A New Distributed Histogram Equalization Processing Remote Sensing Images based on MapReduce Framework

Lipeng Ji [1, a], Xiaohui Hu [2, b]

[1]Beihang University, Beijing 100191, China

[2]Institute of Software Chinese Academy of Science, Beijing 100190, China

[a]jlp_1987@163.com, [b]hxh@iscas.ac.cn

Keywords: Remote Sensing Images, Distributed Histogram Equalization, Mapreduce.

Abstract. With the development of 3S technology and information technology, the remote sensing data is growing dramatically. A new distributed histogram equalization for processing remote sensing images based on MapReduce framework is proposed in this paper. Through distributed histogram equalization method, significant changes of the sensing images will be got by a parallel way. The method is both useful and necessary for processing such a huge amount of remote sensing data.

1. Introduction

Remote sensing data is most important foundation resources of the scientific study and production work in geo-information science and other related field, and its application has already covered all related fields of geo-information science, such as environment monitoring, geo-hazard early warning, land usage monitoring, water and ocean remote sensing, atmosphere study and so on. But with the development of 3S technology and information technology, the remote sensing data is growing dramatically, sometimes the accumulated amounts are several TB or PB [1]. The processing of the image data which has a tremendous volume, requires high computation ability, large storage capacity, which has been beyond the performance of common graphic workstations [2].

MapReduce is a programming model introduced by Google for writing applications that rapidly process vast amounts of data parallel on large clusters of computing nodes [3]. MapReduce will be widely adopted on processing the huge data [4]. Histogram equalization (HE) is a simple and effective technology in image enhancement, and widely used in remote sensing image [5]. Histogram equalization technique is to expand the gray level of the image with more pixels, and compress the gray scale with less number of pixels in the image, so that it can contrast the details and equalize the dynamic range of the image gray level. So, with the development of the huge data technology, a new distributed histogram equalization for processing remote sensing images based on MapReduce model is both needed and proposed.

2. MapReduce

MapReduce model is a parallel programming model serving for processing large data sets in a massively parallel manner [2, 3, 6]. Using MapReduce framework can turn a program processed by a single computer into a distributed form which can be processed parallelly by a computer cluster with Hadoop. The computation takes a set of input <*key*, *value*> pairs, and produces a set of output <*key*, *value*> pairs. The user of the MapReduce library expresses the computation as two functions: map and reduce.

Map, written by the user, takes an input pair and produces a set of intermediate <*key*, *value*> pairs. The MapReduce library groups together all intermediate values associated with the same intermediate key and passes them to the reduce function. The reduce function, also written by the user, accepts an intermediate key and a set of values for that key. It merges these values together to form a possibly smaller set of values. The intermediate values are supplied to the user's reduce function via an iterator. This allows the user to handle lists of values that are too large to fit in memory. In other words, the user writes code to fill in a MapReduce specification object with the names of the input and output files and optional tuning parameters, and then invokes the MapReduce function, passing it to the specification object. The MapReduce implementation of operational processes as Fig.1 shows. More discussion of specific applications of MapReduce can be found elsewhere.

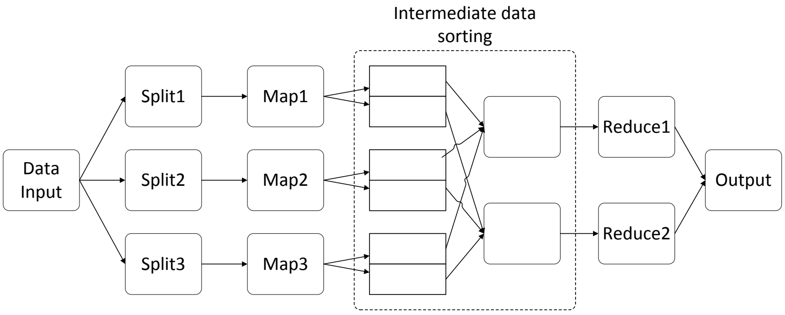


Fig.1 Operation process of MapReduce

3. Distributed histogram equalization

3.1 Basic histogram equalization.

Histogram equalization (HE) is a very popular technique for enhancement images. It is the most commonly used method due to its simplicity and comparatively better performance on almost all types of images. HE performs its operation by remapping the gray levels of the images based on the probability distribution of the input gray levels.

For a given image ***X***, the probability density function $p(X_k)$ is defined as

$$p(X_k) = \frac{n^k}{n} \tag{1}$$

For $k = 0,1, \dots, L-1$, where n^k represents the number of times that the level X_k appears in the input image ***X*** and n is the total number of samples in the input image. Note that $p(X_k)$ is associated with the histogram of the input image which represents the number of pixels that have a specific intensity X_k. In fact, a plot of n^k vs. X_k is known histogram of ***X***. Based on the probability density function, the cumulative density function is defined as

$$c(x) = \sum_{j=0}^{k} p(X_j) \tag{2}$$

where $X_k = x$, for $k = 0,1, \dots, L-1$. Note that $c(X_{L-1})$=1 by definition. HE is a scheme that maps the input image into the entire dynamic range, (X_0, X_{L-1}), by using the cumulative density function as a transform function. Let's define a transform function $f(x)$ based on the cumulative density functions as

$$f(x) = X_0 + (X_{L-1} - X_0)c(x) \tag{3}$$

Then the output image of HE, $\mathbf{Y} = \{Y(i,j)\}$, can be expressed as

$$\mathbf{Y} = f(\mathbf{X}) = \{f(X(i,j)) | \forall X(i,j) \in \mathbf{X}\} \tag{4}$$

Generally, there are many other improved methods based on traditional HE method mentioned above [5, 7, 8, 9, 10].

3.2 Distributed histogram equalization based on MapReduce framework.

Considering the computation of HE is first to get the probability density function of the original image by counting the number of existing pixel gray values, and then changing the original image into a new image with the same distribution over the entire luminance range by seeking a special transform, so it can be applied to the MapReduce framework. Then distributed histogram equalization (DHE) based on MapReduce framework is proposed.

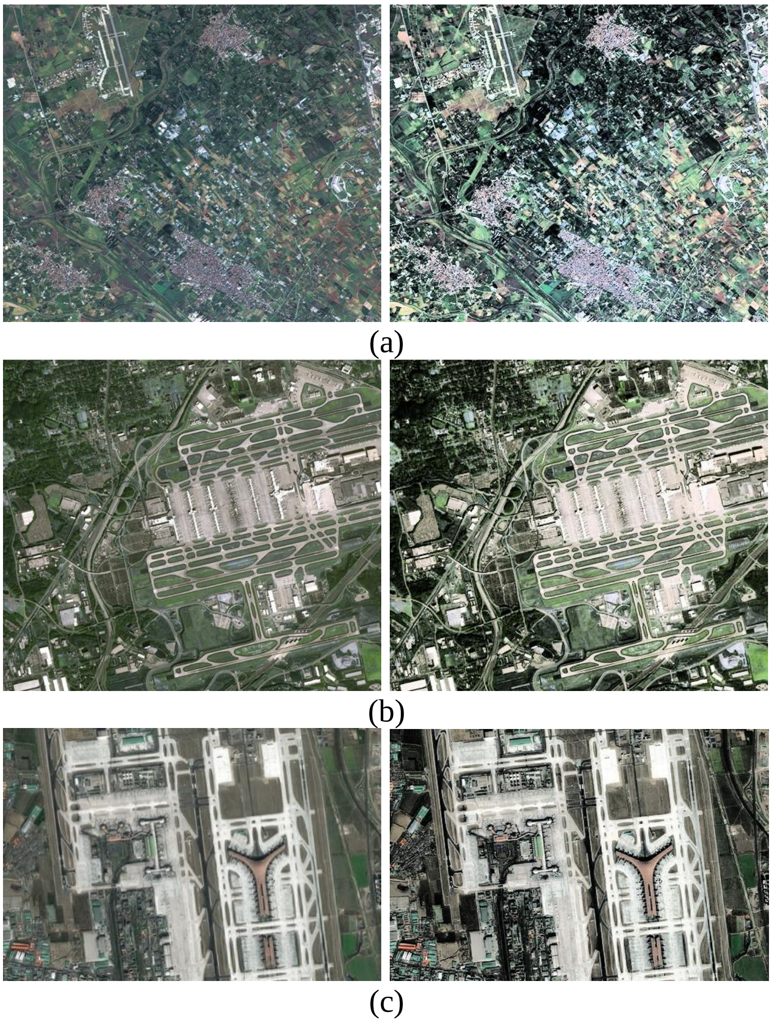
More specifically, counting the number of gray values of different pixels to get the original image's probability density function can be realized in Map function, which are steps (1) and (2). While generating the new image process is done in Reduce function by finding the corresponding pixel after transformation, which are steps (3) and (4).

In order to customize the MapReduce framework, which is essentially designed for text processing, some changes in Java classes are inevitable for image processing needs. The first thing to consider when using MapReduce for image processing algorithms is how to customize its data types, classes, and functions to read, write, and process binary image files as can be seen in sequential Java programs running on a PC. So, it is necessary to design an ImageFileInputFormat class inherits from the FileInputFormat <*Text*, *Image*> class, which takes an image file as an input slice. Then design an ImageRecordReader class inherits from the RecordReader <*Text, Image*> class, which transforms input slices into a <*key*, *value*> pair. The *Text* and *Image* are the original image file's name and its content respectively. Design an ImageOutputFormat class inherits from the FileOutputFormat <*Text, Image*> class. And Design an ImageRecordWriter class inherits from the RecordWriter <*Text*, *Image*> class. The *Text* and *Image* are the generating image file's name and its content. While FileInputFormat and FileOutputFormat are as the basis classes existing in the org.apache.hadoop.mapreduce package API.

Both finishing the four class design and four steps of HE, a DHE can be well designed.

4. Experimentation

The computer cluster was already installed Hadoop for providing MapReduce framework operating environment. The data source was a real color satellite image supplied by Google Earth. Subsequent processing was done on those images to change their sizes so that each of images within an airport. Using three images for instance, their sizes are 512×429 pixels, 493×433 pixels and 500×333 pixels. They were also saved by JPEG format, so that Java compiler can open them successfully. The images left below are the original remote sensing images, and right are the images processed by distributed histogram equalization.



(a)

(b)

(c)

Fig.2 Original remote sensing images and its results

All of these three groups of images can be processed well by distributed histogram equalization proposed in this paper. It can be seen that all of the result images show high performances in enhancing the contrasts. The brightness of images are changed significantly by DHE. Especially in (a) group, the trees surrounding the airport can be well apart.

5. Conclusion

A new distributed histogram equalization for processing remote sensing images based on MapReduce model is proposed in this paper. Through DHE method, significant changes of the sensing images will be got. Due to the fact that DHE is based on MapReduce framework, DHE will be well adapted to processing the huge remote sensing data. DHE is also based on the traditional and simple HE method, it is necessary to research improving HE methods based on MapReduce future.

References

[1]. Ma Y, Wu H, Wang L, et al. Remote sensing big data computing: Challenges and opportunities. Future Generation Computer Systems. Vol. 51(2015) No. 1, p. 47-60.

[2]. Almeer M H. Cloud hadoop map reduce for remote sensing image analysis. Journal of Emerging Trends in Computing and Information Sciences. Vol. 3(2012) No. 4, p. 637-644.

[3]. Dean J, Ghemawat S. MapReduce: Simplified Data Processing on Large Clusters. Conference on Symposium on Opearting Systems Design & Implementation. San Francisco, 2004, p. 107-113.

[4]. Sakr S, Liu A, Fayoumi A G. The family of mapreduce and large-scale data processing systems. ACM Computing Surveys. Vol. 46(2013) No. 1, p. 28-36.

[5]. Wang Y, Chen Q, Zhang B. Image enhancement based on equal area dualistic sub-image histogram equalization method. IEEE Transactions on Consumer Electronics. Vol. 45(1999) No. 1, p. 68-75.

[6]. Orlando S, Palmerini P, Perego R, et al. An efficient parallel and distributed algorithm for counting frequent sets. International Conference on High Performance Computing for Computational Science. Portugal, 2002, p. 421-435.

[7]. Kim Y T. Contrast enhancement using brightness preserving bi-histogram equalization. IEEE Transactions on Consumer Electronics. Vol. 43(1997) No. 1, p. 1-8.

[8]. Chen S D, Ramli A R. Contrast enhancement using recursive mean-separate histogram equalization for scalable brightness preservation. IEEE Transactions on Consumer Electronics. Vol. 49(2003) No. 4, p. 1301-1309.

[9]. Singh K, Kapoor R. Image enhancement using Exposure based Sub Image Histogram Equalization. Pattern Recognition Letters. Vol. 36(2014) No. 1, p. 10-14.

[10]. Kim M, Min C. Recursively separated and weighted histogram equalization for brightness preservation and contrast enhancement. Consumer Electronics IEEE Transactions on. Vol. 54(2008) No. 3, p. 1389-1397.