



A Parallel Retinex Image Enhancement Algorithm Based on OpenMP

Shixiong Cheng¹, Bin Liu^{1,2,3(✉)}, Dongjian He^{2,3,4}, Jinrong He⁵,
Yuancheng Li⁶, and Yanning Du⁷

¹ College of Information Engineering, Northwest A&F University,
Yangling, Shaanxi, China

liubin0929@nwsuaf.edu.cn

² Key Laboratory of Agricultural Internet of Things, Northwest A&F University,
Ministry of Agriculture and Rural Affairs, Yangling 712100, Shaanxi, China

³ Shaanxi Key Laboratory of Agricultural Information Perception and Intelligent
Service, Northwest A&F University, Yangling 712100, Shaanxi, China

⁴ College of Mechanical and Electronic Engineering,
Northwest A&F University, Yangling, Shaanxi, China

⁵ College of Mathematics and Computer Science,
Yan'an University, Yan'an, Shaanxi, China

⁶ School of Computer Science and Technology,
Xi'an University of Science and Technology, Xi'an, China

⁷ School of Computer Science and Engineering,
Xi'an University of Technology, Xi'an, China

Abstract. Retinex image enhancement algorithm occupies an important position in eliminating image uneven exposure, low contrast, and smog influence. However, with the increasing of image resolution, the real-time performance of the serial Retinex algorithm has not satisfied the requirements of practical applications. This paper proposes an OpenMP-based parallel Retinex algorithm. The parallelism of the Retinex algorithm is first identified by theoretical analyses. Then, the time-consuming sub-algorithms such as Gaussian convolution and exponential transformation, of the serial algorithm are designed and executed in parallel. On Tianhe-2 supercomputer platform, the experimental results show that the speedup of the parallel algorithm is significantly improved, and the test image set achieves an average speedup of 12. It indicates that the parallel algorithm can satisfy the needs of real-time processing in image enhancement field.

Keywords: Image enhancement · Parallel algorithm · Retinex · OpenMP · Agricultural image

1 Introduction

Image enhancement algorithms are basic work in many areas especially in the public security, biomedical field, health service, and marine information field, where significant achievements have been made in [1–3]. At present, researchers have proposed a great diversity of parallel image processing algorithms, such as CUDA-based image enhancement algorithms [4], and image processing algorithms based on multicore

DSP [5]. However, as one of the most important images processing technologies, the serial single-scale Retinex (SSR) algorithm is still too slow to finish the image enhancement tasks within an acceptable time.

In order to solve above problem, this paper proposes the parallel SSR image enhancement algorithm based on OpenMP. The parallel SSR image enhancement algorithm is implemented on the Tianhe-2 supercomputer using the OpenMP programming model, which is evaluated and achieves an average speedup of 12. The experimental results show that the proposed parallel algorithm can fulfill the needs of real-time processing in image enhancement field.

2 Parallel Design and Implementation

2.1 Parallelism Analyses

SSR algorithm enhances an image through the implementation of sub-algorithms such as Gaussian template, Gaussian convolution, and exponential transformation. As the data processed by these sub-algorithms is independent of each other, SSR algorithm has good parallelism. As illustrated in Fig. 1, the following three aspects are presented to analyze the parallelism of the serial SSR algorithm.

Parallelism 1: the blurred image estimating the incident illumination component is generated by Gaussian convolution operations. During this, each pixel is not associated with others. So, the image can be divided into sub-blocks for parallel computing.

Parallelism 2: the size of the Gaussian template is determined by the input parameters. When the Gaussian weight is normalized, each pixel is divided respectively by the sum of the weights, This process can be calculated in parallel.

Parallelism 3: the operation of exponential transformation can be also executed in parallel because there is no data dependence directly in those operations.

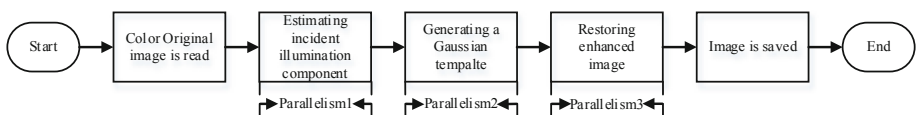


Fig. 1. The parallelism of single-scale Retinex algorithm

2.2 Parallel Design and Implementation of SSR Algorithm

In this section, parallel design of these sub-algorithms are firstly illustrated in Fig. 2 and then parallel implement are presented.

Parallel Design and Implementation of Gaussian Convolution. The subsequent operation of image segmentation into different data blocks is independent in Gaussian convolution serial algorithm. This is consistent with the parallel characteristics of OpenMP because no dependence between the data of non-direct adjacent pixels in the image. And a two-dimensional Gaussian function $G(x, y)$ could be written as the

product of two one-dimensional Gaussian functions $G(x)$ and $G(y)$, meaning that $G(x, y)$ can be calculated serially by convolution of $G(x)\delta(y)$ and $G(y)\delta(x)$. However each of one-dimensional Gaussian functions $G(x)$ could be executed in parallel. So the two-dimensional Gaussian convolution can be generated serially by two one-dimensional Gaussian convolutions performed respectively in parallel in the X and Y directions. And for example, there is an image which size is 7×7 , and the convolution kernel is 3×3 . It can be seen that the convolution operation requires 9 multiplications for each element in the image, so the total number of multiplication operations executed in sequential algorithm is $7 \times 7 \times 9 = 441$ times. In contrast, parallel execution requires only $2 \times 7 \times 9 = 126$ operations in the case of sufficient threads. And the execution time will be reduced and the speed will be increased compared with sequential algorithm.

Parallel Design and Implementation of Gaussian Template. The Gaussian template generation is mainly divided into two steps. The first step is that the weight sum is calculated serially, and the second step is that the normalization Gaussian template is generated in parallel. Supposing a 3 by 3 normalization Gaussian template is generated in serial algorithm with one thread and needs to be executed 9 times. However, in the case of parallelized execution with 9 threads, it only needs to be executed once.

Parallel Design and Implementation of Exponential Transformation. The original image and the Gaussian blurred image are set to the logarithmic domain to obtain a logarithmic image. The function of exponential transformation is to extend the image's high gray level and compress the low gray level. The most critical step in the exponential transformation is the linear mapping of each value. Assuming that the image size is 1000×1000 , it takes a lot of time to go through the linear mapping. If linear mapping is performed in parallel using 24 threads, the image only needs to perform 1737 operations rather than 1,000,000 in the serial algorithm. Therefore, it is very profitable to perform each worthy linear mapping in parallel.

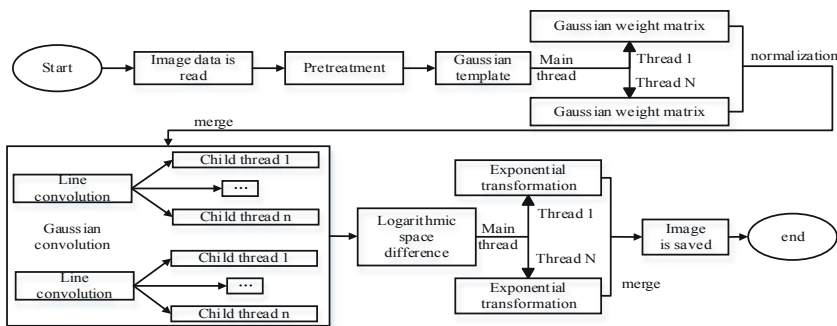


Fig. 2. Parallel design of single-scale Retinex algorithm

3 Experimental Results and Performance Analysis

3.1 Experimental Environment and Test Set

The experiment is performed on Tianhe-2 supercomputer equipped with 16000 nodes, which each note has three coprocessors, two Xeon E5-2692 processors, 24 cores and 64 GB of memory. The experimental environment is shown in Table 1. In this section, 10 different sizes of pictures are used to demonstrate the speedup performance of the parallel algorithm, the minimum size is 1730×883 , the maximum is 4000×3000 , and the format is JPG. These pictures are all agricultural images, including apples, pears, kiwis, farmland and mountain forests. They are from the shooting of the Dajiang UAV. The image test set is shown in Table 2.

Table 1. Experimental environment

Name	Description
Computer	Tianhe-2 Supercomputer
Processor	Intel Xeon E5-2692V2
Coprocessor	Intel Xeon phi
OS	Red Hat 4.4.7-4
Compiler	GCC 4.4.7
Cmake	Cmake-3.2.2
OpenCv	OpenCv-2.4.9

Table 2. Image test set

No	Description	Resolution	Size
img1	Farmland 1	1767×885	620 KB
img2	Farmland 2	1730×883	261 KB
img3	Qingyang Apple	2448×3264	2.20 MB
img4	Ruixue Apple	2784×1856	1.09 MB
img5	Ruiyang Apple	3088×2056	2.07 MB
img6	Kiwi 1	3264×2448	2.19 MB
img7	Kiwi 2	3264×2448	1.03 MB
img8	Farmland 3	4000×3000	4.36 MB
img9	Mountain 1	4000×3000	5.38 MB
img10	Mountain 2	4000×3000	4.72 MB

3.2 Speedup Comparison

The running time of serial and parallel algorithms respectively in Table 3 which Th represents thread. Within a certain range, the parallel SSR algorithm shortens the image processing time with the number of OpenMP threads increasing, and the average speedup is increased by about 12. After the parallel SSR algorithm are executed in parallel from dual thread to 24 thread, the speedup is obviously improved, and the parallel SSR algorithm can achieve near linear acceleration. The speedup curve is shown in Fig. 3. This experiment was carried out on a single node of Tianhe-2 supercomputer, each node had 24 cores, and the speedup reached a peak at 24 threads, making full use of the performance of multi-core. The speedup start to reduce at 32 threads because the number of threads at this time exceeds the number of CPU cores, but the processing time is still better than the serial algorithm. And the experimental results show that the speedup of the proposed parallel algorithm is significantly improved, and can satisfy the needs of real-time processing in image enhancement field.

Table 3. Comparison of running time (s)

Name	Serial	2Th	4Th	8Th	16Th	24Th	32Th
img1	7	4.71	2.442	1.279	0.726	0.542	0.627
img2	7	4.59	2.501	1.341	0.784	0.531	0.681
img3	39	24.799	12.746	6.733	3.897	2.86	3.445
img4	24	15.977	7.944	4.202	2.325	1.719	1.816
img5	30	19.089	9.985	5.128	2.888	2.072	2.39
img6	39	24.151	12.443	6.995	3.667	2.624	2.809
img7	39	24.437	12.281	6.609	3.619	2.626	2.837
img8	58	38.306	21.468	10.172	5.65	4.188	5.065
img9	59	36.762	18.923	9.598	5.549	4.779	4.743
img10	59	38.968	18.781	10.13	5.525	4.089	4.243

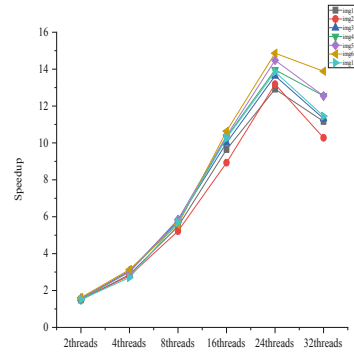


Fig. 3. Speedup comparison

4 Conclusion

This paper proposes a parallel SSR algorithm based on OpenMP. Compared to the serial Retinex algorithm, the proposed parallel algorithm can achieve an average speedup of 12, which represents a significant decrease in execution time. Experimental results show that the proposed parallel algorithm can acquire a significant increase in speedup and can better meet the requirements of real-time processing of the image enhancement algorithm in the image processing field.

Acknowledgment. This research is supported by the National Natural Science Foundation of China under Grant No. 61602388, by the Natural Science Basic Research Plan in Shaanxi Province of China under Grant No. 2017JM6059, by the Fundamental Research Funds for the Central Universities 2452019064, by the China Postdoctoral Science Foundation under Grant No. 2017M613216, by the Postdoctoral Science Foundation of Shaanxi Province of China under Grant No. 2016BSHEDZZ121, by the Fundamental Research Funds for the Central Universities under Grants No. 2452016081, by the Key Program of the National Natural Science Foundation of China under Grant No. 61834005, by China Postdoctoral Science Foundation No. 2018M633585, by Natural Science Basic Research Plan in Shaanxi Province of China No. 2018JQ6060, by the Natural Science Basic Research Plan in Shaanxi Province of China under Grant No. 2015JM6355, and by Doctoral Starting up Foundation of Yan’an University YDBK2019-06.

References

- Salazar-Colores, S., Cabal-Yepez, E., Ramos-Arreguin, J.M., Botella, G., Ledesma-Carrillo, L.M., Ledesma, S.: A fast image dehazing algorithm using morphological reconstruction. *IEEE Trans. Image Process.* **28**(5), 2357–2366 (2019)
- Muslim, H.S.M., Khan, S.A., Hussain, S., Jamal, A., Qasim, H.S.A.: A knowledge-based image enhancement and denoising approach. *Comput. Math. Organ. Theory* **25**, 108–121 (2018)
- Bhowmik, M., Ghoshal, D., Bhowmik, S.: An Improved method for the enhancement of under ocean image. In: 2015 International Conference on Communications and Signal Processing, Melmaruvathur, India, pp. 1739–1742. *IEEE* (2015)

4. Li, H., Xie, W.H., Wang, X.G., Liu, S.S., Gai, Y.Y., Yang, L.: GPU implementation of multi-scale retinex image enhancement algorithm. In: 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications, Agadir, Morocco, pp. 1–5. IEEE (2016)
5. Wang, G., Liu, X.: A parallel image processing platform based on multi-core DSP. In: 2017 IEEE/ACIS 16th International Conference on Computer and Information Science, Wuhan, China, pp. 775–779. IEEE (2017)