Real-time Scan-Matching Using $L_0$-norm Minimization Under Dynamic Crowded Environments

Yusuke Hieida$^1$, Tsuyoshi Suenaga$^1$, Kentaro Takemura$^1$, Jun Takamatsu$^1$ and Tsukasa Ogasawara$^1$

Abstract—We propose real-time scan-matching based on $L_0$-norm minimization under dynamic crowded environment. The prior scan-matching methods are based on $L_2$-norm minimization, because the measurement noise follows the normal distribution in static environments. This assumption is unfortunately broken in dynamic crowded environments.

We propose to use the idea of Locality Sensitive Hashing (LSH) to accelerate the $L_0$-norm minimization, which usually is a time-consuming process. The LSH customized for our issue reduces the calculation time even in the worst cases. The experimental results demonstrate the effectiveness of the proposed method compared with standard $L_2$-norm minimization and its robust version with M-estimator.

I. INTRODUCTION

It is very demanded for an autonomous robot to move under unknown environments [1]. Scan-matching estimates robot’s displacement by aligning time-series measurements. Concatenation of displacement and the aligned measurement generate robot’s trajectory and the map respectively, while may suffer from accumulation of alignment errors. Iterative Closest Point (ICP) method [2] and particle filter [3], [4] improve the performance under static environments [5].

However, the assumption valid for static environments is broken in real-world environments. Under dynamic crowded environments, the accuracy of the alignment deteriorates because both static (referred to as inliers) and moving objects (referred to as outliers) are rigidly aligned, resulting in poor performance of the prior methods (Fig. 1). To robustify outliers in the alignment is inevitable under dynamic environments.

A. Related work

There are two types of approaches for dynamic environments. One approach explicitly detects inliers/outliers by feature extraction, tracking and so on. The other approach gradually or simultaneously estimates the likelihood of outliers and decreases their effect during the alignment.

In the first approaches, Wolf and Sukhatme [6] proposed to use only landmarks, which tend to be static, as inliers. Examples of landmarks are wall planes and corners. Wang et al. [7] propose to interleave object tracking based on motion detection. Hähnel et al. [8] also propose to use object tracking based on the sample joint probabilistic data association filter. Rodriguez-Losada and Minguez [9] prove that the possible displacement to satisfy correspondence of one point between two frames is helix and detect the outliers by using the distance to the helix. The performance of these methods deeply depends on the detection of inliers/outliers.

In the second approaches, Hähnel et al. propose to use the expectation-maximization (EM) algorithm [10]. The E-step evaluates likelihood of outliers and the M-step aligns measurements with weighting according to the likelihood. Ramos et al. [11] propose to use the conditional random field (CRF) to stochastically estimate data association and outliers. Further, Ven et al. [12] estimate moving object and motion as well as data association. Since these methods need to optimize the function with large number of parameters, it is difficult to achieve real-time operation.

Viewing from the outlier detection, use of robust statistics, such as M-estimator [13], is one of the solutions. There are several methods that use M-estimator for scan matching-based alignment [14], [15]. Although the optimization of the M-estimator is simpler and thus faster than the second approaches, it may also lack of inappropriate convergence.

B. Proposed method

We propose real-time scan-matching that suits adapted to dynamic environments. We use a common real-time LIDAR sensor, which measures depths only on the cross-sectional plane. Thus, we assume that the ground is flat and the measuring cross-section is parallel to the ground. We believe that the proposed method can be extended to its 3-D version.

The novelty of the proposed system is to use $L_0$-norm being that prior systems use $L_2$-norm. Although optimization of the $L_0$-norm tends to be time-consuming, we propose to use the Locality Sensitive Hashing (LSH) [16] to make the proposed method suitable for real-time.

The Contributions of this paper are twofold:

- We experimentally prove that the effectiveness of the $L_0$-norm for scan-matching under dynamic environments. Note that such norm is often employed in other applications, such as face recognition [17], image inpainting [18], and denoising [19].
- We propose to accelerate the $L_0$-norm calculation with approximation using the LSH [16]. Unlike the $L_2$-norm calculation, we only need to decide if the other points exist inside the pre-defined radius and it is not necessary to consider their point ID. These aspects stabilize the calculation time even in the worst cases and reduce memory consumption by removing the point ID information.

$^1$They are with Graduate School of Information Science, Nara Institute of Science and Technology, Japan {yusuke-h, tsuyo-s, kenta-ta, j-taka, ogasaw} at is.naist.jp
function represents the translation in robot’s planer displacement. The \( L_0 \) of the points near the distance less than \( \epsilon \). Note that calculation of the \( L_0 \)-norm does not require the explicit point correspondences, just to decide if the other points exist inside the radius \( \epsilon \). We refer to these points as r-near neighbors following the terminology in [16].

**B. Acceleration by Locality Sensitive Hashing**

We employ the Locality Sensitive Hashing (LSH) [16], whose calculation time is \( O(1) \). The LSH limits the search space using the hash function, whose values of two proximity points tend to be the same. In the LSH, only the points with similar hash values are considered. Using multiple hash functions reduces the tendency to failure in the search. Concretely, the hash function is defined as

$$ h(x) = \left\lfloor \frac{a \cdot x + b}{w} \right\rfloor, $$

where the symbol \( \lfloor \rfloor \) means floor operation, \( x \) is the data point, \( w \) is the width of the hash bin and the parameters \( a \) and \( b \) are randomly chosen by \( p \)-degree stable distribution and uniform distribution from \( 0 \) to \( w \), respectively. We only consider the projection of x- and y-axes as the vector \( a \), e.g., \((a_x,0)\) and \((0,a_y)\), and use them by concatenating x- and y-axis projection LSH as shown in Fig. 2. The existence of r-near neighbor is searched by accessing the bin of the table several times (at most, number of tables).

Although the LSH is usually used for nearest neighbor search, the LSH in r-near neighbor search has two more advantages. In the nearest neighbor search, when all the searched points \( \{q_j\} \) have different hash values, it is necessary to continue searching the points whose values are near to the target value. However in r-near neighbor search, the calculation is just terminated and we conclude that no r-near neighbor exists. \( L_0 \)-norm does not consider the position or ID of the r-near neighbor, and depends only on the existence.

**IV. Minimization by Importance sampling**

The approximate robot’s displacement is obtained from odometry or control inputs. Thus, we use the sampling-based optimization method considering importance of searched region. In strict sense, minimization of \( L_0 \) norm is equivalent to combinatorial optimization. In this sense, using the method in [20] is alternative and future work.

We minimize the evaluation function using importance sampling and coarse-to-fine search as shown in Fig. 3:

1) randomly generate \( n_1 \) locations (referred to as sample-A) based on odometry or control inputs
2) calculate the importance of each sample by Eq. (4).

$$ \exp \left( \frac{m - E(R,t)}{k} \right). $$

The parameter \( k \) controls the degree of the importance with respect to the evaluation function.

3) select \( n_2 \) locations (referred to as sample-B) from sample-A following their weights.

Fig. 1. Erroneous estimation of displacement due to dynamic environment. The circular object moves rightward from time \( t-1 \) (middle) to time \( t \) (left), while the L-shape object is fixed. Alignment only using the L-shape object outputs the correct displacement. But the prior scan-matching methods diffuse matching errors uniformly in the space, resulting in erroneously estimating the displacement.

Fig. 2. Several boolean-array tables are generated by the LSH. Each bin represents the existence of r-near neighbors and thus the existence is decided by accessing the corresponding bin in \( O(1) \) time.
High likelihood Position → Extraction → Resampling → Estimated Position

Fig. 3. Minimization by importance sampling and coarse-to-fine search. By gradually changing the distribution \( \sigma \) of sampling range, the samples converge on the minimum in the \( L_0 \)-norm.

(a) LIDAR: SICK LMS100  (b) Mobile robot: EMC-230

Fig. 4. Equipments used in this experiment

<table>
<thead>
<tr>
<th>Specification of SICK LMS100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Viewing angle</td>
</tr>
<tr>
<td>Angular resolution</td>
</tr>
<tr>
<td># of points in each frame</td>
</tr>
<tr>
<td>Accuracy in depth</td>
</tr>
<tr>
<td>Maximum depth</td>
</tr>
<tr>
<td>Frequency</td>
</tr>
</tbody>
</table>

Table I

4) randomly generate \( n_3 \) locations by normal distribution with variance \( (\sigma_x, \sigma_y, \sigma_\theta) \) around from sample-B. \( n_2 \times n_3 \) locations are sampled in total.

5) repeat 2) to 4) in several times, while reducing variances.

V. EXPERIMENTS

A. Experimental setup

We used the LIDAR, SICK LMS100 (Fig. 4 (a)), and a wheelchair, EMC-230 made by Imasen Engineering Corporation as the mobile robot (Fig. 4 (b)). This LIDAR measures uniformly along the angle direction and thus the measured points become sparser as the depth increases. Considering the calculation time and the estimation accuracy, the measured points \( \{p_i\} \) are segmented and thinned out so that the points are distributed uniformly in the space. On the other hand, in the measured points \( \{q_j\} \), additional points are inserted as all the intervals of neighbors are less than the threshold \( \epsilon \) in Eq. (2). We set \( \epsilon \) to 1 [cm] based on the accuracy of the LIDAR used in this experiment (see Table I).

We conducted three types of experiments. The first experiment verifies the effectiveness of the \( L_0 \)-norm in outliers compared with simple ICP, ICP with thresholding, and ICP with M-estimator, which are most related to the proposed method. Note that first approaches described in Section I-A use such methods after detecting outliers. The second experiment verifies the effectiveness of the LSH with respect to the calculation time and accuracy. The third experiment verifies the applicability of the proposed method through actual use. The experimental environment is mainly the inside of a building of our campus.

B. Verification of effectiveness of \( L_0 \)-norm

Figure 5 shows the experimental environment which is about 5 [m] in width and 20 [m] in depth. About 25 people are moving toward the robot within the environment. The robot measures the environment while fixed. Thus, the estimated robot displacement \((x, y, \theta)\) should be always zero, where \( x, y, \) and \( \theta \) are the translation along the x- and y-axes and the orientation, respectively.

To verify the effectiveness without considering the initial guess, we optimize all of the functions by brute-force search, where the ranges of \( x, y, \) and \( \theta \) are \([-5 \text{ [cm]}, 5 \text{ [cm]}]\), \([-5 \text{ [cm]}, 5 \text{ [cm]}]\), and \([-0.3 \text{ [rad]}, 0.3 \text{ [rad]}]\) and sampled uniformly at 0.5 [cm], 0.5 [cm], and 0.01 [rad] (\(\approx 0.57 \text{ [deg]}\)), respectively. There are 24000 samples in total.

In ICP with thresholding, we ignore some ratio of the further corresponding data as outliers. We employ the Cauchy and the Biweight function as M-estimator \( \rho(d) \) and use \( \rho(f(Rp_i + t, \{q_j\})) \) in place of \( f(Rp_i + t, \{q_j\}) \) in Eq. (1). The Cauchy function is defined as Eq. (5) and is often used (e.g. [14], [15]):

\[
\rho(d) = \frac{C^2}{2} \log \left(1 + \left(\frac{d}{C}\right)^2\right),
\]

(c) floor plan

Fig. 5. Experimental environment. (a) appearance of the inside of the building. (b) actual experimental environment crowded by people. (c) floor plan of the environment.
where $C$ is a positive real value. The Biweight function is defined as Eq. (6) and completely removes the effect of the outliers (shape of the function is flat in the region $|e| \geq 0.01$) similar to the proposed $L_0$-norm.

$$\rho(d) = \begin{cases} \frac{B^2}{2} & (d \geq B) \\ \frac{B^2}{2} \left(1 - \left(1 - \left(\frac{d}{B}\right)^2\right)^3\right) & (d < B) \end{cases}$$  

(6)

where $B$ is a positive real value. Both values, $B$ and $C$, are set to 1 [cm] considering the sensor’s accuracy.

Since the estimated displacement should be zero as described before, we quantitatively compare them using the mean square error (MSE) of the displacement. Table II shows the results. The estimation by the $L_0$-norm, $L_2$-norm with higher threshold, and the Biweight function are similar and better than the others, because sensor accuracy is 1.2 [cm]. This result demonstrates the effectiveness of the $L_0$-norm, which regards the point outside of the pre-defined radius as outliers. The Biweight function and ICP with higher threshold have a similar characteristic and thus achieves a better performance. But this means that these three methods have the same disadvantages with respect to the convergence. Note that distance used as threshold in ICP with thresholding method has a similar characteristic and thus achieves a higher threshold, and the Biweight function are similar and better than the others.

<table>
<thead>
<tr>
<th>Method</th>
<th>MSE ($x$ [cm], $y$ [cm], $\phi$ [rad])</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L_2$-norm</td>
<td>(3.37, 2.08, 0.016)</td>
</tr>
<tr>
<td>Threshold (30 percents)</td>
<td>(1.17, 0.033, 0)</td>
</tr>
<tr>
<td>Threshold (50 percents)</td>
<td>(0.19, 0.004, 0)</td>
</tr>
<tr>
<td>Cauchy</td>
<td>(1.33, 0.013, 0)</td>
</tr>
<tr>
<td>Biweight</td>
<td>(0.13, 0.002, 0)</td>
</tr>
<tr>
<td>$L_0$-norm</td>
<td>(0.42, 0.015, 0)</td>
</tr>
</tbody>
</table>

Table III shows the average calculation time for the brute-force search (24000 times of calculation), number of $L_0$-norm calculations per second and the root mean square error (RMSE) of the calculated $L_0$-norm; the simple and kd-tree methods always output correct answers, but the LSH method does not. In calculation time, the LSH method is about five times faster than the kd-tree method, but sacrifices the accuracy; RMSE of the $L_0$-norm is 14.4, while the average of the norms is 280.03.

To evaluate the degeneration caused by the approximation of the $L_0$ norm calculation, we actually construct the map using the LSH method. Figure 10 shows the map which the RBPF-SLAM method [21] estimates from measurement of the static environment and we regard it as ground truth. Figure 11 shows the map which the LSH method estimates from measurement of the same but dynamic environment. Despite the approximation, the estimated map is acceptable in visual inspection.

C. Verification of effectiveness of LSH

We compare the proposed LSH acceleration with the simple method and the kd-tree method [22]. The computer to employ is composed of Intel Core2Duo T7500 (Clock speed: 2.2 [GHz]) and 2 [GB] memory. We conducted this experiment with the same configuration of the previous experiment, but the robot moved.

Table III shows the average calculation time for the brute-force search (24000 times of calculation), number of $L_0$-norm calculations per second and the root mean square error (RMSE) of the calculated $L_0$-norm; the simple and kd-tree methods always output correct answers, but the LSH method does not. In calculation time, the LSH method is about five times faster than the kd-tree method, but sacrifices the accuracy; RMSE of the $L_0$-norm is 14.4, while the average of the norms is 280.03.

To evaluate the degeneration caused by the approximation of the $L_0$ norm calculation, we actually construct the map using the LSH method. Figure 10 shows the map which the RBPF-SLAM method [21] estimates from measurement of the static environment and we regard it as ground truth. Figure 11 shows the map which the LSH method estimates from measurement of the same but dynamic environment. Despite the approximation, the estimated map is acceptable in visual inspection.

D. Verification in Actual Use

We actually estimate the map using the proposed method, which includes the LSH method and minimization by the importance sampling. The localization at each frame is performed at about 5 [fps] on the same computer as that of Section V-C. The parameters are: $n_1 = 300$, $n_2 = 20$, $n_3 = 15$, $k = 20.0$. Steps 2) to 4) are repeated twice,
and \((\sigma_x, \sigma_y, \sigma_\theta) = (0.03, 0.03, 0.025)\) at the first time and 
\((\sigma_x, \sigma_y, \sigma_\theta) = (0.01, 0.01, 0.01)\) at the second time.

In the first experiment, the size of the experimental environment is about 50 [m] \times 30 [m]. About 25 people are moving around the robot as outliers as shown in Fig. 12. 300 frames of measurements are used. Figure 13 shows the estimated map. Figure 14 shows the occupancy grid map obtained from the estimation results using the open SLAM software MRPT 1. We overlay the floor plan of this building to evaluate the accuracy.

Figure 15 shows the estimated map in the second experiment, which includes both indoor and outdoor measurements. The size of the experimental environment is about 100 [m] \times 60 [m]. 700 frames of measurements are used. To evaluate the accuracy of the map, we overlay the aerial photograph downloaded from the Google maps 2.

E. Discussion

Both the \(L^0\)-norm, ICP with higher threshold, and the Biweight function have similar advantages and disadvantages. Unlike the other two methods, by ignoring the distance and ID of the r-near neighbor in the LSH, the \(L^0\)-norm calculation is accelerated. Despite ignoring them, the localization accuracy is surprisingly maintained as shown in the experiments. Note that better results obtained by these three methods strongly support the plausibility of using the \(L^0\)-norm.

VI. Conclusion

In this paper, we proposed real-time scan-matching for dynamic environments, which works at about 5 [fps] on Intel Core2Duo T7500 (Clock speed: 2.2 [GHz]) with 2 [GB] memory. The proposed method uses \(L^0\)-norm, not \(L^2\)-norm, for alignment. First, we proved the robustness of the \(L^0\)-norm to outliers through actual use. Next, we proposed to accelerate r-near neighbor search by the LSH technique. Unlike the nearest neighbor search, we only need to determine if r-near neighbors exist and it is not necessary to consider the IDs of the neighbors. These aspects do not affect the calculation time even in the worst cases and reduce memory consumption by removing the point ID information.

The proposed method estimates the displacement only from two subsequent measurements. Generally, scan-matching suffers from accumulation of the estimation errors. Loop-closing technique [23] is one of the solutions.

REFERENCES

Fig. 12. Snapshots of images obtained from digital camera (left column) and SICK LMS100 (right column) installed in the robot.

Fig. 13. The map estimated by the $L_0$-norm minimization

Fig. 14. The occupancy grid map obtained by the open SLAM software MRPT and the proposed method. We overlay the floor plan of this building to evaluate the accuracy.

Fig. 15. The map estimated by the proposed method. We overlay the corresponding aerial photograph from the Google Maps to evaluate the accuracy.


