A Parallel Image Fusion Algorithm Based on Wavelet Packet

Yinglei Cheng\textsuperscript{1,2}, Ying Li\textsuperscript{2}, Rongchun Zhao\textsuperscript{2}

\textsuperscript{1} The Telecommunication Engineering Institute, Air Force Engineering University  
Xi’an Shaanxi, 710077, E-mail: ylcheng718@163.com

\textsuperscript{2} Department of Computer Science and Engineering, Northwestern Polytechnical University  
Xi’an Shaanxi, 710072

ABSTRACT

The wavelet packet provides an accurate method for image fusion. However, the computation and time cost will increase with the size of image. Therefore it is difficult to achieve real-time fusion. After having analyzed the wavelet packet based image fusion algorithm on the single computer, according to the time complexity and the parallel character of the wavelet packet algorithm, we propose a wavelet packet based parallel image fusion algorithm under the situation of distributed storage. Based on the data local property of wavelet packet transform, we achieve the algorithm on the MPI (Message Passing Interface) parallel computing platform, which is composed by Pentium PC and 1000 Mbps Ethernet. The efficiency of parallel computing is studied under different image size and different cluster size. The experimental result shows that the algorithm has good parallel computing property.

1. Introduction

Recently, image fusion has become one of the focuses in image processing field. Especially with the development of the sensor and computer technologies, and the requirements of remote sensor image processing in many applications, such as meteorologic analysis, geological exploration and celestial bodies prediction, multi-sensor image fusion has been paid more attention. Many algorithms have been proposed via extensive and deep research by scholars concerned in the past 30 years. Among the fusion methods such as pixel level, character level and decision level, pixel level fusion methods are the most mature ones [1]. But a common problem of these methods is the computing cost and computing complexity, therefore it is difficult to implement fast and real-time processing. Especially for fusion method based on the wavelet packet transform, with the increasing of the image size and algorithm decomposition level, its computing cost will increase even more. In recent years the rapid development of computer hardware and parallel computing technique has laid a solid foundation for the complex computing problem, and has explored a broader way to parallel image fusion.

Using parallel technique is an effective way to accelerate image fusion processing. Pudua et al used instruction sets parallel technique to process remote sensor image. But this technique is closely correlated with hardware and general application is poor. Compared with this, computer clusters have become increasingly popular among researchers and users because of quality price ratio and good expansibility. Meislv and Yang et al have realized parallel processing of SAR image and other remote sensor images.

To accelerate the processing of image fusion, the parallel environment in the paper is a high capability parallel cluster with a certain parallel supporting software-MPI parallel program developing library, where message delivery parallel programming model is adopted. The parallel algorithm of multi-sensor image fusion based on the wavelet packet transform is proposed in this paper.

Section 1 introduces wavelet packet based serial fusion algorithm and estimates the computation cost. Section 2 discusses in detail the realization of parallel fusion algorithm. Section 3 presents the actual implementation in computer cluster system and capability test and analysis. Finally, the conclusion is drawn.

2. Wavelet Packet Based Serial Fusion Algorithm Estimation of Computation Cost

According to the Paper 5, the first step of multi-spectral TM image and SAR multi-resolution image serial fusion is the pre-processing of the whole TM image. It is the preparation for the post-processing. The computation cost of this step is small and the relationship with the following steps is serial. The post-processing of the seventh step is similar to the pre-processing. Therefore the first and last step can be excluded from the parallel processing. The steps 2, 3, 4, 5 and 6 are the wavelet packet processing for the images. In one decomposition procedure, the image is decomposed into 4 sub-images: LL, HH, LH and HL, and then each sub-image is decomposed again into 4 sub-sub-images. After that 16 sub-sub-images are produced. Each
corresponding pixel of the both images must be operated. In this case, the computation complexity is high, computation cost is large and it is time-consuming. With the increase of the image size, the decomposition levels increase in a non-linear way. If the image size is $M \times N$ and the decomposition level is $L$, then the computation complexity of the serial fusion algorithm is $O(M \times N \times L)$. When size of $M$, $N$ and $L$ is too large, the time cost is extremely big. Therefore accelerating the wavelet packet decomposition is the key step for accelerating the whole fusion algorithm.

3. Parallel Fusion Algorithm

3.1. The Theory of Parallel Wavelet Packet Algorithm

To achieve the parallel algorithm, the separation of the problem is the first step. There are two methods to separate the problem. One is function separation, i.e. the control parallel; the other is the parameter separation, i.e. the data parallel [5]. Through the analysis of the serial wavelet packet decomposition, we know that the coefficient of the wavelet will be the input of next level decomposition. Therefore every level’s decomposition is strongly relative. Then we cannot design the control parallel between different levels. However, the computation of the wavelet coefficient is independent in every level. And this computation is sequence invariant. Therefore we propose the data parallel method.

The basic idea is to separate the image with the size of $M \times N$ into the machine $P$ parts that contain $M \times N / P$ pixels in each one. Then every computer only needs to compute $M \times N / P$ data. The load for each computer is equal. In every level of the processing, every machine can get $M \times N / P$ mid-result of the coefficients. Then they send their mid-result to the master machine (i.e. the main processor). The final result is given in the master machine. In order to reduce the communicating time between machines, we can store the whole image in every machine. And that can reduce the time taken to send images at the beginning of computation and can improve the efficiency further. Based on this parallel idea to achieve the wavelet decomposition, the computation complexity is $O(N \times L)$.

3.2. Parallelization of the Wavelet Packet Algorithm

The Wavelet Packet transform is the improvement of wavelet transform. For an image with the size $M \times N$, the value of its every point is $x_{k, l}$. Then the image can be decomposed into 4 sequences.

$$C_{m, n} = \sum_{k, l} x_{k, l} x_{k, l}$$
$$D_{m, n} = \sum_{k, l} x_{k, l} x_{k, l}$$

Where $h$ and $g^i$ stand for the scale coefficient and wavelet coefficient respectively. This is the decomposition of the first level. In the next step’s decomposition, each one of the four sequences will be further decomposed into four other sequences. Then we can get altogether 16 sequences in the second level decomposition. And then each one of the 16 sequences will be decomposed,...the number of computers is $P$, input data $x_{k, l}$ ($k = 1, 2, \ldots, N - 1; l = 1, 2, \ldots, M - 1$) are decomposed in a parallel way; then the images of the height $N/P$ and the size $M \times N / P$ to different processors $P$. In general the input image has a large size. At the same time $h$ and $g^i$ have little non-zero value which always concentrates in the neighbor of subscript 0. Therefore computing the output sequence value in position $(m, n)$ only needs the points in the neighbor of $(2m, 2n)$. From this we can say that the computation is only local-dependent. Then we can separate the image into several parts with same size and make every computer calculate one part. When doing the decomposition for the inside points, there will be data exchange between computers. If we let border of every part overlapped, the amount of data exchange will reduce. The size of the overlapped part is decided by the non-zero points of $h$ and $g^i$. After one step decomposition, the distribution of the data should be rearranged between computers and data should be also exchanged. After some certain steps, the decomposition for the sequences between different machines is independent. And the data exchange is no more needed. Then the communication time can be reduced and the efficiency is improved. After the operation mentioned above, the problem of balance and reduction of data exchange in computation has been solved successfully.

3.3. The Achievement of the Parallel Fusion Algorithm

The hardware environment of the algorithm consists of 8 high-performance PCs connected through high speed LAN 1,000M Ethernet. In every machine it has the MPICH software which can support the message transform between the MPI based parallel program. The program is compiled with language VC++. All steps of the algorithm are as follows:

(1) Pre-processing: The multi-spectrum image in RGB space is converted into HIS space. Three components are $I$, $H$ and $S$. This step is similar to the serial
algorithm. The convert is done in the master computer, and then the I component is sent to every computer for further computation.

(2) Data partition: According to the number of computers \( P \), we partition the input image \( I \) and high space resolution image into sub-images of height \( N/P \) and size \( M \times N/P \), leaving one part for each computer.

(3) Locally output range computation: Computing the range of the two sub-images that should be calculated in each machine.

(4) Locally fusion and reconstruction computation: For each sub-image, weighted sum and fusion of the corresponding part of the two images are executed according to liner weighted fusion rule. Wavelet inverse transform for the image fused is executing and then the new I component is obtained.

Post-processing (data collection and arrangement): every machine sends its local computation results to the master machine. And the master machine does the inverse IHS transform with the new I component and the \( H, S \) component obtained in step one to get the final fused image.

4. Analysis of Parallel Algorithm Property

The common evaluation index of parallel algorithm is accelerated ratio and parallel efficiency. It reflects the capacity of algorithm obtained in solving practical problems with parallel computers. Specifically, the accelerated ratio of parallel system refers to the fact that for given application how many times of the operation speed of parallel algorithm have been increased compared with the speed of serial algorithm. It is defined as follows:

\[
S(n) = \frac{t_s}{t_p} \tag{1}
\]

\( t_s \) refers to the time used by a single computer while \( t_p \) refers to the time used by parallel operation of multi-computers. The parallel efficiency is defined as follows:

\[
E = \frac{S(n)}{n} = \frac{t_s}{t_p \times n} \tag{2}
\]

n refers to the number of computers.

To evaluate the capability and efficiency of the wavelet packet transform fusion algorithm proposed, fusion experiment has been done using SAR and TM multi-spectrum remote sensor image. The results indicate that the algorithm proposed in this paper not only can fuse SAR image and TM multi-spectrum image effectively, thereby the image whose space resolution and spectrogram information are superior to those of the source image, but also has comparatively high parallel accelerated ratio and parallel efficiency.

4.1. Linear Weighted Value Comparison

TM multi-spectrum image and SAR high space-resolution image are fused according to the weighted fusion rule in this paper. The weighted computation formula is:

\[
I = \sum_{i=1}^{n} \alpha_i I_i \tag{3}
\]

\[
\sum_{i=1}^{n} \alpha_i = 1 \tag{4}
\]

In the formula, \( \alpha_i \) is weighted coefficient; \( I_i \) refers to different sensor image; \( I \) is fusion image. In the experiment, \( n = 2 \). \( \alpha_1 \) refers to weighted coefficient of multi-spectrum image while \( \alpha_2 \) refers to weighted coefficient of SAR image. Fig.1 shows the fusion results of \( \alpha_1 \) and \( \alpha_2 \) in different value. According to the fusion result: the bigger the weighted coefficient of SAR image, relatively the higher the high space-resolution and the less the spectrum information. Reversely, the high space resolution of fusion image is relatively lower and the spectrum information is more. Generally, the size of weighted coefficient depends on experience. For the wavelet packet fusion, since the image can be decomposed freely according to the use specific requirements of fusion image, the weighted coefficient can be selected freely for different sub-image. It has high flexibility.

The weighted fusion rule does not need extra data communication in parallel computation and it helps improve parallel efficiency. As to the local energy fusion rule in parallel computation, border area fusion needs data communication among the computers. And the bigger the image, the bigger the communication cost with lower parallel efficiency. That is the reason why the former is adopted in the paper instead of the latter.

4.2. Analysis of Parallel Efficiency

It is put forward in this paper that the parallel algorithm can fuse SAR image and TM multi-spectrum image effectively. Table 1 indicates that although two serial steps are kept in the parallel algorithm, their complexity is comparatively trivial compared with that of realized parallel wavelet packet transform algorithm. Therefore the whole parallel algorithm still has very high efficiency. Since the weighted fusion rule is adopted, then computation is operated among more computers, no extra data
communication is needed. Therefore the bigger the image is, the higher the parallel efficiency and the bigger the accelerated ratio are.

For image size $512 \times 512$, the accelerated ratio of two parallel machines and eight machines are as much as 89.18% and 57.33% respectively; for image size $1024 \times 1024$, the accelerated ratio of two parallel machines and eight machines are as much as 91.00% and 79.75%.

<table>
<thead>
<tr>
<th>Image Size</th>
<th>Net-work Scale</th>
<th>Time Cost (seconds)</th>
<th>Accelerated Ratio</th>
<th>Parallel Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>512×512</td>
<td>1</td>
<td>3.2371</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1.8149</td>
<td>1.7796</td>
<td>89.18%</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>13.389</td>
<td>2.3307</td>
<td>58.27%</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>0.7058</td>
<td>4.5864</td>
<td>57.33%</td>
</tr>
<tr>
<td>1024×1024</td>
<td>1</td>
<td>10.7832</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>5.9247</td>
<td>1.8200</td>
<td>91.00%</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3.1705</td>
<td>3.1054</td>
<td>85.03%</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>1.6902</td>
<td>6.3798</td>
<td>79.75%</td>
</tr>
</tbody>
</table>

5. Conclusion

In this paper, we first proposed fusion algorithm of multi-sensor image bases on the wavelet packet transform, then analyzed the potential parallel capability. Based on these, the run environment of MPI parallel implementation is confirmed and relative techniques are presented. Through experimenting in the clusters of different machine numbers, the results show that the parallel implementation proposed in the paper has linear accelerated ratio, stable efficiency and ideal parallel capability.

The wavelet packet transform image fusion is a new kind of fusion technique with excellent performance; much attention has been paid to it. In the view of parallel implementation, this paper discusses its implementation in a rapid and real time way, it is of certain significance to the popularization and application of the wavelet packet based image fusion.

References

[7] Luigi Dadda.: Parallel algorithms and architectures for CPUs and dedicated processors: development and trends.