IMPROVED DICTIONARIES FOR GENERALIZED BITPLANES-BASED MATCHING PURSUITS VIDEO CODING USING RIDGELETS

Leonardo Hidd Fonteles, Rogério Caetano**, Eduardo A. B. da Silva**

*PEE/COPPE/DEL/EE/Universidade Federal do Rio de Janeiro
Cx.P. 68504, Rio de Janeiro, RJ, Brazil
**Fundação Desembargador Paulo Feitoza
Av. Danilo Areosa, S/N, L. 164, Distrito Industrial, Manaus, AM, Brazil
leonardo@lps.ufrj.br, rcaetano@fpf.br, eduardo@lps.ufrj.br

ABSTRACT

A novel decomposition of functions using generalized bit-planes has been recently proposed. Such a decomposition maps a function into a set of indexes, thus performing quantization and signal decomposition in a single step. In a previous work, this signal representation model has been successfully used replacing the decomposition/quantization operations in Neff and Zakhor's matching pursuits video encoder. In this paper we investigate good codebooks for such decompositions whose construction is based on ridgelets. We show that by combining products of properly scaled and rotated ridgelets one can obtain codebooks which provide improved rate-distortion performances. Also, we confirm the theoretical prediction that good codebooks are the ones that have a small maximum angle between any vector in $\mathbb{R}^N$ and the closest one in the codebook.

1. INTRODUCTION

In [1] it has been proposed an algorithm, called Matching Pursuits (MP), that decomposes a signal on a redundant set of functions. It has been successfully used for video coding in [2], being an effective alternative to the standard video encoders based on the discrete cosine transform (DCT). It greatly reduces blocking artifacts and improves the PSNR (Peak Signal-to-Noise Rate).

Such a video encoder decomposes the displaced frame difference $\mathbf{x} \in \mathbb{R}^N$, originated from the motion compensation process, using a dictionary $\mathcal{C} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_M\}$, $\|\mathbf{v}_i\| = 1$, $\forall i$, as:

$$\mathbf{x} \approx \sum_{n=1}^{P} p_n \mathbf{v}_{\gamma_n} \tag{1}$$

In order to accomplish this decomposition a greedy algorithm is used [1]. In signal coding applications, the coefficients p_n must be quantized and encoded along with the indexes γ_n.

Recently, it has been proposed a signal decomposition based on generalized bit-planes [3]. In it, the normalized displaced frame difference (dfd) is decomposed in a greedy fashion generating, given $0 < \alpha < 1$, a decomposition as:

$$\mathbf{x}^{(P)} = \sum_{m=1}^{P} \alpha^{k_m} \mathbf{v}_{r_m} \tag{2}$$

Such a decomposition maps a signal into a sequence of indexes (k_m, r_m), thus performing the decomposition and quantization at same time. The *generalized Bitplane* j is defined as the set of vectors $\{\mathbf{v}_{r_m} \mid k_m = j\}$ [3].

An algorithm to carry out a decomposition as in Eq. (2) adds one $\mathbf{v}_{r_n}$ at a time, until a rate and/or distortion criterion is met. Given a dictionary $\mathcal{C} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_q\}$, $\|\mathbf{v}_i\| = 1$, $\forall i$, it is as follows:

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 $\mathbf{w} \cdot \mathbf{v}_{r_m} = \max_{1 \le j \le q} \{\mathbf{w} \cdot \mathbf{v}_j\}$.
   (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.

Conditions for $\mathbf{x}^{(P)}$ in Eq. (2) to converge to $\mathbf{x}$ as the number of terms P tends to infinite are given by the following theorem [3] :

Theorem 1: Be $\mathbf{x} \in \mathbb{R}^N$, $\|\mathbf{x}\| \le 1$, such that it is approximated by MPGBP Algorithm using a dictionary $\mathcal{C}$ with P steps, generating $\mathbf{x}^{(P)}$ as in Eq. (2), 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 $\|\mathbf{r}^{(P)}\| = \|\mathbf{x} - \mathbf{x}^{(P)}\| \le \beta_c^P$, where $\beta_c = \sqrt{1 - (2\alpha - \alpha^2)\cos^2(\Theta(\mathcal{C}))} < 1$ for every $0 < \alpha < 1$ and $0 \le \Theta(\mathcal{C}) < \frac{\pi}{2}$.

In [3] there is a performance comparison between the MPGBP and the MP algorithms when used within the Neff

This work has been supported in part by CAPES, Brazil

and Zakhor's video encoder framework [2]. The dictionary used in both cases was the one proposed in [2]. The results show that the performance of the algorithm in [3] is consistently better than one in [2]. However, an important question remained without an answer: is there another dictionary that produces a better performance with the MPGBP algorithm?

From Theorem 1, we see that in pass P the error is bounded by β_c^P. Therefore, the smaller β_c is, then the smaller is the distortion after P passes. Since β_c is a decreasing function of $\Theta(\mathcal{C})$, then, for small distortion, we should have $\Theta(\mathcal{C})$ small. However, the rate after P passes increases with the cardinality $q(\mathcal{C})$ of the dictionary (actually, with $\log(q(\mathcal{C}))$). Therefore, for a good rate-distortion compromise, $\Theta(\mathcal{C})$ should be made small provided that $q(\mathcal{C})$ does not increase too much.

Since $\Theta(\mathcal{C})$ is the largest angle between any vector in $\mathbb{R}^N$ and the closest one in $\mathcal{C}$, it is obtained for the region in $\mathbb{R}^N$ with the largest "hole" (see Fig. 1a). From the 2-D example depicted in Fig. 1, we can infer that, given a dictionary $\mathcal{C}$ (Fig. 1a), a reduction in $\Theta(\mathcal{C})$ can be obtained by: **1)** the insertion of extra vectors in order to fill appropriately the "empty" regions of the space (Fig. 1b), or **2)** a better distribution of its vectors (Fig. 1c). Note that in **2)** we preserve both the cardinality $q(\mathcal{C})$ and the dimension N of the dictionary $\mathcal{C}$. On the other hand, in **1)**, the cardinality $q(\mathcal{C})$ is increased, and there is a trade-off between the decrease in $\Theta(\mathcal{C})$ and the increase in $q(\mathcal{C})$.

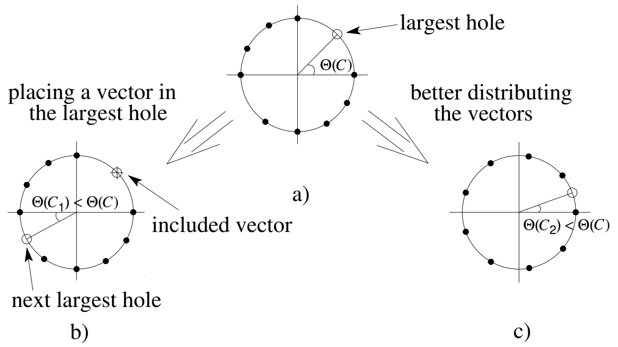

Fig. 1. $\Theta(\mathcal{C})$ in a) can be reduced by: b) placing an extra vector in the largest hole of the dictionary $\mathcal{C}$ in $\mathbb{R}^2$; c) better distributing the vectors in $\mathbb{R}^2$.

This paper deals with searching good dictionaries for MPGBP. From the above discussion, they must present good trade-offs between $q(\mathcal{C})$ and $\Theta(\mathcal{C})$. Ridgelets [4] (as well as curveletes [5] and contourlets [6]) have been attracting a lot of interest from the image coding community. They yield good sparse representations of line discontinuities. Since the edges of an image can be seen as discontinuities along curves, they represent effective alternatives for image decompositions [4]. In this work, we investigate the performance of the MPGBP video encoder when using overcomplete dictionaries based on ridgelets.

This paper is organized as follows. Section 2 discusses dictionary design. In Section 3 we describe the encoder used, that has a rate control algorithm for the MPGBP that provides a precise control over the rate. It enables a fair comparison among different dictionaries. In Section 4 we specify the atoms used, presenting and analyzing the simulation results. The conclusions are presented in Section 5.

2. DICTIONARY DESIGN

As seen in the previous section, we search for codebooks with both $\Theta(\mathcal{C})$ and $q(\mathcal{C})$ as small as possible. One should note that any rotation $\mathcal{C}'$ of the codebook $\mathcal{C}$ has $\Theta(\mathcal{C}') = \Theta(\mathcal{C})$, and thus, according to Theorem 1, should have the same performance when used in MPGBP. However, for the first iteration of the MPGBP algorithm, the reduction in the approximation error depends only on how similar are the atoms in the dictionary to the image features. This is so because an atom which is similar to an image feature has a large inner product to it, and thus, a small value of $\|\mathbf{x} - \alpha^{k_m}\mathbf{v}_{r_m}\|$. Thus, among the codebooks with same value of $\Theta(\mathcal{C})$, one should search for the one whose vectors are more similar to the image features. That is, a good codebook should:

(i) have small values of $\Theta(\mathcal{C})$ and $q(\mathcal{C})$;

(ii) have its atoms as similar as possible to the image features.

One should bear in mind that, in video coding, we want to encode displaced frame differences (dfds). Dfds are composed essentially of contours (edges) oriented in several spacial directions. Thus, atoms based on ridgelets, that are known to be efficient to encode line singularities in arbitrary directions, are good candidates to satisfy condition (ii) above.

However, if we want to satisfy condition (i) above, we cannot use an orthogonal ridgelet transform. This so is because any orthogonal transform is just a rotation of any other orthogonal transform and thus, for a given dimension, they all have exactly the same $\Theta(\mathcal{C})$ (see discussion at the start of this section). Therefore, in order to decrease $\Theta(\mathcal{C})$ we should use overcomplete dictionaries based on ridgelets. Note that, by referring to Fig. 1 this is equivalent to adding atoms to fill in the large holes.

The dictionaries are generated according to the procedure described as follows. We start by defining an infinite discrete ridgelet whose ridge has angle ϖ with the horizontal axis as

$$\rho_{s,l,\varpi}^{(f)}(m,n) = f_s(m \cdot \tan(\varpi) + n) \cdot W_l(m \cdot \tan(\varpi) + n) \quad (3)$$

where $f_s(n)$ is usually a wavelet function at scale s, that is $f_s(n) = \frac{1}{\sqrt{s}} f(n/s)$ and W_l is a window of length l defined as

$$W_l(n) = \begin{cases} 1, & n \in [-\frac{l-1}{2}, \frac{l-1}{2}] \\ 0, & n \notin [-\frac{l-1}{2}, \frac{l-1}{2}] \end{cases} \quad (4)$$

In this work, $f(n)$ can be a Meyer wavelet, a Meyer scaling function [7], or one of the Gabor functions used by Neff and Zakhor [2, 3]. The periodic Meyer wavelets and scaling functions can be defined in the frequency domain as

$$\Psi(\omega) = \begin{cases} \frac{1}{\sqrt{2\pi}} e^{i\omega/2} \sin\left(\frac{\pi}{2}\nu(\frac{3}{2\pi}|\omega| - 1)\right), & \frac{2\pi}{3} \le |\omega| \le \frac{4\pi}{3} \\ \frac{1}{\sqrt{2\pi}} e^{i\omega/2} \cos\left(\frac{\pi}{2}\nu(\frac{3}{4\pi}|\omega| - 1)\right), & \frac{4\pi}{3} \le |\omega| \le \frac{8\pi}{3} \\ 0, & |\omega| \notin [\frac{2\pi}{3}, \frac{8\pi}{3}] \end{cases} \quad (5)$$

$$\Phi(\omega)=\begin{cases}\frac{1}{\sqrt{2\pi}}, & |\omega|\le\frac{2\pi}{3}\\ \frac{1}{\sqrt{2\pi}}\cos\left(\frac{\pi}{2}\nu(\frac{3}{2\pi}|\omega|-1)\right), & \frac{2\pi}{3}\le|\omega|\le\frac{4\pi}{3}\\ 0, & |\omega|\ge\frac{4\pi}{3}\end{cases} \quad (6)$$

where $\nu(a)=a^4(35-84a+70a^2-20a^3), a\in[0,1]$

Using Eq. (3), the atoms in the dictionaries are then

$$g^{(f_1,f_2)}_{s_1,s_2,l_1,l_2,\varpi}(m,n)=\rho^{(f_1)}_{s_1,l_1,\varpi}(m,n)\cdot\rho^{(f_2)}_{s_2,l_2,\frac{\pi}{2}-\varpi}(m,n) \quad (7)$$

The above equation can be regarded as an infinite-length ridgelet windowed along its ridge by another infinite-length ridgelet. Note that the atoms generated according to Eq. (7) have a finite region of support. This is so because each infinite ridgelet has finite length in the direction orthogonal to its ridges (see Eqs. (3) and (4)).

It is important to note that the atoms in Eq. (7) are in general non-separable. The use of non-separable dictionaries can lead to a large increase in computational complexity. Since for each atom encoded we must search the whole dictionary for the largest inner product, the complexity is proportional to both the dictionary size and the dimension of the vector space. In order to speed up the atom search one can use one of the several methods for computational complexity reduction have been proposed in the literature. A good example of such methods is the one in [8].

3. ENCODER USED

The encoder used was essentially the one in [3] with the difference that we employ a simple but precise rate control mechanism. We do so because, in order to have meaningful comparisons among the different dictionaries, we need the rates for them to be exactly the same.

The rate control mechanism essentially computes the rate spent for each new encoded atom and stops adding atoms when the desired bit-rate is achieved. Note that to add an atom is equivalent to adding a pair of indexes (k_m, r_m). The atom index is composed of both the atom shape and its position. The atom shape, as well as the exponent k_m, are encoded in exactly the same way as in [3]. However, the position of the atoms inside a 16 $\times$ 16 macro block is encoded differentially (see [2] and [3]). Then, in order to have precise computation of the bit-rate we should recompute, for each encoded atom added, the bit-rate spent for encoding the differential position of all encoded atoms.

4. SIMULATION RESULTS

In this section we describe the simulation results. We use the encoder described in Section 3. All sequences are composed of QCIF images (176×144 resolution frames) and we input to the codec the first 300 frames, at a frame rate of 10fps (that is, we encode a total of 100 frames for each sequence). In all cases $\alpha = 0.56$ (see Eq. (2) and reference [3]).

In order to evaluate the performance of the codec for different dictionaries, we use the PSNR of the encoded sequences, along with the parameters $\Theta(\mathcal{C})$ and $\overline{\Theta}(\mathcal{C})$. $\Theta(\mathcal{C})$ is as defined in Theorem 1. As mentioned is Section 1, the smaller $\Theta(\mathcal{C})$ is, the better is the dictionary. If we define for each pass j a value $\beta^{(j)}$ such that the magnitude of the residual in pass $j+1$, $\|r^{(j+1)}\|$, is equal to $\beta^{(j)}\|r^{(j)}\|$, then $\overline{\Theta}(\mathcal{C})$ is such that $\beta_c(\overline{\Theta}(\mathcal{C}))=\mathbf{E}[\beta^{(j)}]$ (see Theorem 1). $\mathbf{E}[\cdot]$ is the expected value, obtained over all passes and over the entire sequence. Since we want the largest residual reduction at each pass, then we need $\mathbf{E}[\beta^{(j)}]$ to be as small as possible. This implies that, for a good dictionary, $\overline{\Theta}(\mathcal{C})$ should be as small as possible.

As a starting point, we use the dictionary employed in [2] (from now on referred to as the NZ - Neff and Zakhor dictionary), that has very good performance in Matching Pursuits video coding, and has been shown to also perform well in the MPGBP algorithm [3]. The results obtained with this dictionary are summarized in Table 1 for convenience. The cardinality of the NZ dictionary is $q(\mathcal{C}) = 400$.

Table 1. PSNR, $\Theta(\mathcal{C})$ and $\overline{\Theta}(\mathcal{C})$ for the NZ dictionary.

Seq+Rate (kbps)	NZ		
	PSNR	$\Theta(\mathcal{C})$	$\overline{\Theta}(\mathcal{C})$
Mother24	36.19	86.17	82.29
Weather24	31.75	87.52	82.61
Silent24	32.59	87.08	80.19
Mother64	40.38	87.19	83.51
Weather64	37.56	88.26	83.60
Silent64	37.72	87.81	82.25
Foreman64	33.45	86.80	80.77
Foreman96	35.52	87.07	81.60
Weather96	40.38	88.48	84.00

We now describe the other dictionaries investigated. For each dictionary we define the functions f_1 and f_2 used to form the atoms in Eq. (7), as well as the sampling of the parameter space $(s_1, s_2, l_1, l_2, \varpi)$. For example, in the NZ dictionary [2], we have that for $i = 1, 2$, f_i can assume the values of all the 20 modulated Gaussian functions, l_i is the support region of f_i, $s_i = 1$ and $\varpi = 0$.

One of the reasons for the good performance of the NZ dictionary is that its atoms are similar to the features present in the dfds, that is, they satisfy requirement (ii) in Section 2. Thus, if we design our dictionaries by growing them from the NZ dictionary, we guarantee that requirement (ii) is satisfied at least by some atoms. Indeed, we design our dictionaries by, starting from the NZ dictionary, adding other atoms in order to fill in its holes so as to decrease $\Theta(\mathcal{C})$ (see Fig. 1), and thus satisfy requirement (i). The atoms added are based on ridgelets since we want them to satisfy requirement (ii) (see discussion in Section 2).

In Table 2, we see the results for a dictionary formed by adding, to the NZ dictionary, atoms composed from ridgelets based on Meyer functions (dictionary MEYER). In this case, f_i, $i = 1, 2$, can be either the Meyer wavelets or scaling functions (see Eqs. (5) and (6)). The combinations of lengths and scales used, (l_i, s_i), $i = 1, 2$ (see Eq. (7)), are shown in Table 3. The rotation angles ϖ for each atom with lengths l_1 and l_2 have been chosen according to the criterion suggested by Donoho and Flesia in [7], that is, they are given by

$$\varpi_b=\arctan\left(\frac{2b}{L-1}\right), \quad -\frac{L-1}{2}\le b\le\frac{L-1}{2} \quad (8)$$

where $L = \max\{l_1, l_2\}$. The cardinality of the resulting MEYER dictionary was $q(\mathcal{C}) = 24617$ vectors.

Table 2. PSNR, $\Theta(\mathcal{C})$ and $\overline{\Theta}(\mathcal{C})$ for the MEYER dictionary.

Seq+Rate (kbps)	MEYER			MEYER − NZ	
	PSNR	$\Theta(\mathcal{C})$	$\overline{\Theta}(\mathcal{C})$	ΔPSNR	$\Delta\overline{\Theta}(\mathcal{C})$
Mother24	36.46	85.36	80.81	0.27	−1.48
Weather24	31.85	86.46	81.23	0.10	−1.38
Silent24	32.63	85.86	78.64	0.04	−1.55
Mother64	40.51	86.12	82.25	0.13	−1.26
Weather64	37.80	86.91	82.37	0.24	−1.23
Silent64	37.65	86.56	80.98	−0.07	−1.27
Foreman64	33.83	85.84	78.97	0.38	−1.73

Table 3. Lengths (l) and scales (s) for the atoms used.

l	1	3	5	7	7	9	9	11	13
s	0	1	2	1.5	2	2	2.5	3	3
l	15	21	21	23	27	29	35	35	–
s	3	3	3.5	4	4	4	4.5	5	–

We can see that we have a reduction of the values of $\Theta(\mathcal{C})$ and $\overline{\Theta}(\mathcal{C})$ in all cases, accompanied by an improvement in PSNR as compared to the one obtained with the NZ dictionary. It is important to note that such an improvement occurred for most sequences and rates, even though there was a significant increase in the cardinality $q(\mathcal{C})$. This confirms the conjecture, based on Theorem 1, that one can form good dictionaries for MPGBP by decreasing $\Theta(\mathcal{C})$ even at the expense of increased cardinality. That is, the rate increase produced by the increase in $q(\mathcal{C})$ has been compensated by the decrease in distortion caused by the reduction in $\Theta(\mathcal{C})$.

Finally, in Table 4, we see the results obtained for a dictionary formed from the NZ dictionary by adding atoms based on ridgelets other than the ones generated by the Meyer functions in Eqs. (5) and (6). The atoms used are based on ridgelets generated using the 20 Gabor functions defined in [2]. We refer to it as the GR dictionary. Its cardinality is $q(\mathcal{C}) = 8090$. Again we note that we got a substantial increase in PSNR for all sequences and rates along with a reduction in the values of $\Theta(\mathcal{C})$ and $\overline{\Theta}(\mathcal{C})$. Indeed, it yields a better rate-distortion performance than the one of the MEYER dictionary. This is not unexpected, since the Gabor functions in [2] were obtained from a training process aiming at generating separable 2-D functions similar to the features presented in dfds (that is, they were designed specifically to satisfy requirement (ii)).

5. CONCLUSIONS

In this paper we investigated good dictionaries for the MPGBP algorithm. We defined conditions to be satisfied by a good dictionary, namely, to have a small value of $\Theta(\mathcal{C})$ and to have atoms as close as possible to the image's features. Then we proposed a dictionary design method based on ridgelets. We designed dictionaries obtained by the addition, to the Neff and Zakhor's dictionary [2], of atoms

Table 4. PSNR, $\Theta(\mathcal{C})$ and $\overline{\Theta}(\mathcal{C})$ for the GR dictionary.

Seq+Rate (kbps)	GR			GR − NZ	
	PSNR	$\Theta(\mathcal{C})$	$\overline{\Theta}(\mathcal{C})$	ΔPSNR	$\Delta\overline{\Theta}(\mathcal{C})$
Mother24	36.53	85.69	81.03	0.34	−1.26
Weather24	32.25	86.62	81.27	0.50	−1.34
Silent24	32.74	86.29	78.90	0.15	−1.29
Mother64	40.72	86.51	82.35	0.34	−1.16
Weather64	38.39	87.20	82.34	0.83	−1.26
Silent64	37.96	86.92	81.14	0.24	−1.11
Foreman64	33.93	85.94	79.25	0.48	−1.52
Foreman96	36.01	86.34	80.30	0.49	−1.30
Weather96	41.41	87.30	82.72	1.03	−1.28

based on ridgelets generated by both Meyer's wavelets and 1-D Neff and Zakhor's Gabor functions. The results obtained were encouraging, showing two dictionaries in which, despite the large increase in cardinality (and thus the rate to code each atom), the reduction in $\Theta(\mathcal{C})$ compensates the rate increase, yielding significant improvement in PSNR for all sequences and rates.

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] Rogério Caetano, Eduardo A. B. da Silva, and Alexandre G. Ciancio, "Matching pursuits video coding using generalized bit-planes," *International Conference on Image Processing*, vol. 3, no. 24–28, pp. 677–680, June 2002.

[4] Emmanuel J. Candès and David L. Donoho, "Ridgelets: a key to higher-dimensional intermittency?," *Phil. Trans. R. Soc. Lond. A.*, vol. 357, pp. 2495–2509, June 1999.

[5] Emmanuel J. Candès and David L. Donoho, "Curvelets - a surprisingly effective nonadaptive representation for objects with edges," In A. Cohen, C. Rabut and L. L. Schumaker (eds.)*, Curve and Surface Fitting,* Saint-Malo, Vanderbilt University Press, 1999.

[6] Minh N. Do and Martin Vetterli, "Contourlets: a directional multiresolution image representation," *International Conference on Image Processing*, vol. 1, pp. I–357 – I–360, September 2002.

[7] David L. Donoho and Ana Georgina Flesia, "Digital ridgelet transform based on true ridge functions," In Grant Welland (ed.)*, Beyond Wavelets,* Academic Press, pp. 1–30, September 2003.

[8] Ralph Neff and Avideh Zakhor, "Matching pursuits video coding - part 1: Dictionary approximation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 1, pp. 13–26, January 2002.