3D Microscopic image coding by finite-state vector quantization in an enhanced image pyramid

Yegang Tao[*a], W. Paul Cockshott[a]
[a]Dept. of Computing Science, Univ. of Glasgow / University Ave., Glasgow, UK G12 8QQ

ABSTRACT

A novel approach, based on a 3D difference pyramid structure with vector quantization error feedback, is proposed for microscopic volume image data compression. We have improved the coding performance relative to previous work. A finite-state vector quantizer (FSVQ) is introduced to exploit the correlation between neighbouring vectors to improve the coding efficiency. The effects of FSVQ in conjunction with thresholding are investigated. A distortion minimization algorithm selects both the setting of thresholds and size of state codebook. Experiments have been performed on data sets obtained by confocal laser scanning microscopy (CLSM) scans of human arteries. Results demonstrate that our new coding technique substantially improves the subjective and objective quality of the decompressed images over Moving Picture Expert Group (MPEG)-1 with more than 5dB gain. Compared to the state-of-the-art 3D volume coder 3D-Set Partition in Hierarchical Trees (SPIHT), our method also offers better coding performance with roughly 0.1 dB higher at high rate and more than 0.6 dB higher at very low bit rate.

Keywords: Confocal Laser Scanning Microscopy, 3D Microscopic Image Compression, 3D Difference Pyramid, Vector Quantization, Finite-State Vector Quantization

1. INTRODUCTION

We have used confocal laser scanning microscopy (CLSM; single photon microscopy)[2] to collect 3D volumetric data. Serial optical sections (x, y plane) are collected at specific intervals down through the axial plane (z-axis) to produce a 'stack' of optical planes which can be processed as a 3D volume. Processing, analysis and transfer of the resulting data volumes is time consuming and therefore a dedicated compression routine would be of great value for biomedical purpose.

Different approaches may be considered for volumetric microscopic image compression. Since lossless coding approaches result in a small compression ratio, we consider lossy compression techniques, which would be attractive for non-diagnostic purpose, like reviewing. Among the main approaches developed, we can note:

1. Methods, closed to the MPEG standard[4], using block-matching algorithm for motion estimation to code 2D image sequences[5, 6]. Such frame-by-frame algorithms are not adapted to the compression of real 3D medical images because of the artifacts introduced when quantizing the motion residual images along the third dimension which are not acceptable for static objects.
2. Methods using subband decomposition. Such a decomposition takes the advantage of the voxel correlation in the three directions as the volumes produced from CLSM devices are isotropic. More recently, a 3D zerotree quantizer has been associated to such a subband decomposition[7]. Both the zerotree methods, like 3D-EZW and its enhanced version, like 3D-SPIHT[8] present good results for 3D medical image coding.
3. Methods using 3D Discrete Cosine Transform applied to blocks[9]. 3D DCT-based techniques are not widely adopted, because unaccepted blocking artifacts distort the visual quality at low bit rate.

This work continues from the 'Advanced Compression for Microscopy Images' project, in which we presented a technique combining vector quantization (VQ) with a 3D difference image pyramid data structure and demonstrated good coding performance at compressing volumetric confocal microscopic images[1]. In this study, we improve the previous work by applying adaptive block-based splitting thresholding on pyramid-transformed coefficients and introducing the finite-state vector quantization scheme to improve performance over ordinary VQ by exploiting the

[*] tao@dcs.gla.ac,uk; tel 44 141 339-8855 ext. 0174; fax 44 141 330-3119; http://www.dcs.gla.ac.uk/~tao

correlation between neighboring vectors. Section 2 gives a brief review of pyramid decomposion and finite-state vector quantization for image coding. Section 3 describes the complete coding scheme of the enhanced 3D pyramid coder. Finally, in section 4, we present coding results on human arteries data set issued from CLSM devices using our approach compared with video coder MPEG-1 and the state-of-the-art image volume coder 3D-SPIHT.

2. BASIC IDEA OF USING VQ IN IMAGE PYRAMID STRUCTURES

2.1. Image pyramid

Image pyramids[10-16] provide a multi-resolution model of images and have been found attractive for compression, thanks to their good sub-band decomposition property, low computational cost and inherent characteristics of progressiveness. A difference pyramid (DP) is computed from an image pyramid (IP) to exploit the redundancy between each level of image pyramid, which is generated by taking the difference between the IP from the same level and the upsampled version of the IP from next above. A difference pyramid provides more efficient representation for transmission when combined with entropy coding. Figure 1 illustrates the construction of the image pyramid and the difference pyramid, and the procedure can be formulated as:

$$IP_i = \begin{cases} original\ image, & i = 0\ (bottom\ level) \\ F_{reduce}(IP_{i-1}), & i = 1, \dots, N-1 \end{cases} \qquad (1.1),$$

$$DP_i = \begin{cases} IP_i, & i = N-1\ (top\ level) \\ IP_i - F_{expand}(IP_{i+1}), & i = 0, \dots, N-2 \end{cases} \qquad (1.2),$$

where N is the number of pyramid level.

Generally, F_{reduce} represents a decimation process, where an image is first lowpass filtered by a 2-D decimation filter, then subsampled by a factor of two horizontally and vertically. F_{expand} represents an interpolation process, where we first upsample a decimated image by a factor of two horizontally and vertically, and then perform lowpass filtering of the upsampled image using an interpolation filter.

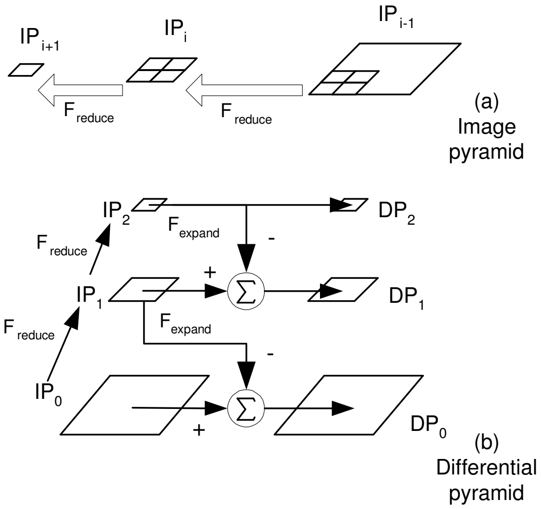


Figure 1. (a) Image pyramid and (b) difference pyramid construction.

With all-unit quantization step sizes, the image pyramid can be perfectly reconstructed, warranting a lossless reconstruction[13-15]. In this case, nonexpensive pyramids, like reduced-difference pyramid, are more efficient in terms of overall rate for lossless coding compared to expansive versions, like difference pyramid, since the latter have larger number of coefficients to be encoded. When we choose larger step size or vector quantization (VQ) to encode the difference pyramid, the quantization errors will be taken into account. The effects of quantization noise on the entropy of Laplacian pyramids was firstly investigated in Burt's Laplacian pyramid[10], in which the overall distortion is shared between the various pyramid layers. The optimum rate and distortion allocation scheme of the Laplacian pyramid was discussed in[11]. Such techniques, however, can be used only for irreversible coding schemes. Therefore, schemes with quantization noise feedback have been introduced into pyramid structures when we compute the difference pyramids. They demonstrate better coding performance compared to the approaches without feedback[16, 12].

2.2. Finite-state VQ

A memoryless VQ exploits the statistical redundancy between pixels within a vector[20, 21]. But each vector is independently encoded without using the high statistical correlation among its neighbouring vectors. Finite-state vector quantization (FSVQ) is designed for exploiting vector-to-vector correlation within the image to reduce the bit rate[20-22]. FSVQ selects a smaller state codebook from a large master codebook by the current encoder's state. Hence, a FSVQ can maintain the image quality of a large VQ codebook and achieve the bit rate efficiency of a small VQ codebook.

A FSVQ (see Fig. 2) uses the previously encoded blocks to establish a uniquely defined state. For each state s_i, we select N_f codewords from the master codebook Y as the s_i's *state codebook* SC_s. For encoding an input vector x, the encoder finds the current state s and then searches the state codebook SC_s, not the master codebook Y, to find the best-matched codeword to x. The decoder will find the same current state s and the corresponding codeword in the same state codebook SC_s according to the reproduction vector(s). In general, size of SC_s is much smaller than that of Y. Hence, the FSVQ uses a large codebook for improving the image quality and uses the encoder state to select a smaller codebook for reducing the bit rate and the encoding time. The selected codeword may not be equal to the closest codeword that is selected by full searching the master codebook in which the closest codeword is found.

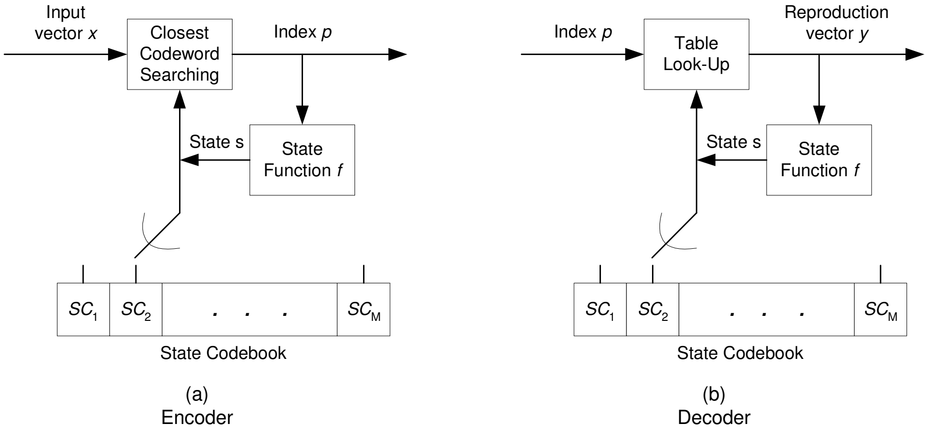


Figure 2. Finite-State VQ (a) encoder and (b) decoder.

3. ENHANCED IMAGE PYRAMID CODER

The 3D version of our algorithm is used to compress the sequence of slices obtained from CLSM devices. Unlike other 3D image data, e.g. video sequence, each frame in the CLSM sequence presents one slice of an object at specific depth. The 3D pyramid coder treats the whole sequence as a 3D volume data and exploits the multi-dimensional redundancy with a single procedure.

In Figure 3, we gave an example of constructing a four-level 3D pyramid. We incorporate block-based splitting thresholding scheme and FSVQ into the pyramid structures with vector quantization noise feedback. . In this case, the ith level of the difference pyramid is generated by taking the difference between the i^{th} level of the original image pyramid and the i+1th level of the reconstructed version of the image pyramid. Therefore, errors introduced by the thresholding and quantization can be recovered at higher-resolution pyramid layers. We generalize the F_{reduce} and F_{expand}, previously used on 2D image data, to 3D voxel data. T_{bb} refers to block-based splitting thresholding. We can formulate the construction of 3D pyramid as follows (refer to Formula (1)):

$$IP_i = \begin{cases} original\ image, & i = 0 \quad (bottom) \\ F_{reduce}(IP_{i-1}), & i = 1, \ldots, N-1 \end{cases} \qquad (2.1),$$

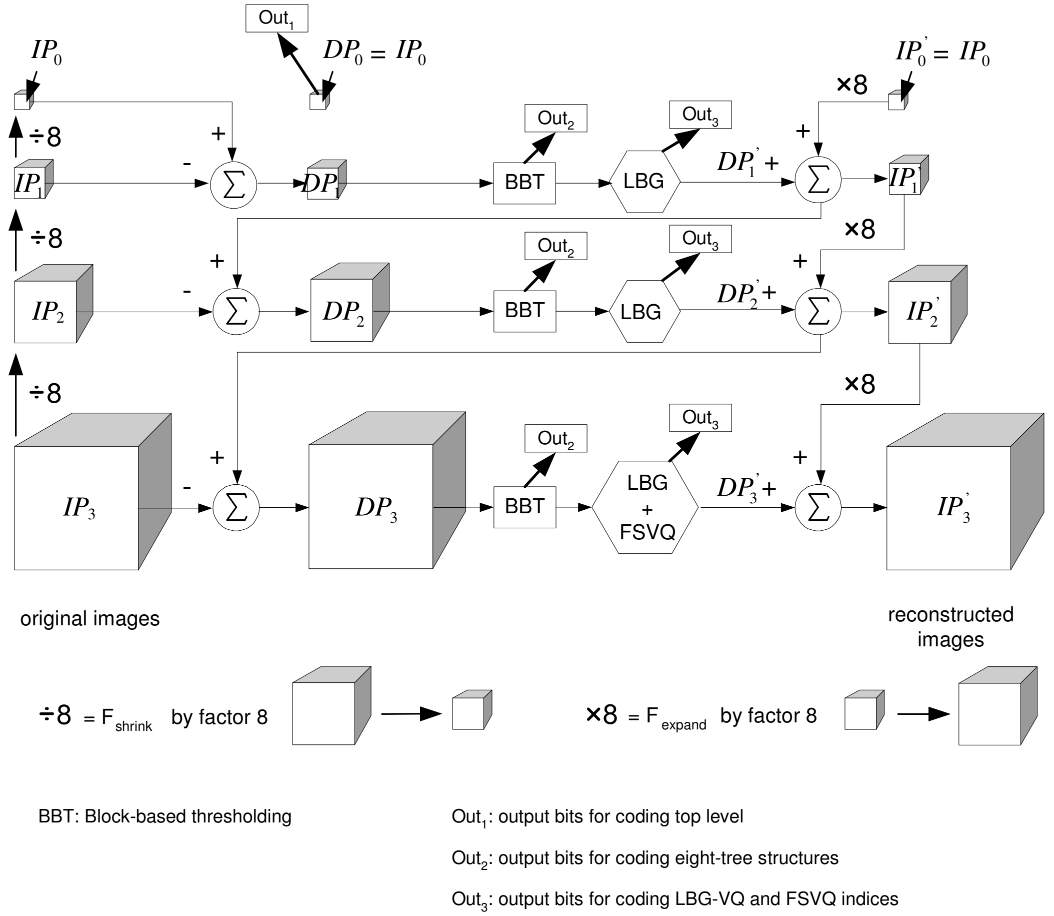
$$DP_i^{'} = VQ_i^{-1}(VQ_i(T_{bb}(DP_i))), \qquad i = 0, 1, \ldots, N-2 \qquad (2.2),$$

$$IP_i^{'} = \begin{cases} IP_i, & i = N-1 \quad (top) \\ DP_i^{'} + F_{expand}(IP_{i+1}^{'}), & i = 0, \ldots, N-2 \end{cases} \qquad (2.3),$$

$$DP_i = \begin{cases} IP_i, & i = N-1 \\ IP_i - F_{expand}(IP_{i+1}^{'}), & i = 0, \ldots, N-2 \end{cases} \qquad (2.4),$$

where N is the number of pyramid level.

Now we will describe the approach combining thresholding, FSVQ and distortion minimization scheme on the pyramid-transformed coefficients to improved rate-distortion performance over the previously proposed 3D image pyramid coder[1].



Out$_1$
IP_0
$DP_0 = IP_0$
$IP_0^{'} = IP_0$
×8
÷8
+
-
IP_1
DP_1
Out$_2$
Out$_3$
BBT
LBG
$DP_1^{'}$+
$IP_1^{'}$
÷8
×8
IP_2
DP_2
$DP_2^{'}$+
$IP_2^{'}$
÷8
×8
IP_3
DP_3
LBG + FSVQ
$DP_3^{'}$+
$IP_3^{'}$
original images
reconstructed images
÷8 = F$_{shrink}$ by factor 8
×8 = F$_{expand}$ by factor 8
BBT: Block-based thresholding
Out$_1$: output bits for coding top level
Out$_2$: output bits for coding eight-tree structures
Out$_3$: output bits for coding LBG-VQ and FSVQ indices

Figure 3. Enhanced 3D image pyramid coder with the introduction of block-based splitting thresholding and finite-state vector quantization applied on bottom layer.

3.1. Block-based splitting thresholding

When thresholding the pyramid-transformed coefficients, a block-based splitting technique is applied on each level of the difference pyramid to discard a large part of insignificant coefficients and isolate smaller entities, containing the significant information. Such thresholding technique is based on three important observations from the properties of pyramid-transformed coefficients:

1. Difference images in pyramid have the characteristics that most of the coefficients energies are concentrated around zero.
2. The coefficients with larger magnitude are more important than smaller magnitude coefficients (low energy) because they contribute more to the decreasing of distortion after receiving by decoder.
3. When an image is pyramid decomposed the intra-band variations will be smaller on lower pyramid level than on the higher pyramid level.

A block will be regarded as significant if it has one or more coefficients whose magnitudes exceed a threshold *T*. The significant block will be split into eight sub-blocks and the same significant test will be applied on each sub-block until the sub-block's size reaches the predefined smallest sub-block's size. Such a block-based method can capture and separate the interesting objects from the uninteresting background, which is more suitable for images of medical purpose than the previously proposed thresholding method[1], which applies thresholding on VQ output indices, because a vector containing one significant coefficient might be discarded with a threshold by regarding it as a low energy vector. In Fig. 4, the initial size of the block is 16 × 16 × 16 and the smallest sub-cubes have the size of 4 × 4 × 4. We will refer to these smallest sub-cubes as the leaf nodes.

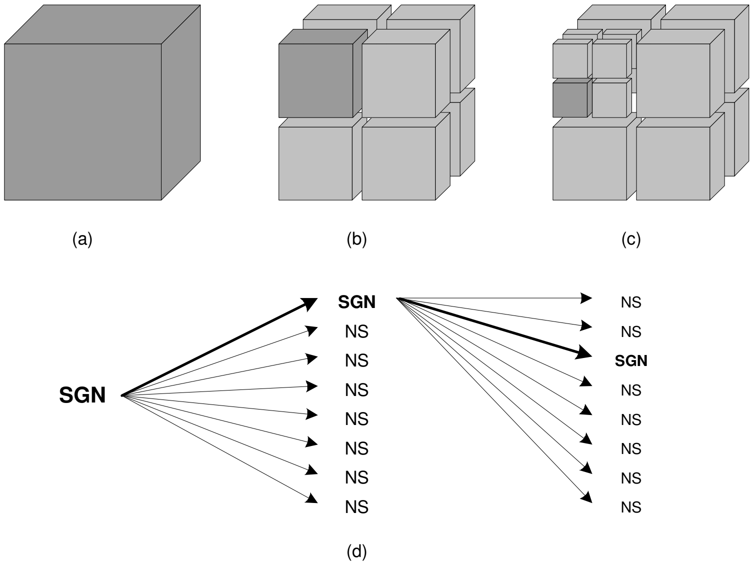


Figure 4. When a significant pyramid coefficient is encountered, the cube (a) is spited into eight sub-cubes (b), and further on (c) up to the leaf node level (smallest sub-cube). The result is an eight-tree structure (d) (SGN = significant node; NS = non-significant node).

3.2. Adaptive vector quantization

We use vector quantizers to exploit the correlations between coefficients within the leaf nodes[17, 19]. We use small size vectors, for instance 2 × 2 × 2, to avoid the irritating blocky artefacts introduced by large size vectors. Since each level of difference signal contains the details of signal at different scales, we hypothesise these signals will have different frequency characteristics. We use different codebooks to quantize the coefficients from each layer. Each codebook is generated using LBG (a codebook designing algorithm proposed by Linde, Buzo and Gray) algorithm[18]. Such VQ is adaptive because we determine the shape of the vector according to the variations σ_x, σ_y and σ_z of distribution of one layer of the difference pyramid along x-y planer (x and y) directions and depth (z) direction, respectively. Suppose the l-th (l=N-2, …, 0) layer of the difference pyramid is of X, Y and Z in size, we can define the variations along three directions as:

$$\begin{cases} \sigma_x = \dfrac{1}{(X-1)YZ}\displaystyle\sum_{k=1}^{Z}\sum_{j=1}^{Y}\sum_{i=1}^{X}(c_{i,j,k}-\hat{c}_{j,k})^2, \quad \hat{c}_{j,k} = \dfrac{1}{X}\displaystyle\sum_{i=1}^{X}c_{i,j,k} \\ \sigma_y = \dfrac{1}{X(Y-1)Z}\displaystyle\sum_{k=1}^{Z}\sum_{i=1}^{X}\sum_{j=1}^{Y}(c_{i,j,k}-\hat{c}_{i,k})^2, \quad \hat{c}_{i,k} = \dfrac{1}{Y}\displaystyle\sum_{j=1}^{Y}c_{i,j,k} \\ \sigma_z = \dfrac{1}{XY(Z-1)}\displaystyle\sum_{j=1}^{Y}\sum_{i=1}^{X}\sum_{k=1}^{Z}(c_{i,j,k}-\hat{c}_{i,j})^2, \quad \hat{c}_{i,j} = \dfrac{1}{Z}\displaystyle\sum_{k=1}^{Z}c_{i,j,k} \end{cases} \qquad (3).$$

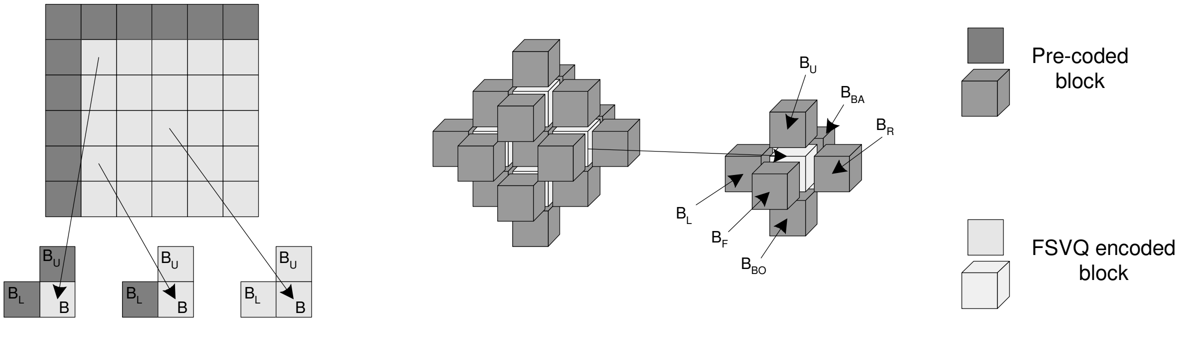
If the distribution has low variations along a certain direction, we can expect to achieve better rate-distortion performance of VQ by increase the size of the vector along this direction.

3.3. Surface-match FSVQ

We use side match VQ[22], which is a kind of FSVQ which exploits contiguity across block boundaries to establish the state. In 2D images two-sided side match is commonly used because blocks are coded in the raster-scan order. The side information of upper B_U, and left B_L, neighbouring blocks produce the state codebook for each input vector B (see Fig. 5.a). This kind of prediction may be poor if the correlation among B_U, B_L, and B is not high enough. We solve this problem by using all the neighbouring sides of an input block, which in 3D case it will be six surfaces from six neighbouring blocks as illustrated in Fig. 5.b). Here, 50% of input blocks will be precoded before using all adjacent surfaces to guide the match. Let the size of blocks be $X\times Y\times Z$. We present the measurement of surface match distortion $d_{sm}(y)$ of a codeword y in super codebook as the similarity of the outer surfaces of y to the inner surfaces forming by its six adjacent blocks. The suface match distortion $d_{sm}(y)$ is defined to be the summation of the surface match distortions d_U from upper block B_U, d_{BO} from bottom block B_{BO}, d_L from left block B_L, d_R from right block B_R, d_F from front block B_F and d_{BA} from back block B_{BA}, as

$$\begin{aligned} d_{sm}(y) &= d_U(y)+d_{BO}(y)+d_L(y)+d_R(y)+d_F(y)+d_{BA}(y) \\ &= \sum_{k=1}^{Z}\sum_{i=1}^{X}(B_U(i,Y,k)-y(i,1,k))^2+\sum_{k=1}^{Z}\sum_{i=1}^{X}(B_{BO}(i,1,k)-y(i,Y,k))^2 \\ &+\sum_{k=1}^{Z}\sum_{j=1}^{Y}(B_L(X,j,k)-y(1,j,k))^2+\sum_{k=1}^{Z}\sum_{j=1}^{Y}(B_R(1,j,k)-y(X,j,k))^2 \\ &+\sum_{j=1}^{Y}\sum_{i=1}^{X}(B_F(i,j,Z)-y(i,j,1))^2+\sum_{j=1}^{Y}\sum_{i=1}^{X}(B_{BA}(i,j,1)-y(i,j,Z))^2 \end{aligned} \qquad (4).$$

The state codebook SC_s of state s consists of N_f codewords with the N_f smallest surface-match distortions that are selected in the super codebook C.



(a)

(b)

Figure 5. (a) Two-sided side match in 2D case, where side match distortion is determined by the upper block B_U and left block B_L of current block B; (b) six-surface surface match in 3D case, where surface match distortion is determined by all the adjacent blocks: B_U, B_{BO}, B_L, B_R, B_F, B_{BA}.

In our method, FSVQ is applied on the pyramid coefficients on bottom level only. This is because firstly, the bottom pyramid level contains the majority of pyramid coefficients. Secondly, we noticed that when applied to higher pyramid levels, FSVQ performs worse than when applied to lower pyramid levels, because the variations of coefficients increase with pyramid level, and the inter-vector correlations will not be strong for data with high-variation distribution. Thirdly, as we introduced error feedback in our pyramid structure, the poor performance of FSVQ on higher pyramid levels will introduce more errors and increase the variation of coefficients on lower levels. Therefore, applying FSVQ to the bottom level is preferable to applying it on all levels.

3.4. Distortion minimization algorithm

We observed, however, that at a given rate, different settings of thresholds and state codebook size give different *mean squared errors* (MSEs) distortion *D*. Our task is to select the thresholds and size of state codebook to optimise the rate-distortion performance under the constraint of a total number of coding bits *R* lower than a coding bit budget R_b. This is a minimization problem under a constraint given by $\underset{R\leq R_b}{Min(D)}$.

We illustrate the procedure of such minimization on Fig. 6. We compute the (R^i_j, D^i_j) with i^{th} thresholding scheme and j^{th} state codebook size. Suppose the size of master codebook is 2^N, the state codebook size can be chosen from the set $\{2^j \mid j=1, 2, \ldots, N\text{-}1)$. The thresholding values are selected with two constraints:

- According to the observations described in Section 3.1, the lower the pyramid level *l*, the larger is the thresholding value: $Th_l > Th_{l+1}$.
- Given a thresholding scheme {Th}, if R with 2^{N-1} state codebook size is smaller than R_b, we will regard this thresholds as 'over-thresholded', and discard all the thresholding shemes which have the same and larger thresholds than {Th}; If R with 2^1 state codebook size is larger than R_b, we will regard this thresholds as 'less-thresholded', and discard all the thresholding shemes which have the same and smaller thresholds than {Th}.

We compute all the possible (*R*, *D*) points, coming from all the combinations of (R^i_j, D^i_j). The optimal solution $(R_{\text{opt}}, D_{\text{opt}})$ is on the convex hull of the cluster of these points. With the constraint $R \leq R_b$, we use a Lagrangian cost function *J* which merges the rate and the distortion in the Lagrangian multiplier λ, and define *J* as: $J = D + \lambda R$ to obtain $(R_{\text{opt}}, D_{\text{opt}})$ by searching λ_{opt}.

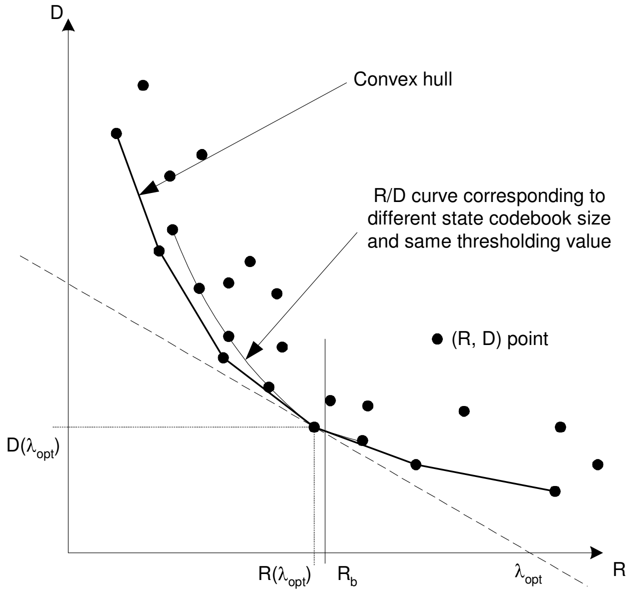

D
Convex hull
R/D curve corresponding to different state codebook size and same thresholding value
● (R, D) point
D(λopt)
R(λopt)
Rb
λopt
R

Figure 6. The illustration of *R/D* optimisation method: we compute all the possible (*R*, *D*) points, coming from all the combinations of (R^i_j, D^i_j) with i^{th} thresholding scheme from $\{T_i\}$ and j^{th} state codebook size from $\{N_j\}$. The optimal solution $(R_{\text{opt}}, D_{\text{opt}})$ is on the convex hull of the cluster of these points.

4. RESULTS

The coding experiments of the proposed coder are evaluated on a set of microscopic volumetric data obtained with CLSM device. This set of stacks describes the cellular organization through the vascular wall of small segments of human arteries, which are labelled as G××HG70[3]. Briefly, arterial segments are stained with fluorescent markers for the cell nucleus (propidium iodide 1μg/ml) Tissues are then slide mounted on the stage of a NORAN[1] (nuclear work) CLSM. Serial optical sections (x, y plane) represented as 2D images are collected at interval of 1μm down through the axial plane (z-axis).

We built a 4-level 3D pyramid (L=4) for each image stack of size X×Y×Z as described in Figure 3. The top layer l_3 was encoded using LZ entropy coder. The coefficients from bottom layer l_0 were encoded as 12-D intra-band vectors and other layers (l_i, i=1, 2) were encoded as 6-D intra-band vectors. The codebook for each layer had 256 codewords trained by the LBG algorithm. We choose from six kinds of vector-forming strategies for a 6-D vector and three kinds for a 12-D vector as shown in Fig. 7 (a) and (b), respectively. An automatic pre-analysis of variations is used to determine which strategy will be preferable. Given the l-th (l=0, 1, 2) layer of the difference pyramid, we computed the variations of the probability distributions of coefficients along x, y and z directions using (3) and the shape of the vector is determined by σ_x, σ_y and σ_z, such that the lower the variation along a direction, the larger the size of vector along this direction. For example, suppose at level l_2, we have $\sigma_x \geq \sigma_y \geq \sigma_z$, we will use strategy 6, having largest size along z direction and smallest size along x direction, to quantize this level.

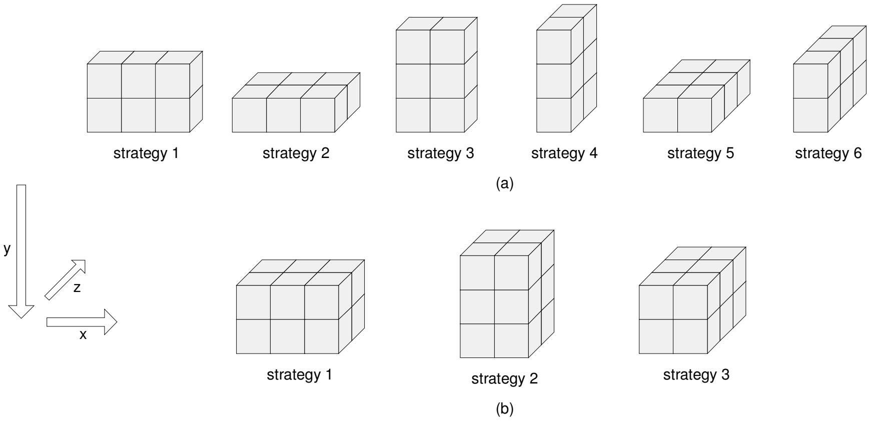


Figure 7. (a) Six kinds of 6-D vector-forming strategies and (b) three kinds of 12-D vector-forming strategies.

The peak signal-to-noise ratio (PSNR) is measured on two stacks 'G25HG70' (256×256×144×8bits) and 'G27HG70' (256×256×96×8bits) from data sets obtained from human arteries described above at six different bit-rates: 0.25, 0.2, 0.16, 0.1, 0.08 and 0.04 bits-per-pixel (bpp), which correspond to 32:1, 40:1, 50:1, 80:1, 100:1 and 200:1 compression ratios. The PSNRs and the corresponding settings of thresholds and FSVQ at each bit-rate are listed in Table I. The settings of thresholds and the size of state codebook have been optimised by optimisation process described in Section 3.4.

Table I. Coding results of our method and the settings of thresholds and FSVQ at each bit-rate.

G25HG70				G27HG70			
Bit-rate (bpp)	PSNR (dB)	Thresholds (l_0, l_1, l_2, l_3)	State codebook size	Bit-rate* (bpp)	PSNR (dB)	Thresholds (l_0, l_1, l_2, l_3)	State codebook size
0.2556	44.5992	(6, 2, 2, 0)	64	0.2463	43.0030	(8, 3, 2, 0)	128
0.1959	43.9455	(7, 3, 2, 0)	32	0.2038	42.4830	(9, 4, 2, 0)	64
0.1572	43.4897	(9, 3, 2, 0)	64	0.1603	41.9783	(11, 4, 2, 0)	64
0.1003	42.5215	(11, 4, 2, 0)	64	0.1015	41.0920	(14, 5, 2, 0)	8
0.0804	42.0845	(13, 5, 3, 0)	16	0.0800	40.5621	(17, 6, 3, 0)	64
0.0400	40.6767	(22, 10, 4, 0)	64	0.0397	38.9130	(24, 14, 5, 0)	128

* Sizes of codebooks were considered when computing the rate.

[1] NORAN (Formerly Noran Instruments - No longer in business)

We compared the new 3D pyramid method (3D-PYR) with standard video coder MPEG-1[2] and the state-of-the-art 3D image coder 3D-SPIHT. The PSNRs for each coder are calculated at almost the same bit-rates on G25HG70 and G27HG70 respectively. The comparison results are illustrated in Fig. 8. We noticed that for these two stacks 3D-PYR outperforms the other coders at all rates. At higher rates above 0.1 bpp, the difference between 3D-PYR and 3D-SPIHT is not significant. For G27HG70, for example, the difference is of 0.12 dB and 0.10 dB at 0.16 bpp and 0.25 bpp respectively. However, at lower rates, like 0.04 bpp, the 3D-PYR yields a PSNR of 38.91 dB, which is 0.63 dB higher than 38.28 dB provided by the 3D-SPIHT. For G25HG70, the difference is not as significant as shown on G27HG70. This is because our surface-matching strategy applied on FSVQ is not optimal for stacks having low-variation distribution. A more suitable surface-matching scheme for 3D FSVQ will be investigated. Both methods have a superior quality compared to MPEG-1 with up to 6dB higher at low rate. This is because in 3D microscopy the raw data is digitised as a sequence of 2D images, but inherently the data is 3 dimensional. This contrasts with the data in a movie sequence, where the physical process is 3D surfaces moving in time, which are then projected onto the 2D image plane of the camera. Because of this, the higher order statistics of 3D microscopy data will differ from those of film. In particular as the block-matching process used in MPEG-1 depends upon finding similar luminance values, this can be confused by objects moving into shade or fades.

Fig. 9 illustrates the visual performance of 3D-PYR, 3D-SPIHT, MPEG-1 for 45th slice taken from G27HG70. For MPEG-1, blocking artifacts distort the visual quality because of the block-based DCT technique. It is difficult to distinguish a visual quality difference between 3D-PYR and 3D-SPIHT.

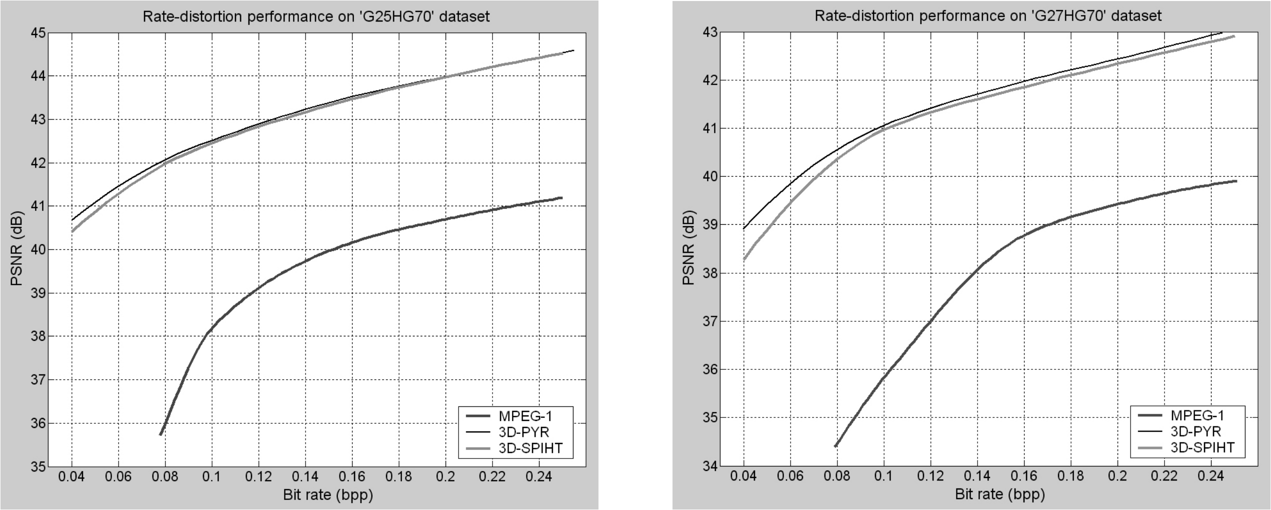
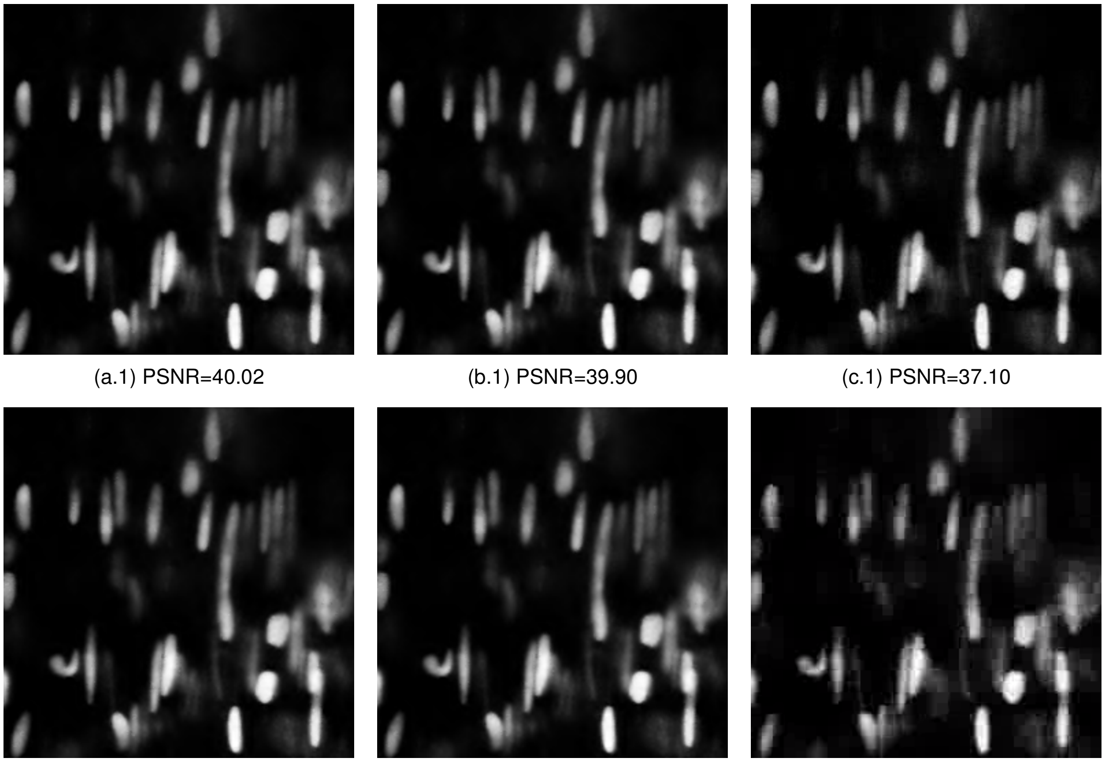


Figure 8. Coding results on G25HG70 (left) and G27HG70 (right) with 3D-PYR, 3D-SPIHT and MPEG-1.

5. CONCLUSION

In this study, we propose a new 3D pyramid coder by introducing FSVQ into the difference pyramids, which is an enhanced version of the previous work[1]. It is effective for microscopic volumetric data compression, especially for volumes having low spatial variations. Results show that our method using 3D difference pyramid structures combined with VQ techniques yields as good as or better rate-distortion performance than 3D wavelet coder and standard video coder. They also demonstrate the effectiveness in preventing visible artifacts, e.g., blocking and ringing, from being introduced into textured and homogeneous areas, even at very low rates. Moreover, we noticed the 3D techniques are suitable for coding volumetric images of high spatial resolution along the slice axis, while frame-by-frame compression schemes don't work well to take the advantage of such spatial correlations.

[2] In our program, the MPEG-1 coder cannot achieve the rate as low as 0.04 bpp.



(a.1) PSNR=40.02

(b.1) PSNR=39.90

(c.1) PSNR=37.10

(a.2) PSNR=37.73

(b.2) PSNR=37.46

(c.2) PSNR=30.84

Figure 9. G27HG70 compressed at (1) 0.2 bpp and (2) 0.08 bpp using (a) 3D-PYR, (b) 3D-SPIHT, (c) MPEG-1. Slice 45 is depicted.

REFERENCES

1. W. P. Cockshott, Y. Tao, G. Gao, P. Balch, A. M. Briones and C. Daly, "Confocal Microscopic Image Sequence Compression using Vector Quantization and 3D Pyramid", *SCANNING*, **25**, 247-256, 2003.

2. J. B. Pawley, *Handbook of Biological Confocal Microscopy (2nd Edition)*, Plenum Press, New York, 1995.

3. C. Daly, A. McGee, E. Vila, A. M. Briones, J. Giraldo, S. Arribas, C. González, J. M. González, B. Somoza, S. N. Pagakis, J. Adler, J. C. Provost, A. Merle, J. Maddison, J. Pederson, and J. C. McGrath, "Analysing the 3D Structure of Blood Vessels using Confocal Microscopy", Microscopy and Analysis (UK), 92, 5-8, 2002.

4. D. L. Gall, "The MPEG Video Compression Algorithm", *Signal Processing: Image Communication*, **4**, 129-140, 1992.

5. B. K. T. Ho, P. Saipetch, J. Wei, M. Ma, J. Villansenor and M-J. Tsai, "Video Compression Algorithm for Dynamic Angiographic Images" *Proc. SPIE Medical Imaging*, 2164, 302-309, Newport Beach, California, 1994.

6. H. Lee, Y. Kim, A. H. Rowbey and E. A. Riskin, "Statistical Distributions of DCT Coefficients and Their Applications to An Interframe Compression Algorithm for 3D Medical Images", *IEEE Trans. on Medical Imaging*, **12**, 478-485, 1993.

7. J. Luo, X. Wang, C. W. Chen and K. J. Parker, "Volumetric Medical Image Compression with Three Dimensional Wavelet Transform and Octave Zerotree Coding", *Proc. SPIE Visual Communications and Image Processing*, 2727, 579-590, Orlando, Florida, 1996.

8. Y. Kim and W. A. Pearlman, "Lossless Volumetric Medical Image Compression", *Proc. SPIE on Applications of Digital Image Processing XXII*, 3808, 305-312, 1999.

9. G. P. Abousleman, M. W. Marcellin and B. R. Hunt, “Compression of Hyperspectral Imagery Using the 3D DCT and Hybrid DPCM/DCT”, *IEEE Trans. on Geosci. Remote Sensing*, **33**, 26-34, 1995.

10. P. J. Burt and E. H. Adelson, “The Laplacian Pyramid as a Compact Image Code”, *IEEE Trans. on Comm.*, **31**, 532-540, 1983.

11. J. M. Salinas and R. L. Baker, “Laplacian Pyramid Encoding: Optimum Rate and Distortion Allocations”, *Proc. of International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 1957-1960. 1989.

12. L. Wang and M. Goldberg, “Progressive Image Transmission Using Vector Quantization on Images in Pyramid Form”, *IEEE Trans. on Comm.*, **37**, 1339-1349, 1989.

13. L. Wang and M. Goldberg, “Comparative Performance of Pyramid Data Structures for Progressive Image Transmission”, *IEEE Trans. on Comm.*, **39**, 540-548, 1991.

14. D. Houlding and J. Vaisey, “Low Entropy Image Pyramids for Efficient Lossless Coding”, *IEEE Trans. on Image Processing*, **4**, 1150-1153, 1995.

15. B. Aiazzi, L. Alparone and S. Baronti, “A Reduced Laplacian Pyramid for Lossless and Progressive Image Communication”, *IEEE Trans. on Comm.*, 44, 18-22, 1996.

16. B. Aiazzi, L. Alparone, S. Baronti and F. Lotti, “Lossless Image Compression by Quantization Feedback in a Content-Driven Enhanced Laplacian Pyramid”, *IEEE Trans. on Image Processing*, **6**, 831-843, 1997.

17. R. M. Gray, D. L. Neuhoff, “Quantization”, *IEEE Trans. on Inform. Theory*, **44**, 2325-2383, 1998.

18. Y. Linde, A. Buzo and R. M. Gray, “An Algorithm for Vector Quantizer Design”, *IEEE Trans. on Comm.*, **28**, 84-95, 1980.

19. N. M. Nasrabadi and R. A. King, “Image Coding Using Vector Quantization: A Review”, *IEEE Trans. on Comm.*, **36**, 957-970, 1988.

20. J. Foster, R. M. Gray and M. O. Dunham, “Finite-state Vector Quantization for Waveform Coding”, *IEEE Trans. on Inform. Theory*, **31**, 348-359, 1985.

21. R. Aravind and A. Gersho, “Image Compression based on Vector Quantization with Finite Memory”, *Optical Engineering*, **26**, 570-580, 1987.

22. T. Kim, “Side match and overlap match vector quantizers for images”, *IEEE Trans. on Image Processing*, **1**, 170-185, 1992.