Journal of Physics: Conference Series

PAPER • **OPEN ACCESS**

Using external sensors in solution of SLAM task

To cite this article: V S Provkov and I S Starodubtsev 2018 *J. Phys.: Conf. Ser.* **1015** 032184

IOP Conf. Series: Journal of Physics: Conf. Series **1015** (2018) 032184 doi:10.1088/1742-6596/1015/3/032184

Using external sensors in solution of SLAM task

V S Provkov[1], **I S Starodubtsev**[1,2]

[1] Ural Federal University, Yekaterinburg, Russia
[2] Krasovskii Institute of Mathematics and Mechanics UB RAS, Yekaterinburg, Russia

E-mail: ProvkovV@mail.ru; StarodubtsevIS@robolab.tk

Abstract. This article describes the algorithms of spatial orientation of SLAM, PTAM and their positive and negative sides. Based on the SLAM method, a method that uses an RGBD camera and additional sensors was developed: an accelerometer, a gyroscope, and a magnetometer. The investigated orientation methods have their advantages when moving along a straight trajectory or when rotating a moving platform. As a result of experiments and a weighted linear combination of the positions obtained from data of the RGBD camera and the nine-axis sensor, it became possible to improve the accuracy of the original algorithm even using a constant as a weight function. In the future, it is planned to develop an algorithm for the dynamic construction of a weight function, as a result of which an increase in the accuracy of the algorithm is expected.

1. Introduction

Modern algorithms of spatial orientation, such as EKF SLAM, FastSLAM, DP-SLAM, PTAM, were described in articles in the mid 2000s and were of a significant importance as the algorithms used to solve problems in computer vision.

At the moment, these algorithms are used by programmers to create their own solutions to solve the problems of spatial orientation of robots [1], which improve the performance of these methods, increasing the accuracy, speed; eliminate the noise of sensors using filters; get an average reading of the movement in accordance with the indicators of several sensors, etc.

Nevertheless, these algorithms have their drawbacks, for example, the accumulating error or the progressing uncertainty of the robot's location (more details are given below). However, these disadvantages can be partially compensated, using a bunch of additional sensors, the readings of which can be used to correct the trajectory. Further, let us offer a solution using external sensors, which can be used to smooth out inaccuracy and reduce errors.

Figure 1. Example of capture of key points using PTAM algorithm

2. The principal approaches overview

Let us consider two principal approaches to solving the SLAM problem.

2.1. The PTAM algorithm

PTAM (Parallel Tracking and Mapping) solution allows one to build a map of the area and to display elements of augmented reality using only one camera [2]. Solution does not require the use of markers. The search and orientation are based on the key points (Fig.1). The depth of the scene at a single point is calculated on the basis of the parallax effect and perspective distortion.

The features of this algorithm are:

- tracking and mapping are separated and run in two parallel threads (processes);
- a large number of points (several thousand) are mapped;
- map construction is based on keyframe. Keyframe is selected if the Euclidean distance of the new frame relative to all keyframes exceeds 12 % of the average depth of the scene space;
- new points are initialised in the epipolar search;
- the initial position for the first 2 keyframes is evaluated at boot time.

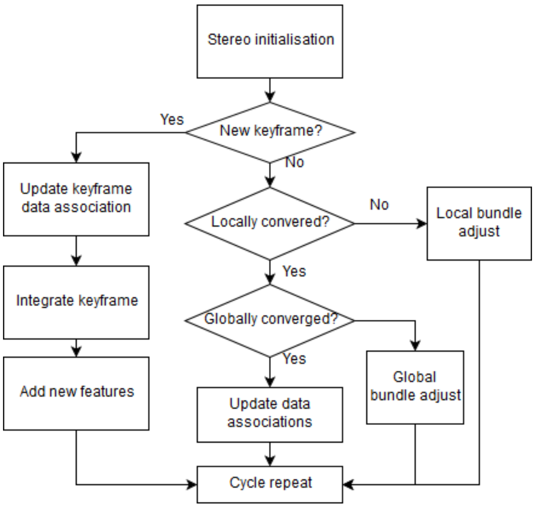

Figure 2. The block diagram of the PTAM algorithm.

Figure 2 shows a block diagram of the operation of the PTAM algorithm.

The map construction takes place in 2 stages:

- Initially, the map is built using stereo technology by capturing 2 frames using the horizontal movement of the camera, then the map is constantly being refined and expanded as new key frames are added from the tracking system.
- Bundle adjusts local and global packaging. The image is divided into 2 samples to create a five-level pyramid image. There is an optimization of default intensive. [3]

Pros of this algorithm:

- The error does not accumulate, as the frames are taken into account by key points and each key point is assigned its own weight (degree of confidence).
- Proceeding from this property, there are positive sides - high accuracy of the algorithm and high resistance to erroneous measurements.

Cons of the algorithm is that the work of the whole algorithm depends on the first key frames; when working, situations arise when the algorithm can not correctly determine where the plane is located (Figure 3). A high dependence on the equipment can be considered as another, more significant disadvantage. Since only one camera is used, it is very difficult, and sometimes, impossible to conduct a tone compensation or compensate for the change in illumination. Also, the algorithm is very sensitive to digital noise.

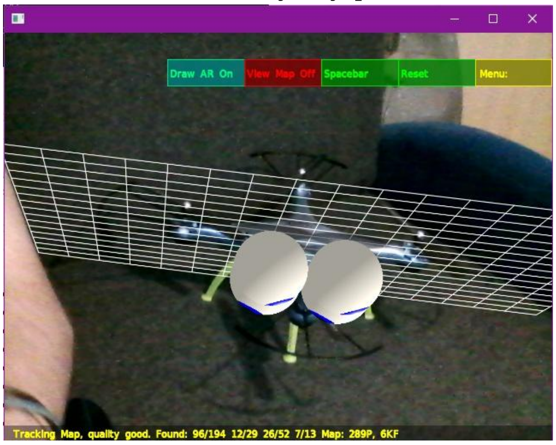

Figure 3. Incorrectly selected key frames lead to a wrong determination of the plane. in space.

2.2. Algorithm EKF – SLAM (Extendet Kalman Filter)

Another principal approach is the probabilistic evaluation of the actor's own position on the scene on the basis of visible landmarks, for example, using the advanced Kalman filter [4].

The difference between EKF (extended Kalman filter) and Kalman filter is that in this method, it is possible to use nonlinear motion models [5].

Figure 4 shows the scheme of the robot movement relative to the control points (landmarks given in advance). The dotted line is the trajectory of the robot. The gray oval region is the evaluation of the position of the robot. 6 green dots and transparent oval area are the mark positions and their position evaluation. In figure 4, the robot sees the first landmark and the uncertainty of the location of the other landmark decreases, "loop closure" occurs. However, the uncertainty of assessing one's own position is reduced insignificantly.

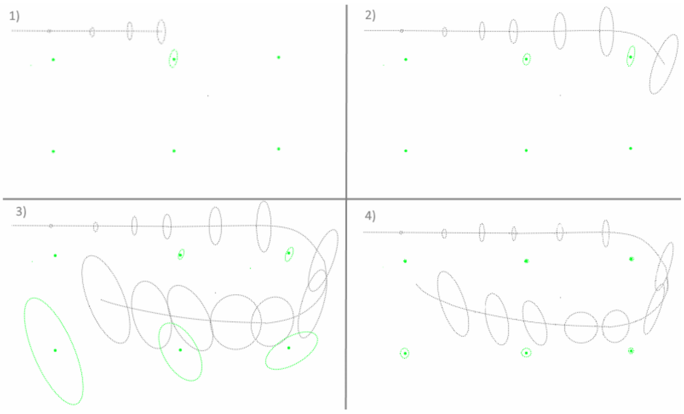

Figure 4. An example of the EKF SLAM algorithm.

This feature is a significant drawback. Even after loop closure, there is no concrete reliable understanding of the position, in which the robot is. Also, in the course of further work, an error is accumulated and this accumulation is in a quadratic dependence on the number of observed landmarks.

3. Implementation with the addition of new sensors

To increase the accuracy of positioning and to reduce the effect of the accumulated error, an experiment was carried out to implement the SLAM method, taking into account data from additional sensors.

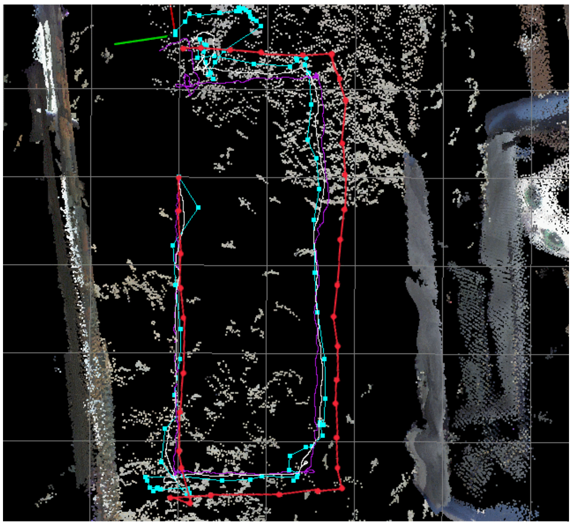

Figure 5. The result of the algorithm with the display of different trajectories.

The implemented method uses a depth camera (Kinect) as a basic. As the basic algorithm of SLAM. there is the algorithm RGBD-SLAM [6]. As external sensors, let us use a bunch of gyroscope + accelerometer + magnetometer (MPU-9250) This sensors allow one to obtain additional accuracy and, as a consequence, a decrease in the accumulated error for the determination of the robot in space and the construction of the terrain map. Gyroscope allows one to avoid the error of orientation by the jump or vibration, magnetometer helps to determine the orientation of the robot, based on the magnetic vector of the planet; the accelerometer corrects the accumulation of errors in motion. These sensors complement each other and, after calibration, show good data with low error. Also, MPU-9250 is presented in the form of a small device that combines all 3 sensors, which simplifies the placement of the sensor on a moving robot.

To calculate the position at time t, the authors used a weighted linear combination of the positions obtained from data of the RGBD camera and the nine-axis sensor.

$$\bar{x}_t = (\alpha(t))R_t + (1 - \alpha(t))G_t \tag{1}$$

where $\alpha(t)$ – weighting function;

$$\forall t: t \in (t_0, t_{end}), \alpha(t) \in [0,1] \tag{2}$$

where R_t, G_t – position data at time t, obtained from data from the RGBD camera and external sensors, respectively. Next, the value of $\bar{x}_t$ is passed to the Extended Kalman Filter as an observable at time t [7, 8].

Figure 5 shows an example of the work of the algorithm.
Here:

- The violet line - the trajectory of the robot in accordance with the sensor measurements.
- The blue line - the motion display in accordance with the data obtained by the RGBD-camera.
- The white line is the calculated motion obtained by the algorithm taking into account the accelerometer sensor, the magnetometer, the gyroscope and the constructed map using the RGBD-camera.
- The red line is the approximate movement of the robot in accordance with the photo-images (was entered manually).

In this experiment, constant $\alpha(t)=c$, $c=0,4$ was taken as a weight function

4. Conclusion

The results of the experiment are as follows: the orientation based only on optical information is possible, as well as the orientation based on sensors. Figure 5 shows that both violet and blue lines as a whole are fairly close in the real trajectory of the robot. However, both methods have their own features. So, electromagnetic sensors work much more precisely when turning, while the optical method has errors. In this case, the optical method shows itself better in relatively straight sections without sharp turns. Another conclusion is that using a weighted combination of data from two different types of sensors improves accuracy of the optical method by 12% and reduce the error of the sensors by 5% (Table 1), even when using a constant as a weight function.

Table 1. The average deviation of the constructed trajectory from the real.

	Direct trajectory					Rotation area				
Optical method, mm	69	43	39	45	43	47	67	74	68	70
Sensor readings, mm	56	51	55	53	55	51	55	49	55	53
Combination of methods, mm	55	44	43	47	47	49	56	53	57	55

As a further work, it is planned to develop an algorithm for dynamical construction of a weight function as a function of the motion vector. Also, the plans include optimization of the algorithm, parallelization of tasks to accelerate the work, testing the work of the complex when building a map on large polygons with various obstacles.

The developed algorithm allows one to build a map of the area, and, using additional sensors, to correct the trajectory of the robot movement, which increases the accuracy of the work.

The results of this experiment will be used to realize project of the test area for the robotic security complex [9].

5. References

[1] Retrieved from: http://openslam.org/

[2] Klein G, Murray D 2007 Parallel Tracking and Mapping for Small AR Workspaces *6th IEEE and ACM International Symposium on Mixed and Augmented Reality* (Washington: IEEE Computer Society)

[3] Retrieved from: https://github.com/Oxford-PTAM/PTAM-GPL

[4] Siciliano B, Khatib O 2008 *Springer Handbook of Robotics* (Luxembourg: Springer)

IOP Conf. Series: Journal of Physics: Conf. Series **1015** (2018) 032184 doi:10.1088/1742-6596/1015/3/032184

[5] Calonder M 2006 *EKF SLAM vs. FastSLAM - A Comparison.* Retrieved from: https://www.oalibrary.org/papers2/236cc809-ef45-4251-bf50-bc95dcad999a/
[6] Retrieved from: https://github.com/felixendres/rgbdslam_v2
[7] Welch G, Bishop G 2006 *An Introduction to the Kalman Filter* (Chappel Hill: ACM)
[8] Grewal M, Andrews A 2001 *Kalman filtering: theory and practice using MATLAB* (New York: John Wiley and Sons)
[9] Mikhailov I O, Starodubtsev I S, Provkov V S 2017 Virtual test area for robotic security complex *AIP Conference Proceedings* **1886** 367-369