academicJournals

Vol. 8(33), pp. 1672-1675, 9 September, 2013
DOI: 10.5897/IJPS2013.3996
ISSN 1992 - 1950 © 2013 Academic Journals
http://www.academicjournals.org/IJPS

International Journal of Physical Sciences

Full Length Research Paper

Multi-core large-scale image-based modeling

Kun Li, Chao Song, Jingyu Yang* and Jianmin Jiang

Tianjin University, Tianjin 300072, China.

Accepted 4 September, 2013

Large-scale image-based modeling is a challenging and important research topic, which has wide applications in various areas. Existed serial methods consume a significant amount of time for large problems. With the emergence of multi-core computers, it is possible to speed up the method using central processing unit (CPU) parallelism. In this paper, we present the design and implementation of multi-core large-scale image-based modeling that exploits hardware parallelism to efficiently solve the large-scale 3-D scene reconstruction problem. We explore the use of multi-core CPU to achieve high speedups for the whole problem. The speedup ratio can be close to the number of cores. Experimental results show that our method achieves the acceleration of the whole algorithm and also ensures the accuracy of the 3-D reconstruction.

Key words: Large-scale image-based modeling, multi-core, central processing unit (CPU) parallelism, 3-D reconstruction, speedup ratio.

INTRODUCTION

Image-based modeling is a classic and hot issue in computer vision and image processing, and its goal is to create 3-D models from a collection of 2-D image measurements. Some methods (Lavoie et al., 2004; Wang et al., 2012) achieve the 3-D reconstruction with the help of structured light, but it is difficult for them to recover the textures of the scene. Another category of methods (Rander et al., 1997; Liu et al., 2011) depend on multi-camera arrays to obtain high quality 3-D models, but the costs of building multi-camera systems are very high. Recently, there has been a renewed interest in large-scale image-based modeling systems, especially those of which use the images captured by uncalibrated cameras, or collected from the Internet (Agarwal et al., 2010; Snavely et al., 2006). The problem for this kind of methods is usually large-scale: with hundreds of or thousands of input images. Many methods adopt serial processing mode, which needs several days to reconstruct a 3-D model from several hundreds of images. With the rapid development of parallel computation resources, Agarwal et al. (2009) achieve distributed computation running on a cluster of computers, and significantly shorten the running time. Since the hard disk space is local and not shared, they need to consider the problem of data distribution and communication. The emergence of multi-core computers motivates Wu et al. (2011) to propose a multi-core bundle adjustment algorithm (a key component of the problem) and achieve a 10x to 30x boost in speed over existing systems.

In this paper, we present a multi-core implementation for the whole framework of large-scale image-based modeling, which significantly improves the computational efficiency without loss of reconstruction quality. We use central processing unit (CPU) parallelism to achieve high speedups over existing serial systems. The speedup ratio can be close to the number of cores.

THE PROPOSED METHOD

Pipeline of the large-scale image-based modeling

As show in Figure 1, the large - scale image - based modeling

*Corresponding author. E-mail: yjy@tju.edu.cn.

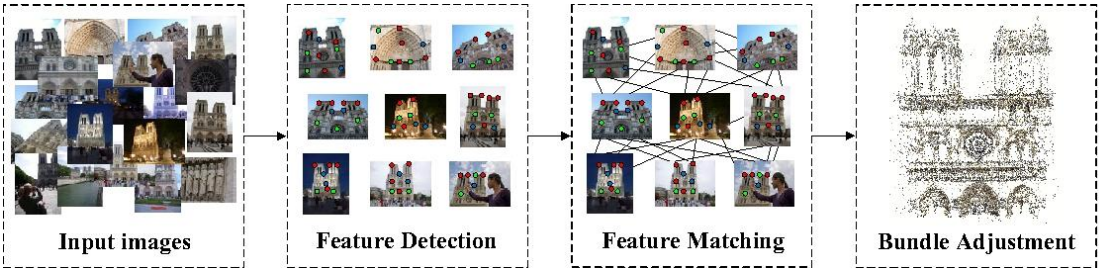

Figure 1. The pipeline of classic large-scale image-based modeling.

method mainly contains three steps: feature detection, feature matching, and bundle adjustment. The input images are captured with uncalibrated cameras or collected from the Internet, and the number of images is usually large. In the first step, existed methods process the images one by one: extracting the EXIF tags, recording the focal length, and generating a shell script which processes the operations of converting file formats, detecting the features with scale-invariant feature transform (SIFT), and compressing the features in a serial manner. The second step uses the Approximate Nearest Neighbor (ANN) library (Arya et al., 1998) to match SIFT features, which costs about half of the whole processing time. Given a set of measured image feature locations and correspondences, the goal of the final step is to find 3-D point positions and camera parameters that minimize the reprojection error. The optimization problem is usually formulated as a non-linear least squares problem and solved by the Levenberg-Marquardt (LM) algorithm (Nocedal and Wright, 2006).

Multi-core parallelized scheme

In this section, we present our multi-core parallelized scheme for the three steps of the large-scale image-based modeling.

Feature detection

Although shell programming does not support multi-thread, there is still a way to parallelize the work by creating children processes in a multi-core environment. We first divide the input images into M lists that are saved in M files, where M is the number of available cores. Then, we generate a script of extracting the EXIF tags and recording the focal length for each list, and create a child process to complete the corresponding operation. The same is used for generating the shell script for feature detection.

Feature matching

We use message passing interface (MPI), instead of multi-thread, to parallelize this step in distributed environment. MPI does not share variables in the global scope but maintains independent code and data for each process. These processes communicate with each other by the respective data buffer. Therefore, the features generated in the first step are not shared by these processes, which must be read more than once. Fortunately, we find that reading these data does not cost too much time and the time of reading keys for all these processes is the same as reading these keys once. We adopt master-slave mode to achieve process collaboration. The main process is responsible for dividing tasks and then also participates in the calculation. Denote $f(i)$ as the matching number for the image i. The amount of computation for the process j that is assigned the task of matching the images from m to n is:

$$F(j)=\sum_{i=m}^{n} f(i). \quad (1)$$

Hence, the main process divides the tasks so that the computation of each process is roughly equal, and then sends the image indexes to the corresponding process. Finally, the main process maintains the last task for its own computation.

Bundle adjustment

In this step, we use the computation interface provided by the PBA method (Wu et al., 2011). The reconstruction accuracy of the PBA method is less than that of the original serial method (Snavely et al., 2006), which will affect the initialization and further the final reconstruction. Therefore, for some important initialization, we use the original bundle adjustment method (Snavely et al., 2006), and then adopt the PBA method to speed up.

RESULTS

We evaluate the performances of the proposed method on three datasets: *Rilievo* (93 pictures) captured with a Samsung digital camera, *NotreDame1* (103 pictures) and *NotreDame2* (204 pictures) provided by Noah Snavely (Snavely et al., 2006). All the experiments are individually performed on two workstations: one with three-core CPUs clocked at 3.2 GHz and 4 GB RAM running a 32-bit Ubuntu operating system, and the other with eight-core CPUs clocked at 0.8 GHz and 16 GB RAM running a 64-bit OpenSUSE operating system (Figure 2).

Table 1 gives the quantitative evaluation of the proposed method for the three datasets, compared with a serial method (Snavely et al., 2006). The speedup ratio is $S=t_s/t_p$, which is the ratio of serial time t_s over parallel time t_p. This is an intuitive evaluation of how much is accelerated. The efficiency is $E=S/n$, where n is the number of threads or processes. The cost is $C=t_p\times n=t_s/E$. The degree of contribution of the

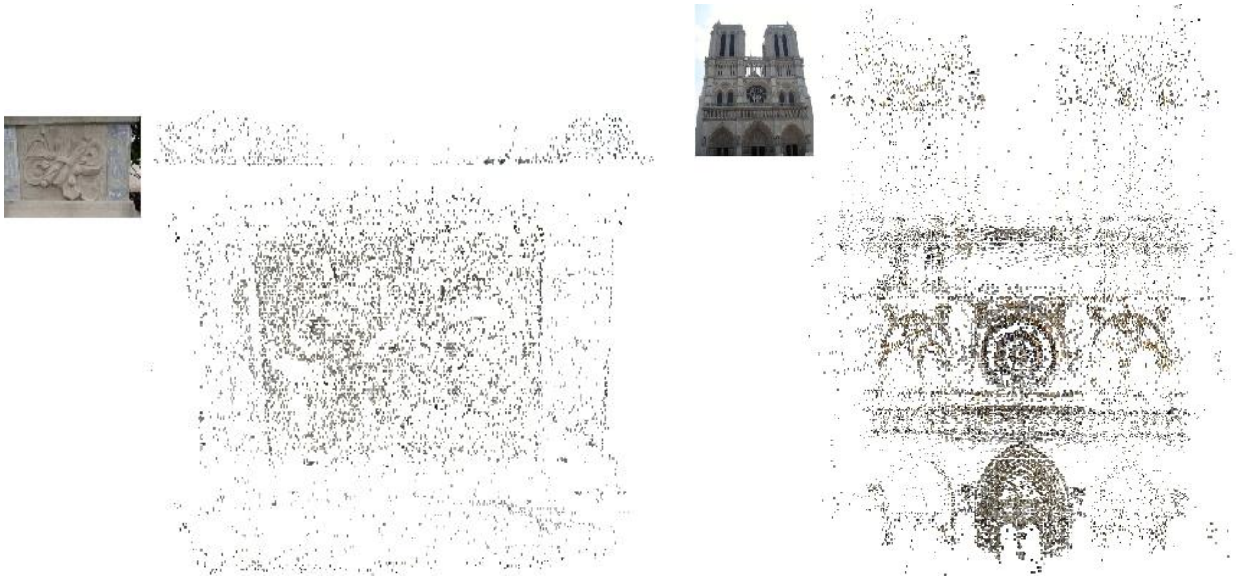

Figure 2. 3-D point clouds of (a) *Rilievo* and (b) *NotreDame1* reconstructed by our method.

Table 1. Quantitative evaluation of the proposed method for three datasets.

Dataset	Step	Serial time	Parallel time	Cores	Speedup ratio	Efficiency (%)	Cost	Proportion (%)	Degree of contribution
R	FD	5m22s	1m56s	3	2.78	92.53	5m48s	17.09	1.12
	FM	9m44s	3m27s	3	2.82	94.04	10m21s	31.00	1.25
	BA	16m18s	1m58s	3	8.29	276.27	5m54s	51.91	1.84
	All	31m18s	7m7s	3	4.33	131.24	23m51s		
N1	FD	40m20s	9m1s	5	4.58	91.68	45m5s	26.68	1.26
	FM	1h21m47s	21m27s	5	3.81	76.25	1h47m15s	54.10	1.66
	BA	29m3s	6m28s	5	4.49	89.85	32m20s	19.22	1.18
	All	2h40m43s	35m33s	5	4.52	90.42	2h57m45s		
N2	FD	1h23m2s	19m17s	5	4.31	86.12	1h36m25s	14.93	1.13
	FM	6h32m44s	1h32m4s	5	4.27	85.32	7h40m20s	70.92	2.19
	BA	1h20m22s	52m5s	5	1.54	30.86	4h50m25s	14.45	1.05
	All	8h36m31s	2h41m28s	5	3.20	63.98	13h27m20s		

R, *Rilievo*; *N1*, *NotreDame1*; *N2*, *NotreDame2*; FD, feature detection; FM, feature matching; BA, bundle adjustment.

step i is $Q_i = \frac{1}{1+q_i(1/s_i-1)}$, where q_i is the proportion of the computation of the step i in the whole system, and s_i is the speedup ratio of the step i. The more the degree of contribution is, the larger the played acceleration role is.

As shown in Table 1, for feature detection step, the speedup ratio is close to the number of cores due to the small computation of serial component, which complies with Amdahl's law. The same is used for feature matching step. For *Rilievo* dataset in the bundle adjustment step, the speedup ratio is larger than the number of cores, and the execution efficiency is even equal to 276.27%. This is because our approach is not simply the parallelization of the original program, but replacing its part with other parallelized process, which does not meet the condition of Amdahl's law. The total

execution time of each dataset is less than the sum of phased execution times, since the separate implementation of each step requires some additional overhead such as process creation. It can be seen that the proportion of the feature matching step increases with the expansion of the scale of the problem. The high degree of parallelism in the feature matching step has an important role to improve the execution efficiency of the whole system. Figure 1 gives the 3-D point clouds of *Rilievo* and *NotreDame1* reconstructed by our method. The left-top corner of each subfigure shows one of the input images. It can be observed that our method maintains high quality of reconstruction when speeding up the algorithm.

Conclusion

In this paper, we present multi-core solutions to the problem of large-scale image-based modeling that run on currently available CPUs. These systems deliver a significant boost in speed over existing serial systems while maintaining the high quality of 3-D reconstruction. In the future, we would like to further improve the parallelism of various steps, especially for bundle adjustment, and at the same time ensure the high accuracy of 3-D reconstruction.

ACKNOWLEDGEMENTS

This work is supported by the NSF of China (61072062), Ph.D. Programs Foundation of Ministry of Education of China (20120032120040), the Tianjin Research Program of Application Foundation and Advanced Technology (12JCYBJC10300, 13JCQNJC03900), and the Research Academy of China Space Technology.

REFERENCES

Agarwal S, Snavely N, Simon I, Seitz S, Szeliski R (2009). Building Rome in a day. In: Proceeding of Int. Conf. Computer Vision, pp. 72–79.

Agarwal S, Snavely N, Seitz S, Szeliski R (2010). Bundle adjustment in the large. In: Proceeding of Eur. Conf. Computer Vision, pp. 29–42.

Arya S, Mount DM, Netanyahu NS, Silverman R, Wu AY (1998). An optimal algorithm for approximate nearest neighbor searching fixed dimensions. J. ACM. 45(6):891–923.

Lavoie P, Ionescu D, Petriu EM (2004). 3-D object model recovery from 2-D images using structured light. IEEE Trans. Instrumentation and Measurement. 53(2):437–443.

Liu Y, Stoll C, Gall G, Seidel HP, Theobalt C (2011). Markerless Motion Capture of Interacting Characters Using Multi-view Image Segmentation. In: Proceeding of IEEE Int'l Conf. Computer Vision and Pattern Recognition.

Nocedal J, Wright SJ (2006). Numerical Optimization. (2ed.). Springer.

Rander P, Narayanan PJ, Kanade T (1997). Virtualized reality: constructing time-varying virtual worlds from real world events. In: Proceeding of IEEE Conf. Visualization, Phoenix, Arizona, United States, pp. 277–284.

Snavely N, Seitz S, Szeliski R (2006). Photo tourism: exploring photo collections in 3D. ACM Transactions on Graphics (TOG). 25(3):835–846.

Wang J, Zhang C, Zhu W, Zhang Z, Xiong Z, Chou PA (2012). 3D scene reconstruction by multiple structured-light based commodity depth cameras: In: Proceeding of Int. Conf. Acoustics, Speech, and Signal Processing, Kyoto, Japan.

Wu C, Agarwal S, Curless B, Seitz S (2011). Multicore bundle adjustment. In: Proceeding of IEEE Conf. Computer Vision and Pattern Recognition, pp. 3057–3064.