An Experimental Distributed Framework for Distributed Simultaneous Localization and Mapping

Ruwan Gamage, Mihran Tuceryan
Department of Computer and Information Science
Indiana University - Purdue University - Indianapolis
Indianapolis, Indiana 46202
Email: rjegodag@iupui.edu, tuceryan@iupui.edu

***Abstract*—Simultaneous Localization and Mapping (SLAM) is widely used in applications such as rescue, navigation, semantic mapping, augmented reality and home entertainment applications. Most of these applications would do better if multiple devices are used in a distributed setting. The distributed SLAM research would benefit if there is a framework where the complexities of network communication is already handled. In this paper we introduce such framework utilizing open source Robot Operating System (ROS) and VirtualBox virtualization software. Furthermore, we describe a way to measure communication statistics of the distributed SLAM system.**

***Keywords*—Simultaneous Localization and Mapping, SLAM, Distributed SLAM, ROS**

I. Introduction

The robotics community defines Simultaneous Localization and Mapping(SLAM) problem as an agent creating a map of an unknown environment using sensors, while localizing itself in it. To localize the agent properly, an accurate map is required. To generate an accurate map, localization has to be done properly. Which means these two tasks have to be done simultaneously to benefit each other.

Many researchers investigated on how to use multiple agents to perform SLAM: called collaborative or distributed SLAM. Distributed SLAM increases the robustness of SLAM process and makes it less vulnerable for catastrophic failures. The main challenge in distributed SLAM is to share information between agents with limited communication bandwidth.

In this paper we introduce an experimental distributed framework for distributed SLAM. We utilize many aspects of already available free and open source Robot Operating System (ROS) [1][2, About ROS]. The maturity of the ROS communication system is a major factor for selecting ROS as our underlying communication platform. Furthermore, a large number of robotics and related libraries are already supported in ROS, meaning our framework would attract many researchers.

During the experimentation stage, we implement distributed system in virtual machines. In our work we used VirtualBox virtualisation software [3], given it suited most of the network requirements to simulate a distributed framework. To test our framework we used the LSD-SLAM implementation by Engel et al. [4].

This paper is organized as follows. First, in section II we discuss related work. Next, in section III we introduce how to adopt our framework for a distributed SLAM environment. Finally in section IV we provide conclusion and what we would do with this framework in the future.

II. Related Work

An Extended Kalman Filter based SLAM Solution (EKF-SLAM) was first introduced in a seminal paper by Smith et al. [5]. Since then, many contributed using algorithms based on Monte Carlo Sampling (FastSLAM) [6] & Unscented Kalman Filter based approach (UKF-SLAM)[7], etc.

Davison et al. [8] introduced a VisualSLAM method of capturing the path of a freely moving camera (6 Degrees of Freedom — DoF), while generating a sparse map. The method was called Monocular Visual SLAM (MonoSLAM). A more robust MonoSLAM method called Parallel Tracking and Mapping (PTAM) was introduced by Klein et al. in [9]. Recent MonoSLAM contributions such as DTAM by Newcombe et al. in [10] and LSD-SLAM by Engel et al. [4], utilize image pixel intensities directly, and generate dense or semi-dense maps of the environment.

To handle a distributed SLAM environment consisting of multiple agents, a naive brute-force method is to communicate all sensor observations and map updates between each and every agent. However, the bandwidth and computational resources available for an agent are typically limited. Furthermore, the distributed network is subject to failures of nodes and links. So it is required to come up with an intelligent approach to cope with these challenges. Cunningham et al. in [11][12] formulates the distributed SLAM problem using a graphical model. In their fully decentralized system, each agent maintains a consistent local map augmented with information shared in a neighborhood of agents. In the system proposed in [13], local feature matches are propagated through the low-bandwidth communication network. This method helps agents to find global correspondences with other agents with no direct connections.Work done in [14] discusses about most informative features to transmit, to reduce bandwidth requirements.

The distributed tracking system (DTS) by Joshi et al. [15] and its enhancement by Rybarczyk et al. [16] [17] describe an efficient distributed tracking infrastructure for indoor usages. Their design is capable of communicating and fuse tracking results from multiple cameras within the system. They utilize a discovery service to locate camera services.

978-1-4673-9985-2/16/$31.00 ©2016 IEEE

III. METHODOLOGY

Our experimental distributed framework consists of multiple nodes and a communication network. Robot Operating System[2] is used as the underlying platform for nodes and communication network of the distributed framework. For virtualization we used VirtualBox[3] virtualisation software. Furthermore, an existing SLAM implementation for ROS, LSD-SLAM [4], is used to test our framework.

A. Robot Operating System

A node in ROS is responsible of performing computations. ROS also provides a message passing communication framework between nodes. A single project could consist of multiple ROS nodes. For example, there could be a node to acquire sensor data, a second node to process it, and another to visualize the results.

In its communication framework, ROS provides named communication buses called **topics**. Multiple nodes can publish messages to a topic while multiple subscribed nodes could receive them. Based on the requirement, ROS could either use UDP or TCP for message passing. In addition to that ROS allows remote procedure calls (RPC) between nodes. These are called ROS **services**. During a service call, a node sends a request message and waits for a response.

ROS consists a master server to list all available topics of a system. After identifying providers and subscribers of a topic by communicating with master server, ROS nodes can communicate with each other via topics.

B. LSD-SLAM based monocular visual SLAM system

To test our framework we have used a modified version of semi-dense SLAM implementation for ROS, called LSD-SLAM, found in the work by Engel et al. [4][18].

LSD-SLAM implementation contain two nodes. the first lsd_slam_core node receives camera frames via ROS topic $/image$. The lsd_slam_core node generate keyframes and a pose graph after performing its SLAM operation. Results are made available via ROS topics $/lsd\_slam/graph$, $/lsd\_slam/keyframes$ and $/lsd\_slam/liveframes$. The second node lsd_slam_viewer node displays resultant point cloud by listening to these topics.

C. The framework: Network configuration

The distributed framework needs to support free flow of information between nodes. For that, we configured virtual machines with the options provided by VirtualBox virtualisation software.

As shown in Figure1, each virtual machines are configured to contain 2 network adapters.

- First network adapter operates in host-only mode. This host-only network allows the host computer to communicate with each virtual machine. Furthermore, it allows virtual machines to communicate with each other. A static IP address is provided to each network interface.

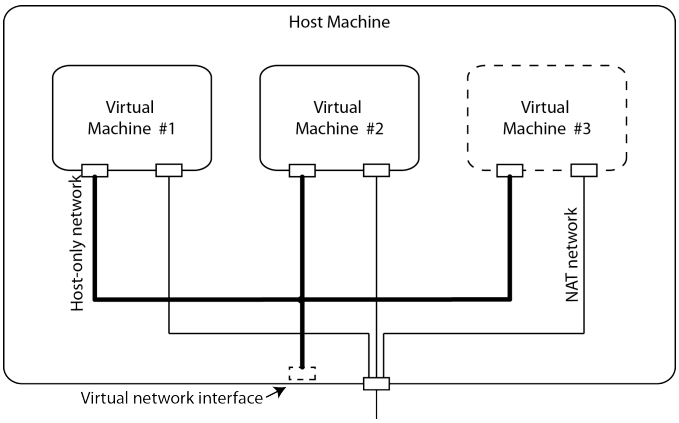

Fig. 1: Communication network using 2 types of network interfaces. Virtual machines use host-only network to communicate with each other and NAT network to access outside.

- The second network adapter operate in NAT mode. This NAT network allows virtual machines to access the network beyond the host machine.

D. The framework: SLAM Nodes

Each node is deployed in a VirtualBox virtual machine having Ubuntu as the operating system. The lsd_slam_core node is deployed in each virtual machine. Each machine is given a unique host name as well.

ROS master is deployed in the host machine where all virtual machines are deployed. To use remote ROS master, each virtual machine has to be configured as shown in listing 1. In this example, the host machine has a virtual network interface configured with IP address 10.1.2.2

Next lsd_slam_viewer node is deployed in the host machine and successfully received topic data from each lsd_slam_core node deployed in different virtual machines. Furthermore, lsd_slam_viewer is deployed in a virtual machine, and successfully received topic data from a lsd_slam_core node deployed in a virtual machine.

E. Multiple instances of the same ROS node

For a distributed SLAM system containing multiple instances of a ROS node, we need to find a way to uniquely identify each node's topics. For example, LSD-SLAM core publishes messages to $/lsd\_slam/graph$ topic. If we have two instances, namely A and B, we would like to see these topics as $/A/lsd\_slam/graph$ and $/B/lsd\_slam/graph$. In this way, node A could listen to topic $/B/lsd\_slam/graph$ and node B could listen to topic $/A/lsd\_slam/graph$.

Fortunately, ROS provides a way to remap a topic to a different name. We use $roslaunch$, a tool designed launch ROS nodes with a set of given parameters. $roslaunch$ can read environment variables, hence we set following environment variables before calling the $roslaunch$ tool.

Listing 1: Setting up environment variables

```
export ROS_HOSTNAME=c3po
export ROS_IP=10.1.2.3
export ROS_MASTER_URI=http://10.1.2.2:11311
export LSD_SLAM_NODE=c3po
roslaunch lsd_slam.launch
```

We configure the lsd_slam.launch file for topic remapping as shown in listing 2.

Listing 2: ROS node launch script

```
<launch>
  <node pkg="lsd_slam_core"
  type="live_slam"
  name="slam_$(env LSD_SLAM_NODE)"
  args="image:=/image_raw
  camera_info:=/camera_info"
  output="screen">
    <remap from="/lsd_slam/graph"
     to="/$(env LSD_SLAM_NODE)/graph" />
    <remap from="/lsd_slam/keyframes"
     to="/$(env LSD_SLAM_NODE)/keyframes" />
  </node>
</launch>
```

Above launch file remap topics $/lsd\_slam/graph$ & $/lsd\_slam/keyframes/$ to $/c3po/graph$ & $/c3po/keyframes$ respectively. Now the node $r2d2$ can listen to the topic $/c3po/graph$ and $c3po$ can listen to the topic $/r2d2/graph$.

F. The framework: Network Statistics

Distributed SLAM could easily reach the bandwidth limit of the network, specially, if nodes transfer map data between each other for fusion. We could generate statistics of our system's bandwidth utilization and accordingly do necessary changes. Given each node's output topics are now uniquely defined with its name as a prefix, we use ROS **Topic Statistics** to generate statistics per node. In addition to that, topic statistics could measure, number of dropped messages, mean & standard deviation of the age of messages, and period of messages by all providers.

IV. Conclusion & Future Work

In conclusion we have developed an experimental framework for distributed SLAM. As a case study, we have used the LSD-SLAM implementation to test our framework. We have tested different subscriber and provider configurations and found our framework support data communication between nodes with no issues. Similarly, one can adapt our proposed distributed framework for different distributed SLAM implementations.

In future, we are going to use this framework to perform Distributed SLAM, where multiple agents generate a single fused dense map of the environment.

References

[1] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, 2009.

[2] Robot operating system. [Online]. Available: http://www.ros.org/about-ros/

[3] Virtualbox: An open source, general-purpose full virtualizer for x86 hardware. [Online]. Available: https://www.virtualbox.org/wiki/Downloads

[4] J. Engel, T. Schps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *Computer Vision ECCV 2014*, ser. Lecture Notes in Computer Science. Springer International Publishing, 2014, vol. 8690, pp. 834–849. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-10605-2_54

[5] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," in *Autonomous Robot Vehicles*, I. Cox and G. Wilfong, Eds. Springer New York, 1990, pp. 167–193. [Online]. Available: http://dx.doi.org/10.1007/978-1-4613-8997-2_14

[6] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "Fastslam: A factored solution to the simultaneous localization and mapping problem," in *In Proceedings of the AAAI National Conference on Artificial Intelligence*. AAAI, 2002, pp. 593–598.

[7] R. Martinez-Cantin and J. Castellanos, "Unscented slam for large-scale outdoor environments," in *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, Aug 2005, pp. 3427–3432.

[8] A. Davison, I. Reid, N. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 6, pp. 1052–1067, June 2007.

[9] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, Nov 2007, pp. 225–234.

[10] R. A. Newcombe, S. Lovegrove, and A. Davison, "Dtam: Dense tracking and mapping in real-time," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, Nov 2011, pp. 2320–2327.

[11] A. Cunningham, M. Paluri, and F. Dellaert, "Ddf-sam: Fully distributed slam using constrained factor graphs," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 3025–3030.

[12] A. Cunningham, V. Indelman, and F. Dellaert, "Ddf-sam 2.0: Consistent distributed smoothing and mapping," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 5220–5227.

[13] E. Montijano, R. Aragues, and C. Sagues, "Distributed data association in robotic networks with cameras and limited communications," *Robotics, IEEE Transactions on*, vol. 29, no. 6, pp. 1408–1423, 2013.

[14] E. Nettleton, S. Thrun, H. Durrant-Whyte, and S. Sukkarieh, "Decentralised slam with low-bandwidth communication for teams of vehicles," in *Field and Service Robotics*. Springer, 2006, pp. 179–188.

[15] G. G. Joshi, R. R. Raje, and M. Tuceryan, "Designing and experimenting with a distributed tracking system," in *Parallel and Distributed Systems, 2008. ICPADS '08. 14th IEEE International Conference on*, Dec 2008, pp. 64–71.

[16] R. Rybarczyk, R. Raje, and M. Tuceryan, "edots 2.0: A pervasive indoor tracking system."

[17] R. T. Rybarczyk, "E-dts 2.0: A next-generation of a distributed tracking system," Master's thesis, Purdue University, 7 2010.

[18] J. Engel, J. Sturm, and D. Cremers, "Semi-dense visual odometry for a monocular camera," in *Computer Vision (ICCV), 2013 IEEE International Conference on*, Dec 2013, pp. 1449–1456.