Large-scale loop-closing by fusing range data and aerial image

C. Chen and H. Wang*

INRIA Rhône-Alpes
Montbonnot, 38334 Saint Ismier Cedex, France
Email: chen.cheng@inrialpes.fr

School of EEE, Nanyang Technological University,
50 Nanyang Avenue, Singapore 639798
Email: hw@ntu.edu.sg

Abstract

This paper presents a mapping method that can accurately map large environment with one single robot by visiting the environment for only once, and the resulting map can provide thorough 3D description for the environment in a predefined global coordinate system. Our first contribution is to represent the map as a collection of submaps arranged in a deformable configuration, and to perform loop-closing by registering this submap configuration to an aerial image. The second contribution is to introduce the active contour technique to the SLAM domain, so that the registration is efficiently solved in an iterative energy minimization process. The constraints from robot mapping are modeled as forces trying to keep the submaps consistent to each other, while the pictorial matching is represented by forces guiding submaps to a configuration consistent with the aerial image. In the experiment, we demonstrate the proposed algorithm’s capability to close a 1,890 meters loop with only one visiting. The result is compared with ground truth, and high accuracy is observed.
1. Introduction

Going through the same environment again and again till the map converges may be very time-consuming for one single robot in the large-scale outdoor navigation tasks. Actually this motivated the development of collaborative mapping using multiple robots, as in [1]. But it is often desirable to reduce the operational time and number of robot whenever possible. On the other hand, even if above converged map could be finally obtained, it only provides the relative locations among map items. Although such relative map is already enough for a lot of path planning tasks, it may not be able to support complex outdoor navigation tasks, such as surface-air collaborations, human-robot interactions, multi-sensor multi-platform tracking, in which exchanges of information are often preferred to be performed in a global coordinate system.

In this work, the loop-closing is modeled as the process of registering the mapping result to pictorial information (e.g., aerial photos or satellite images) captured from a global perspective, and solved as a general energy minimization problem. By exploiting the pictorial information, the presented algorithm can accurately map large environment with only one robot, by visiting the environment for only once. In the meanwhile, the resulting map provides thorough 3D information about the environment in a predefined global coordinate system.

The major contribution of this paper is to introduce the global pictorial information to SLAM as a constraint on the loop-closing optimization process. The global pictorial information describes how the mapped environment should look like from a global perspective. Placed into the mapping context, it attracts submaps to a configuration which best matches the input image. The process is named loop-closing with pictorial matching (shorted as LCPM in this context) in this context.

In this paper, the LCPM is formulated in a probabilistic manner which is consistent with the generic stochastic SLAM algorithms. The goal of LCPM is to compute the maximum a posteriori (MAP) estimate
for the submap configuration based on both mapping information $\Phi$ and pictorial information $I$: $X^* = \arg\max_X (X|\Phi, I)$

The second contribution of this work is to introduce the active contour technique \cite{2} to the SLAM domain, to guide the submap configuration to its desired position. The MAP estimation is therefore solved in an iterative energy minimization process. A potential field is constructed in the domain of the input aerial image. The two constraints in the LCPM optimization process are represented as two forces in the image domain. These two forces deform the submap configuration in every iteration until a balance is achieved.

This paper is organized as following, in the next section, some related SLAM techniques will be reviewed; then in Section 3, we will lay out the probabilistic formulation of LCPM; the iterative solution for the LCPM optimization will be elaborated in Section 4; finally, the experimental results and performance analysis will be given.

2. Background

In the SLAM process \cite{3}, while sensor fusion process iterates, the error in raw sensor measurements and sensor data approximation will accumulate. The error contained in the raw data is especially difficult to handle because it always exists even we could obtain perfect motion model (with no linearization error) and sensor data model. Such error is inherent to the SLAM problem and not caused by any specific modelling or approximation. The linearization error in robot motion modelling also plays an important role in SLAM. The linearization error is determined by the difference between the estimate and the one used to compute the Jacobians. This difference could become very large when closing a large loop.

Due to above errors, when the robot re-visits a place and tries to close a loop, there always exists inconsistency between the new and old maps. To accurately and efficiently perform loop-closing for large environment is widely regarded as a major problem in SLAM. In the left of this section, some most relevant approaches will be reviewed.
The Kalman filter based approach in [4] presents a generic and elegant approach to handle errors in the SLAM problems. However, the Kalman filter and its derivatives basically cannot scale well to large SLAM problem, because it explicitly represents the correlation between each pair of map items. Montemerlo [5] developed FastSLAM which approximates the probability distribution of map using Monte Carlo sampling. However, this algorithm’s high efficiency depends critically on the number of particles involved.

By representing the relative information among map items, the map can be represented by a graph-like model, such a strategy has gained tremendous popularity during the past few years. Lu [6] proposed the Consistent Pose Estimation (CPE) that directly builds a sparse linear equation system from measured relative position between adjacent robot poses. Thrun [7] developed another graph model based SLAM algorithm called Sparse Extended Information Filter (SEIF), in which the uncertainties of SLAM is stored in the information matrix. SEIF is efficient to incorporate a new landmark in the information matrix. However, it could be nevertheless time consuming to obtain a useful map representation when this information matrix is projected back to its covariance form. A different implementation based on graph models is the thin junction tree filter SLAM (TJTF) proposed by Paskin [8]. TJTF represents the filtered information by a Gaussian graphical tree model, as the tree model grows when the robot navigates, new edges are added to the tree. An efficient approximation is periodically performed by maximum likelihood projections. A similar but independently developed tree-structure-based graphical model is the treemap algorithm proposed by Frese [9]. Relaxation [10] provides yet another efficient way to analyze the graphical model. Recently, Frese [11] further developed multi-level relaxation and demonstrated highly efficiency.

Dividing the whole map into submaps is yet another way to perform robot mapping. Bosse [12] proposed the Atlas framework which segments the map into a graph of interconnected submaps, there is no global coordinate in the Atlas, so that the deployments of submaps are all relative. Another technique called ‘map joining’ was independently developed by Tardós et.al. [13], in which how the submaps are jointed is modeled in a relative manner similar to the graphical model discussed previously. Estrada [14] further introduced the hierarchy SLAM which imposes the consistency at the global level. This is achieved by
ignoring the correlations between map items from different submaps.

All above techniques infer the map solely based on the measurements of on-board perceptual sensors and odometry, so they can be generally regarded as ML estimators. A comprehensive discussion on solving this ML estimation problem can be found in [15]. During the past few years, new techniques have also been developed to fuse global a priori information [16] [17]. These techniques incorporate a strong constraint on the local data fusion process. Such constraints can be regarded as onboard sensors, whose errors are bounded over time and distance. However, these constraints often take advantage of the heuristic that the robot only travels in certain pre-defined places, e.g. roads.

3. Probabilistic Loop-closing with Pictorial Matching

3.1 Submap Configuration

As elaborated in [14], when the robot navigates and segments the whole environment map, the submaps, the transformations between submaps and the uncertainties of these transformations can be estimated. In this context, they are referred as initial submap configuration, or shorted as mapping information. Mathematically, at the time of loop-closing \( l \), given the history of perceptual data \( Z_l \), and interior motion readings \( U_l \), the initial submap configuration \( \Phi \) can be constructed from a SLAM routing \( \Lambda \):

\[
\Phi = \{\bar{\mathcal{X}}, \mathcal{P}, \mathcal{M}\} = \Lambda(Z_l, U_l),
\]

where \( \Phi \) comprises submap transformations \( \bar{\mathcal{X}} \), transformation covariances \( \mathcal{P} \), and submaps themselves \( \mathcal{M} \). For instance, \( \mathcal{X} = (\cdots \bar{x}_s \cdots) \), here \( s = 1, 2, \cdots, N_m \) and \( N_m \) is the total number of submaps. Each submap \( M_s \) is represented as a 3D point cloud \( \{x_{s,j}, y_{s,j}, z_{s,j}\} \), while \( j = 1, 2, \cdots, N_s \), here \( N_s \) is the number of points inside submap \( s \).

These submaps and their deployments represent the information collected during the mapping process, based on which the a posteriori map configuration \( \mathcal{X} \) can be estimated. In the mapping context, only the transformation between adjacent submaps is the target variable to be estimated, so in this paper, \( \mathcal{X} \), and all its inside items, are marked with bar. As depicted in Fig. 1, at the time when a loop-closing is detected, this configuration resembles a circle, and each arc \( s - 1 \rightarrow s \) represents a known transformation.
Fig. 1. (a) The initial submap configuration. The deployment of all the 9 submaps is defined by the sequence of coordinate transformations between submaps, which is denoted as $\bar{X}$. When a loop-closing situation is confirmed (as indicated), the submap $M_9$ should be shifted to the position of submap $M_1$. (b) The afterriori submap configuration corresponding to (a). The deployment of all the 8 submaps can be defined by either the sequence of relative transformations $X$, or equivalently, the sequence of absolute global positions $L$. From submap $M_{s-1}$ to $M_s$. This arc is denoted as $\bar{x}_s$, which is a 3-vector. These three items represent the translation of a submap relative to the coordinate of its adjacent submap. The estimated transformations between submaps come with uncertainties. Here they are approximated to be Gaussian, and represented by a covariance matrix $P$. It can be noticed that a resulting submap $M_s$ is independent of any previous ones, because it is built relative to the vehicle location at the moment it was built. Therefore, any two submaps within $\mathcal{M}$ are statistically uncorrelated.

Similar to the definition of $\bar{X}$, the afterriori map estimate can be denoted as the collection of transformations between local submaps, $\mathcal{X} = (\cdots x(s) \cdots)$.

The pose of submap $s$ is denoted as $L(s)$, which comprises the submap’s origin $\{n(s), e(s)\}$ and direction $\tau(s)$ in the fixed north-east global coordinate frame. Therefore, the mapping information can also be formulated in an absolute form: $\mathcal{X} \equiv \mathcal{L} = (\cdots L(s) \cdots)$. A pair of operators $\oplus$ and $\ominus$ are introduced here to represent the coordinate transformations in the Euclidean space. Therefore, by defining $L(0)$ as the origin of the fixed global coordinate system, there exist following relations between submap
locations $L$ and submap transformations $x$: $L(s) = L(0) \oplus x(1) \oplus x(2) \oplus x(3) \ldots \oplus x(s)$, and respectively, the pose of submap $s$ with respect to $s-1$ is calculated as $x(s) = \oplus L(s-1) \oplus L(s)$.

### 3.2 Probabilistic Loop-Closing With Pictorial Matching

Based on above mapping information, the task of LCPM is to calculate the *a posteriori* probability density function (PDF) of the submap configuration $\mathcal{X}$, with additional pictorial information $I$. The target probability to be estimated is therefore denoted as $p(\mathcal{X}|\bar{\mathcal{X}}, \mathcal{M}, \mathcal{P}, I)$ According to the Bayes rule, this target probability can be marginalized as:

$$p(\mathcal{X}|\bar{\mathcal{X}}, I, \mathcal{M}, \mathcal{P}) \propto p(I|\mathcal{X}, \bar{\mathcal{X}}, \mathcal{M}, \mathcal{P})p(\mathcal{X}|\bar{\mathcal{X}}, \mathcal{M}, \mathcal{P})$$

(1)

Apparently, the image appearance is only determined by the status of the real world, rather than how it appears in the robot’s sensor observation, so the $\bar{\mathcal{X}}$ in the first component on the right-side of (1) can be omitted.

Taken the fact that observing $I$ is only dependent on the layout of the submaps, the $\mathcal{P}$ in $p(I|\mathcal{X}, \mathcal{M}, \mathcal{P})$ on the right side can be removed. Similarly, the prior probability of $\mathcal{X}$ is independent of the details of submaps, so the $\mathcal{M}$ in the second item $p(\mathcal{X}|\bar{\mathcal{X}}, \mathcal{M}, \mathcal{P})$ is omitted too. The target probability is thus refined as:

$$p(\mathcal{X}|\bar{\mathcal{X}}, I, \mathcal{M}, \mathcal{P}) \propto p(I|\mathcal{X}, \bar{\mathcal{X}}, \mathcal{M})p(\mathcal{X}|\bar{\mathcal{X}}, \mathcal{P})$$

(2)

The probability $p(I|\mathcal{X}, \mathcal{M})$ captures the imaging process, and measures the likelihood of observing pictorial information $I$ given the map configuration. It is not difficult to understand that, one submap’s appearance will not affect the appearances of others. So this probability can be decomposed into:

$$p(I|\mathcal{X}, \mathcal{M}) \propto \prod_{s=1}^{N_m} p(I|L_s, M_s)$$

(3)

where each $p(I|L_s, M_s)$ characterizes the match between submap $s$ and the image. The distribution $p(\mathcal{X}|\bar{\mathcal{X}}, \mathcal{P})$ models the conventional loop-closing optimization process. Its constraint on the whole submap
graph can be decomposed into constraints on each individual transformation:

\[ p(\mathcal{X}|\bar{\mathcal{X}}, \mathcal{P}) = \prod_{s=1}^{N_m} p(x(s)|\bar{x}_s, P_s) \propto \prod_{s=1}^{N_m} \exp \left( -\frac{1}{2} (x(s) - \bar{x}_s)^T P_s^{-1} (x(s) - \bar{x}_s) \right) \tag{4} \]

By substituting (3) and (4) into (2), we can reach the final formulation of the posterior probability distribution of the submap graph \( \mathcal{X} \):

\[ p(\mathcal{X}|I, \bar{\mathcal{X}}, \mathcal{M}, \mathcal{P}) \propto \prod_{s=1}^{N_m} p(I|L(s), M_s) \prod_{s=1}^{N_m} \exp \left( -\frac{1}{2} (x(s) - \bar{x}_s)^T P_s^{-1} (x(s) - \bar{x}_s) \right) \tag{5} \]

Taken the objective of LCPM, the MAP loop-closing estimation is detailed as:

\[ \mathcal{X}^* = \arg \max_{\mathcal{X}} \prod_{s=1}^{N_m} p(I|L(s), M_s) \prod_{s=1}^{N_m} \exp \left( -\frac{1}{2} (x(s) - \bar{x}_s)^T P_s^{-1} (x(s) - \bar{x}_s) \right) \tag{6} \]

3.3 An Energy-Minimization Formulation

Taking the negative logarithm of the right side of (6) yields:

\[ \mathcal{X}^* = \arg \min_{\mathcal{X}} \epsilon \times \sum_{s=1}^{N_m} f \left( L(s), M_s \right) + \sum_{s=1}^{N_m} (x(s) - \bar{x}_s)^T P_s^{-1} (x(s) - \bar{x}_s) \tag{7} \]

where \( \epsilon \times f \left( L(s), M_s \right) = -\log p(I|L(s), M_s) \) is a match cost measuring how well submap \( M_s \) matches the image \( I \) at the pose \( L(s) \), and \( \epsilon \) is a weight determining how much the mapping information and pictorial matching contribute in the minimization process. So the objective of LCPM is to minimize the target energy function \( E^{LC} \), where \( E^{LC}(L) = \epsilon \times E^{Ext}(L) + E^{M}(L) \).

In this energy model, \( E^{M}(L) \) characterizes the cost of deforming from initial submap configuration; while \( E^{Ext}(L) \) gives penalty to \( L \) that does not match the pictorial information.

4. SNAKE-based Energy Minimization

In this work, a greedy searching strategy based on SNAKE [2] is developed to solve the LCPM problem. The basic idea is to sequentially adjust the pose for each single submap based on the available local pictorial information and mapping information, and iterate this procedure until convergence. With respect to the LCPM in (7), the configuration of submaps that minimizes \( E^{LC} \) must satisfy the Euler equation.
\[ \epsilon \times \nabla E^\text{Ext}(L) - \nabla E^M(L) = 0. \]

To solve this equation, submap configuration is made dynamic by treating \( X \) as a function of iteration \( t \), as well as \( s \). Then the configuration at iteration \( t \) is denoted as \( X^{(t)} \), or equivalently, \( L^{(t)} \), and the poses of submaps within it are consistently written as \( L^{(t)}(s) \). Here we define time ‘-1’ as the moment right before loop-closing is detected. Correspondingly, time ‘0’ is the moment when loop-closing is confirmed, i.e., submap \( N_m \) is replaced with submap 1. Thereafter, \( t = 1, 2, 3, \ldots \) represents the index of iteration.

Taken the Euler function, the submap configuration can be iteratively updated as:

\[
L^{(t+1)} = L^{(t)} + \epsilon \times \nabla E^\text{Ext}(L^{(t)}) + \left( -\nabla E^M(L^{(t)}) \right)
\]

This is essentially a force balance process, in which these two forces push/attract the submap configuration \( L \) to the desired position.

### 4.1 Constraint From Pictorial Matching

The active contour [2] technique is employed in this work. SNAKE is essentially a gradient-like greedy shape fitting algorithm, which tries to minimize an energy function derived from the image so that it takes on its smaller values at the features of interest. In the scenario of SLAM, edges are natural features to be exploited, because the edges often represent structures in the environment, e.g., roads and buildings.

The image is denoted as \( I(u, v) \) here, its edge map can be conveniently calculated by well-established techniques such as Canny edge detector \( f^\text{edge}(u, v) = -|\nabla I(u, v)|^2 \). As can be observed, \( f^\text{edge}(u, v) \) gains high values at position close to edges (corresponding to real-world structures), while has low values in the homogeneous regions. The gradient of edge map, \( \nabla f^\text{edge} \), has vectors pointing toward the edges. This is a very important characteristic in the context of LCPM, because it gives a clue that how the deployment of a submap should be adjusted for a better match to the pictorial information. However, the 2D vectors in the gradient of edge map \( \nabla f^\text{edge} \) basically have very limited ‘capture range’, they gain high values only within the immediate vicinity of the edges, if the submap configuration is initially deployed outside this vicinity, the contribution of pictorial matching in the loop-closing optimization will be trivial.
The solution proposed by Xu [18] is to replace the vector field of edge image’s gradient by a dense vector field $V$. For each pixel $(u,v)$ on the image, a vector $V(u,v)$ is generated to measure, if a point is deployed at this pixel, how its position should be adjusted. A popular way to denote the vectors in $V$ is to use their projection in the x-axis and y-axis: $V(u,v) = [v_x(u,v), v_y(u,v)]$. So that we could compute the ‘force’ acting on a point by combining the force acting on it in the direction of x-axis and y-axis.

This vector field is derived from the image to minimize following target energy function:

$$
\varepsilon = \int \int \mu(v_x(u)^2 + v_x(v)^2 + v_y(u)^2 + v_y(v)^2) + |\nabla f_{\text{edge}}|^2 |V - \nabla f_{\text{edge}}|^2 dxdy
$$

(9)

where $\{v_x(u), v_x(v), v_y(u), v_y(v)\}$ is the partial derivatives of the vector field and $\mu$ is a constant. This variational formulation has several desirable characteristics. First, in the homogeneous regions where there is no particular information about the environment, in other words, $|\nabla f_{\text{edge}}|$ is small, the energy is dominated by the sum of squares of the partial derivatives of the vector field, which is denoted by the first item on the right side. This yields a smoother vector field compared to the one of edge gradient, so that the ‘capture range’ of the edges can be drastically increased. Second, when $|\nabla f_{\text{edge}}|^2$ is large, the second term dominates the integrand. By minimizing it, the $V$ is kept nearly equal to the gradient of the edge map in the vicinity of the edges, so that the edge information is preserved.

Submaps in this work are implemented as point clouds containing $N_s$ points. Here we assume that the camera has already been calibrated and its position is also available. To make the formulation concise, the projection from 3D world coordinate system to image coordinate system is wrapped by an operator $\odot$, which projects the $j^{th}$ 3D point in submap $M_i$ to the image plane $(u_{i,j}, v_{i,j}) = (x_{i,j}, y_{i,j}, z_{i,j}) \odot L(t)(s)$. Since each submap $M_s$ is modeled as a rigid body, when deployed within the potential field $V$, it moves as metal in a magnetic field. The force acting on the submap can be calculated by averaging the forces acting on all the $N_s$ 3D points inside it. Following the definition of the dense vector field, this external
force can be decomposed in the direction of x-axis and y-axis:

\[ F^{ext}(L^t(s), M_s) = \begin{pmatrix} \frac{\sum_{j=1}^{N_s} v_s(u_{i,j}, v_{i,j})}{N_s} \\ \frac{\sum_{j=1}^{N_s} v_s(u_{i,j}, v_{i,j})}{N_s} \end{pmatrix} \] (10)

\section*{4.2 Constraint From Initial Submap Configuration}

\subsection*{4.2.1 The constraint in a general form}

Before loop-closing is detected, say, at time \(-1\), the last component in the submap configuration is

\[ L(-1)(N_m) = L_0 \oplus \bar{x}_1 \oplus \bar{x}_2 \ldots \oplus \bar{x}_{N_m}. \]

This configuration from the mapping information is illustrated in Fig. 1(a), in which \(N_m = 9\). When a loop-closing is detected at time 0, the last submap is associated with submap 1, so the new pose of the last submap will be \(L(0)(N_m) = L_0 \oplus \bar{x}_1\).

Due to the errors in SLAM, \(L(-1)(N_m)\) will not be equal to \(L(0)(N_m)\). The difference between them is often referred as loop-closing error. To propagate the loop-closing error backward, each submap’s pose should be adjusted accordingly. Such adjustment can be imagined as ‘force’ acting on the submaps, which is similar to the force calculated from pictorial matching.

For convenience, here we define \(\varphi(s, s - 1) = \ominus L(s - 1) \oplus L(s) - x(s)\), which can be linearized at the linearization point \(\hat{L}(a), \hat{L}(b)\) using the Jacobian term:

\[ J_a = \left. \frac{\partial \varphi(a, b)}{\partial L(a)} \right|_{\hat{L}(a), \hat{L}(b)} \quad J_b = \left. \frac{\partial \varphi(a, b)}{\partial L(b)} \right|_{\hat{L}(a), \hat{L}(b)} \]

Using above Jacobian terms, the transformation between two submaps can be linearized as following:

\[ \varphi(a, b) = J_a L(a) + J_b L(b) - \left( J_a \hat{L}(a) + J_b \hat{L}(b) - \psi(\hat{L}(a), \hat{L}(b)) + \bar{x}_{N_m} \right) \] (11)

According to the elaboration in [11], the energy function in (7) can be finally formulated as:

\[ E^M(L) = L^T A L - 2L^T B + C \] (12)
where

\[
A = \sum_{\{a,b\} \subset X} \begin{pmatrix} 
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & J_a^T(P_b^{-1})J_a & \ldots & J_a^T(P_b^{-1})J_b & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & J_b^T(P_b^{-1})J_a & \ldots & J_b^T(P_b^{-1})J_b & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots 
\end{pmatrix}
\]

\[
B = \sum_{\{a,b\} \subset X} \left( \begin{pmatrix} 
J_a^T(P_b^{-1})y & \ldots & J_b^T(P_b^{-1})y & \ldots 
\end{pmatrix} \right)^T
\]

the \( y \) and \( C \) are constants and each pair of \( \{a, b\} \) represents a transformation between two submaps. For simplicity of notation, here we define:

\[
\begin{pmatrix} 
J_a^T(P_b^{-1})J_a & J_a^T(P_b^{-1})J_b \\
J_b^T(P_b^{-1})J_a & J_b^T(P_b^{-1})J_b 
\end{pmatrix} = \begin{pmatrix} 
Q_{a,a}^b & Q_{a,b}^b \\
Q_{b,a}^b & Q_{b,b}^b 
\end{pmatrix}
\]

(13)

In a similar manner, we can write \( J_a^T(P_b^{-1})y = R_a^b \). Based on the loop graph we introduced in Fig. 1, there should be:

\[
B = \begin{pmatrix} 
R_1^2 & R_2^3 & \ldots & \ldots & R_{N_m-1}^{N_{m-1}} + R_{N_m-1}^{N_m} & R_{N_m}^{N_m} 
\end{pmatrix}^T
\]

(14)

Therefore, the force representing the mapping information can be computed according to (8):

\[
-\nabla E^M(\mathcal{L}) = -\frac{\partial (\mathcal{L}^T A\mathcal{L} - 2\mathcal{L}^T \mathcal{B} + \mathcal{C})}{\partial \mathcal{L}} = (B - A\mathcal{L}).
\]

(15)

Please note that the iteration index \( t \) is omitted in this equation.

5. Experimental Results

Several experiments were conducted to examine the performance of the proposed loop-closing algorithm. The testing platform is a 4 × 4 vehicle with a range scanners mounted on the top. This vehicle is also equipped with a Inertial Science DMARS-I IMU to estimate its motion. A high accuracy GPS is used to provide ground truth for reference.
The total length of the vehicle’s trajectory is 1,890 meters. During the experiment, 6,237 frames of 2D range scans were collected. New submap is initialized when the number of scans in the old submap reaches a certain pre-defined threshold. Using such a strategy, totally 32 submaps have been constructed.

The submap’s origin is set to be the vehicle’s pose at the moment when this submap is constructed. All the 3D points within the submap are then stored relative to its local coordinate system. Since the objective of this work is to achieve global consistency, the local map building is naively performed by directly projecting the range scans from the vertical laser scanner to the vehicle’s poses read from IMU. Since the submap’s length is constrained, i.e., generally less than 30 meters, the accumulated error of IMU readings within each submap is moderate. However, we believe that incorporating a local SLAM scheme with EKF or particle filter will surely benefit the submap building process and achieve higher accuracy within the local submaps, as in [19] [20] [21]. Using above approach, a 3D submap $M_s$ can be imaged as a pile of 2D laser slices horizontally deployed. By accumulating such ‘slices’ of range data, a submap can be conveniently represented by a point cloud. One of these submaps is depicted in Fig. 2.

![Fig. 2. The front view of the 3D submap 1.](image)

5.1 Uncertainty model $\mathcal{P}$

If the deformable submap graph is imaged as a sequence of blocks connected by springs, the value of $\mathcal{P}$ determines the toughness of the springs. In this work a synthesis $\mathcal{P}$ is employed to examine the proposed
In the experiments, the transformation uncertainties are set to be proportional to the aptitude of vehicle’s transformation with respect to the previous submap. Let \( \bar{x}_s = \begin{pmatrix} \Delta \bar{x}_s & \Delta \bar{y}_s & \Delta \bar{\gamma}_s \end{pmatrix} \) denotes the transformation between submap \( s-1 \) to \( s \), the corresponding synthesis \( P_s \) is calculated as a diagonal matrix \( \text{dia}(\varsigma_x, \varsigma_y, \varsigma_z) \), where \( \{\varsigma_x, \varsigma_y, \varsigma_z\} \) are constants controlling the amount of error. Since the vehicle is moving on the road, we set \( \varsigma_x \) and \( \varsigma_y \) to be small to represent the small translation error. Meanwhile, the \( \varsigma_\gamma \) is set to be comparatively large to model the error during the turning. The initial submap configuration constructed by the above uncertainty model is depicted in Fig. 3(a).

It can be observed that this synthetic uncertainty model is consistent with the general characteristic of SLAM’s error: the longer the vehicle travels, the more error will be accumulated.

5.2 Limitation of Using Only Mapping Information

Without pictorial matching, it is very difficult for the SLAM to converge by visiting the environment for only once, as showed in Fig. 3. Each two consecutive submaps are connected in a smooth manner, and very little conflict can be observed, but the whole submap structure is erroneous.Shortly speaking, the map is only consistent on the local level, but incorrect on the global level. This result proves that the guidance from global information is definitely necessary if we want the map to converge within just one loop.

5.3 Result of Loop-Closing With Pictorial Matching

The final mapping result of our proposed algorithm is showed in Fig. 4. The parameter setting we used for this result is : \( \epsilon = 20 \). Accuracy of the algorithm is measured by the root-mean-square (RMS) error between submap configuration and the ground truth state \( \mathcal{X}^g = (L_1^g, L_2^g, \cdots, L_{N_m}^g) \). The ground truth was obtained by the onboard GPS/INS system.

The submap graph takes 6 iterations to converge to an optimal configuration, which is showed in Fig. 4(a). During these 6 iterations, the submap graph moves gradually to the correct configuration, which
can be observed through the reduction of RMS error in Fig. 4(b). By comparing Fig. 4 with the ground truth acquired from GPS data, it can be observed that our algorithm can achieve both local and global consistencies by visiting the environment for only once.

After rendering the points cloud $\mathcal{M}$ to the maximum a posteriori submap configuration using VRML, the final 3D mapping result for the target environment is depicted in Fig. 5.

To demonstrate the superiority of the proposed algorithm, the accuracies of different loop-closing techniques are depicted in Fig. 6. For each submap, we compare the Euclidean distance between the LCPM estimate and GPS ground truth: the proposed technique achieves much higher accuracy than the other two methods. We can further notice that, for generic loop-closing with only mapping information (marked with squares), the error grows almost monotonously from the beginning to the middle of the submap configuration. Such error is caused by accumulation of local filtering errors: the longer the robot travels, the bigger the error will be. Loop-closing tries to re-distribute this error to obtain a more consistent map, which leads to the drastic decrease of error from the middle to the end of submap configuration.
Fig. 4. (a) The mapping result using the proposed algorithm. The loop has been accurately closed by visiting the environment for only once; (b) The loop-closing result using only mapping information.

Fig. 5. The 3D environment map rendered by VRML, using the ground truth from GPS/INS and the result of LCPM. It can be observed that the environment can be accurately mapped by one robot visiting the environment for only once.

5.4 The parametric \( \epsilon \)

The setting of \( \epsilon \) plays an important role in the LCPM, it controls how much the pictorial matching can deform the initial submap configuration. However, we noticed that to derive \( \epsilon \) analytically is a challenging task, because the quality of the input image (with respect to matching with real world structure) is difficult to measure. In this work, using the available data we try to qualitatively analyze the characteristics of the \( \epsilon \).

During the experiment, it is observed that smaller value should be given to \( \epsilon \) when the image’s quality
Fig. 6. Accuracy comparison between LCPM, mapping information only (generic SLAM), and pictorial matching only.

is poor, e.g., blurry or noisy. Basically, the blurry image provides much less information about the environment than the original image. Therefore, if the same $\epsilon$ is employed in this case, we actually ‘over-weight’ the contribution of pictorial matching in the LCPM. Consequently, the final mapping result is distorted by the error coming with the blurry image, as in Fig. 8.

Fig. 7. The LCPM result using blurry image, when $\epsilon = 0.05$. Significant distortion can be observed in the indicated regions as A and B.

A smaller $\epsilon$ puts more weight on the loop-closing using only mapping information, which is equivalent to release the constraint from pictorial matching. This can to some extent alleviate the distortion existing in Fig. 7. The mapping result using bigger $\epsilon$ is depicted in Fig. 8. Being blurry, the input image now is less informative than the original one, consequently, LCPM basically cannot achieve the performance as using non-blurred image: there are still minor distortions existing in the final map, as in the regions C.
and $D$ indicated by the arrows.

![Mapping result](image1)

(a) Mapping result

![Error analysis](image2)

(b) Error analysis

Fig. 8. LCPM using blurred input image when $\epsilon = 11$. By reducing $\epsilon$, the distortion can also be reduced. However, some minor distortions can still be observed.

The setting of $\epsilon$ is also variant to the scale of input image. In the experiments, only two scales of the input image are available, one is at the scale of 1 pixel = 1.189 meter, as used in Fig. 4. The other one is at a higher scale: 1 pixel = 3.303 meter.

When image has a larger scale, i.e., larger meter per pixel value, the constraint calculated from mapping information is reduced when projected into the image domain, while the external force from pictorial information will be about the same due to the characteristic of the dense vector field. Therefore, the increase of scale essentially amplifies the constraint from pictorial matching. If the same $\epsilon = 20$ is employed as the one used in the small scale image, the resulting map could be seriously distorted. For example, in the regions marked as A and B in Fig. 9(a), the constraint from mapping information has been released in scaling, so it cannot compensate the force from pictorial matching which strongly attracts submaps to wrong places. For large-scale input image, a smaller value of $\epsilon$ should be employed. The distortion existing in Fig. 9(a) is reduced when $\epsilon$ is set to be 11, the improvement can be found in Fig. 9 (b). Please note that there are still errors existing in the result using large $\epsilon$, as indicated by $C$. This can be attributed to the less detailed information provided by the large-scale image, which has a lower resolution than the small scale one. It can be observed that, although the accuracy is improved by
reducing the $\epsilon$, the overall performance is not as good as the one using original image. This can also be explained by the lose of information when the image is zoomed out.

5.5 Algorithmic Complexity

The computational complexity of the proposed algorithm comprises two parts: offline computation and online computation. The off-line processing’s objective is to construct the potential field from input image, whose operation is linear to image size and loop iterations. The more iterations it loops, the further the field can affect. How many iterations should be used may vary with the applications.

For the online computation, the proposed algorithm exploits the sparse matrices $A$ and $B$ introduced by the relative pose representation. Therefore, the presented algorithm can perform one iteration of deformation, as in (8), in $O(N_m^1)$ computation time, where $N_m$ is the size of submap transformation $\mathcal{X}$. So, the online operation of LCPM scales linearly with three terms, i.e., submap number, submap size, and iteration number. The number of submaps is determined by the submap segmentation scheme. The submap’s size depends critically on submap representation.
6. Conclusion and Future Work

In this paper, a pictorial matching based technique is presented to conduct large scale mapping by only one single robot going through the environment for only once. The map is represented as a sequence of submaps deployed as a deformable configuration in a global coordinate system. The submap graph is estimated by *maximum a posteriori*, based on both mapping information and pictorial matching. The MAP estimation is finally formulated as an energy minimization problem, and active contour algorithm is introduced to optimize the submap configuration efficiently.

Although this paper is dedicated to the loop-closing problem for the cyclic environment, we expect that the LCPM can nevertheless be applied to more complex environment with multiple loop-closings. New experiments are now undergoing.

A major limitation of LCPM in its current form is the setting of parameter $\epsilon$. It plays an important role in the iterative energy minimization process, however, to derive the $\epsilon$ is a challenging task as it is difficult to obtain a quantification of the error in the image. In the future, we argue that more research work should be conducted to investigate LCPM using different pictorial information with different setting of $\epsilon$.

It is also attractive to use the pictorial information from the very beginning, rather than only in the loop-closing stage, so that the accumulation of error could be prevented. This technique essentially conducts the energy minimization in Section 4 incrementally, i.e., submap by submap. Such a strategy deserves more investigation in the future work.

References


