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 effectively 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 retain 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
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.

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).

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 implementing procedure is described as the following steps:

- Connecting the two farthest edge points \(A\) and \(B\) into a line segment \(L_{AB}\), and then 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):

\[
x(y_A - y_B) + y(x_A - x_B) + x_Ay_B - x_By_A = 0 \quad (1)
\]

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

\[
d_s = \frac{r_s}{\Delta_{AB}} \quad (2)
\]
Where, $s$ denotes the position of any point along the curve $AB$, and
\[ r_s = x_s(y_A - y_B) - y_s(x_A - x_B) + x_Ay_B - x_By_A \] (3)

\[ \Delta_{AB} = \|A - B\| \] (4)

Thus, the maximum absolute error is:
\[ MAE = \max_{s \in [A, B]} |d_s| \] (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 $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ 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):
\[ F_m(x) = \sum_{i=0}^{m} C_i P_i(x) \]
\[ = C_0P_0(x) + C_1P_1(x) + \cdots + C_mP_m(x) \] (6)

Our purpose is to calculate $P_0, P_1, \ldots, P_m$, where $m$ denotes order of the fitted polynomial. The orthogonal function family can be constructed by (7):
\[ P_0(x) = 1 \]
\[ P_1(x) = (x - A_1)P_0(x) \]
\[ P_2(x) = (x - A_2)P_1(x) - B_2P_0(x) \] (7)
\[ \ldots \]
\[ P_{j+1}(x) = (x - A_{j+1})P_j(x) - B_{j}P_{j-1}(x) \]

Where,
\[ A_{j+1} = \sum_{i=1}^{j+1} x_iP_{i+1}^2(x_j), B_j = \sum_{i=1}^{j} P_{i+1}^2(x_j), C_j = \sum_{i=1}^{j} P_{i}^2(x_j) \] (8)

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

Finally, we may calculate the polynomial coefficients $P_0, P_1, \ldots, P_m$ 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:
\[ \rho(u,v) = \frac{MN}{(M - x)(N - y)} \sum_{u=1}^{M-1} \sum_{v=1}^{N-1} I(u, v)I(u + x, v + y) \] (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:
\[ y = P_0 + P_1x + P_2x^2 + \cdots + P_mx^m \] (10)

Where, the polynomial coefficients $P_0, P_1, \ldots, P_m$ 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 $\tilde{x} = \{x_m, x_{m+1}, \ldots, x_n\}$, where each element is the $x$-coordinate of one point on the curve. Substitute elements of $\tilde{x}$ 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](image)

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

![Figure 4. The OpenMP parallel code](image)

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).

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.

<table>
<thead>
<tr>
<th>Original Image</th>
<th>Fig.6 (c)</th>
<th>Fig.6 (d)</th>
<th>Fig.6 (f)</th>
<th>Fig.6 (g)</th>
<th>Fig.6 (j)</th>
<th>Fig.6 (m)</th>
<th>Fig.6 (p)</th>
</tr>
</thead>
<tbody>
<tr>
<td>JPEG</td>
<td>Bytes</td>
<td>30656</td>
<td>29254</td>
<td>36512</td>
<td>19876</td>
<td>30252</td>
<td>6197</td>
</tr>
<tr>
<td>JPEG2000</td>
<td>Bytes</td>
<td>30088</td>
<td>29238</td>
<td>29475</td>
<td>12553</td>
<td>14381</td>
<td>5264</td>
</tr>
<tr>
<td>Our algorithm</td>
<td>Q</td>
<td>1264</td>
<td>13872</td>
<td>832</td>
<td>5772</td>
<td>2848</td>
<td>8649</td>
</tr>
<tr>
<td></td>
<td>T</td>
<td>736</td>
<td>2362</td>
<td>4935</td>
<td>6361</td>
<td>2958</td>
<td>5060</td>
</tr>
</tbody>
</table>

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

\[ \Omega = \sum_{i=1}^{N} (Q_i + T_i) \]  

(11)

Where, \( \Omega \) 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


