A New Video Coding Algorithm using 3D-Subband Coding and Lattice Vector Quantization

Joong-Han Choi and Keun-Young Lee

Abstract

In this paper, we propose an efficient motion adaptive 3-dimensional (3D) video coding algorithm using 3D subband coding (3D-SBC) and lattice vector quantization (LVQ) for low bit rate. Instead of splitting input video sequences into the fixed number of subbands along the temporal axes, we decompose them into temporal subbands of variable size according to motions in frames. Each spatio-temporally splitted 7 subbands are partitioned by quadtree technique and coded with lattice vector quantization(LVQ). The simulation results show 0.1 ~4.3dB gain over H.261 in peak signal to noise ratio(PSNR) at low bit rate (64Kbps).

I. Introduction

Since the digital image preserves enormous amount of data, data compression techniques are required to process, store and communicate digital image data efficiently. The standardization activities for video coding are in progress, and some of them have already been done, for instance H.261[1] and MPEG. A conventional approach to achieving low bit rate is to employ hybrid coding with ME/MC along the temporal axes and 2D-DCT in the spatial domain as present digital video standards. In some other works, the 2D-DCT was replaced by 2D-SBC filter banks. This subband coding approach removes the blocking artifacts inherent in DCT based coding techniques. In case of H.261, it has been widely accepted as audiovisual coding standard which operate at $p \times 64$ Kbps. While $p = 1$ or 2 are more emphasized on H.261, the MPEG is focused on rate above 1 Mbps. The DCT-based H.261 provides impressive results in video conferencing image. But the blocking effects will degrade the visual quality especially at low bit rates. Also H.261 references two frames for motion compensation. So, it does not reflect characteristics of whole image sequence. SBC, an alternative method for block coding, was introduced by Crochirere[2], and became very popular in speech and image compression areas. The main idea for SBC was to decompose an original signal into multiple independent frequency bands by filter banks. The compression was accomplished by using appropriate coding techniques according to their decomposed signal characteristics. This 1D-SBC technique can be extended to 2D[3] and 3D[5-8] image sequences. The first 3D-SBC technique, proposed by Vitterli et. al[4] decomposed image sequences into 11 spatio-temporal subbands. Image sequences were separated into two temporal subbands and each temporal subbands was decomposed into several spatial subbands. Afterwards, the spatio-temporally decomposed subbands were coded in various schemes such as ADPCM, GVQ and UTSVQ. Most of the reported 3D-SBC techniques employed fixed tap sizes without considering the motion characteristics of the video sequences. Kim[8] used rate distortion theory to decide the variable number of taps along time axes and found a promise performance of 3D-SBC. Here, an alternative 3D-SBC technique with variable tap sizes considering the motion characteristics of video sequence was proposed. Also we proposed an efficient coding techniques adequate for ISDN basic rate interface(BRI) mode. This paper is organized as follows. Section II presents our motion adaptive coding algorithm including 3D decomposition, tap number decision, ME/MC, and LVQ. Then in section III, simulation results are presented to demonstrate the performance of proposed techniques. Finally, conclusions are presented in section IV.

II. Our 3D-SBC Algorithm

The temporal decomposition transforms original video sequences into multiple subbands, each of which contains the information about temporal change. If the amount of motion is small, the energy in "low subbands" will be high, otherwise the energy in "high subbands" will be high. If the amount of motion is large, aforementioned phenomenon deteriorates the coding efficiency. Motion adative 3D-SBC was introduced to overcome these artifacts

Manuscript received June 19, 1997; accepted November 5, 1997.

J. H. Choi is with Dept. of Elcetronics engineering, Taejon junior college, Taejon, Korea.

K. Y. Lee is with Dept. of Electronics Engineering, Sung Kyun Kwan University, Suwon, Korea.

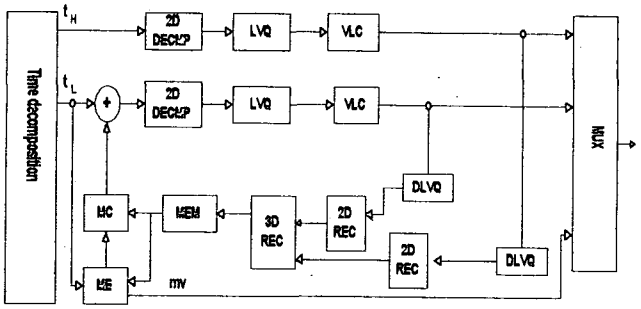


Fig. 1. Proposed encoder block diagram.

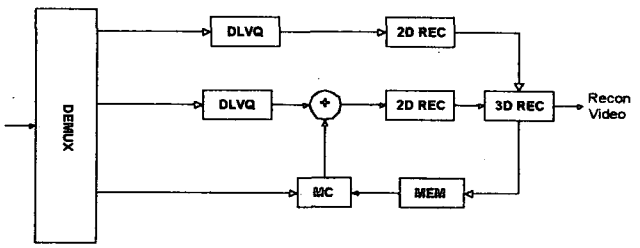


Fig. 2. Proposed decoder block diagram.

[7]. In case of small motion, the large tap size reflecting many frame characteristics would be preferable but it requires increased memory and coding complexity. Here we limit tap size to 1, 2, 4 taps for the proposed algorithm. Although the image sequence is decomposed into multiple temporal subbands, there exists a correlation between temporal baseband and previously decoded frame. ME/MC will be taken to remove temporal redundancy between temporal baseband and previously decoded frame. When the small amount of motion occurs, variance of MCFD is also small, so large number of taps will be taken. In this paper, the tap size is determined by utilizing the ratio of MCFD of various tap sizes. And then temporal decomposition is performed according to the decided tap size. The decomposition of video signals along time with tap size M results in one temporal baseband and M-1 temporal highbands. Fig. 1 and Fig. 2 show the block diagram of our motion adaptive encoder and decoder. Fig. 3 shows the decomposition flow of the our algorithm and the motion adaptive 3D-SBC decomposition flow.

1. Temporal Decomposition

First, the input frames are decomposed into 1-tap, 2-tap, 4-tap along time axes. Here the chosen 2-tap filter can be described in equation (1) through (4). The notation i and h represent low pass and high pass filter with delay 1, and H and G represent analysis and synthesis filters respectively.

$$H_l(z) = \frac{1}{2}(1+z^{-1}) \quad (1)$$

$$H_h(z) = \frac{1}{2}(1-z^{-1}) \quad (2)$$

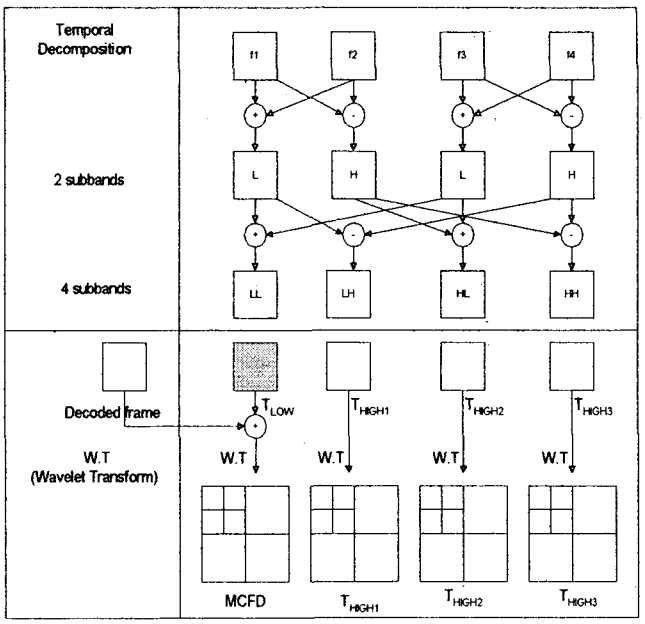


Fig. 3. 3D-SBC decomposition flow.

$$G_h(z) = -2H_l(-z) \quad (3)$$

$$G_l(z) = 2H_h(-z) \quad (4)$$

After temporal decomposition, variance of MCFD and the number of tap M are computed. Decomposition is performed by using the computed tap size M and resulted in one baseband and M-1 highbands. Since the baseband has characteristics similar to that of the previous decoded frame, applying the motion compensation between these frames is sensible. As a result, the motion compensation is carried out to the baseband of decomposed frame from the previous decoded frame. Motion compensation, however, is not applied to the highbands because it has no correlation between previous decoded frame and temporal high subbands.

2. Motion Adaptive Tap Size Decision

As described in the previous section, number of taps should be calculated according to their motion characteristics. Here, we define 3 motion estimation parameters as equations (5) through (7), where $\sigma^2_{0,1}$, $\sigma^2_{0,2}$, $\sigma^2_{0,4}$ are the variances of MCFD corresponding to each of 1-tap, 2-tap, 4-tap respectively. Also $R_{1,2}$, $R_{2,4}$, $R_{1,4}$ are the extension criteria for 1-tap to 2-tap, 2-tap to 4-tap and 1-tap to 4-tap respectively.

$$R_{1,2} = \frac{\sigma^2_{0,2}}{\sigma^2_{0,1}} \quad (5)$$

$$R_{2,4} = \frac{\sigma^2_{0,4}}{\sigma^2_{0,2}} \quad (6)$$

$$R_{1,4} = \frac{R_{2,4}}{R_{1,2}} \quad (7)$$

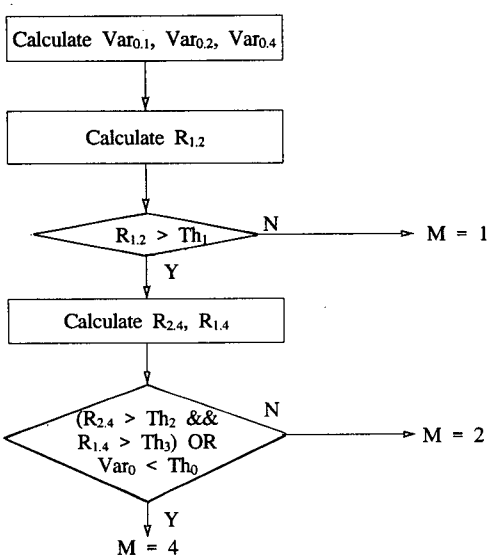


Fig. 4. Tap number decision flow.

If $\sigma^2_{n,m}$ is the variance of MCFD of n-th band splitted with m-tap, $\sigma^2_{0,1}$, $\sigma^2_{0,2}$, $\sigma^2_{0,4}$ are the variances of MCFD of the baseband decomposed with 1-tap, 2-tap, 4-tap respectively. If the video sequences have correlations between frames, it will satisfy the following relationships.

$$\sigma^2_{0,1} < \sigma^2_{0,2} < \sigma^2_{0,4} \tag{8}$$

The above relation would not be true, however, in the case of uncorrelated frames. When the amount of motion is small, the value of $\sigma^2_{0,1}$, $\sigma^2_{0,2}$, $\sigma^2_{0,4}$ will be small but the extension criterion $R_{1,2}$ for one tap to two taps will be large according to equations (5)-(8). The same result is true for $R_{2,4}$ and $R_{1,4}$. Tap size is extended from 1 to 2 if $R_{1,2}$ exceeds the threshold which is computed by experimentally. On the contrary, when the amount of motion is large, the opposite result holds true because of the smoothing effect of signals. As shown in Fig. 4 aforementioned tap number decision algorithm is summarized as follows.

1) Decompose input frames with 1-tap, 2-taps and calculate $\sigma^2_{0,1}$, $\sigma^2_{0,2}$.
2) Calculate $R_{1,2} = \frac{\sigma^2_{0,2}}{\sigma^2_{0,1}}$.
3) If $R_{1,2} < \mathrm{Th}_1$, decide 1-tap and goto step 1), otherwise skip to step 4) .
4) Decompose input frames with 4-taps and calculate $\sigma^2_{0,4}$
5) Calculate $R_{2,4} = \frac{\sigma^2_{0,4}}{\sigma^2_{0,2}}$.
6) If (($R_{2,4} > \mathrm{Th}_2$ && $R_{1,4} > \mathrm{Th}_3$) || $\sigma^2_{0,1} < \mathrm{Th}_0$) decide 4-tap , otherwise decide 2-tap.
7) Go to step 1).

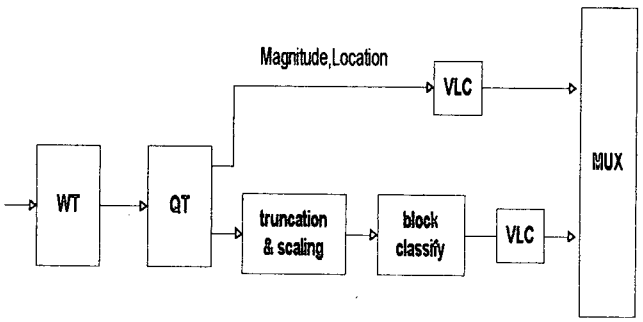


Fig. 5. Lattice VQ block diagram.

3. Spatial Decomposition

Input frames are decomposed into 1, 2, 4 subbands along the temporal axes. MCFD is obtained by applying ME/MC from the previous decoded frame to the baseband of the decomposed frame. Most of the MCFD and highbands signals show Laplacian distribution with zero mean and small variance. The temporal MCFDs of the baseband and highbands are further decomposed into 7 spatio subbands to derive sharper Laplacian distribution. Here, we used wavelet transform basis represented in M. Antonini, et al.[9].

4. Coding

As shown in Fig. 5, the 3D decomposed subband signals are partitioned into various block size and coded with LVQ.

1) Quadtree Techniques

The decomposed subbands are partitioned into various size square blocks in accordance with variance of the block by using quadtree techniques. Our quadtree techniques are as follows. First, all the spatio-temporally decomposed MCFD and highband coefficients are partitioned into 2×2 size block. If the sum of variance for the adjacent four 2×2 size block falls below threshold value, they are merged into 4×4 size blocks. In the same way, all of the blocks are merged into one of 2×2, 4×4, 8×8, 16×16 sizes block. Size and position of classified blocks resulted from quadtree technique are coded with VLC and transmitted to mux. As the variance of each block becomes smaller, the bigger block size. Therefore the large block will show a sharper distribution curve than a small block.

2) Truncation and Scaling

VQ is now considered one of the most powerful tools for image compression. Linde, Buzo and Gray proposed an efficient codebook design algorithm which is locally optimal with training sequences[10]. This algorithm, however, involves enormous memory and encoding complexity as the vector dimension increases. An alternative method for constructing a codebook is LVQ proposed by Conway and Sloane to reduce these drawbacks

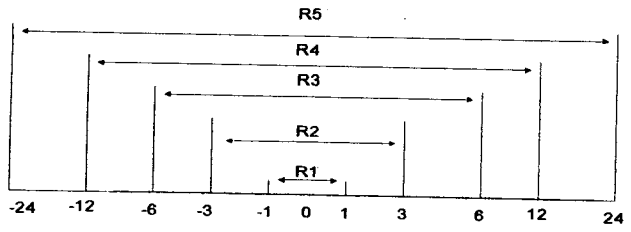


Fig. 6. Truncation and scaling for 1-D.

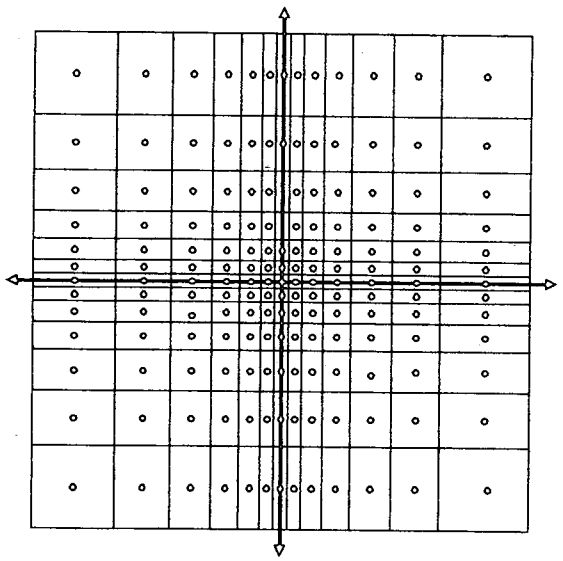


Fig. 7. Truncation and scaling for 2-D.

[11, 12]. But these do not always provide an efficient entropy coding without reducing codebook sizes. Here we reduce codebook sizes by introducing truncation and scaling techniques. A truncation and scaling example of the wavelet transformed coefficients for 1-D is shown in Fig. 6. First, the values above $\pm$ 24 are truncated to $\pm$24. The Voronoi region is therefore confined to $\pm$24. After truncation, scaling techniques are applied to quantize the input vectors. The scaling steps represented in Fig. 6 are obtained by analyzing the coefficients of the each subbands experimentally. The values between -1 and 1 are classified in region 1 and the values between -3 and 3 except in region 1 are classified region 2, and so on. Fig. 7 illustrates the truncation and scaling for 2-D. As shown in this figure, the scaling step sizes are increased when going far away from zero point. Dotted points within cubic lattice represent the reconstruction value. Most of wavelet coefficients usually have zero value and non-zero values occurred intermittently. By exploiting the characteristics of quadtree techniques 2×2, 4×4, 8×8 blocks are truncated to $\pm$24, $\pm$12, $\pm$6 respectively and 16×16 blocks are discarded.

3) Block Classification

① 2×2 vector

The partitioned blocks resulting from quadtree are further

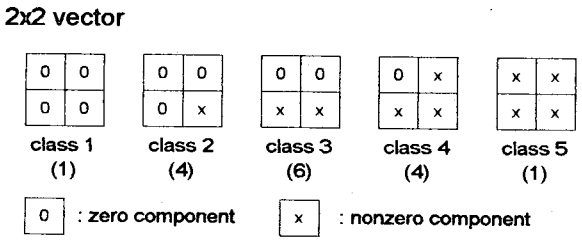


Fig. 8. Block classes for 2×2 vector.

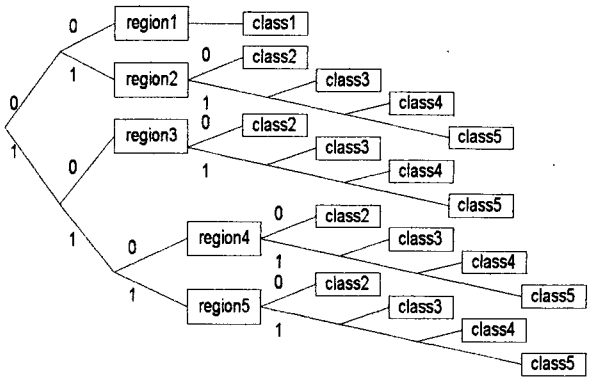


Fig. 9. Bit assignment tree for 2×2 vector.

classified by utilizing the position of the coefficients, that are zeros and non-zeros. A 2×2 size blocks are grouped by the number of zeros and then are further classified into various sub-types depending on the position of zeros and non-zeros. Fig. 8 illustrates the block classes of the 2×2 vector which are classified depending on the number of zeros. Since all of the components in region 1 contain zero, only one class would exist. Other regions contain from class 2 to class 5 depending on the number of zeros. In this figure, we assign short codewords to blocks containing many zeros to improve coding efficiency. Class 2 containing one non-zero component and three zero components has 4 sub-types depending on the position of this non-zero coefficients. The sub-type classification procedures for classes 3, 4, 5 should be performed according to the aforementioned procedure. The region and class information are coded with VLC. Fig. 9 shows the bit assignment tree for region and classes belonging to each region. Table 1 shows the bit assignment results for 2×2 vector. The number of lattice with four dimension vector in each region are 1, 80(3^4-1^4), 544(5^4-3^4), 1776 (7^4-5^4) and 4160 respectively. The number of subtypes that exist in each class of region 2 are 8, 24, 32,16 according to the position of non-zero points. The number of subtypes belonging to class 5 are obtained by subtracting the summmation from class 2 to class 4 from the total number of lattice at each regions.

② 4×4, 8×8 vector

Fig. 10 and 12 shows the block classes and bit assignment tree for 4×4 vector. Since the 4×4 vectors are composed of four

Table 1. Bit assignment table for 2×2 vector.

Region	# of lattice in region	Class	#of sub-type in class	Required bits		Total bits
				Region /Class	Sub-type id	
1	1	C1	1	2	0	2
2	80	C2	8	3	3	6
		C3	24	4	5	9
		C4	32	5	5	10
		C5	16	5	4	10
3	544	C2	8	3	3	6
		C3	24	4	5	9
		C4	32	5	5	10
		C5	480	5	9	14
4	1776	C2	8	4	3	7
		C3	24	5	5	10
		C4	32	6	5	11
		C5	1712	6	11	17
5	4160	C2	8	4	3	7
		C3	24	5	5	10
		C4	32	6	5	11
		C5	4096	6	12	18

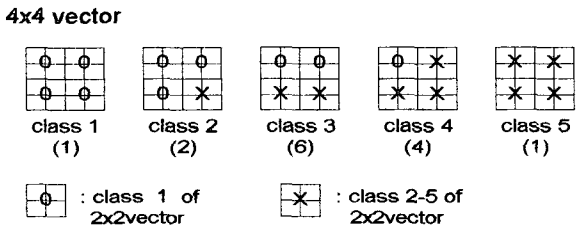


Fig. 10. Block classes for 4×4 vector.

adjacent 2×2 vectors, we used the same region information and class identification as in the 2×2 vector. A 4×4 class identification bits, however, should be appended to identify the position of the 2×2 vector '0' and 'x' in each class and case identification bits indicating the number ofcases within class. As shown in Fig. 10, 8×8 vectors are composed of four 4×4cases within class. As shown in Fig. 10, 8×8 vectors are composed of four 4×4 vectors. The aforementioned 4×4 techniques are applied to 8×8 vectors. 16×16 vectors are discarded for it does not degrade the reconstructed image. One should keep in mind that truncation is readjusted according to various block sizes.

III. Simulation Results

Simulations were performed on typical QCIF video conferencing sequences, Salesman and Miss America. The video sequences were filtered temporally using the 2-tap Harr filters, which are mean and difference between frames. 2-tap filters are extended to 4-tap filters by exploiting them. Wavelet basis proposed by M.

Table 2. Bit assignment for 4×4 vector.

Class	bit assignment description			Max expected bits
	4×4 Class	case (bit)	vector index (2×2)	
Class1	2	1(0)	0	2
Class2	2	4(2)	×1	22(4+18×1)
Class3	2	6(3)	×2	41(5+18×2)
Class4	3	4(2)	×3	59(5+18×3)
Class5	3	1(0)	×4	75(3+18×4)

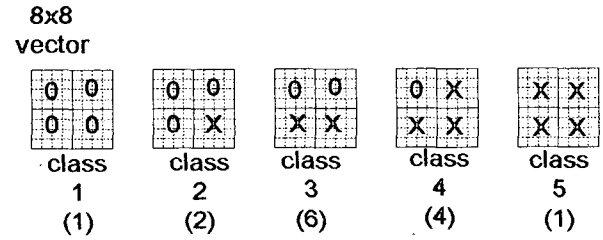


Fig. 11. Block classes for 8×8 vector.

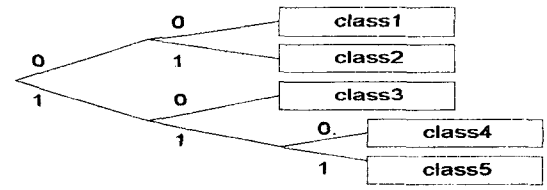


Fig. 12. Bit assignment tree for 4×4 vector.

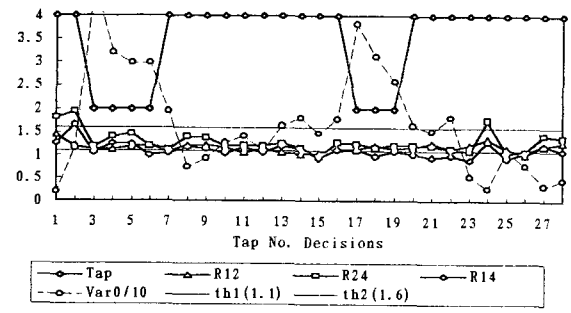


Fig. 13. The number of taps for Salesman.

Antonini[9] was used to decompose spatial domain. If $\sigma^2_{0,1}$ falls below threshold, we take tap size 4 by assuming that these video sequences have very small motion within 4 frames. Fig. 13 shows the decided number of tap for the Salesman by using the proposed algorithm. In this figure, the values $var_0/10$ representing the variance of 1-tap MCFD are displayed to indicate the amount of motion. The experimental threshold values of Th_0, Th_1, Th_2, Th_3 are taken 20, 1.1, 1.6, 1 respectively. Motion vectors were

Table 3. Comparison of time adaptive SBC and time fixed SBC for salesman.

	1-tap fixed	2-tap fixed	4-tap fixed	motion adaptive
PSNR(dB)	29.95	33.00	33.26	33.28

Table 4. The coding results(64, 128, 256Kbps).

Video \ Quality		PSNR(dB)					
		64Kbps		128Kbps		256Kbps	
		H.261	3D SBC	H.261	3D SBC	H.261	3D SBC
Sales-man	Y	28.70	33.08	33.66	35.78	37.60	39.20
	U	35.99	38.6	38.90	38.21	41.82	39.23
	V	36.61	37.97	39.61	39.38	42.64	39.35
Missa	Y	35.55	37.96	39.39	39.28	40.87	40.00
	U	36.79	37.63	38.39	37.83	39.03	38.72
	V	34.60	37.37	38.51	38.26	39.96	39.17

coded with VLC defined in H.261 table. The quadtree informations which represent block size and position of coefficients were coded with VLC. Bit rate is controlled by varying the threshold value of quadtree. Low bit rate can be obtained by taking large threshold value since it generates more big blocksize than small value. Table 3 shows the coding results of various fixed tap size and adaptive techniques we proposed. In this table, decomposition with 1-tap showed worst performance, since the motion vectors need to be coded at every frames. The proposed motion adaptive scheme showed best performance in view of PSNR. It is a natural consequence that most of the frames were decomposed to 4-tap, and remainings were decomposed to 2-tap. Table 4 compares the coding results between H.261 and the proposed scheme at low bit rate (64Kbps, 128Kbps, 256Kbps). In this table, we conclude that proposed scheme is superior to that of H.261 in 64Kbps and 128Kbps.

Ⅳ. Conclusions

We proposed an efficient 3D-SBC coder suitable for low bit rate application, especially for p=1,2. Instead of splitting fixed number of temporal subbands, the frames were splitted into variable temporal subbands from 1 to 4 depending on motion information. To remove the temporal redundancy existing in video sequence, motion compensation was performed between baseband and previous frame decoded. By considering the statistical characteristics of spatio-temporally decomposed signals, the LVQ codebook was designed by truncation and scaling for low bit rate application. MVs are one of the most difficult to implement high performance in low bit rate application such as H.261. But, as in 3D-SBC, the required bits to code them were reduced considerably. Since this LVQ codebook is designed to 64Kbps and 128Kbps fine truncation and scaling steps of LVQ codebook is required to support above 256Kbps.

References

[1] ITU-T Recommdation H.261, "Video Codec for Audiovisual Services at p×64kbps," Geneva, 1990.

[2] R. E. Crochiere, S. A. Webber, and J. L. Flangan, "Digital Coding of Speech in Subbands," Bell Syst. Tech., J., Vol. 55, pp. 1069-1085, Oct. 1976.

[3] J. W. Woods and S. D. O'Neil, "Subband Coding of Images," IEEE Trans. on Acoustic., Speech, and Signal Processing, Vol. 34, No. 5, pp. 1278-1288, Oct. 1986.

[4] G. Karlsson and M. Vetterli, "Three dimensional subband coding of video," Proc. of ICASSP., pp. 1100-1103, 1988

[5] G. Karlsson and M. Vetterli, "Three Dimensional Subband Coding of Video," IEEE Trans. on Image Processing, Vol. 4, No. 2, Feb. 1995.

[6] H. Gharavi, "Subband Coding Algorithm for Video Applications : Videophone to HDTV-Conferencing," IEEE Trans. on Circuits and Systems for Video Technology, Vol. 1, No. 2, pp. 174-183, June 1991.

[7] K. N. Ngan and W. L. Chooi, "Very Low Bit Rate Video Coding using 3D Subband Approach," IEEE Trans. on Circuits and Systems for Video Technology, Vol. 4, No. 3, pp. 309-316, June 1994.

[8] Y. K. Kim, R. C. Kim, and S. U. Lee, "On the Adaptive 3D Subband Coding," Proc. SPIE Visual Communications and Image Processing, Vol. 2727, pp. 123-132. 1996.

[9] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechiess, "Image Coding using Wavelet Transform," IEEE Trans. on Image Processing, Vol. 1, No. 2, pp. 205-220, Apr. 1992.

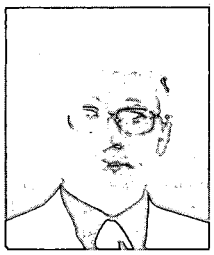
[10] Y. Linde, A. Buzo, and R. M. Gray, "An Algorithm for Vector Quantizer Design," IEEE Trans. on Commun., Vol. 28, No. 1, pp. 84-95, 1980.

[11] J. H. Conway and N. J. A. Solane, "Voronoi Regions of Lattices, Second Monents of Polytopes, and Quantization," IEEE Trans. on Information Theory, Vol. 28, pp. 211-226, 1982.

[12] J. H. Conway and N. J. A. Solane, "Fast Quantizing and Decoding Algorithms for Lattice Quantizers and Codes," IEEE Trans. on Information Theory, Vol. 28, pp. 227-232, 1982.

[13] J. W. Lee, C. Moon, B. W. Hwang, and K. Y. Lee, " Image Coding using Variable Block Lattice Vector Quantization of Wavelet Coefficient," Proc. of ITC-CSCC, pp. 225-228, July 1996.

Joong-Han Choi was born in Korea, in 1958. He received the B.S., M.S. and Ph. D. degrees in electronic engineering from Sung Kyun Kwan University in 1982, 1991, and 1996 respectively. From 1982 to 1991, he worked as a senior researcher of the ADD. From 1991 to 1993, he was a senior researcher of the ETRI. Currently, he is an assistant professor in Taejon junior college. His research interests are image communication, computer communication, u-processor application, mobile communication, etc.



Keun-Young Lee was born in Korea in 1947. He received the B.S. degree in electrical engineering from Chonnam University. He received the M.S. and Ph. D. degrees in electronic engineering from Han Yang University in 1975, 1978 respectively. From 1979 to 1980, he worked as a visiting professor of the Denmark University. During 1987 to 1988, he was a visiting professor of the Loughborough University. Since 1981, he is a professor in the department of Electrical, electronic and computer engineering, Sung Kyun Kwan University, Suwon, Korea. Currently his research interests are image processing, compression, character recognition, neural networks, u-processor application, power control system, etc.