Charles University in Prague
Faculty of Mathematics and Physics

BACHELOR THESIS

Lukáš Jelínek

Graph Based SLAM on NDT Maps

Name of the department

Supervisor of the bachelor thesis:	Supervisor's Name
Study programme:	study programme
Study branch:	study branch

Prague 2015

I declare that I carried out this bachelor thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University in Prague has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In ..... date ..... signature of the author

Title: Graph Based SLAM on NDT Maps

Author: Lukáš Jelínek

Department: Name of the department

Supervisor: Supervisor's Name, department

Abstract: Abstract.

Keywords: key words

Dedication.

Contents

Introduction **3**

1 Used algorithms and key concepts **4**
1.1 SLAM problem definition . . . 4
1.1.1 SLAM categories . . . 4
1.1.2 Graph-based SLAM . . . 5
1.1.3 Pose graph creation . . . 5
1.1.4 Optimization . . . 6
1.2 Map representation . . . 7
1.2.1 NDT grid . . . 8
1.2.2 NDT-OM extension . . . 9
1.3 Scan registration . . . 10
1.3.1 Point to distribution NDT registration . . . 10
1.3.2 Distribution to distribution NDT registration . . . 11
1.3.3 Correlative scan registration . . . 12
1.4 NDT graph slam analyses . . . 13

2 System overview **14**
2.1 System architecture . . . 14
2.2 NDT frame . . . 15
2.3 (Robust D2D-NDT registration) . . . 16
2.3.1 Results . . . 16
2.4 Running window . . . 17
2.5 Loop closures . . . 18
2.6 Slam back-end . . . 19
2.7 Main algorithm . . . 20

3 Implementation and API **21**
3.1 Used libraries . . . 21
3.2 NDTGrid2D . . . 21
3.2.1 VoxelGrid . . . 21
3.3 Registration algorithms . . . 21
3.4 NDTCell . . . 21
3.5 Algorithm API . . . 21
3.6 Graph SLAM API . . . 21
3.7 ROS API . . . 21

4 Evaluation **22**
4.1 Datasets . . . 22
4.2 Results . . . 22

Conclusion **23**

Bibliography **24**

List of Figures **25**

List of Tables **26**

List of Abbreviations **27**

Attachments **28**

Introduction

Humanity has envisioned many tasks which could be carried out by robots including transportation, health care, save and rescue and many more. Robots of current world can efficiently operate only in very limited conditions. To solve problems of the future we need fast and reliable algorithms for our robots. Big question in field of mobile robotics is how to efficiently localize robot and create map as precise as possible. This problem is often referred to as simulations localization and mapping (SLAM) problem.

Good localization is crucial part of any good navigation software. Generated map plays important role in path planning and multi-robot coordination. SLAM algorithm should relay mostly on robots internal sensors like, e.g., sonars, cameras, wheel encoders. Using global localization systems(GPS) is only possible in outdoor environment. Precision of this localization is very often not good enough to successfully navigate robot.

In last decade many high quality SLAM algorithms where presented. Solution to full problem of map building and robot positioning is usually done by combining state-of-the-art approaches in related fields of study. This work combines normal distribution transforms (NDT) map building process and scan registration with well developed research branch of graph-based SLAM optimizing engines. To achieve this combination this work presents extended implementation of NDT mapping process and new way of robust scan-matching on NDT map. Implementation is done in robotic operating system (ROS) to make it easy to use.

This text is structured as follows: Key algorithms and concepts used in whole problem solution are explained in Section 1. In Section 2, is described process of scan matching, graph building, optimization and NDT mapping. Section 3 presents implementation details of this algorithm. Section 4 discuss mapping results and comparison to well known implementations in ROS.

An example citation: Anděl [2007]

1. Used algorithms and key concepts

This section offers basing introduction to multiple state-of-the-art algorithms used in this work. Good understanding of these concepts is crucial for full comprehension of section 3 . ref

1.1 SLAM problem definition

Successfully solving SLAM problem means to find location of robot in every time step and be able to create map at that time-stamp. In real world we deal with robot's sensors which have always some inherited noise. This means we are not able to fully say exact position of robot. This is main reason why to use probabilistic definition of problem. Robot moves through unknown space along trajectory expressed as variables $\mathbf{x}_{1:T} = \{\mathbf{x}_1, ..., \mathbf{x}_T\}$. While moving robot is taking odometry measurements $\mathbf{u}_{1:T} = \{\mathbf{u}_1, ..., \mathbf{u}_T\}$ and perception of environment $\mathbf{z}_{1:T} = \{\mathbf{z}_1, ..., \mathbf{z}_T\}$ Solving SLAM than means finding out probability of the robot's trajectory $\mathbf{x}_{1:T}$ and a map $\mathbf{m}$ of local environment given all the measurements and initial pose $\mathbf{x}_0$:

$$p(\mathbf{x}_{1:T}, \mathbf{m} \mid \mathbf{z}_{1:T}, \mathbf{u}_{1:T}, \mathbf{x}_0) \tag{1.1}$$

Odometry is usually acquired by robots wheel encoders or by incremental scan-matching. Odometry is represented in 2D by triple (x, y, θ) or by three dimensional transformation matrix. Perceptions of environment might come from different sources. In this work we expect distance measurements from laser scanner or kinect sensor. Initial pose can be interpreted as origin of coordinate system for global map. Representation of the map is more described in section 1.2.

1.1.1 SLAM categories

1.1.2 Graph-based SLAM

A graph-based SLAM solves SLAM problem by constructing graph representation of the problem. This graph is commonly called a pose graph. Nodes of the graph represent potential poses of robot at certain time stamp T. Therefore, nodes are representing our trajectory $\{\mathbf{x}_1, ..., \mathbf{x}_T\}$. Nodes also hold state of the current map. Edges in the graph represent possible spatial transformation between nodes. Edge is generated out of noisy sensor data. Therefore, we need to model uncertainty of this movement. It is represented by probability distribution and included in the edge. Generation of constrains is done by algorithm's front-end. It creates them either from odometry $\mathbf{u}_T$ or by measurement data $\mathbf{z}_T$ registration. Once the graph is completed ,it is optimized by algorithms back-end. Result of this process is the most likely position of all nodes in the graph.

1.1.3 Pose graph creation

Loop closure edge generation

1.1.4 Optimization

1.2 Map representation

Successful solving SLAM problem should output map of the unknown environment. This map needs to be stored for local path planning and obstacle avoidance. It is also needed for scan registration. Algorithms for avoiding obstacles very often need precise map. Map should keep low memory consumption, because robots very often have limited access to memory. Scan registration algorithms usually might benefit from maps with high precision.

Point-cloud is map representation which stores all measured points. This is very precise representation. All input data is still in its raw form. We are not loosing any information. Scan-matching algorithms e.g. ICP —— is working on top of this datastructure. It is very easy to convert from this model to different type of map if needed. Problematic is memory consumption. If robot runs for long period with higher frequency of data production, it is likely that robot will run out of memory.

citace

Occupancy map is grid based type. It consists of grid with cells. In every cell we have just one value describing probability that this cell is occupied. Value becomes higher with more incoming data measurements. It has constant memory consumption with respect to time of robot's run-time. It is possible to use this representation for map to map registration process. This model is also possible to represent empty spaces (low probability). This feature is used by many path planning and obstacle avoidance algorithms. That is why, occupancy maps are main output format for SLAM algorithms in ROS. It is important to select good resolution of grid. Finer grid offers better detail but higher memory consumption.

Quad-tree is a tree data structure. Each node of the tree has exactly four children. Nodes are decomposing space into sub-areas. Every node has its threshold. When it is reached, cell subdivides into four smaller cells. This process dynamically change resolution of the grid. This way we get higher precision in places where it matters more. It is crucial part of speeding up performance of ICP registration ———-.

citace

Normal distributions transform (NDT) representation uses grid based datastructure. Each cell has normal distribution parameters calculated based on inserted points. This model offers constant memory consumption over time. In comparison, NDT has better representation of inner points than octree (3D case of quad-tree). This was proven as convenient by . They have shown that coarser NDT grid can have similar results as finer octree map. NDT grids have their own class of registration whoch will be explained in next sections.

citace

picture of occupancy grid, point-cloud ndt grid

1.2.1 NDT grid

The normal distributions transform(NDT) grid representation was first time used by Biber and Strasser [2003] in their scan registration process. Central idea was to convert laser scan into grid with cells containing normal distributions. Points in space from laser scanner are first separated into corresponding cells. From points in sigle cell we approximate normal distribution (μ_i, P_i) by calculating mean and covariance:

$$\mu_i = \frac{1}{n}\sum_{k=1}^{n} x_k \tag{1.2}$$

$$P_i = \frac{1}{n-1}\sum_{k=1}^{n}(x_k - \mu_i)(x_k - \mu_i)^t \tag{1.3}$$

NDT grid was than used for registration.Originally proposed grid could be updated with new laser scans only by keeping used points and recalculating all cells again. This has changed with proposed recursive covariance update step by Saarinen et al. [2013a]. Their update step offers way how to fuse in new measurements. First it calculate normal distributions for added points. In second step, it merges old covariances with new one.

Consider two sets of measurement $\{x_i\}_{i=1}^{m}$ and $\{y_i\}_{i=1}^{n}$ than formula for mean calculation is in equation (1.4). Recursive update for covariance (RCU) is in equation (1.7)

$$\mu_{x\oplus y} = \frac{1}{m+n}(\sum_{i=1}^{m} x_i + \sum_{i=1}^{n} y_i) = \frac{1}{m+n}(m\mu_x + n\mu_y) \tag{1.4}$$

$$S_x = \sum_{i=1}^{m}(x_i - \mu_x)(x_i - \mu_x)^T \quad S_y = \sum_{i=1}^{m}(y_i - \mu_y)(y_i - \mu_y)^T \tag{1.5}$$

$$S_{x\oplus y} = S_x + S_y + \frac{mn}{m+n}(\mu_x - \mu_y)(\mu_x - \mu_y)^T \tag{1.6}$$

$$P_{x\oplus y} = \frac{1}{m+n-1}S_{x\oplus y} \tag{1.7}$$

Proof and further explanation for these equations can be found in work of Saarinen et al. [2013a].

In addition to fusing in new laser measurements we can also easily generated coarser grid by merging cells from higher resolution grid to grid with lower resolution. This mechanism is useful in path planning where we can plan on coarser grid which could be faster. Also, we can use multi-level scan matching approaches, which will be discussed in next section 1.3. Small disadvantage of this method is that we need to keep number of points used in every cell. It is worth noting that, this number can potentialy rise into infinity making distribution fusion numerically unstable. This problem will be resolved in next section.

1.2.2 NDT-OM extension

NDT grids offers good compromise between space and precision, but it lacks information about occupied space and unoccupied space. This is crucial for planning algorithms. This functionality was added to NDT by Saarinen et al. [2013a] and later improved by same authors in later work Saarinen et al. [2013b]. Every cell in NDT-OM is represented with parameters $c_i = \{\mu_i, p_i, N_i, p_i\}$, where μ_i and P_i are parameters of estimated normal distribution, N_i is number of points in cell and p_i is probability of the cell being occupied.

Calculation of occupancy parameter is done by ray-tracing. Consider that we have current map m_x. We have calculated new NDT map m_y from incoming distance measurements. Both maps needs to be in the same coordinate system. Ray-tracing starts at current robot position in map m_x and ends in cell of map m_x, where we will later merge in cell from m_y. Program visits every cell along the line and updates covariance based on equation **??**. When is ray-tracing over ref we merge in all cells from m_y into m_x with RCU update rule.

1.3 Scan registration

Scan registration is the esential part of SLAM front-end.

1.3.1 Point to distribution NDT registration

1.3.2 Distribution to distribution NDT registration

1.3.3 Correlative scan registration

1.4 NDT graph slam analyses

2. System overview

2.1 System architecture

2.2 NDT frame

2.3 (Robust D2D-NDT registration)

2.3.1 Results

2.4 Running window

2.5 Loop closures

2.6 Slam back-end

2.7 Main algorithm

3. Implementation and API

3.1 Used libraries

3.2 NDTGrid2D

3.2.1 VoxelGrid

3.3 Registration algorithms

3.4 NDTCell

3.5 Algorithm API

3.6 Graph SLAM API

3.7 ROS API

4. Evaluation

4.1 Datasets

4.2 Results

Conclusion

Bibliography

J. Anděl. *Základy matematické statistiky.* Druhé opravené vydání. Matfyzpress, Praha, 2007. ISBN 80-7378-001-1.

P. Biber and W. Strasser. The normal distributions transform: a new approach to laser scan matching. In *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 3, pages 2743–2748 vol.3, Oct 2003. doi: 10.1109/IROS.2003.1249285.

J. Saarinen, H. Andreasson, T. Stoyanov, J. Ala-Luhtala, and A. J. Lilienthal. Normal distributions transform occupancy maps: Application to large-scale online 3d mapping. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 2233–2238, May 2013a. doi: 10.1109/ICRA.2013.6630878.

J. Saarinen, T. Stoyanov, H. Andreasson, and A. J. Lilienthal. Fast 3d mapping in highly dynamic environments using normal distributions transform occupancy maps. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4694–4701, Nov 2013b. doi: 10.1109/IROS.2013.6697032.

List of Figures

List of Tables

List of Abbreviations

Attachments