Robotics and Autonomous Systems 62 (2014) 401–413

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

C^{2}TAM: A Cloud framework for cooperative tracking and mapping

L. Riazuelo *, Javier Civera, J.M.M. Montiel

Robotics, Perception and Real-Time Group, Aragón Institute of Engineering Research (I3A), Universidad de Zaragoza, 50018 Zaragoza, Spain

HIGHLIGHTS

- Allocate the expensive map optimization process out of the robot platform.
- Interoperability between different visual sensors.
- The raw keyframe images are stored in the Cloud along with the point-based map.
- The proposed framework naturally adapts to the cooperative SLAM problem.

ARTICLE INFO

Article history:
Received 19 December 2012
Received in revised form
12 November 2013
Accepted 22 November 2013
Available online 4 December 2013

Keywords:
SLAM
Visual SLAM
Cloud SLAM
Cloud Robotics
Cloud Computing

ABSTRACT

The Simultaneous Localization And Mapping by an autonomous mobile robot – known by its acronym SLAM – is a computationally demanding process for medium and large-scale scenarios, in spite of the progress both in the algorithmic and hardware sides. As a consequence, a robot with SLAM capabilities has to be equipped with the latest computers whose weight and power consumption might limit its autonomy.

This paper describes a visual SLAM system based on a distributed framework where the expensive map optimization and storage is allocated as a service in the Cloud, while a light camera tracking client runs on a local computer. The robot onboard computers are freed from most of the computation, the only extra requirement being an internet connection. The data flow from and to the Cloud is low enough to be supported by a standard wireless connection.

The experimental section is focused on showing real-time performance for single-robot and cooperative SLAM using an RGBD camera. The system provides the interface to a map database where: (1) a map can be built and stored, (2) stored maps can be reused by other robots, (3) a robot can fuse its map online with a map already in the database, and (4) several robots can estimate individual maps and fuse them together if an overlap is detected.

1. Introduction

The acronym *SLAM*, standing for Simultaneous Localization and Mapping, refers to the problem of simultaneously estimating a model of the surroundings of a mobile robot – the "map" – and the robot's location into it from a stream of sensor data [1]. SLAM is a problem of key importance in robotics; as an accurate model of the environment is a prerequisite of most of the mobile robots' tasks (e.g., navigation, exploration or manipulation). In a practical robotic setting, the computation and memory requirements of the SLAM algorithms are two aspects of prime interest: SLAM algorithms tend to be computationally demanding and the onboard resources of a mobile robot are limited. Also, SLAM has strong real-time constraints as it is integrated in the control loop of the robot.

In recent years the possibility of massive storage and computation in Internet servers – known as *Cloud Computing* and *Cloud Storage* – has become a reality. The availability of such technology and its possible use in robotics have opened the door to a whole new line of research called *Cloud Robotics* [2]. Regarding SLAM, robots could benefit from the use of the Cloud by moving part of the SLAM estimation from their limited computers to external servers; saving computational and power resources. This paper tries to answer the following question *How should a SLAM system be partitioned in order to leverage the storage and computational resources in the Cloud?* Notice that the answer to this question is not trivial. Due to the real-time constraints of SLAM algorithms and the network delays the naïve solution of moving *all* the computation to the Cloud would be unfeasible. In order to guarantee the real-time, part of the computation must be performed on the robot's computers.

The contribution of this paper is the partition of a real-time SLAM algorithm that allows part of the computation to be moved to the Cloud without loss of performance. Our experimental results show that the bandwidth required in all cases does not exceed

---

* Corresponding author. Tel.: +34 876554559.
E-mail address: riazuelo@unizar.es (L. Riazuelo).

0921-8890/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.robot.2013.11.007

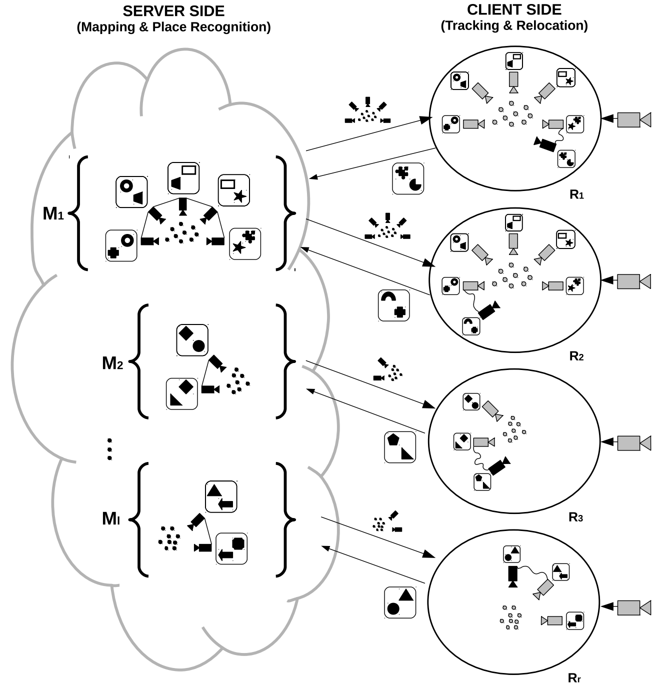

Fig. 1. Computer intensive bundle adjustment is performed as a Cloud service running on a high performance server. Camera location with respect to the map is computed in low performance mobile devices. Several tracking threads can be run on the same map data. The data flow from tracking to mapping is composed of the new keyframes when gathered images contain new information with respect to the available map; and from mapping to tracking the flow is the computed map.

a standard wireless connection. We demonstrate the capabilities of the framework to provide the interface to a map database in a multi-map multi-camera experiment where the users can: create and save several maps, relocate within them and improve them as new areas are explored, and fuse several maps into one if an overlap is detected. As part of the paper, we plan to release the software after acceptance.

We take as a starting point the monocular SLAM algorithm described in [3], the so-called *Parallel Tracking and Mapping* or PTAM. The processing of PTAM is based on two parallel threads. On one hand, a geometric map is computed by non-linear optimization over a set of selected keyframes usually known as Bundle Adjustment. This background process is able to produce an accurate 3D map at a low frame rate. On the other hand, a foreground tracking process is able to estimate the camera location at the frame rate assuming a known map. This method is able to produce maps composed of thousands of points using standard computers for room-sized environments. From this SLAM system, we propose C^{2}TAM standing for *Cloud framework for Cooperative Tracking And Mapping* that moves the non-linear map optimization thread to a service operating in the Cloud.

On top of the computational advances of the keyframe based methods, the resulting communication between the tracking and mapping processes requires low bandwidth. The tracking process sends a new raw keyframe only when the gathered image contains new information with respect to the available map. The mapping sends a new map after every iteration of the Bundle Adjustment, at a frequency substantially lower than the frame rate. Even in exploratory trajectories, the number of new keyframes is small compared to the frame rate; and in the case of already visited areas no new keyframes are sent. See Fig. 1 for a scheme of the framework. The low communication bandwidth allows us to use a standard wireless connection and run tracking and mapping on different computers. Also, in both data flows the algorithm is robust to latencies. The camera tracking thread can run on suboptimal maps until the next global optimization is finished and sent. Also, with an appropriate policy on map management, the camera tracking is robust to delays in keyframe addition.

We believe that a SLAM system partially running on the Cloud has a wide array of benefits and potential applications:

- allocating the expensive map optimization process out of the robot platform allows a significant reduction in the onboard computational budget, hence reducing the payload and power consumption; both critical factors for field robotics (e.g., unmanned aerial [4] or underwater vehicles). More importantly, it provides the foundations to accommodate SLAM algorithms within the distributed computation framework, which makes it possible to exploit the newly available Cloud computation resources;
- the interoperability between different visual sensors comes as a prerequisite in our system, as very different robots with different cameras could connect to the mapping service;
- the raw keyframe images are stored in the Cloud along with the point-based map. Optimizing a sparse 3D scene of salient points is just one of the Cloud services that can be run over the keyframes. Additionally, other background processes at different time scales can handle map management operations aiming at life-long mapping [5], semantic mapping [6,7], layout estimation [8] or computing free space for navigation [9];
- the centralized map building also allows a straightforward massive data storage of robotic sequences and geometric estimations that could be used to provide a significant training sample for learning. It can be seen that the size of the computer vision datasets [10] tends to be much larger than those of the robotics [11]. The creation of datasets with data exclusively from robots is essential for exploiting the commonalities in the robotic data [12];
- the proposed framework naturally adapts to the cooperative SLAM problem [13,14]; where several robots have to build a joint map of the environment. The server can operate on different maps and fuse them independently of the trackers running on the robots. As the number of clients grows the server computation can be parallelized. The bandwidth required for each tracker is low enough to be provided by a standard wireless connection;
- the technologies involved in the proposed system are in an advanced state of maturity: cloud Computing and storage have already been successfully incorporated in multiple domains and the keyframe-based SLAM is one the most promising mapping methods available [15]. Additionally, the proposed framework can provide the interface to an advanced database of visual maps in the Cloud.

The rest of the paper is organized as follows: Section 2 refers to the related work; Section 3 discusses the main SLAM components, Section 4 provides the formulation of the SLAM problem and Section 5 gives the details of the proposed system, C^{2}TAM. Finally, Section 6 shows the experimental results and Section 7 sets out the conclusions and presents lines for future work.

2. Related work

In recent years, the Cloud Computing paradigm has revolutionized almost every field related to computer science [16]. The idea in a few words is that the software pieces are replaced by services provided via the Internet. In the robotics community the potential applications, the benefits and the main lines for research regarding Cloud Computing have already been foreseen [17,18] and some platforms are already starting to emerge [19]. Nevertheless, with some recent exceptions referred to in the next paragraph, there is still a lack of specific algorithms and specific realizations of these ideas. This paper aims to contribute with a concrete realization of