Parallel Tracking and Mapping for Large AR Workspace on a Cluster

SeongHu Hong, Jae Min Seo, Do Hyoung Kim, Chang-Sung Jeong
Dept. of Electrical Engineering, Korea University, Seoul, South Korea
{valvenis, onlytjwo, 2015010681, csjeong}@korea.ac.kr

Abstract

In this paper, we present a new fast Parallel Tracking and Mapping (PTAM) for Large AR Workspace on a cluster. PTAM limits map size, since large map size cause a lot of overhead. Therefore, our system is designed to overcome this limitation by separating map building part to cluster and tracking on local device. It can make possible that construct more large size map in similar operation time of original PTAM. Our system achieves 156% speed up than the PTAM[1].

Keywords: hardware acceleration; parallel computing; clustering;

1. Introduction

Recently trend of CPU is multi-core CPU which has decisive effect on computer vision algorithms. Using hardware parallelism will be significant part to getting scalability of computer vision algorithms. One way to exploit parallelism is to build our systems on low level libraries which have already been optimized to use hardware parallelism.

PTAM is one of the popular algorithms. Its response time is very fast and has conspicuous object detection accuracy. However, this algorithm has limitation depending on map size. A more practical limit at which the system remains well usable is around 6000 points and 150 key frames[2]. This limitation comes from bundle adjustment for refinement constructed map. The bundle adjustment has very high cost with large scale data. As a result, PTAM could be applied to small local area. So PTAM research team invited new algorithm for camera moving environment, especially for wearable device. The basic idea is to spread map into several small maps. It could help reduce bundle adjustment cost. It secured stability on moving situation [3]. It could provide mobility to AR application on wearable device. Nevertheless, application had no information between every map. If users want to get information about wide area, there is no way to provide it. The main reason of this situation is the high order time complexity of operation during bundle adjustments.

This paper proposes operating with remote cluster. We expected this architecture can expand working area for AR and breakthrough PTAM's limitation. Our system achieves 156% speed up than [1].

Our paper is organized as follows: In section 2, related works are introduced. In section 3, we describe our PTAM for large AR workspace on a cluster. In Section 4, we report the performance of our system. In section 5, conclusion and future work are showed.

2. Related Works

2.1 Cluster environment with hardware acceleration

The HPC (High Performance Computing) is the best way to solve complicated problems. For example, DNA sequencing problem, analyzing traffic flow in large city, volume rendering for medical pictures and so on. However, it causes great expense when operating HPC. Surely a lot of power should be used. Users could get some following benefits when configure cluster by using hardware acceleration. 1) Saving server operation cost 2) Conserving electricity 3) Reducing communication time. We could easily find precedents. A range of previous deployment with hardware acceleration added server-especially user GPGPU was very wide. From general purposed cluster [4][5][6] to specific application [7][8][9] there are a lot of works. Sometimes it configures with normal server [9]. At this cluster or cloud environment, users could write code with parallel platform such as CUDA, OpenCL and MPI. We present using MPI with Cluster.

2.2 PTAM

PTAM (Parallel Tracking and Mapping) is Visual SLAM (Simultaneous Localization and Mapping) Technology using key frame. The frame which is saved whole images when compact video is key frame. A key frame is a drawing the starting and ending points of any smooth transition.

In robotic mapping, SLAM is the computational problem of constructing or updating a map of and unknown environment while simultaneously keeping track of agent's location within it. SLAM can be using a lot of sensors. Which using camera image is Visual SLAM and especially using single camera is mono SLAM[10].

Existing Visual SLAMs are doing feature point matching and map construction via all of the frames. So processing speed is very low and this cause that SLAM cannot use precise algorithms. But PTAM separate tracking part which trace position of camera and mapping part which construct feature map and run two parts are separately using thread. Each part works on its own thread. It removes dependency between two parts and get accuracy and real-time.

PTAM is appropriate method for bounded and not deformable space. But our system proposed mapping part running of cluster side. The benefit of this change will be described in next section.

3. PTAM for Large AR Workspace on a Cluster

Originally, PTAM works on one standalone machine. It has limitation as we mentioned before. So we separate original PTAM to mobile side and cluster side for overcome the limitation. Cluster side construct map. During map construction, bundle adjustment occupies the largest portion in processing. We focus on speed up of bundle adjustment operation and extending size of map through cluster environment. And our system communicates between mobile side and cluster side via network. So the important thing is that performance improvement have to exceed network overhead.

Mobile side working flow is exactly same as legacy since the tracking module's performance has no problem. In legacy method reference map data works directly and surely on same machine. We add map repository instead of direct map data accessing. It reduces cost for modifying PTAM source code and can easily update source code later on mobile side. Map repository is managed by agent. It manages map data in repository and control communication on cluster. The key frame data that comes from camera does not need to process on mobile side. It just passes through cluster side. If cluster side cannot handle key frame data, it discards them. There is one more benefit of map repository. PTAM only executes bundle adjustment when it is need. In normal situation, tracking thread only needs to key frame information. Proposed architecture saves key frame and map information mobile side. It makes there is no communication between mobile device and cluster. So PTAM works without communication with cluster if there is no need for bundle adjustment.

On cluster side, there is a main controller which is named stream processing controller. The rules are to distribute processing resource (server node resource), create processing element and invoke parallel processing element if needed. On cluster side, working process is very similar with map construction part in PTAM. The changes are these. 1) Each module or processing is mapped with every single processing element. 2) Optimizing map step can run with parallel processing elements. Using how many parallel processing elements depends on processing data volume and it will be deter-mined by optimizing processing element. After operating map construction, map data is sent back to mobile side. A repository agent receives map data, and save it to repository. It does not need to send notice to tracking module, since tracking module and map construction module working threads are separated from the beginning. A tracking module just checks and

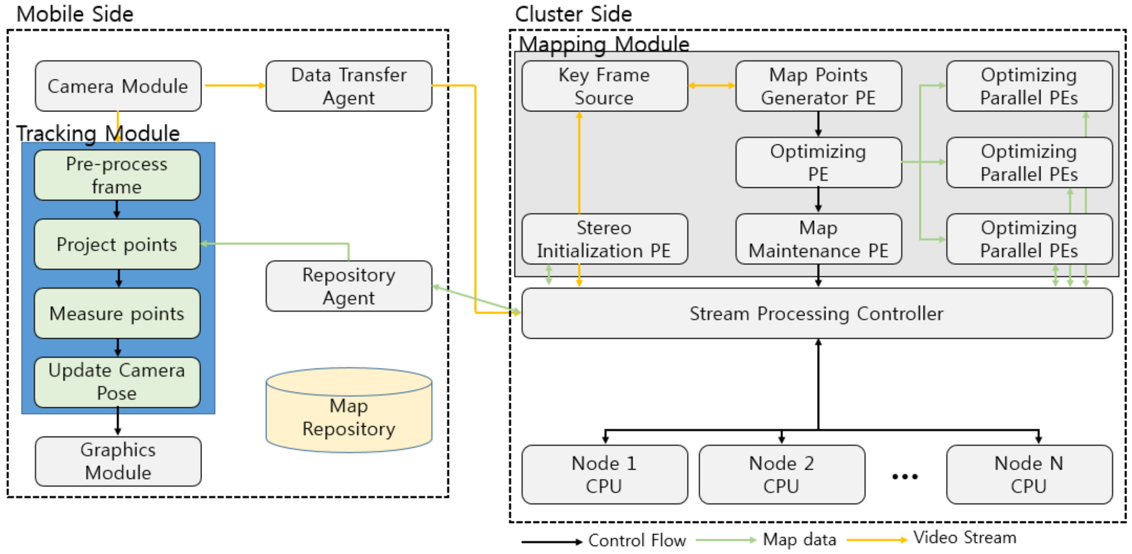

Fig. 1. PTAM for Large AR Workspace on a Cluster

dispatches map data when needed. Each modules and sequence diagram are described in figure 1 and figure 2.

The cluster side can serve several device or computer. If geological data is considered, cluster could build map with different data stream from several input devices. A map construction works on cluster side, because of this reason we can expect more rich user experience at augmented reality. For example, 1) As we mentioned collaboration augmented reality possible 2) Mobile device can be mapped with already constructed map. PTAM considering volatility map data. If map construction is working on cluster side, we can reuse map.

4. Performance Evaluation

To test our system, we experimented in our laboratory. Visual textures were placed on the walls and textured 3D objects were on the desk. Cam was mounted at the corner of the room. To create a global reconstruction of the scene, we captured video sequence with cam.

Performance evaluation environment is: Intel Xeon CPU (2.13GHz) x 8, 24GB Memory. Cluster with that spec was used for the mapping module. While the tracking system scales fairly well with increasing map size. The largest map we have produced is map of our laboratory containing about 2,500 map points.

As the map grows, bundle adjustment can keep up with exploration. The original PTAM need 0.969 seconds to construct map in average. Compared with original PTAM, the processing time of our system which is required for constructing map is decreased 64% (0.621 seconds in average). And there is a lot of network overhead between tracking module and mapping module but the total operation time is nearly same as original PTAM. We can construct large map in same time as original PTAM.

5. Conclusion and Future Works

We presented our new PTAM system for large AR workspace. The original PTAM has limitation when the cost of bundle adjustment operation for map construction is extremely high. In order to overcome that limitation, we separate tracking module and mapping module to mobile side and cluster side. Mapping module can construct large size map rapidly since it works on cluster which has outstanding performance than the mobile device. Nevertheless, there is a lot of network communication between mobile side and cluster side, we can get performance improvement because using cluster cause increasing of computing power. So we can achieve two times fast performance at map construction by cluster.

Later, we will experiment our system at multi cluster environment with MPI. And we can apply another hardware accelerator such as Intel Xeon Phi or GPGPU instead of cluster. Especially we interested in Intel Xeon Phi. Intel Xeon Phi is for parallel operation coprocessor. They can operate independently of CPUs and they don't require special code to program. We will optimize our system on Intel Xeon Phi cluster soon.

References

[1] G. Klein and D. W. Murray, "Parallel Tracking and Mapping for Small AR Workspaces," IEEE ACM Int. Symp. Mix. Augment. Real., pp. 1–10, 2007.

[2] R. Castle, G. Klein, and D. W. Murray, "Video-rate localization in multiple maps for wearable augmented reality," in 2008 12th IEEE International Symposium on Wearable Computers, 2008, pp. 15–22.

[3] V. V Kindratenko, J. J. Enos, G. Shi, M. T. Showerman, G. W. Arnold, J. E. Stone, J. C. Phillips, and W.-M. Hwu, "GPU clusters for high-performance

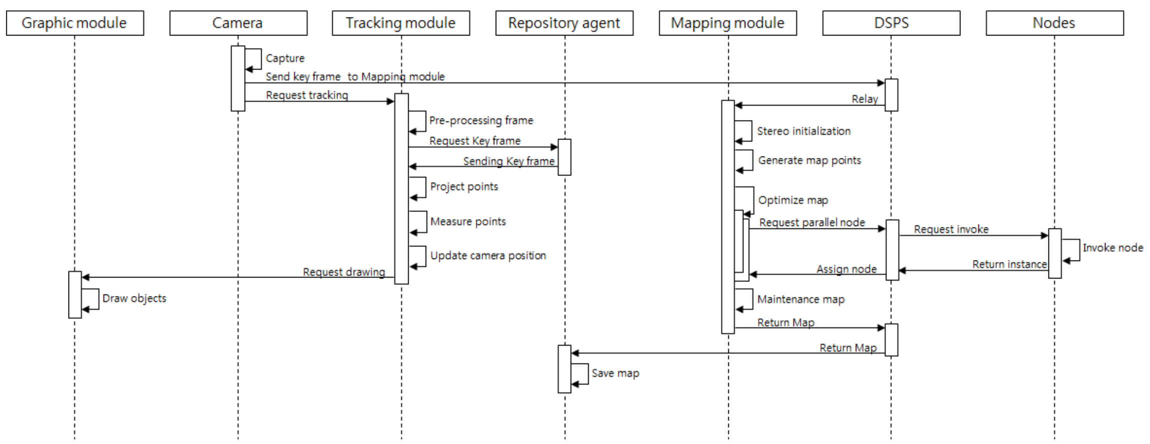

Fig. 2. Sequence diagram of stream processing system for PTAM

computing," in Cluster Computing and Workshops, 2009. CLUSTER '09. IEEE International Conference on, 2009, pp. 1–8.

[4] L. Hu, X. Che, and Z. Xie, "GPGPU cloud: A paradigm for general purpose computing," Tsinghua Sci. Technol., vol. 18, no. 1, pp. 22–23, 2013.

[5] R. R. Expósito, G. L. Taboada, S. Ramos, J. Touriño, and R. Doallo, "General-purpose computation on GPUs for high performance cloud computing," Concurr. Comput. Pract. Exp., vol. 25, no. 12, pp. 1628–1642, 2013.

[6] S. Georgescu and P. Chow, "GPU accelerated CAE using open solvers and the cloud," ACM SIGARCH Comput. Archit. News, vol. 39, no. 4, p. 14, 2011.

[7] K. Wang and Z. Shen, "Artificial Societies and GPU-Based Cloud Computing for Intelligent Transportation Management," IEEE Intell. Syst., vol. 26, no. 4, pp. 22–28, 2011.

[7] J. Yimu, K. Zizhuo, P. Q. Yu, S. Yanpeng, K. Jiangbang, and H. Wei, "A Cloud Computing Service Architecture of a Parallel Algorithm Oriented to Scientific Computing with CUDA and Monte Carlo," Cybern. Inf. Technol., vol. 13, no. Special Issue, pp. 153–166, 2013.