Implementing Automated Photogrammetry for the New Zealand eScience Infrastructure (NeSI) Facilities

Steven Mills[1], David Eyers[1], Zhiyi Huang[1], Kai-Cheung Leung[1], and Xiaoxin Tang[2]

[1]Department of Computer Science
University of Otago. Dunedin, New Zealand
Phone: +64 3 479-5450 Fax: +64 3 479-8311
Email: `{steven,dme,hzy,kcleung}@otago.ac.nz`

[2]Shanghai Key Laboratory of Scalable Computing and Systems
Department of Computer Science
Shanghai Jiao Tong University, China
Email: `tang.xiaoxin@sjtu.edu.cn`

Presented at SIRC NZ 2013 – GIS and Remote Sensing Research Conference
University of Otago, Dunedin, New Zealand
August 29th – 30th 2013

***Keywords and phrases*:** Photogrammetry, Image-based reconstruction, High performance computing, New Zealand eScience Infrastructure

1.0 INTRODUCTION

In this paper we discuss current work to implement automated photogrammetry software for the New Zealand eScience Infastructure (NeSI) high performance computing (HPC) facilities. While the problem of reconstructing a 3D scene from a set of 2D images has been well studied, relatively little research has been published on how to exploit HPC infrastructure in order to process very large scenes. Recent results have shown that many thousands of images can be processed to create reconstructions of large areas (Agarwal, et al., 2009). This work makes use of a cluster of computers, but relatively little information is provided as to how the various tasks are distributed among the nodes in the cluster. Other research has focussed on a single computer containing multiple processors, using of both traditional CPUs and graphics processing units (GPUs) to perform this task (Fram, et al., 2010).

The techniques discussed here are suitable for both terrestrial and aerial photogrammetry. In aerial photogrammetry it is more common to have good priors for the camera positions (from GPS) and possibly orientation (from inertial sensors), and surveyed ground control points are often used to provide either absolute positioning or an independent verification/validation of the results. This additional information provides useful constraints on the solution, but we are primarily concerned with image-based reconstruction.

2.0 THE NEW ZEALAND ESCIENCE INFRASTRUCTURE

NeSI's aim is to provide "High Performance Computing (HPC) facilities to New Zealand, supporting researchers across the public research sector and private industry" (New Zealand eScience Infrastructure, 2013). The NeSI resources include the BlueFern supercomputer and several clusters located at the University of Canterbury, the University of Auckland, and the National Institute for Weather and Atmospheric Research (NIWA). Their main HPC facilities at the time of writing are summarised in Table 1**.**

Facility	Nodes	Cores/Node	Total Cores	Clock Speed	RAM/Node	Total RAM
BlueGene/P	-	-	8196	0.8 GHz	-	8192 GB
P575/POWER6	58	32	1856	4.7 GHz	64-128 GB	5376 GB
P755/POWER7	13	32	416	3.3 GHz	128 GB	1664 GB
Intel Cluster	76 194	12 16	4016	2.8 GHz 2.7 GHz	96 GB 128 GB	32 TB
Intel Big-memory nodes	3	10	30	2.4 GHz	512 GB	1.5 TB
Intel Visualisation Cluster	5	8	40	3.03 GHz	96 GB	480 GB

Table 1: Summary of NeSI high performance computing facilities – http://www.nesi.org.nz/facilities.

In addition to conventional CPUs, 21 of the Intel Cluster nodes and all 5 of the Visualisation Cluster nodes are equipped with Nvidia Tesla graphics processing units (GPUs). These are very efficient at performing a single instruction on multiple data streams (SIMD computing). While initially developed for graphics processing tasks, where many thousands of points or triangles are processed every frame, GPUs are increasingly being used for other computationally intensive tasks (Huang, et al., 2008), including those related to photogrammetry (Fram, et al., 2010). The Tesla M2090 cards in the general purpose Intel Cluster provide 448 processor cores at 1.15 GHz, and 16 of the nodes have two GPU cards each. The Visualisation Cluster machines each have two cards with 512 cores at 1.3 GHz. A further 5 nodes have two Nvidia Tesla K20 GPU cards, and each card provides 2496 cores.

3.0 COMPUTATION AND COMMUNICATION IN PHOTOGRAMMETRY

The computational process for taking two-dimensional images and producing three-dimensional model can be divided into three main phases, based on the degree of interaction between data from each image. This is illustrated in Figure 1, and shows an increase in the degree of interaction that must be modelled as processing is carried out.

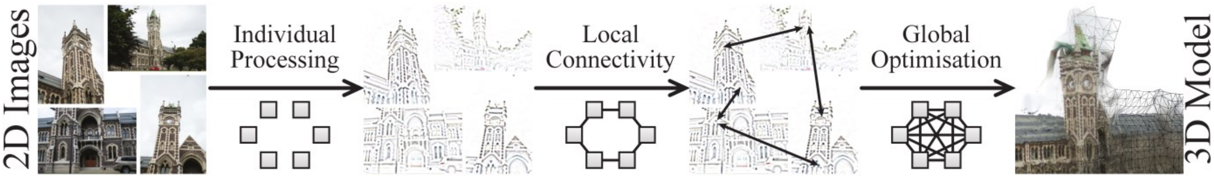

Figure 1: Main stages in an image-based reconstruction pipeline.

In the first stage images are processed individually. This typically includes correction for lens distortion; adjustment for photometric and lighting changes; and keypoint detection and description. The second stage considers local connections between images. The main task here is usually the computation of fundamental or essential matrices between image pairs, or of trifocal tensors between triplets of images. These computations are based on pairwise correspondences computed between overlapping images. The final stage is bundle adjustment to globally estimate the scene structure. While parameters associated with disparate parts of the scene may not directly influence each other, this process involves all of the data in a single computation.

3.1 Individual Image Processing

The initial stages of computation can be carried out independently on each image. The main computation at this stage is the extraction and description of keypoints for later matching (Lowe, 2004), but correction for lens distortion and lighting changes may also be required. Since the processing of each image is independent, and we typically have many more images than processors, this stage is ‘embarrassingly parallel’ – distributing the tasks across a number of processors is simple, and we do not need to consider communication between processes.

3.2 Local Connectivity

The next phase of computation involves only local connectivity. The most computationally demanding task at this stage is feature matching (Sipla-Anan & Hartley, 2008). These feature matches are then used to estimate the relative pose of the cameras. This may be done between each pair of cameras to compute fundamental or essential matrices, or using triplets of cameras to compute the trifocal tensor (Hartley & Zisserman, 2003). These tasks are still relatively independent, but the fact that overlapping pairs or triplets of images are used means that there is some communication between computational nodes. Our recent research (Mills, et al., 2013) has shown that dynamic scheduling algorithms can be used to distribute these tasks across a cluster of computers in such a way that these communication costs do not impact on performance. In some cases we have even observed super-linear speedups, where improved cache and memory use means that doubling the number of computers in the cluster more than halves the computation time required.

3.3 Global Optimisation

The final stage of processing is bundle adjustment – a large, sparse, non-linear least squares optimisation task (Triggs, et al., 1999). This problem is computationally very demanding, and recent research has investigated implementations on single multi-core machines (Agarwal, et al., 2010). The techniques identified in this research reduce the time and memory requirements, and could be well-suited to application on NeSI’s BlueGene/P HPC facility. Implementation on a cluster of compute nodes, where communication costs between nodes are much higher could prove more difficult, and remains an open area for research.

4.0 HETEROGENEOUS COMPUTING

As well as considering the patterns of computation and communication it is important to consider the range of processors that are available in modern computers. Mutli-core processors have largely replaced single core machines with quad-core processors being typical in consumer level computers. High performance workstations typically contain two or more individual processors, each with four to eight processing cores, and HPC facilties often provide clusters of such workstations. Alongside this, there is increasing use of GPUs for general purpose computing. No single solution will work well in all cases.

For example, while kd-trees are often used for efficient matching of features, they require many test-and-branch operations, which are not well suited to GPU implementation (Garcia, et al., 2008). Our research has supported this claim, and shown that while this task scales well across a cluster of computers, scaling on a single multi-core machine is more difficult (Mills, et al., 2013; Tang, et al., 2013).

5.0 CONCLUSIONS

Large-scale photogrammetric reconstruction from imagery is a computationally demanding task. As the number of images grows, the use of HPC facilities such as those provided by NeSI for this task becomes increasingly important. Our current research has identified three phases in the image-based reconstruction pipeline with increasingly complex communication required between sub-tasks. These communication patterns lead to a corresponding increase in complexity when implementing these algorithms for HPC facilities. In addition to the communication costs, attention must be paid to the heterogeneous nature of modern computing hardware, with multi-core processors, general purpose GPU computation, and clusters of compute nodes potentially requiring different approaches to each problem.

REFERENCES

New Zealand eScience Infrastructure (2013). *New Zealand eScience Infrastructure.* [Online] Available at: http://www.nesi.org.nz/ [Accessed 22 05 2013].

Agarwal, S., Snavely, N., Seitz, S. M. & Szeliski, R. (2010). Bundle Adjustment in the Large. *European Conference on Computer Vision*, pp. 29-42.

Agarwal, S. et al. (2009). Building Rome in a Day. *International Conference on Computer Vision*, pp. 72-79.

Fram, J.-M.et al. (2010). Building Rome on a Cloudless Day. *European Conference on Computer Vision (ECCV)*, pp. 368-381.

Garcia, V., Debreuve, E. & Barlaud, M. (2008). Fast k Nearest Neighbour Search using GPU. *Computer Vision and Pattern Recognition Workshops (CVPRW)*.

Hartley, R. & Zisserman, A. (2003). *Multiple View Geometry in Computer Vision, second edition.* Cambridge University Press.

Huang, Q., Huang, Z., Werstein, P. & Purvis, M. (2008). GPU as a General Purpose Computing Resource. *Int. Conf. Parallel and Distributed Computing, Applications and Techniques (PDCAT)*, pp. 151-158.

Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision,* 60(2), pp. 91-110.

Mills, S. et al. (2013). Large Scale Feature Matching with Distributed and Heterogeneous Computing. *Int. Conf. on Computer Vision*, p. under review.

Sipla-Anan, C. & Hartley, R. (2008). Optimised kd-Trees for Fast Image Descriptor Matching. *Computer Vision and Pattern Recognition (CVPR)*.

Tang, X. et al. (2013). Performance Tuning is Demanding on Multicore and Manycore Systems: A Case Study on Large Scale Feature Matching within Image Collections. *Int. Workshop on Parallel Programming Models and Systems Software*, p. under review.

Triggs, B., McLauchlan, P., Hartley, R. & Fitzgibbon, A. (1999). Bundle Adjustment - A Modern Synthesis. *Vision Algorithms: Theory and Practice*, pp. 298-372.