MICROSCOPIC VOLUMETRIC IMAGE DATA COMPRESSION USING VECTOR QUANTIZATION AND 3D PYRAMID

W. Paul Cockshott, Yegang Tao, Gang Gao

Department of Computing Science
University of Glasgow
Glasgow, Scotland
wpc, tao, gaog@dcs.gla.ac.uk

Craig Daly

Faculty of Biomedical & Life Sciences
University of Glasgow
Glasgow, Scotland
c.daly@bio.gla.ac.uk

ABSTRACT

The 3D pyramid compressor project at the University of Glasgow has developed a compressor for images obtained from CLSM device. The proposed method using a combination of image pyramid coder and vector quantization techniques has good performance at compressing confocal volume image data. An experiment was conducted on several kinds of CLSM data using the presented compressor compared to other well-known volume data compressors, such as MPEG. The results showed that the 3D pyramid compressor gave higher subjective and objective image quality and better minutia preservation on reconstructed images at the same compression ratio.

1. INTRODUCTION

The 3D pyramid compressor project at the University of Glasgow was funded by the Scottish Enterprise with a scheme of Proof of Concept Awards. The objective of this project is to provide a 3D compressor for confocal microscopic images. The basic concept of the 3D compressor is to read a stack of two-dimensional images, for example, a stack of microscopic images, sequentially into a three-dimensional array and compress the three-dimensional array. Here, we present a technique combining *vector quantization* (VQ) with a 3D differential image pyramid data structure for volume image data compression.

Confocal laser scanning microscopy (CLSM; single photon microscopy) has been available to biomedical scientists for almost 20 years [10]. However, it is only very recently that affordable computer power has enabled biologists to fully exploit the data contained within the large (> 100Mb) image volumes. In this study we have used CLSM to collect 3D volumetric data describing the cellular organization and receptor protein distribution through the vascular wall of small segments of human and rat arteries. Serial optical sections (x, y plane) are collected at intervals of 1um 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 robust non-lossy or low distortion lossy compression routine would be of great value for biomedical purpose, e.g. in studying vascular structure [11].

In 3D microscopy the raw data correspond to tracer densities at sub volumes in a 3D grid, with the size of the sub volumes constrained by the microscopy optics. The data is typically digitized as a sequence of 2D images, but this is an artificial presentation, inherently the data is 3 dimensional. This contrasts with the data in a movie sequence which is also captured as a sequence of 2D images, but in this case the generating physical process is 3D surfaces moving in time, which are then projected onto the 2D image plane of the camera. Because of this, we hypothesise that the higher order statistics of 3D microscopy data will differ from those of film. In particular planar motion normal to the camera axes - which motion compensation algorithms capture - has no corresponding generating physical process in microscopy. We thus hypothesise that the optimal compression strategy will differ from that used in film and video applications.

In this paper, we describe the use of 3D pyramid data structures to compress microscopy data. These exploit the inherent redundancy associated with correlation between tracer densities in 3 dimensions. We describe experimental results on several kinds vascular structural data. We compare the image quality with video coders currently in use. Finally, a brief conclusion is given in section V.

2. PREVIOUS WORK

2.1. Image Pyramid

Image pyramid data structures were originally developed for 2D image lossless coding. In this data structure, a *differential pyramid* (DP) is generated from an *image pyramid* (IP), which provides a multi-resolution model of image. In the image pyramid, an image is filtered producing a series of levels of images. The higher the level of image pyramid, the lower the resolution presented (see Fig. 1-a) [6, 7]. The scale factor for shrinking is usually 4. F_{shrink} and F_{expand} are two image scale transformation filters, where F_{shrink} decreases the image size and F_{expand} enlarges the image size. Many interpolation methods can be used for these two transformations, such as *nearest neighbor*, *bilinear* and *bi-cubic*. Image pyramid provides a reasonable solution for progressive transmission of images: the top level will be transmitted first to reconstruct the image with lowest resolution, and the following levels will refine the reconstructed image stage by stage.

The differential pyramid is computed from the image pyramid to exploit the redundancy between each level of the image pyramid, and provides more efficient representation for transmission when combined with entropy coding. Fig. 1-b illustrates the construction of the differential pyramid. Suppose a IP and a DP with N-levels, we can formulate the construction as follows:

$$IP_i = \begin{cases} original\ image & i = N-1(bottom) \\ F_{shrink}(IP_{i+1}) & i = N-2, \cdots, 0 \end{cases} \quad (1)$$

$$DP_i = \begin{cases} IP_i & i = 0(top) \\ IP_i - F_{expand}(IP_{i-1}) & i = 1, \cdots, N-1 \end{cases} \quad (2)$$

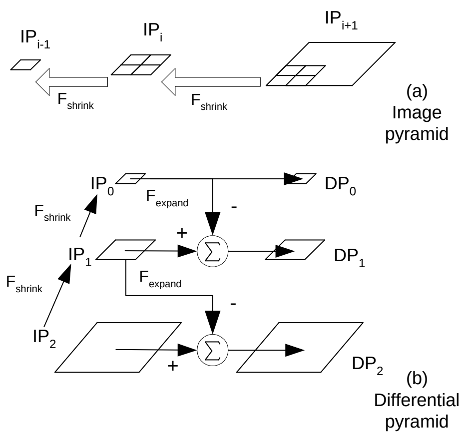


Fig. 1. Image pyramid and differential pyramid compositions.

Image pyramid is an redundant subband decomposition. That means the decomposed subbands need more storage requirement than the original image. Given L as the pyramid level, and s as the scale factor, number of pixels in pyramid will be

$$\begin{aligned} &NumOfPixels_{pyramid} \\ &= (1 + \tfrac{1}{s} + \tfrac{1}{s^2} + \cdots + \tfrac{1}{s^{L-1}})NumOfPixel_{original} \\ &= (1 + \tfrac{s^{L-1}-1}{s^{L-1}-s^{L-2}})NumOfPixel_{original} \end{aligned} \quad (3)$$

when compared with that of the original image. For instance, given $L = 5$ and $s = 4$, theres nearly $1/3$ extra pixels in pyramid. However, since the histogram of the DP, as in lossless DPCM, is highly peaked around zero, some advanced entropy coding can take advantages of this.

2.2. Burt's Pyramid Coder

In 1983, Burt and Adelson introduced quantization into the image pyramid structure and proposed a multi-resolution lossy compression technique [6]. Using a scalar quantizer, only the pixels with high energy are transmitted to the decoder side, and the entropy can be substantially reduced by quantizing the pixel values in each level of the differential pyramid. Fig. 2 illustrates the block diagram of Burt and Adelsons pyramid coder. Their method, introducing quantization into a pyramid structure has certain disadvantage. The quantization errors from upper levels would be magnified as they propagate down the pyramid during reconstruction. For example, an error affecting one pixel at the top of a three-level pyramid ends up corrupting sixteen pixels at the bottom layer. This disadvantage means Burt and Adelsons pyramid coding model doesn't give good results under high compression ratios, since increased errors are introduced when we set fewer quantization levels. In the next section, we extend the Burts pyramid by introducing vector quantization when constructing the differential pyramid with quantization noise feedback.

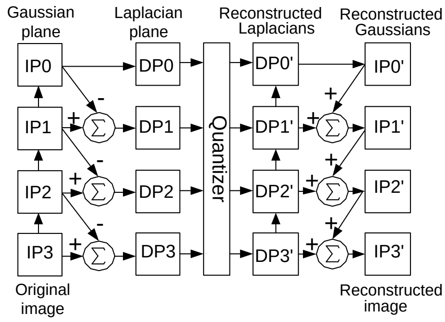


Fig. 2. Block diagram of Burt's pyramid coder.

3. PROPOSED 3D PYRAMID CODER

3.1. 3D Pyramid Structure

The 3D version of our algorithm is used to compress the sequence of slices obtained from CLSM device. Unlike other 3D image data, e.g. video sequence, each frame in the CLSM sequence presents one slice of an object at specific depth. That means inter-frame correlations between pixels are as strong as or even higher than intra-frame ones [10]. The 3D pyramid coder treats the whole sequence as a 3D volume data and exploits the multi-dimensional redundancy with only one procedure.

In Fig. 3, we gave an example of building a four-level 3D pyramid with VQ introduced. Wherein we generalize the F_{shrink} and F_{expand}, previously used on 2D image data, to 3D voxel data. Each voxel of level n maps to 8 voxels at level $n+1$. We can formulate the construction of 3D pyramid as follows (refer to Formula (1)-(2)):

$$IP_i = \begin{cases} original\ image & i = N-1(bottom) \\ F_{shrink}(IP_{i+1}) & i = N-2, \cdots, 0 \end{cases} \quad (4)$$

$$DP'_i = VQ^{-1}(VQ(DP_i)) \quad i = 1, \cdots, N-1 \quad (5)$$

$$IP'_i = \begin{cases} IP_i & i = 0(top) \\ DP'_i + F_{expand}(IP_{i-1}) & i = 1, \cdots, N-1 \end{cases} \quad (6)$$

$$DP_i = \begin{cases} IP_i & i = 0(top) \\ IP_i + F_{expand}(IP'_{i-1}) & i = 1, \cdots, N-1 \end{cases} \quad (7)$$

There are several advantages of 3D pyramid structures. Firstly, they organize sequential images as 3D volume data. This can capture the correlation in 3 rather than 2 spatial dimensions. Another advantage is that although the 3D pyramid structure, like the 2D pyramid is redundant, when F_{shrink} and F_{expand} are applied to 3D volume data, the scale factor in formula (3) would be 8, not 4. As a result the redundancy is only 1/7 rather than 1/3 for the 2D case.

3.2. Vector Quantizing Differential Pyramid

Vector quantization is an efficient technique for image compression [2, 3, 5]. It encodes a group of neighboring pixels together rather than individual pixel in scalar quantization. Since the neighboring pixels from an image are strongly correlated, according to Shannons rate-distortion theory [1], a better performance is achievable by coding vectors instead of scalars.

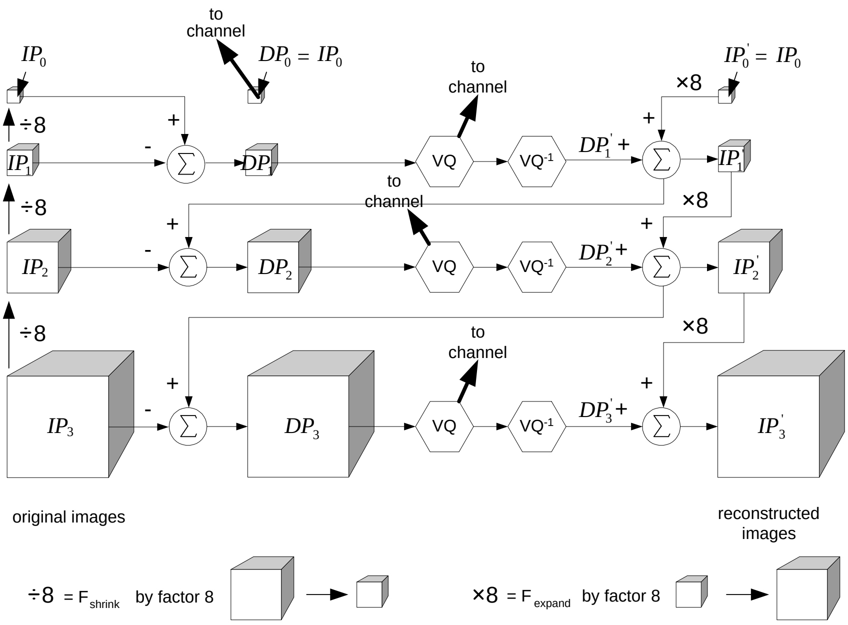


Fig. 3. Architecture of proposed compression technique using VQ with 3D image pyramid.

In a 3D pyramid, the output of the differential pyramid could either be scalar or vector quantized, we chose VQ rather than scalar quantization because of the higher compression ratio obtained at the same image quality. We use VQ to exploit the 3D correlations between voxels from intra-bands. The shape of a vector is specified by its *width*, *height* and *depth*. For instance, we construct an eight-dimensional vector $2 \times 2 \times 2$ $(w \times h \times d)$ by sampling four neighboring pixels from frame i and four neighboring pixels at the same position from its next frame $i + 1$. The shapes we typically use are $2 \times 2 \times 2$ or $4 \times 4 \times 2$, but the choice is programmable. In experiment, we observed that, for some image stacks like BxCGP/BxCGS, higher image quality can be achieved using vectors having larger size along the depth axis than that along planar axes, such that vectors of shape $2 \times 1 \times 4$ are preferable to those of shape $2 \times 2 \times 2$. This is based on the fact that for these stacks, correlation in depth direction is higher than in planar direction, for reasons pertaining to microscope optics.

3.3. Thresholding

Differential images in pyramid have the characteristics that most of the energy of pixels intensities is concentrated around zero[1] (see Fig. 4), we use a thresholding method to discard the low energy blocks to yield a data stream suitable for entropy encoding.

Assuming we have a codebook matrix M with n rows, and each row represents a code vector. When designing a codebook there will always be a vector in the codebook matrix with a minimum energy. In what follows we will assume that the codebooks are so designed that the minimum energy vector actually has zero energy. That is to say all elements are zero. We call this vector Z, and assuming it is the z-th row in codebook matrix. The thresholding algorithm scans each index i from the encoder and checks if $\mid M_i \mid^2 > T$ where T is some energy threshold. If the answer is positive, the i is transmitted to the entropy encoder. Otherwise, the index i is replaced by z, which is the index of Z, before being transmitted to the decoder. An entropy encoder such as LZ, or a Huffman codec is placed downstream from the vector quantizer so that the vector quantizer data stream is, in most cases, mapped to a shorter bit stream containing the same information content. A corresponding LZ or Huffman codec is used at the decoder side to reconstruct the VQ data stream.

Thresholding algorithm provides an efficient data format for an entropy encoder such as LZ, or Huffman. T is individual in each layer, usually the lower the pyramid layer, the larger the thresholding value.

[1]We define the zero energy of a pixels intensity as its intensity is of mid-grey, e.g., 128 for 256 grey levels.

4. EXPERIMENTAL RESULTS

We have performed coding experiments on two kinds of eight-bit CLSM image volumes captured by the *EC FP5* partnership for vascular imaging (VASCAN 2000). The first data sets are from rat mesenteric artery. Another data sets are from human resistance artery. Table 2 describes these data sets.

Table 2. Description of CLSM data

Filename	Description	Volume size (voxels)	Data size (bytes)
B1CGS	Rat mesenteric artery	$256 \times 256 \times 168$	11,303,376
B3CGP		$256 \times 256 \times 178$	11,976,196
B4CGS		$256 \times 256 \times 170$	11,445,476
B6CGS		$256 \times 256 \times 183$	12,320,285
G25_HG70	Human resistance arteries	$512 \times 512 \times 135$	35,523,360
G27_HG70		$512 \times 512 \times 89$	23,419,104

Table 1. Performance of 3D Pyramid Compressor on 6 data-sets

Filename	Original data size (bytes)	Output data size (bytes)	Compression ratio (C/R)	Ave. PSNR (dB)	3D Pyramid specifications				
					Pyramid levels	Entries in codebook	Vector shape $(w \times h \times d)$	VQ training algorithm	Thresholds (different between layers)
B1CGS	11,303,376	712,666	15.86	30.8155	4	256	$2 \times 2 \times 3$	LBG [4]	top - 0:0:1:4 - bottom
B3CGP	11,976,196	566,132	21.15	33.2712	4	256	$2 \times 2 \times 3$	LBG	0:0:2:7
B4CGS	11,445,476	608,267	18.82	33.5211	4	256	$2 \times 2 \times 3$	LBG	0:0:1:4
B6CGS	12,320,285	750,093	16.43	31.8014	4	256	$2 \times 2 \times 3$	LBG	0:0:1:4
G25_HG70	35,523,360	211,152	168.24	35.3181	5	256	$4 \times 4 \times 4$	LBG	0:0:1:4:16
G27_HG70	23,419,104	246,371	95.06	35.7988	5	256	$4 \times 4 \times 3$	LBG	0:0:0:2:8

We tested the performance of a 3D pyramid compressor and listed the result in Table 1. The data sets of human resistance arteries have many regions, in which there are very low variations between voxels. We can get compression ratio as high as about 150:1 on these data sets. Data sets of rat mesenteric artery have much more details. We can get about $15 \sim 20 : 1$ compression ratio on these data sets with acceptable image quality. The *Peak-Signal-to-Noise Ratio (PSNR)* measure used in the table is defined by

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \tag{8}$$

where MSE is the mean-squared-error between the original and reconstructed images.

We compared the performance of 3D pyramid method with two video compressors: MPEG and Indeo©Video Codec 5.10. MPEG uses *motion compensation* and *DCT* techniques [8, 9], while Indeo©Video Codec 5.10 is based on *hybrid wavelet compression technology*. We used two data sets, one from rat mesenteric data and the other from artery human resistance arteries data. The comparison results show that the proposed method offers better image quality than MPEG and Indeo Codec 5.10 at almost the same compression ratio. Fig. 4 shows the compression results B3CGP rat mesenteric data sets. The average PSNR for the 3D pyramid method is roughly 0.11 dB and 1.70 dB better than that of MPEG and Indeo. Fig. 5 shows the compression result on G27_HG70 human resistance arteries data set. For this data set, we also get better image quality with the 3D Pyramid, with gains of 0.59 dB and 0.63 dB over MPEG and Indeo. The reconstructed images from Indeo look slightly blurred (see Fig. 5b-4), and blocky artifacts are visible in the reconstructed MPEG images, which are more irritating to the human visual system(see Fig. 5b-3). Another point worth mentioning is that while MPEG's coding rate varies with every frame, 3D pyramid scheme has a fixed rate allocation over whole stack, which makes the PSNR curves of 3D pyramid more smooth than that of MPEG.

We also examine the performance on a standard video sequence 'Claire' $(174 \times 144 \times 200\ voxels; 8bpp)$. We tested image quality using the three codecs at a rate 0.26 bpp. The 3D pyramid method offers almost the same image quality as that using Indeo video codec with 37.67 dB for average PSNR, but cannot compete with MPEG, which gains roughly 4 dB over the 3D pyramid and Indeo. As we discussed in the introduction section, the video sequence doesn't have information about a 3D object, but the information of a surface (2D or 3D surface projected on 2D) moving in time. Motion compensation based techniques are good at capturing the moving information, but are not suitable for 3D microscopy data. In contrast, subband coding techniques, such as using 3D pyramid structures, perform better on microscopic volumetric data compression.

5. CONCLUSION

In this paper, a 3D lossy confocal microscopy image compression scheme, using 3D pyramid structures and vector quantization, is introduced. The 3D pyramid structures utilize the correlations between voxels in 3 spatial dimensions and decompose the source signal into a series of levels of subbands, which would be more suitable for quantization and entropy coding.

The experimental results show that the 3D pyramid method provides good qualities of reconstructed images at $15 \sim 20 : 1$ compression ratio on rat mesenteric data sets; and 100:1 or even higher compression ratio on human resistance arteries ones. Both offer better image quality than that using MPEG and Indeo©Video codec 5.10 at the same compression ratios.

6. REFERENCES

[1] C. E. Shannon. *Coding Theorems for a Discrete Source with a Fidelity Criterion.* IRE Convention Record, part 4, pp. 142-163, Mar. 1959.

[2] R.M. Gray, D.L. Neuhoff. *Quantization.* IEEE Trans. Inform., vol. 44(6), pp. 2325-2383, Oct. 1998.

[3] Nasser M. Nasrabadi, Robert A. King. *Image Coding Using Vector Quantization: A Review.* IEEE Trans. Commun., vol. 36(8), pp. 957-970, Aug. 1988.

[4] Y. Linde, A. Buzo, R. M. Gray. *An Algorithm for Vector Quantizer Design* IEEE Trans. Commun., vol. 28, pp. 84-95, Jan. 1980.

[5] P. Cosman, R. Gray, M. Vetterli. *Vector Quantization of Images Subbands: A Survey.* IEEE Trans. Image Processing, vol. 5, pp. 202-225, Feb. 1996.

[6] Peter J. Burt, Edward H. Adelson. *The Laplacian Pyramid as a Compact Image Code.* IEEE Trans. Commun., vol. 31, pp. 532-540, Apr. 1983.

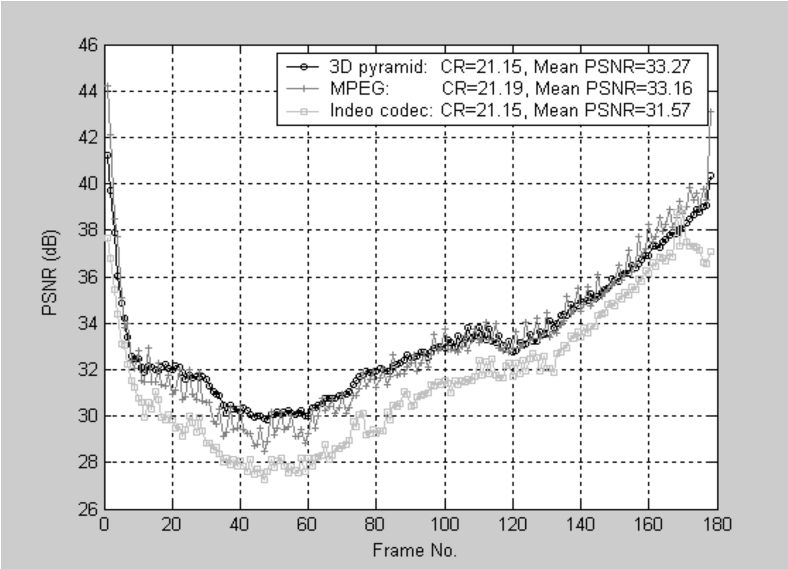
[7] E.H. Adelson, C.H. Anderson, J.R. Bergen, P.J. Burt, J.M. Ogden. *Pyramid Methods in Image Processing.* RCA Engineer, vol. 29(6), Nov/Dec 1984.

[8] Barry G. Haskell, Atul Puri, Arun N. Netravali. *Digital Video: An Introduction to MPEG-2.* Kluwer Academic Publisher, 1997.

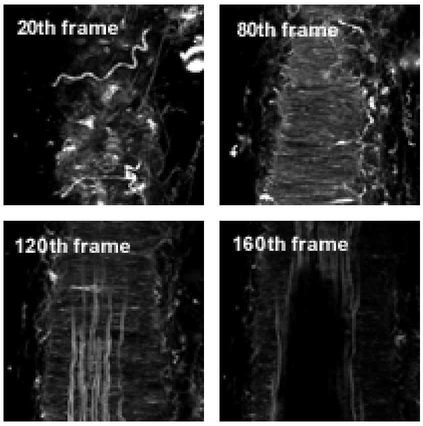
[9] Borko Furht, Joshua Greenberg, Raymond Westwater. *Motion Estimation algorithm for Video Compression.* Kluwer Academic Publishers, 1997.

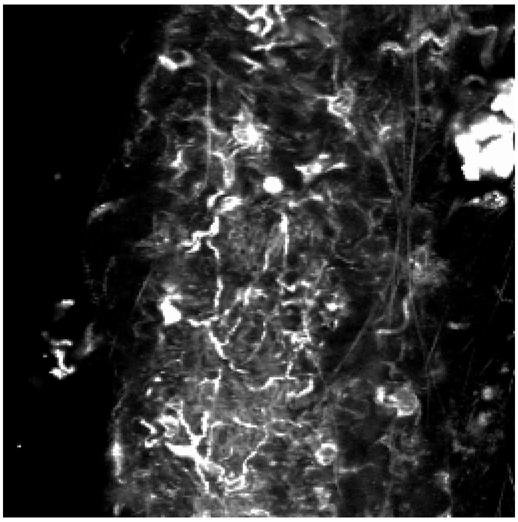
[10] James B. Pawley. *Handbook of biological confocal microscopy.* Plenum Press, 1990.

[11] Craig Daly, *et. al*. *Analysing the 3D Structure of Blood Vessels using Confocal Microscopy.* Microscopy and Analysis (UK), vol. 92, pp. 5-8, 2002.

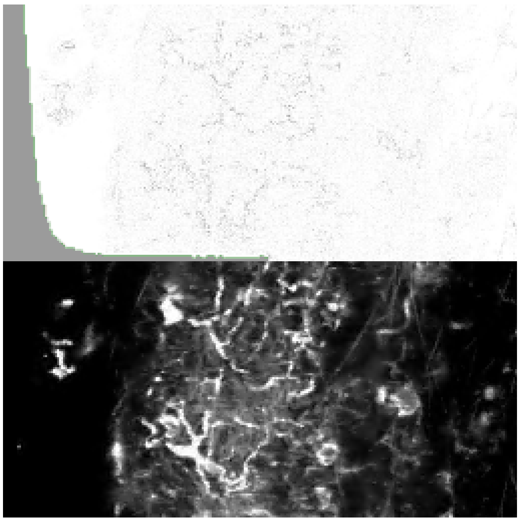


(a) Coding results using 3D pyramid method compared to MPEG and Indeo Video Codec 5.10 at the almost the same compression ratio (CR), Peak Signal to Noise Ratio (PSNR) for each frame.

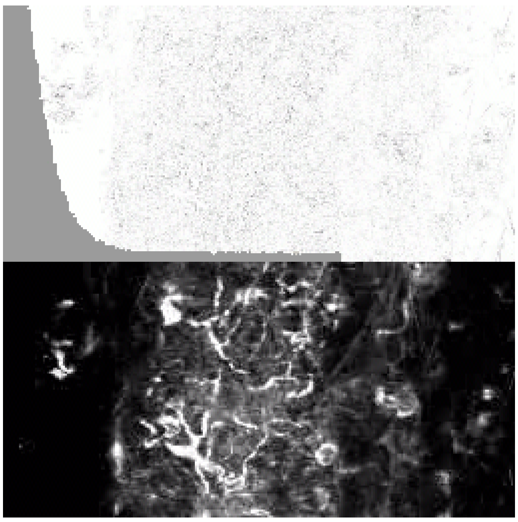




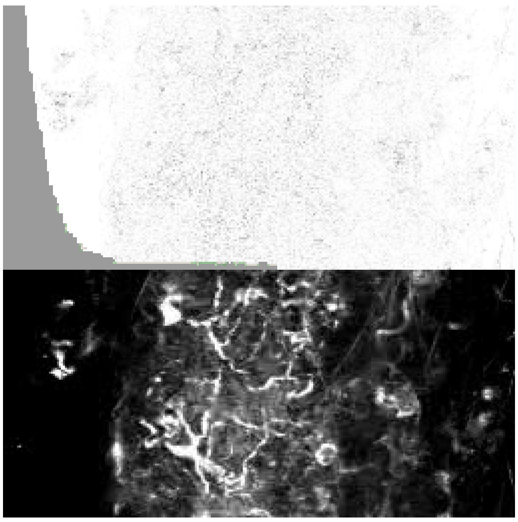
(b-1) The 47th frame from original image data set (256 x 256 x 8-bit)



(b-2) The reconstructed image using 3D pyramid codec, PSNR = 29.88. Upper half is the lower half part of the error image (white means no error), and the histogram of error image.

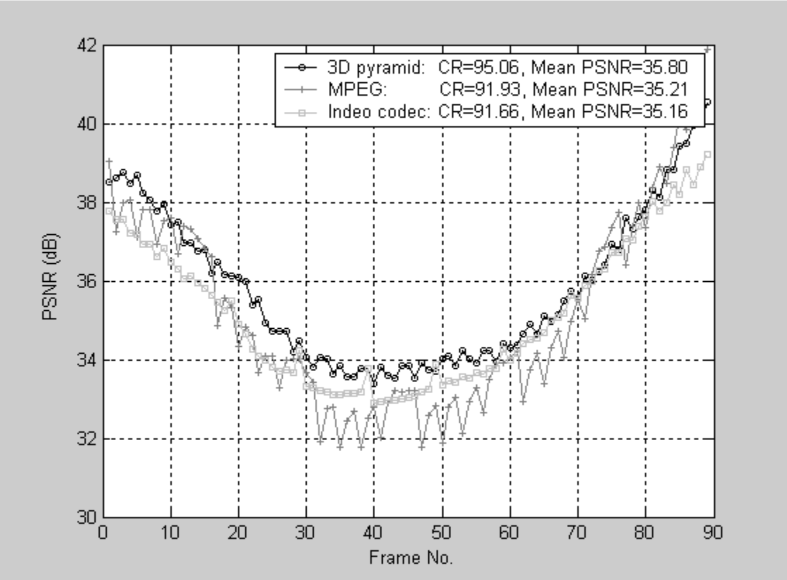


(b-3) The reconstructed image using MPEG video codec, PSNR = 28.50

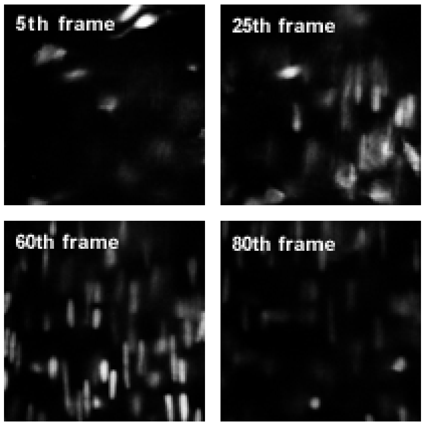


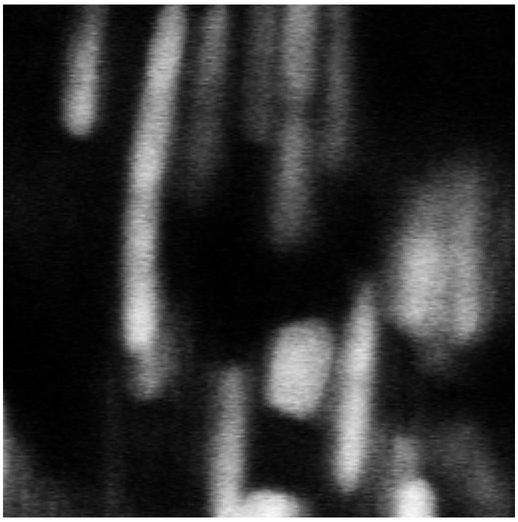
(b-4) The reconstructed image using Indeo© Video codec 5.10, PSNR = 27.24

Fig. 4. Compression results using 3D pyramid codec on B3CGP data set compared to MPEG and Indeo©Video codec 5.10



(a) Coding results using 3D pyramid method compared to MPEG and Indeo Video Codec 5.10 at the almost the same compression ratio (CR), Peak Signal to Noise Ratio (PSNR) for each frame.





(b-1) An 256 x 256 area clipped from the 41st frame from original image data set (512 x 512 x 8-bit)

(b-2) The reconstructed image using 3D pyramid codec, PSNR = 33.76

(b-3) The reconstructed image sequence using MPEG video codec, PSNR = 32.01

(b-4) The reconstructed image sequence using Indeo Video codec 5.10, PSNR = 32.91

Fig. 5. Compression results using 3D pyramid codec on H27HG data set compared to MPEG and Indeo© Video codec 5.10