Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE @ DIRECT®

Signal Processing 83 (2003) 1937–1943

SIGNAL PROCESSING

www.elsevier.com/locate/sigpro

Gain–shape optimized dictionary for matching pursuit video coding

Yao-Tang Chou[a], Wen-Liang Hwang[b,*], Chung-Lin Huang[a]

[a] *Electrical Engineering Department, National Tsing Hua University, Taiwan*
[b] *Institute of Information Science, Academia Sinica, Nankang, Taipei, Taiwan*

Received 14 September 2001; received in revised form 10 October 2002

Abstract

We show that the vector quantizer (VQ) techniques can be used to optimize a matching pursuit dictionary and improve the PSNR performance at low bitrates. The basis functions are the shape part, and the inner product values are the gain part of the VQ. The performance is evaluated based on a comparison of the PSNR in encoding motion residuals obtained using a matching pursuit coder with a reference dictionary and that obtained from the coder with a gain–shape optimized dictionary.

Keywords: Matching pursuit; Vector quantization; Video compression

1. Introduction

The matching pursuit (MP) algorithm was first proposed to analyze the time–frequency structures of a signal using an over-complete basis set (dictionary) [8]. In this article, vector quantizer (VQ)-based techniques are proposed for optimizing an MP dictionary for video coding. Since Vetterli and Kalker [15] proposed their first MP video coder, various approaches to improving the coding efficiencies and enhancing the functionalities of an MP video codec have been proposed. Results presented in [9] indicate that MP coding of motion residuals yields performance superior to that of H.263 and MPEG-4 from both quantitative (PSNR) and subjective points of view. In [1], efficient methods were proposed to encode atom positions. Proposed in [2] was a procedure for selecting between MP and block-DCT. In [1,6,16], SNR scalability was added to an MP video codec. A simple MP-based multiple description strategy was presented in [14]. Dictionary approximation techniques aimed at speeding up an MP-encoder were proposed in [10]. In this article, we extend our results in [3] by providing a VQ-based atom optimization method that is particularly necessary for an MP video codec. The VQ approach to designing MP dictionaries for video coding was also studied in [13]. A technique for designing frames to approximate each training vector for an MP was proposed in [4] for speech and electrocardiogram(ECG) signals analysis.

In Neff and Zakhor's MP codec [9], an efficient subset of a large basis set is selected as a dictionary, according to the residual patterns in some video sequences, to reduce the number of inner products at each iteration. However, their method simply selects an efficient subset from among these sequences, and the subset is not necessarily optimized to the

[*] Corresponding author.
E-mail address: whwang@iis.sinica.edu.tw (W-L. Hwang).

doi:10.1016/S0165-1684(03)00112-9

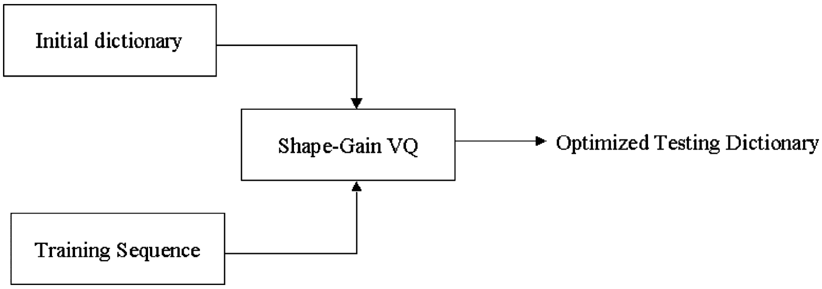

Fig. 1. Block diagrams of dictionary optimization methods.

sequences. For this reason, we adopt a popular VQ-based optimization technique for designing the dictionary of an MP codec. We show that it is worthwhile to use VQ design methods to learn MP dictionaries to encode motion residuals. Fig. 1 shows block diagrams of the proposed method. The advantage of the proposed method is that once a test sequence is approximated efficiently using the trained-dictionary (i.e. codebooks), the proposed method provides good performance for the test sequence. However, like most of the VQ-based algorithms, our approach suffers when the test sequence and the training sequences for the trained dictionary have different characteristics. This problem is well known in VQ design, and many interesting related subjects have been discussed in [12].

The implementation of our MP video codec is not optimal. The performance of our video codec could be improved if it were implemented according to those algorithms proposed in [9,1] for atom positions. In Section 2, the gain–shape product VQ techniques for optimizing the dictionary of an MP codec will be introduced. In Section 3, experimental results will be given. Conclusions will be drawn in Section 4.

2. Matching pursuit dictionary optimization using VQ

We employ VQ techniques to optimize an MP dictionary. Before describing our method, we will briefly review the matching pursuit theory and the gain–shape product VQ.

2.1. Matching pursuit theory

The matching pursuit algorithm represents an image by means of linear expansion of the image with respect to a dictionary, D, which is a collection of many complete bases; for example, both the Fourier and wavelet bases can belong to the dictionary, and the dictionary is dense for all $L^2(R^2)$ functions [8]. The image function f can be decomposed into

$$f = \langle f, g_{\gamma_0} \rangle g_{\gamma_0} + Rf,$$

where Rf is the residual image after approximating f in the direction of g_{γ_0}, and g_{γ_0} is chosen such that $|\langle f, g_{\gamma_0} \rangle|$ is maximum in D. This procedure is performed each time on the following residue that is obtained. It was showed that for an image f,

$$f = \sum_{n=0}^{+\infty} \langle R^n f, g_{\gamma_n} \rangle g_{\gamma_n}.$$

2.2. Gain–shape vector quantizer

The purpose of a VQ is to approximate a random input vector to its close representative vector (called a codeword); then, the index of the codeword can be effectively encoded and transmitted [7,5]. The decoder can easily, with the help of codebooks, recover the corresponding codeword from an index. However, in high-dimensional space, the vector space is too big to manage efficiently, in terms of both computational storage and time needed by a VQ. Therefore, the VQ is usually constrained. The constrained VQ leads to increased efficiency in terms of time and storage, but the accuracy of the VQ is degraded. The gain–shape product VQ is a constrained VQs. It has been successfully used to code the waveforms as well as the parameters of speech signals [11]. The gain–shape product VQ assumes that the element X in the training random variable $\underline{X}$ has a product code structure:

$$X = gS,$$

where g is a scale (called the gain) and S is a normalized vector (called the shape). For a detailed description of the codec structure and the optimized codebook design for a gain–shape product VQ, the reader is referred to [5,11].

2.3. Dictionary optimization

Let $\underline{X}$ be our random variable for dictionary optimization, and let the original dictionary contain only separable basis: $\{g_\alpha(x)g_\beta(y)|\alpha,\beta\}$, where $\|g_\alpha(x)g_\beta(y)\| = 1$ and the size of the basis function in the dictionary is truncated to $b \times b$. $g_\alpha(x)$ and $g_\beta(y)$ each is a 1D basis functions at which index is, respectively, to be α and β. Let $f_k(x,y)$ be the kth motion residual frame in the training sequence, and let the lth matching pursuit residual of the kth frame be $R^l f_k(x,y)$. The matching pursuit decomposition of $f_k(x,y)$ up to $N(k)$ atoms is

$$f_k(x,y) = \sum_{n=0}^{N(k)-1} \langle F(R^n f_k), g_{\alpha_n}(x)g_{\beta_n}(y)\rangle g_{\alpha_n}(x)g_{\beta_n}(y) + R^{N(k)} f_k(x,y), \quad (1)$$

where F is composed of the following steps: finding the largest energy block (the highlighted block shown in the left part of Fig. 2) and then restricting the matching pursuit on the block(as shown in the right part of the figure). The image patch, say of size $b \times b$, where the first atom is selected by applying matching pursuit to the block, is the result of F. Our training random variable $\underline{X}$ is obtained from all the matching pursuit residuals in our training sequence:

$$\underline{X} = \{F(R^n f_k)(x,y); k = 1,2,\ldots; n = 0,1,\ldots,N(k)-1\}.$$

The vector space of our random variable $\underline{X}$ is of dimension $b \times b$, which is the size of a basis function in our dictionary. A basis function is a vector mapped to the points $\vec{s}$ on the radial 1 sphere in a vector space. Similarly, an element in the random variable $\underline{X}$ is mapped to a point, say $\vec{p}$, in the vector space. The inner product of the element and the basis function is equivalent to the projection of $\vec{p}$ in the direction of $\vec{s}$. In our implementation, b is set to 35; the resultant vector space is of size 35×35, which is too big for an efficient VQ design. We thus consider the gain–shape product VQ for our task because the basis functions form the shapes, and the inner product values form the gains.

Our objective is to find the shape and gain codebooks that minimize the average distortion incurred in encoding the training vectors in $\underline{X}$. Since the dictionary used in our implementation is constructed from the tensor product of two one-dimensional basis sets, two shape codebooks, corresponding to the shape along the x-dimension and that along the y-dimension, respectively, are required. Let N_g and N_s indicate, respectively, the number of regions of the gain and shape partitions, and let R be the regions in the vector space. We have $R = \{R_{i,j,k}; i,j = 1,\ldots,N_s; k = 1,\ldots,N_g\}$.

Three codebooks are adopted in our method:

- the gain codebook $C_g = \{\hat{g}_k; k = 1,2,\ldots,N_g\}$, where $\hat{g}_k$ is the centroid of the region $G_k = \bigcup_{i,j=1}^{N_s} R_{i,j,k}$, which is the region of the input vector space that maps into the gain codeword $\hat{g}_k$;
- the x-axis shape codebook $C_{sx} = \{\hat{S}x_i; i = 1,\ldots,N_s\}$, where $\hat{S}x_i$ is the centroid of the region $Ax_i = \bigcup_{j=1}^{N_s}\bigcup_{k=1}^{N_g} R_{i,j,k}$, which is the region of the input vector space that maps into the x-axis shape codeword $\hat{S}x_i$;
- the y-axis shape codebook $C_{sy} = \{\hat{S}y_j; j = 1,\ldots,N_s\}$, where $\hat{S}y_j$ is the centroid of the region $Ay_j = \bigcup_{i=1}^{N_s}\bigcup_{k=1}^{N_g} R_{i,j,k}$, which is the region of the input vector space that maps into the y-axis shape codeword $\hat{S}y_j$.

Finally, we adopt a partition $R = \{R_{i,j,k}; i,j = 1,\ldots,N_s; k = 1,\ldots,N_g\}$ of $N_s \times N_s \times N_g$ cells describing the encoder, that is, if X is an element of $\underline{X}$ and $X \in R_{i,j,k}$, then X is mapped into (i,j,k). We express this as $g(X) = \hat{g}_k$, $Sx(X) = \hat{S}x_i$, and $Sy(X) = \hat{S}y_j$. The average distortion is defined by

$$D(C_g, C_{sx}, C_{sy}, R) = E\{d(\underline{X}, g(\underline{X})Sx(\underline{X})Sy(\underline{X}))\},$$

where d is the Euclidean distance. After slight modification of the VQ theory, there are four necessary

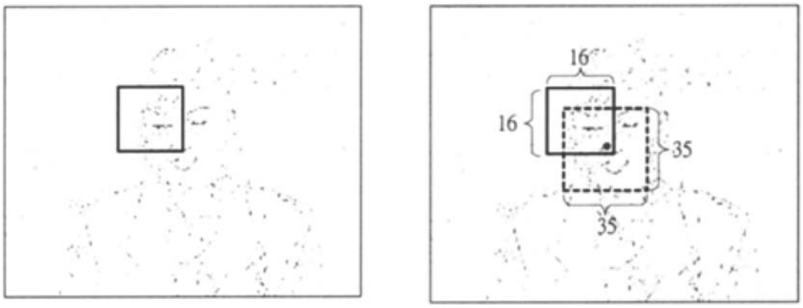

Fig. 2. Left: Highlighted block has the largest energy. Right: Inner product block search is limited around the block.

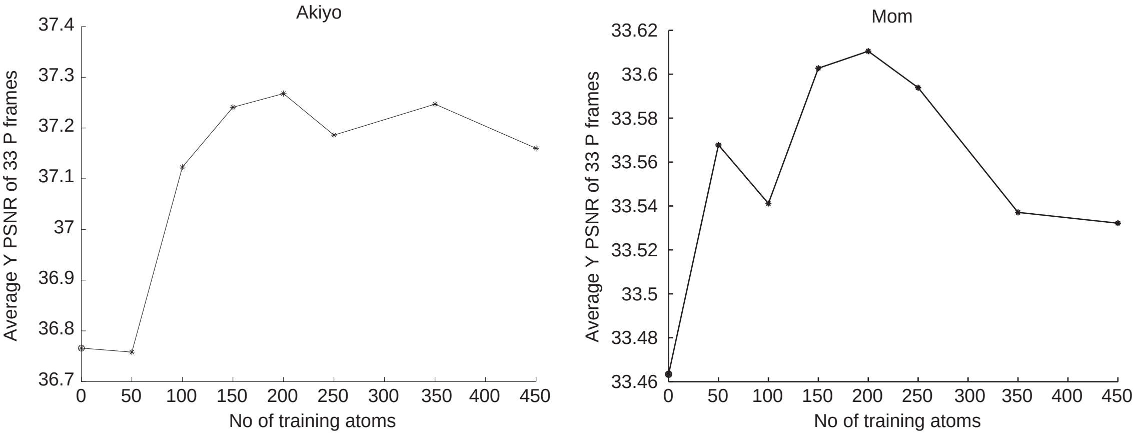

Fig. 3. The PSNR performance of our MP codec for various testing sequences using the dictionary trained by selecting various numbers of atoms from each residual frame in all the training sequences. Left: Testing Akiyo. Right: Testing Mother and Daughter. Dictionary with 200 atoms for each residual gives the best PSNR for these sequences.

conditions for optimal codebook design:

- optimal partition for fixed C_g, C_{sx}, and C_{sy};
- optimal C_g for fixed C_{sx}, C_{sy}, and partition R;
- optimal C_{sx} for fixed C_{sy}, C_g, and partition R;
- optimal C_{sy} for fixed C_{sx}, C_g, and partition R.

Given an initial set of three codebooks and the four necessary equations, the well-known iterative joint optimization algorithm can be used to obtain a locally optimal quantizer [5].

3. Experimental results

Various performance evaluations were carried out to demonstrate the performance of an MP codec using a dictionary optimized by gain–shape VQ. Five video sequences; Akiyo, Mother and Daughter, Sean, Miss America, and Salesman, were used in our experiments for obtaining a trained dictionary. The common features of these sequences are that they contain mostly one or two head-and-shoulder-type objects, and that there is not much fast global motion of these objects against their backgrounds. Each video sequence is encoded at 10 frame/s and in QCIF format for 3 s. Fig. 3 plots the performance evaluation of our MP codec for various testing sequences using the dictionary trained by selected from each residual frame a given number of atoms from all the testing video sequences. The test sequences are coded at 24 kbit/s, 10 frame/s, and in the QCIF format. The motion estimation part of our method for all the sequences is identical to that of H.263, which is available publicly

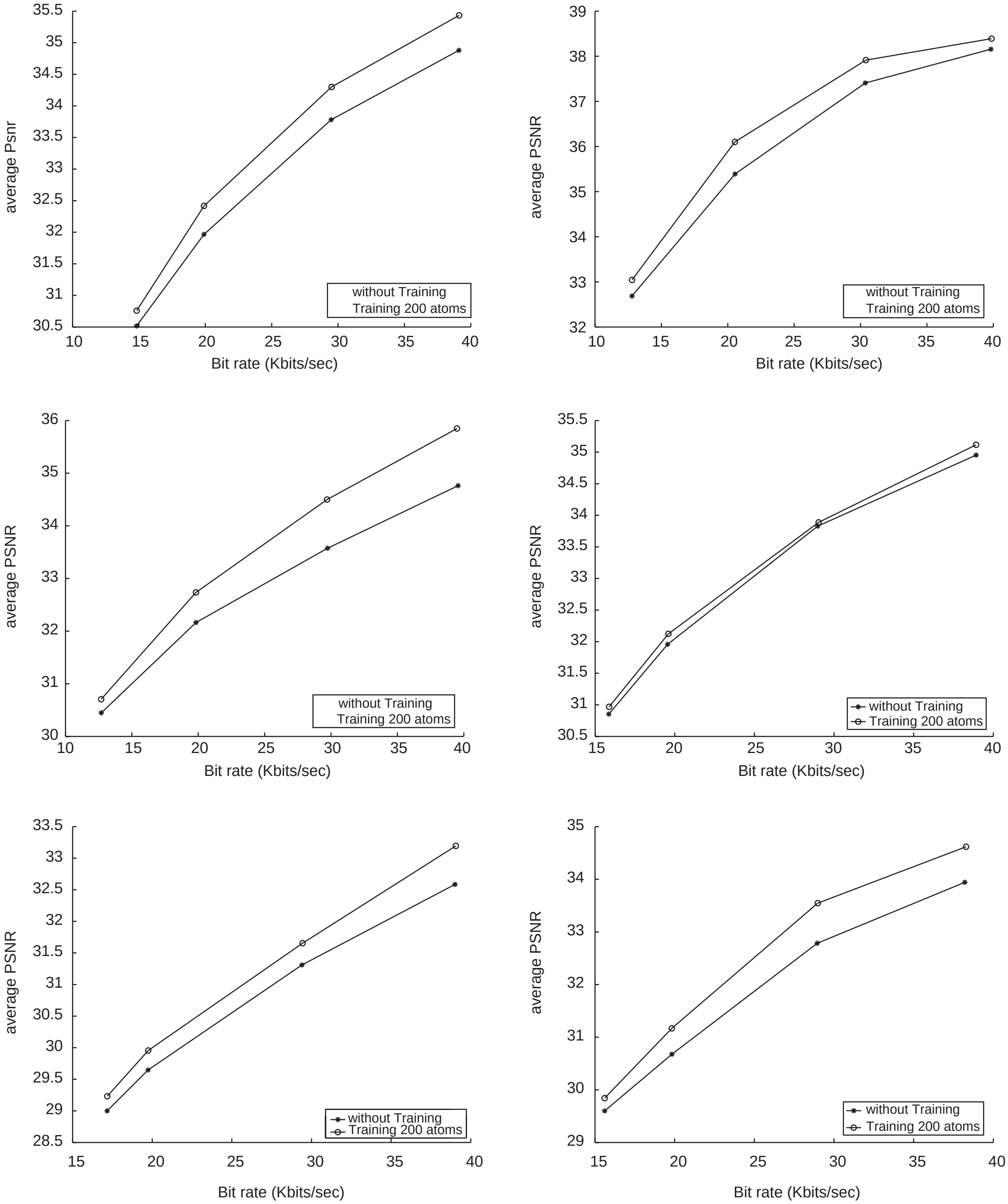

Fig. 4. PSNR comparisons with different dictionaries at low bit rates for testing sequences. Top Left: Average PSNR performance of all the test sequences. Top Right: Claire. Middle Left: Container. Middle Right: Mother and Daughter. Bottom Left: Salesman. Bottom Right: Sean.

at http://www.nta.no/brukere/DVC/. Also, our rate control for a testing sequence is synchronized with H.263 such that our MP codec consumes approximately the same number of bits in encoding each motion residual frame as does that in H.263. Atoms are selected according to the method proposed in [9]. An atom is encoded in our MP as follows: 16 bits for the locations of a basis, fixed Huffman codings for the horizontal and vertical indexes, respectively, and the adaptive arithmetic code for the quantized inner product values. The initial shape codebooks are those in the 2D separable Gabor dictionary given in [9]. Here, the parameter $\alpha=(s,\xi,\phi)$ is a triplet consisting, respectively, of a positive scale, a modulation frequency, and a phase shift. The constant K_α is chosen such that the following 1D Gabor sequence is of unit norm

$$g_\alpha(i)=K_\alpha g\left(\frac{i-N/2+1}{s}\right)\times\cos\left(\frac{2\pi\xi(i-N/2+1)}{16}+\phi\right)\quad i\in\{0,1,\ldots,N-1\}.$$

Each shape codebook has $20(=N_s)$ codewords. Totally, we have 400 basis functions. The initial gain codebook is a uniform scalar quantizer of size $10(=N_g)$. The results of the figure show that the dictionary trained by selecting 200 atoms from each residual frame in all the training videos gives the best PSNR performances for our testing sequences. According to the experimental results, this dictionary is used for our further performance evaluation on the efficiency of our method at various low bit rates.

Fig. 4 gives the PSNR performance comparisons at various low bit rates using our dictionary with the dictionary proposed in [9]. All the curves in the figure were obtained by applying our MP codec with different dictionaries to various testing sequences encoded in QCIF format at 10 frame/s for the first 3 s. There are five testing videos: Container, Claire, Mother and Daughter, Salesman, and Sean. Among them, The videos Container and Claire are not used in the experiment for obtaining our dictionary and Container is not a head-and-shoulder-type video. As in our previous experiment, the algorithm in H.263 gives the motion vector estimations and the rate control of assigning bits to each residual frame. The parameters of the testing sequences are 10 frame/s for the first 3 s in QCIF format. The top left subfigure gives an averaged PSNR comparison of dictionary efficiency. All these curves indicate that the proposed VQ-based dictionary optimization method can indeed the PSNR performance of an MP codec at low bit rates. Table 1 gives a summary of the average PSNR of all the testing sequences of different dictionaries versus bit rates. According to the results given in the table, the average improvements of using our gain–shape optimized dictionary are between 0.25 and 0.5 dB when the bit rates are ranging from 15 to 40 kbit/s.

Table 1
Average PSNR with different dictionaries at various bit rates

Bit rate	14.8	19.9	29.4	39.1
PSNR initial dictionary	30.5175	31.9666	33.7810	34.8798
PSNR trained dictionary	30.7582	32.4164	34.3002	35.4331
Gain with trained	+0.2407	+0.4498	+0.5191	+0.5533

4. Conclusion

We have shown that the techniques used to design gain–shape product VQ codebooks can be used to optimize the dictionary of matching pursuit and improve the PSNR performance at low bit rates for an MP codec. Experimental results how that between 0.25 and 0.5 dB improvement was obtained for our video sequences when encoded with bit rates ranging from 15 to 40 kbit/s, using a gain–shape optimized dictionary than that of the initial dictionary.

Acknowledgements

We would like to thank Dr. Neff for discussion and also to Mr. Jian-Liang Lin for helping us prepare simulation data.

References

[1] O.K. Al-Shaykh, E. Miloslavsky, T. Nomura, R. Neff, A. Zakhor, Video compression using matching pursuits, IEEE

Trans. Circuit Systems Video Technol. 9 (1) (February 1999) 123–143.

[2] M.R. Banham, J.C. Braillean, A selective update approach to matching pursuits video coding, IEEE Trans. Circuit and Systems Video Technol. 7 (1) (February 1997) 119–129.

[3] Y.T. Chou, W.L. Hwang, C.L. Huang, Very low-bit video coding based on gain–shape VQ and matching pursuit, IEEE ICIP, Kobe, Japan, 1999, Vol. 2, pp. 76–80.

[4] J. Engan, S.O. Aase, J.H. Husoy, Designing frames for matching pursuit algorithms, IEEE ICASSP, Turkey, 1998, Vol. 3, pp. 1817–1820.

[5] A. Gersho, R.M. Gray, Vector Quantization and Signal Compression, Kluwer Academic Publishers, Dordrecht, 1992.

[6] J.L. Lin, W.L. Hwang, S.C. Pei, The fine-grained scalable video coding based on matching pursuits, IEEE ICIP, Rochester, NY, 2002, Vol. 2, pp. II-53–II-56.

[7] Y. Linde, A. Buzo, R.M. Gray, An algorithm for vector quantizer design, IEEE Trans. Comm. COM-28 (January 1980) 84–95.

[8] S. Mallat, Z. Zhang, Matching pursuits with time-frequency dictionaries, IEEE Trans. Signal Process. 41 (December 1993) 3397–3415.

[9] R. Neff, A. Zakhor, Very low bit-rate video coding based on matching pursuits, IEEE Trans. Circuit Systems Video Technol. 7 (1) (February 1997) 158–171.

[10] R. Neff, A. Zakhor, Dictionary approximation for matching pursuit coding, IEEE ICIP, Vancouver, Canada, 2000, Vol. 2, pp. 828–831.

[11] M.J. Sabin, R.M. Gray, Product code vector quantizers for waveform and voice coding, IEEE Trans. ASSP 32 (3) (June 1984) 474–488.

[12] K. Sayood, Introduction to Data Compression, 2nd Edition, Morgan Kaufmann, Los Altos, CA, 2000.

[13] P. Schmid-Saugeon, A. Zakhor, Learning dictionaries for matching pursuits based video coders, IEEE ICIP, Thessaloniki, Greece, 2001, Vol. 3, pp. 562–565.

[14] X. Tang, A. Zakhor, Matching pursuits multiple description coding for wireless video, IEEE ICIP, Thessaloniki, Greece, 2001, Vol. 1, pp. 926–931.

[15] M. Vetterli, T. Kalker, Matching pursuit for compression and application to motion compensated video coding, IEEE ICIP, Austin, TX, 1994, Vol. 1, pp. 725–729.

[16] C.D. Vleeschouwer, B. Macq, SNR scalability based on matching pursuits, IEEE Trans. Multimedia 2 (4) (2000) 198–208.