An ICP-based Navigation System for Human Support Robot

Zhang Haibin (Tokyo Metropolitan University)

As concern for Japan's aging population and lack of workforce, many researches are focusing on nursing and healthcare-related robot to help ease the burden on society. We use the human support robot (HSR) to assist with independent living in the home for handicapped people. In this paper, we proposed a way to estimate the position of robot while moving as part of our navigation system. Using RGB image and depth maps captured extracted by Xtion, we calculate the 3D coordinate's position by a pair of features based on ICP algorithm. Lastly, we use the TUM benchmark to validate our system's accuracy.

1. Introduction

For our whole robot navigation system, we use figure below as explanation.

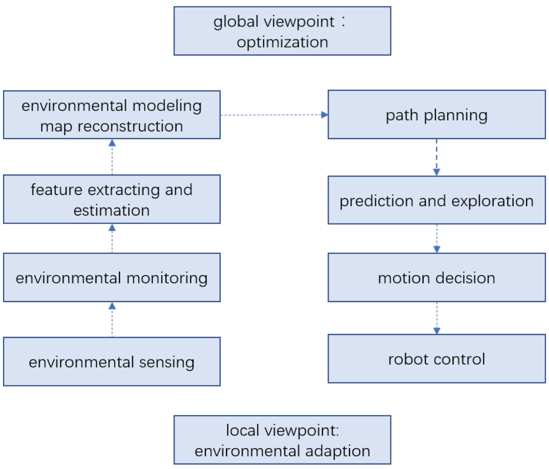

Fig. 1 Flowchart of robot navigation

The figure is a complete robot navigation system flow chart. Depending on robot's environmental sensors to obtain environmental information, we can extract image feature and set those features as landmark. Ultimately, we get the environmental model. while the left part briefly introduce robot localizing and map reconstruction, the right part is that robot navigation and control for a specific target. With the map model, mobile robots can then navigate autonomously, exploring in the map and deciding the optimizing way.

Location estimation is the basis of navigation. Various techniques have been described for estimating the position and orientation of a robot. The main problem point is how to estimate camera's motion depending on sequence of images. In order for a mobile robot to navigate through an unknown environment, the robot has to estimate a map of the environment while at the same time localizing itself with respect to this map.

2. Background

HSR is being developed to support independent living for people with limited arm or leg mobility.

Fig. 2 Human Support Robot

The Xtion set on HSR's head is capable of generating a 640×480 image where each pixel contains a distance from the depth camera (precision 1mm). This image can be updated at 30fps. In this paper we will primarily focus on the RGB color data and depth maps.

3. System Overview

In Fig. 3, we illustrate an overview of our system for estimating the robot's position by ICP algorithm. The input data to the system are an RGB color image and a depth image captured using a Xtion PRO LIVE sensor mounted on the head of HSR. Then the processing is like below:

(1) For the new arrival of the current frame, extract the key points and descriptors.

(2) If the system is not initialized, take this frame as a reference frame, calculate the 3D position of the key points according to the depth map, and return to the first step.

(3) Estimate the motion T_k of the reference frame and the current frame

(4) Determine whether the above estimation is successful or not

(5) If successful, the current frame as a new reference frame, return to the first step. If failed, record the number of consecutive lost frames. When the continuous loss of more than a certain number of frames, set VO state is lost, then set algorithm ends. If not exceeded, return to the first step.

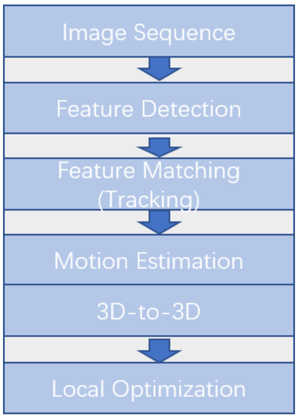

Fig.3 Pipeline of visual odometry

We first define the concepts of reference frame and current frame. With the reference frame as the coordinate system, we match the current frame with it and estimate the motion relationship. It is assumed that the transformation matrix of the reference frame relative to the world coordinate system is $\mathrm{T_{rw}}$, and the current frame relative to the world coordinate system are T_{cw}, so the estimated motion T_{cr} can be calculated due to $T_{cw} = T_{cr}T_{rw}$.

From time t-1 to time t, we take t-1 as reference to find the motion at time t.

3.1 Feature extraction and matching

ORB feature consists of two parts, which are key-point and descriptor. Therefore, the task that extract orb features is divided into the following two steps.

(1) Fast corner extraction: find the corner of the image. Comparing with the original FAST, the main direction of the feature point is calculated in ORB, which adds rotational invariance to the subsequent BRIEF descriptor.

(2) BRIEF descriptor: Describe the surrounding image area of the feature point extracted in the previous step.

Next, we using feature matching between two frames. Feature matching is a crucial part of visual slam, which can determine the corresponding relationship between the currently seen signpost and the previous signpost. One of the simplest methods of feature matching is Brute-Force Matcher. That is, for each feature point and all feature points measure the distance between descriptors, and then sort, only take the nearest one as the matching point. We use the Hamming distance as a descriptor distance, and we can express the similarities between the two features.

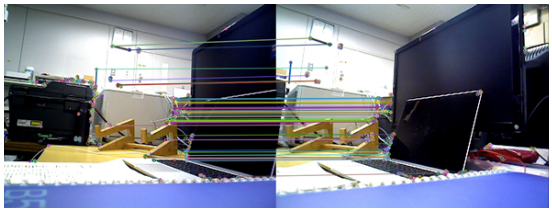

Fig.4 Feature matching of two frames

3.2 Iterative Closest Point

We get the corresponding relationship between the feature points of the image, so we can use 3d-3d matching points' pair to estimate the motion of camera.

If we have two group 3d matching points,

$$\mathrm{P} = \{\mathrm{p}_1, \dots, p_n\},\ P' = \{p'_1, \dots, p'_n\}$$

to make $\mathrm{p_i} = \mathrm{Rp'_i} + \mathrm{t}$. We can use ICP to solve it and get pose estimation of to two group matching points.

For the errors of i_{th} points,

$$\mathrm{e_i} = \mathrm{p_i} - (\mathrm{Rp'_i} + \mathrm{t}).$$

Building the least squares problem,

$$\min_{R,t} J = \frac{1}{2}\sum_{i=1}^{n} \|(p_i - (Rp_i{}' + t))\|^2.$$

To solve this problem are in below. If we define points' centroid,

$$\mathrm{p} = \frac{1}{\mathrm{n}}\sum_{i=1}^{n}(p_i),\ p' = \frac{1}{n}\sum_{i=1}^{n}(p'_i)$$

then, $\min_{R,t} J = \frac{1}{2}\sum_{i=1}^{n} \|(p_i - (Rp_i{}' + t))\|^2$

Because of $p_i - p - R(p_i' - p') = 0$, therefore,

$$R^* = arg\min_R \frac{1}{2}\sum_{i=1}^{n} \|q_i - Rq_i'\|^2,$$
$$t^* = p - Rp'$$

The main steps are below:

(1) Calculate two group points' centroid p, p', and then for every point distract the centroid, $q_i = p_i - p, \quad q_i' = p_i' - p'$

(2) Solve the optimization problem,

$$R^* = arg\min_R \frac{1}{2}\sum_{i=1}^{n} \|q_i - Rq_i'\|^2,$$
$$t^* = p - Rp'$$
$$\frac{1}{2}\sum_{i=1}^{n} \|q_i - Rq_i'\|^2 = -tr(R\sum_{i=1}^{n} q_i' q_i^T)$$

(3) Use SVD to solve this question. Suppose that, $W = \sum_{i=1}^{n} q_i q_i'^T$, $W = U\Sigma V^T$.

Therefore, $R = UV^T$ when W are full rank, we calculate for t^*, $t^* = p - Rp'$.

In the Fig. 4 below, we show the camera's motion trajectory based on color and depth maps, which is the change of the coordinate of the current frame relative to the frame of reference.

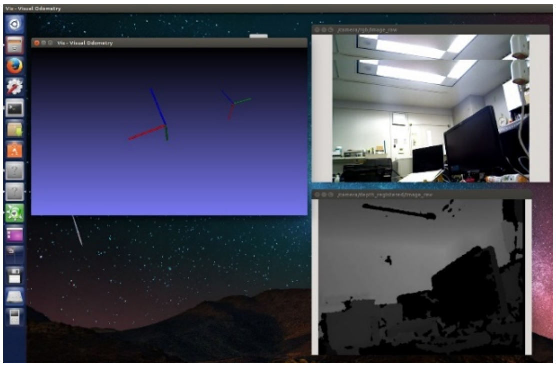

Fig. 5 The estimation of position using RGB-D image

4. Experiment

We use the benchmark for the evaluation of RGB-D visual odometry system. The dataset contains color images, depth maps, and associated ground-truth camera pose information. the data sequence "freiburg1_desk" contains several sweeps over four desks in a typical office environment. Further, we proposed an evaluation metrics that can be used to assess the performance of a visual odometry system.

Additionally, the ground-truth file included contains the ground truth trajectory stored as a timestamped translation vector and unit quaternion (format: timestamp tx ty tz qx qy qz qw).

Duration	23.40s
Duration with ground-truth	23.35s
Ground-truth trajectory length	9.263m
Avg. translational velocity	0.413m/s
Avg. angular velocity	23.327deg/s
Trajectory dim.	2.42m * 1.34m * 0.66m

Table 2: Basic information of "freiburg1_desk" in TUM Benchmark

For a navigation system, additionally the global consistency of the estimated trajectory is an important quantity. The global consistency can be evaluated by comparing the absolute distances between the estimated and the ground truth trajectory.

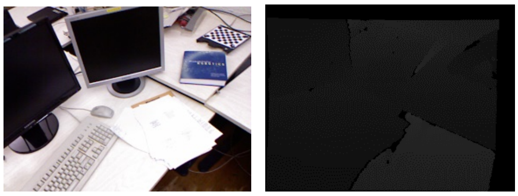

Fig. 6 Example of freiburg1_desk dataset

We calculate the cumulative error of the true trajectory and the estimated trajectory according to the time.

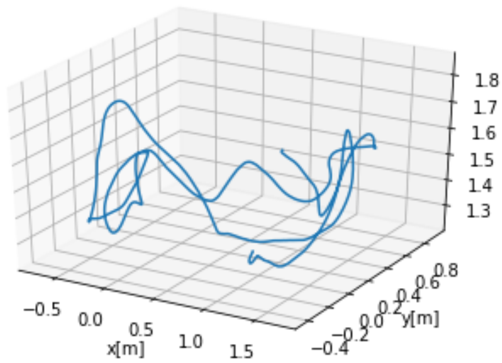

Fig. 4 Visualization of the ground-truth trajectory on the "fr1/desk1" sequence.

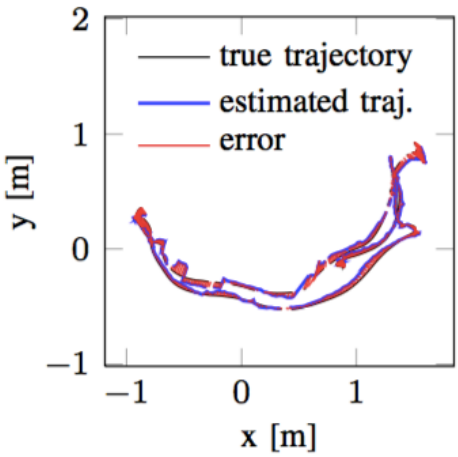

Fig. 5 Visualization of the absolute trajectory error on the “fr1/desk1” sequence.

5. Conclusion

We see that the visual odometer can estimate the camera movement and the location of the feature points over a period of time, but this partial approach has obvious drawbacks:

(1) Easy to lose. When the key frame moves too fast, it will cause the feature point to be lost. Once lost, we either wait for the camera to turn around, save the reference frame and compare with the new frame, or reset the entire VO to track the new image data.

(2) Trajectory drift. The main reason is that the error of each position estimation will be accumulated to the next estimation, resulting in inaccurate long-time trajectory. Larger local maps may alleviate this phenomenon, but it is always there.

6. Future Works

Compared with human, a robot’s ability to autonomously execute tasks is still very limited. For the whole navigation system, we hope that, when people set the target location and orientation, the HSR is able to autonomously move to the new position, detecting and avoiding obstacles as needed. It is necessary to have previously created 3D map of the environment using the equipped Laser Range Finder and Xtion. Until now, we have only looked at the part of the robot vision odometer. Next, using the environment map obtained from Xtion, we study the robot's path planning part of the map and expect the optimal path.

In the following, we will conduct research on other aspects of the navigation system along the route proposed in the previous article. We hope that robots can fulfill their tasks according to human instructions as soon as possible so as to serve human beings to reduce their workload.

Reference

[1] K. Hashimoto, F. Saito, T. Yamamoto and K. Ikeda, "A field study of the human support robot in the home environment," 2013 IEEE Workshop on Advanced Robotics and its Social Impacts, Tokyo, 2013, pp. 143-150.

[2] Krystof Litomisky, Consumer RGB-D Cameras and their Applications, University of California, Riverside.

[3] K. Nagahama, H. Yaguchi, H. Hattori, K. Sogen, T. Yamamoto and M. Inaba, "Learning-based object abstraction method from simple instructions for human support robot HSR," 2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM), Banff, AB, 2016, pp. 468-475.

[4]J. Sturm, N. Engelhard, F. Endres, W. Burgard and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, 2012, pp. 573-580.

[5]高翔，张涛，颜沁睿，刘毅，视觉 SLAM 十四讲：从理论到实践，电子工业出版社, 2017

[6] “Human Support Robot (HSR)” http://www.brain.kyutech.ac.jp/~hma/wordpress/robots/hsr/

[7] “Aging of Japan”, https://en.wikipedia.org/wiki/Aging_of_Japan