論文2003-40SD-5-6

웨이블릿 계수에 대한 효율적인 무손실 부호화 및 복호화기 설계

(Design of an Efficient Lossless CODEC for Wavelet Coefficients)

李 宣 永 *, 趙 敬 淳 *

(Seonyoung Lee and Kyeongsoon Cho)

요 약

웨이블릿 변환에 기반을 둔 영상 압축은 기존의 JPEG과 비교했을 때, 블록 형태의 잡음이 나타나지 않고 화소 당 비트 수를 적게 압축할 때의 화질이 우수하므로 산업계에서 널리 사용되고 있다. 이산 웨이블릿 변환에 의해서 생성되는 계수들은 양자화 과정을 거쳐서 코드 비트 수를 줄이게 된다. 양자화 다음에는 무손실 부호화 과정을 통해서 코드 비트 수를 더 감소시킨다. 본 논문은 생성된 계수들의 통계적 특성을 바탕으로 양자화된 계수들에 대하여 효율적으로 무손실 부호화를 수행하는 새로운 알고리즘을 제시하고 있다. 이산 웨이블릿 변환과 양자화 과정을 결합하여 본 알고리즘을 0.5㎛ 표준 셀 방식의 영상 압축 칩으로 구현한 결과, 효율성과 성능을 확인할 수 있었다.

Abstract

The image compression based on discrete wavelet transform has been widely accepted in industry since it shows no block artifacts and provides a better image quality when compressed to low bits per pixel, compared to the traditional JPEG. The coefficients generated by discrete wavelet transform are quantized to reduce the number of code bits to represent them. After quantization, lossless coding processes are usually applied to make further reduction. This paper presents a new and efficient lossless coding algorithm for quantized wavelet coefficients based on the statistical properties of the coefficients. Combined with discrete wavelet transform and quantization processes, our algorithm has been implemented as an image compression chip, using 0.5㎛ standard cells. The experimental results show the efficiency and performance of the resulting chip.

Keywords : Image Compression, Discrete wavelet transform, Lossless coding

I. Introduction

For the advanced multimedia systems and internet applications, the image data that should be processed is getting bigger and needs to be handled more efficiently. A 16-bit VGA (video graphics array) color image with 640 x 480 pixels requires 600 Kbytes of memory to store. It requires 18 Mbytes of memory to store 30 frames of image per second for real-time applications. If we want to store one hour of video data, 64 Gbytes of memory is required. This makes it very difficult to store or transmit the image

* 正會員, 韓國外國語大學校 電子情報工學部
(School of Electronics and Information Engineering, Hankuk University of Foreign Studies)
接受日字:2002年12月20日, 수정완료일:2003年4月29日

data in the multimedia systems. The solution to this problem will be the compression of the image data to make it smaller. JPEG (joint photographic coding experts group) based on DCT (discrete cosine transform) is one of the most famous image compression methods. DWT (discrete wavelet transform)[1] can characterize the signals locally in spatial domain while DCT cannot. Hence, image compression based on DWT is regarded as a next-generation technique. Compared to JPEG, the image compression based on DWT shows no block artifacts and provides a better quality of the restored image when compressed to low BPP (bits per pixel). In addition, DWT is more suitable for noisy communication channels. These advantages make this technique attractive to many applications such as security systems. For this reason DWT was included in JPEG2000 Standard[2]. DWT can also be applied to other areas[3-6] in addition to image compression.

The image pixel values in the spatial domain are transformed into the coefficients in the frequency domain through DWT. The coefficients are then quantized to reduce the number of code bits to represent them. Quantization means the mapping of a broad range of input values to a limited number of output values. Since it is an irreversible operation, it is regarded as lossy data compression. After quantization, lossless coding processes are usually applied to make further reduction of the number of code bits. Lossless data compression is an error-free compression technique and performed by the reduction of coding redundancy. Run-length coding and Huffman coding[7] are the variable-length coding techniques for lossless compression. Run-length coding is to represent an image by a sequence of lengths that describe successive runs of source symbols. Huffman coding assigns the smallest number of code bits for the most probable source symbol. This paper presents a new and efficient lossless coding algorithm for quanized wavelet coefficients. We developed this algorithm by simplifying Huffman coding algorithm based on the observation of the statistical properties of the wavelet coefficients and applying run-length coding technique for zero-valued coefficients only. The advantage of our algorithm is its simplicity, which makes it possible to implement an ASIC (application specific integrated circuit) chip with a small number of gates while it maintains a reasonable compression ratio, compared to the original Huffman coding algorithm. We have implemented two-dimensional DWT core with the goal of small gate count by using external SDRAM (synchronous dynamic random access memory) and carefully scheduling the operations of the core[8]. Combining with this DWT core, we implemented our coding algorithm as an image compression chip. We described the RTL (register-transfer level) circuit in Verilog HDL (hardware description language)[9] and verified it using Verilog-XL of Cadence Design Systems. We synthesized the gate-level circuit with the 0.5μm standard cell library[10] of Samsung Electronics Company using Design Compiler of Synopsys, Inc. Xilinx FPGA (field programmable gate array) was used for prototyping the circuit. The whole circuit including DWT, inverse DWT, quantization, un-quantization and CODEC (coder and decoder) modules was fabricated with the 0.5μm standard cell manufacturing process of Samsung Electronics Company. Our circuit is using 27-MHz clock and requires approximately one clock cycle to encode each quantized wavelet coefficient, making it usable for real-time applications. The number of equivalent two-input NAND gates in CODEC module is 16,118, which is small enough to be included in our image compression chip.

In Section II, we summarize the theory of the two-dimensional DWT as background and describe the modifications that have been made to the original theory to reduce the computational complexity. The quantization method adopted in our image compression chip is also explained. Section III presents our lossless coding algorithm with the comparison of the original Huffman coding algorithm.

Section IV shows the ASIC chip implementation results for our algorithm. The experimental results are also presented in this section. Finally, in Section V, we conclude the paper.

II. Background

We have designed a two-dimensional DWT core module and reported the implementation results. This core performs two-dimensional DWT and also inverse DWT based on the 9-tap Daubechies filter[11] as in JPEG2000. We utilized an SDRAM chip[12] external to the core in order to reduce the core size by minimizing the internal memory. The resulting circuit is considerably small compared to other approaches such as systolic array method[13 14]. The computation speed of the core was achieved by carefully scheduling the operations of the core. This section summarizes the theory and implementation results of the two-dimensional DWT core reported in[8].

DWT can be described as a multi-resolution analysis combined with a sub-band coding[1]. The original signal is passed through the high-pass and low-pass filters, followed by sub-sampling by two. The output of the low-pass filter is then passed through the same high-pass and low-pass filters followed by the sub-sampling process. This process continues until the specified level is reached. Two-dimensional DWT can be performed with a separable extension of the one-dimensional decomposition algorithm described above. At each level we convolve the rows with one-dimensional high-pass and low-pass filters, retain every other row, convolve the columns of the resulting signals with the same one-dimensional high-pass and low-pass filters, and retain every other column. The image can be reconstructed by inverse DWT, in which we add a column of zeros, convolve the rows with one-dimensional high-pass and low-pass filters, add a row of zeros between each row of the resulting image, and convolve the columns with the same one-dimensional high-pass and low-pass filters.

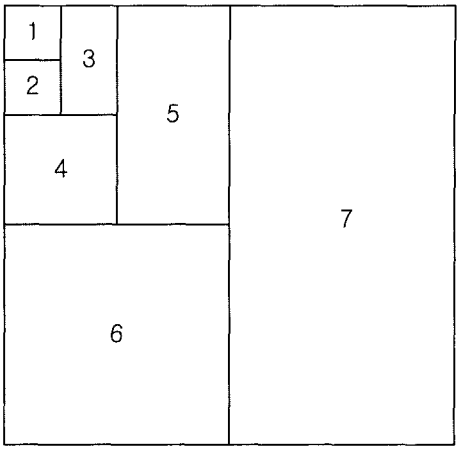

그림 1. 수정된 DWT에서의 이차원 분할
Fig. 1. Two-dimensional decomposition in the modified DWT.

Since the visually important information in the image is concentrated in the low-frequency region, we omitted the column-wise decomposition steps in high-frequency region to reduce the computational efforts for two-dimensional DWT. The number of clock cycles required in performing two-dimensional DWT for a given image data can be reduced by 25% with this modification. However, it causes some degradation of compression ratio to maintain the same quality of the restored image. According to our experiments, the degradation in the compression ratio with the same PSNR (peak signal to noise ratio) is quite small, usually 1 ~ 2%. Detailed description has been given in[8]. <Fig. 1> illustrates the decomposed regions in our modified two-dimensional DWT.

The coefficients generated from two-dimensional DWT are passed to the quantization module to reduce the number of code bits to represent them. The number of code bits is derived from the quantization bin size. Bigger the quantization bin size, smaller the number of code bits. In our implementation, all of the wavelet coefficients in one region are divided by the same quantization bin size. Seven regions are defined for a given image, as illustrated in <Fig. 1>. Smaller quantization bin size

표 1. 7개 영역에 대한 양자화 크기
Table 1. Quantization bin sizes for seven decomposed regions.

Region	1	2	3	4	5	6	7
Bin size option 1	1	1	1	4	4	16	16
Bin size option 2	2	4	4	16	16	64	32
Bin size option 3	4	4	8	16	32	64	128
Bin size option 4	8	8	16	32	64	128	256

is used for higher-level and lower-frequency regions. <Table 1> shows the various combinations of the quantization bin size in our quantization module. We made the experiments using various combinations of the quantization bin size and selected these values yielding the better quality. The user of our image compression chip is allowed to select one of them.

Ⅲ. Coding algorithm

The next step of quantization in the image compression is a lossless coding process. Through this process, more reduction of the number of code bits to represent the coefficients is achieved by the elimination of coding redundancy. Run-length coding and Huffman coding are the typical lossless variable-length coding techniques. Variable-length coding means that the number of code bits can be varied for each coefficient. Run-length coding is to represent the quantization results by a sequence of lengths that describe successive runs of source symbols, that is, coefficient values. By Huffman coding, the source symbol with the highest probability of occurrences is assigned the smallest number of code bits. This section describes the original run-length coding and Huffman coding algorithms, and then explains our lossless coding algorithm.

As an example of run-length coding, consider the sequence of source symbols "*AAAAAAABBC DDDDDEEE*", where A, B, C, D and E represent the quantized wavelet coefficient values. If we represent each wavelet coefficient in 16 bits, 288 bits (16 bits × 18 coefficients) will be needed. Instead, run-length code "A7B2C1D5E3" is used for the above example, which means that 7 runs of A's, 2 runs of B's, 1 run of C, 5 runs of D's and 3 runs of E's. By doing so, we can reduce the total number of bits to represent the given stream.

The most popular technique to eliminate coding redundancy is Huffman coding, which yields the smallest possible number of code bits per source symbol. In Huffman coding, the number of occurrences is counted for each source symbol and all the symbols are ordered according to their probability. Based on this sorted histogram, a binary tree is constructed and the labels '0' and '1' are assigned to all the branches of the tree. The bit representation of each symbol is extracted by traversing the tree from the root to the leaves. <Fig. 2> shows the binary tree for the source symbol stream "*AAAAAAABBCDDDDDEEE*", which is constructed according to the number of occurrences. When traversing the tree, label '1' is assigned to the branch representing the smaller number of occurrences and label '0' is assigned to the other branch. Consequently, the codes for the source symbols are A : "1", B : "0000", C : "0001", D : "01" and E : "001". The Huffman code for the above source symbol stream will be "11111110000000000010 1010101010 01001001", resulting in 38 bits.

Even though Huffman coding is known as optimal,

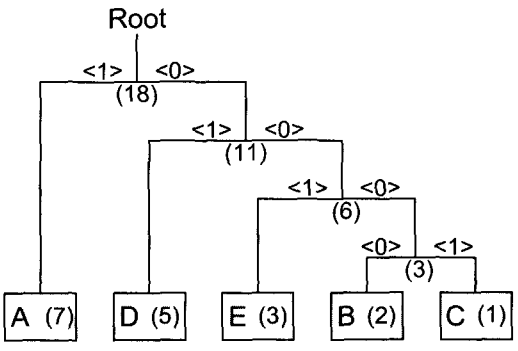

그림 2. 허프만 부호화 예제
Fig. 2. Huffman coding example.

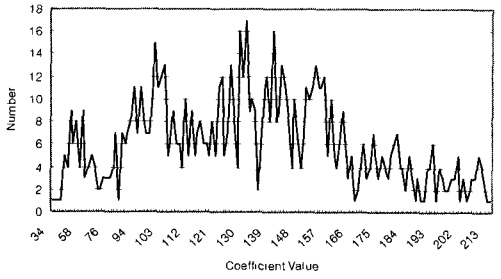

그림 3. 레나 영상에서 영역 1의 웨이블릿 계수 분포
Fig. 3. Distribution of the wavelet coefficient values for region 1 of Lena image.

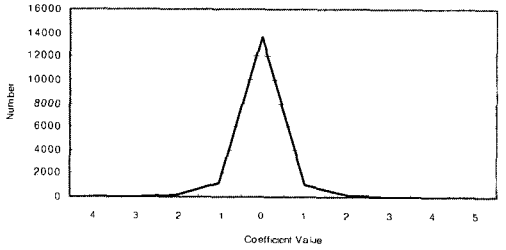

그림 4. 레나 영상에서 영역 7의 웨이블릿 계수 분포
Fig. 4. Distribution of the wavelet coefficient values for region 7 of Lena of image.

the problem is that it requires the ordered histogram for all the source symbols. It requires excessive data access and computational efforts to count the number of occurrences of all symbols and to order them. In addition, binary tree information should also be stored as well as coded bit stream. To overcome this problem, Huffman code tables may be used as in JPEG. This approach, however, still requires that additional code tables should be built and stored for later use.

To investigate the statistical properties of wavelet coefficients, we examined the distribution of the coefficient values for various image samples. <Fig. 3> and <Fig. 4> illustrate the distribution of the coefficient values for region 1 and region 7, respectively. Regions 2 ~ 6 have the similar distribution characteristics to region 7. The definition of each region is given in <Fig. 1>. The sample image used for <Fig. 3> and <Fig. 4> is black-and-white Lena image with 8-bit 256 × 256 pixels shown in <Fig. 5>. As seen in these pictures, the coefficients in region 1 are distributed in much smoother fashion than other regions. For regions 2 ~ 7, the coefficients are showing Laplacian distribution characteristics with a sharp peak at value zero. Based on this observation, we made our strategy to handle region 1 as one group (called group LL) and the other regions as another group (called group Non-LL). We are going to describe our algorithm for these two groups separately.

The wavelet coefficients of group LL (region 1) represent the lowest-frequency data for both of row and column directions. Since the distribution characteristics of the coefficient values are smooth, we decided to use fixed-length coding for this region. Fixed-length coding means that the same number of code bits is used for all coefficients in this region. To do this, the minimum value m and maximum value M for all coefficients are searched first. The difference Δ ($\Delta = M - m$) is used to determine the number of code bits N($N = \lceil log2\ \Delta \rceil$). Then, the difference between each coefficient value and m is stored in N bits. For example, consider six 8-bit coefficients "2 6 23 10 9 8", where M = 23, m = 2, Δ = 21 and N = 5. Hence, the above coefficients

그림 5. 256 × 256 레나 영상
Fig. 5. 256 × 256 Lena image.

are coded as follows: 2 - 2 = 0 (00000), 6 - 2 = 4 (00100), 23 - 2 = 21 (10101), 10 - 2 = 8 (01000), 9 - 2 = 7 (00111), 8 - 2 = 6 (00110). Through our fixed-length coding, total number code bits to store six coefficients are reduced from 48 bits (6 × 8 bits) to 41 bits (6 × 5 bits + 8 bits to represent m + 3 bits to represent N).

As illustrated in <Fig. 4>, the distribution of the wavelet coefficient values of group Non-LL (regions 2 ~ 7) is characterized as Laplacian. The number of zero-valued coefficients is outstandingly more than that of others. The number of non-zero-valued coefficients decreases very rapidly as the absolute values of the coefficients increase. We can find that the shape of distribution curve becomes sharper with larger peak value as the compression ratio gets higher. Based on this observation, we coded zero-valued coefficients and non-zero-valued coefficients in a different way. For the zero-valued coefficients, we applied run-length coding technique. We will describe our algorithm for the non-zero-valued coefficients later.

To distinguish the number of runs of zero-valued coefficient R_{ZERO} from the value of non-zero-valued coefficient $V_{NONZERO}$, a special code bit is inserted at the start of the code. Code bits '0' and '1' are inserted for R_{ZERO} and $V_{NONZERO}$, respectively. Hence, if a code starts from '0', it means R_{ZERO}. Otherwise it represents $V_{NONZERO}$. Since the number of bits to denote R_{ZERO} can be varied according to its value, it is not reasonable to represent it with a fixed-length number. Instead, we adopted a semi-variable-length approach by partitioning R_{ZERO} into the tokens with two bits and inserting '0' bit at the start of each token. For example, if R_{ZERO} = 57, then its binary representation will be "111001". We divide "111001" into three tokens with two bits, that is, "11", "10" and "01". At the start of each token, '0' bit is inserted to create three tokens, "011", "010" and "001". These new tokens are concatenated to yield "001010011", where the order of tokens are reversed to make the hardware implementation easy. The number of bits for each token NB_{TOKEN} is two. As NB_{TOKEN} gets smaller, the resolution will be better but the number of '0's inserted will increase. If we use a larger NB_{TOKEN} , higher-order bits may be wasted in case of small R_{ZERO}. Through experiments with various image samples, we tried to find the value of NB_{TOKEN} which makes the smaller number of code bits and decided that NB_{TOKEN} = 2.

The value of non-zero-valued coefficient $V_{NONZERO}$ is coded in a different way. The code for $V_{NONZERO}$ consists of two fields. One field is to represent the number of bits for its value. The other field is to denote its value itself. <Table 2> shows the coding rules. The columns with label $V_{NONZERO}$ are the values of the coefficients, one for negative value and the other for positive value. The column with label *#Bits* represents the number of code bits for each $V_{NONZERO}$. The column *Size* is the field for the number of bits in $V_{NONZERO}$ and the column *Value* is the field for the value of $V_{NONZERO}$. The column *Size* consists of starting '1' bit, ending '1' bit and '0' bits between them. The number of '0' bits inserted between two '1' bits *NZB* is defined as the number of bits to denote $|V_{NONZERO}| - 1$. For example, if $V_{NONZERO}$ = -6, then $|V_{NONZERO}| - 1 = |-6| - 1 = 5$ and $NZB = \lceil \log_2 5 \rceil = 3$, yielding *Size* = "10001" by inserting three '0' bits. The column *Value* consists of *NZB* bits if $|V_{NONZERO}|$ is not 1. For negative $V_{NONZERO}$, *Value* is defined as $(|V_{NONZERO}| - Delta) * 2$, where $Delta = 2^{NZB-1} + 1$. For positive $V_{NONZERO}$, *Value* is defined as $(|V_{NONZERO}| - Delta) * 2 + 1$. In case that $V_{NONZERO}$ = -6, $Delta = 2^{3-1} + 1 = 5$ and $Value = (|V_{NONZERO}| - Delta) * 2 = (|-6| - 5) * 2 = 2$ = "010", consisting of *NZB* (= 3) bits. As special cases, when $V_{NONZERO}$ = 1 and $V_{NONZERO}$ = -1, *Value* = "1" and *Value* = "0", respectively. Let's take another example with a positive value: $V_{NONZERO}$ = 128. Since $|V_{NONZERO}| - 1 = |128| - 1 = 127$, $NZB = \lceil \log_2 127 \rceil = 7$ and *Size* = "100000001". With $Delta = 2^{NZB-1} + 1 = 2^{7-1} + 1 = 65$, Value = $(|V_{NONZERO}| - Delta) * 2 + 1 = (|128| - 65) * 2 + 1 = 127$ = "1111111".

Hence, $V_{NONZERO}$ = 128 is coded as "1000000011111111", resulting in 16 bits.

Ⅳ. ASIC implementation and experimental results

This section describes how our lossless coding algorithm presented in Section Ⅲ was designed as a hardware module. We merged this encoder module with two-dimensional DWT and quantization modules that were reported in[8] to implement an ASIC chip with image compression capability. <Fig. 6> and <Fig. 7> show the block diagrams of two-dimensional wavelet-based image compression circuit and the lossless encoder module in the compression circuit, respectively. The outputs of the DWT module are applied to the inputs of the quantization module. Then, the outputs of the quantization module are used as the inputs of the encoder module. The qunatized wavelet coefficients of group LL are written into the external SDRAM, computing their minimum value m, maximum value M, the difference Δ and the number of code bits N. After writing all the group-LL coefficients, we read them from the SDRAM and perform our fixed-length coding. The qunatized wavelet coefficients of group Non-LL are classified into two categories. While zero-valued-coefficients are fed to run-length coder block, the coding rules in <Table 2> are applied to non-zero-valued-coefficients. The final bit stream is passed to FIFO (first-in first-out) memory block to make an interface to the external chip.

Our lossless coding algorithm was verified by writing the C program. The RTL description was written in Verilog HDL and verified using Verilog-XL 2.8 logic simulator of Cadence Design Systems. We synthesized the gate-level circuit by Design Compiler 1999.02 of Synopsys, Inc. For prototype verification, Virtex XCV800-4 HQ240 C FPGA of Xilinx was used. After prototyping, we synthesized the gate-level circuit with 0.5μm Standard Cells Library STDM85 of Samsung Electronics Company. 16,118 equivalent two-input NAND gates were used in the gate-level circuit of the CODEC module: 6,028 gates in encoder and 10,090 gates in decoder. SADAS of Samsung Electronics Company calculated pre-layout and post-layout propagation delays for timing verification. The CODEC module was merged with DWT, inverse DWT, quantization and un-quantization modules. For testability enhancement, partial scan chains were inserted through Test Compiler of Synopsys, Inc. and the fault coverage of 93% was achieved. For placement and routing, Apollo v99.2 of Avant! was used. The layout is shown in <Fig. 8>. The final gate count is 60,027 and chip size is 4362 × 4262μm^2. The chip was fabricated with 0.5μm standard cell manufacturing process of Samsung Electronics Company and packaged in 100-TQFP. We assembled a PCB (printed circuit board), shown in <Fig. 9>, and confirmed that the resulting chip compresses and decompresses the images from CMOS sensor correctly with the main clock of 27 MHz.

In order to compare the performance of our coding algorithm with the original Huffman coding algorithm, we experimented on Lena image with 256

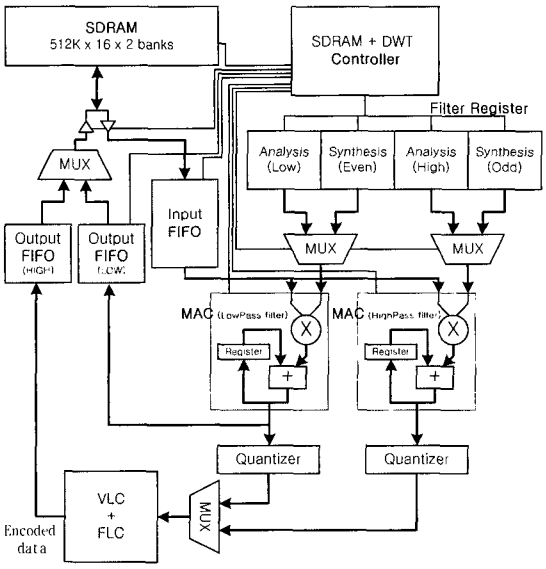

그림 6. 이차원 웨이블릿 기반 영상압축기 회로도
Fig. 6. Block diagram of two-dimensional wavelet-based image compression circuit.

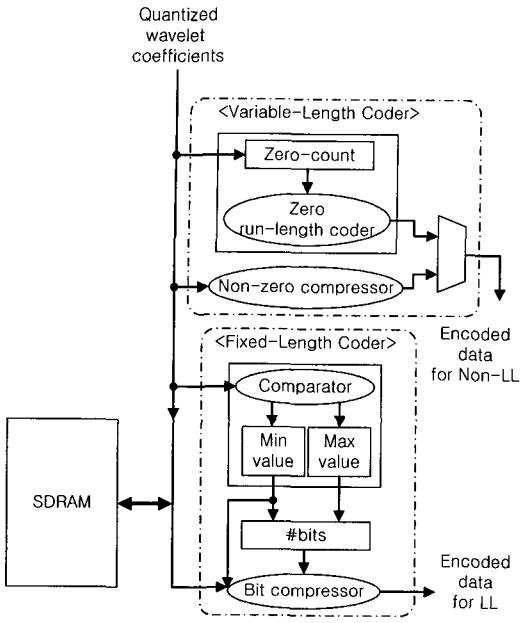

그림 7. 무손실 부호화기 회로도
Fig. 7. Block diagram of the lossless encoder module.

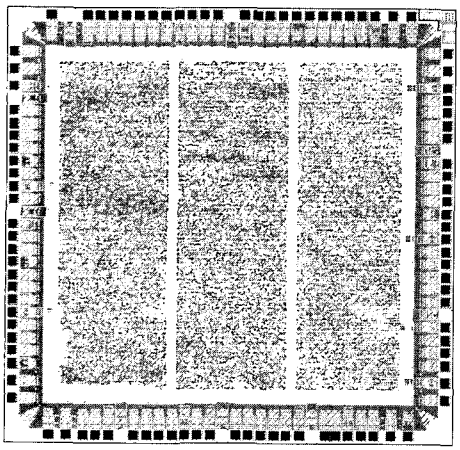

그림 8. 영상 압축 및 복원 ASIC 칩의 설계도면
Fig. 8. Layout of image compression and decompression ASIC chip.

× 256 pixels in <Fig. 5>. The run-length coding for zero-valued coefficients was added to the original Huffman coding. We applied the same DWT, quantization and zero-valued run-length coding to both of the the original Huffman coding and ours. As shown in <Table 3>, the compression ratio of our algorithm is better than that of the original Huffman coding for all combinations of bin size

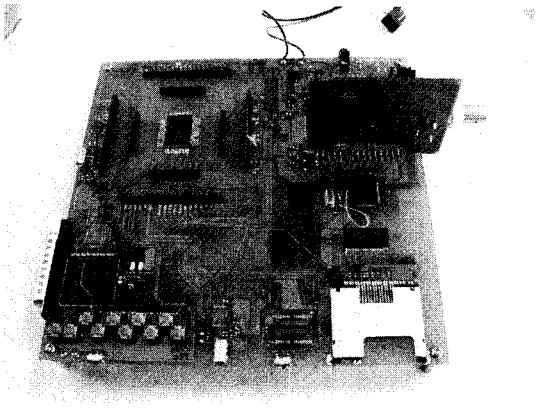

그림 9. 실험에 사용된 PCB 사진
Fig. 9. Photo of PCB used in the experiments.

options. The bin size options are defined in <Table 1>. This is due to the fact that our algorithm does not require any information on the Huffman tree because it uses fixed coding rules in <Table 2>. <Table 4> illustrates the comparison of the compression ratio for other image samples. In this experiment, bin size option 2 was used.

V. Conclusions

In the image compression procedures, lossless coding is to reduce the number of code bits by the elimination of coding redundancy. Huffman coding, which is the most popular variable-length coding, is known as optimal, but the problem is that it requires an ordering of all source symbols according to their probability of occurrences. It requires excessive data access and computational efforts. In this paper, we proposed a new and efficient lossless coding algorithm for quantized wavelet coefficients. Our algorithm is based on the observation that the wavelet coefficients have Laplacian distribution characteristics with a sharp peak at value zero except the lowest-frequency region. The lowest-frequency coefficients are coded by fixed-length coding method. Other coefficients are classified into two categories. Zero-valued-coefficients are fed into the run-length coder and the fixed coding rules are applied to non-zero-valued coefficients. The

표 2. Non-LL 그룹에서 0이 아닌 계수에 대한 부호화 규칙

Table 2. Coding rules for non-zero-valued coefficients of group Non-LL.

Code		$V_{NONZERO}$	#Bits	$V_{NONZERO}$	Code	
Size	Value				Size	Value
11	0	-1	3	+1	11	1
101	0	-2	4	+2	101	1
1001	00 10	-3 -4	6	+3 +4	1001	01 11
10001	000 010 100 110	-5 -6 -7 -8	8	+5 +6 +7 +8	10001	001 011 101 111
100001	0000 0010 ~ 1100 1110	-9 -10 ~ -15 -16	10	+9 +10 ~ +15 +16	100001	0001 0011 ~ 1101 1111
1000001	00000 00010 ~ 11100 11110	-17 -18 ~ -31 -32	12	+17 +18 ~ +31 +32	1000001	00001 00011 ~ 11101 11111
10000001	000000 000010 ~ 111100 111110	-33 -34 ~ -63 -64	14	+33 +34 ~ +63 +64	10000001	000001 000011 ~ 111101 111111
100000001	0000000 0000010 ~ 1111100 1111110	-65 -66 ~ -127 -128	16	+65 +66 ~ +127 +128	100000001	0000001 0000011 ~ 1111101 1111111
1000000001	00000000 00000010 ~ 11111010 11111100	-129 -130 ~ -254 -255	18	+129 +130 ~ +254 +255	1000000001	00000001 00000011 ~ 11111011 11111101

표 3. 256 × 256 레나 영상에 대한 압축비 비교

Table 3. Comparison of compression ratios for 256 × 256 Lena image.

	Compression ratio		PSNR (dB)
	Huffman coding	Our coding	
Bin size option 1	5.060	6.044	35.578
Bin size option 2	9.955	12.803	30.649
Bin size option 3	18.210	24.273	27.717
Bin size option 4	29.415	41.063	25.725

표 4. 다양한 영상 예제에 대한 압축비 비교

Table 4. Comparison of compression ratios for various image samples.

Image samples		Compression ratio		PSNR (dB)
Size	Name	Huffman coding	Our coding	
256 × 256	Barbara	6.913	8.044	28.131
	Goldhill	9.779	12.365	28.493
640 × 480	Zebra	21.603	29.860	35.826
640 × 240	Flower	13.195	16.587	31.055
	House	6.355	6.615	25.459
	Rain	21.378	25.549	32.333
	Sky	15.717	19.423	29.438
	Snow	9.933	11.633	27.339

coding rules do not require any code table or Huffman tree information. The advantage of our algorithm is its simplicity, which makes it possible to implement a chip with a small number of gates while it provides competitive compression ratio, compared to the original Huffman coding algorithm. Our algorithm was implemented as an ASIC chip and verified by compressing and decompressing the images from CMOS sensor successfully.

REFERENCES

[1] S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674~693, July 1989.

[2] David S. Taubman and Michael W. Marcellin, JPEG 2000: *Image Compression Fundamentals, Standards and Practice*, KAP, 2002.

[3] Mehran Aminian and Farzan Aminian, "Neural-network based analog-circuit fault diagnosis using wavelet transform as preprocessor," *IEEE Trans. on Circuit and Systems-II: Analog and Digital Processing*, vol. 47, no. 2, pp. 151~156, Feb. 2000.

[4] Guoliang Fan and Xiang-Gen Xia, "Improved hidden markov models in the wavelet domain,"

IEEE Trans. on Signal Processing, vol. 49, no. 1, pp. 115~120, Jan. 2001.

[5] Nick Soveiko and Michel S. Nakhla, "Efficient capacitance extraction computations in wavelet domain," *IEEE Trans. on Circuit and Systems-I : Fundamental Theory and Applications*, vol. 47, no. 5, pp. 684~701, May 2000.

[6] Aleksandra Mojsilovic, Miodrag V. Popovic and Dejan M. Rackov, "On the selection of an optimal wavelet basis for texture characterization," *IEEE Trans. on Image Processing*, vol. 9, no. 12, pp. 2043~2050, Dec. 2000.

[7] D. A. Huffman, "A method for the construction of minimum-redundancy codes," *Proc. of IRE*, vol. 40, pp. 1098~1101, 1952.

[8] S. Lee, K. Cho, and S. Hong, "Design of a two-dimensional discrete wavelet transform core for image compression," *Journal of Korean Physics Society*, vol. 38, no. 3, pp. 224~231, Mar. 2001.

[9] D. E. Thomas and P.R. Moorby, *The Verilog Hardware Description Language*, KAP, 1996.

[10] Samsung Electronics Co., Ltd., *0.5㎛ High Density CMOS Standard Cell Library Data Book*, June 1997.

[11] I. Daubechies, "*Orthonormal bases of compactly supported wavelets,*" *Communication Pure Applied Math.*, vol. 41, pp. 909~996, Nov. 1988.

[12] Samsung Electronics Co., Ltd., *KM416S1120D : 1M × 16 SDRAM*, Revision 1.4, June 1999.

[13] M. Vishwanath, R. M. Owens and M. J. Irwin, "VLSI architectures for the discrete wavelet transform," *IEEE Trans. on Circuits and Systems-II : Analog and Digital Signal Processing*, vol. 42, no. 5, pp. 305~316, May 1995.

[14] A. Grzeszczak, M. K. Mandal, S. Panchanathan and T. Yeap, "VLSI implementation of discrete wavelet transform," *IEEE Trans. on VLSI Systems*, vol. 4, no. 4, pp. 421~433, Dec. 1996.

저 자 소 개

李 宣 永(正會員) 第37卷 SD編 第1號 參照
현재 : 한국외국어대학교 전자정보공학부 박사과정

趙 敬 淳(正會員) 第37卷 SD編 第1號 參照
현재 : 한국외국어대학교 전자정보공학부 교수