MATCHING PURSUITS VIDEO CODING USING GENERALIZED BIT-PLANES

Rogério Caetano, Eduardo A. B. da Silva, Alexandre G. Ciancio

PEE/COPPE/DEL/EE/Universidade Federal do Rio de Janeiro
Cx.P. 68504, Rio de Janeiro, RJ, Brazil
caetano@lps.ufrj.br, eduardo@lps.ufrj.br, ciancio@lps.ufrj.br

ABSTRACT

The Matching Pursuits (MP) coding method has been employed successfully in video coding. In it, the motion compensated frame difference is decomposed on an overcomplete dictionary of atoms, generating a sequence of pairs specifying the atoms used and their corresponding coefficients. A particular rate × distortion trade-off is achieved by setting both the number of atoms and the quantizers of their coefficients. Several strategies have been presented in order to set this trade-off. In this paper we propose a novel method for performing matching pursuits quantization, based on the notion of decomposition in generalized bit-planes. Such decompositions generate just a sequence of indexes. They provide an elegant solution to the trade-off between quantization of coefficients and number of passes in the MP algorithm. We show that they can be regarded as generalizations of decompositions followed by linear quantization of the coefficients. In addition, we state a theorem that sets bounds for their R-D performance. We test the effectiveness of the proposed method using the framework of Neff and Zakhor's MP video encoder. The results obtained are promising, presenting, without any ad-hoc assumptions about the R-D behavior of the coded frames or any increase in computational complexity, a significant improvement over the classical MP video coders. Also, the results are as good as the ones obtained employing more sophisticated strategies.

1. INTRODUCTION

The classical algorithms used in video coding are based on the block discrete cosine transform (DCT). An effective alternative for such methods is given by decompositions over redundant dictionaries using the Matching Pursuits (MP) Algorithm [1]. An efficient video encoder using the MP Algorithm has been presented by Neff and Zakhor [2]. It provides good coding efficiency and is free from blocking artifacts. Its success has encouraged research on this topic.

In the MP algorithm we usually decompose a signal $\mathbf{x}$ of dimension N on a redundant dictionary $\mathcal{D} = \{\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_M\}$, $\|\mathbf{g}_i\| = 1$, $\forall i$. The $\mathbf{g}_i$ are in general referred to as atoms. The dictionary is said to be redundant because, in general, $M > N$. The signal $\mathbf{x}$ is then approximated in P passes as [1]

$$\mathbf{x} \approx \sum_{n=1}^{P} p_n \mathbf{g}_{\gamma_n} \qquad (1)$$

This work has been supported in part by CNPq, Brazil, grant nº 522381/95-2 and FAPERJ, Brazil, grant nº 170.331/2000

The index γ_1 corresponds to the atom in the dictionary over which the projection of $\mathbf{x}$ is largest, with p_1 being the value of this projection. We then compute the residual $\mathbf{r}_1 = \mathbf{x} - p_1\mathbf{g}_{\gamma_1}$ and find the pair (p_2, γ_2) corresponding to the largest projection of $\mathbf{r}_1$ on every atom of $\mathcal{D}$. From these we compute $\mathbf{r}_2 = \mathbf{r}_1 - p_2\mathbf{g}_{\gamma_2}$. This process is repeated recursively until we reach the number of passes P that meets a predefined rate and/or distortion criterion. Details of this algorithm can be found in [1]. There, it is shown that the energy of the residuals decrease monotonically as the number of passes P is increased, and tends to zero as P tends to infinity.

From the above, we see that the MP performs a kind of successive approximation of a signal $\mathbf{x}$, since, for each atom added, the error in the approximation decreases. Therefore, in principle, the approximation error can be controlled by the number of atoms used. However, when the coefficients p_n are quantized, the approximation error also depends on how the quantization is performed. Several strategies have been proposed for dealing with this problem in the literature. In [3] the quantizer of the coefficients p_n in each frame is chosen using a two-pass procedure. In the first pass, the frame is decomposed without the coefficients p_n being quantized. Then, the quantizer of the second pass is chosen as 60% of the smallest $|p_n|$ used. Rate×distortion approaches have also been proposed, as the ones in [4], where it was developed operational R-D models for matching pursuits. In [5], an adaptive entropy-constrained quantization scheme is used, based on the fact that the magnitude of the coefficients are bounded by an exponential function of the number of passes.

It is interesting to observe that many of the state of the art image compression methods use successive approximation [6]. They achieve successive approximation by encoding the wavelet transform coefficients in bit-planes. For each added bit-plane, the error in the representation decreases. This is for example the case of the JPEG2000 standard [7]. Taking this into consideration, it is natural to wonder whether it could advantageous to perform the quantization of the MP coefficients in bit-planes. In this paper we propose a novel algorithm to perform an MP-like decomposition in which a signal is decomposed in generalized bit-planes, each bit-plane being composed by a set of atoms. In it, unlike the classical MP, there are no coefficients to be quantized, that is, only the atoms corresponding to each generalized bit-plane need to be transmitted. It provides an elegant solution to the coefficient quantization problem in the MP algorithm, and gives improvements over the existing MP-based encoders. This paper is organized as follows: Section 2 outlines the theory of signal decomposition in generalized bit-planes, that is the base of the proposed algorithm. In section 3 we describe a practical video encoder using generalized bit-planes, with the experimental results described in section 4. Section 5 presents the conclusions.

2. SIGNAL DECOMPOSITION IN GENERALIZED BIT-PLANES

Suppose $\mathbf{x}$ is a signal that can be decomposed in a redundant dictionary $\mathcal{D} = \{\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_M\}$, $\|\mathbf{g}_i\| = 1, \forall i$ as

$$\mathbf{x} = \sum_{n=1}^{M} c_n \mathbf{g}_n \tag{2}$$

Without loss of generality, we are assuming that $\|\mathbf{x}\| \leq 1$. Also, note that we are considering that the dictionary $\mathcal{D}$ is complete, so that an expansion in M terms can represent $\mathbf{x}$ with zero distortion. In addition, since $\|\mathbf{g}_i\| = 1$, there is an expansion in the form of equation (2) such that $|c_n| \leq 1$.

Since $|c_n| \leq 1$, we can write the binary representation for c_n as $c_n = s_n \sum_{j=1}^{\infty} 2^{-j} b_{j,n}$. $s_n \in \{-1, 1\}$ is the sign of c_n, and $b_{j,n} \in \{0, 1\}$. Replacing this value of c_n in equation (2) we have that

$$\begin{aligned} \mathbf{x} &= \sum_{n=1}^{M} s_n \sum_{j=1}^{\infty} 2^{-j} b_{j,n} \mathbf{g}_n = \sum_{j=1}^{\infty} 2^{-j} \sum_{n=1}^{M} b_{j,n} s_n \mathbf{g}_n \\ &= \sum_{j=1}^{\infty} 2^{-j} \sum_{n=1}^{M} b_{j,n} \overline{\mathbf{g}}_n \end{aligned} \tag{3}$$

Note that since $s_n \in \{-1, 1\}$, then $\overline{\mathbf{g}}_n = s_n \mathbf{g}_n \in \overline{\mathcal{D}} = \{\pm\mathbf{g}_1, \pm\mathbf{g}_2, \ldots, \pm\mathbf{g}_M\}$. Now, defining the indexes $i_{j,l}$ such that, for $l \in \{1, 2, \ldots, L_j\}$, $b_{j,i_{j,l}} = 1$, and zero elsewhere, the summation in equation (3) can be expressed as

$$\mathbf{x} = \sum_{j=1}^{\infty} 2^{-j} \sum_{l=1}^{L_j} \overline{\mathbf{g}}_{i_{j,l}} \tag{4}$$

Equation (4) can be regarded as a generalized bit-plane decomposition of the signal $\mathbf{x}$. The bit-plane j is composed by the functions $\overline{\mathbf{g}}_{i_{j,l}}$ for $l = 1, \ldots, L_j$. In [8] a convergent algorithm for finding such decompositions has been proposed, in the same philosophy of the MP algorithm. In fact, the algorithm proposed in [8] finds decompositions of the following form

$$\mathbf{x} = \sum_{j=1}^{\infty} \alpha^j \sum_{l=1}^{L_j} \overline{\mathbf{g}}_{i_{j,l}} \tag{5}$$

These decompositions are more general then the one in equation (4), since the term 2^{-j} has been replaced by α^j, for $0 < \alpha < 1$. In [8] there have been derived conditions for the algorithm to be convergent (that is, for any signal $\mathbf{x}$ be approximated with arbitrary precision by adding a sufficient number of terms to the summations). These conditions impose that $\Theta(\overline{\mathcal{D}}) \leq \frac{\pi}{3}$, where $\Theta(\overline{\mathcal{D}})$ is the largest angle between any signal $\mathbf{x} \in \mathbb{R}^N$ and the closest atom in dictionary $\overline{\mathcal{D}}$. However, even for signals of moderate dimension (e.g., $N \geq 64$), the dictionaries that could provide $\Theta(\overline{\mathcal{D}}) \leq \frac{\pi}{3}$ would have very large cardinality. This would lead to inefficient decompositions from an R-D perspective, since a large number of bits would be needed to encode the indexes $i_{j,l}$.

In this paper we propose a novel algorithm for finding such decompositions, that is convergent whenever $0 < \alpha < 1$ and $\Theta(\overline{\mathcal{D}}) \leq \frac{\pi}{2}$. The advantage of this algorithm is that $\Theta(\overline{\mathcal{D}}) \leq \frac{\pi}{2}$ is only a very mild restriction, being satisfied whenever $\mathcal{D}$ is complete [1]. In this algorithm a greedy decomposition is carried out by adding one $\overline{\mathbf{g}}_{i_{j,l}}$ at a time, until a rate and/or distortion criterion is met. Given a dictionary $\mathcal{C} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_q\}$, $\|\mathbf{v}_i\| = 1, \forall i$, the algorithm is as follows (the input signals are normalized so that $\|\mathbf{x}\| \leq 1$):

Algorithm 1

1. Start with $\mathbf{w} = \mathbf{x}$, $m = 1$.
2. Repeat until a stop criterion is met
   (a) Choose $r_m \in \{1, \ldots, q\}$ such that $\mathbf{w} \cdot \mathbf{v}_{r_m} = \max_{1 \leq j \leq q} \{\mathbf{w} \cdot \mathbf{v}_j\}$.
   (b) Choose $k_m = \left\lceil \frac{\ln(\mathbf{w} \cdot \mathbf{v}_{r_m})}{\ln(\alpha)} \right\rceil$.
   where $\lceil y \rceil$ is the smallest integer larger than or equal to y.
   (c) Replace $\mathbf{w}$ by $\mathbf{w} - \alpha^{k_m} \mathbf{v}_{r_m}$.
   (d) Increment m.
3. Stop.

Note that Algorithm 1 approximates $\mathbf{x}$ in P passes as

$$\mathbf{x}^{(P)} = \sum_{m=1}^{P} \alpha^{k_m} \mathbf{v}_{r_m} \tag{6}$$

If we define L_j as the number of values m such that $k_m = j$, we can rename the corresponding indexes r_m as $i_{j,l}$ for $l = 1, \ldots, L_j$. Therefore, if we make the dictionary $\mathcal{C}$ in Algorithm 1 equal to $\overline{\mathcal{D}}$, then equation (6) is equivalent to equation (5) for $P \to \infty$.

We can say that Algorithm 1 is convergent if $\lim_{P\to\infty} \mathbf{x}^{(P)} = \mathbf{x}$. In that sense, its convergence is guaranteed by Theorem 1.

Theorem 1: Be $\mathbf{x} \in \mathbb{R}^N$, $\|\mathbf{x}\| \leq 1$, such that it is approximated by Algorithm 1 using a dictionary $\mathcal{C}$ with P steps, generating $\mathbf{x}^{(P)}$ as in equation (6), and be $\Theta(\mathcal{C})$ the largest angle between any signal $\mathbf{y} \in \mathbb{R}^N$ and the closest atom in dictionary $\mathcal{C}$. We have that $\|\mathbf{r}^{(P)}\| = \|\mathbf{x} - \mathbf{x}^{(P)}\| \leq \beta_c^{(P)}$, where $\beta_c = \sqrt{1 - (2\alpha - \alpha^2)\cos^2(\Theta(\mathcal{C}))} < 1$ for every $0 < \alpha < 1$ and $0 \leq \Theta(\mathcal{C}) < \frac{\pi}{2}$.

The following points regarding Algorithm 1 should be highlighted:

(i) Algorithm 1 performs a decomposition such that, for every atom added, the distortion in the approximation of $\mathbf{x}$ decreases by at least $\beta < 1$. Thus, when the number of passes $P \to \infty$, $\|\mathbf{r}^{(P)}\| \to 0$, that is, algorithm 1 is convergent.

(ii) The representation output by Algorithm 1 is given by just a sequence of pairs of indexes (k_m, r_m), $m = 1, 2, \ldots, P$. This implies that there is no need for coefficients quantization as in the classical MP algorithm (see equation (2) and the discussion that follows). In other words, it can be said that Algorithm 1 performs both the decomposition and quantization at the same time. Thus, it presents an elegant solution to the coefficient quantization problem inherent in the classical MP algorithm, described in section 1.

(iii) The decomposition obtained can be organized in bit-planes as in equation (5). This can be done by noting that, in equation (6), the indexes r_m for the values of m such that $k_m = j$ correspond to the atoms comprising bit-plane j.

(iv) The number of atoms used in the decomposition can be set arbitrarily and each atom corresponds to a pair (k_m, r_m). This permits a precise rate control, since the decomposition can be stopped when the bit-budget is exhausted. This feature can be very useful in more sophisticated R-D schemes.

3. IMPLEMENTATION OF THE VIDEO ENCODER

In this section the effectiveness of Algorithm 1 will be evaluated by employing it in the framework of Neff and Zakhor's MP video encoder [2]. Essentially, Algorithm 1 will replace the decomposition and quantization strategy employed in [2], using exactly the same dictionary $\mathcal{D}$. The indexes r_m (see equation (6)) are encoded in the same way as the atom's indexes in [2]. On the other hand, instead of encoding the value of the inner product p_n (see equation (1)), the index k_m of the bit plane corresponding to the atom of index r_m is encoded. An adaptive arithmetic coder [9] is used for this purpose. Since we do not know at first what is the maximum value that k_m can assume (see equation (6)), we had to perform a slight modification to the arithmetic encoder in [9]. The initial number of possible indexes k_m is set to two (k_m=1 and k_m=2) plus an escape code. If we need to transmit k_m=3 we first transmit the escape code to indicate an increase in the number of symbols and then transmit the code for k_m=3. At this point the possible symbols are k_m=1,2,3 plus an escape code. The same process is repeated for each new value of k_m that is out of the current range. Also, when we start coding the next frame the number of possible values of k_m is the same as the one at the end of the previous frame.

Since Algorithm 1 assumes that the norm of the input signal is $\|\mathbf{x}\| \leq 1$, we need to compute, for each video frame, the largest norm of the macroblocks, $X_{\max}$. It is important to note that, for each macroblock, as in [2], we search for the closest atom by centering every atom in every pixel of the macroblock. This implies that the atoms searched for in a macroblock B_i overlap the neighboring macroblocks. Then, effectively, it is as if the dimension of the signal we are decomposing is not the one of macroblock B_i, but the dimension of B_i plus the pixels of the neighboring macroblocks overlapping with the atoms used to decompose B_i. Referring to figure 1, since the luminance macroblocks are 16 × 16, the value of $X_{\max}$ is computed for a region of $(15 + n_{\max}) \times (15 + n_{\max})$ centered in the macroblock ($n_{\max} \times n_{\max}$ is the support of the atom having largest support). In our case, $n_{\max} = 35$, and $X_{\max}$ is computed considering 50×50 windows centered in every macroblock (see figure 1). We also need to send the approximation scaling factor (α) at the header of the video sequence. Note that since the use of Algorithm 1 permits precise bit-rate control (see comment (iv) at the end of section 2), then the strategy used for bit-rate allocation was to divide the bit-budget of the sequence equally among all its frames. Clearly other more sophisticated rate control algorithms could be used taking advantage of the precise rate control that such decompositions may provide, as in [10].

4. EXPERIMENTAL RESULTS

We have coded the sequences Container, Hall-monitor, Mother-and-daughter, Silent-voice and Foreman with 300 QCIF frames at 30 frames/s, sub-sampled in time by factors of 4 (rates under 20kbits/s) and 3 (other rates) to generate 7.5 frames/s and 10 frames/s, respectively. Coding was done only on luminance component in bit-rates that vary in the range 10-100kbits/s.

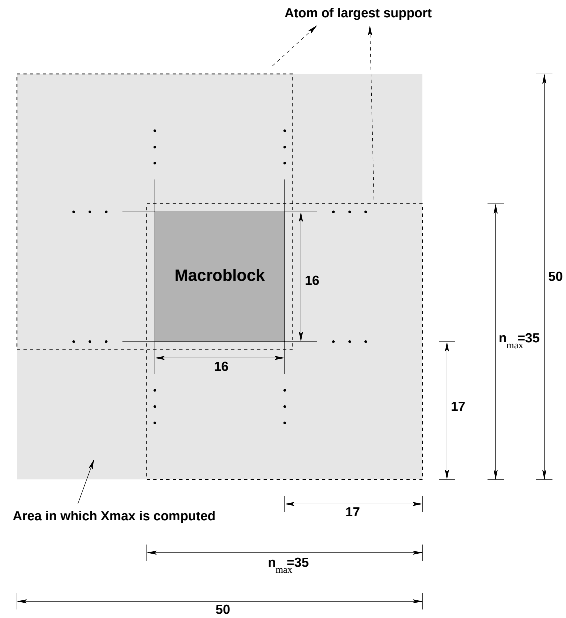

Fig. 1. Illustration of the area in which $X_{\max}$ is computed.

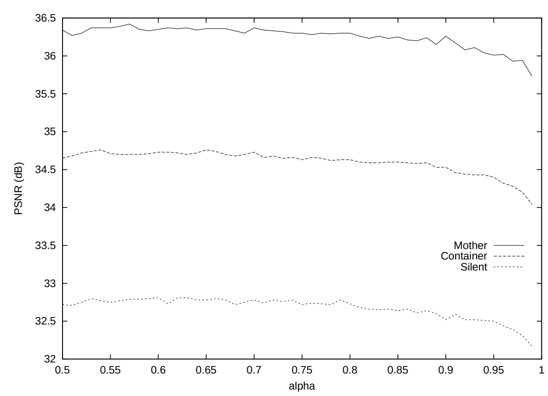

Fig. 2. Variation of the average PSNR with alpha parameter for Mother, Silent and Container sequences for 24kbits/s.

The value of α (see equation (6)) chosen at the beginning of coding interferes with the number of vectors used to code each frame. Small values of α lead to small values of k_m and consequently to a large number of vectors. Likewise, large values of α lead to large values of k_m and consequently to a small number of vectors. We can see then that there is a trade-off among the value of α, the number of vectors and the range of values of k_m. Therefore, the value of α can potentially affect the rate × distortion characteristics of the encoder. In figure 2 we can see the variation of the average PSNR with α. We can verify that the variation of α does not interfere significantly with the results, except when this parameter is next to one, when there is a significant drop in performance. Based in figure 2 an $\alpha = 0.56$ would be a good choice.

Table 1 compares the average peak signal to noise (PSNR) of the original matching pursuits video encoder (MP) [2] with our adaptation using generalized bit-planes (MPGBP) for some rates. The MPGBP encoder uses an α=0.56.

Seq + Rate	MPGBP	MP [2]	MPGBP - MP
Container10	32.54	32.45	0.06
Mother10	33.42	33.35	0.07
Hall10	33.43	33.30	0.13
Container24	34.70	34.47	0.23
Mother24	36.39	36.18	0.21
Hall24	36.59	36.13	0.46
Silent24	32.77	32.73	0.04
Container48	36.95	36.43	0.52
Mother48	39.21	38.45	0.76
Hall48	39.14	38.00	1.14
Silent48	36.35	35.90	0.45
Mother96	42.27	41.02	1.25
Foreman96	35.54	35.35	0.19

Table 1. Comparison, in terms of PSNR, between the two matching pursuits implementations.

Figure 3 shows the variation of the average PSNR with rate for both implementations of the matching pursuits encoders. In our scheme we used an α=0.56. We can see from this figure that the use of the generalized bit-planes scheme consistently improves the performance of the matching pursuits encoder from [2] for all rates. In addition, this improvement increases with the bit rate. The knee on the curves around 20kbps is due to the increase of the frame rate from 7.5 fps to 10 fps.

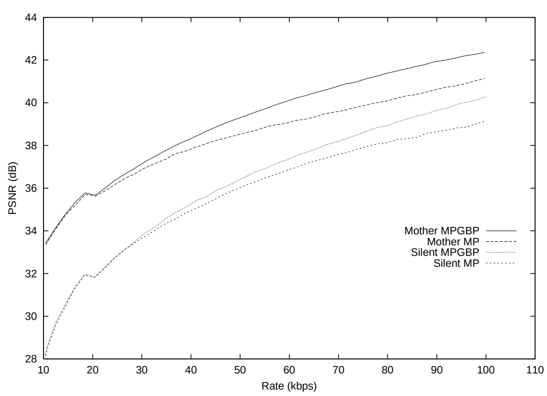

Fig. 3. Variation of the average PSNR with rate for Mother and Silent sequences.

In [3] Neff and Zakhor presented a more sophisticated quantization scheme for the inner product. It provides a significant improvement over the one in [2]. It is a two-pass algorithm in which, in the first pass, the frame difference is decomposed without quantization of the inner product in order to determine the number of atoms and the quantization step to be used in the second pass. In this algorithm, there is an obvious increase in computational complexity because the decomposition process must be repeated twice per frame. The MPGBP algorithm has an average performance as good as the one in [3] without need for any extra suppositions or any increase in computational complexity.

5. CONCLUSIONS

In this paper we have proposed a novel algorithm for performing matching pursuits decomposition. Instead of generating at its output a sequence of pairs comprising atoms indexes and corresponding coefficients, as in the classical MP algorithm, it generates just a sequence of atoms indexes. These indexes can be grouped in generalized bit-planes. The proposed algorithm has the advantage of obviating the need for setting up arbitrary trade-offs between number of atoms used and coefficients quantization. We have shown that the proposed algorithm corresponds to a generalization of the usual decomposition on a dictionary or basis followed by uniform scalar quantization. Also, we have stated a theorem setting a bound for the distortion obtainable for a decomposition in generalized bit-planes using a given number of atoms.

We have implemented an MP video encoder using the proposed algorithm replacing the classical MP decomposition and quantization. The results obtained are very promising, yielding a significant improvement over the classical video-MP algorithm [2]. It also provides a performance comparable to one obtained by the more sophisticated algorithms as, for example, the one in [3]. The generalized bit-plane decomposition obtained with this algorithm opens the possibility for more flexible implementations, as, for example, quad-tree based encoders using generalized bit-planes.

6. REFERENCES

[1] Stéphane G. Mallat and Zhifeng Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, December 1993.

[2] Ralph Neff and Avideh Zakhor, "Very low bit rate video coding based in matching pursuits," *IEEE Transactions Circuits and Systems*, vol. 7, no. 1, pp. 158–171, February 1997.

[3] Ralph Neff and Avideh Zakhor, "Modulus quantization for matching pursuits video coding," *IEEE Transactions Circuits and Systems for Video Technology*, vol. 10, pp. 895–912, 2000.

[4] Ralph Neff and Avideh Zakhor, "Matching pursuits video coding–part II: Operational models for rate and distortion," *IEEE Transactions Circuits and Systems for Video Technology*, vol. 12, pp. 27–39, 2002.

[5] Pierre Vandergheynst and Pascal Frossard, "Adaptive entropy-constrained matching pursuits quantization," *IEEE International Conference on Image Processing*, pp. 423–426, 2001.

[6] Amir Said and William A. Pearlman, "A new, fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 243–250, June 1996.

[7] ISO/IEC JTC1/SC29/WG1 (ITU/T SG28), "JPEG2000 verification model 5.3," 1999.

[8] Marcos Craizer, Eduardo Antônio Barros da Silva, and Eloane Garcia Ramos, "Convergent algorithms for successive approximation vector quantization with applications to wavelet image compression," *IEE Proceedings - Part I - Vision, Image and Signal Processing*, vol. 146, no. 3, pp. 159–164, June 1999.

[9] T. C. Bell, J. G. Cleary, and I. H. Witten, *Text Compression*, Prentice Hall, Englewood Cliffs, NJ, 1990.

[10] Rogério Caetano and Eduardo A. B. da Silva, "Rate control strategy for embedded wavelet video coders," *Electronics Letters*, vol. 35, no. 21, pp. 1815–1817, October 1999.