# Tree of Words for Visual Loop Closure Detection in Urban SLAM

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

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

## Abstract

This paper introduces vision based loop closure detection in Simultaneous Localisation And Mapping (SLAM) using Tree of Words. The loop closure performance in a complex urban environment is examined and an additional feature is suggested for safer matching. A SLAM ground experiment in an urban area is performed using Tree of Words, a delayed state information filter and planar laser scans for relative pose estimation. Results show that a good map estimation using our vision based loop closure detection can be obtained in near real, yet constant, time. It is shown that an odometry supported recall rate of almost 70% can be obtained with a false detection rate of about 0.01%.

## 1 Introduction

The Simultaneous Localisation And Mapping (SLAM) problem is one of the most central in robotics research [Durrant-Whyte and Bailey, 2006; Bailey and Durrant-Whyte, 2006]. It asks if a mobile robot, put in an unknown location, can incrementally build a consistent map of the environment and simultaneously determine its location within this map. A general solution is required to enable truly autonomous robots.

In SLAM, the ability to detect that a previously mapped area is being revisited, a so called loop closure, is crucial, since it enables an adjustment of the environment mapped after the area was last visited. A loop closure is usually detected either by identifying a previously specified landmark [Bailey, 2001; Davison *et al.*, 2004; Nieto *et al.*, 2005; Ramos *et al.*, 2007b] or by acknowledging the similarity between two sceneries [Gutmann and Konolige, 1999; Ho and Newman, 2005]. In this paper we use the latter, detecting loop closures using vision based scene similarity detection. Since the images are treated as a scenery, two images must be taken from the same area facing the same way for a loop closure to be detected.

Tree of Words (ToW) was first proposed by Nistér and Stewénius [Nistér and Stewénius, 2006] as a way of finding the closest match of an image in a large database. We propose using Tree of Words for loop closure detection in a SLAM algorithm.

Tree of Words was introduced as a hierarchical approach to Bag of Words [Sivic and Zisserman, 2003], which suggests how an image can be represented by predefined features for fast database querie. The image is simply compressed into a vector of the predefined features, or words, it contains. Bag of Words compares images by matching clusters of words from a small vocabulary, i.e. set of predefined words. Tree of Words on the other hand uses a larger vocabulary and no clustering. The latter showed promising results and had significant speed advantages, making it appropriate for loop closure detection in a dense urban environment. Bag of Words has been used in numerous SLAM implementations [Cummins and Newman, 2008; Newman *et al.*, 2006; Ho and Newman, 2005; Angeli *et al.*, 2008]. In these experiments, loop closure detection is performed without exploiting the estimated vehicle location.

After a loop closure has been detected, the relative difference in position and heading is calculated by aligning laser scans of the surroundings, acquired where the images were taken. Nieto *et al* have studied laser based SLAM using Iterative Closest Point (ICP) for scan alignment [Nieto *et al.*, 2007]; Ramos *et al* suggest aligning the scans with Conditional Random Fields (CRF) [Ramos *et al.*, 2007a]. In ICP a mean square cost function is minimized to align the scans, while CRF estimates the conditional probability of the joint association of the scans. We use CRF to get an initial alignment followed by ICP to adjust the result. The alignment cannot be performed by ICP exclusively, since the cost function has many local minima resulting in misalignments.

## 2 SLAM

### 2.1 Filter Options

Slighly simplified, there are two main approaches to a SLAM solution. The first approach is landmark based SLAM where the map consists of landmarks such as trees and houses. The state vector maintains the robot state and each landmark's position. Consequently, the dimensionality of the state vector grows with the number of landmarks being observed. Loop closures are detected by correctly associating a currently observable landmark to a landmark previously mapped at an earlier stage of the experiment.

The second approach is trajectory based SLAM where the state vector consists of the current vehicle state and past vehicle states, called a delayed state vector. Each vehicle state is associated to a measurement of the spatial appearence of the surrounding area. Loop closures are detected by associating the spatial appearance of the current surroundings to a previous one. After a loop closure has been detected, the estimated trajectory is adjusted. Note that no map features are included in the state vector. Instead the environment is represented by the measurements associated to each pose.

In landmark based SLAM two solutions stand out; the Extended Kalman Filter solution, called EKF-SLAM, and the particle filter based solution, called FastSLAM [Montemerlo *et al.*, 2002]. Both have been used in numerous experiments and have been proven to work both in- and outdoors. In an urban environment maintaining a landmark based solution is challenging due to the complexity of the environment. The number of landmarks grows rapidly and the computations scales as the square of the number of landmarks for EKF-SLAM [Durrant-Whyte and Bailey, 2006].

Eustice *et al* [Walter *et al.*, 2007; Eustice *et al.*, 2006] have examined delayed state filters for view-based SLAM. They show that the information matrix is exactly sparse if the state vector is in delayed state format. This results in constant execution time for time and measurement updates, i.e. an upscaling of the experiment does not result in a slower filter. The filter is called an Exactly Sparse Delayed-state Filter (ESDF) and is the filter used in this paper. Exactly sparse refers to the fact that the vast majority of the elements in the information matrix are exactly zero. Other information filters, such as the Sparse Extended Information Filter (SEIF) [Thrun *et al.*, 2005], approximate small elements in the information matrix as zero, resulting in a sparse matrix. ESDF incurrs no sparse approximation errors, producing results that can be compared to the full covariance matrix solution. ESDF has previously been used in outdoor SLAM by Newman *et al* [Cole and Newman, 2006; Newman *et al.*, 2006].

A different approach to large scale SLAM is using submaps [Estrada *et al.*, 2005; Guivant and Nebot, 2002; Chong and Kleeman, 1999]. Using submaps in the ATLAS framework proposed by Bosse has been shown to work well in a large scale urban environment [Bosse and Roberts, 2007].

### 2.2 Exactly Sparse Delayed-state Filter

ESDF utilises a delayed state vector containing a history of past vehicle poses (positions and headings).

The vehicle pose vector is

$$\mathbf{x}(t_k) = \begin{bmatrix} x(t_k) \\ y(t_k) \\ \phi(t_k) \end{bmatrix} \tag{1}$$

where $x(t_k)$ and $y(t_k)$ are the vehicle's position in a two dimensional plane, and $\phi(t_k)$ is its heading in the same plane. Vehicle pose uncertainty is represented by the covariance matrix

$$\mathbf{P}(t_k) = \begin{bmatrix} \sigma_{xx}(t_k) & \sigma_{xy}(t_k) & \sigma_{x\phi}(t_k) \\ \sigma_{yx}(t_k) & \sigma_{yy}(t_k) & \sigma_{y\phi}(t_k) \\ \sigma_{\phi x}(t_k) & \sigma_{\phi y}(t_k) & \sigma_{\phi\phi}(t_k) \end{bmatrix}. \tag{2}$$

The current delayed state vector $\mathbf{X}(t_k)$ and its associated covariance matrix $\mathbf{P}(t_k)$ have the following format

$$\mathbf{X}(t_k) = \begin{bmatrix} \mathbf{x}_t(t_k) \\ \mathbf{x}_v(t_k) \end{bmatrix},$$

$$\mathbf{P}(t_k) = \begin{bmatrix} \mathbf{P}_{tt}(t_k) & \mathbf{P}_{tv}(t_k) \\ \mathbf{P}_{vt}(t_k) & \mathbf{P}_{vv}(t_k) \end{bmatrix},$$

where $\mathbf{x}_t(t_k)$ is the history of past vehicle poses augmented at times $t_0$ to $t_N$, $[\mathbf{x}_{t_0}(t_k) \; \ldots \; \mathbf{x}_{t_N}(t_k)]^T$, and $\mathbf{x}_v(t_k)$ is the current vehicle pose. All vehicle poses and uncertainties, past and current, have the format given by (1) and (2).

Prediction only introduces connections between the current pose and the previous pose, producing an information matrix with elements along a tri-diagonal,

$$\mathbf{\Lambda}(t_k) = \begin{bmatrix} \mathbf{\Lambda}_{t_0t_0}(t_k) & \mathbf{\Lambda}_{t_0t_1}(t_k) & & & \\ \mathbf{\Lambda}_{t_1t_0}(t_k) & \mathbf{\Lambda}_{t_1t_1}(t_k) & \mathbf{\Lambda}_{t_1t_2}(t_k) & & \\ & \mathbf{\Lambda}_{t_2t_1}(t_k) & \mathbf{\Lambda}_{t_2t_2}(t_k) & \mathbf{\Lambda}_{t_2t_3}(t_k) & \\ & & \ddots & \ddots & \ddots \end{bmatrix}.$$

Non-tri-diagonal elements are induced by loop closures.

For a thorough description of ESDF filter procedures such as prediction with and without state vector augmentation, observation, measurement update and state vector recovery, see Callmer and Granström [Callmer and Granström, 2008].

Figure 1: Vehicle used in the experiments.

## 2.3 Vehicle Model

The utility vehicle, Fig. 1, used for data acquisition is fitted with a wheel encoder that measures vehicle speed and an inertial measurement unit (IMU) that measures turn rates. A turn rate vehicle model was used in the experiments. It has two input signals $\mathbf{u}(t_k) = [u_1(t_k)\ u_2(t_k)]^T$, corresponding to speed [m/s] and turn-rate [rad/s], respectively. The coordinate system for this model is centered at the laser, which is attached to the front of the car. The model equations are

$$\begin{aligned}
\mathbf{x}_v(t_{k+1}) &= \begin{bmatrix} x_v(t_{k+1}) \\ y_v(t_{k+1}) \\ \phi_v(t_{k+1}) \end{bmatrix} = \mathbf{f}(\mathbf{x}_v(t_k), \mathbf{u}(t_{k+1})) \\
&= \begin{bmatrix} x_v(t_k) \\ y_v(t_k) \\ \phi_v(t_k) \end{bmatrix} + T_s B \begin{bmatrix} u_1(t_{k+1}) \\ u_2(t_{k+1}) \end{bmatrix}, \\
B &= \begin{bmatrix} \cos(\phi_v(t_k)) & -h\cos(\phi_v(t_k)) - T_1 \\ \sin(\phi_v(t_k)) & -h\sin(\phi_v(t_k)) + T_2 \\ 0 & 1 \end{bmatrix}, \\
T_1 &= a\sin(\phi_v(t_k)) + b\cos(\phi_v(t_k)), \\
T_2 &= a\cos(\phi_v(t_k)) - b\sin(\phi_v(t_k)),
\end{aligned}$$

where $a$, $b$ and $h$ are vehicle specific parameters explaining the position of the laser sensor, and $T_s$ is the sampling time.

# 3 Tree of Words

To classify an image using ToW, feature descriptors are first extracted from the image using the feature extractor SURF [Bay *et al.*, 2006]. Each descriptor is compared to a large number of predefined descriptor vectors, called words, using a hierarchical tree search to find its nearest match. If descriptor $a$ is classified as word $m$, the image is said to contain word $m$, no matter exactly how well $a$ and $m$ matches. The image is thus compressed into a list of the words it contains. This list can be readily stored and compared to a database of classified images.

In a SLAM solution, this enables a fast way to compare a new scene to a large number of other scenes saved in the experiment, in order to detect loop closures.

## 3.1 Building a Vocabulary Tree

The vocabulary is built using a large number of feature descriptors, extracted from an image database independent of the future SLAM experiments. These descriptors are clustered into a tree structure using k-means clustering [Hartigan and Wong, 1979]. Each leaf in the tree represent a word.

The rootnode uses all descriptor vectors to compute $k$ clusters. Each cluster, described by its cluster centre vector, is represented in the tree as a node. These clusters in turn forms $k$ new subclusters, using the descriptors assigned to the cluster. Subsequently a tree of arbitrary depth $n$ can be formed.

In our SLAM experiments a tree with $n = 5$ and $k = 10$ is used. This results in a vocabulary of 100000 words.

## 3.2 Descriptor Classification

A descriptor vector is classified using the vocabulary tree. For each level of the tree, the descriptor is compared to the $k$ child nodes' cluster vectors using angular comparison. The descriptor vector is eventually classified as its most similar leaf. Intuitively, ignoring a nearest neighbours search such as Depth First [Friedman *et al.*, 1977] or Best Bin First [Beis and Lowe, 1997] should make classification more uncertain for a larger tree since descriptor noice would cause missclassifications, but in [Nistér and Stewénius, 2006] it was shown that a larger tree only improves classification.

After all descriptors of an image have been classified as words, the image is described as a list of these words, called a word vector.

## 3.3 Add Image to the Database

A new image is added to the database by simply converting it into a word vector which is saved.

## 3.4 Compare Image to the Database

An image is compared to a database of classified images by converting it into a word vector. This word vector is compared to the stored vectors using a weighted comparison called scoring. In our SLAM experiments, this comparison is limited to a smaller set of likely candidates, see Section 3.6.

Each word in the vocabulary tree has been assigned a significance known as its weight $w_i$. Weighing using Inverse Document Frequency (IDF) [Jones, 1972] is defined as

$$w_i = \ln \frac{N}{N_i}, \tag{4}$$

where $N$ is the number of images in the database and $N_i$ is the number of images in which the word has occured. A common word is given a low weight and an unusual word a high weight. The elements of the query and database vectors $q$ and $d$ are

$$q_i = n_i w_i, \tag{5a}$$

$$d_i = m_i w_i, \tag{5b}$$

where $n_i$ and $m_i$ are the number of times the word $i$ has occurred in the query image and database image, respectively.

The scoring of a database image is defined as

$$s(q,d) = \left\| \frac{q}{\|q\|} - \frac{d}{\|d\|} \right\| = \|\mathbf{q} - \mathbf{d}\| \tag{6}$$

under some norm. According to Nistér and Stewénius [Nistér and Stewénius, 2006], scoring using the $L_1$ norm consequently gives a slightly better result than the $L_2$ norm, but the latter was chosen for loop closure detection because it is computationaly cheaper.

For the $L_2$ norm, (6) can be rewritten as a scalar product according to

$$\|\mathbf{q} - \mathbf{d}\|_2^2 = 2 - 2\ (\mathbf{q}^T\mathbf{d}). \tag{7}$$

For simplicity, $L_2$ scoring is implemented as

$$s_{L_2}(q,d) = \mathbf{q}^T\mathbf{d}. \tag{8}$$

In this case $s_{L_2}(q,d) \in [0,1]$ with 1 being the score for identical match.

## 3.5 Tree of Words with Spatial Consistency

For SLAM experiments, Tree of Words has too many false positives when used for loop closure detection. The problem is caused by the compression of an image into just a list of words. Two completely dissimilar scenes can still be similar in this representation since there is no information describing how the features relate to each other. We suggest a way to save a simple representation of the image's spatial relations for safer matching.

The feature extractor provides not only the descriptors, but also the center of the regions in the image where the features were located. In the image, the region where descriptor $b$ was extracted, is the region nearest to the region where descriptor $a$ was extracted. This makes descriptor $b$ the nearest neighbour of $a$. Note that $a$ is not necessarily the nearest neighbour of $b$. $a$ and $b$ are thereafter classified as words $m$ and $n$, and $n$ is saved as $m$'s nearest neighbour. Should word $m$ appear more than once, the distances to the nearest neighbours are compared and the closest neighbour is chosen. Every image is now compressed into two vectors; $\mathbf{d}$ containing the list of words and $\mathbf{d_{nn}}$ containing the nearest neighbour of each one of these words.

Scoring finds the possible loop closure matches $\mathbf{d^m}$ and spatial relations are used to confirm them. $\mathbf{d^m_{nn}}$ is compared to $\mathbf{q_{nn}}$ to get a list of the common words. For these words, the nearest neighbour of each word is compared and the ratio of correct neighbours is computed according to (9).

$$\frac{N_{\text{same nn}}}{N_{\text{common}}} = \epsilon, \tag{9}$$

where $N_{\text{same nn}}$ is the number of common words with the same nearest neighbour and $N_{\text{common}}$ is the number of common words. If $\epsilon$ is above a certain threshold, the image is considered to be a match. This spatial consistency (SC) representation is only a backup system for the scoring, but has shown to be efficient.

## 3.6 Tree of Words in SLAM

Tree of Words always finds a best match, even if there is no image in the database that is actually similar to the query image. In most cases, like an image search on the Internet, getting a dissimilar best match is no disaster. In a SLAM experiment, an incorrect match used as a loop closure can destroy the entire estimation. To avoid this, only a match with a score above a certain threshold and a spatial consistency ratio above another threshold will be considered a loop closure.

In our SLAM experiment using an ESDF, every image acquired is processed according to Algorithm 1.

**Algorithm 1** Image SLAM: Closing the Loop

```
 1: Classify new image as a list of words
 2: Find set of Loop Closure candidates.
 3: if Loop Closure candidates found then
 4:   Compare image to loop closure candidates.
 5:   if Image match/es found with score above threshold then
 6:     Check match/es for spatial consistency.
 7:     if Image match/es SC ratio is above threshold then
 8:       Calculate relative pose
 9:       Update filter
10:     else
11:       Ignore image match.
12:     end if
13:   end if
14: end if
15: Add image to database
```

Before ToW can be used in SLAM experiments, a tree must be prepared. The necessary size of the tree depends on the environment in which future experiments will take place. For an urban experiment, a tree with 100000 words produces acceptable results. In an environment where all features look more or less the same, like an underwater environment, a much larger tree is needed. Offline experiments on images from the chosen environment, can be used for performace evaluation. The size of the tree will, besides performance, also effect execution time and memory requirements. The weight of each word can now also be calculated. Thereafter, threshold values for score and spatial consistency ratio are determined empirically.

**Loop Closure Candidate Selection**
Loop closure candidates are obtained utilising the known estimated covariance of the current vehicle pose, computed using the Mahalanobis distance. The 50 last poses are excluded since they are too close to be possible loop closures, indicated by the ellipse in Fig. 2.

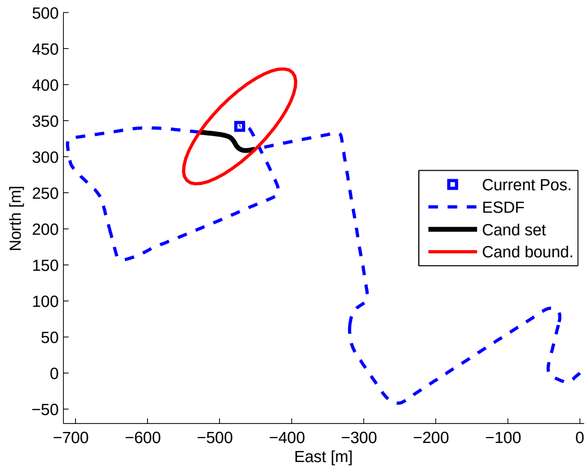

Figure 2: Loop closure candidates, indicated by the ellipse. The 50 last augmented states are excluded.

Loop closure candidates are not primarily selected to reduce computational time. Execution time for image comparison under the $L_2$ norm is almost independent of the number of loop closure candidates. Comparing an image to a database of 5000 images takes roughly 20% longer than comparing it to a database of 100 images – 0.30 sec compared to 0.25 sec for some example images. Instead, candidates are selected to make the loop closure comparison consistent with the filter estimation. It is unnecessary to compare the scene of the current pose with all other images acquired during the experiment, since the filter estimate can provide a set of likely candidates. Ignoring the estimated pose and covariance, results in an experiment where the filter is considered reliable for map building but not for loop closure candidate selection.

## 4 Experimental Results

### 4.1 Loop Closure Performance

Loop closing capability of Tree of Words with different thresholds for scoring and spatial consistency ratio, was examined in ground experiments. An urban data set almost 2 kilometers in length with a total of about 155 loop closures was used. A new image was acquired every 1.5 m or 15° turn. The experiments were performed in conjunction with an ESDF, estimating the vehicle trajectory and its uncertainty. Loop closure candidates were chosen as in Sec. 3.6.

A total of 43000 loop closure candidates were selected throughout the experiment. Of these, about 42000 were false candidates.

Using the detected loop closures in filter updates, changes the filter's estimated trajectory and its uncertainty. This in turn changes the loop closure candidates selected for comparison. To compare the different threshold settings fairly, the loop closure detections were not used to update the filter. The loop closure candidates were therefore constantly chosen by using only the odometry induced trajectory and its estimated uncertainty.

The loop closure performance was evaluated by the charateristics true positive rate and false positive rate. There are different ways to define the true positive rate and false positive rate. Here, true positive rate is defined as the number of detected loop closures divided by the total number of loop closures. False positive rate is defined as the number of incorrect loop closures divided by the total number of non matching data pairs.

A receiver operating characteristic curve (ROC) displays the relationship between the two characteristics of a classifier, true positive (detection) rate versus false positive (false alarm) rate, as the thresholds for detection are changed. For any probability of false positive, as high as possible probability of true positive is wanted.

**Loop closure performance without SC**
The score threshold is varied from 0.2 to 0.6. The performance is presented as a ROC curve in Fig. 3. It shows that false positives can be eliminated at the price of fewer detected loop closures, by raising the threshold. A detection rate of about 70% is achieved with a false alarm rate of 0.08%.

**Loop closure performance with SC**
The introduction of spatial consistency is intended to reduce the false positive ratio while keeping the detected loop closure ratio high. An SC ratio threshold is selected and the score threshold is varied to see how SC affects the ROC curve.

Four SC ratio thresholds are selected; 0, 0.1, 0.2 and 0.25. For these, the score threshold is varied from 0.2 to 0.3. The SC ratio threshold 0 is equal to using no SC and is included for comparison clarity.

The results can be studied in Fig. 4. The solid line is scoring without using SC ratio. The pale solid line has a fixed SC ratio threshold of 0.1 and a varying score

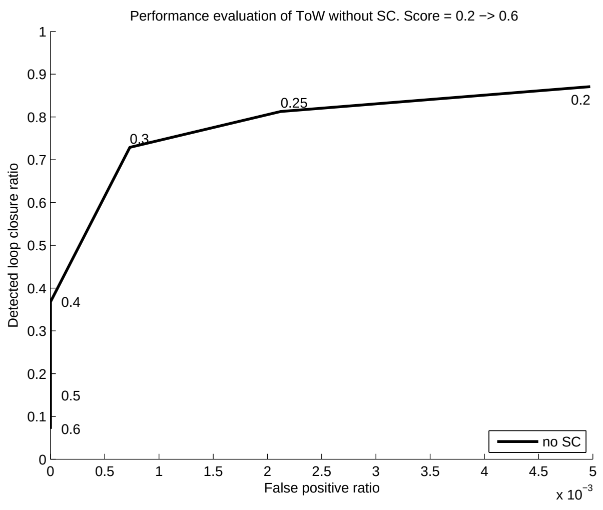

Figure 3: Loop closure performance of ToW without SC.

threshold. The dash-dotted line has a fixed SC ratio threshold of 0.20 . Finally, the dashed line has a fixed SC ratio threshold of 0.25 with a varying SC ratio.

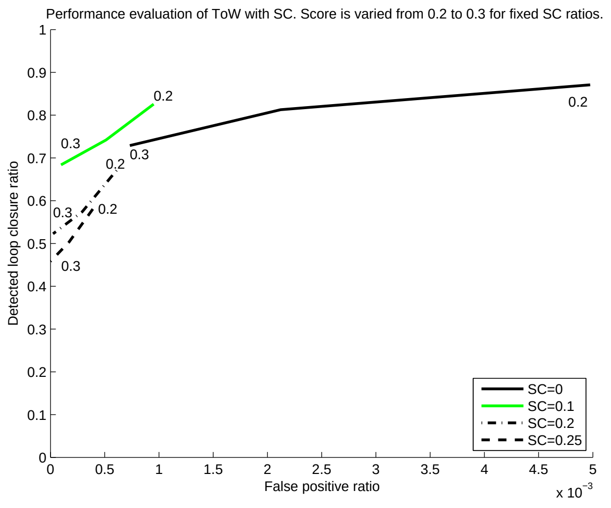

Figure 4: Loop closure performance of ToW with fixed SC ratio and varying score.

It is clear that a combination of thresholds for score and spatial consistency enables a lower false positive ratio with a higher detected loop closure ratio, than using just a threshold for score. For example, a combination of a SC ratio threshold of 0.1 and a score threshold of 0.3 results in a false positive ratio of 0.01% while still detecting almost 70% of the loop closures. Without the SC ratio threshold, the false positive ratio is almost 10 times higher.

## 4.2 SLAM ground experiments

Tree of Words is used in SLAM ground experiments to detect loop closures using images from a forward facing camera. The filter used in the experiment is an ESDF and the relative pose is estimated by aligning laser scans of the surroundings using CRF followed by ICP. The usage of a forward facing camera requires the vehicle to revisit an area from the same directions in order to detect a loop closure. Multiple cameras facing different directions could be a solution to that problem.

The urban dataset constitutes of a short loop, ending with a few loop closures. A score threshold of 0.2 and a SC ratio of 0.2 is used to keep the number of false positives low.

A total of 613 poses are augmented into the state vector and three loop closures are detected. The experiment takes 14 min, or 1.3 sec per pose. About half the execution time is spent aligning laser scans after the detected loop closures.

The estimated trajectory compared to GPS ground truth and dead reckoning can be studied in Fig. 5. The estimated trajectory is good since the loop closure is connecting the end of the data set with the beginning.

The forward laser scans are plotted on each estimated pose, resulting in a laser map. This laser map is overlaid on an aerial photograph of the experiment area, to illustrate the correspondance between the estimated map and the real environment. This can be studied in Fig. 6. It shows how well the filter has estimated the vehicle trajectory compared to the true environment. Three loop closures were discovered and the quality of the alignments makes the end result quite good.

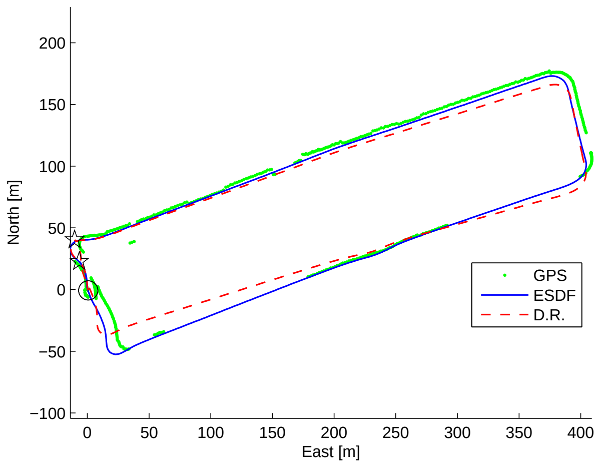

Figure 5: Comparison of dead reckoning and estimated trajectory. The ring marks the starting point, stars mark end points.

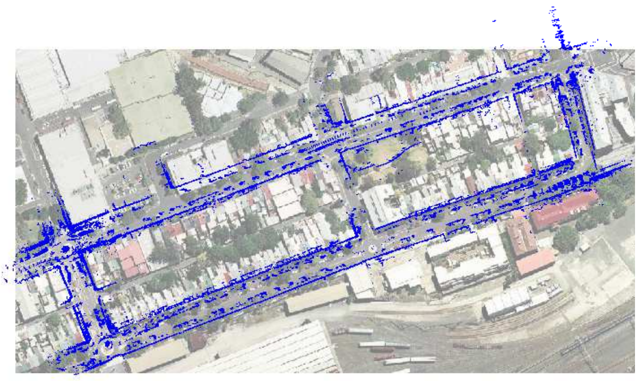

Figure 6: Laser map overlaid on aerial photograph.

## 5 Conclusions

Tree of Words with Spatial Consistency has been shown to be a simple and fast, yet reliable way to detect loop closures in an urban environment. The introduction of Spatial Consistency has shown promising results, enabling a significantly lower false positive ratio for a given true positive rate. This is achieved with a negligible increase in computational complexity. The performance has been evaluated and the results have been used in a SLAM experiment in a complex urban area. The loop closures detected resulted in a good estimation of the trajectory.

## References

[Angeli *et al.*, 2008] A. Angeli, D. Filliat, S. Doncieux, and J.-A. Meyer. Real-time visual loop-closure detection. In *Proc. IEEE International Conference on Robotics and Automation (ICRA'08)*, Pasadena, California, April 2008.

[Bailey and Durrant-Whyte, 2006] T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (SLAM): part II. *Robotics & Automation Magazine, IEEE*, 13(3):108 – 117, 2006.

[Bailey, 2001] T. Bailey. *Mobile Robot Localisation and Mapping in Extensive Outdoor Environments.* PhD thesis, University of Sydney, 2001.

[Bay *et al.*, 2006] H. Bay, T. Tuytelaars, and L Van Gool. Surf: Speeded up robust features. In *9th Europ. Conf. on Computer Vision*, Graz Austria, May 2006.

[Beis and Lowe, 1997] J. Beis and D Lowe. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *Conf. on Computer Vision and Pattern Recognition*, Puerto Rico, 1997.

[Bosse and Roberts, 2007] M. Bosse and J. Roberts. Histogram matching and global initialization for laser-only SLAM in large unstructured environments. *2007 IEEE International Conference on Robotics and Automation*, pages 4820–4826, 2007.

[Callmer and Granström, 2008] J. Callmer and K. Granström. Large Scale SLAM in an Urban Environment. Master's thesis, Linköping University, May 2008.

[Chong and Kleeman, 1999] K.S. Chong and L. Kleeman. Feature-based mapping in real, large scale environments using ultrasonic array. *Int. J. Robot. Res.*, 18(1):3–19, 1999.

[Cole and Newman, 2006] D. Cole and P. Newman. Using laser range data for 3d slam in outdoor environments. In *Proc. IEEE International Conference on Robotics and Automation, (ICRA'06)*, Florida, 2006.

[Cummins and Newman, 2008] M. Cummins and P. Newman. Accelerated appearance-only SLAM. In *Proc. IEEE International Conference on Robotics and Automation (ICRA'08)*, Pasadena, California, April 2008.

[Davison *et al.*, 2004] A. Davison, Y.G. Cid, and N. Kita. Real-time 3D slam with wide-angle vision. In *Proc. IFAC/EURON Symp. Intelligent Autonomous Vehicles*, 2004.

[Durrant-Whyte and Bailey, 2006] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping (SLAM): part I. *Robotics & Automation Magazine, IEEE*, 13(2):99 – 110, 2006.

[Estrada *et al.*, 2005] C. Estrada, J. Neira, and J.D. Tardos. Hierarchical slam: Real-time accurate mapping of large environments. *Robotics, IEEE Transactions on [see also Robotics and Automation, IEEE Transactions on]*, 21(4):588–596, August 2005.

[Eustice *et al.*, 2006] R. M. Eustice, H. Singh, and J. J. Leonard. Exactly sparse delayed-state filters for view-based SLAM. *IEEE Transactions on Robotics*, 22(6):1100–1114, December 2006.

[Friedman *et al.*, 1977] J. Friedman, J. Bentley, and R. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Trans. Math. Software*, 3:209–226, 1977.

[Guivant and Nebot, 2002] J. Guivant and E. Nebot. Improving computational and memory requirements of simultaneous localization and map building algorithms. *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, 3:2731–2736, 2002.

[Gutmann and Konolige, 1999] J.-S. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. *Computational Intelligence in Robotics and Automation, 1999. CIRA '99. Proceedings. 1999 IEEE International Symposium on*, pages 318–325, 1999.

[Hartigan and Wong, 1979] J. A. Hartigan and M. A. Wong. A K-Means Clustering Algorithm. *Applied Statistics*, 28(1):100–108, 1979.

[Ho and Newman, 2005] K. Ho and P. Newman. Combining visual and spatial appearance for loop closure detection. *Proceedings of the European Conference on Mobile Robotics*, September 2005.

[Jones, 1972] K. S. Jones. Exhaustivity and specificity. *Journal of Documentation*, 28(1):11 – 21, 1972.

[Montemerlo *et al.*, 2002] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proc. AIII Nat. Conf. on Artif. Intel.*, Edmonton, Canada, 2002.

[Newman *et al.*, 2006] P. Newman, D. Cole, and K. Ho. Outdoor slam using visual appearance and laser ranging. In *Proc. IEEE International Conference on Robotics and Automation, (ICRA'06)*, Florida, 2006.

[Nieto *et al.*, 2005] J. Nieto, J. Guivant, and E. Nebot. Scan-slam: Combining EKF-SLAM and scan correlation. In *Proc. IEEE Int. Conf. Field Service Robotics*, 2005.

[Nieto *et al.*, 2007] J. Nieto, T. Bailey, and E. Nebot. Recursive scan-matching SLAM. *Robotica and Autonomous Systems*, 55(1):39–49, 2007.

[Nistér and Stewénius, 2006] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 2161–2168, June 2006.

[Ramos *et al.*, 2007a] F. Ramos, D. Fox, and H. Durrant-Whyte. CRF-matching: Conditional Random Fields for feature-based scan matching. In *Proceedings of Robotics: Science and Systems*, Atlanta, Georgia, United States of America, June 2007.

[Ramos *et al.*, 2007b] F. Ramos, J. Nieto, and H.F. Durrant-Whyte. Recognising and modelling landmarks to close loops in outdoor slam. In *Proc. IEEE International Conference on Robotics and Automation (ICRA'07)*, Rome, 2007.

[Sivic and Zisserman, 2003] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proceedings of the Int. Conf. on Computer Vision*, volume 2, pages 1470–1477, October 2003.

[Thrun *et al.*, 2005] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics.* MIT Press, Cambridge, MA, 2005.

[Walter *et al.*, 2007] M.R. Walter, R. Eustice, and J. Leonard. Exactly sparse extended information filters for feature-based slam. *International Journal of Robotics Research*, 26(4):335–359, 2007.