future directions for work include the implementation of refined motion compensation methods, a filter for removing blocking artifacts, a better quantization scheme of the sparse decomposition coefficients and other forms of prediction residue coding using both matching pursuits and dictionaries created by K-SVD. Such changes can significantly improve the resulting image quality.

Acknowledgments

The authors are grateful to FAPESP and CNPq for their financial support.

References

1. Aharon, M., Elad, M., Bruckstein, A.: K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. IEEE Transactions on Signal Processing 54(11), 4311–4322 (2006)
2. Aizawa, K., Harashima, H., Saito, T.: Model-Based Analysis Synthesis Image Coding (MBASIC) System for a Person's Face. Signal Processing: Image Communications 1(2), 139–152 (1989)
3. Al-Shaykh, O., Miloslavsky, E., Nomura, T., Neff, R., Zakhor, A.: Video Compression using Matching Pursuits. IEEE Transactions on Circuits and Systems for Video Technology 9(1), 123–143 (1999)
4. avcodec: libavcodec: A Library containing Decoders and Encoders for Audio/Video Codecs (2010), `http://www.ffmpeg.org/`
5. Bhaskaran, V., Konstantinides, K.: Image and Video Compression Standards: Algorithms and Architectures. Kluwer Academic Publishers, Norwell (1997)
6. CCITT: Video Codec for Audiovisual Services at p × 64 kbit/s, CCITT Recommendation H.261, CDM XV-R 37-E (August 1990)
7. Davis, G.: Adaptive Nonlinear Approximations. Ph.D. thesis, Department of Mathematics, New York University (1994)

8. Elad, M., Aharon, M.: Image Denoising Via Sparse and Redundant Representations Over Learned Dictionaries. IEEE Transactions on Image Processing 15(12), 3736–3745 (2006)
9. Furht, B., Furht, B.: Motion Estimation Algorithms for Video Compression. Kluwer Academic Publishers, Norwell (1996)
10. H263: ITU-T Recommendation H.263, Video Coding for Low Bit Rate Communication (September 1997)
11. Jacquin, A.: Image Coding Based on a Fractal Theory of Iterated Contractive Image Transformations. IEEE Transactions on Image Processing 1(1), 18–30 (1992)
12. Mallat, S., Zhang, Z.: Matching Pursuit with Time-Frequency Dictionaries. IEEE Transactions on Signal Processing 41, 3397–3415 (1993)
13. Media, X.T.: Video Sequences (2010), `http://media.xiph.org/video/derf/`
14. Neff, R., Nomura, T., Zakhor, A.: Decoder Complexity and Performance Comparison of Matching Pursuit and DCT-based MPEG-4 Video Codecs. In: International Conference on Image Processing, Chicago, IL, USA, pp. 783–787 (October 1998)
15. Neff, R., Zakhor, A.: Very-Low Bit-Rate Video Coding Based on Matching Pursuits. IEEE Transactions on Circuits and Systems for Video Technology 7(1), 158–171 (1997)
16. NVIDIA: CUDA - Parallel Computing Architecture (2010), `http://www.nvidia.com/`
17. Rebollo-Neira, L., Lowe, D.: Optimized Orthogonal Matching Pursuit Approach. IEEE Signal Processing Letters 9(4), 137–140 (2002)
18. Said, A.: Arithmetic Coding. Communications, Networking, and Multimedia. In: Lossless Compression Handbook. Academic Press, London (2003)
19. Sculley, D., Brodley, C.: Compression and Machine Learning: A New Perspective on Feature Space Vectors. In: Data Compression Conference, Snowbird, UT, USA, pp. 332 (March 2006)
20. Shapiro, J.: Application of the Embedded Wavelet Hierarchical Image Coder to Very Low Bit Rate Image Coding. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, Minneapolis, MN, USA, vol. 5, pp. 558–561 (April 1993)
21. Wang, B., Wang, Y., Yin, P.: A Two Pass H.264-Based Matching Pursuit Video Coder. In: IEEE International Conference on Image Processing, Atlanta, GA, USA, pp. 3149–3152 (October 2006)
22. Zhang, H., Wang, X., Huo, W., Monro, D.: A Hybrid Video Coder Based on H.264 with Matching Pursuits. In: IEEE International Conference on Acoustics, Speech and Signal Processing, Toulouse, France, vol. 2, pp. 889–892 (July 2006)