ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/316273339

ORB-SHOT SLAM: Trajectory correction by 3D loop closing based on bag-of-visual-words (BoVM) model for RGBb-D visual...

Article *in* Journal of Robotics and Mechatronics · April 2017

DOI: 10.20965/jrm.2017.p0365

CITATIONS
0

READS
81

2 authors, including:

Takafumi Matsumaru
Waseda University
110 PUBLICATIONS **382** CITATIONS

SEE PROFILE

All content following this page was uploaded by Takafumi Matsumaru on 20 April 2017.

The user has requested enhancement of the downloaded file. All in-text references underlined in blue are added to the original document and are linked to publications on ResearchGate, letting you access and read them immediately.

Paper:

ORB-SHOT SLAM: Trajectory Correction by 3D Loop Closing Based on Bag-of-Visual-Words (BoVW) Model for RGB-D Visual SLAM

Zheng Chai and Takafumi Matsumaru

Graduate School of Information, Production and Systems, Waseda University
2-7 Hibikino, Wakamatsu-ku, Kitakyushu 808-0135, Japan
E-mail: icecc_sunny@163.com, matsumaru@waseda.jp
[Received August 1, 2016; accepted November 14, 2016]

This paper proposes the ORB-SHOT SLAM or OS-SLAM, which is a novel method of 3D loop closing for trajectory correction of RGB-D visual SLAM. We obtain point clouds from RGB-D sensors such as Kinect or Xtion, and we use 3D SHOT descriptors to describe the ORB corners. Then, we train an offline 3D vocabulary that contains more than 600,000 words by using two million 3D descriptors based on a large number of images from a public dataset provided by TUM. We convert new images to bag-of-visual-words (BoVW) vectors and push these vectors into an incremental database. We query the database for new images to detect the corresponding 3D loop candidates, and compute similarity scores between the new image and each corresponding 3D loop candidate. After detecting 2D loop closures using ORB-SLAM2 system, we accept those loop closures that are also included in the 3D loop candidates, and we assign them corresponding weights according to the scores stored previously. In the final graph-based optimization, we create edges with different weights for loop closures and correct the trajectory by solving a nonlinear least-squares optimization problem. We compare our results with several state-of-the-art systems such as ORB-SLAM2 and RGB-D SLAM by using the TUM public RGB-D dataset. We find that accurate loop closures and suitable weights reduce the error on trajectory estimation more effectively than other systems. The performance of ORB-SHOT SLAM is demonstrated by 3D reconstruction application.

Keywords: trajectory correction, loop closing, bag-of-visual-words (BoVW), 3D vocabulary, RGB-D SLAM

1. Introduction

1.1. Background of SLAM

In the field of robotics, simultaneous localization and mapping (SLAM) is a computational problem of constructing or updating a map of an unknown environment while simultaneously keeping track of an agent's location within it. Visual SLAM, which is becoming increasingly popular, is a type of SLAM that uses visual cameras such as RGB cameras [1] or RGB-D cameras [2] to capture images. State-of-the-art systems similar to ORB-SLAM2 [3] can work with both RGB and RGB-D cameras. Even if there is a stereo vision system [4] based on field programmable gate array in unmanned aerial vehicles, SLAM is also used as in [5]. LIDAR (light detection and ranging or laser imaging detection and ranging)-aided method [6] is also used with ORB-SLAM2.

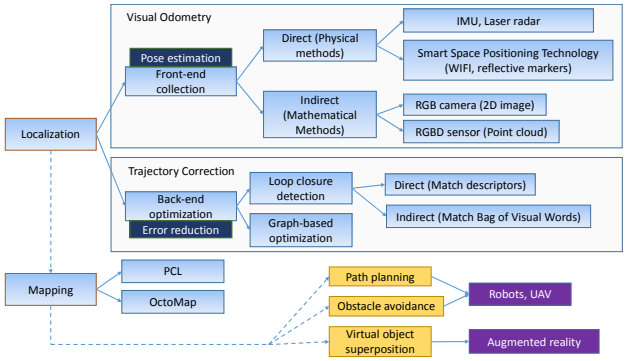

Fig. 1. SLAM framework.

SLAM is usually divided into two parts, namely, localization and mapping (**Fig. 1**). In this study, we focus on the localization part of the methodology, and then we show an application related to the mapping part at the end. The localization part is also divided into two parts: front-end and back-end. The front-end part is used to estimate the sensor pose (position and orientation), which we also call "visual odometry." The back-end part is used to reduce the accumulated error during continuous motions, which we also call "trajectory correction." Below, we present visual odometry and trajectory correction.

1.2. Visual Odometry

In robotics and computer vision, visual odometry is the process of determining the position and orientation of a robot equipped with sensors by analyzing the associated visual images. Visual odometry is divided into two parts:

feature tracking and pose estimation. In this study, we select the corner as the feature by using the ORB (oriented fast and rotated brief) feature descriptor. ORB is a fast robust local feature detector, first presented by Ethan Rublee et al. in 2011 [7]. We detect the ORB corners and use brute-force matching algorithm to track them. The corresponding 3D points are computed by a coordinate transformation based on the depth image captured by an RGB-D sensor. The pose estimation part is a perspective-n-point (PnP) [8] problem on estimating the sensor pose because we have n number of 2D keypoints on the 2D images and corresponding 3D points in 3D point clouds. We solve the PnP problem by minimizing the reprojection error based on an iterative optimization method.

1.3. Trajectory Correction

Visual odometry is a frame-to-frame operation that produces accumulated errors during exploration. A common method to reduce the accumulated error is to find a loop closure and use it in the graph-based optimization process. We build a graph for visual odometry. The vertices are the locations of a robot at different times, and the edges are the transformations of different poses. A loop closure means that the robot has almost returned to where it has been before. If we detect the correct loop closure, we adjust all sensor poses based on this new constraint.

There are two ways to detect loop closures: direct (match by descriptors) and indirect (match by visual words) methods. The direct method detects the features and computes the descriptors that are used to match images. This process is extremely slow on the second time scale. A random strategy [2] saves time for direct methods; however, it also makes the system unstable. The indirect method is based on the bag-of-words (BoW) model, which is commonly used in text analyses and language processing areas. In recent years, many researchers use it to recognize similar scenes [9, 10]. It converts the images to vectors and matches the images using those vectors [11]. This method is fast and allows all images to be compared with the new image. Moreover, the probability of missing the true loop closure is greatly reduced.

1.4. Motivation

In state-of-the-art systems, there is only a 2D constraint (adequately matched features on keypoints and 2D descriptors) for loop closures; this causes large errors on loop closure detection, and the weights of corresponding edges are single species and are indistinguishable. The weights should represent the reliability of loop closures. More details are discussed in Section 3.3. To improve the accuracy of loop closure detection, we train a visual vocabulary based on 3D descriptors and add a 3D constraint (adequately matched features on keypoints and 3D descriptors) for loop closure detection. In addition, to correct the trajectory more efficiently, we assign different weights to loop closures in the pose graph.

1.5. Paper Structure

In this paper, first we present the background and motivation of our research in Section 1. Then in Section 2, we discuss the related works as well as their limitations and remaining problems. Next in Section 3, we describe the details of the proposed ORB-SHOT SLAM (or OS-SLAM) system for offline vocabulary training as well as online loop closing by adding a 3D constraint and assigning different weights for loops in the pose graph. Then in Section 4, we discuss the experiments that we performed to determine the best feature detector-descriptor combination and to prove that the OS-SLAM system has the best performance for trajectory correction. Finally, we describe the design of a 3D reconstruction application based on the optimal trajectory in Section 5 and present the conclusion of our research in Section 6.

2. Related Works

2.1. RGB-D SLAM

2.1.1. Description

The first state-of-the-art SLAM system is RGB-D SLAM [2] for RGB-D cameras designed by the famous Computer Vision Group of Technical University of Munich (TUM). For the visual odometry part, this system has three choices on keypoints tracking which are used to calculate the camera pose: ORB, SIFT [12], and SURF [13]. This system requires landmark positions as keypoints to estimate the robot motion between two states.

For the trajectory correction part, the RGB-D SLAM system first computes a minimal spanning tree of limited depth from the pose graph, with the sequential predecessor as root node. Then, it removes n immediate predecessors from the tree, and randomly draws k frames from the tree with a bias toward earlier frames. When the robot revisits a place, once a loop closure is found, this procedure exploits the knowledge on the loop by preferring candidates near the loop closure in the sampling. To find large loop closures, RGB-D SLAM randomly samples l frames from a set of designated keyframes. A frame is added to the set of keyframes when it cannot be matched with the previous keyframe. In this way, the number of frames for sampling is greatly reduced, while keeping two keyframes having some part of the shared view.

2.1.2. Limitations

The RGB-D SLAM system uses a direct method to check whether two images are similar or not. It matches a pair of keypoint descriptors by using their distance in the descriptor space. This matching method could find the corresponding keypoints in two images; however, it requires too much time. The random strategy also has a problem of missing the true loop closure sometimes.

View publication stats