Visual Odometry

Christos Kokas

May 2022

Contents

1 Introduction

In robotics and computer vision, visual odometry is the process of determining the position and orientation of a robot by analyzing images from the robot's camera. It has been used in a wide variety of robotic applications, such as on the Mars Exploration Rovers [1].

The accurate localization of the robot poses a great challenge for many applications, even more in GPS-Denied environments (e.g. indoors) where GPS systems, such as RTK GPS, have difficulty producing an accurate estimate of the position of the robot. Visual Odometry has gained popularity because it can produce accurate estimates of the position of the camera even in GPS-Denied environments and at the same time be cheaper than other alternatives (e.g. RTK GPS, LiDARS). Combinations of Visual Odometry and other techniques or products, such as Visual Odometry combined with an IMU (also know as Visual Inertial Odometry) or Visual Odometry combined with Deep Learning, can also yield satisfying results.

2 Definitions/Appendix

Bundle Adjustment is simultaneous refining of the 3D coordinates describing the scene geometry, the parameters of the relative motion, and the optical characteristics of the camera(s) employed to acquire the images, given a set of images depicting a number of 3D points from different viewpoints. It amounts to an optimization problem on the 3D structure and viewing parameters (i.e., camera pose and possibly intrinsic calibration and radial distortion), to obtain a reconstruction which is optimal under certain assumptions regarding the noise pertaining to the observed image features.

Reprojection Error is a geometric error corresponding to the image distance between a projected point and a measured one.

Pose Graph contains nodes connected by edges that represents the pose of the robot.

Bayesian inference is a method of statistical inference in which Bayes' theorem is used to update the probability for a hypothesis as more evidence or information becomes available.

Voxel each of an array of elements of volume that constitute a notional three-dimensional space, especially each of an array of discrete elements into which a representation of a three-dimensional object is divided.

Euclidean clustering groups points that are close together. You must set a "closeness" threshold, such that points within this threshold are considered to be part of the same cluster.

3 Literature Review

In this section papers using Visual Odometry will be presented each using a different method of producing an accurate estimate of the camera.

3.1 ORB_SLAM2

3.1.1 Abstract

We present ORB-SLAM2 a complete SLAM system for monocular, stereo and RGB-D cameras, including map reuse, loop closing and relocalization capabilities. The system works in real-time on standard CPUs in a wide variety of environments from small hand-held indoors sequences, to drones flying in industrial environments and cars driving around a city. Our back-end based on bundle adjustment with monocular and stereo observations allows for accurate trajectory estimation with metric scale. Our system includes a lightweight localization mode that leverages visual odometry tracks for unmapped regions and matches to map points that allow for zero-drift localization. The evaluation on 29 popular public sequences shows that our method achieves state-of-the-art accuracy, being in most cases the most accurate SLAM solution. We publish the source code, not only for the benefit of the SLAM community, but with the aim of being an out-of-the-box SLAM solution for researchers in other fields [2].

3.1.2 System Overview

ORB-SLAM2 for stereo and RGB-D camera has three main parallel threads.

- Find Feature Matches to the local map with every frame to localize the camera, and apply motion-only Bundle Adjustment (BA) to minimize the reprojection error. (Tracking)
- Perform local BA to manage and optimize the local map. A local map is a simplified representation of the immediate environment around the robot. (Local Mapping)

- Detect large loops and correct the accumulated drift by performing a pose-graph optimization. (Loop Closure)

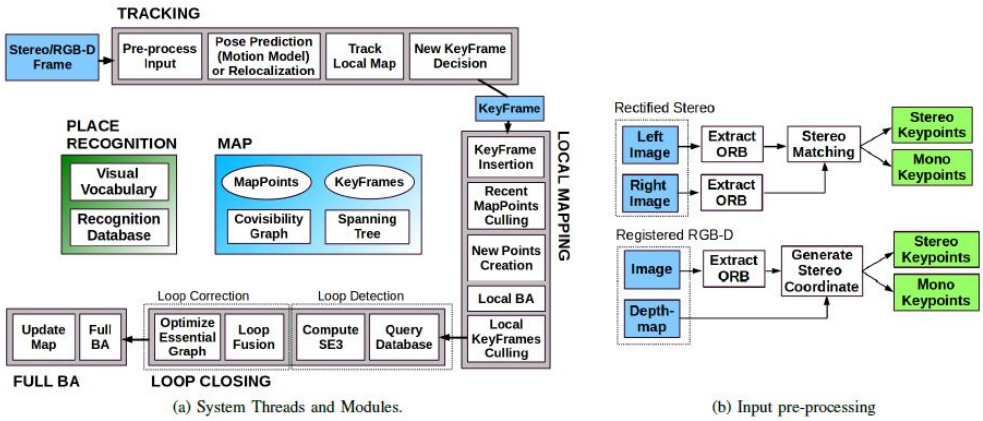

Figure 3.1: ORB-SLAM2 Diagram of Parallel Threads

3.1.3 Key Points

ORB-SLAM2's key points include :

- ORB-SLAM2 treats differently close and far points, first described in the work of Paz et al. [3], which indicates that the points' depth cannot be reliably estimated due to little disparity in the stereo camera. If the points' depth is more than ~ 40 times the stereo baseline it cannot be reliably estimated.

3.2 Hydra: A Real-time Spatial Perception Engine for 3D Scene Graph Construction and Optimization

3.2.1 Abstract

3D scene graphs have recently emerged as a powerful high-level representation of 3D environments. A 3D scene graph describes the environment as a layered graph where nodes represent spatial concepts at multiple levels of abstraction (from low-level geometry to high-level semantics including objects, places, rooms, buildings, etc.) and edges represent relations between concepts. While 3D scene graphs can serve as an advanced "mental model" for

robots, how to build such a rich representation in real-time is still uncharted territory.

This paper describes the first real-time Spatial Perception engINe (SPIN), a suite of algorithms to build a 3D scene graph from sensor data in real-time. Our first contribution is to develop real-time algorithms to incrementally construct the layers of a scene graph as the robot explores the environment; these algorithms build a local Euclidean Signed Distance Function (ESDF) around the current robot location, extract a topological map of places from the ESDF, and then segment the places into rooms using an approach inspired by community-detection techniques. Our second contribution is to investigate loop closure detection and optimization in 3D scene graphs. We show that 3D scene graphs allow defining hierarchical descriptors for loop closure detection; our descriptors capture statistics across layers in the scene graph, ranging from low-level visual appearance, to summary statistics about objects and places. We then propose the first algorithm to optimize a 3D scene graph in response to loop closures; our approach relies on embedded deformation graphs to simultaneously correct all layers of the scene graph. We implement the proposed SPIN into a highly parallelized architecture, named Hydra, that combines fast early and midlevel perception processes (e.g., local mapping) with slower highlevel perception (e.g., global optimization of the scene graph). We evaluate Hydra on simulated and real data and show it is able to reconstruct 3D scene graphs with an accuracy comparable with batch offline methods, while running online [4].

3.2.2 Building 3D scene Graphs

A 3D scene graph is a layered graph where nodes represent spatial concepts at multiple levels of abstraction (from low-level geometry to objects, places, rooms, buildings, etc.) and edges represent relations between concepts.

- Euclidean Signed Distance Function (ESDF) : From a field of booleans and produces a field of scalars such that each value in the output is the distance to the nearest "true" cell in the input. Unfortunately, ESDFs scale poorly in the size of the environment [5].

This paper focuses on indoor environments and adopts the 3D scene graph introduced in [6]. The 3D scene graph consists of 5 layers that represent semantics, objects and agents, places, rooms and lastly a building node connecting all rooms. Edges connect nodes within each layer or across layers.

- Mesh and Objects (Layers 1-2) : The construction of the metric-semantic 3D mesh is an extension of the approach in [7]. The authors in [7] use Voxblox to integrate semantically-labeled point clouds into a monolithic Truncated Signed Distance Field (TSDF) and an ESDF of the environment, while also performing Bayesian inference (Defined in Chapter 2) over the semantic label of each voxel. This paper forms a volumetric model of the robot's surroundings within a radius to bound the memory used by the ESDF. Subsequently, they extract he 3D metric-semantic mesh using Voxblox' marching cubes implementation and the places, and they are passed to the Scene Graph Frontend. They segment objects by performing euclidean clustering (EC) of the 3D meshes. The results of the EC are then used to create a bounding box for each putative object.

- Places (Layer 3) : To extract the subgraph of places the authors of this paper use a Generalized Voronoi Diagram (GVD). After the GVD they identify distinctive points and connect them with edges to form the graph of places.

- Rooms (Layer 4) : The segmentation of rooms was created directly from the sparse subgraph of places. Firstly, by inflating objects, small apertures in the environment (i.e., doors) will gradually close, naturally partitioning the voxel-based map into disconnected components (i.e., rooms). Secondly, with a dilation of the map by a distance δ, every place with obstacle distance smaller than δ will disappear from the graph (since it will no longer be in the free space). A visualization of this idea is given in Figure 3.2.

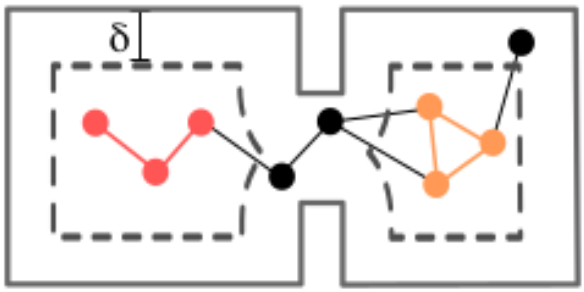

Figure 3.2: Map Dilation for Room Detection

3.2.3 Key Points

This paper overcomes that ESDFs scale poorly depending on the size of the environment by implementing algorithms that reconstruct a local ESDF of the robot's surroundings and incrementally convert the ESDF into a metric-semantic 3D mesh as well as a Generalized Voronoi Diagram, from which a topological graph of places can be quickly extracted. The main contribution of this paper is the development of the first real-time Spatial Perception engINe (SPIN), a suite of algorithms and implementations to build a 3D scene graph from sensor data in real-time.

This paper additionally investigates loop closure detection and optimization in 3D scene graphs. It proposes a hierarchical approach for loop closure detection :

- top-down loop closure detection that uses hierarchical descriptors —capturing statistics across layers in the scene graph— to find putative loop closures
- a bottom-up geometric verification that attempts estimating the loop closure pose by registering putative matches
- optimize a 3D scene graph in response to loop closures, which relies on embedded deformation graphs to simultaneously correct all layers of the scene graph, from the 3D mesh, to places, objects, and rooms.

Lastly the paper proposes a highly paralllelized implementation, named Hydra,that combines fast early and mid-level perception processes (e.g., local

mapping) with slower high-level perception (e.g., global optimization of the scene graph).

3.2.4 Paper

List of Figures

References

[1] Mark Maimone, Yang Cheng, and Larry Matthies. Two years of visual odometry on the mars exploration rovers. *Journal of Field Robotics, Special Issue on Space Robotics*, 24:2007, 2007.

[2] Raúl Mur-Artal and Juan D. Tardós. ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

[3] Lina M. Paz, Pedro PiniÉs, Juan D. TardÓs, and JosÉ Neira. Large-scale 6-dof slam with stereo-in-hand. *IEEE Transactions on Robotics*, 24(5):946–957, 2008.

[4] Nathan Hughes, Yun Chang, and Luca Carlone. Hydra: A real-time spatial perception engine for 3d scene graph construction and optimization, 2022.

[5] Helen Oleynikova, Zachary Taylor, Marius Fehr, Roland Siegwart, and Juan Nieto. Voxblox: Incremental 3d euclidean signed distance fields for on-board MAV planning. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, sep 2017.

[6] Antoni Rosinol, Arjun Gupta, Marcus Abate, Jingnan Shi, and Luca Carlone. 3d dynamic scene graphs: Actionable spatial perception with places, objects, and humans, 2020.

[7] Antoni Rosinol, Andrew Violette, Marcus Abate, Nathan Hughes, Yun Chang, Jingnan Shi, Arjun Gupta, and Luca Carlone. Kimera: From slam to spatial perception with 3d dynamic scene graphs. *The International Journal of Robotics Research*, 40(12-14):1510–1546, 2021.