

1 Structure

introduction motivation state of the art systems system design and architecture

background, what is system, what features, how does it works data processing slam and graph minimization factor graphs, distance function, graph matching trajectory generation walking model

map construction observations to map (approximation, regression), map to observations (as a sensor) (sensor model), propagation model, slam / sam conditioning on magnetic field data what graphs / figures can be plotted

2 Background and problem statement

The topic of the project is related to the problem of indoor navigation. The context of the problem states that no GPS data are available at hand, which makes the use of usual navigation services impossible.

Most existing systems of indoor navigation require special mapping stage. Novel indoor navigation systems utilize the data recorded from users. This is what is called the crowd-source approach.

For real-time indoor navigation, in conditions where is no initial data is available, we have to pass stages of localization and mapping, which can be done simultaneously in SLAM approach. We aim to combine both SLAM and crowd-source approach for the best performance of positioning system.

The innovation of this research is in ability to provide same navigation services with less information and in more natural way, which means also the reduced cost of the system overall.

3 Objectives

We perform this research to create an indoor positioning system with special features. The objective of this research is to develop all algorithms needed to obtain these features.

We define an approach for dealing with the dead reckoning system. We propose a mapping algorithm using ARcore visual odometry for training and all-time pedestrian dead reckoning and magnetic mapping for operation.

We aim to develop the system, working without complicated sensors or physical landmarks as beacons. This is only a research interest because the current SOTA PDR systems are only secondary to other approaches.

The problem we solve and the scope can be formulated as the following: design the algorithm of collection and processing the data for mapping indoor

location for the indoor localization application. The algorithm should be solvable, by means that it always converge if sufficient data is given.

We can write the criteria for the positioning system we develop.

Criteria for the proposed system:

1. no prior map is available: the system can work as SLAM system (real-time navigation with no prior map)
2. no special hardware for operation except smartphones: positioning accuracy enough for operation (1-2m is the usual accuracy in this conditions)
3. the system aggregate data from many sources (crowd-source) and improves the localization accuracy

We formulate several hypotheses we evaluate during research:

Hypotheses: The technology of magnetic field navigation can be implemented and fine-tuned for indoor crowd-source SLAM

The data from magnetic field and inertial sensors is enough for running SLAM

The crowd-source system satisfies the system optimality conditions and improve the accuracy

The hypothesis 2 was not proved in existing systems. For most system, additional prior knowledge is needed. This is more the scientific interest to prove this hypothesis.

Two other hypotheses are more the engineering questions. We have to compare performance and robustness of our algorithm and systems to other state of the art approaches. However for our problem statement, there is no much systems that have achieved any reasonable accuracy (1-2m). So we aim to achieve the accuracy that can be compared to other methods only.

4 Introduction

4.1 Related work

We review the field of human indoor localization. We focus on crowd-source mapping, or mapping with limited sensor model (limited to existing infrastructure in building space and to sensors in human smartphones). In this field there are many approaches so solve the sub-problems or parts of given problem.

We know paper GraphSLAM based Crowd sourcing framework for indoor Wi-Fi fingerprinting [1]. The approach of GraphSLAM is promising. We want to utilize more available spatial information.

The most promising results in terms of cheap sensor spatial information are magnetic maps. For magnetic fingerprinting there are well developed approaches.

One of well-known papers in magnetic fingerprinting is [2]. The authors present an approach for magnetic field mapping, in terms of not fingerprints, but a full grid mapping procedure. As a result of this mapping procedure we obtain a full field image that is easy to work with. The problem with this approach that we have to collect full grid measurements, which is a time consuming procedure. We want to get magnetic field map without additional mapping procedure, this approach is called crowd-source mapping. We collect the data from a set of human travel trajectories. Then we merge them in single noised image (only analogy representation for better understanding), and apply optimization procedure. We require the resulting image be smooth, so the optimization constraints must be set to satisfy the image denoising procedure. This is just our idea and assumption that is have to be proved in further research. We see some similarities here with [3].

However they are highly dependent(?? prove statement) on the data collection procedure. This algorithms also are not intended for standalone usage in terms of sensors fusion. E.g. requires either special mapping procedures or additional hardware devices - beacons.

There are few papers on magnetic and inertial based navigation. However they were not implemented in SLAM frameworks such as [1]. Some of frameworks are a part of commercial interest and are not open source.

The interesting map construction algorithm were proposed in [3].

The full system consists of three subsystems, i.e. Dead Reckoning Subsystem (DRS), Map Construction Subsystem (MCS), and Localization and Navigation Subsystem (LNS).

Our intuition gives that Dead Reckoning Subsystem performs sensor fusion and smoothing. The localization and navigation subsystem utilizes the existing map and the output of smoothed sensors measurements.

The map construction process is an optimization problem with some constraints.

In [3], authors propose universal framework with no prior information of building map structure. But for real situation, the map of the building is known. We can define an indicator function similar to SLAM occupied cells mapping (????) With indicator function defined, we may fit the graph of trajectories to the building inner structure graph - topological map.

The similar approach with map constraints correction and re-sampling

was shown in [4]. The paper utilizes a particle filter that combines PDR and RSSI data.

5 Methodology / theoretical framework

Graph matching

Our case: formulate situation with no wifi available data in local area
(A specific case of this problem) The graph matching problem is the “maximum weighted bipartite matching”,

which is defined as a matching on a bipartite graph with maximum sum of the weights of selected edges.

This problem is also known as the “assignment problem”. We utilize factor Graphs for such assignment of new data.

Matching problem proposed:

We now describe the problem formulation by modelling two graphs: the ground truth graph (location based information - building map) and the data graph, constructed during the online training phase of system from the crowd-sourced data of users walking in the environment.

We do not obtain a radio map which is needed for RSS-based localization. Instead, we collect a data-set of magnetic field fingerprints, tagged with their relative physical coordinates to previous position. This relative coordinates (graph type trajectory with approximate information on edges lengths).

From the data of two graphs: location map and fingerprints collection, we perform matching procedure, using multiple available methods.

The first procedure to apply is accept-reject method: all points in restricted location are blocked (person can’t go through walls and etc.). Secondly, we perform loop closure and data association using common algorithms:

probabilistic approach (hidden Markov models)

Similarity measures:

What data we obtain in the data graph: heading, relative position, magnetic field direction. For multiple locations in same domain there can be lots of point with same of similar magnetic field direction. Instead, between any two points, there can be enough magnetic field disturbances, which will create enough information for distinguishing data, and mapping only location related data.

We can measure the similarity only between long enough tracklets(parts of trajectory e.g. frames) / edges.

The signal similarity measure cab be just a cross-covariance

5.1 System architecture

MCS MCS first partitions the trajectories into segments, and then clusters them into a cluster set C . For each cluster in C , with trajectory matching procedure, we construct the graph by connecting the crossing trajectories. We estimate the boundaries of the graph (trajectories closest to the walls), its width (adjust to corridor dimension) and location of each trajectory given the position of trajectory in graph and graph absolute position. We store the walls relative position to the processed graph in additional structure W .

From generated graph data, we extract / resample the representative trajectory, or field approximation. The representative trajectory set is denoted by R . This trajectory set has no relation to trajectories and input data, but only to the magnetic field in the given location. The output of MCS is the constructed digital map characterized by X , R , and W .

LNS LNS performs particle-filter-based localization and routing-like navigation easily without any human intervention. The particle filter (PF) for localization utilizes the physical constraint that one cannot walk through the walls in estimating the posterior probability distribution of the user's location.

6 measurements model

acceleration data from IMU
magnetic field orientation vector
RSSI

7 Transition model definition

We consider the process is Markovian. Copied all from [4]:

With the development of microelectromechanical systems(MEMS), a few MEMS-based sensors have been built and incorporated into smartphones: accelerometers, gyroscopes, magnetometers, etc. These sensors can be used to provide information on the user's actions. Pedestrian dead reckon-ing (PDR) [10] is a relative navigation technique that uses these sensors.

we propose a PDR-based indoor positioning method that integrates RSSI with indoor environment map constraints by using particle filters.

PDR-based indoor positioning can be expressed by the following equations

$$X_k = \begin{bmatrix} x_{t-1} \\ y_{t-1} \end{bmatrix} = \begin{bmatrix} x_{t-1} & +l_k \sin(\theta_t) \\ y_{t-1} & +l_k \cos(\theta_t) \end{bmatrix} + \begin{bmatrix} \delta_x \\ \delta_y \end{bmatrix} \quad (1)$$

The orientation updates is independent from X_k updates, depends on accelerometer and gyro, same model.

Step Detection: Android mobile phones have two kinds of sensors to monitor steps: step counters and step detectors [25].

Peak detection is used in this paper for step detection

nonlinear step length estimation model based on statistics proposed by Weinberg [27]

$$s = k \sqrt[4]{a_{zmax} - a_{zmin}} \quad (2)$$

The Android system computes orientation angles by using the device's geomagnetic sensors in combination with its accelerometers [25]. Using these two hardware sensors, the system provides data for the following three orientation angles

The device is NOT assumed to be pointing in the heading direction. Need to apply additional space transformation.

7.1 RSSI methode

Usual path-loss model:

$$PL(d) = PL(d_0) + 10\alpha \log d/d_0 + \omega = PL(d_0) + 10\alpha \log d/d_0 + N(0, \sigma_\omega^2) \quad (3)$$

where d represents the Euclidean distance between the anchor node and the receiver, d_0 represents a specified distance, $PL(d)$ and $PL(d_0)$ represent the RSSI at d and d_0 , respectively (in dBm), and α represents the path loss exponent, which is closely related to the ambient environment; ω is a zero-mean Gaussian distribution variable with variance σ_ω^2 .

The inverse function:

$$d_i = d_0 \cdot 10^{\frac{PL(d_i) - PL(d_0)}{10\alpha}} \quad (4)$$

Landmark model:

$$f_i(x, y) = d_i - \sqrt{(x - x_i)^2 + (y - y_i)^2} \quad (5)$$

$$\min(x, y) = \min \sum_{i=1}^m [f_i(x, y)]^2, m \geq 3. \quad (6)$$

In paper author transform this to linear system,

$$AZ = b \quad (7)$$

$$A = \begin{bmatrix} 1 - 2x_1 - 2y_1 & \\ & \dots \\ 1 - 2x_m - 2y_m & \end{bmatrix} \quad (8)$$

$$Z = \begin{bmatrix} x^2 + y^2 \\ x \\ y \end{bmatrix} \quad (9)$$

$$b = \begin{bmatrix} d_1^2 - x_1^2 - y_1^2 \\ \dots \\ \dots \end{bmatrix} \quad (10)$$

We have to check it

Good alternative was proposed in [1] to bound region by min and max RSSI according to its variance - linear system.

7.2 Processing

Update step, prediction, correction State Estimation

Resample: In this step, importance resampling [20] is used to obtain a new particle set

some papers prove importance resampling can't be applied for this task because pf noise.

7.3 RSSI Graph SLAM

Definition taken from [1].

Trajectory Modelling with GraphSLAM [5] is a classical framework for SLAM optimization.

GraphSLAM extends the traditional SLAM framework by considering the poses in a trajectory as nodes and raw measurements between poses and landmarks as edges in a graph. Note that each edge is attached with a probability distribution over the relative positions of its two vertexes since inherent noise in sensors need to be considered. In 2-D case, a pose x_t consists of a 2-D coordinates (x_1, t, x_2, t) and an yaw angle θ_t .

$$e_{ij}(x_i, x_j) = z_{ij} - \hat{z}_{ij}(x_i, x_j) \quad (11)$$

The item ω_{ij} in the equation is the information matrix representing Gaussian noise in sensor measurement. The GraphSLAM problem is now reduced

to a constrained least square problem and it can be solved by some standard optimization techniques, like Gaussian-Newton or Levenberg-Marquardt algorithms, as proposed in [16]. Either of the two methods is based on local iterative linearization:

Then, the goal of GraphSLAM is to find an configuration of poses x^* that minimizes the squared error $F(x)$ of all observations given a set of constraint edges C :

$$F(X) = \sum_{(i,j) \in C} e_{ij}^T W_{ij} e_{ij} \quad (12)$$

$$x^* = \operatorname{argmin}_x F(x) \quad (13)$$

8 factor graphs, distance function, graph matching

We define a distance function as a space measure/metric to construct a formal space as the Cartesian product of spaces. We construct it in a way of traditional Borel set definition.

For any collection $E_{1..i}$ of a countable (possibly finite) number of pairwise disjoint, measurable sets, let E denote their union. The measure μ for the iteration of the whole collection must satisfy:

$$\mu(E) = \sum_i \mu(E_i)$$

If we given the space of observations (z, ρ_z) and the space of states (x, ρ_x) . And let $(z, \rho_z), (x, \rho_x)$ be two metric spaces.

We may define a composition distance function with the Cartesian product of given spaces:

$$\rho_{x,z}(X_1, X_2) = c_1 \rho_x(x_1, x_2) + c_2 \rho_z(z_1, z_2) \quad (14)$$

The traditional restriction is for the positive weights.

We plan use distances as the likelihood function, so here we do not really care about distances itself on a large scale, only as the similarity measure.

The ρ_x is the traditional L2 distance, plus added angle delta function. We refer to example from Lavalle book, *SO(2) Metric by Comparing Angles*:

$$\rho(\theta_1, \theta_2) = \min\{|\theta_1 - \theta_2|, 2\pi - |\theta_1 - \theta_2|\} \quad (15)$$

Or better in the continuous form as the inverse cosine of the complex number representation of angles $a + bi$:

$$\rho(\theta_1, \theta_2) = \rho(a_1, b_1, a_2, b_2) = \cos^{-1}\{a_1 a_2 + b_1 b_2\} \quad (16)$$

for two points (a_1, b_1) and (a_2, b_2)

The alternative here is using the SE3 metric or quaternion based representation, but we don't consider them here for simplicity.

8.1 SAM for magnetic localization problem

$$P(X, M, Z, U) = p(x_0) \prod_{i=1}^M p(x_i | x_{i-1}, \mu_i) \cdot \prod_{k=1}^K p(z_k | x_{i_k}) \quad (17)$$

$$p(z_k | x_{i_k}) = \log(z_k - h(x_{i_k})) \quad (18)$$

Objective: maximize the joint probability using graph theory and linear Algebra.

SAM problem is represented as a factor graph - bipartite graph with weighted nodes and edges.

$$\theta = \{X, M\}, \quad (19)$$

$$p(\theta) = \prod \phi_i(\theta_i) \cdot \prod \xi_{ij}(\theta_i \theta_j) \quad (20)$$

$$\phi_0(x_0) \propto p(x_0) \quad (21)$$

$$\phi_{i-1,i}(x_{i-1}, x_i) \propto p(x_i | x_{i-1}, u_i) \quad (22)$$

$$\xi_{ij}(\theta_i \theta_j) \propto p(z_k | x_{i_k}, M(x_{i_k})) \quad (23)$$

Transition model

$$p(x_i | x_{i-1}, u_i) = N(x_i; g_i(x_{i-1}, u_i), \Sigma_{u_i}) \quad (24)$$

The covariance matrix Σ_{u_i} is constant here.

The function $g()$ is defined by the propagation model. We have the traditional pedestrian dead reckoning problem statement. The inertial motion model is the following:

Observation model

$$p(z_k | x_{i_k}) = N(z_k; h_k(x_{i_k}, m_{j_k}), \Sigma_k) \quad (25)$$

$$m_{j_k} = M(x_{j_k}) \quad (26)$$

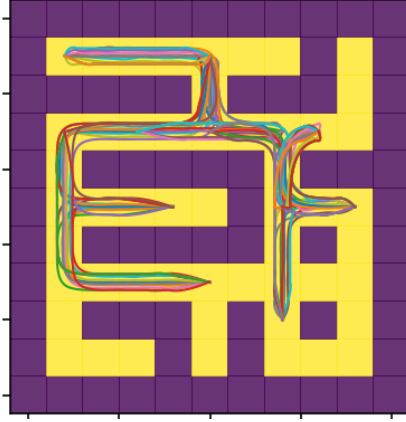


Figure 1: Random trajectories on a generated building map.

8.2 Clustering application for mapping

The optimization task: global minimization, matrix reduce,

States are only locally dependent, thus the task can be approximately solved for local clusters. The single task can be parallelized and all minimization problems can be formulated and solved independently for sub-locations.

We approximate the space, so we have to split a continuous space in a decent manner. The good way would be to use a constant area discrepancy trick as in regular random sampling methods. In Lavalle book there are decent examples, but for simplicity we stay with a regular grid and RBF functions.

The grid and a measure function together define a cluster shape and specifics. We can use rbf or either an ellipse function (sum of distances to two points).

For each point in coordinates x - y of state space, we assign cluster $c(x, y, m)$

Clusters are an additional variable in state, linkages in lookup tables.

We are not restricted to lattices, because we allow clusters to overlap a bit. Thus the optimization problem will be solved twice or more time for overlapping vertices or states, which will guarantee smoothness in resultant solution.

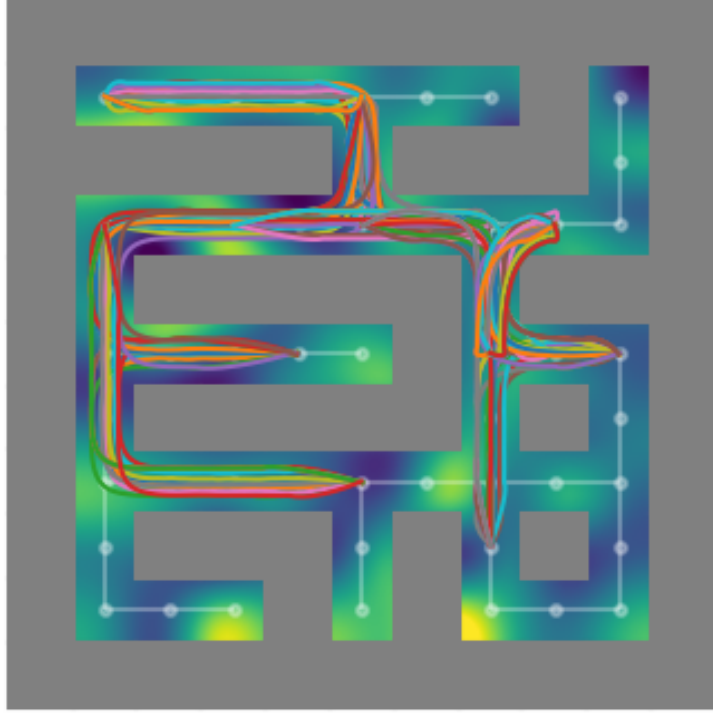


Figure 2: The background simulates continuous random field we want to reconstruct

9 Experiments

9.1 Pose estimation and IMU Modelling

We generate the series of data using maze generation algorithm. On a fixed map of corridors we generate routes and trajectories, then we add noise to simulate the walking human.

9.2 Data processing

First we focus on a key part of a given research. The main part of this paper requires the transformation of observation to the solid continuous map.

After we highlighted the map structure and features, we can say that we need to reconstruct the magnetic field.

A magnetic field is continuous potential field with certain known features. Each point has its own field direction and magnitude. We can encode the field as a number of three dimensional vectors or as 3 images, representing the field magnitude in each axis direction.

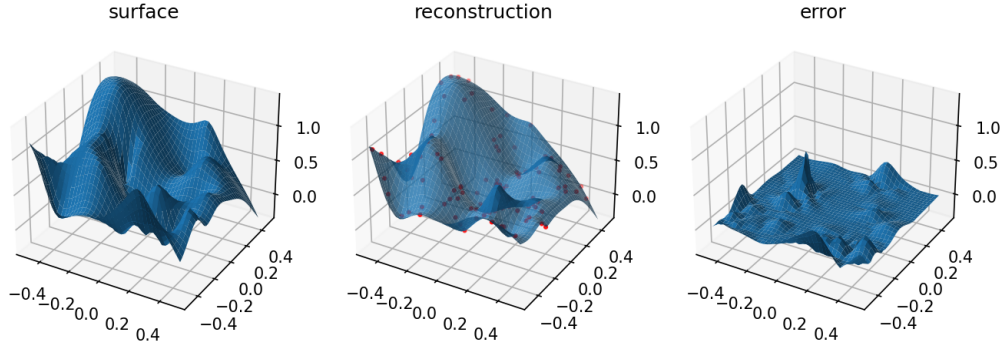


Figure 3

First we model the field construction and reconstruction.

9.2.1 image approximation

We generate a surface from array with random noise values as a mesh and apply interpolation and smoothing. On the random surface we sample a series of points. The problem is to accurately reconstruct the initial surface from samples.

There are several possible solutions for such problem. The most promising and accurate seems kernel training as in Kernel Ridge Regression or other similar methods. From the other side, there are other more straightforward solutions.

Solution 1: interpolation. Solution is not accurate and not feasible because resulting surface is very noisy. We need a solution to deal with many observations and noisy data.

The promising solution was shown in repository of [6]. We aim to solve the problem in similar problem statement with more traditional methods.

We perform the noisy surface approximation using RBF function from the scipy package.

On the figure ??, the right plot represents the distance between initial and approximated surfaces.

[6]

References

- [1] Min Zhang, Ling Pei, and Xiaotie Deng. Graphslam-based crowdsourcing framework for indoor wi-fi fingerprinting. In *2016 Fourth International*

- Conference on Ubiquitous Positioning, Indoor Navigation and Location Based Services (UPINLBS)*, pages 61–67, 2016.
- [2] E. Grand and S. Thrun. 3-axis magnetic field mapping and fusion for indoor localization. *2012 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 358–364, 2012.
 - [3] X. Zhang, Y. Jin, H. Tan, and W. Soh. Cimloc: A crowdsourcing indoor digital map construction system for localization. In *2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, pages 1–6, 2014.
 - [4] Hao Xia, Jinbo Zuo, Shuo Liu, and Yanyou Qiao. Indoor localization on smartphones using built-in sensors and map constraints. *IEEE Transactions on Instrumentation and Measurement*, PP:1–10, 08 2018.
 - [5] Brian Ferris, Dieter Fox, and Neil Lawrence. Wifi-slam using gaussian process latent variable models. volume 7, pages 2480–2485, 01 2007.
 - [6] Surface approximation gp.