Hybrid MPI-OpenMP Parallelization Of Image Reconstruction

Jinliang Wan
College of Computer and Information Engineering, Henan University of Economics and Law
Zhengzhou, China
Email: wanqq_2006@hotmail.com

Yanhui Liu
State Key Laboratory for Seismic Reduction/Control & Structural Safety (Cultivation), Guangzhou University
Guangzhou, China
Email: Liuyanhui19@163.com

***Abstract*—Perceptual coding should take full advantage of the results from human visual system (HVS) studies. In this paper, we exploit an image reconstruction algorithm, which can simply and reliably represent images using regional shape and texture information and effectually eliminate perceptual redundancy. But real-time application of the algorithm is limited due to its inherent complex and intensive computation. Therefore, we implement image reconstruction with a hybrid message passing interface (MPI) and OpenMP approach on the basis of a standard MPI implementation. Experimental results confirm our image reconstruction algorithm can get optimal restoration results. Compared with JPEG and JPEG2000 compression, this algorithm can not only get more coding gain but remain most information people care. Our hybrid MPI-OpenMP approach can significantly improve time performance of image reconstruction algorithm.**

***Index Terms*—human visual system, image reconstruction, parallel processing**

I. INTRODUCTION

The mainstream of image coding scheme is based on hybrid coding framework integrated with three classes of traditional technologies. Transform coding, prediction coding and entropy coding are used to reduce and remove spatial redundancy, temporal redundancy and statistic redundancy. But human eyes are the terminal receptors for images, and thus there still is a lot of perceptual redundant information according to the characteristics of human visual system (HVS). However, it is difficult to achieve additional significant progress for compression efficiency because of the limitation of pixel-based image representation [6]. Therefore, how to find a more efficient image representation method has always been coding scientist's ardent pursuit.

The second-generation image coding techniques were proposed by Kunt [1], which can take advantage of the human visual perceptual characteristics. In recent years, with the establishment and development of the wavelet transform theory [2], fractal theory [3, 4] and visual simulation theory [5], Image processing has been considerable development. And, actually many still images contain massive textures like grass, brick, cloud, sand, etc., which compose the still backgrounds. Ndjiki-Nya [7] assumed that the textures in an image could be classified into two categories: textures with unimportant subjective details and the remainder. Sun [8] utilized this idea and considered textures as unimportant parts in terms of the inherent imperfections of the HVS. Change of texture details will not affect the subjective comprehension toward the original textures. It indicates that these visual features can be utilized to eliminate perceptual redundancy.

In recent years, there have emerged large quantities of efficient and effective algorithms of image texture analysis and synthesis [9-13]. A non-parametric method for texture synthesis was proposed in [11]. The method can produce good synthesized results for a wide range of textures by stitching together small texture patches. In addition, the detection and extraction of linear structures which usually refers to the contours used to define the target shape or the boundaries used to demarcate the regions also have been researched far and wide [17-20]. Most of the early shape description techniques used binary images. At present, these researches have a wide range of applications in the fields of image processing and computer vision.

In this study, we propose an image reconstruction algorithm and its hybrid MPI-OpenMP approach. This algorithm can select the smallest region texture sample and obtain optimal restoration images. But with the image size increasing, image reconstruction will cost a large executing time. To improve its time efficiency, we discuss the hybrid MPI-OpenMP approach in this paper. The parallel programming approach is implemented on a dedicated cluster. The remainder of this paper is organized as follows. In the next section, the details of the proposed image reconstruction algorithm are presented. And Sect.3 gives the details of the hybrid MPI-OpenMP approach. The proposed image reconstruction

doi:10.4304/jsw.8.3.687-693

algorithm and hybrid MPI-OpenMP approach are tested in Sect.4. Finally, this paper is concluded in Sect.5.

II. Image Reconstruction Algorithm

Diagram of image reconstruction algorithm is shown in Fig.1. The implementing procedure of the image texture substitution algorithm consists of three phases: multi-regions extraction, image feature representation and multi-regions image reconstruction.

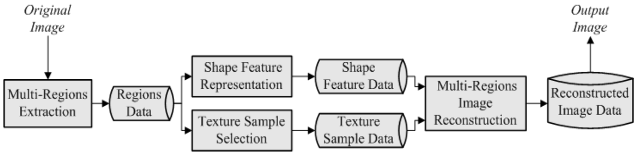

Figure 1. Diagram of the image reconstruction algorithm

A. Multi-Regions Extraction

An image is first segmented into many small regions in terms of homogeneous color and texture. The segmentation method is the so-called JSEG algorithm [14, 15]. Considering that texture sample selection is sensitive to segmentation results, we merge connected small regions if they are nearby in space so that each image can obtain bigger regions [16]. And, number of regions which each image gets may be variable in order to obtain optimal restoration result.

Through multi-regions extraction, homogeneous pixels belong to the same region. Each region has a unique index and every pixel belonged to one region is assigned the same region index. Region indexes of all pixels compose of region index list. The region index list is as the same size as the original image if the input one is grayscale, and it is one third size as the original color image naturally. Each element of index list corresponds to the pixel of original image. Specifically, the region index list plays an important role in the image feature representation and the hybrid MPI-OpenMP parallel programming approach.

B. Image Feature Representation

Shape and texture are salient features of the appearance of objects in natural scenes. The mid-level region is image representing and reconstructing unit which contains shape and texture characteristics. Describing approaches of shape and texture features will be discussed respectively.

The implementing procedure of regional shape feature representation is described as the following steps: binarization processing, contour tracking, down sampling and piecewise iterative curve fitting. Fig.2 indicates the schematic drawing of four stages above. In order to elaborate conveniently, we just take into account one region.

Firstly, the image should be binarized according to region index list. This coding region corresponds to the white area and other regions correspond to the black area, as is depicted in Fig.2 (a).

Secondly, an ordered sequence of contour points approximating the regional shape is extracted. For this purpose, a contour tracing algorithm [17] is applied to the input binary image. The shape of this coding region is drawn by the white closed curve, as is shown in Fig.2 (b).

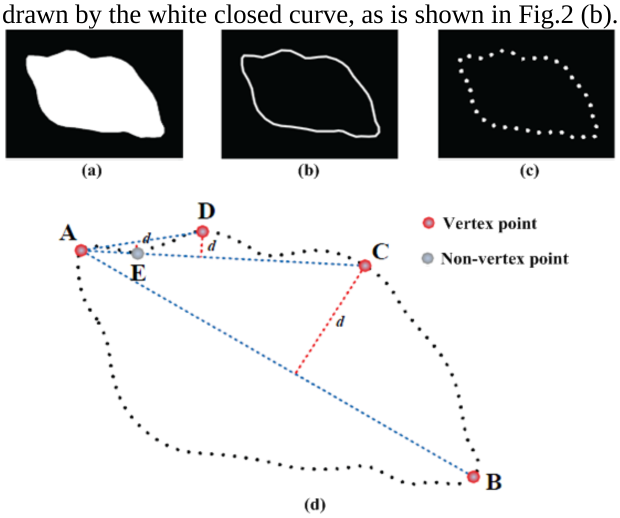

Figure 2. Schematic drawing of regional shape feature representation

Thirdly, a vector composed of equally distributed points along the regional contour is extracted from the ordered sequence of contour pixels. Contour points are extracted in such a way that they will be approximately equally distributed along the regional contour curve, showed as Fig.2 (c). The regional contour is represented by the ordered sequence (x_s, y_s) of contour points, where s denotes the position along the regional contour curve.

Finally, the piecewise iterative curve fitting [8] combined with top-down split algorithm and curve-fitting principle is implemented to represent region structure feature. Its implementation procedure is described as the following steps:

- Connecting the two farthest edge points A and B into a line segment L_{AB} , and thus split the closed contour into two curves AB and BA, as is shown in Fig.2 (d).
- Calculate the distance d of each edge point on curve AB to this line segment L_{AB} using (2), and determine the edge point farthest from the line segment L_{AB}.

Here, suppose the coordinates of A and B are (x_A, y_A) and (x_B, y_B) , so the formula for line segment L_{AB} passing from the two end points is shown as (1):

$$x(y_A - y_B) + y(x_A - x_B) + x_A y_B - x_B y_A = 0 \quad (1)$$

The distance d of edge point (x_s, y_s) to the line segment L_{AB} is:

$$d_s = {r_s}/{\Delta_{AB}} \quad (2)$$

Where, s denotes the position of any point along the curve AB, and

$$r_s = x_s(y_A - y_B) - y_s(x_A - x_B) + x_A y_B - x_B y_A \quad (3)$$

$$\Delta_{AB} = \| A - B \| \quad (4)$$

Thus, the maximum absolute error is:

$$MAE = \max_{s \in [A,B]} | d_s | \quad (5)$$

Clearly, point C with MAE is the farthest edge point from the line segment L_{AB}.

- .If the maximum error of that point from the line segment L_{AB} is above a threshold, split the curve AB into two curve segments at that point (i.e. new vertex point C), one between A and C, the other between C and B.
- Repeat second step and third step separately for the above two sub-curve segments AC and CB to determine more and more vertices until the error is smaller than the "split threshold" set in advance.
- Implement polynomial curve fitting on each sub-curve segment and the polynomial coefficients are deemed as region shape features.

Simply, we only take sub-curve segment AC as an example. The curve AC is represented by the order sequence $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$ of edge points, where n denotes the number of elements in this order sequence. A polynomial always may be expressed by the sum of some orthogonal polynomial. Suppose the fitted polynomial can be expressed by (6):

$$\begin{aligned} F_m(x) &= \sum_{i=0}^{m} C_i P_i(x) \\ &= C_0 P_0(x) + C_1 P_1(x) + \cdots + C_m P_m(x) \\ &= P_0 + P_1 x + P_2 x^2 + \cdots + P_m x^m \end{aligned} \quad (6)$$

Our purpose is to calculate $P_0, P_1, \cdots, P_m$, where m denotes order of the fitted polynomial. The orthogonal function family can be constructed by (7):

$$\begin{aligned} & P_0(x) = 1 \\ & P_1(x) = (x - A_1) P_0(x) \\ & P_2(x) = (x - A_2) P_1(x) - B_1 P_0(x) \\ & \cdots\cdots \\ & P_{j+1}(x) = (x - A_{j+1}) P_j(x) - B_j P_{j-1}(x) \end{aligned} \quad (7)$$

Where,

$$A_{j+1} = \frac{\sum_{i=1}^{n} x_i P_j^2(x_i)}{\sum_{i=1}^{n} P_j^2(x_i)}, B_j = \frac{\sum_{i=1}^{n} x_i P_j^2(x_i)}{\sum_{i=1}^{n} P_{j-1}^2(x_i)}, C_j = \frac{\sum_{i=1}^{n} y_i P_j^2(x_i)}{\sum_{i=1}^{n} P_j^2(x_i)} \quad (8)$$

And $j = 0, 1, \cdots, m$.

Finally, we may calculate the polynomial coefficients $P_0, P_1, \cdots, P_m$ in terms of C_j and $\{P_j(x)\}$ above. In this way, the contour curve of one region is figured by these coefficients and necessary endpoints.

Regional texture feature is represented by regional texture sample selected. A regional texture sample should contain a majority of local and global characteristics of the inner texture. The inner texture is analyzed by a 2D-autocorrelation statistic analysis method [21]. The autocorrelation function of an image can be used to assess the amount of regularity as well as the fineness/coarseness of the texture present in the image. The autocorrelation function of an image is defined as follows:

$$\rho_{\mathrm{II}}(x, y) = \frac{MN}{(M - x)(N - y)} \bullet \frac{\sum_{u=1}^{M-x} \sum_{v=1}^{N-y} I(u, v) I(u + x, v + y)}{\sum_{u=1}^{M} \sum_{v=1}^{N} I^2(u, v)} \quad (9)$$

For regular structural textures, the function will exhibit peaks and valleys. We can easily determine the scale of the texture primitives from the given texture. The texture sample should contain 2~5 texture primitives. For stochastic textures, 90% above values of the function are bigger than 0.9 generally. Correspondingly, the size of texture sample not only may be set smaller but also can be controlled by the researchers.

C. Multi-Regions Image Reconstruction

The implementing procedure of multi-regions image reconstruction is described as the following stages: region contour reconstruction, region texture synthesis and region restoration.

In the first stage, the contour curve of one region can be described as a polynomial function as follows:

$$y = P_0 + P_1 x + P_2 x^2 + \cdots + P_m x^m \quad (10)$$

Where, the polynomial coefficients $P_0, P_1, \cdots, P_m$ have been attained through region structure feature representation. According to the x-coordinates of two endpoints, such as x_m and x_n, it is easy to attain an abscissa vector $\vec{x} = \{x_m, x_{m+1}, \cdots, x_n\}$, where each element is the x-coordinate of one point on the curve. Substitute elements of $\vec{x}$ in (10) one by one, then the y-coordinate of each point will be gained. In this way, the region contour is reconstructed. In order to make the connection between two neighboring curves perfect and guarantee the contour closed, a generic algorithm of dilation is applied.

In the second stage, a texture synthesis algorithm is performed in the manner of Efros and Freeman's image quilting in [11]. And many implementation details are introduced in [22]. For instance, a bigger patch means better capturing of texture characteristics in the texture patches and thus more similarity between the original texture and the synthesized one. It is usually assumed that the patch contains one texture element at least.

Finally, region restoration is achieved through such a way that synthesized region texture is filled to the reconstructed region contour. Multi-regions image reconstruction is achieved when all regions are successfully restored.

III. HYBRID MPI-OPENMP APPROACH

To improve the efficiency of image reconstruction algorithm, during the implementing procedure, parallel techniques are utilized to achieve image feature extraction and multi-regions image reconstruction. The implementing procedure of this parallel programming approach consists of two phases: MPI parallelism and OpenMP parallelism. In order to illustrate the implementing process of the proposed hybrid MPI-OpenMP approach, we give its parallel implementing flowchart, as shown in Fig.3.

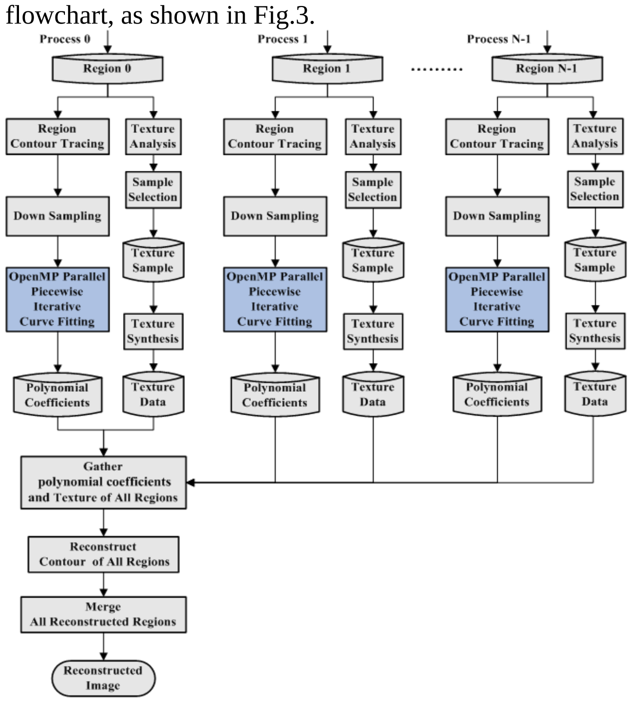

Figure 3. The hybrid MPI-OpenMP parallel implementing flowchart

A. MPI Parallelism

The MPI parallelism is designed based on the Master-Slave design pattern [23] and its methodology is based on the Task-Channel model. In order to elaborate conveniently, we assume that number of regions is N. And then indexes of these regions are $0,1,2,\cdots,N-1$, respectively. Each region is mapped as a task and each task corresponds to a process. ID number of master is 0; ID numbers of all slavers are $1,2,3,\cdots,N-1$, respectively. The processor whose ID number is i takes charge of region which has same index. For example, the master takes charge of region whose index is 0.

In the MPI parallel implementing process, Firstly, according to region index list, The master and all slaves obtains regional structure feature through performing contour tracking, down sampling and piecewise iterative curve fitting and regional texture sample through performing sample selection. After that, the master and all slaves accomplish regional texture synthesis according to regional texture sample.

Secondly, because contour curve of each region is different and polynomial coefficients are generated randomly, number of polynomial coefficients of region contour curve is different. Therefore, the master and all slaves gather numbers of these polynomial coefficients of contour curves of all regions from all processors, which could be implemented by using the MPI_Allgather() function. After that, the master gathers these polynomial coefficients of contour curves of other regions from all slavers, which could be implemented by using the MPI_Gatherv() function.

Then, the master gathers synthesized texture of other regions from all slavers, which could be implemented by using the MPI_Gather() function.

Finally, the master achieves contour reconstruction of all regions. Then, the master accomplishes multi-regions image reconstruction through completing restoration of all regions successively.

B. OpenMP Parallelism

```
..................
#pragma omp parallel sections
{
    #pragma omp section
    {
        /* compute polynomial coefficients of sub-curve segment */
    }
    #pragma omp section
    {
        /* compute polynomial coefficients of sub-curve segment */
    }
    ⋮
    #pragma omp section
    {
        /* compute polynomial coefficients of sub-curve segment */
    }
}
..................
```

Figure 4. The OpenMP parallel code

The piecewise iterative curve fitting plays a key role in the process of region structure feature representation and its execution time can not be ignored. In addition, with the image size increasing, the execution time consumed by piecewise iterative curve fitting will also increase. That will affect the efficiency of image reconstruction. In order to solve this problem, we take full advantage of potential parallel features of image reconstruction algorithm and propose a multi-threaded piecewise iterative curve fitting which is based on the OpenMP technique [23]. At the same time, multi-threaded parallelism based on OpenMP does not produce communication overhead. Fig.4 illustrates the OpenMP parallel piecewise iterative curve fitting in the form of code.

C. Effective Load-Balancing

There is effective load-balancing in our hybrid MPI-OpenMP parallel programming approach.

On the one hand, because all regions have same texture sample size and texture synthesis parameters, the time consumed by sample selection and texture synthesis is also same. Because the contour of each region is different, the time consumed by contour tracing, down sampling, piecewise iterative curve fitting and contour restoration is also not same. But in the real environment, since the time consumed by the latter is smaller than the time consumed by the former, the difference of total execution time among processors is small.

On the other hand, in the implementing procedure of the piecewise iterative curve fitting, we execute a multi-threaded parallelism based on OpenMP. This way, though the contour of each region is different, the parallel execution time consumed by piecewise iterative curve fitting is almost same. Therefore, the difference of total execution time among processors will become smaller. Clearly, multi-threaded parallelism based on OpenMP is beneficial to load-balancing.

IV. Experimental Results And Performance Analysis

The proposed framework is composed of two parts. The first part of the proposed framework is in Sect.2, which derives the image reconstruction algorithm. In Sect.3, the second part of the proposed framework shows the details of the hybrid MPI-OpenMP approach. Experimental results are then given according to these two parts. For the first part, we perform experiments in Sect.4.1 to evaluate our image reconstruction algorithm. For the second part, we perform experiments in Sect.4.2 to evaluate the time performance of our hybrid MPI-OpenMP approach.

A. Evaluation on Image Reconstruction Algorithm

We test our image reconstruction algorithm on some natural images with massive textures. For multi-regions extraction, the JSEG algorithm has three parameters which need to be specified artificially [15]. And we specify the three parameters as follows: TQUAN = -1, NSCALE = -1, threshcolor = 0.8, which make the performance of homogeneity detection perfect. For the piecewise iterative curve fitting, the split threshold amounts to 5~10 and the degree of polynomial of curve fitting is specified as 4 generally. Here, the JPEG version used is JPEG Imager 2.1.2.25 and the JPEG2000 version used is Kakadu_V2.2.3.

In order to illustrate the implementing procedure of the proposed image reconstruction algorithm, we show the results of each stage in Fig.5. Two regions of original image are obtained through multi-regions extraction, as is shown in Fig.5 (b). Fig.5 (c) and Fig.5 (d) show the reconstructed contour and the dilated contour of the upper region, respectively. Fig.5 (e) and Fig.5 (f) display the reconstructed contour and the dilated contour of the lower region, respectively. Fig.5 (g) displays the synthesized texture of the upper region and Fig.5 (h) displays the synthesized texture of the lower region. Finally, the reconstructed image is show in Fig.5 (i).

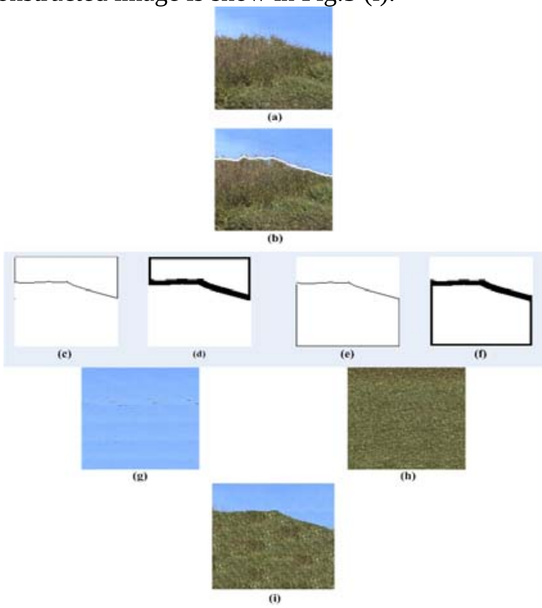

Figure 5. Diagram shows results of each phase

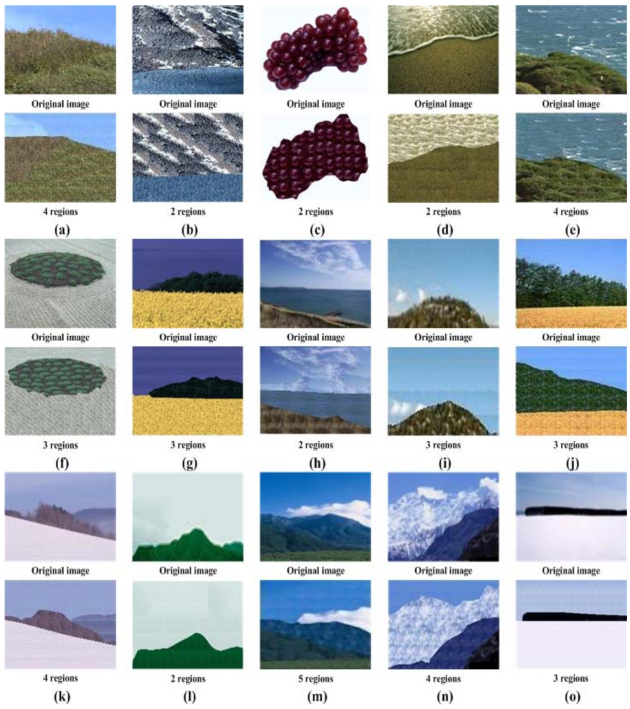

Figure 6. Reconstruction results

There are many reconstruction results with different number of regions in Fig.6. Experimental results show these images reconstructed by our algorithm have visual impairment. But visual impairment of JPEG images and JPEG2000 images is almost imperceptible. The reason why reconstructed results have visual impairment is that loss of non-texture detailed information. This loss is the main cause which causes the visual impairment of reconstructed results. Distortion of regional contour isn't obvious.

TABLE I.
COMPARISON OF BYTES REQUIRED BY JPEG, JPEG2000 AND OUR ALGORITHM

Original Image		Fig.6 (c)		Fig.6 (d)		Fig.6 (f)		Fig.6 (g)		Fig.6 (j)		Fig.6 (l)		Fig.6 (m)		Fig.6 (n)	
		2 regions		2 regions		3 regions		3 regions		3 regions		2 regions		5 regions		4 regions	
JPEG	Bytes	39656		28954		39512		15876		32922		6197		6822		11335	
JPEG2000	Bytes	30988		25538		29475		12553		14481		5364		6752		9393	
our algorithm	Q / T	1664	13872	832	8772	2848	8649	1344	3591	736	5625	832	2156	1504	3556	1760	3972
	Bytes	15536		9604		11497		4935		6361		2988		5060		5732	

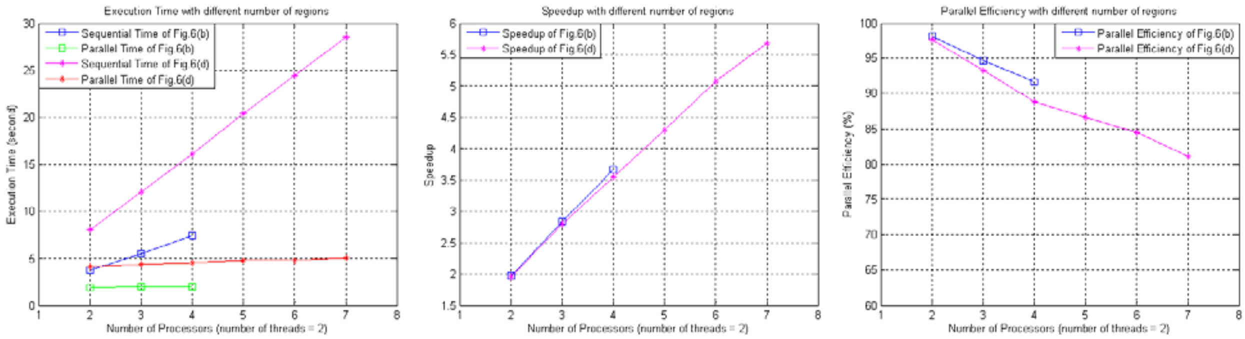

Figure 7. Execution time, Speedup and Parallel efficiency with different number of regions

In our image reconstruction algorithm, bytes required by image reconstruction are computed by (11):

$$\Omega = \sum_{i=1}^{N}(Q_i + T_i) \tag{11}$$

Where, Ω denotes bytes required by image reconstruction; N is number of regions; i is index of region; Q_i is bytes of polynomial coefficients representing regional contour with index i ; T_i is bytes of regional texture sample with index i .

In Table 1, we show number of bytes used in our image reconstruction algorithm and compare them to the JPEG and JPEG2000 compression for the equivalent images. Our reconstructed images are fair approximations and we may do more useful attempt for establishing more sophisticated image coding algorithms to reconstruct very realistic images.

B. Evaluation on Hybrid MPI-OpenMP Approach

We implement the hybrid MPI-OpenMP approach using standard C with Microsoft MPI (MS-MPI) and OpenMP. A series of tests have been performed on a dedicated cluster of 8 computing nodes running Windows HPC Server 2008, each with 2 Quad-core Xeon X5430 (2.66GHz) and 16GB memory and connected by an Infiniband switch.

Now we turn to the time performance measurements of the proposed hybrid MPI-OpenMP approach. We use original images in Fig.6 (b) and Fig.6 (d) to measure the execution time, speedup and parallel efficiency of our implementation. In addition, considering the actual characteristics of image shape, number of regions of original image in Fig.6 (b) is 2, 3 and 4, respectively; and number of regions of original image in Fig.6 (d) is 2, 3, 4, 5, 6 and 7, respectively. Although the number of threads can be arbitrarily set, the number of threads is specified as two in really experimentation. These measurements are demonstrated in Fig.7.

As shown in Fig.7, we can see clearly that the execution time can be reduced greatly by image feature extraction and reconstruction in parallel. Moreover, the more number of regions is, the more the saved time is. In addition, high speedups are also obtained. Moreover, the more number of regions is, the higher the speedup is. However, further research and improvement are still required for the low parallel efficiency when the number of processors is increasing. To sum up, we believe that image feature extraction and reconstruction in parallel is a feasible approach for image reconstruction algorithm. And it will perform better as the further development of parallel techniques.

V. CONCLUSIONS

Perceptual coding should take full advantage of the results from human visual system (HVS) studies. Image can be parsed into medium-level vision representation: regional contour curve and texture. An effective image reconstruction algorithm is proposed in this paper. Through selecting the smallest region texture sample, the image reconstruction algorithm can effectually eliminate perceptual redundancy. Compared with JPEG and JPEG2000 compression, our algorithm can get more coding gain but remain most image information people care. Simulation results show that our algorithm can obtain optimal reconstructed images. The hybrid MPI-OpenMP approach proposed is implemented on a dedicated cluster. This parallel approach can significantly improve efficiency of image reconstruction. Response time of restoring image from original image with many regions can be reduced greatly.

In our experiments, several limitations are found for the image reconstruction algorithm. One case is regional shape distortion. Another case is how to handle the loss of non-texture detailed information. Future research work is on how to solve these problems and establish more sophisticated coding algorithms to reconstruct very realistic images.

ACKNOWLEDGMENT

We acknowledge the funding provided by the National Natural Science Foundation of China key project (Grant No. 60832004) and youth project (Grant No. 51008009).

REFERENCES

[1] M. Kunt, A. Ikonomopoulos, M. Kocher, "Second-Generation Image Coding Techniques", Proc. IEEE, Vol. 73, No. 4, Apr. 1985, pp.549-574.
[2] R.A. DeVore, B. Jawerth, B.J. Lucier, "Image Compression Through Wavelet Transform Coding", IEEE Transactions on Information Theory, Vol. 38, No. 2 , Mar. 1992, pp.719-746.
[3] L. Thomas, F. Deravi, "Region-based Fractal Image Compression using Heuristic Search", IEEE Transactions on Image Processing, Vol. 4, No. 6, Jun. 1995, pp.832-838.
[4] H. Hartenstein, M. Ruhl, D. Saupe, "Region-based Fractal Image Compression", IEEE Transactions on Image Processing, Vol. 9, No. 7, Jul. 2000, pp.1171-1184.
[5] J.W. Han, K.N. Ngan, "Unsupervised Extraction of Visual Attention Objects in Color Images", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 16, No. 1, Jan. 2006, pp.141-145.
[6] T. Sikora, "Trends and Perspectives in Image and Video Coding", Proc. IEEE, Vol. 93, No. 1, Jan. 2005, pp. 6-17.
[7] P. Ndjiki-Nya, B. Makai, G. Blattermann, "Improved H.264/AVC Coding using Texture Analysis and Synthesis", Proc. IEEE Int. Conf. Image Processing, Sept. 14-17, 2003, pp. II-849-852.
[8] K. Sun, Y.Y. Yang, L. Ye, Q. Zhang, "Image Restoration Using Piecewise Iterative Curve Fitting and Texture Synthesis", Proc. 4th Int. Conf. Intelligent Computing, Shanghai, China, Sept. 15-18, 2008, pp. 1056-1063.
[9] R.J. Chen, L.G. Liu, G.C. Dong, "Local Resampling for Patch-based Texture Synthesis in Vector Fields", International Journal of Computer Applications in Technology, Vol. 38, No. 1/2/3, Jul. 2010, pp.124-133.
[10] D.S. Wickramanayake, E.A. Edirisinghe, H.E. Bez, "Transform Domain Texture Synthesis", Signal Processing: Image Communication, Vol. 23, No. 1, Jan. 2008, pp.1-13.
[11] A.A. Efros, W.T. Freeman, "Image Quilting for Texture Synthesis and Transfer", ACM SIGGRAPH, 2001, pp.341-346.
[12] G. Peyre, "Texture Synthesis with Grouplets", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 32, No. 4, Apr. 2010, pp.733-746.
[13] P.P. Bartakke, S.A. Vaidya, M.S. Sutaone, "Refining Structural Texture Synthesis Approach", IET Image Processing, Vol. 5, No. 2, Mar. 2011, pp. 184-189.
[14] Y. Deng, B.S. Manjunath, H. Shin, "Color Image Segmentation", Proc. IEEE Computer Vision and Pattern Recognition, Fort Collins, USA , Jun. 23-25,1999, pp. 446-451.
[15] Y. Deng, B.S. Manjunath, "Unsupervised Segmentation of Color-Texture Regions in Images and Video", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23, No. 8, Aug. 2001, pp.800-810.
[16] F. Jing, M. Li, H.J. Zhang, B. Zhang, "An Efficient and Effective Region-based Image Retrieval Framework", IEEE Transactions on Image Processing, Vol. 13, No. 5, May 2004, pp.699-709.
[17] T. Adamek, N. O'Connor, "Efficient Contour-based Shape Representation and Matching", Proc. 5th ACM SIGMM Int. Workshop on Multimedia Information Retrieval, Nov. 2003, pp.138-143.
[18] P. Dollar, Z.W. Tu, S. Belongie, "Supervised Learning of Edges and Object Boundaries", Proc. IEEE Computer Vision and Pattern Recognition, 2006, pp.1964-1971.
[19] D.R. Martin, C.C. Fowlkes, J. Malik, "Learning to Detect Natural Image Boundaries using Local Brightness, Color, and Texture Cues", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 26, No. 5, May 2004, pp.530-549.
[20] Q.L. Tang, N. Sang, T.X. Zhang, "Contour Detection based on Contextual Influences", Image and Vision Computing, Vol. 25, No. 8, Aug. 2007, pp.1282-1290.
[21] C.H. Chen, P.S.P. Wang, Handbook of Pattern Recognition and Computer Vision, World Scientific Publishing Co., 1999, Ch. 2, pp 218-222.
[22] L. Liang, C. Liu, Y. Xu, "Real-Time Texture Synthesis by Patch-based Sampling", ACM Transactions on Graphics, Vol. 20, No. 3, Jul. 2001, pp.127-150.
[23] M.J. Quinn, Parallel Programming in C with MPI and OpenMP, McGraw-Hill Companies, Inc., 2004.

Jinliang Wan is lecturer from Henan University of Economics and Law. My research area focuses on image and video coding, real-time digital image processing and high performance computing. I received the B.E. degree in Communication Engineering from Zhengzhou University, Zhengzhou, China, and the M.Sc. degree in Communication and Information System from Communication University of China, Beijing, China. I am currently lecturer in Henan University of Economics and Law, Zhengzhou, China, and Ph.D. candidate in Information Engineering School, Communication University of China, Beijing, China. Now, I take part in one National Natural Science Foundation of China key project and one youth project.

Yanhui Liu, Postdoctor, Certified structural engineer, 1st. grade of China. State Key Laboratory for Seismic Reduction /Control & Structural Safety (Cultivation), Guangzhou University. Research field: structure vibration control.

In 2003, 2007 and 2009, I obtained the bachelor's degree of engineering in Hainan University, master degree of engineering in Lanzhou University of Technology and doctor degree of engineering in Lanzhou University of Technology respectively. In 2009-2011, I went in for postdoctor in Beijing University of Technology. Now, I am working in State Key Laboratory for Seismic Reduction /Control & Structural Safety (Cultivation), Guangzhou University. Now, I hold one National Natural Science Foundation of China and one Postdoctoral Science Foundation of China, obtain one patent of China and have published twenty academic paper of which most have been indexed SCI or EI.