2009 IEEE International Conference on Robotics and Automation
Kobe International Conference Center
Kobe, Japan, May 12-17, 2009

Learning to Detect Loop Closure from Range Data

Karl Granström and Jonas Callmer
Div. of Automatic Control, Dept. of Electrical Engineering
Linköping University, Sweden
{karl, callmer}@isy.liu.se

Fabio Ramos and Juan Nieto
Australian Centre for Field Robotics
University of Sydney, Australia
{f.ramos, j.nieto}@acfr.usyd.edu.au

***Abstract*—Despite significant developments in the Simultaneous Localisation and Mapping (SLAM) problem, loop closure detection is still challenging in large scale unstructured environments. Current solutions rely on heuristics that lack generalisation properties, in particular when range sensors are the only source of information about the robot's surrounding environment. This paper presents a machine learning approach for the loop closure detection problem using range sensors. A binary classifier based on boosting is used to detect loop closures. The algorithm performs robustly, even under potential occlusions and significant changes in rotation and translation. We developed a number of features, extracted from range data, that are invariant to rotation. Additionally, we present a general framework for scan-matching SLAM in outdoor environments. Experimental results in large scale urban environments show the robustness of the approach, with a detection rate of 85% and a false alarm rate of only 1%. The proposed algorithm can be computed in real-time and achieves competitive performance with no manual specification of thresholds given the features.**

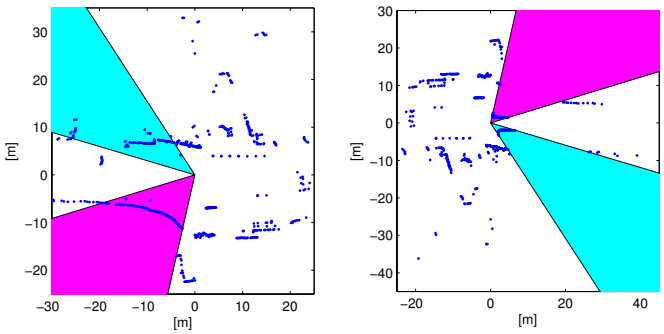


Fig. 1. Illustrative example of the loop closure detection problem. Despite the significantly different appearance of two laser scans depicted in the picture, both scans were obtained in the same location but rotated 180 degrees with respect to each other. Further changes include two cars observed in the right scan that are not present in the left scan. The highlighted areas correspond to regions of significant differences between the scans due to occlusions.

I. Introduction

For the last fifteen years, the robotics community has experienced a tremendous effort to find robust and general solutions for the Simultaneous Localisation and Mapping (SLAM) problem. The main motivation is the primary importance of this task for reliable autonomy in unknown environments. Despite significant developments in reducing the computational cost and increasing the robustness of SLAM algorithms, operation in large scale environments is still difficult mainly due to data association issues. In particular, the loop closing problem, where the robot needs to identify previously visited locations, is of crucial importance. An incorrect loop closure detection can significantly jeopardise the consistency of the map. In a robot configuration where only range sensors are available, identifying loop closures can be very challenging especially due to changes in the robot's viewpoint or dynamic objects in the environment.

To illustrate the difficulty of this problem, consider the example shown in Figure 1. A quick look at the laser scans depicted in the figure would indicate that they were obtained at different locations. In reality, the scans were obtained from very close positions, but at different times and with different orientation. The right scan is rotated 180 degrees with respect to the left scan, and in the right scan two cars have been parked along the side of the road (the L-shaped point clusters, slightly right of the origin in the highlighted areas, are two vehicles). This example demonstrates that identifying loops can be very difficult, especially when the environment is observed from different orientations. In addition to the vantage point problem, it is very common in practical applications to close a loop after several hundreds of metres or even kilometres. As a consequence, the robot's pose uncertainty can be significantly large, further complicating data association.

In this paper, we cast the problem of loop closure detection as a classification task. By introducing a number of features, especially designed to have small variance against different viewpoints, we are able to learn a classifier for real-time loop closure detection. The classification technique employed is based on AdaBoost [1] which builds a strong classifier by concatenating very simple decision rules. The result is a powerful non-linear classifier with very good generalisation properties [2], [3].

The main contribution of this paper is an automatic procedure for loop closure detection using elements of statistical learning. This is achieved by using a combination of rotation invariant features extracted from laser scans. The approach is extensively evaluated using 800 laser scan pairs from three different urban data sets. As a secondary contribution, the loop closure detection algorithm is integrated into a scan-matching SLAM framework using the Exactly Sparse Delayed-State Filter (ESDF), and combined CRF-matching [4] and ICP [5] for scan alignment. This is demonstrated in a data set about 2 kilometres long.

The paper outline is as follows. The subsequent section presents related work. The loop closure detection algorithm

is presented in Section III. Section IV presents the SLAM framework adopted which efficiently handles long trajectories. The features are evaluated in Section V-A. Experiments on loop closure detection are presented in Section V-B and results from a full SLAM experiment are provided in Section V-C. Finally, Section VI concludes the paper.

II. Related Work

In this section we summarise relevant work on loop closure detection and large-scale SLAM.

SLAM algorithms based on raw laser scans have been shown to present a more general solution than classic feature-based [6]. For example, in [7]–[9], raw laser scans were used for relative pose estimation. The mapping approach presented in [6] joins sequences of laser scans to form local maps. The local maps are then correlated with a global laser map to detect loop closures. Laser range scans are used in conjunction with EKF-SLAM in [10]. The authors introduced an algorithm where landmarks are defined by templates composed of raw sensed data. The main advantage claimed is that the algorithm does not need to rely on geometric landmarks as traditional EKF-SLAM. When a landmark is re-observed, raw points could be augmented with new sensor measurements, thus improving the representation of landmarks. The authors also introduced a shape validation measure as a mechanism to enhance data association when landmarks are re-observed. In summary, the main advantage in all these works is the ability of the algorithms to work in different environments due to the general environment representation obtained from raw sensor data.

Mapping algorithms based on laser scans and vision have shown to be robust. The work presented in [11] performs loop closure detection using visual cues and laser data. Shape descriptors such as angle histograms and entropy are used to describe and match the laser scans. A loop closure is only accepted if both visual and spatial appearance comparisons credited the match. In [12], laser range scans are fused with images to form descriptors of the objects used as landmarks. The laser scans are used to detect regions of interest in the images through polynomial fitting of laser scan segments while the landmarks are represented using visual features.

The approach presented in this paper uses only laser information. Perhaps the most relevant work is the algorithm presented in [8], [13] where consecutive laser scans comprise submaps. Feature descriptors of the maps are composed using a histogram representation. The feature representation allows the authors to match local maps without prior knowledge of their relative position. The histogram method utilises entropy metrics, weighed histograms and quality metrics. The results presented in [13] show 48% detection rate for a 1% false alarm rate. These results are improved slightly in [8] to a 51% detection rate for the same false alarm rate.

In this paper we present a solution to the loop closure problem based on a machine learning approach. A similar classification approach based on AdaBoost was used by Arras *et al* [14] for detecting people from laser scanners in a cluttered office environment. The approach was based on the classification of laser segments as whether or not belonging to a pair of legs. Detection rates of over 90% were achieved. Using the same ideas, place recognition was performed in indoor environments in [15].

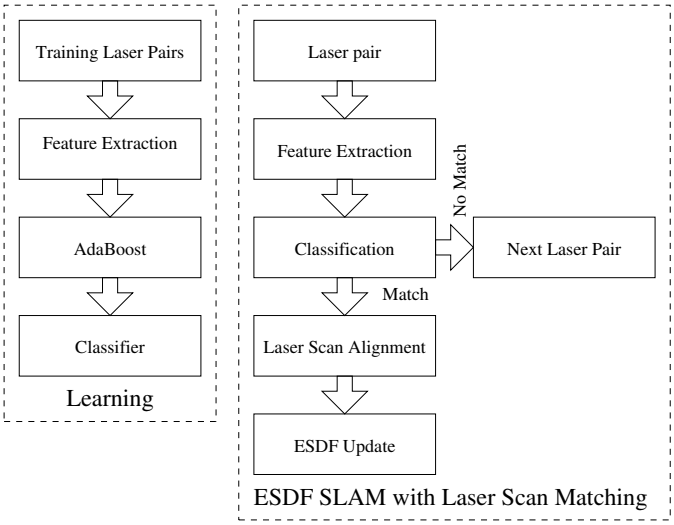


Fig. 2. Diagram depicting the learning and SLAM phases of algorithm.

III. Loop Closure Detection

This section describes the main algorithm of the paper; the loop closure detection procedure. In the following section, loop closure is integrated with a SLAM framework for large scale mapping.

A. Algorithm Overview

We perform loop closure detection from a pair of 2D laser scans composed of range and bearing data. Our loop detection algorithm uses the same principle as standard scan matching algorithms; loops are detected by comparison of laser scans. The main difference between our algorithm and traditional scan matching approaches is the introduction of rotation invariant features describing the laser scans. These features are combined in a non-linear manner using a boosting classifier which outputs the likelihood of the two scans being matched.

Figure 2 presents a diagram with the stages of the algorithm. In the learning phase, pairs of laser scans and the corresponding assignments (match or non-match) are input to AdaBoost. From the laser points, rotation invariant features are initially extracted. Examples of the features employed are length, area, curvature of the scan, etc. (a detailed description of the features is presented in the next subsection). AdaBoost greedily builds a strong classifier by a linear combination of simpler, weak, classifiers. In our implementation these classifiers are decision stumps which provide very nonlinear decision boundaries. The same strategy has been employed for face detection in [16]. This procedure notoriously enhances the capabilities of the resulting classifier. As more decision stumps are added, the classification error on the training data goes to zero. Although

this might be interpreted as overfitting, [1] shows that it also generalises well on testing data.

Once the classifier has been built, loop closure detection can be performed in a SLAM framework as Figure 2 (right) indicates. If a loop closure is detected, a laser scan alignment procedure is performed with a corresponding update in the map. We describe the particular SLAM framework employed in Section IV.

B. Laser Features

The laser range sensors used in experiments have a 180 degree field of view. The sensors deliver scans $\boldsymbol{l} = \{r_i, \alpha_i\}_{i=1}^{N}$ where r_i is range, α_i is bearing, and N is the number of laser returns in the scan. Together, a forward scan and a backward scan create $\mathbf{L}$, which gives a full 360 degree view of the surroundings. A laser scan can be described in Cartesian coordinates $\mathbf{L} = \{\mathbf{x}_i\}_{i=1}^{N} = \{x_i, y_i\}_{i=1}^{N}$, where $x_i = r_i \cos(\alpha_i)$ and $y_i = r_i \sin(\alpha_i)$. We use 360 degree laser scans to achieve a general solution that is invariant to rotation. One might think that considering 360 degree laser scans guarantees rotation invariance, however as two laser scans are rarely taken in the exact same position that is not the case. We also show that the method works in the 180 degree case as well, although in that case loop closure can only be detected from the same direction.

A 360 degree laser scan consists of 722 laser points, and a data set can contain up to several hundred thousand laser scans. Hence there is a need to reduce the size of the data dimension. The method of reducing the dimension we use is to employ features, thus reducing the dimension of each laser scan to the number of features used. A feature f is defined as a function that takes a laser scan $\mathbf{L}$ and returns a real value. In this paper we are interested in features that describe different geometric properties of the laser scan, such as the area covered by the scan, the average range, the circularity of the scan and the sum of the distances between consecutive points.

We use 20 features in this paper, and thus each laser scan is compressed into a feature vector $\mathbf{f}$ of length 20. Some of the features we use are closely related, e.g. $\#1$ and $\#4$ as well as $\#5$, $\#10$ and $\#11$. It is not easy to discern why the use of all of them is better than just using e.g. $\#1$ and $\#10$. However we do not concern ourselves with that problem, instead we feed all 20 features to AdaBoost. Given the training data, finding out which features are better than others, and how to best combine them, is a task performed entirely by AdaBoost.

The laser range sensors time out after a certain distance determined by the sensor itself. Therefore a maximum range gate r_{max} is used in some features to remove measurements whose range is too long. In experiments we set $r_{\text{max}} = 50$ metres. Some of the features described here were also employed in [14]. All features listed below are invariant to rotation:

1) Area: Measures the area covered by a laser scan. Points whose range is greater than r_{max} have their range set to r_{max}.

$$f_{\text{area}} = \sum_{i:1}^{N-1} r_i r_{i+1} \sin\left(\frac{\alpha_{i+1} - \alpha_i}{2}\right) \tag{1}$$

2) Average Range: Measures the average range of a scan. Ranges greater than or equal to r_{max} are set equal to r_{max}.

$$f_{\text{average range}} = \frac{1}{N} \sum_{i=1}^{N} \min\left(r_i, r_{\text{max}}\right) \tag{2}$$

3) Centroid: Measures the distance from the origin to the mean position. This feature captures whether the laser points are concentrated to one part of the euclidean space around the robot (longer distance), or are scattered evenly around the robot (shorter distance). The mean position is calculated as

$$\begin{bmatrix} x_{\text{mean}} \\ y_{\text{mean}} \end{bmatrix} = \begin{bmatrix} \frac{1}{N} \sum_{i:\ r_i < r_{\text{max}}} x_i \\ \frac{1}{N} \sum_{i:\ r_i < r_{\text{max}}} y_i \end{bmatrix}. \tag{3}$$

The distance to the origin is then calculated as

$$f_{\text{mean centroid}} = \sqrt{x_{\text{mean}}^2 + y_{\text{mean}}^2} \tag{4}$$

4) Close Area: Measures the area covered by the laser scan, excluding the area covered by range measurements whose range is greater than or equal to r_{max}. This feature will be significantly different from feature $\#1$ if the robot is standing in an open area, e.g. a field.

$$f_{\text{close area}} = \sum_{i:\ r_i < r_{\text{max}}} r_i^2 \sin\left(\frac{\delta_\alpha}{2}\right), \tag{5}$$

where δ_α is the angle interval at which range measurements are taken. If 361 range measurements are acquired in a 180 degree field of view, then $\delta_\alpha = \frac{180}{361-1}$ degrees.

5) Close Distance: Measures the sum of the distances between consecutive points whose range is smaller than r_{max}, excluding distances that are larger than a maximum distance gate, $g_{\text{max dist}}$. In experiments we set $g_{\text{max dist}} = 2.5$ metres.

$$d_i = \|\mathbf{x}_i - \mathbf{x}_{i+1}\| \qquad i:\ r_i, r_{i+1} < r_{\text{max}} \tag{6a}$$

$$f_{\text{close dist}} = \sum_{j:\ d_j < g_{\text{max dist}}} \left(d_j\right) \tag{6b}$$

where $\|.\|$ is defined as the Euclidean distance.

6) Circularity Radius: The circularity feature fits a circle to the points in the laser scan, whose range is smaller than r_{max}, in a least squares sense. This returns a centre point x_c, y_c and a range r_c for the fitted circle. The value of the feature is the radius of the circle r_c.

7) Circularity Residual: This feature is defined as the residual sum of squares, after fitting a circle to the points as with the previous feature:

$$f_{\text{circularity}} = \sum_{i:\ r_i < r_{\text{max}}} \left(r_c - \sqrt{(x_c - x_i)^2 + (y_c - y_i)^2}\right)^2 \tag{7}$$

8) Curvature Mean: The curvature features are based on the curvature along the points in the laser scan. Let $\mathbf{x}_a = [x_a, y_a]^T$, $\mathbf{x}_b = [x_b, y_b]^T$ and $\mathbf{x}_c = [x_c, y_c]^T$ be three consecutive points, let A be the area covered by the triangle with corners in $\mathbf{x}_a$, $\mathbf{x}_b$ and $\mathbf{x}_c$, and let d_a, d_b and d_c be the distances between the points. The curvature of the boundary at $\mathbf{x}_b$ is calculated as

$$k = \frac{4A}{d_a d_b d_c} \tag{8}$$

The curvatures over all points, excluding points whose range is greater than or equal to $r_{\max}$, are calculated. This feature returns the mean value of the curvatures.

9) Curvature Standard Deviation: This feature is defined as the standard deviation of the curvatures computed above.

10) Distance: Measures the sum of the distances between consecutive points, excluding points whose range is greater than or equal to $r_{\max}$.

$$f_{\text{dist}} = \sum_{i:\ r_{\{i,i+1\}} < r_{\max}} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} \tag{9}$$

11) Far Distance: Measures the sum of the distances between all consecutive points, i.e. including points whose range is greater than or equal to $r_{\max}$.

$$f_{\text{far dist}} = \sum_{i=1}^{N-1} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} \tag{10}$$

12) Number of Groups: This feature measures the number of groups (clusters) in the scan. A group is defined as a cluster of laser points in which the distance between consecutive points is less than a maximum distance gate $g_{\text{max dist}}$, also used in feature $\#5$. To be considered a group, the cluster has to contain more than a certain number of points specified by the minimum group size gate $g_{\text{min size}}$. In experiments we set $g_{\text{min size}} = 3$ points.

13) Mean Group Size: This feature is defined as the mean group size after detecting and clustering the laser points into groups as in the previous feature.

14) Maximum Range: Measures the number of points in the laser scan whose range is greater than or equal to the maximum range gate.

$$f_{\text{max range}} = \sum_i \mathbf{1}\left\{r_i \geq r_{\max}\right\}, \tag{11}$$

where

$$\mathbf{1}\left\{r_i \geq r_{\max}\right\} = \begin{cases} 1 & \text{if } r_i \geq r_{\max} \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

15) Mean Angular Difference: Measures the sum of the angles between consecutive point to point vectors. Given two consecutive laser points L_i and L_{i+1}, a vector that connects the points is given as $\bar{\mathbf{x}}_{i,i+1} = [x_{i+1} - x_i\ ,\ y_{i+1} - y_i]^T$. The feature is calculated as

$$f_{\text{MAD}} = \sum_{i:r_{\{i,i+1,i+2\}} < r_{\max}} \arccos\left(\frac{\bar{\mathbf{x}}_{i,i+1}^T \bar{\mathbf{x}}_{i+1,i+2}}{||\bar{\mathbf{x}}_{i,i+1}||\ ||\bar{\mathbf{x}}_{i+1,i+2}||}\right). \tag{13}$$

16) Mean Deviation: Measures the mean deviation from the mean of the laser scan. The feature is calculated as

$$f_{\text{mean deviation}} = \frac{1}{N} \sum_{i:r_i < r_{\max}} \sqrt{(x_i - x_{\text{mean}})^2 + (y_i - y_{\text{mean}})^2}, \tag{14}$$

where x_{mean} and y_{mean} is calculated as in (3).

17) Regularity: Measures the regularity of the laser scan, which is defined as the standard deviation of the distances between consecutive points in the laser scan. Laser points whose range is greater than or equal to $r_{\max}$ are excluded.

Let $d_{i,i+1}$ be the distance between the laser points with indices i and $i+1$, and let $\bar{d}$ be the mean value of $d_i,\ \forall\ i : r_i < r_{\max}$. The regularity feature is then calculated as

$$f_{\text{regularity}} = \sqrt{\frac{1}{N-1} \sum_{i:\ r_{\{i,i+1\}} < r_{\max}} \left(d_{i,i+1} - \bar{d}\right)^2} \tag{15}$$

18) Size: Measures the number of points which has a range shorter than $r_{\max}$.

$$f_{\text{Size}} = \sum_i \mathbf{1}\left\{r_i < r_{\max}\right\}, \tag{16}$$

where

$$\mathbf{1}\left\{r_i < r_{\max}\right\} = \begin{cases} 1 & \text{if } r_i < r_{\max} \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

19) Standard Deviation of Distance to Mean: Measures the standard deviation of the point-wise distances to the mean position. The mean position is calculated as in (3), and the distance from point i to the mean is

$$d_{i,mean} = \sqrt{(x_i - x_{mean})^2 + (y_i - y_{mean})^2}. \tag{18}$$

The feature is given as the standard deviation of $d_{i,mean}$ for $i : r_i < r_{max}$.

20) Standard Deviation of Range: Measures the standard deviation of all the ranges that are less than or equal to $r_{\max}$. The feature is calculated as

$$f_{\text{std range}} = \frac{1}{N-1} \sum_{i:r_i < r_{\max}} \sqrt{(r_i - r_{\text{mean}})}, \tag{19}$$

where r_{mean} is the mean of all the ranges that are less than or equal to $r_{\max}$.

These 20 features are computed for both scans in the pair and the absolute difference between them is passed to the classifier in the next step. Given two scans k and $k+1$, the set of extracted features is $\mathbf{f}(\mathbf{L}^k, \mathbf{L}^{k+1}) = \left[f_1(\mathbf{L}^k, \mathbf{L}^{k+1}), \ldots, f_{20}(\mathbf{L}^k, \mathbf{L}^{k+1})\right]$, where $f_i(\mathbf{L}^k, \mathbf{L}^{k+1}) = \|f_i(\mathbf{L}^k) - f_i(\mathbf{L}^{k+1})\|$.

C. Classification and Boosting

We briefly review boosting in this section. As training data, n pre-labeled laser pairs are provided,

$$\left(\mathbf{f}(\mathbf{L}_1^1, \mathbf{L}_1^2), y_1\right), \ldots, \left(\mathbf{f}(\mathbf{L}_n^1, \mathbf{L}_n^2), y_n\right), \tag{20}$$

where y_i is a binary variable, $y_i = \{0, 1\}$ for negative (non-matching) and positive (matching) laser pairs, respectively and $\mathbf{f}$ is a set of features. Let N_n and N_p denote the number of negative pairs and positive pairs respectively. AdaBoost is an iterative procedure that consecutively adds weak classifiers to a set of previously added weak classifiers to find a good combination that constitutes a strong classifier. The weak classifiers adopted are decision stumps defined as:

$$c(\mathbf{f}(\mathbf{L}_i^m, \mathbf{L}_i^n), \theta) = \begin{cases} 1 & \text{if } pf < p\lambda \\ 0 & \text{otherwise} \end{cases} \quad (21)$$

with parameters $\theta = \{f, p, \lambda\}$, where p is the polarity ($p = \pm 1$), f is the particular feature selected and λ is a threshold.

To add a new weak classifier to the set, the training data is classified using the set of previously added weak classifiers. The weak classifier that improves the classification the most is added to the set of weak classifiers. The training data is weighted to ensure that the newly added classifier was the one that minimized the misclassified data the most. After the classifier has been added, the weights are updated. The procedure is repeated until T weak classifiers have been added. Each weak classifier can be added several times, each time with a new threshold. The set of T weak classifiers together create the strong classifier. AdaBoost is described in Algorithm 1.

IV. Simultaneous Localisation and Mapping

The section before presented the procedure used to associate scans. In order to build a global map of the environment we need to build a framework that stores the information acquired during the data collection process. We use a SLAM algorithm based on a Exactly Sparse Delayed-state Filter (ESDF) [17]. Each pose in the trajectory based state vector is associated to a laser scan acquired at that location. The classifier presented before is used to detect loop closures between poses. Odometry and relative pose estimation after loop closure detection (difference in position and heading) are calculated using laser scan alignment.

Once we have detected an association between scans, an alignment process estimates the sensor displacement. A Conditional Random Field-match (CRF-match) [4] followed by Iterative Closest Point (ICP) is used for the scans' alignment. The ICP algorithm [5] is used to refine the scan alignment result obtained by the CRF-match.

A. Exactly Sparse Delayed-state Filters

The ESDF maintains a delayed state vector containing the poses of the vehicle's trajectory. The state vector is augmented with a new pose when a new laser scan is acquired. In information form the information matrix is sparse without approximation, which results in an estimation comparable to the full covariance matrix solution while prediction and update can be performed in constant time regardless of the information matrix size.

B. Laser Scan Alignment

CRF-match [4] is a feature based probabilistic method that finds the most likely of all point to point associations between two laser scans. The method can align scans without the need for an initial guess of the alignment. ICP can give an alignment with better quality, however the square cost function minimized by ICPcontains many local minima. Therefore the algorithm requires good initialisation to ensure correct convergence. CRF-match followed by ICP gives a very robust alignment process.

Algorithm 1 AdaBoost

Input: $\left(\mathbf{f}(\mathbf{L}_1^1, \mathbf{L}_1^2), y_1\right), \ldots, \left(\mathbf{f}(\mathbf{L}_n^1, \mathbf{L}_n^2), y_n\right)$

Initialize weights: $W_1^i = \frac{1}{2N_n}$ if $y_i = 0$, $W_1^i = \frac{1}{2N_p}$ if $y_i = 1$

1: **for** $t = 1, \ldots, T$ **do**

2: Normalise the weights:

$$\tilde{W}_t^i = \frac{W_t^i}{\sum_{j=1}^{N_n+N_p} W_t^j}, \quad i = 1, \ldots, N_n + N_p \quad (22)$$

3: Select the best weak classifier, i.e. the one that minimizes the weighted error:

$$\epsilon_t = \sum_{i=1}^{n} \tilde{W}_t^i \left| c(\mathbf{f}(\mathbf{L}_i^1, \mathbf{L}_i^2), \theta) - y_i \right| \quad (23)$$

4: Define $c_t(\mathbf{f}(\mathbf{L}^1, \mathbf{L}^2)) = c(\mathbf{f}(\mathbf{L}^1, \mathbf{L}^2), \theta_t)$ where θ_t is the minimizer of ϵ_t.

5: Update the weights:

$$W_{t+1}^i = \tilde{W}_t^i \beta_t^{1-e_i}, \quad (24)$$

where $e_i = 0$ if $\mathbf{f}(\mathbf{L}_i^1, \mathbf{L}_i^2)$ is classified correctly and 1 otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.

6: **end for**

The strong classifier is:

$$\mathbf{c}\left(\mathbf{f}(\mathbf{L}_i^1, \mathbf{L}_i^2)\right) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t c_t \left(\mathbf{f}(\mathbf{L}_i^1, \mathbf{L}_i^2)\right) \geq K \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases} \quad (25)$$

where $K \in [0, 1]$ and $\alpha_t = \log \frac{1}{\beta_t}$.

Output: $\mathbf{c}\left(\mathbf{f}(\mathbf{L}_i^1, \mathbf{L}_i^2)\right)$

C. Vehicle Motion Model

The vehicle motion model is

$$\mathbf{x}_v(t_{k+1}) = \mathbf{x}_v(t_k) \oplus \mathbf{u}(t_{k+1}) \quad (26a)$$

$$= \begin{bmatrix} x_v(t_k) \\ y_v(t_k) \\ \phi_v(t_k) \end{bmatrix} \oplus \begin{bmatrix} u_1(t_{k+1}) \\ u_2(t_{k+1}) \\ u_3(t_{k+1}) \end{bmatrix} \quad (26b)$$

$$= \begin{bmatrix} x_v + u_1 \cos(\phi_v) - u_2 \sin(\phi_v) \\ y_v + u_1 \sin(\phi_v) + u_2 \cos(\phi_v) \\ \phi_v + u_3 \end{bmatrix}, \quad (26c)$$

where time indices have been omitted in the last row for the sake of brevity. $\mathbf{x}_v(t_k)$ is the current vehicle pose at time t_k, $\oplus$ is the compounding operator from [18]. In our implementation, the input signal $\mathbf{u}(t_k) = [u_1(t_k)\ u_2(t_k)\ u_3(t_k)]^T$ corresponds to translation and rotation, calculated from alignment of consecutive laser scans using ICP.

V. Experimental Results

We performed experiments using data from four data sets. The first two data sets were collected along residential and business streets in the vicinity of the University of Sydney, Australia. Both data sets were acquired during day time and contain moving objects such as cars and people. The data sets are approximately 0.65 and 2 kilometres long.

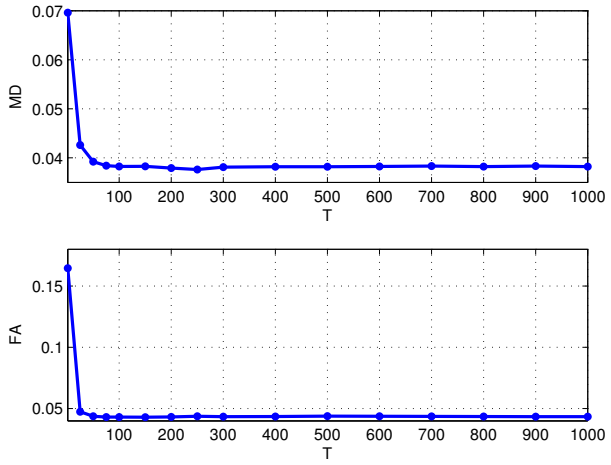


Fig. 3. Error rates for different values of T. The values of T used are marked with dots, the steady error levels of around 4% suggest that there is no clear overfitting.

TABLE I
BEST FEATURES FOR LOOP CLOSING

TEST 1						
Training Round	1	2	3	4	5	50
Added Feature	1	4	15	12	10	...
Total Error [%]	12.0	12.0	9.5	8.6	8.0	4.0
TEST 2						
Feature Removed	12	4	15	1	3	17
Total Error [%]	4.44	4.39	4.39	4.38	4.30	4.16

The third data set was obtained from the Robotics Data Set Repository (Radish) [19][1]. This data set was collected in Kenmore, QLD, Australia. It is about 18 kilometres long. From these three data sets we identified a set of 400 matching and 400 non-matching laser pairs. The fourth data set, also collected around the University of Sydney, is approximately 2 kilometres long. It was used in a SLAM experiment, where we also used GPS to collect ground truth data.

10-fold cross validation was used to estimate the false alarm and missed detection error rates. The results from each of the ten folds are pooled together. As the shuffling of the laser pairs have a slight impact on the results, 10-fold cross validation can be performed several times, each time with a new order. The results from all 10-fold cross validations are then averaged. Unless otherwise stated, all error rates are estimated from 100 10-fold cross validations.

To determine a good number of training rounds, we trained strong classifiers for T between 1 and 1000 rounds. The error rates for each T_i are shown in Figure 3. Since the error rates remains approximately constant after $T = 50$, we choose to train the strong classifier for 50 rounds in all our tests. A lower number of training rounds is preferred, since the computation time for classification of laser pairs increases when more features are added to the strong classifier. Figure 3 also suggests that overfitting is not a concern for the task.

[1]Thanks to Michael Bosse for providing the data set.

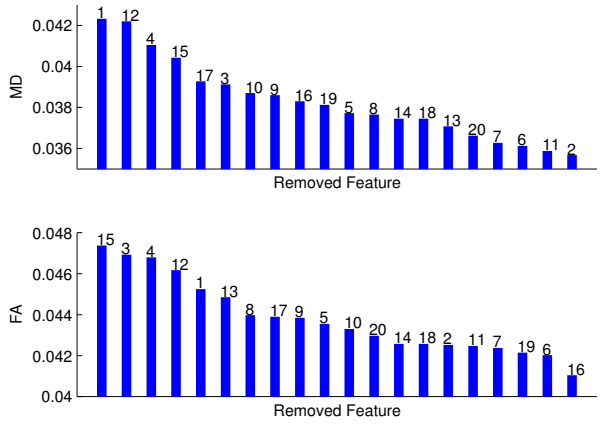


Fig. 4. Missed Detection and False Alarm error rates after feature have been removed one at a time.

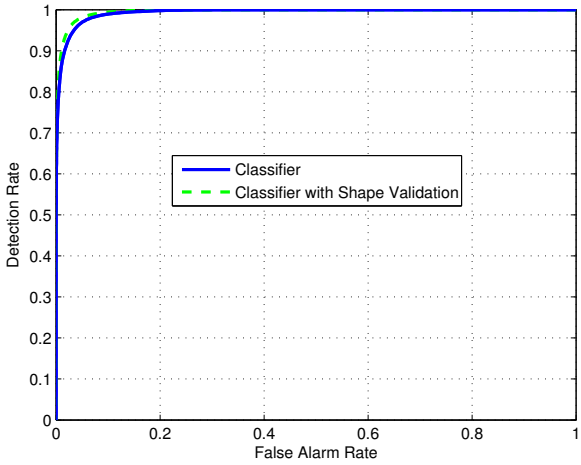


Fig. 5. The blue solid graph shows the ROC curve for the strong classifier, and the green dashed graph shows the ROC curve for the strong classifier combined with scan alignment and shape validation.

A. Loop Closure Feature Analysis

During training, AdaBoost selects the best feature in each training iteration. By examining which features are chosen earlier during training, it can be determined which are the most significant features for classification. Table I shows results for two tests using the 800 pre-labeled laser pairs. The table presents total error rates, i.e. the sum of false alarm and missed detection. The error rates can be compared to blind guessing which would yield a 50% error rate.

1) Test 1: A strong classifier is trained and Table I shows which features are chosen in the first rounds, and how the error rate decreases as more features are added. The first two added features, #1 and #4, correspond to Area and Close area features. They represent the most informative features and are closely related since both are area measures of the scan. The

third feature, #15, is Mean Angular Difference, and the fourth and fifth, #12 and #10, are Mean Group Size and Distance. As can be seen in Table I, the reduction in error rate decreases as more features are added.

2) Test 2: For this test we start by training a strong classifier, the False Alarm and Missed Detection rates are estimated to 4.26% of the 400 non-matching pairs and 3.75% of the 400 matching pairs, giving a total error rate of 4.0% of the 800 pairs. We then proceed to remove each feature, one at a time, and train new classifiers on the remaining 19 features. Results from this are presented in Figure 4, where the indices on top of the bars denote the eliminated features.

Figure 4 shows that removing feature #1 increases Missed Detection rate the most, and removing feature #15 increases the False Alarm rate the most. If total error is considered, removing feature #12 has the largest negative impact. Results for the 6 features whose removal have the most negative impact on total error are presented under Test 2 in Table I.

The four features chosen first in Test 1, #1, #4, #15 and #12, also have the most negative impact on the Missed Detection rate, and together with feature #3 have the most negative impact on the False Alarm rate.

B. Loop Closure Results

The two most important characteristics for a classifier are false alarm and detection rates. We examined the two rates for different match thresholds by changing K in Eq. (25). The detection and false alarm rates for each threshold are estimated using 400 10-fold cross validations on the set of 800 pre-labeled data pairs.

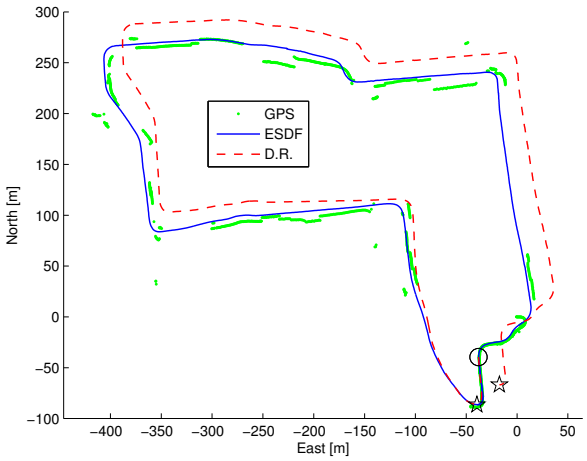
1) Classification Accuracy: We measure the accuracy of the resulting classifier using the area under the Receiver Operating Characteristic (ROC) curve. The ROC curve is shown as the solid blue curve in Figure 5. A threshold $K = 0.59$ gives a false alarm rate of 1% and a detection rate of 85%. The area under the curve is approximately 0.99.

The classifier's invariance to rotation was tested on a large set of laser scan pairs. Each pair was initially classified, then one of the laser scans was rotated arbitrarily between 90 and 180 degrees and the pair was classified again. Out of 50451 laser scan pairs, 98.4% received the same classification as in the previous case.

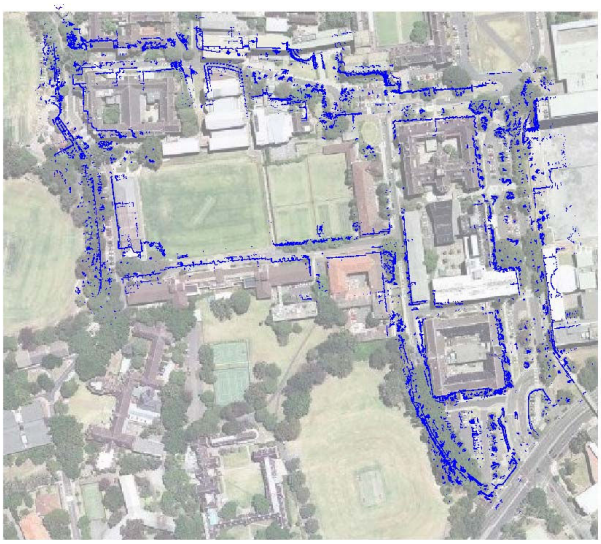
2) Shape Validation Supported Classifier: The false alarm rate is further reduced when the classifier is combined with laser scan alignment using CRF-match, ICP and shape validation. Shape validation evaluates the laser scan alignment by finding the percentage of nearest neighbour point pairs that fall within a certain distance d. If the number is above a threshold $N\%$, the validation test is passed.

In this setting, a loop closure is accepted if a pair of scans is classified as a match and the computed alignment passes the shape validation test. This, however, will also decrease the detection rate so the shape validation thresholds must be a compromise between false alarm rate and detection rate. Empirically, we have found that $N = 90\%$ and $d = 1$ metre works well in the present application. A shape validation supported classifier with a threshold $K = 0.57$ gives a false alarm rate of 1% with a detection rate of 89%. In Figure 5, the dashed green curve is a ROC curve for the same 400 cross validations as were used to draw the solid blue curve. The area under the green ROC is just over 0.99.

3) Time Complexity: Our implementation classifies 800 pairs of laser scans in just under 32 seconds, on average 0.04 seconds per pair. About 95% of the computation time is spent calculating the feature values, which in a SLAM setting only has to be performed once per laser scan. The time spent by the



(a)



(b)

Fig. 6. (a) Estimated vehicle trajectory, GPS and dead reckoning. The ring marks the starting point, the stars mark the end points. (b) Laser map overlaid on an aerial photograph. Each laser scan was transformed to its respective pose and plotted on top of the photograph.

classifier (without feature extraction) averages at only 0.002 seconds per pair.

C. SLAM *Experiment*

For the laser based SLAM experiment, a strong classifier was trained on the second data set which contains forward and backward facing laser scans. However, the fourth data set used for localisation and mapping only contains forward laser scans.

The resulting state vector contains 1800 augmented poses, each one associated to a laser scan. In the information matrix, 98.5% of the $(1800 * 3)^2$ elements are exactly zero, and thus the matrix is highly sparse. In total, 85759 pairs of laser scans were tested against each other, out of which 85 were classified as matching. 42 were correct loop closures, the remaining 43 were false alarms giving a false alarm rate of 0.05%. All false alarms were refused by the shape validation, i.e. no incorrect updates were made in the filter. There were a few missed detections due to occlusion from other vehicles. Vehicle movement was estimated by the alignment of consecutive laser scans using ICP. The estimated trajectory is compared to GPS (estimated ground truth) and SLAM with only dead reckoning (without loop closure detection) in Figure 6a. The performance with our loop closure detection; ESDF in Figure 6a, is clearly better than the performance without it; D.R. in Figure 6a.

A laser map from the data set is overlaid on an aerial photograph in Figure 6b. The map shows a good fit to the image.

Another interesting observation is that the laser matching method, designed and trained for a full 360 degree view, performs well in the 180 degree view.

VI. Conclusions

This paper presented a machine learning procedure for loop closure detection. Features invariant to viewpoint were designed and combined into a boosting classifier. Using the proposed method, laser scans can be correctly matched regardless of the alignment, enabling loop closure detection from arbitrary directions. The classifier performance is encouraging, with good detection rates for low false alarm rates. Additionally, a SLAM experiment demonstrates that reliable localisation and mapping can be achieved in a complex outdoor environment using our framework. The classifier, designed and trained for 360 degree laser scans, performs well even if only 180 degree laser scans are available.

The work by Bosse *et al* [8] is, to the best of our knowledge, the largest results for laser scan-based SLAM. While our maps are smaller in size than their maps, at a false alarm rate of 1% we achieve a detection rate of 85% compared to their lower rate of 51%.

References

[1] Y. Freund and R. Shapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *Proceedings of European Conference on Computational Learning Theory*, Barcelona, Spain, 1995.

[2] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *In Proceedings of the Thirteenth International Conference on Machine Learning*. Morgan Kaufmann, 1996, pp. 148–156.

[3] R. E. Schapire, "Theoretical views of boosting," in *Computational Learning Theory: 4^{th} European Conf., EuroCOLT'99*, 1999, pp. 1–10.

[4] F. T. Ramos, D. Fox, and H. F. Durrant-Whyte, "Crf-matching: Conditional random fields for feature-based scan matching," in *Proceedings of Robotics: Science and Systems*, Atlanta, USA, 2007.

[5] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," *Image Vision Comput.*, vol. 10, no. 3, pp. 145–155, 1992.

[6] J. S. Gutmann and K. Konolige, "Incremental mapping of large cyclic environments," in *Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation*, 1999.

[7] D. Hahnel, W. Burgard, D. Fox, and S. Thrun, "An efficient fastslam algorithm for generating maps of large-scale cyclic environments from raw laser range measurements," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003.

[8] M. C. Bosse and R. Zlot, "Map matching and data association for large-scale two-dimensional laser scan-based slam," *International Journal of Robotics Research*, vol. 27, no. 6, pp. 667–691, 2008.

[9] P. Newman, D. Cole, and K. Ho, "Outdoor slam using visual appearance and laser ranging," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2006.

[10] J. Nieto, T. Bailey, and E. Nebot, "Recursive scan-matching slam," *Jrnl. of Robotics and Autonomous Systems*, vol. 55, no. 1, pp. 39–49, 2007.

[11] K. Ho and P. Newman, "Combining visual and spatial appearance for loop closure detection in slam," in *Proceedings of European Conference on Mobile Robots (ECMR)*, Ancona, Italy, September 2005.

[12] F. T. Ramos, J. Nieto, and H. F. Durrant-Whyte, "Recognising and modelling landmarks to close loops in outdoor slam," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, Rome, Italy, 2007.

[13] M. C. Bosse and J. Roberts, "Histogram matching and global initialization for laser-only slam in large unstructured environments," *Proceedings of IEEE International Conference on Robotics and Automation*, 2007.

[14] K. O. Arras, O. M. Mozos, and W. Burgard, "Using boosted features for the detection of people in 2d range data," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2007.

[15] O. M. Mozos, C. Stachniss, and W. Burgard, "Supervised learning of places from range data using adaboost," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2005.

[16] P. Viola and M. Jones, "Robust real-time object detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[17] S. Thrun, Y. Liu, D. Koller, A. Ng, Z. Ghahramani, and H. Durrant-Whyte, "Simultaneous localization and mapping with sparse extended information filters," *International Journal of Robotics Research*, vol. 23, no. 7–8, pp. 693–716, 2004.

[18] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," *Autonomous Robot Veh.*, pp. 167–193, 1990.

[19] A. Howard and N. Roy, "The robotics data set repository (radish)," 2003. [Online]. Available: http://radish.sourceforge.net/