

# Hybrid vector quantization

Kuei-Ann Wen  
Chung-Yen Lu  
National Chiao Tung University  
Institute of Electronics  
1001 Ta Shieh Road  
Hsinchu 300, Taiwan

**Abstract.** Vector quantization (VQ) is a powerful technique for low-bit-rate image coding. However, initial studies of image coding with VQ have revealed that VQ causes degradations, most notably around edges. Moreover, the computational complexity is high. Although a few algorithms have been developed to reduce edge degradation, such as block truncation coding (BTC) with VQ (BTC/VQ) or classified vector quantization, their compression ratios are not satisfactory. Discrete cosine transformation with VQ (DCT/VQ) has been applied to image compression, showing a high compression ratio, but the edge degradation problem still exists. We present an image compression algorithm that takes advantage of the merits of DCT/VQ and BTC/VQ to achieve a high-quality and low-bit-rate compression of images. High quality images can be achieved at rates of 0.34 to 0.46 bit/pixel.

*Subject terms:* visual communication; hybrid vector quantization; block truncation coding; discrete cosine transform.

*Optical Engineering* 32(7), 1496–1502 (July 1993).

## 1 Introduction

Natural images can be segmented into regions with widely varying perceptual importance. There are three types of regions in a typical image. The region in which the contrast between objects and background is high is categorized as the *edge*. Due to the high contrast, human eyes will naturally pay much attention to the edges. Thus, edge regions are very important for human perception. The second type of region is the *smooth* area, in which the brightness of neighboring pixels changes slowly. The third type of region is the *texture* part. Pixels in the texture region change slower than the edge region but faster than the smooth region. Typically, in a photograph such as “Lena” [see Fig. 5(a)], the boundary of the face is categorized as the edge part, the cheek is the smooth part, and the hair is viewed as the texture part. We propose a new and simple method for classifying the edge and smooth parts; and then we encode the image using block coding with discrete cosine transform/vector quantization (DCT/VQ)<sup>1</sup> and the modified block truncation coding/vector quantization (BTC/VQ) derived from BTC/VQ.<sup>2</sup>

In our coding algorithm, we use block coding and, according to the eye model, treat the texture part as a composition of some smooth blocks. We then devote our attention

to the coding of the edge part to maintain the high quality of the image. Thus, more bits are allocated for edge regions to reduce the edge degradation. On the other hand, the major compression of the image is dependent on coding the smooth part. Block coding algorithms are usually applied to one of three rudimentary coding techniques: transform coding (TC),<sup>3</sup> vector quantization (VQ),<sup>4</sup> and block truncation coding (BTC).<sup>5</sup> In each scheme, the block size is a parameter. The block size of TC and BTC are usually  $8 \times 8$  and  $4 \times 4$ , respectively. The block sizes of VQ may vary for different applications. We chose a block size of  $8 \times 8$  for the DCT and  $4 \times 4$  for the BTC, and block sizes such as  $3 \times 1$  and  $4 \times 4$  were used for different applications. We proposed a hybrid VQ algorithm that combines DCT, BTC, and classified VQ (CVQ) to classify edges from smooth parts.

## 2 Hybrid VQ Algorithm

The hybrid VQ algorithm is illustrated in Fig. 1. The smooth part and the edge part are first distinguished and then coded with different approaches. Each  $8 \times 8$  block is first transformed with a DCT to obtain important coefficients. Because we do not need all the coefficients in the DCT domain, we use a technique, called the partial discrete cosine transform (PDCT), to obtain four coefficients. These four coefficients represent the four lowest frequency coefficients, denoted as  $d(0,0)$ ,  $d(0,1)$ ,  $d(1,0)$ , and  $d(1,1)$ , respectively. Two of these four coefficients are used to classify each  $8 \times 8$  block into either a smooth region or edge region. The decision is made

Paper VCI-20 received Oct. 15, 1992; revised manuscript received Feb. 18, 1993; accepted for publication March 11, 1993.  
© 1993 Society of Photo-Optical Instrumentation Engineers. 0091-3286/93/\$2.00.

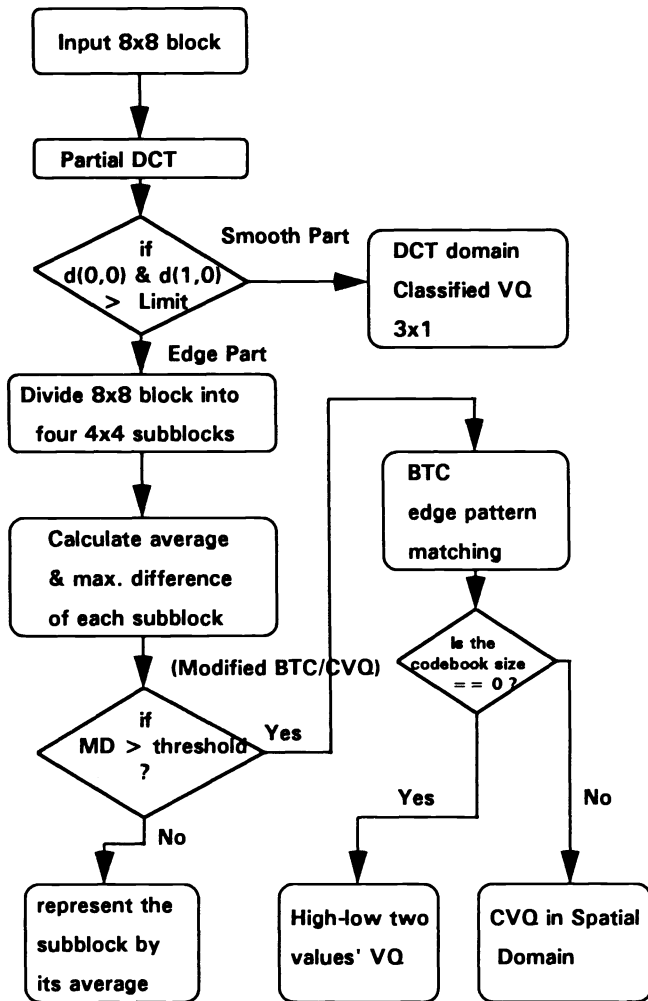


Fig. 1 Flowchart of hybrid VQ: threshold preset at 85 and edge region marked by the white line.

by comparing  $d(0,1)$  and  $d(1,0)$  with a threshold value. If both of them are less than the threshold value, the block is classified as a smooth block; otherwise, it is an edge block. Because these two coefficients represent the amount of brightness change in the horizontal and vertical direction, we choose the threshold experimentally. Typically, the threshold value is 85, which is determined by a set of training images shown in Fig. 2.

For smooth blocks, we use CVQ<sup>6</sup> to code three DCT coefficients, that is,  $d(0,1)$ ,  $d(1,0)$ , and  $d(1,1)$ . For  $d(0,0)$ , 8 bits are allocated.

For edge blocks, we divide the  $8 \times 8$  block into four  $4 \times 4$  subblocks,<sup>7</sup> and process them sequentially. As illustrated in Fig. 1, the process for the  $4 \times 4$  subblocks is denoted as modified BTC/VQ. The maximum difference, which is defined as the difference between the maximum and the minimum intensities in the subblock, is found first. Then, we compare the maximum difference with a preset threshold value, typically 18 (see Ref. 8), to determine whether or not the subblock is uniform (smooth). If the subblock is uniform, we use its average to represent all the intensities in the subblock; otherwise, we use the BTC technique to determine its edge pattern and compare the edge pattern with the edge pattern

table (shown in Table 1) to choose the nearest pattern. The edge pattern table represents normal edge patterns in normal images. Then, according to the index of its edge pattern, the VQ code book size is found by looking up a size table. The size table is obtained by calculating the frequency of the edge patterns in the training set images. The size is not fixed—its value may be 0, 16, 32, or 64. If the size is zero, it means that the edge pattern is seldom used. Thus, the  $4 \times 4$  edge subblock is coded with VQ by using the two output values of BTC. Otherwise, we use VQ to code the subblock.

### 2.1 Smooth Area Processing

For processing the smooth region, we use the CVQ proposed in Ref. 6 to encode the three DCT coefficients. We classify the  $8 \times 8$  block according to the sign of these three coefficients. In total, we have eight subcode books (+ + +, + + -, + - +, + - -, - + +, - + -, - - +, - - -). Each code book size is 128 code words. We send the dc value, namely  $d(0,0)$ , with eight bits. Finally, we code the smooth block by 19 bits in total, which is the sum of the decision bit (1 bit), dc value (8 bits), classifying index (3 bits), and VQ index (7 bits) (e.g.,  $19 = 1 + 8 + 3 + 7$ ).

### 2.2 Edge Area Processing

The goal in processing the edge region is to obtain high-quality compressed images with edge preservation. Consider a  $4 \times 4$  subblock of an image. It is desired that the  $4 \times 4$  block be placed in one of two categories: either in the areas in which the intensities of the pixels in the block are relatively uniform or the block contains an edge.

Coding the image efficiently, without introducing unreasonable distortion, requires an auxiliary method for each case. For the first case, the block is represented by its average (mean), with 10 bits in total representing the sum of the decision bit (1 bit), decision bit (1 bit), and the bits representing the mean value (8 bits). For the second case, the block is coded with the modified BTC/VQ. The test to determine whether or not a block is relatively uniform (smooth) can be performed by using the mean (average) and the deviation, which is the difference between the maximum and the minimum intensities in the  $4 \times 4$  block. This is compared to a preset threshold usually in the range of 13 to 50 (see Ref. 8). We set it to be 18. Higher threshold values result in more blockiness in the image. For blocks containing edges, we use look-up tables to represent cases of edges, in which the 16 binary bits of BTC are classified into 64 edge patterns. If the size is not equal to zero, we send the  $4 \times 4$  block to the CVQ coder. The total number of bits for CVQ block coding is equal to the sum of the decision bit (1 bit), decision bit (1 bit), classification index (6 bits), and VQ index (4 to 6 bits). Note that only 32 cases are used for CVQ. The others are coded with high/low VQ that utilizes the high and low levels from the BTC method. In this case, the total bits for high/low block coding is equal to the sum of the decision bit (1 bit), decision bit (1 bit), classification index (6 bits), and high/low VQ index (8 bits). As a result, the total bits for an image = the number of smooth blocks ( $8 \times 8$ ) \* bits of a smooth block coding + number of mean blocks ( $4 \times 4$ ) \* bits of mean block coding + number of BTC/CVQ blocks \* bits of a BTC/CVQ block coding + number of high/low blocks \* bits of high/low block coding.

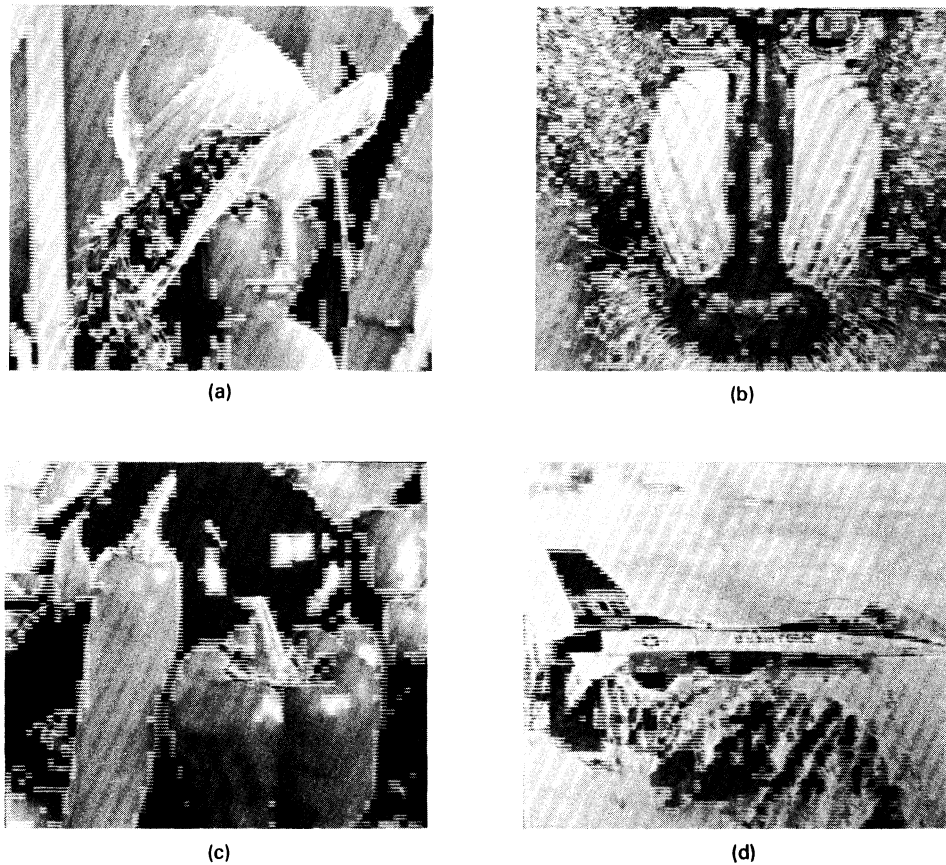


Fig. 2 Experimental images for the classification of the smooth and edge regions.

### 3 Elements of Hybrid VQ

Some key elements of the hybrid VQ, including PDCT, BTC, CVQ, and code-book training are described below.

#### 3.1 Partial Discrete Cosine Transform

The partial discrete cosine transform (PDCT) is used to obtain the most important DCT coefficients. For 1-D PDCT, where  $d0$  and  $d1$  are the first two coefficients in the 1-D DCT domain:

$$d0 = \frac{\sqrt{2}}{4} * \sum_{k=0}^7 I_k$$

$$d1 = \sum_{k=0}^7 C_k * I_k,$$

where  $I_k$  ( $k=0..7$ ) is the pixel intensity and  $C_k$  ( $k=0..7$ ) is the 1-D DCT transform coefficient.

Each row of an  $8 \times 8$  block is transformed with a 1-D PDCT. Then, the 1-D PDCT is applied to the first two columns. This procedure is called the 2-D PDCT, and is shown in Fig. 3.

#### 3.2 BTC and Edge Pattern Table

BTC encodes every  $4 \times 4$  block of the image by its average,

Table 1 Different edge patterns.

1110	1111	0011	0111	1111	1111	1100	0011
1100	1110	0001	0011	1111	1111	1100	0011
1000	1100	0000	0001	1100	0011	1111	1111
0000	1000	0000	0000	1100	0011	1111	1111
0011	0001	1100	1000	1111	0001	1000	1111
0110	0011	0110	1100	1000	0001	1000	0001
1100	0110	0011	0110	1000	0001	1000	0001
1000	1100	0001	0011	1000	1111	1111	0001
1111	1111	1111	0001	1111	1111	1111	1001
0000	1111	1111	0001	0011	1100	1111	1001
0000	0000	1111	0001	0011	1100	1001	1111
0000	0000	0000	0001	1111	1111	1001	1111
0011	0111	0110	0000	0010	0000	0100	0000
0011	0111	0110	1111	0100	0001	0010	1000
0011	0111	0110	1111	1000	0010	0001	0100
0011	0111	0110	0000	0000	0100	0000	0010

its variance, and by 16 binary bits. Each bit indicates whether or not the original intensity at a pixel is above the average intensity value. The two levels A and B are the averages of all the pixels whose intensities are above and below the mean, respectively. According to the 16 binary bits, appropriate VQ is found for each  $4 \times 4$  block by matching the 16-bit pattern with an edge pattern table. The edge pattern table is shown in Table 1, which lists half of all the edge patterns. The other half is complementary to Table 1.

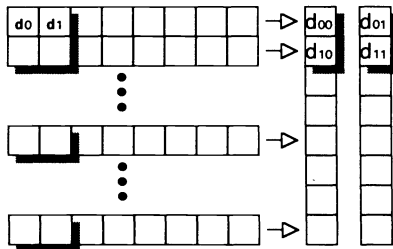


Fig. 3 Structure of the 2-D partial DCT.

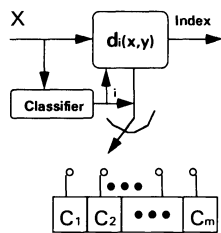


Fig. 4 Classified VQ.

### 3.3 Classified VQ

The CVQ coder is depicted in Fig. 4. There are  $M$  classes. If the input  $X$  belongs to class  $i$ , the  $i$ 'th subcode book  $C_i$  of size  $N_i$  is employed to encode  $X$ , by using the distortion measure  $d_i(\cdot)$ . In the hybrid VQ algorithm, we have two types of CVQ: one is  $3 \times 1$  in the DCT domain and the other is  $4 \times 4$  in the spatial domain. The classification for the former is decided by the signs of the three DCT coefficients. In this case,  $M=8$  and the size  $N_i$  of each subcode book  $C_i$  is 128. The latter is classified by the BTC 16 binary bits with an edge pattern table. There are 64 edge patterns, but  $M=32$  classes for this case. This is because some edge patterns are seldom used and are processed with block truncation coding with high/low VQ. The other patterns are fed to CVQ with the subcode book size varying from 16 to 64.

### 3.4 VQ Code-Book Training

The LBG algorithm<sup>9</sup> with least-squares-error measurement is used to train the VQ code book. The training image set is shown in Fig. 5.



(a)



(b)



(c)



(d)

Fig. 5 Training set images ( $512 \times 512$ ): (a) "Lena," (b) "baboon," (c) "peppers," and (d) "jet1."

**Table 2** Results of training images coded by HVQ.

	PSNR	Compression rate	Total bits of encoded image	Amount of Smooth blocks (8x8)	Amount of edge subblocks (4x4 each)		
					Mean	High-Low	CVQ
Lena	29.52	20.10 (0.3979bpp)	104320	3196	506	116	2978
Baboon	21.74	17.69 (0.4523bpp)	118556	2756	101	512	4747
Paper	28.96	19.70 (0.4062bpp)	106476	3096	648	143	3209
Jet1	29.06	19.57 (0.4087bpp)	107138	3066	666	171	3283

**Table 3** Results of nontraining images coded by HVQ.

	PSNR	Compression rate	Total bits of encoded image	Amount of Smooth blocks (8x8)	Amount of edge subblocks (4x4 each)		
					Mean	High-Low	CVQ
Rocket	33.05	23.39 (0.3421bpp)	89676	3667	462	86	1168
Bicycle	31.78	22.28 (0.3590bpp)	94116	3506	610	101	1649
Beach	29.89	20.99 (0.3811bpp)	99902	3327	455	165	2456
Vocano	29.24	23.73 (0.3371bpp)	88370	3739	170	107	1151
Model	28.56	20.56 (0.3891bpp)	102009	3287	244	218	2774
Girl	28.06	21.92 (0.3650bpp)	95682	3488	339	140	1953

**4 Experimental Results**

The peak signal-to-noise ratio (PSNR) and the compression rates are shown in Tables 2 and 3. The formula of PSNR is

$$PSNR = 10 \log_{10} \frac{255^2}{\text{Mean Square Error}}$$

The original images are shown in Figs. 5 and 6 and the compressed images are shown in Figs. 7 and 8. These show that, without entropy coding and with purely hybrid VQ, the image quality at bit rates ranging from 0.34 to 0.46 bit/pixel is high.

**5 Conclusion**

The hybrid VQ algorithm is a technique combining DCT, BTC, and CVQ for image compression. This algorithm takes advantage of DCT/VQ and modified BTC/VQ to achieve a high compression ratio and images of high-quality. The key element is the classification of the edge and smooth parts. We use the characteristics of each part to obtain adequate compression so that there are more bits for the edge region and less bits for the smooth region. The results from the hybrid VQ are shown at bit rates ranging from 0.34 to 0.46 bit/pixel with high quality.



**Fig. 6** Nontraining images (512 × 512): (a) “rocket,” (b) “bicycle,” (c) “beach,” (d) “volcano,” (e) “model,” and (f) “girl.”

HYBRID VECTOR QUANTIZATION

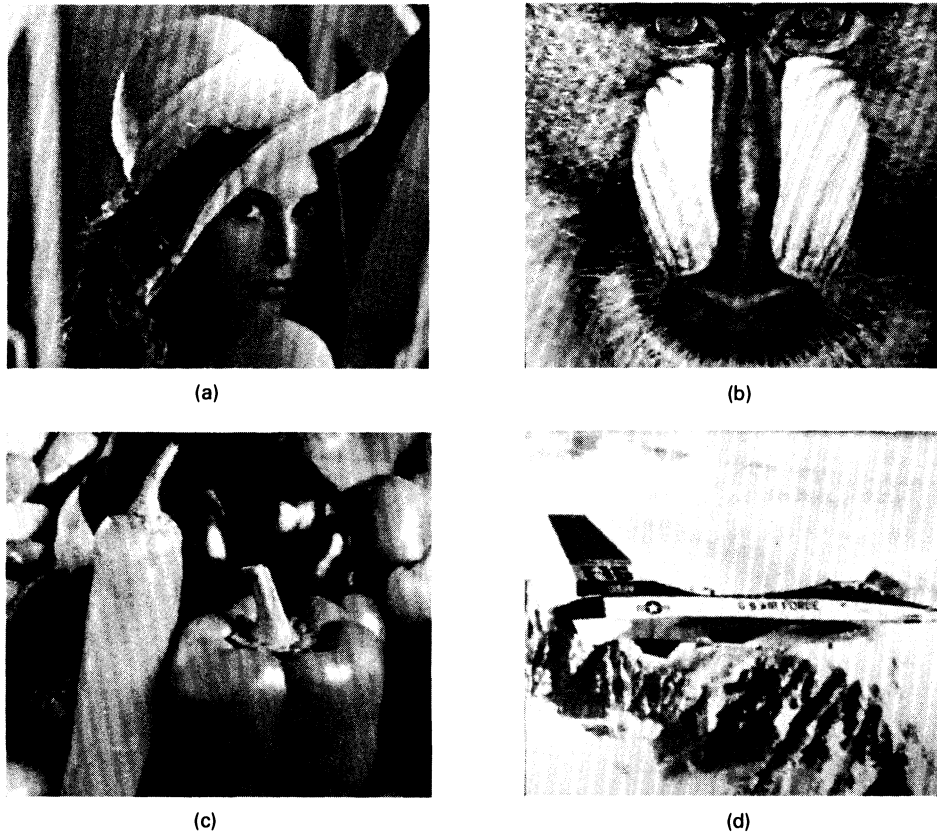


Fig. 7 Compression images: (a) "Lena," (b) "baboon," (c) "peppers," and (d) "jet1."



Fig. 8 Compression images: (a) "rocket," (b) "bicycle," (c) "beach," (d) "volcano," (e) "model," and (f) "girl."

### Acknowledgment

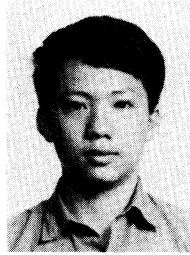
This work was sponsored by the National Science Council under the project "VLSI Design for HDTV Signal Processing."

### References

1. T. Satio, H. Takeo, K. Aizawa, H. Harashima, and H. Miyakawa, "Adaptive discrete cosine transform image coding using gain/shape vector quantizers," in *Proc. ICASSP '86*, pp. 129-132 (1986).
2. N. Efrati, H. Licitin, and H. B. Mitchell, "Classified block truncation coding-vector quantization: an edge sensitive image compression algorithm," *Sig. Process. Image Commun.* **3**, 275-283 (1991).
3. R. Clarke, "Transform coding of images," Academic Press, New York (1985).
4. N. M. Nasrabadi and R. A. King, "Image coding using vector quantization: a review," *IEEE Trans. Commun.* **COM-27**, 957-971 (1979).
5. E. J. Delp and R. Mitchell, "Image compression using block truncation coding," *IEEE Trans. Commun.* **8**, 1245-1254 (Aug. 1991).
6. B. Ramamurthi and A. Gersho, "Classified vector quantization of images," *IEEE Trans. Commun.* **COM-34**(11), 1105-1115 (Nov. 1986).
7. J. Vaisey and A. Gersho, "Image compression with variable block size segmentation," *IEEE Trans. Sig. Process.* **8**, 2040-2060 (Aug. 1992).
8. P. Nasiopoulos, R. K. Ward, and D. J. Morse, "Adaptive compression coding," *IEEE Trans. Commun.* **8**, 1245-1254 (Aug. 1991).
9. Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.* **COM-28**(1), 84-95 (Jan. 1980).



**Kuei-Ann Wen** received the BEE, MEE, and PhD from the Department of Electrical Engineering at the Institute of Electrical and Computer Engineering at National Cheng Kung University, Tainan, Taiwan, in 1983, 1985, and 1988, respectively. She is presently a professor in the Department of Electrical Engineering, National Chiao Tung University, Hsinchu, Taiwan, where she has joined the Center for Telecommunication Research. Her current teaching and research interests are in the areas of high-speed digital signal processing, parallel processing and VLSI circuit design, and visual signal processing.



**Chung-Yen Lu** received the BS degree in electronics engineering from National Chiao Tung University, Taiwan, in 1991, and he is now a graduate student at the Institute of Electronics at the same university. His research interests include image processing and video compression.