

Pixel-level Image Fusion Based on Programmable GPU

Li Hong

School of Information
Engineering
Xianyang Normal
University Shaanxi,China
110508@sina.com

Tang Gao-Feng

Department of science and
technology
Xianyang Normal University
Shaanxi,China
tangaofeng0043@163.com

Wu Fen-Xia

School of Information
Engineering
Xianyang Normal University
Shaanxi,China
wufeixia@126.com

Tan Cong-e

College of Chinese traditional
herb
Shaanxi university of Chinese
medicine
Shaanxi,China
tce2000@163.com

Abstract—A novel algorithm which is image fusion based on GPU is proposed. The fused rule is regional energy. In recent years, the power of the computing of GPU has been greatly improved, which results that using it for the general-purpose computing has a rapid development. The essay researches on implementing the oriental field algorithm on GPU, including selecting GPU memories and dividing blocks and threads of GPU kernel functions. The results of experiment on the GPU of NVIDIA GTX560 are given, which shows that our proposed algorithm can be applied to the field of image fusion. Experiment shows the proposed algorithm has faster calculation velocity and higher evaluation accuracy. The speed of the parallel algorithm is 200 times faster than that of the CPU-based implementation. Meanwhile the mutual information and $Q^{AB/F}$ parameters are higher than that of the CPU-based algorithm.

Keywords- image fusion, local feature, GPU

I. INTRODUCTION

Image fusion is a process of combining information from multiple images into a fused image. These images may come from one sensor or multiple sensors. The fused image can provide more interpretation capabilities, and is more useful and gives a better perceptual quality. The fused image can improve edge detection, image segmentation and object recognition in medical imaging and military applications. There are different methods available to implement image fusion[1]. Image fusion algorithms can be categorized into three levels: pixel level, feature level and decision level[2]. In all of those methods, multiscale transform based methods are the most successful category of techniques.

Typical multiscale transforms include the Laplacian pyramid[3], morphological pyramid[4], discrete wavelet transform[5], stationary wavelet transform[6], and dual-tree complex wavelet transform[7]. Recently developed multiscale geometry analysis, such as ridgelet transform[8], curvelet transform[9], the nonsubsampling contourlet transform[10], are also applied to image fusion. The discrete wavelet transform is a good time-frequency analysis features that make it widely used in image aspects, and it is suitable to represent image details. Ridgelet bases are suitable to express lines, and discrete cosine transform bases to textures which are periodic. However, the procedure of multiresolution image fusion are three steps. Firstly, the source images are decomposed into sub-images by a certain pyramid

algorithm. In the transform domain, the sub-images are combined by the fusion rule. A new set of coefficients can thus be obtained. By applying the inverse transformation, the fused image can be constructed.

The traditional fusion methods are based on the CPU. This process took a lot of time. In recent years, the programmable GPU is being studied and applied to image processing field. A GPU highly parallel structure makes them more effective than general-purpose CPUs for algorithms. GPU can get higher speed than by optimizing the GPU thread partition, and therefore can improve the system overall performance. In order to get higher evaluation accuracy and faster calculation velocity, image fusion based on programmable GPU is proposed.

The traditional fusion rules such as average method and the maximum absolute value method has simple principle and easy to achieve. In this paper, We make full use of the characteristics of the GPU, and the fusion rule is regional energy. The simulation results show that the proposed algorithm has higher evaluation accuracy and faster calculation velocity. The speed of the parallel algorithm is 78 times faster than that of the CPU-based implementation. Meanwhile the mutual information and $Q^{AB/F}$ parameters are higher than that of the CPU-based algorithm.

The remainder of the paper is organized as follows. Section 2 provides background on GPU and introduces its principle of operation. In Section 3, a fusion problem in the spatial domain is described. Simulation results and discussions are presented in Section 4. Finally, conclusions and suggestions for future work are given in Section 5.

II. GRAPHICS PROCESSING UNIT

In recent years, multiple processors technology is a research hot spot in the field of high performance computing[11]. A graphics processing unit (GPU) is a specialized electronic circuit designed to rapidly manipulate and alter memory to accelerate the building of images in a frame buffer intended for output to a display. Along with the improving performance of GPU, it can be programmed and operated. GPUs are widely used in image processing, embedded systems, mobile phones, personal computers, workstations and game consoles. Modern GPUs are very efficient at manipulating computer graphics, and their highly parallel structure makes them more effective than general-purpose CPUs for algorithms where processing of large

blocks of data is done in parallel[12-14]. GPU have many advantages in accelerated parallel processing, namely a large number of threads, high precision operation, rapid accessing graphics memory and minimal data coupling.

A CPU shares computing tasks across calling GPU-Kernel function. The GPU's core is Streaming Multi-processor(SM) array. Each SM has multiple Streaming Processor(SP). GPU memory composition mainly has two parts. One is Globla Memory, and the other is Shared Memory. The Shared Memory is divided into more equal banks. At the same time, the requests which are reading and writing are sent concurrently by multiple threads to different bank[15,16].

A GPU can play its powerful computation in programming model of single instruction and most data via Compute Unified device Architecture(CUDA). A GPU is a hardware coprocessor that accelerates computations for computer graphics applications in CUDA modal. In this modal, the GPU is coprocessor, while CPU is host. The responsibility of CPU is serial calculation, and the duty of GPU is parallel processing. The CPU and the GPU have their address space, respectively. The computational task is decomposed into a large number of threads which can be parallel processing. These threads are scheduled dynamically.

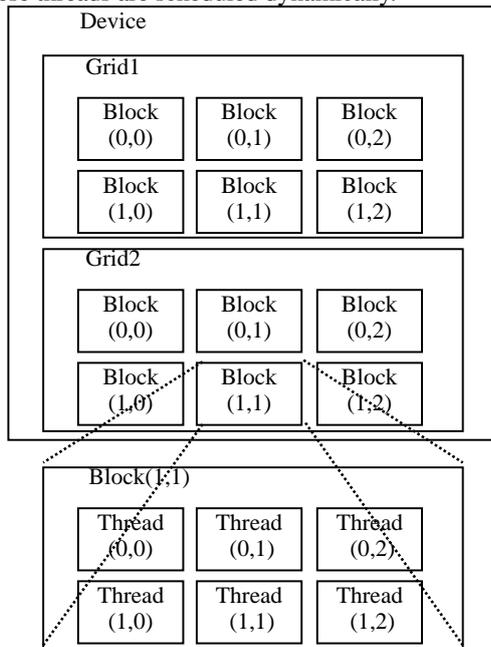


Figure 1. Linearly Separable for CUDA

III. IMAGE FUSION

Program on CUDA is composed of functions of host and kernel function of GPU. In this paper, program of host is responsible for the I/O data transmission, program control flow, allocating and freeing required memory of device and invoking kernel functions. Image fusion is executed in the module of device. Blocks are equal to the height of image each grid, and threads of GPU are 512 every block.

The local features of the images is expressed by pixels. We cannot ignored the correlation between pixels. Strong correlation exists in the image of a local area between pixels. The simple image fusion based on a pixel is logical but is one-sided. In order to get the better visual effect and the high quality fusion results, the algorithm based on image area is necessary. This paper put to use the regional energy. In this method, we should consider to correlation between pixels, and fusion pixels is in the center of the area.

The regional energy consider the correlation between pixels. The advantage of this method is edge smoothing for images and best visual effects. Sometimes the result image lost edge and texture information because of edge smoothing. Equation (1) is the regional energy. The reach size of local field is $3 \times 3, 5 \times 5$ or 7×7 .

They are defined by the following expression:

$$E(x, y) = \sum_{m \in M, n \in N} [D(x+m, y+n)]^2 \quad (1)$$

$$MED(x, y) = \sum_{m \in M, n \in N} \frac{D(x+m, y+n)}{M \times N} \quad (2)$$

$$Match(x, y) = \sum_{m \in M, n \in N} \frac{2 \times D_A(x+m, y+n) \times D_B(x+m, y+n)}{E_A(x, y) + E_B(x, y)} \quad (3)$$

$$F(x, y) = \begin{cases} \begin{cases} D_A(x, y) & E_A(x, y) \geq E_B(x, y) \\ D_B(x, y) & E_A(x, y) < E_B(x, y) \end{cases} \\ \quad \quad \quad Match(x, y) \leq Thr \\ \\ \frac{MED(x, y) + MED(x, y)}{2} \\ \quad \quad \quad Match(x, y) > Thr \end{cases} \quad (4)$$

where $MED(x, y)$ is mid-value. $Match(x, y)$ is matching degree. $F(x, y)$ is fusion coefficients. The local field of proposed algorithm is 3×3 .

IV. SIMULATION RESULTS AND DISCUSSIONS

In this section, we use proposed algorithm to fuse the images. Two sets of test images are employed for performance evaluation. In the medical images Med, one is captured by MRI and the other is captured by CT. The size of Med is 256×256 . For the multifocus images Clock, one is focused on left clock and the other is focused on right clock. The size of Clock is 512×512 . All experiments are implemented on an CPU of Intel(R) Core(TM) i5-2380P and GPU of NVIDIA GTX560. The simulation software is VS2010 and CUDA 4.2.

Fig.2 and Fig.3 show the fused result. The fusion rule is regional energy in spatial domain. One method is a serial execution on CPU, and the other is parallel execution on GPU. The image visuals are quite between (c) and (d). The fused images have clear object outline, good performance

and brightness. The results contain the most information of source images.

Table1 gives the statistical time of the serial and the parallel execution. Compared with on CPU, time of parallel execution would greatly cut. Meanwhile, The experimental results indicate that the larger size of image is, the higher the speedup ratio is. An objective evaluation criteria is applied to compare fusion results obtained using different methods. Table2 shows image quality via parameters. The results show that the scheme based on GPU is a little higher than on CPU both MI and $Q^{AB/F}$. But one reason is because image edge is automatic processing on GPU. Above all the proposed method has fast operation and high computing precision.

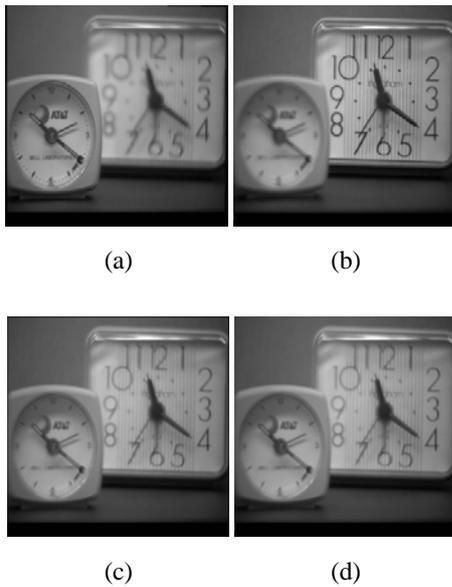


Figure 2. fusion results for Clock ((a)and(b) source images,(c) Fused image on CPU,(d) Fused image on GPU)

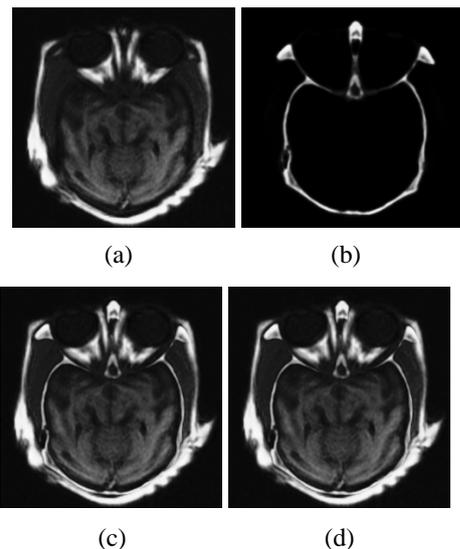


Figure 3. fusion results for Med ((a)and(b) source images,(c) Fused image on CPU,(d) Fused image on GPU)

TABLE I. SPEEDUP RATIO

image	size	Time(ms)		Speedup ratio
		CPU	GPU	
Clock	256 x 256	12.70194	0.82008	15.48865
	512 x 512	58.70829	0.75094	78.17973
	1024X1024	204.36244	0.86329	236.72513
Med	256 x 256	14.5327	0.80512	18.05035
	512 x 512	51.64376	0.79166	65.23477
	1024X1024	207.96473	0.77787	267.35152

TABLE II. FUSED RESULTS

Image	size	Fusion	MI	$Q^{AB/F}$
Clock	512 x 512	CPU	9.1196	0.5307
		GPU	9.1514	0.5315
Med	256 x 256	CPU	6.0225	0.8233
		GPU	6.0553	0.8251

CONCLUSION

In this paper, a novel scheme which is image fusion on GPU is proposed. The fusion rule is regional energy in spatial domain. One method is a serial execution on CPU, and the other is parallel execution on GPU. The essay also researches on implementing the oriental field algorithm on GPU, including selecting GPU memories and dividing blocks and threads of GPU kernel functions. Compared with serial execution on CPU, the advantage of parallel execution on GPU is high operation and high computing precision. Experiment shows the proposed algorithm has faster calculation velocity and higher evaluation accuracy. The speed of the parallel algorithm is 200 times faster than that of the CPU-based implementation. Meanwhile the mutual information and $Q^{AB/F}$ parameters are higher than that of the CPU-based algorithm. So our method have more advantages such as high degree of precision and fast computation speed.

ACKNOWLEDGMENT

This work is funded by the National Natural Science Foundation of China(NO.81072731), the Scientific research plan projects of Shaanxi Educational Committee (No.09JK811), the Scientific research projects of Xianyang Normal University (No.11XSYK329, 12XSYK072, 08XSYK339), Pedagogical Reforms Project funded by Xianyang Normal University(No.200902026)

REFERENCES

[1] S. Daneshvar, H. Ghassemian, "MRI and PET image fusion by combining IHS and retina-inspired models, " Information Fusion, vol.11, pp. 114-123, Feb 2010.
 [2] Z. Zhang, R. S. Blum. "A categorization of multiscale decomposition based image fusion schemes with a performance study for a digital

- camera application, " Proceedings of the IEEE, vol. 87, pp. 1315-1326, Aug 1999.
- [3] P. T. Burt, E. H. Andelson, "The Laplacian pyramid as a compact image code, " IEEE Transactions on Communications, vol. 31, pp. 532–540, Apr 1983.
- [4] V. S. Petrovic and C. S. Xydeas, "Gradient-based multiresolution image fusion," IEEE Trans. Image Process, vol.13, pp. 228-237, Feb 2004.
- [5] G. Pajares, J. Cruz, "A wavelet-based fusion tutorial," Pattern Recognition, vol. 37, pp 1855-1872, Sep 2004.
- [6] M. Beaulieu, S. Foucher, L. Gagnon, "Multi-spectral image resolution refinement using stationary wavelet transform, in: Proceedings of the International Geoscience and Remote Sensing Symposium, " pp. 4032–4034, 1989.
- [7] J. J. Lewis, R. J. O'Callaghan, S. G. Nikolov, D. R. Bull, C. N. Canagarajah, "Region-based image fusion using complex wavelets, in: Proceedings of the 7th International Conference on Image Fusion," pp. 555–562, 2004..
- [8] Chen, J. P. Zhang, Y. Zhang, "Remote sensing image fusion based on ridgelet transform, in: Proceedings of International Conference on Geoscience and Remote Sensing Symposium, "2005, pp. 1150–1153.
- [9] L. Tessens, A. Ledda, A. Pizurica, W. Philips, "Extending the depth of field in microscopy through curvelet-based frequency-adaptive image fusion, in:Proceedings of the International Conference on Acoustics, Speech and Signal Processing, "pp. I-861–I-864, 2007.
- [10] L. D. Cunha, J. P. Zhou, "The nonsubsampling contourlet transform: theory,design, and applications," IEEE Transactions on Image Processing, vol. 15, pp. 3089–3101, Oct 2006.
- [11] YU Wen-guang, WANG Wei-ping ,Hou Hong-tao, LI Qun. "Parallel Agent-based simulation on multi-core CPU and GPU heterogeneous platforms," Systems Engineering and Electronics, vol. 34, pp. 1716-1722, Aug 2012.
- [12] NVIDIA, CUDA C Programming Guide 3.2[EB/OL]. http://developer.download.nvidia.com/computer/cuda/3_2/toolkit/docs/CUDA_C_Programming_Guide.pdf.
- [13] NVIDIA CUDA [EB/OL].http://www.nvidia.com/object/cuda_home_new.html.
- [14] Harish P, Narayanan P J. "Accelerating large graph algorithms on the GPU using DUDA[c]," pp. 197-208, Dec 2007[Pro.14th Int'l Conf.High Performance Computing(HiPC'07)].
- [15] Katz G J, Kider J T, Jr. "All-pairs shortest-paths for large graphs on the GPU[C]," [Proc. of the 23rd ACM].
- [16] DI Peng, HU Chang-jun, LI Jian-jiang. "Efficient method for histogram generation on GPU," Computer Science, vol. 39, pp. 304-307, Mar 2012.