Chinese Journal of Electronics
Vol.12, No.2, Apr. 2003

A Parallel Implementation of Improved Fractal Image Coding Based on Tree Topology*

SUN Yunda, ZHAO Yao and YUAN Baozong

(Institute of Information Science, Northern Jiaotong University, Beijing 100044, China)

Abstract — One of the main drawbacks of fractal image coding (FIC) is its time-consuming encoding process. So how to speed up the encoding process is a challenging issue of FIC research. As both sequential solutions and parallel ones have their advantages and disadvantages, we combine them together to further speed up the encoding phase. In this paper a derivative tree topology is first proposed to provide support for complex parallelism. Then a dual-classification technique is designed for speeding up the fractal image coding with Same-Sized Block Mapping, which improves the decoded image quality. Finally, some experimental results with good performance are presented.

Key words — Fractal image coding (FIC), Parallel image processing, Tree topology.

I. Introduction

FIC, which is based on the theory of contractive transformations, can find its origin from the papers of M. F. Barnsley and A. E. Jacquin[1,2]. It has generated more interests in image compression community since Y. Fisher's outstanding work[3]. Compared with other mature or newly developed compression techniques, FIC has its particular advantages, such as high compression ratio, resolution independence, etc. However, it is primarily placed in the class of archival coding algorithms due to its computationally expensive encoding algorithm. After original image to be compressed is divided by different ways, resulting in two sets of pixel blocks, range blocks and domain blocks, the exhaustive search must be made on the domain pool to find out the best matched domain and affine transform for each range block. So if the image has $n \times n$ pixels, the computational complexity is about $O(n^4)$, while its decoding procedure is much simpler, on the order of $O(n^2)$[4]. Therefore, it becomes a big problem to speed up the encoding step.

All the speeding solutions can be classified into two types: sequential ones and parallel ones. The sequential ones, aiming at decreasing the encoding complexity, is a flexible and economical way. Now several techniques, such as classification and feature vector methods, have been proposed and extensively studied[3,5,6]. Unfortunately, in all these methods, the performance degradation is inevitable due to the decrease of potential domains. In addition, they are still not practical for real-time applications on most sequential machines.

The parallel ones, which distribute the encoding task among a certain number of processor elements (PEs), can achieve better performance. So far, there has been much work on implementing FIC with diverse parallel systems: SIMD (single instruction, multiple data) arrays, MIMD (multiple instruction, multiple data) architectures and ASIC (application specific integrated circuit) machines. Corresponding algorithms have been proposed[7-11]. Generally speaking, parallel processing is an effective but costly way; moreover, the encoding complexity remains unchanged in these solutions.

In order to further speed up the encoding phase, we combine the sequential approach together with the parallel one in our study. While improving conventional schemes to reduce the computation complexity, we seek for the scalable parallel structures, application specific or not. Then these improved methods can be applied into parallel systems. So far, we have introduced a dual-classification technique into FIC with Same-Sized Block Mapping, a newly proposed improvement on conventional fractal methods, and devised the parallel algorithm correspondingly on MIMD system with a derivative tree topology. Our solution makes fractal coding more suitable for real-time application and for the compression of large images where single speedup technique does not offer enough efficiency.

The rest of this paper is organized as follows: Section II proposes an efficient topology for MIMD parallel systems. Its computational complexity is also analyzed. Section III proposes some improvements on conventional fractal methods. The experimental results and conclusion are given respectively in Section IV and Section V.

II. Proposed Tree Topology for MIMD Systems

We choose transputer, a typical multi-processor building block produced by the INMOS division of SGS-Thomson, as the basic processing element in our studies. A transputer network is a MIMD machine because it is built with separately programmable general-purpose processors connected through fast interconnection links.

*Manuscript Received Jan. 2002; Accepted Nov. 2002. This work is supported jountly by Ministry of Education of China, Huo Yingdong Young Teachers Foundation and the National Natural Science Foundation of China (No.69802001 and 60172062).

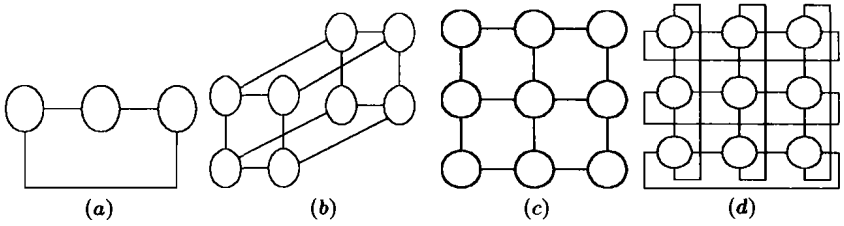

Fig. 1. (a) Ring topology, (b) Hypercube topology, (c) Mesh topology, (d) Torus topology

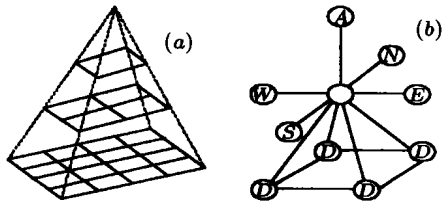

Fig. 2. (a) The overview of pyramid structure, (b) The connection relationship in pyramid

PEs can be configured in a variety of topologies so long as the incoming and outgoing connections to each processor do not exceed the number of links. Some popular topologies of transputer are ring, mesh, torus, and hypercube, as shown in Fig.1.

The topology used in our study is derived from standard pyramid structure[13], a particular parallel computing topology, which can find its origin from multi-resolution image processing and wide appliance in computer vision. Fig.2(a) shows the overview of pyramid structure. It is a multiple-layer model, and each small square represents a PE in each layer. The connection relationship among PEs is given in Fig.2(b), where each PE has four descendants D in the lower layer, one ancestor A in the upper layer, four brothers W, N, E, S in the same layer. However, such pyramid structure is unsuitable for FIC. The reason lies in two facts: a waste of communication resources and an increase in communication burden. Firstly, there is no necessity for interconnection among peer PEs, only if the method is subtly designed. Secondly, as each ancestor is limited to connect four descendants, more PEs must be arranged in too many layers, which inevitably increases communication burden.

To meet the need of FIC, we disconnect PEs within the same layers and use all links to communicate with one ancestor and many descendants. In this way, each PE's communication capacity is fully exploited in a very compact fashion. We call it as tree topology, which is just like a tree growing downwards, composed of root, branches and treetop. As shown in Fig.(3), this topology takes on multiple-layer architecture outside and master/slave structure inside. From the function point of view, within one layer, each processor can receive task from its master processor (ancestor in the upper layer) and subsequently allocate partial task to its slave processors (descendants in the lower layer). Based on the above design, a d-layer tree can support up to $n = \sum_{i=0}^{d-1}(a-1)^i$ processors, where a is the number of links for selected processor.

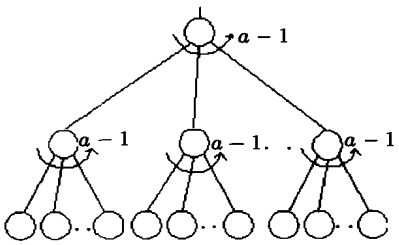

Fig. 3. Tree topology

To evaluate these topologies, let's take the time cost into consideration. For a parallel structure with n processors, the computational complexity mainly consists of two parts: task allocation and execution. On an ideal condition, when the encoding work is evenly partitioned into n parts, time cost can be counted as the ratio of total data volume to average network bandwidth, plus $1/n$ of the execution time under sequential method. Besides, the former can reach a maximum if all assignments are transmitted through the whole diameter, which is known as the maximum distance in terms of number of links between the entrance processor (processor connected to the host machine) and any other processors in the network. That is:

$$\begin{aligned}\tau &\approx \frac{V_d}{\overline{B}} + \frac{V_c}{n \cdot C} \leq \frac{V_d}{B_{\min}} + \frac{V_c}{n \cdot C} \\ &= n \cdot \frac{V_d/n}{R/d} + \frac{V_c}{n \cdot C} = d \cdot \frac{V_d}{R} + \frac{V_c}{n \cdot C}\end{aligned} \tag{1}$$

where V_d is total data volume (bits), including input and output, $\overline{B}$ and $B_{\min}$ is average and minimum bandwidth of the network (bps), V_c is computation volume of the sequential method, C is computation capacity of a single processor (flops), R is I/O throughput for a processor (bps) and d is diameter of a topology. From Eq.(1) we know that encoding time τ is approximately proportional to diameter d on the condition of fixed n processors, so such topology with the smallest diameter tends to achieve the best performance. Referring to Ref.[12], Fig.4 gives the performance comparison of the five topologies, among which tree topology is the most scalable and effective due to its smallest diameter. Thus it can provide support for complex parallelism easily by alterable system configuration and controllable assignment, and then it is more suitable for such computationally intensive work as FIC.

III. Improved FIC with Same-Sized Block Mapping

In conventional FIC schemes, domain blocks are always constrained to be larger (usually twice) than range blocks to ensure the convergence of the iterative decoding procedure, which means only the self-similarity of different scales is exploited. However, in real natural images, there exists not only similarity of different scale but also similarity at the same scale. For example, the left eye of a person is very similar to the right; they are of the same size instead of different size. In order to overcome this drawback, T. Bedford[14] and Zhao[15] put forward the idea of same-sized block mapping (SSBM), in which domain blocks same-sized as range blocks besides twice larger

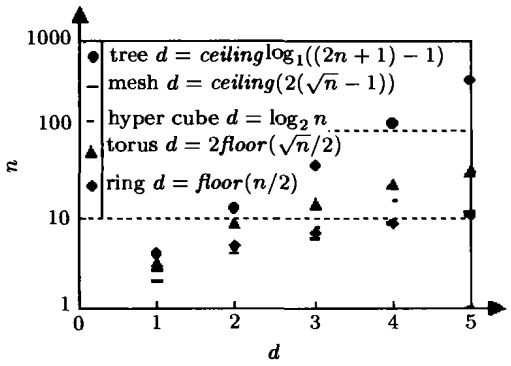

Fig. 4. The number of transputers vs. diameter of five topologies

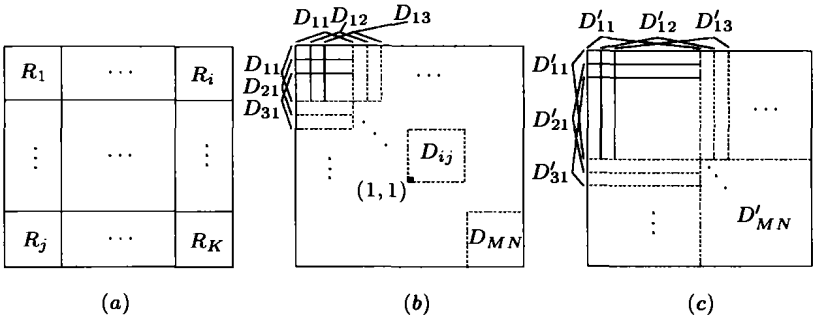

Fig. 5. (a) Range blocks R, (b) Same-sized domain pool D, (c) Double-sized domain pool D'

ones are used. This scheme increases the number of all potential domains and the similarity degree between a range and its matched domain, leading to better reconstructed quality.

As it is shown in Fig.5(a), an original image is first partitioned evenly into range blocks $R_1 \cdots R_K$. Then the whole domain pool is constructed, including newly-utilized same-sized domain pool D as well as conventional double-sized domain pool D', see Figs.5(b) and 5(c). With the nearly doubled domain pool, SSBM acquires better fidelity but requires more time on exhaustive full search, which is about 10 times slower than Fisher's scheme when encoding images of medium size. This greatly prohibits its further application. Therefore, in order to minimize the number of domains compared with a range, we use a dual-classification technique to improve FIC with SSBM. For any $N \times N$ block in D (take D_{ij} for example), its mean intensity AVE_{ij} and class number x_{ij} are defined as:

$$AVE_{ij} = \frac{1}{N^2}\sum_{m=1}^{N}\sum_{n=1}^{N} B(m,n) \tag{2}$$

$$x_{ij} = floor(AVE_{ij}/INTERVAL) \tag{3}$$

Then it is classified into one of $X = NUM/INTERAVL$ classes, where $B(m,n)$ denotes each pixel's gray-level in D_{ij}, NUM and $INTERVAL$ are pre-selected parameters, meaning total gray-levels (usually 256) and gray-level interval (8 proposed) respectively; this avoids reclassification of same-sized domains later. On the other hand, all elements of D' are classified as same as the Fisher's[3] or Hurtgen's scheme[5]. In the former, because a square block can be oriented (rotated and flipped) in such a way that the average intensities of its four quadrants are ordered in one of three ways, three major classes are constituted. In addition, there exist 24 possible orderings of the quadrants' variances which define 24 subclasses for each major class, yielding 72 classes in all. In Hurtgen's scheme, an image block and its four quadrants' mean intensities are first computed; then each quadrant is assigned a bit which is 1 if its mean is above the overall mean and 0 otherwise. In this way, each block is associated with one of 15 possible classes (the 16^{th} class is always empty). To further characterize the blocks, the variances can be used like Fisher's scheme, obtaining 360 classes in all. Accordingly, our scheme is named as FX (HX) scheme, where X denotes number of classes in D; F(isher), H(urtgen) refer to the classification technique used for D'.

During the encoding, the optimal matched domain block called the domain match must be found and recorded for each range block. Let's take R_i for example. On the one hand, if R_i has been used less than MAX times in the same-sized way (i.e. when encoding previous range blocks $R_1 \cdots R_{i-1}$, any part of R_i is contained in same-sized domain matches for less than MAX times, where MAX is a pre-selected maximum number used to ensure convergence and fidelity), both D and D' are available. R_i is first classified into one of X classes, and only elements with identical (or near) classification in D are compared with R_i. After the same-sized domain match is found, whether its root mean square (rms) can satisfy a pre-selected threshold RMS or not determines its usage. Either it is recorded by quantizing and storing scale factor and domain position, or R_i is reclassified just as elements in D'. To find the double-sized domain match quickly, only elements with identical (or near) classification in D' are compared with R_i. If this match is still insufficient on a pre-selected threshold TOL, R_i has to be partitioned into four quadrants and encoded recursively. Otherwise, the double-sized domain match is recorded by quantizing and storing symmetry operation, scale factor, brightness offset and domain position. On the other hand, if R_i has been used MAX times in the same-sized way, D is unavailable and the domain match can only be retrieved from D'. If this double-sized domain match is insufficient on TOL, R_i has to be partitioned into four quadrants and encoded recursively. Otherwise, this match is recorded. The dual-classification technique significantly reduces the number of domain-range comparisons. Furthermore, without consideration of symmetry operation and brightness offset in SSBM, it is almost impossible to match a range block with a same-sized domain block of different class when coding on a low threshold, so we can speedup the encoding step after dual-classification with almost no loss in fidelity.

IV. Experimental Results

We demonstrate the FX (HX) scheme on a tree topology based MIMD system, using four IMS T800 Transputers at a processor speed of 20MHz. As shown in Fig.6, it is an ex-

perimental two-layer model, which consists of one root, or master processor (M) connected to a host machine (H) and three slave processors (S). At the beginning, all data including instructions are loaded from the host machine to the master processor, and then allocated to slave ones. During the coding procedure, the above FX (HX) scheme is implemented in all processors in parallel, while slave processors can submit requests to the master for interaction. Once a processor has finished its allocation, it sends a signal to the master, which is responsible for collecting results and feeding them back to the host machine. In the end, all coding data is reorganized and written to a disk file.

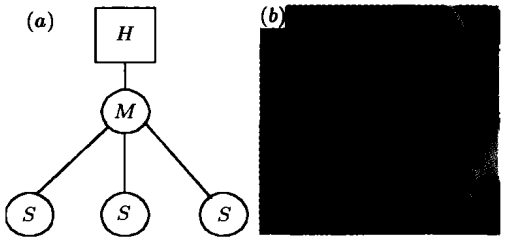

Fig. 6. (*a*) An experimental two-layer model of tree topology based MIMD system, (*b*)The test original image: standard "Lena"

The first experiment aims at choosing the best *INTERVAL* (gray-level interval) for our FX and HX scheme, which determines the number of classes in D (i.e. X). Fig.7 shows the curves of 256 × 256 "Lena" encoded with TOL = 8, MAX = 5 and RMS = 10. As the increasing of *INTERVAL*, bit rate decreases exponentially with almost unchanged PSNR, the execution time increases due to the expanding class. We know that *INTERVAL* = 8 can achieve the best performance. Therefore, F32 and H32 schemes will be used in following experiments.

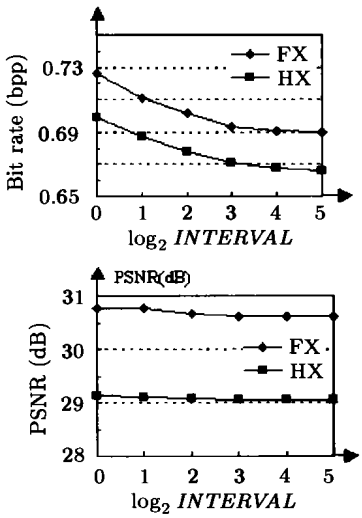

Fig. 7. The bit rate and PSNR vs. $\log_2$ *INTERVAL*

The second experiment is done to compare the performance among Fisher's quadtree scheme, Hurtgen's scheme, Zhao's scheme, and F32(H32) scheme. The rate-distortion curves of 256 × 256 "Lena" are shown below, among which F32 and Zhao's scheme basically overlap each other, meaning little loss in fidelity after dual-classification. From Figs.8 and 9, we know that performance of Fisher's classification technique surpasses that of Hurtgen's on the whole; however, the latter behaves better in keeping reconstructed image quality with the increasing of compression ratio. At the same bit rate, PSNR of our F32 and H32 scheme can reach about 0.8dB higher than Fisher's scheme and Hurtgen's scheme respectively.

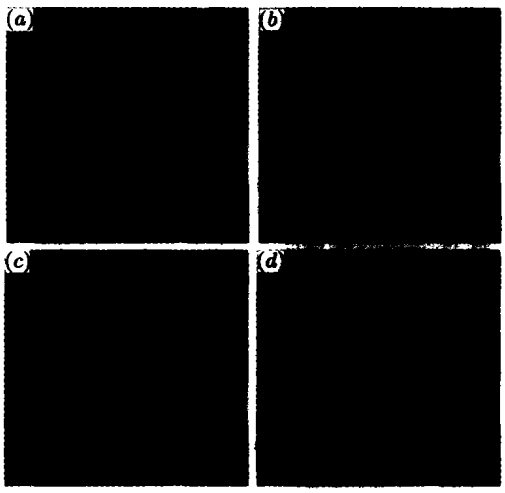

Fig. 8. Reconstructed images of the four schemes at 0.5bpp. (*a*) F32 (Zhao), PSNR=29.65dB; (*b*) Fisher, PSNR=28.86dB; (*c*) H32, PSNR=28.48dB; (*d*) Hurtgen, PSNR=27.87dB

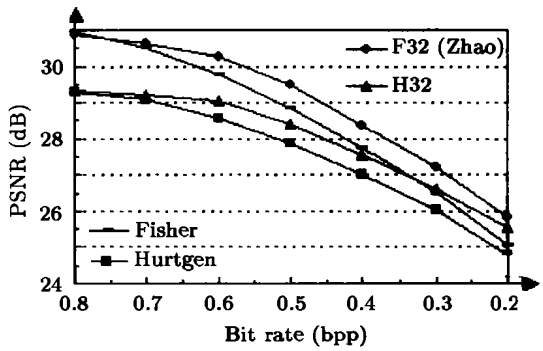

Fig. 9. PSNR vs. bit rates of the four schemes

Table 1. Execution times for different schemes

Scheme \ Image size / Timing	64 × 64	128 × 128	256 × 256
Zhao's scheme	2 sec	16 sec	110 sec
F32 scheme	0.5 sec	3 sec	13 sec
H32 scheme	0.3 sec	2 sec	10 sec

Another experimental result is about the absolute times for encoding "Lena" with different resolutions. As shown in Table 1, our F32(H32) scheme greatly reduce the execution time compared with Zhao's scheme. The larger original image is, the higher degree of speedup is. Due to application of parallelism and improvement on FIC with SSBM, we achieve a rather good performance in an economical way. Moreover,

our tree topology based MIMD architecture can be extended easily by adding chips into the system to meet the need of given occasion.

V. Conclusions

1. In this paper, sequential approach and parallel processing are combined together to further speed up the encoding phase, thus their advantages are exploited simultaneously in FIC.

2. A derivative tree topology is used to construct effective processor networks. Due to the small diameter, it is more suitable for our computationally intensive work than other popular topologies.

3. Via dual-classification technique, considerable speed-up is made with almost no loss in fidelity. So our FX(HX) scheme greatly improves FIC with SSBM.

4. With the highly scalable parallel structure and improved method, our solution may easily meet the need of given application through network extension. It also provides us with an economical choice.

References

[1] M.F. Barnsley, *Fractals Everywhere,* Academic Press, New York, 1988.

[2] A.E. Jacquin, "A novel fractal block-coding technique for digital images", in *Proc., ICASSP,* Vol.4, pp.2225–2228, 1990.

[3] Y. Fisher, editor, *Fractal Image Compression: Theory and Application,* Springer-Verlag, New York, 1995.

[4] M. Xue, T. Hanson and A. Merigot, "A massively parallel implementation of fractal image compression", In A. Bovik, editor, *Proceedings of the First IEEE International Conference on Image Processing,* pp.II/640-II/644, IEEE Press, 1994.

[5] B. Hurtgen and C. Stiller, "Fast hierarchical codebook search for fractal coding of still images", in *Proc. EOS/SPIE Visual Communications PACS Medical Application'93,* Berlin, Germany, 1993.

[6] Mario Polvere and Michele Nappi, "Speed-up in fractal image coding: comparison of methods", *IEEE Transactions on Image Processing,* pp.1002–1009, Vol.9, No.6, Jun. 2000.

[7] Christian Hufnagl and Andreas Uhl, "Algorithms for fractal image compression on massively parallel SIMD arrays",*Real Time Imaging,* Vol.6, pp.267–281, 2000.

[8] J. Hammerle and A. Uhl, "Parallel algorithms for fractal image coding on MIMD architectures", In *Proceedings of The First International Conference on Visual Information Systems,* pp.182–191, Melbourne, Feb. 1996.

[9] A. Uhl and J. Hammerle, "Issues in implementing block-based image compression techniques on parallel MIMD architectures" In J. Biemond and E. J. Delp, editors, *Visual Communication and Image Processing'97 (VCIP'97),* Volume Proc. SPIE 3024, pp.494–501, San Jose, Feb. 1997.

[10] Kevin P. Acken, Mary Jane Irwin and Robert M. Owens, "A parallel ASIC architecture for efficient fractal image coding", *Journal of VLSI Signal Processing,* Vol.19, pp.97–113, 1998.

[11] D. Vidya, Ranjani Parthasarathy, T.C. Bina, N.G. Swaroopa, "Architecture for fractal image compression", *Journal of System Architecture,* Vol.46, pp.1275–1291, 2000.

[12] P. Saratchandran, N. Sundararajan and Shou King Foo, *Parallel Implementations of Back Propagation Neural Networks on Transputers — A study of Training Set Parallelism,* World Scientific, Singapore, 1996.

[13] J.D. Becher and I. Eisele, "Pyramidal architecutures for image processing", *Proc. Workshop on Parallel Processing: Logic Organization and Technology,* (Eds.), Berlin, West Germany, Springer-Verlag, pp.58–74, 1986.

[14] T. Bedford, F.M. Dekking, M. Breeuwer, M.S. Keane, D. van Schooneveld, "Fractal coding of monochrome images", *Signal Processing: Image Communication,* Vol.6, pp.405–419, 1994.

[15] Yao Zhao, "Exploiting same scale similarity in Fisher's scheme", *Chinese Journal of Electronics,* vol.10, No.2, pp.153–155, Apr. 2001.

SUN Yunda received his B.S. degree from the School of Mechanics and Electrics, Northern Jiaotong University in 2000. Since then he has been working towards the Ph.D. degree in signal and information processing in Computer Vision Lab, Institute of Information Science, Northern Jiaotong University, Beijing. His current research interests include image and video coding, 3D reconstruction and viewpoint synthesis.(Email:iirainbow@yeah.net)

ZHAO Yao was born in Jiangsu Province, China in 1967. He received B.E. degree from Fuzhou University in 1989 and M.E. degree from Southeast University in 1992, both in Radio Engineering Department. He received the Ph. D. degree in the Institute of Information Science, Northern Jiaotong University (NJTU) in 1996. He became an associate professor with NJTU in 1998 and became a professor in 2001. From 2001 to 2002, he worked as a Senior Research Fellow in Information and Communication Theory Group, Faculty of Information Technology and Systems, Delft University of Technology, Netherlands. He is an editor of a national journal SIGNAL ACQUISITION AND PROCESSING. His research interest includes image coding, fractals, digital watermarking and content-based image retrieval. He has published more than 40 papers in international journals and conferences. Now he is leading several national research projects from National Natural Science Foundation of China and Huo Yingdong Young Education Foundation.

YUAN Baozong was born in Jiangsu Province, China, in 1932. He received the Ph.D. degree in electrical engineering from Leningrad Institute of Railway Engineering, USSR, in 1960. He has joined the Northern Jiaotong University since 1953. He was a visiting professor at the University of Pittsburgh, USA, and the University of Wales, UK in 1982, 1983, and 1988 respectively. Dr. Yuan is Chairman of Computer Chapter of IEEE Beijing Section, Fellow of British Royal Society, IEE Fellow, Vice Chairman of IEE Beijing Center Development. His research interests include computer vision, virtual reality, image processing, computer graphics, speech signal processing, and multimedia information processing and data communication.