Comparison of indoor robot localization techniques in the absence of GPS

Regis Vincent*, Benson Limketkai, Michael Eriksen
SRI International, 333 Ravenswood Avenue, Menlo Park, CA, USA 94025

ABSTRACT

When available, GPS is the quick and easy solution to localizing a robot. However, because it is often not available (e.g. indoors) or not reliable enough, other techniques, using laser range finders or cameras have been developed that offer better performance. For 2D localization,lLaser range finders are far more precise and easier to work with than cameras. We report here on the performance of several implementations of the main class of localization algorithms that use a laser, Simultaneous Localization And Mapping (SLAM) on the RAWSEEDS benchmark. SRI International's SLAM system has an RMS error in XY of 0.32m (0.22%). This is the best reported performance on this benchmark.

Keywords: robot, GPS, localization, SLAM

1. INTRODUCTION

Localization for a robot in the absence of a global positioning system (GPS) is a hard problem and all techniques require external sensors. The most common sensors currently in use in robotics are laser range finders, cameras, and sonar. Sonar sensors are the least reliable and the most difficult to work with. Although they can be used to provide 3D localization, cameras are also unreliable indoors, unless relatively constant lighting can be guaranteed. For 2D localization, the laser range finder is the most precise and easiest to use.

For environments where the precise layout of a building is already known (e.g., from blueprints or previous mapping), probabilistic localization algorithms can be used. The most common of these algorithms are EKF Localization [2] and Monte Carlo Localization (MCL) [2].

When a building layout is not known , the most widely used class of algorithms using laser range finder is Simultaneous Localization And Mapping (SLAM). The most commonly used SLAM algorithms are EKF-SLAM [1], FastSLAM [10], Rao-Blackwellized Particle Filter (RBPF) [11], and GraphSLAM [2].

Comparing these algorithms is not easy because it is difficult to acquire absolute ground truth. Recently, the European project RAWSEEDS [4] has produced a set of data designed to facilitate the comparison.

In this paper we present the results of an implementation of MCL and two implementations of SLAM contained in SRI International's KARTO™ toolkit [3] on the RAWSEEDS dataset. We compare our results to published results using the same dataset.

1.1 RAWSEEDS Data

RAWSEEDS is a robotic dataset with which robotic algorithms can be compared. The data was collected at the University of Milan from laser range finders, odometry, and cameras. In addition, the project has provided the ground truth for all datasets, making it easy to compare the localization precision of any algorithms.

For measuring the precision of our localization systems, we use the Bicocca-10-25b dataset, with the robot having taken the path shown in Figure 1.

*vincent@ai.sri.com; phone 1 650 859-4037; fax 1 650 859-3735; www.sri.com

Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XV,
edited by Russell S. Harmon, John H. Holloway Jr., J. Thomas Broach, Proc. of SPIE
Vol. 7664, 76641Z · © 2010 SPIE · CCC code: 0277-786X/10/$18 · doi: 10.1117/12.849593

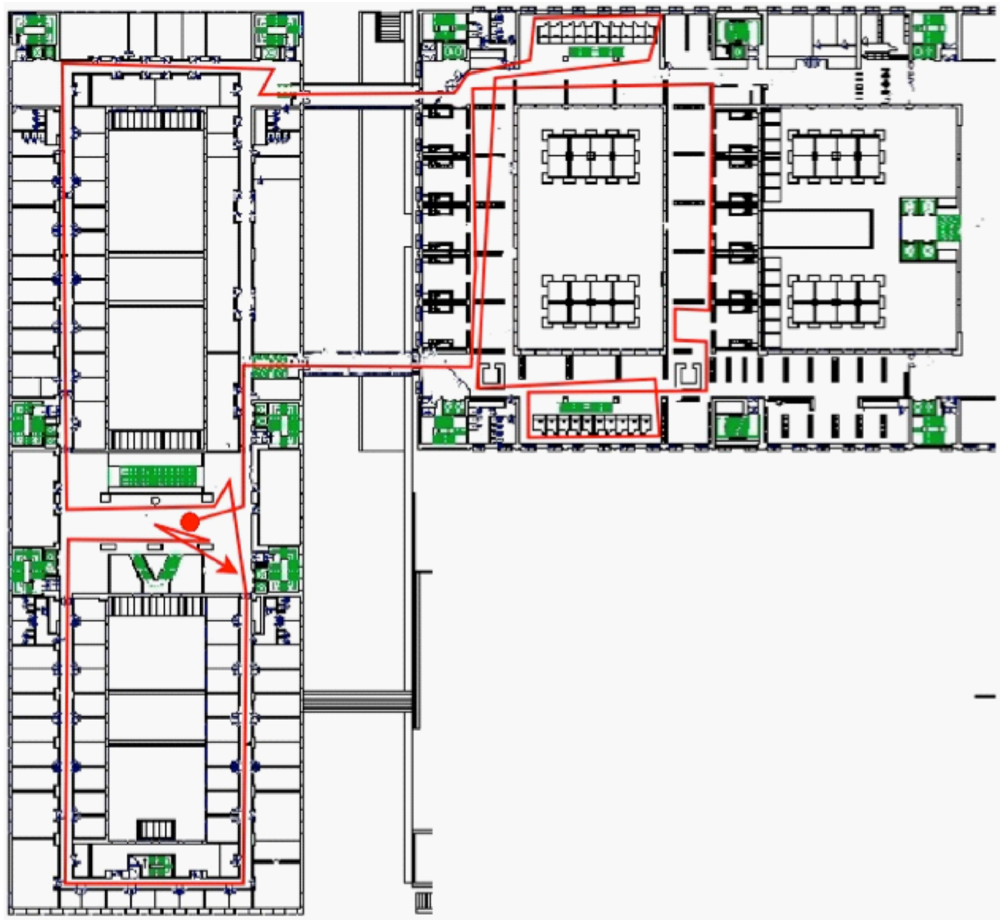

Figure 1: Path followed by the robot.

The dataset produced by RAWSEEDS includes raw odometry data as well as raw data from several lasers. For our comparison, we used the front [what is the "front" laser?] laser (SICK LMS-291). This dataset contains 131626 laser scans.

The RAWSEEDS team had set up some networked stationary lasers that were tracking the position of the robot, it produces the ground truth of where the robot was exactly. The resulting findings were limited, since only a small portion of the floor (around the starting area) was covered by this system. This ground truth system has a precision of 5 cm. Using this dataset, the University of Freiburg computed an extended ground truth by "extrapolating" the real ground truth. The method used was to have GraphSLAM produce a map and then manually adjusting the graph to best match the reference floor plan. By using this dataset, the precision of the localization can be scientifically validated and, more important, different localization algorithms can be compared.

1.2 Algorithms

We have tested one implementation of Monte Carlo localization and three implementations of GraphSLAM,

1.2.1 Monte Carlo Localization

Monte Carlo Localization is a relatively new probabilistic algorithm [2, p. 252]., which follows the recursive Bayesian filtering scheme. The key idea of this approach is to maintain a probability density of the location of the robot at a particular time, given all observations and all control inputs. MCL is a variant of particle filtering [6], where each particle corresponds to a possible robot pose and has an assigned weight. Furthermore, the particle set needs to be resampled according to the assigned weights to obtain a good approximation of the pose distribution with a finite number of particles.

1.2.2 GraphSLAM

GraphSLAM ([2, page 337] , computes a map by means of graph optimization [9]. The idea is to construct a graph out of the sequence of measurements. Every node in the graph represents a pose along the trajectory taken by the robot and the corresponding measurement obtained at that pose. The edge between nodes represents the motion between two consecutive poses. A least squares error minimization approach is applied to obtain the most likely configuration of the graph. The map is then computed by traversing the graph and combining the laser measurements.

Karto's implementations of GraphSLAM use a loop closure technique developed by Lu and Milios [5]. The Karto mapping produces a corrected pose in real time as the robot moves into the environment. A by-product of this computation is the map of the building. Figure 2 shows the resulting map of the environment after processing the Bicocca-10-25b.

Freiburg's GraphSLAM uses a similar technique for the scan-matching but uses a different solver for the loop closure . Freiburg's solver is called TORO. Karto uses a Cholesky decomposition of a dense matrix as its solver and more recently Sparse Pose Adjustment [7].

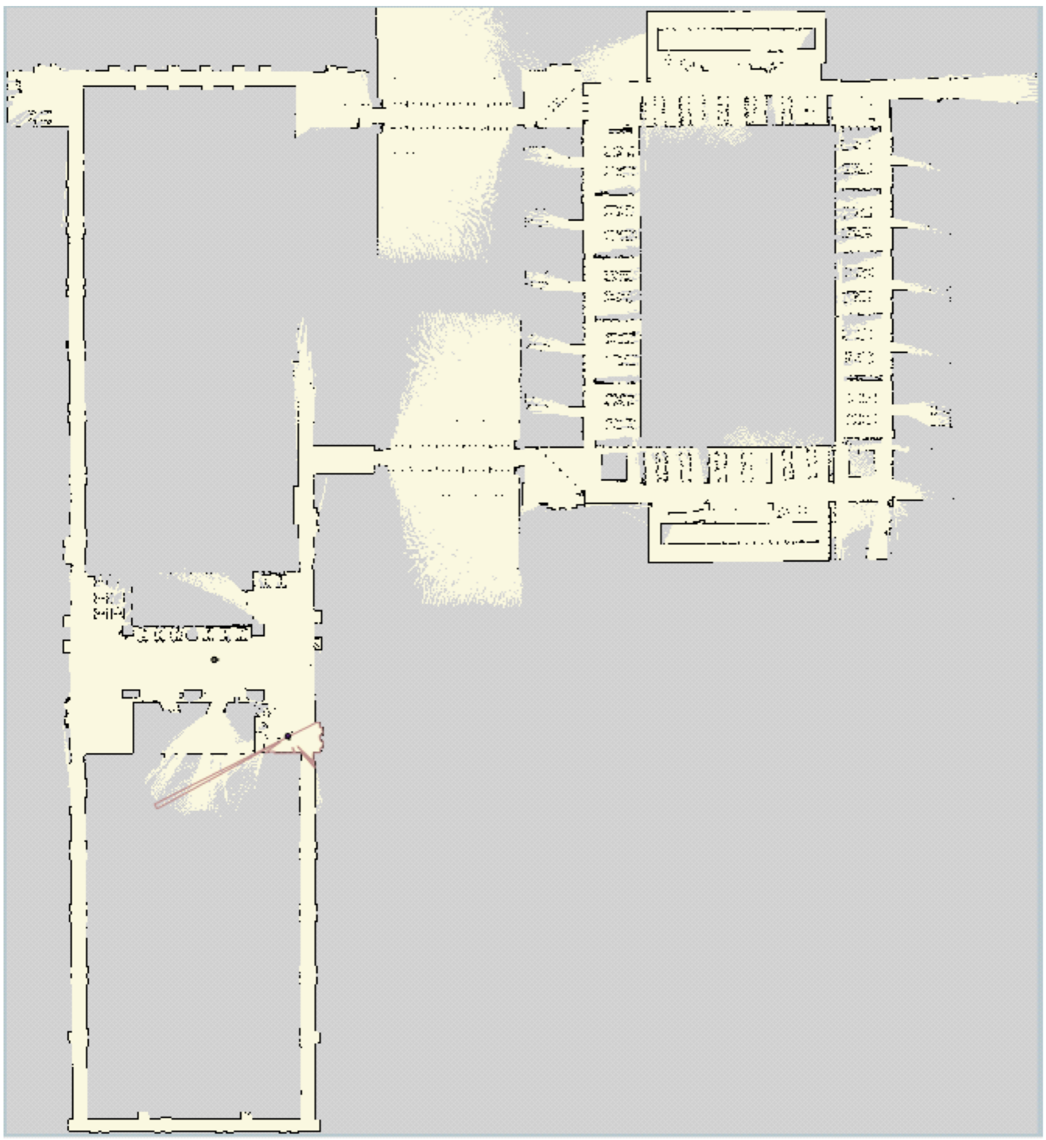

Figure 2: Map produced by Karto.

2. RESULTS

We have tested four different algorithms:

- SLAM with Karto 1.1 that uses Cholesky decomposition of a dense matrix as the solver
- GraphSLAM from University of Freiburg that uses TORO as the solver
- SLAM with Karto 2.0 that uses Sparse Pose Adjustment [7] as the solver
- Monte Carlo localization that is part of Karto

Overlaying both paths, one from ground truth and one produced by any of these algorithms, does little to help in visualizing any errors (Figure 3). The building measures 112.5 m by 140 m. An error of 1 m corresponds to an error of 0.7%. All the algorithms, we have tested, have an error between 0.12 m (0.08%) and 0.63 m (0.45%), which is not visible.

In order to visualize any error, we had to zoom to the lower right corner of the map where the maximum error occurs. At that point the maximum error is 1.21 m (see Figure 4).

Figure 3: Comparison between the ground truth and the position computed by Karto 2.0.

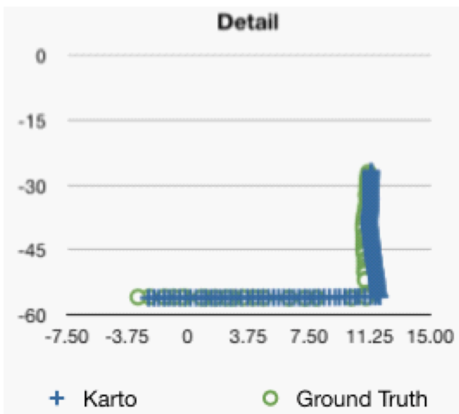

Figure 4: Details of the lower left corner of the building, where the error is maximum.

Kummerle et al have already shown that GraphSLAM is the best of the current SLAM algorithms [8]. Table 1 shows that the KARTO SLAM algorithms yield the best RMS and maximum error of all the algorithms that do not depend on a reference map. .

The maximum error can seem large (between 0.6 and 1.5m) but this error occurs mainly in one corner where the slight angle error (less than 2 degrees) in our matching algorithm cascades into a large error at the corner of the building. The lack of features in the environment does not allow our scan-matching algorithm to recover.

Table 1. Comparison of RMS error and maximum error in meters between the four different algorithms we have tested.

Algorithm	RMS error	Maximum Error
SLAM (Karto 1.1)	0.3207 m	1.21 m
GraphSLAM	0.6335 m	1.59 m
SLAM (Karto 2.0)	0.3642 m	0.66 m
MCL (Karto 1.1)	0.1234 m	0.66 m

3. CONCLUSION

For the first time, we compared scientifically different localization algorithms for indoor and outdoor robots. The key was to have access to a quality dataset that allows an objective evaluation. We were able to compare our solution to a different implementation of the same algorithm or compare our solution to a completely different algorithm. The result is consistent with other papers, where graph-based SLAM outperforms any other solution in terms of precision.

Until recently, however, solving the nonlinear set of constraints required in a GraphSLAM was CPU intensive and was not compatible with real-time processing. In [7], we describe another method for solving this set of constraints 100 times faster than the previous method. As an added benefit, the solution is even more precise.

REFERENCES

[1] P. Moutarlier, R. Chatila, Stochastic multisensory data fusion for mobile robot location and environment modelling, in: Proc. 5th International Symposium on Robotics Research, 1989, pp. 85-89.

[2] Thrun, D., W. Burgard, and D. Fox. Probabilistic Robotics, MIT Press, 2009.

[3] Karto http://www.kartorobotics.com

[4] Bonarini, A., W. Burgard, G. Fontana, M. Matteucci, D. G. Sorrenti, and J. D. Tardos. RAWSEEDS: Robotics advancement through Web publishing of sensorial and elaborated extensive data sets, IROS'06 Workshop on Benchmarks in Robotics Research, Beijing, China.

[5] Lu, F. and E. Milios. Globally consistent range scan alignment for environment mapping, Autonomous Robots, 4:333–349, 1997.

[6] Doucet, A., N. deFreitas, and N. Gordan, editors. Sequential Monte Carlo Methods in Practice, Springer Verlag, 2001.

[7] Konolige, K., G. Grisetti, R. Kummerle, B. Limketkai, and R. Vincent. Sparse pose adjustment for 2D mapping, submitted to IROS 2010.

[8] Kummerle, R., B. Steder, C. Dornhege, M. Ruhnke, G. Grisetti, C. Stachniss, and A. Kleine. On measuring the accuracy of SLAM algorithms, Autonomous Robots, Vol. 27, No. 4, November 2009.

[9] Grisetti, G., C. Stachniss, S. Grzonka, and W. Burgard. A tree parameterization for efficiently computing maximum likelihood maps using gradient descent. Proc. of Robotics: Science and Systems (RSS), 2007.

[10] Michael Montemerlo, Sebastian Thrun: Simultaneous localization and mapping with unknown data association using fastSLAM. ICRA 2003: 1985-1991

[11] G. Grisetti, C. Stachniss, and W. Burgard. Improved techniques for grid mapping with rao-blackwellized particle filters. IEEE Transactions on Robotics, 23:34–46, 2007.