A Novel Non-Iterative Scheme for Fractal Image Coding

HSUAN T. CHANG AND CHUNG J. KUO*

Department of Information Management
Chaoyang University of Technology
Taichung, Taiwan 413, R.O.C.
*Signal and Media (SAM) Laboratory
Department of Electrical Engineering
National Chung Cheng University
Chia-yi, Taiwan 621, R.O.C.

Iteration in conventional fractal coding schemes not only leads to a high computation complexity, but also requires a large amount of memory. It unavoidably prolongs the decoding process and precludes the use of high speed applications. To overcome these problems, we propose a non-iterative method based on a novel domain pool design for fractal image coding. The domain pool used in the encoder is on-line transmitted to the decoder. Therefore, the domain blocks are generated from the same mean image existing in both the encoder and decoder. We next utilize contractive affine transformations to encode/decode the image with iterations. From the simulation results, we have successfully speeded up the decoding process, and the coding performance for the test images is good as or even better than that of the conventional schemes.

Keywords: non-iterative, fractal image coding, on-line transmission, domain pool, contractive affine transformation

1. INTRODUCTION

The fractal coding scheme is a new technique for image compression and has evolved greatly from the first version proposed by Jacquin [1, 2]. In conventional fractal coding schemes, an image is partitioned into non-overlapping range blocks. The larger domain blocks are selected from the same image and can overlap. A grayscale image is encoded by mapping the domain block $D$ to the range block $R$ with the contractive affine transformation [2]

$$
\hat{R} = l(\alpha \cdot (S \circ D) + \Delta g),
$$

where $S^o$ represents the contraction operation that maps the size of a domain block to the size of a range block. Then, the parameters (called the fractal code) describing the contractive affine transformation that has the minimum matching error between the original range block $R$ and the coded range block $\hat{R}$ are transmitted or stored. The fractal code
consists of the contrast scaling $\alpha$, luminance shift $\Delta g$ or block mean (the average pixel value of the range block) $\mu_R$ [3], isometry $\iota$, and the position $P_D$ of the best-matched domain block in the domain pool. In the decoding stage, an arbitrary image is used as the initial image, and the decoded image is repeatedly reconstructed by applying contractive affine transformation to the iterated image. The iteration process does not stop until a predefined convergence criterion is satisfied. Obviously, the complicated iteration process in the decoding stage is a main drawback in fractal coding techniques.

Many modified versions have been proposed to improve the fractal coding techniques [4-10]. However, only a few fast algorithms [11-15] have been proposed to speed up the decoding process. These fast algorithms can only reduce the iteration number or perform non-iterative decoding in special cases. It is, thus, desirable to find a universal scheme that is non-iterative in the decoding stage for fractal image compression, such that, the decoding speed can be greatly improved. Furthermore, limitations due to iterations, for example, contractivity, excessive computation burden, serial block processing, and the large amount of memory required for decoding, will be relaxed.

In this paper, we propose a fractal coding scheme that is non-iterative in the decoding stage. Since the block mean can be used as one of the parameters in the fractal code in a modified contractive affine transformation [3], we can generate the domain pool by using the mean image whose pixel values are the block means of all the range blocks in the encoder. The decoder receives the on-line transmitted mean information, and as a result, an identical domain pool can be obtained. We generate the domain blocks that are directly selected from the mean image. The size of the domain block is the same as that of the range block. The domain blocks can be chosen from the mean image with a sampling period or from the blocks nearest the pixel corresponding to the mean of the coded range block. Therefore, the encoding and decoding results are exactly identical since we use the same domain blocks in both the encoder and decoder. Simulation results show that the decoding speed is greatly improved while the coding performance for the test images is as good as or even better than that of the conventional fractal coding schemes that require iterations.

The organization of this paper is as follows: We first introduce in Section 2 the conventional fractal coding scheme that uses an iterative decoding process. Section 3 describes the proposed non-iterative fractal image codec, which uses on-line domain pool transmission. The computer simulation described in Section 4 verifies the improvement offered by the proposed non-iterative scheme. Finally, a conclusion is given in Section 5.

### 2. CONVENTIONAL FRACTAL CODING SCHEME

The proposed non-iterative scheme is compared here with the conventional fractal coding schemes that require iterations. The domain pool designs in conventional fractal coding schemes are described in the following: Basically, the domain blocks are selected from the original image, and the block size is four times the size of the range block. We investigate here the coding performance of the conventional fractal coding scheme that uses two methods to design the domain pool. First, the domain pool consists of the domain blocks subsampled from the original image, which is called the ‘subsampling’ method. For an image of size $M \times M$, the sampling period ($T$) in both the horizontal and vertical directions is determined by
\[
T = \left\lfloor \frac{M - B}{\sqrt{N_D} - 1} \right\rfloor, \quad T \geq 1,
\]

where \(B \times B\) is the domain block size, \(N_D\) is the number of domain blocks in the domain pool, and \(\lfloor . \rfloor\) denotes choosing a smaller and the integer closest to the real number in the bracket. Second, we choose \(N_D\) domain blocks neighboring the range block, which is called the ‘neighboring’ method. Then, contractive affine transformation is used to find the fractal code for each range block.

In decoding stage, the decoded image is iteratively reconstructed using the contractive affine transformations denoted in the fractal codes. Since the initial image in the decoder is different from the original image in the encoder, the domain blocks in the encoder are different from that found in the decoder. There exists a distortion between the encoded image in the encoder and the decoded image in the decoder. To reduce this distortion, it is desirable to generate the same domain pool in both the encoder and decoder. The proposed non-iterative scheme can solve this problem.

The criterion for the decoded image to achieve convergence is determined as follows: Let the \(n\)th iterated image be denoted as \(f^{(n)}\). The average error \(e(n)\) between the \(n\)th and \((n-1)\)th decoded images is calculated by

\[
e(n) = \frac{1}{512^2} \sum_{i=1}^{512} \sum_{j=1}^{512} (f_{i,j}^{(n)} - f_{i,j}^{(n-1)})^2,
\]

where \(f_{i,j}^{(n)}\) denotes the gray level of the \((i,j)\)th pixel in \(n\)th decoded image. If the ratio

\[
\gamma = \frac{|e(n) - e(n-1)|}{e(n-1)}
\]

is less than a threshold value \(\gamma_0\), then the decoded image converges, and the iteration process terminates. Otherwise, the iteration process does not stop until the convergence criterion \(\gamma \leq \gamma_0\) is satisfied.

### 3. NON-ITERATIVE DESIGN

#### 3.1 Encoder

The flow chart of the encoder in the proposed non-iterative scheme is shown in Fig. 1. The input \(M \times M\) image is partitioned into non-overlapping range blocks of size \(B \times B\). First of all, we sequentially measure the mean and variance of each range block. After all the means of the range blocks are obtained, we can generate a mean image of size \(MIB \times MIB\) with each pixel corresponding to the block mean. If the variance of the range block,

\[
\text{Var}[R] = \sum_{0 \leq i, j < B} (r_{i,j} - u_R)^2,
\]

(where \(r_{i,j}\) denotes the gray level of the \((i,j)\)th pixel in the range block) is smaller than the threshold value \(E_{th}\), then the range block is coded by the block mean. Otherwise, the
Fig. 1. The flow charts of the encoder in the proposed non-iterative fractal coding scheme.

range block is coded using affine transformation. Note that in this case, the size of the mean image should be much larger than the size of the domain block, \( \frac{M}{B} \times \frac{M}{B} \gg B \times B \). Otherwise, it will not be easy to find a good mapping between the domain and range blocks because it will be possible to take only a few domain blocks from the mean image. The size of the domain block is the same as that of the range block; thus, the contraction procedure in conventional fractal coding schemes is eliminated. We, therefore, proceed with new contractive affine transformation between the range block and the domain blocks generated from the mean image.

There are two methods which can be used to design the domain pool in the proposed non-iterative scheme. In the first method, we chose the neighboring blocks of the pixel which correspond to the mean of the range block. On the other hand, the domain blocks are obtained by subsampling the mean image with a sampling period \( T' \), defined as

\[
T' = \frac{M/B - B}{\sqrt{N_D} - 1} \geq 1.
\]
The parameters used in our contractive affine transformation are specified as follows: The luminance shift is replaced by the block mean [3], which is coded by six bits. The contrast scaling can be greater than 1.0 because our scheme is non-iterative. As shown in [16], for some range blocks, if the contrast scaling is greater than one, then the minimum distortion can be achieved between the range block and the transformed domain block. In our design, the contrast scaling is determined from the values in the set \( \{n/4, n = 1, 2, 3, \ldots, 8\} \) so that we can find the one that best minimizes the distortion. We, thus, need three bits to code the contrast scaling. On the other hand, the eight isometries for shuffling the pixels in the block are the same as those in [2] and are coded using three bits.

The new contractive affine transformation can be expressed by

\[
\hat{R} = (\alpha \cdot D + \mu_R - \alpha \cdot \mu_D) = (\alpha \cdot (D - \mu_D) + \mu_R),
\]

where \( \mu_D \) is the mean of domain block. Note that in Eq. (7) the contraction procedure is eliminated and the term \( \mu_D - \alpha \cdot \mu_D \) is equivalent to the luminance shift shown in Eq. (1).

To determine the fractal code for the original range block \( R \), we test each of the domain block \( D \) in the domain pool. For each domain block substituted into Eq. (7), we use eight contrast scaling factors and eight isometries to obtain all possible results for the transformed block \( \hat{R} \). After testing all the combinations of different domain blocks, contrast scaling factors, and isometries in Eq. (7), the fractal code is determined such that the coded block \( \hat{R} \) has the minimum distortion compared to the original range block \( R \).

The distortion between the original and coded range blocks is represented by the mean-squared-error (MSE) measurement, defined as

\[
\text{MSE}(R, \hat{R}) = \frac{1}{B \times R} \sum_{i,j} (r_{i,j} - \hat{r}_{i,j})^2,
\]

where \( \hat{r}_{i,j} \) denotes the gray level of the \((i,j)\)th pixel in the coded range block. We finally attach a header for each range block to denote its coding status (coded using either mean or affine transformation). Therefore, the decoder can correctly reconstruct each range block according to the header.

### 3.2 Decoder

Fig. 2 shows the flow chart of the decoder in the proposed non-iterative scheme. We first receive all the fractal codes and determine whether or not the range block is coded by the mean from its header. At the same time, the mean image is reconstructed with the on-line transmitted mean information in the fractal codes. Note that this mean image is exactly identical to the mean image used in the encoder since both are constructed using the same block means. Therefore, the domain blocks generated from two identical mean images are also the same; thus, the decoded image is the same as the coded image in the encoder. If the block is coded by the mean, then the value of each pixel in the decoded block is equal to the mean value. Otherwise, we perform the new contractive affine transformation to reconstruct the coded range block. The decoding process ends when contractive transformation is applied only once.

From another point of view, the mean image can be extended to be the initial image
in the conventional decoding algorithm. With the first iteration, the reconstructed image is just identical to the encoded result in the encoder. It's not necessary to proceed with the second iteration because the result in the next iteration will be the same. That is, the decoded image converges in only one iteration. In conventional fractal schemes, the decoding process requires two iteratively refreshed images in the iteration process. On the contrary, only the fixed mean image that can be reconstructed from the received fractal codes is required in our non-iterative scheme. Hence, the required amount of memory in the proposed non-iterative decoder is much less than that in the conventional fractal image decoder. On the other hand, without iterations means that the range blocks can be reproduced without requiring information about the previously iterated images as is needed in conventional schemes. Once the mean image is reconstructed, it provides all the information for each range blocks. Therefore, all of the range blocks can be decoded independently and in parallel. The architectural complexity of the proposed decoder is obviously lower than that of the conventional decoder, which requires iterations. Therefore, the proposed decoder is very suitable for hardware implementation and high speed applications.
4. COMPUTER SIMULATION

In our computer simulation, four 512×512 images (shown in Fig. 3(a)-(d)) with eight-bit grayscale resolution were used to test the proposed non-iterative fractal coding scheme. The performance of the decoded image quality was evaluated based on the peak signal-to-noise-ratio (PSNR) and the bit rate (the required number of bits per pixel). In our simulation, an image was partitioned into range blocks with a single size, either 8×8 or 4×4, or with two-level sizes (both 8×8 and 4×4). Therefore, a general form for the PSNR of the decoded image was defined as

\[
\text{PSNR} = 10 \log_{10} \left( \frac{512^2 \cdot 255^2}{\sum_{i=1}^{N_i} \text{MSE}(R_i, \hat{R}_i) + \sum_{i=1}^{N_i} \text{MSE}(R_i, \hat{R}_i)} \right) \text{ dB},
\]

where \(\text{MSE}(R_i, \hat{R}_i)\) is the mean squared error between the original range block \(R_i\) and the reconstructed range block \(\hat{R}_i\).

Fig. 3. The original images (512 × 512, 8 bit/pixel) used to test the proposed non-iterative fractal coding scheme: (a) Lena, (b) Jetplane, (c) Building, and (d) Harbour.
where \( N_8 \) and \( N_4 \) are the total numbers for the 8x8 range block \( R_8 \) and the 4x4 range block \( R_4 \), respectively. Calculation of the bit rate will be given in the following subsections. For all the schemes used in our simulation, we set the threshold value \( E_{th} \) for the variance of 8 \times 8 \) and 4 \times 4 range blocks to be 1600 and 400, respectively. The size of the domain pool was represented by the number of domain blocks in it. There are four sizes for the domain pool: 16, 64, 256, and 1024 domain blocks used in our simulation. On the other hand, the sampling periods \( T \) and \( T' \), in the conventional domain pool design and our method, were determined using Eqs. (2) and (6), respectively.

4.1 Single Block Size

First of all, the range block with a single size (8 \times 8 or 4 \times 4) was considered. The length of the header \( I_h \) attached to the fractal code for each range block was only one bit (i.e., \( I_h = 1 \)) because it only denoted whether or not the range block was coded by the block mean. Therefore, the bit rate could be calculated as

\[
B_1 = \frac{(N_\mu + N_{af})(I_\mu + I_\alpha) + N_{af}(I_{af} + I_\iota + I_{PD})}{512^2} \text{ bit/pixel,}
\]

for a single block size, where \( I_\mu, I_\alpha, I_\iota, \text{ and } I_{PD} \) denote the required bits for the block mean, contrast scaling, isometry, and the position of the domain pool, respectively. In addition, \( N_\mu \) and \( N_{af} \) denote the number of blocks coded by the block mean and affine transform, respectively.

For an image partitioned into 8x8 range blocks, we measured every block mean and obtained a 64x64 mean image. Fig. 4(a) shows that the mean image was very similar to the original Lena image. We therefore constructed domain pools of different sizes from the mean image. The image was coded by using contractive affine transformation with domain pools of different sizes. Fig. 5(a) shows the simulation results obtained for the Lena image. The numbers shown in the figure represent the different sizes of the domain pool. The bit rates of all the schemes using the same size domain pool were the same. A smaller size for the domain pool resulted in a lower bit rate and PSNR, and vice versa.

![Fig. 4. Two mean Lena images of size (a) 64 \times 64 and (b) 128 \times 128.](image-url)
For the image partitioned into 4×4 range blocks, a 128×128 mean image of Lena was obtained and is shown in Fig. 4(b). We constructed domain pools of different sizes from this mean image. The simulation results for the Lena image based on domain pools of different sizes are shown in Fig. 5(b). The proposed non-iterative scheme had performance similar to that of the conventional fractal coding scheme. The PSNR of the decoded image partitioned into 4 × 4 blocks was much higher than that partitioned into 8×8 blocks since a smaller block size led to a smaller matching error for affine transformation. However, the bit rate increased significantly because the number of 4×4 range blocks was four times the number of 8×8 range blocks.

Fig. 6(a)-(d) shows the graphic results of the decoded images in Fig. 5(b). These four images are based on the proposed non-iterative scheme, which uses a single block size of 4×4 and the subsampling method to choose domain pools of four different sizes. As the size of the domain pool increased, the fidelity of the decoded image increased very much and could be better than that obtained using the conventional fractal scheme.

4.2 Two-Level Block Sizes

From the results shown in Figs. 5(a) and (b), the chosen block size greatly affected the bit rate and the PSNR of the coded image. In order to find a compromise between the bit rate and PSNR for the coded image, it was desirable to partition an image into the range blocks with two-level (parent 8×8 and child 4×4) sizes. An image was first partitioned into parent range blocks, and the coding procedures used were the same as those described in Subsection 4.1. Suppose that the parent range block was coded using contractive affine transformation. If the distortion between the original and the coded range blocks, MSE(\(R_8, \hat{R}_8\)), was greater than the threshold value 1600, then the parent range block was split into four child range blocks. The coding procedures for child range blocks were the same as those described in Subsection 4.1.

Now, the bit rate is affected by the number of the partitioned parent and child range blocks. The more parent range blocks in the coded image, the lower the final bit rate.
If we choose a larger size for the domain pool in the parent level, then more parent range blocks can satisfy the criterion $\text{MSE}(R_8, \hat{R}_8) \leq 1600$. We, thus, chose the maximum domain pool size ($N_D = 1024$) for the parent range block such that the number of coded parent range blocks increased. At the same time, the number of child range blocks was decreased to obtain a lower bit rate. Finally, the size of the child domain pool was varied to examine the PSNR performance of the proposed non-iterative scheme under different bit rates.

To identify the different partitions for the parent range block, we attached a variable-length header to the fractal code. Table 1 shows the header and the bit allocation for the parent range block $R_8$. We assigned ‘0’ as the header of the mean-coded parent range block. For the parent range block coded using affine transformation, ‘10’ was the

Fig. 6. The decoded Lena images based on the proposed non-iterative scheme with different domain pool sizes: (a) 16, (b) 64, (c) 256, and (d) 1024.
Table 1. Header and bit allocation for both parent and child range blocks.
(AT: affine transform)

<table>
<thead>
<tr>
<th>Block type</th>
<th>Header</th>
<th>Bit allocation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_8$ coded by $\mu$</td>
<td>‘0’</td>
<td>$I_1$, $I_{\mu}$, $I_{\alpha}$, $I_{PD}$</td>
</tr>
<tr>
<td>$R_8$ coded by AT</td>
<td>‘10’</td>
<td>3, 6, 3, 6</td>
</tr>
<tr>
<td>$R_8$ split into four $R_4$</td>
<td>‘11’</td>
<td></td>
</tr>
<tr>
<td>$R_4$ coded by $\mu$</td>
<td>‘0’</td>
<td>6</td>
</tr>
<tr>
<td>$R_4$ coded by AT</td>
<td>‘1’</td>
<td>3, 6, 3, 6</td>
</tr>
</tbody>
</table>

header. The header ‘11’ indicated that a parent range block was split into four child blocks. Then, ‘0’ and ‘1’ indicated that the child range block was coded using the mean and affine transformation, respectively. Therefore, the header had various lengths (one, two, and six bits) for different parent range blocks. There was some overhead in the two-level block sizes since the header in a single block size was only one bit. The bit rate could be calculated as

$$B_j = \frac{N_{8\mu} + 2N_{8\alpha} + 6N_{4s} + (N_8 + N_4)I_\mu + (N_{8\alpha} + N_{4\alpha})(I_\alpha + I_{PD})}{512^2} \text{ bit/pixel, (11)}$$

where $N_{8\mu}$, $N_{8\alpha}$, $N_{4s}$, and $N_{4\alpha}$ denote the number of parent range blocks coded using the mean, that coded using affine transformation, that partitioned into four child range blocks, and the number of child range blocks coded using affine transformation, respectively.

Fig. 7(a) shows the simulation results of the Lena image based on the proposed non-iterative and conventional fractal coding schemes. Using two-level block sizes, the bit rate and PSNR performance of the proposed methods were within moderate ranges. This figure shows that in the non-iterative scheme, more parent range blocks were split into child range blocks because they could not satisfy the MSE criterion we set. Therefore, higher bit rates were obtained. Since the image partition was dependent on its content and the threshold value we set, the bit rate of the coded image could vary a lot. In order to show that the proposed methods also work well for other images, the simulation results for three other images, Jetplane, Building, and Harbour (shown in Fig. 3(b)-(d)), are given in Figs. 7(b)-(d). Apparently, the performance revealed by these images is good when the domain pools are constructed from the mean image. Based on these simulation results, we conclude that the proposed methods achieve competitive or even better coding performances than the conventional iteration schemes.

Here, we also list the computation time needed on a Sparc_Ultra1 workstation when the proposed and conventional methods were used. The threshold value $\gamma_0$ for the convergence criterion in the conventional fractal coding scheme was set to be 0.005. Table 2 shows their CPU times (in seconds, where the decoding program was not optimized) for decoding the Lena image. The proposed non-iterative scheme saved about 89% of the decoding time required by the conventional fractal coding scheme. With the proposed domain pool design for the non-iterative fractal coding scheme, we can greatly speed up the decoding procedure, and the fidelity of the decoded image is well preserved.
Fig. 7. Coding results of the proposed non-iterative scheme using two-level sizes for the range block. (a) Lena, (b) Jetplane, (c) Building, (d) Harbour.

Table 2. Decoding time (in seconds) for decoding the Lena image.

<table>
<thead>
<tr>
<th>Fractal coding scheme</th>
<th>Domain pool size</th>
<th>Average time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>16</td>
<td>64</td>
</tr>
<tr>
<td>Conventional (subsampling)</td>
<td>13.2</td>
<td>16.7</td>
</tr>
<tr>
<td>Conventional (neighboring)</td>
<td>13.4</td>
<td>12.2</td>
</tr>
<tr>
<td>Non-iterative (subsampling)</td>
<td>1.7</td>
<td>1.7</td>
</tr>
<tr>
<td>Non-iterative (neighboring)</td>
<td>1.9</td>
<td>2.0</td>
</tr>
</tbody>
</table>

4.3 Comparison and Discussion

The conventional fractal coding scheme takes more than eight iterations to achieve convergence in decoding the Lena image [2]. Usually, the iteration number is image
dependent, and so is the decoding time. Although some methods have been proposed to speed up the iteration process for the conventional fractal coding schemes, at least three or four iterations are required [12], [13] to reach the convergence criterion in decoding the Lena image. The fractal coding scheme proposed in [11] is non-iterative in some special cases. The proposed non-iterative scheme accomplishes decoding by applying contractive affine transformation only once. According to the simulation results shown in Table 2, the non-iterative scheme greatly speeds up the decoding process in fractal image compression.

The proposed non-iterative scheme can be applied to other block-based fractal coding schemes, e.g., [17] and [18], which hold the mean information in the fractal codes. A higher compression ratio can be achieved by using lossless coding on the mean image. Furthermore, we can use a variable-size partition for the image. For example, by using quadtree partitioning [17], the source image can be partitioned into range blocks of maximum size 32×32 and minimum size 4×4, according to their complexity. We can only encode range blocks whose sizes are larger than 8×8 using the mean since the range block with a large size is hard to find a fine matching result. Therefore, the bit rate can be reduced although the image quality might be slightly degraded.

5. CONCLUSIONS

In this paper, we have proposed a fractal image codec that is non-iterative in the decoding stage. The decoder receives the on-line transmitted domain pool; thus, affine transformation can be applied only once. Due to the absence of iterations, the decoding time is greatly reduced. Simulation results show that we could obtain significant improvement in the decoding speed, and that the coding performance for the test images was comparable to or even better than that of the conventional fractal coding schemes. Since the decoder in the proposed fractal image codec is non-iterative, the required amount of memory is small and the computational complexity is much less than that in the conventional fractal coding schemes. Therefore, we will design a very-large-scale-integrated (VLSI) architecture for the proposed non-iterative decoder in our future research.

ACKNOWLEDGMENT

This research was partially supported by National Science Council, Taiwan, under contract NSC 88-2612-E-324-001.

REFERENCES

3/1-3/5.


**Hsuan T. Chang** (張軒庭) received his B.S. degree in Electronic Engineering (EE) from the National Taiwan Institute of Technology, Taiwan, in 1991, and the M.S. and Ph.D. degree in Electrical Engineering from National Chung Cheng University, Taiwan, in 1993 and 1997, respectively. He was an adjunct lecturer in the Department of Automation at National Yun-Lin Polytech Institute from 1993 to 1994 and a visiting researcher at the Laboratory for Excellence in Optical Data Processing, the Department of
Electrical and Computer Engineering, Carnegie Mellon University, from 1995 to 1996. Dr. Chang was an assistant professor in the EE Department of the Chein Kuo Institute of Technology (CKIT), Taiwan, from 1997 to 1999. He currently is an assistant professor in the Department of Information Management, the Chaoyang University of Technology (CYUT), Taiwan. Dr. Chang research interests are in image/video processing, optical information processing, computing, and grey theory. He was the director of the Signal Processing Laboratory of the Mechatronic and Optical Technology Research Center in CKIT. He has been an invited speaker and program committee of local conferences. Dr. Chang is member of SPIE, OSA, and The Chinese Image Processing and Pattern Recognition Society.

Chung J. Kuo (郭鎮榮) received the BS and MS degree in Power Mechanical Engineering from National Tsing Hua University, Taiwan, in 1982 and 1984, respectively, and the Ph.D. degree in Electrical Engineering (EE) from Michigan State University (MSU) in 1990. He joined the EE Department of National Chung Cheng University (NCCU) in 1990 as an associate professor and then became a full professor in 1996. He is now the chairman of the Graduate Institute of Communications Engineering NCCU. Dr. Kuo was a visiting scientist of the Opto-Electronics & System Lab, Industrial Technology Research Institute in 1991 and at the IBM T.J. Watson Research Center from 1997 to 1998, and has been a consultant to several international/local companies. He is also an adjunct professor at National Cheng Kung University.

Dr. Kuo research interests are in image/video signal processing, VLSI signal processing, and optical information processing/computing. He is a co-director of the Signal and Media (SAM) Labs. at NCCU. Dr. Kuo received the Outstanding Research Award from National Chung Cheng University in 1998, the Overseas Research Fellowship from the National Science Council (NSC) in 1997, the Outstanding Research Award from the College of Engineering, NCCU, in 1997, the Medal of Honor from NCCU in 1995, the Research Award from NSC every year since 1991, the Best Engineering Paper Award from Taiwan’s Computer Society in 1991, the Electrical Engineering Fellowship from MSU in 1989, and the Outstanding Academic Achievement Award from MSU in 1987. He was a guest editor for two special sections of Optical Engineering and 3D Holographic Imaging (to be published by John, Wiley and Sons) and has been an invited speaker and program committee chairman and member at several international/local conferences. He also serves as an Associate Editor of IEEE Signal Processing Magazine and President of SPIE Taiwan Chapter. Dr. Kuo is a senior member of IEEE, is a member of Phi Kappa Phi, Phi Beta Delta, IEEE, OSA, and SPIE, and is listed in Who's Who in the World.