Robotics and Autonomous Systems 57 (2009) 1188–1197

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

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

A comparison of loop closing techniques in monocular SLAM

Brian Williams [a,*], Mark Cummins [a], José Neira [b], Paul Newman [a], Ian Reid [a], Juan Tardós [b]

[a] Department of Engineering Science, University of Oxford, United Kingdom

[b] Department of Informática e Ingeniería de Sistemas, Universidad de Zaragoza, Spain

ARTICLE INFO

Article history:
Available online 10 July 2009

Keywords:
Loop closing
Monocular SLAM
Submapping

ABSTRACT

Loop closure detection systems for monocular SLAM come in three broad categories: (i) map-to-map, (ii) image-to-image and (iii) image-to-map. In this paper, we have chosen an implementation of each and performed experiments allowing the three approaches to be compared. The sequences used include both indoor and outdoor environments and single and multiple loop trajectories.

1. Introduction

Loop closure detection is an important problem for any SLAM system and, since cameras have become a common sensor in robotics applications, more people are turning towards vision based methods to achieve it. In this paper, we compare three quite different approaches to loop closure detection for a monocular SLAM system. The approaches essentially differ in where the data association for detecting the loop closure is done — in the metric map space or in the image space. The three approaches are as follows:

- **Map-to-map** — Correspondences are sought between features in two submaps taking into account both their appearance and their relative positions. In this paper we look at the method of Clemente et al. [1], who applied the variable scale geometric compatibility branch and bound (GCBB) algorithm to loop closing in monocular SLAM. The method looks for the largest compatible set of features common to both maps, taking into account both the appearance of the features and their relative geometric location.
- **Image-to-image** — Correspondences are sought between the latest image from the camera and the previously seen images. Here, we discuss the method of Cummins et al. [2,4]. Their method uses the occurrences of image features from a standard vocabulary to detect that two images are of the same part of the world. Careful consideration is given to the distinctiveness of the features — identical but indistinctive observations receive a low probability of having come from the same place. This is done to minimise false loop closures.
- **Image-to-map** — Correspondences are sought between the latest frame from the camera and the features in the map. We examine the method of Williams et al. [5] who find potential correspondences to map features in the current image and then use RANSAC with a three-point-pose algorithm to determine the camera pose relative to the map.

First, we describe the underlying monocular SLAM system used during the experiments. Then, we outline in more detail the chosen implementation of each of the different approaches to loop closure. Results are then given on the performance of each algorithm at closing loops in three different environments. Then one of these sequences is used for more extensive experiments to allow quantitative comparisons to be made between the three methods.

2. The monocular SLAM system

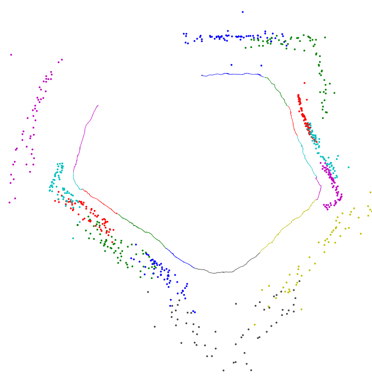
The monocular SLAM system we use is derived from Davison's original system [6,7], but with a few improvements to bring it up to date. The underlying system is essentially the same as the system described in [1] but with our own relocalisation module [3] to recover from situations where the system becomes lost. We have also added a system to prevent premature loop closure and added the ability to perform independent map joining. Here we give a brief description of the system, so details of the loop closing system can be better understood.

2.1. Map building

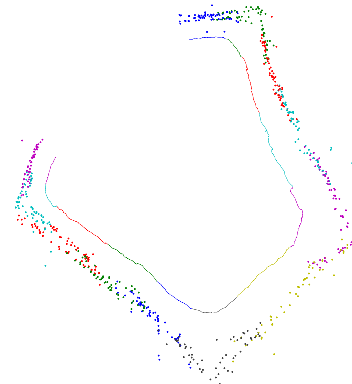
The monocular SLAM system tracks the pose of a handheld camera while simultaneously building a map of point features in 3D using the EKF. The points are initialised using the inverse depth parameterisation [8], and they are recognised in subsequent frames via normalised cross correlation. An image patch is stored when the feature is initialised, but is warped to correspond with

* Corresponding author. Tel.: +44 1865283049.
E-mail address: bpw@robots.ox.ac.uk (B. Williams).

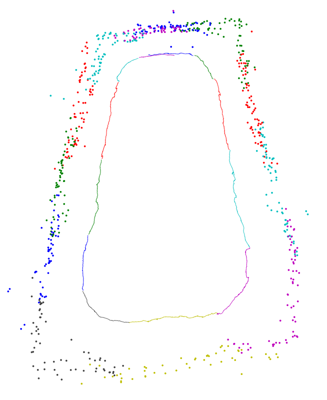
0921-8890/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.robot.2009.06.010



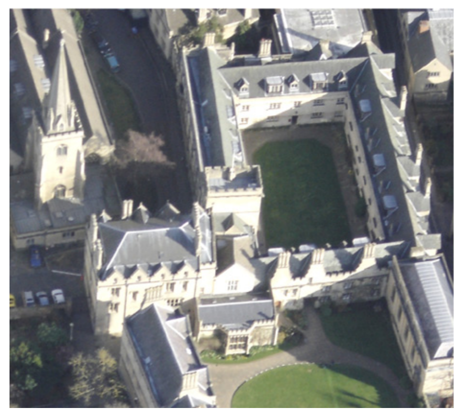
(a) Local maps obtained with pure monocular SLAM.



(b) Local maps auto-scaled.



(c) After loop closure.



(d) Aerial view of the courtyard.

Fig. 1. Pembroke college sequence: Twelve submaps with a total of 848 features were made during the 70 m trajectory around the courtyard.

the current camera pose estimate. To speed up the observation of features, the image is only searched in an ellipse given by the uncertainty in the camera and feature estimate in a process called active search. By gating the search in this way the chances of incorrect data association are reduced. This is further helped by the use of the joint compatibility branch and bound algorithm (JCBB) [9] which detects observations which are incompatible with the others and rejects them.

Despite the improvement given by active search and JCBB, there is still a chance of incorrect data association, particularly near loop closures when the system can believe that distant features are again visible and attempt to measure them. If the system is allowed to observe these features as usual, it will likely make incorrect data association due to the large uncertainty in the camera pose relative to these features. Our approach is to prevent the system from making these observations and delay the loop closure until a separate loop close module has detected it (Section 3). To determine which observations to attempt, we make use of the covisibility data from all the features in the map.

With every set of observations, a tally is updated indicating which features have been successfully observed together. Using this information, a simple graph is constructed where a vertex corresponds to each feature, and the edges indicate those that have been observed together. This graph provides an easy way of determining which features are in the local neighbourhood and which are not. Those which are distant in the graph are not eligible for observation since their relative position to the local features is very uncertain and attempting their observation would likely lead to incorrect data association. Readers should note that another way of determining feature covisibility in a stochastic map is to compute the inverse covariance, the information matrix. Features that have been covisible at some point will have a high value of co-information.

2.2. Larger maps

Due to the accumulation of linearisation errors in the EKF algorithm as well as the increase in update time, we limit our system to quite small local maps (around 70 features). To map larger regions, the Hierarchical SLAM [10] technique is used. This allows the system to map an environment by building a series of submaps, each of which is small enough to allow the system to be run in real-time as well as reducing linearisation errors. This method was already applied to monocular SLAM in [1] but here we show it working in more complex environments with multiple loops.

As each new submap is created, the origin of its base reference frame is stored in the state vector of the submap from which branched off. This transformation is then used to determine the relative position of the two submaps. However, for monocular SLAM, this transformation must also include the scale difference which is determined as follows. Each new submap is created with new features initialised at the location of some of the features in the previous map. The geometry of these common features in each submap is used to determine the relative scale. Since the features were newly initialised, information is not shared between the submaps and they remain independent. This scale correction can be seen in Fig. 1(a) and (b).