International Journal of Computer Vision 74(3), 261–286, 2007

DOI: 10.1007/s11263-006-0020-1

Detecting Loop Closure with Scene Sequences

KIN LEONG HO AND PAUL NEWMAN
Oxford Robotics Research Group, University of Oxford, Parks Road, OX1 3PJ, United Kingdom
klh@robots.ox.ac.uk
pnewman@robots.ox.ac.uk

Received October 27, 2005; Accepted December 6, 2006

First online version published in January, 2007

Abstract. This paper is concerned with "loop closing" for mobile robots. Loop closing is the problem of correctly asserting that a robot has returned to a previously visited area. It is a particularly hard but important component of the Simultaneous Localization and Mapping (SLAM) problem. Here a mobile robot explores an *a-priori* unknown environment performing on-the-fly mapping while the map is used to localize the vehicle. Many SLAM implementations look to internal map and vehicle estimates (p.d.fs) to make decisions about whether a vehicle is revisiting a previously mapped area or is exploring a new region of workspace. We suggest that one of the reasons loop closing is hard in SLAM is precisely because these internal estimates can, despite best efforts, be in gross error. The "loop closer" we propose, analyze and demonstrate makes no recourse to the metric estimates of the SLAM system it supports and aids—it is entirely independent. At regular intervals the vehicle captures the appearance of the local scene (with camera and laser). We encode the similarity between all possible pairings of scenes in a "similarity matrix". We then pose the loop closing problem as the task of extracting statistically significant *sequences* of similar scenes from this matrix. We show how suitable analysis (introspection) and decomposition (remediation) of the similarity matrix allows for the reliable detection of loops despite the presence of repetitive and visually ambiguous scenes. We demonstrate the technique supporting a SLAM system driven by scan-matching laser data in a variety of settings. Some of the outdoor settings are beyond the capability of the SLAM system itself in which case GPS was used to provide a ground truth. We further show how the techniques can equally be applied to detect loop closure using spatial images taken with a scanning laser. We conclude with an extension of the loop closing technique to a multi-robot mapping problem in which the outputs of several, uncoordinated and SLAM-enabled robots are fused without requiring inter-vehicle observations or *a-priori* frame alignment.

Keywords: loop closing, SLAM, mobile robotics, scene appearance and navigation, multi-robot navigation

1. Introduction and Background

This paper is concerned with detecting "Loop Closure"—correctly asserting that a mobile robot has returned to a previously visited area. This is a particularly important competency in relation to the Simultaneous Localization and Mapping (SLAM) problem.

To motivate the work discussed in this paper we begin by briefly reviewing the SLAM problem. SLAM is concerned with having an autonomous vehicle continuously localize in an *a-priori* unknown environment without recourse to infrastructure and using only onboard sensors. The idea is that a suitable representation (map) of the environment can be constructed on-line that allows fast metric localization of the vehicle. The overwhelming majority of successful SLAM algorithms couch the SLAM problem in a probabilistic framework. By various means, they try to estimate the joint probability of map and vehicle trajectory conditioned on control inputs, sensor measurements and data association. The latter is the process of deciding how to associate sensor measurements with new or existing elements of the map. The loop closing problem, with which this paper is concerned, is essentially a data association problem. A positive loop closure occurs when the robot *recognizes* the local scene to be one it has previously visited. We wish the robot to be able to do this irrespective of internal map-vehicle estimates. After all, loop closing is hard (and needed) precisely because

these estimates are often in error. Although we do not preclude the use of SLAM estimates in loop closure detection, the advantages of having an independent loop closure detection system appear clear.

It is important to point out the differences between the problem of localization and loop closing. In localization, it is known *a priori* that the robot is operating in a previously visited environment. The appearance of the local scene must match one in a "database" of previously visited scenes. Additionally, the goal of a localization system is to locate the robot accurately at every possible pose. In contrast, it is not crucial that loop closure detection occurs at every opportunity. If and when a robot returns to a previously mapped region, a single, reliable detection of loop closing is all that is required for the map to be corrected.

In many contemporary feature-based SLAM algorithms, simple geometric primitives such as corners or lines are used as features. These features, by themselves, are not particularly discriminative. They are distinguishable only by their global or relative locations. Many data association techniques (Bar-Shalom, 1987; Neira and Tardõs, 2001) used for loop closing are therefore dependent on accurate map-vehicle estimates. These techniques can close small loops, using Mahalonobis distances, reliably but are less successful when faced with the large errors that inevitably accumulate while traversing loops (see Fig. 2).

Problems remain when removing the need to create maps as a collection of predefined geometric primitives. A popular approach is "scan matching" in which laser range scans are rigidly attached to vehicle poses. Sequential scans (and hence vehicle poses) are registered to each other resulting in a trajectory of vehicle poses and a map made implicitly from the now aligned raw laser scans. In Gutmann and Konolige (1999), an approach called "Local Registration and Global Correlation" was introduced to determine topologically correct relations between new and old poses after long cycles. To identify loop closure, a large "scan patch" is correlated (over motion in the plane) with a partition of the global map. The intuition is that the larger scan patch will be more reliable than a single laser scan in rejecting false positives. However, the location of the search space is still dependent on the robot pose estimate.

Vision has also been used successfully for localization of a mobile robot (Wang et al., 2005; Wolf et al., 2005). Location neighborhood relationships captured by a Hidden Markov Model were exploited in Torralba et al. (2003) and Kosecka et al. (2005) to localize a robot into one of a small number of pre-defined locations. Considering the temporal relationship between images provides robustness against dynamic changes and inherent appearance ambiguities. However the supervised labelling of images following an initial exploration stage does not lend itself to the SLAM problem where there is no controlled mode change from exploration to localization. Vision-based global localization and mapping was achieved in Se et al. (2005) by matching SIFT features detected in a local submap to a pre-built SIFT database map. 3D submaps are built by aligning multiple frames while the global map is built by aligning and merging multiple 3D submaps. Visual loop closure detection was attempted in Levin and Szeliski (2004) through correspondences between omnidirectional images and in the context of this work, the similarity between all images are expressed in a distance matrix. The quality of the distance matrix is improved by imposing epipolar constraints. It was observed in both (Levin and Szeliski, 2004; Silpa-Anan and Hartley, 2005) that off-diagonals can be found in the matrix when the same path is travelled again and reverse-diagonals in the matrix when the reverse path is taken. However, this observation was not exploited further in either paper. The visual appearance aspects of the work in Ranganathan et al. (2006) has a common ground with this paper—visual appearance is used to help close loops. However the focus is on topological reasoning rather than on to build a reliable loop closure detector.

Vision has also been used in SLAM for the extraction of geometric features as natural landmarks from the environment (Davison and Murray, 2002; Se et al., 2002). Given feature correspondences from images captured from multiple calibrated cameras, 3D positions of image features relative to the robot can be computed and used as a measurement stream. In Davison (2003) an Extended Kalman Filter is used to achieve realtime 3D-SLAM over a modest (desk size) area with a single camera given an initialization of scale. The relatively small map size means that, in this system, loop closing is not an issue as the vehicle (a hand held camera) does not execute large looping trajectories and pose errors do not accumulate.

The paper will proceed as follows. We begin in Section 2 by briefly describing the platform and sensors that were used to generate all the data sets used in this paper. This data will be used to illustrate aspects of the loop-closing problem throughout the paper—hence the introduction of experimental apparatus in an early section. For a similar reason, we go on to describe in Section 2 a laser-based SLAM system that, where applicable, is used to generate maps—maps that have substantial loops that need detecting. In Section 3 we describe how scene appearance is encoded and how any two scenes can be compared. In Section 4 we focus on how sequences of matches can be used to affirm or discredit loop closure hypotheses. Section 5 introduces a method in which spectral decomposition is invoked to remove the effects of common-mode similarity and ambiguity across captured scenes—something which impedes successful execution of the sequence extraction described in Section 4. The section concludes by discussing how to adjudicate the statistical significance of loop closure

detections. Section 6 offers an extensive set of results demonstrating and analyzing the performance of our approach in a variety of indoor and outdoor settings and also its application to multi-robot navigation. Finally the paper concludes with Section 8 which summarizes the contributions of this work.

2. Infrastructure

2.1. Hardware and Instrumentation

The research vehicle used is a small all-terrain vehicle, equipped with wheel encoders, a calibrated camera, a 2D laser scanner and a 3D laser scanner. Figure 1 displays an image of the platform. At the top is a pan-tilt EVI-D30 camera used to capture all the images discussed from here on. Below the camera is a SICK LMS-200 laser scanner mounted on a mechanical oscillator allowing the 3D geometry of the local workspace to be captured. Behind the front bumper is an additional laser which is used for obstacle avoidance and, when suitable, SLAM.

2.2. A SLAM System

This section provides a brief summary of a SLAM system which we will augment with loop closure detection. It is provided here to provide a technical context for the loop closure problem. If the reader is familiar with the field (s) he may move straight to Section 2.3. The SLAM system we use is the delayed state formulation used in Bosse et al. (2004) and Leonard and Newman (2003) and has much in common with work in Gutmann and Konolige (1999) and Lu and Milios (1997) and more recently (Eustice et al., 2005). There is nothing special about the choice of SLAM filter employed here and any technique that produces a metric vehicle position estimate could have been used.

Figure 1. The research platform. At the top is a pan-tilt camera. In the middle is a laser scanner mounted on a mechanical oscillator that allows a 2D laser scanner to capture 3D scans of the environment. At the bottom is a fixed laser scanner which is used to obtain 2D scans parallel to the ground plane.

2.2.1. Formulation. The estimated quantity is a state vector $\mathbf{x}(i \mid j)$ which initially contains a single vehicle pose $\mathbf{x}_v(0 \mid 0)$. In the 2D case, $\mathbf{x}_v$ is a three element vector of $[x, y, \theta]^T$ whereas $\mathbf{x}_v$ is a six element vector of $[x, y, z, \theta, \phi, \psi]^T$ for the 3D case. Associated with it is a covariance matrix $\mathbf{P}(0 \mid 0)$. Here we are adopting the usual notation that the quantity $\mathbf{x}(i \mid j)$ is the estimate of the true state $\mathbf{x}$ at time i given measurements up until time j.

At some time $k+1$ the vehicle is subject to a noisy control vector $\mathbf{u}(k+1)$ such that the new position of the vehicle can be written as a function of the control and the last state estimate.

$$\mathbf{x}_v(k+1 \mid k) = \mathbf{x}_v(k \mid k) \oplus \mathbf{u}(k+1) \tag{1}$$

where $\oplus$ is a 3D or 6D transformation composition operator. The second order statistics of $\mathbf{x}(k+1 \mid k)$ following a control input to be written as

$$\begin{aligned}\mathbf{P}_v(k+1 \mid k) = {} & \mathbf{J}_1(\mathbf{x}_v, \mathbf{u})\, \mathbf{P}(k \mid k)\, \mathbf{J}_1(\mathbf{x}_v, \mathbf{u})^T \\ & + \mathbf{J}_2(\mathbf{x}_v, \mathbf{u})\, \mathbf{U}\, \mathbf{J}_2(\mathbf{x}_v, \mathbf{u})^T\end{aligned}$$

where the $(k \mid k)$ and $(k+1)$ indices have been dropped from $\mathbf{x}_v$ and $\mathbf{u}$ respectively for clarity and $\mathbf{U}$ is the covariance of the noise process in control $\mathbf{u}$. $\mathbf{J}_1(a, b)$ and $\mathbf{J}_2(a, b)$ are the jacobians of $a \oplus b$ w.r.t a and b respectively.

We employ a delayed state model in which at every time step the state vector is augmented as follows:

$$\mathbf{x}(k+1 \mid k) = \begin{bmatrix} \mathbf{x}(k \mid k) \\ \mathbf{x}_{vn}(k \mid k) \oplus \mathbf{u}(k+1) \end{bmatrix} \tag{2}$$

$$= \begin{bmatrix} \mathbf{x}_{v1} \\ \vdots \\ \mathbf{x}_{vn} \\ \mathbf{x}_{vn+1} \end{bmatrix} (k+1 \mid k) \tag{3}$$

The state vector is simply a history of previous vehicle poses where we extend the notation to write the ith pose as $\mathbf{x}_{vi}$. No environment features are stored. Associated with each vehicle pose is a set of laser scans (2D or 3D)

and the latest image captured (the image will be used later to aid loop closing). The augmented covariance matrix, $\mathbf{P}(k+1 \mid k)$, following the transformation of Eq. (2), can be written as

$$\mathbf{P}(k+1 \mid k) = \begin{bmatrix} \mathbf{P}(k \mid k) & \mathbf{P}_{vp}(k+1 \mid k) \\ \mathbf{P}_{vp}(k+1 \mid k)^T & \mathbf{P}_v(k+1 \mid k) \end{bmatrix}. \quad (4)$$

It should be noted that k is not incremented at every iteration of the algorithm. The odometry readings of the vehicle are compounded until the overall change in pose is significant. This overall, compounded transformation becomes $u(k)$ and the k is incremented and the above state projection step described above is undertaken. In this way the state vector grows linearly with the driven path length and not with time.

2.2.2. Inter-Pose Measurements. The scan-matching part of the algorithm works as follows. Consider two poses at times i and j. Poses at times i and j have associated laser scans L_i and L_j, each containing n_i and n_j sets of x, y points (or x, y, z points in 3D) in the vehicle frame of reference. Assuming that there is a substantial overlap between the surfaces sampled in these two scans, it finds a transformation T parameterized by the vector $\mathbf{z}_{ij} = [x, y, \theta]^T$ or $\mathbf{z}_{ij} = [x, y, z, \theta, \phi, \psi]^T$ such that

$$\eta = \sum_{k=1:n_j} \Phi\big(L_i, T\big(L_j^k, \mathbf{z}_{ij}\big)\big) \quad (5)$$

is minimized. The function $\Phi(L_i, T(L_j^k, \mathbf{z}_{ij}))$ returns the unsigned distance between the kth point in scan j transformed by $\mathbf{z}_{ij}$, and all of scan i. Note that we need not assigning rigid point to point associations as is common in Iterated Closest Point (ICP) (Fitzgibbon, 2001) like algorithms. In our 2D implementation, Φ uses the distance transform of L_i and uses the coordinates of the transformed points of L_j to calculate the distance to the template scan L_i. The further details of the scan matching procedure are beyond the scope of this paper. However two important points must be made. Firstly, the scan-matcher needs to be seeded with an approximate initial estimate of $\mathbf{z}_{ij}$. Our current implementation has a convergence basin for typical indoor environments (labs, offices and corridors) of around $+/-$ 30 degrees and $+/-5$ meters and takes 40 ms to compute. The need for a ball-park initial estimate is not surprising as scan-matching is a non-linear optimization problem and as such is vulnerable to the presence of local minima. Secondly, as Lu and Milios (1997) described, scan matching can be used to provide constraints or "measurements" of the relationship between poses. In this case the output of the scan matcher is the transformation between pose i and pose j in the state vector. For example matching between scan $k+1$ and k allows the following measurement equation to be formed:

$$\mathbf{x}_v(k+1 \mid k) = \mathbf{x}_v(k \mid k) \oplus \mathbf{z}_{k,k+1} \quad (6)$$

There are several ways to use such an observation. It could simply be stored and used (in linearized form) as an observation in a sparse bundle adjustment as proposed in Konolige et al. (2004). Or, as we choose here, it can be used in a minimum mean squared error update step. Essentially we linearize the equation and use it as an observation in a non-linear Kalman filter which explains the observation as a function of just the last few entries in the state vector. Nevertheless it is important to note the update will alter the entire state vector (which is the vehicle's past trajectory) as shown in Fig. 2.

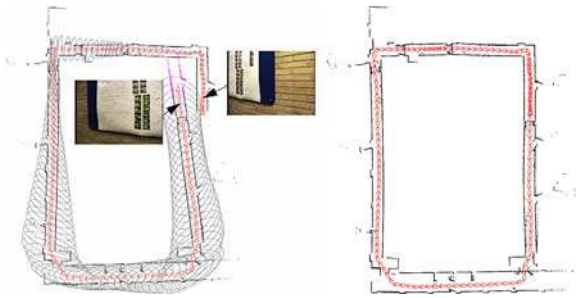

Figure 2. Pre and post loop closing. *Left*: a small angular error while passing through a swing door causes a gross error in position estimate which places the true location outside the 3-sigma bounds of the vehicle marginal. *Right*: After loop closure has been enforced a crisp map results. The issue at hand is how to detect that loop-closure should occur. Here the similarity between two views is used.

2.3. Limitation of Current Loop Closing Techniques

Figure 2 illustrates the limitation of some traditional loop closing techniques which rely on pose estimates. This is by no means a large loop or an extremely challenging environment for contemporary SLAM algorithms. However the accumulated spatial error is significant. Note how, at the end of the loop, the true location of the vehicle is well outside the 3-sigma marginal for the vehicle location. We maintain that regardless of which SLAM approach is used, however good the odometry is or what onboard inertial sensors are employed, a data set could be generated over some terrain or scale that results in accumulation of gross errors in both map and pose estimate p.d.f. Figure 2 also shows the final map after applying the loop closing constraint. Exactly how the loop closure is detected will be explained in following sections.

3. Scene Similarity

The loop-closure detection discussed in this work is achieved by scene sequence recall, principally using captured visual images but also later using laser images. For clarity we shall proceed by initially concerning ourselves with visual images only. In Section 6.5 we shall go on to use the same techniques to work with laser images. A fundamental entity in this work is the "scene database" often referred to as simply the "database". In the abstract, the database is considered to be a "black-box" with time stamped images as input and loop-closure detections as output. Internally the database is a data structure of images, patches and descriptors which admits search and retrieval operations. A key competency is the evaluation of the similarity between any two images I_u and I_v. Sections 3.1, 3.2 and 3.3 describe how this is achieved.

3.1. Interest Point Selection and Description

An image-based loop closure detection system was developed in Newman and Ho (2005) where the most recently captured image against all previous images stored in a database. This was the system used to detect the loop closure in Fig. 2. Each image is described by visually salient features, which are used for image similarity comparison. In contrast to Newman and Ho (2005), the only interest point detector that is adopted in this paper to extract features from images is the detector developed in Mikolajczyk and Schmid (2004), which finds "Harris-Affine Regions". Harris-Affine Regions offer significant invariance under affine transformations. Having found a set of image features, we encode them in a way that is both compact to allow swift comparisons with other features, and rich enough to allow these comparisons to be highly discriminatory. We use the SIFT descriptor (Lowe, 2004) which has become immensely popular in global visual localization applications (Se et al., 2002, 2005). In earlier work (Newman and Ho, 2005), we used Maximally Stable Extremal Regions (Matas et al, 2002) to select regions for SIFT encoding but found that Harris Affine regions lead to a greater diversity of SIFT features—an advantage for our approach. We describe the transformation of an image, I_u into a set of n descriptors as $\mathcal{D} : I_u \rightarrow \{d_1 \cdots d_n\}$, where n is itself a function of the input image. In the case of SIFT features, each d_i is a 128-dimensional vector.

3.2. Descriptor Relevance

As suggested in Sivic and Zisserman (2003), each image can be considered to be a document consisting of "visual words". In this case, each SIFT descriptor, d_i, is associated with a visual word $\hat{\mathbf{d}}_\mathbf{i}$, in a "visual vocabulary," $\mathcal{V} = \{\hat{\mathbf{d}}_\mathbf{i}, \hat{\mathbf{d}}_\mathbf{2} \ldots \hat{\mathbf{d}}_{|\mathcal{V}|}\}$. The vocabulary $\mathcal{V}$ is constructed by clustering similar descriptors (in terms of euclidean distance) into clusters. Each cluster of SIFT descriptors is considered to be associated with a visual word, represented by its cluster center.

A two-stage clustering procedure is used to build the visual vocabulary. The first is a seeding stage that employs a simple "leader follows" (Duda et al., 2001) strategy. The algorithm walks through the set of all descriptors creating a new cluster if no existing cluster center is within some threshold $d_{\min}$ or alternatively assigning it to the closest cluster center. In the latter case the cluster center is shifted to the mean of all assigned descriptors. The second stage builds a KD-tree from the cluster centers (words) and performs nearest neighbour vector quantisation of all descriptors onto them. This has the effect of compacting clusters which otherwise can become extended in descriptor space owing to the motion of the cluster center during the initial seeding stage.

Weights, w_i, are assigned to each word (cluster center), $\hat{\mathbf{d}}_\mathbf{i}$, according to its frequency in the entire image database. The motivation here is that not all words (descriptors) are equally good index terms—frequently occurring words do not make good indexes. The weighting, w_i, of each word is based on the inverse document frequency (Sparck Jones, 1972) formulation: $w_i = \log_{10}(N/n_i)$ where N is the total number of images stored in the image database and n_i is the number of images containing $\hat{\mathbf{d}}_\mathbf{i}$.

Note that we do not preordain the size of the vocabulary. Instead it is a consequence of the choice of the $d_{\min}$ parameter. We typically choose $d_{\min}$ so we end up working with vocabularies of around six thousand words ($d_{\min} \approx 300$). The vocabulary generation can be run as an off-line process operating on training images to produce a static lexicon of terms with which to describe future images. This is of course not the only way clustering could be undertaken; more sophisticated schemes exist and would surely produce slightly different vocabularies. Section 3.3 will describe how the vocabulary can be used to build a similarity function S between image pairs. The paper will proceed to use S for loop-closure detection in a way that is independent of its internals. Adopting a different, perhaps as yet unknown superior clustering algorithm would simply constitute a different implementation of a similarity function—one that could be adopted with ease.

3.3. Calculating Scene Similarity

The vector space model (Sivic and Zisserman, 2003) which has been successfully used in text-based image retrieval is employed in this work. To measure the similarity between two images, I_u and I_v, we examine their cosine distance. Each image, I_u, has become a collection

of words with different weights. If $\mathcal{V}$ contains $|\mathcal{V}|$ distinct words (clusters) we now create a vector $\overrightarrow{I}_u = [u_1 \dots u_{|\mathcal{V}|}]^T$ where

$$u_i = \begin{cases} w_i & \text{if for } \hat{\mathbf{d}}_\mathbf{i} \in I_u, \quad \min\limits_{k=1:|\mathcal{V}|} ||\hat{\mathbf{d}}_\mathbf{i} - \hat{\mathbf{d}}_\mathbf{k}|| < \epsilon \\ 0 & \text{otherwise} \end{cases}$$

for some distance threshold ϵ (typically 300 in all our vision based experiments). The normalized inner product of $\overrightarrow{I}_u$ and $\overrightarrow{I}_v$ can now be used to measure the similarity, $S(I_u, I_v) \in [0, 1]$, between images I_u and I_v:

$$S(I_u, I_v) = \frac{\sum_{i=1}^{|\mathcal{V}|} u_i v_i}{\sqrt{\sum_{i=1}^{|\mathcal{V}|} u_i^2} \sqrt{\sum_{i=1}^{|\mathcal{V}|} v_i^2}}. \tag{7}$$

The similarity function allows the creation of a "similarity matrix" which is a simple but central construct in this paper. A typical visual similarity matrix (VSM) is shown in Fig. 3.[1] Each element $\mathtt{M}_{i,j}$ is the similarity score, $S(I_i, I_j)$, between image i and image j from an image sequence $\mathcal{I} = [I_1, I_2 \dots]$. The diagonal elements are unity because all images are self-similar. Note how loop closures appear as a connected sequence of off-diagonal elements with high similarity scores. Note also that there are several, small, isolated off-diagonal dark (high similarity) patches. These are caused by similar scenes that do not constitute genuine loop closures. In the next section we will discuss how to detect loop closure by looking for *sequences* of similar images and hence reduce false positives. The motivation is that extended spatial regions should appear similar under genuine loop closure.

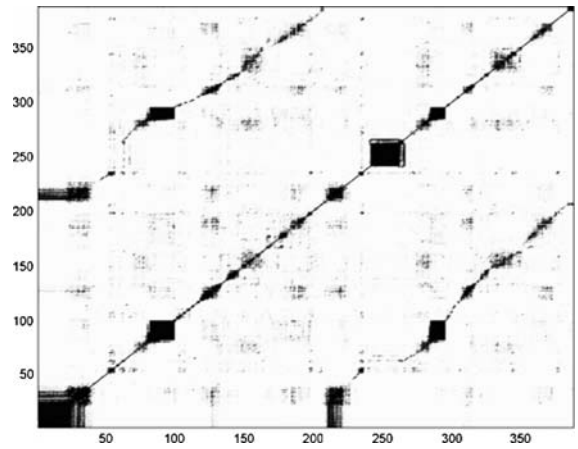

Figure 3. A typical visual similarity matrix with an obvious loop closure. Each element $\mathtt{M}_{i,j}$ is the similarity score, $S(I_i, I_j)$, between image i and image j from an image sequence $\mathcal{I} = [I_1, I_2 \dots]$. Cells with high similarity scores are colored in dark tone while cells with low similarity scores are colored in light tone. The diagonal elements are unity because all images are self-similar. Loop closures appear as a connected sequence of off-diagonal elements with high similarity scores. The dark "squares" are caused by repeated visually similar scenes in the environment.

As an implementation note, Eq. (7) need not explicitly create and then dot product $\overrightarrow{I}_u$ and $\overrightarrow{I}_v$ to calculate the elements of $\mathtt{M}$. Instead an inverted file index returns all images containing a particular word. By iterating over all words the similarity matrix can be populated by accumulating word by word the contributions to the cosine distance for each cell.

4. Detecting Sequences for Loop Closing

We wish to extract a sequence of matching images from a visual similarity matrix. As the vehicle moves through its work space it creates a sequence of images $\mathcal{I} = [I_1, I_2 \dots]$. We pose the loop closure detection problem as finding two subsequences of $\mathcal{I}$, $\mathcal{A} = [a_1, a_2 \dots]$ and $\mathcal{B} = [b_1, b_2 \dots]$ where a_i and b_i are index variables, whose overall similarity strongly suggests that the vehicle is revisiting a region.

Importantly, there is nothing to say that $a_i = b_j$ should imply $a_{i+1} = b_{j+1}$. It could be that a_{i+1} matches b_j as well, perhaps implying that two sequential images in $\mathcal{I}$ are identical because the vehicle has stopped or, more troubling, is imaging a scene with a repetitive structure. We use a modified form of the Smith-Waterman algorithm (Smith and Waterman, 1981), which is a dynamic programming algorithm, to find $\mathcal{A}$ and $\mathcal{B}$.

The algorithm proceeds by constructing a matrix H. Each element, $H_{i,j}$, is the maximal cumulative similarity score of a pairing of images ending with pairing I_i and I_j. Since the visual similarity matrix is symmetric, we will need only to work with the lower triangular matrix of M, excluding the main diagonal. For a practical implementation, a band of elements close to the main diagonal are masked out. This is equivalent to not looking for loop closure with images captured at locations that are less than a fixed distance away. Such small loop closures can be easily handled with existing SLAM techniques.

The element $H_{i,j}$ is a cumulative sum of the costs of sequence of moves through M. The moves are parallel to the direction on the principal diagonal. Three move types are possible: diagonal, horizontal and vertical. The latter two, although viable, are less preferable (one-to-many matching) and so have a penalty term δ (0.1 in this case) associated with them. Moving from $H(I_{i-1}, I_{j-1})$, $H(I_i, I_{j-1})$ and $H(I_{i-1}, I_j)$, $H_{i,j}$ becomes

$$H_{i,j} = \begin{cases} H_{i-1,j-1} + S(I_i, I_j) & \text{if } H(I_{i-1}, I_{j-1}) \text{ is maximal,} \\ H_{i,j-1} + S(I_i, I_j) - \delta & \text{if } H(I_i, I_{j-1}) \text{ is maximal,} \\ H_{i-1,j} + S(I_i, I_j) - \delta & \text{if } H(I_{i-1}, I_j) \text{ is maximal} \\ 0 & \text{if } S(I_i, I_j) < 0 \end{cases}$$

Table 1. Sequence extraction from a similarity matrix.

		I_1	I_2	I_3	I_4	I_5	I_6
M	I_1	1	−2	−2	0.64	−2	0.88
	I_2	−2	1	0.23	0.21	0.65	−2
	I_3	−2	0.23	1	0.37	0.25	0.71
	I_4	0.64	0.21	0.37	1	−2	−2
	I_5	−2	0.65	0.25	−2	1	0.22
	I_6	0.88	−2	0.71	−2	0.22	1
H	I_1	0	0	0	0	0	0
	I_2	0	0	0	0	0	0
	I_3	0	0.23	0	0	0	0
	I_4	0.64	0.75	1.02	0	0	0
	I_5	0	1.29	1.44	0	0	0
	I_6	0.88	0	2.0	0	0.22	0

The top matrix is an example of a simple visual similarity matrix where each cell i, j is the similarity score between the images i and j. Cells below a threshold (0.1) are re-scored to −2. Below is the corresponding H-matrix calculated from the lower triangular matrix of the visual similarity matrix shown above. A penalty (δ) of 0.1 is used for this example. The sequence selected is underlined.

where "is maximal" refers to the largest of $H(I_{i-1}, I_{j-1})$, $H(I_i, I_{j-1})$ and $H(I_{i-1}, I_j)$.

A requirement of the Smith-Waterman algorithm is that the similarity function must give a negative score when two elements are very dissimilar. In our implementation, image pairs with a similarity score that falls below a set threshold are deemed to be dissimilar and are re-scored with a fixed negative value (−1).

The maximum value in the H-matrix, the "maximal alignment score" $\eta_{\mathcal{A},\mathcal{B}}$, is the endpoint of a pair of image subsequences with the greatest similarity. From the H-matrix at the bottom of Table 1, the maximal alignment score is $H(I_6, I_3)$, which is an accumulation of similarity scores of the subsequence of underlined elements from $S(I_4, I_1)$ to $S(I_6, I_3)$.

To take into account that the robot might have traversed through the same area in opposite directions, the row order of the visual similarity matrix is reversed and the algorithm is repeated for that matrix order. The larger of the two alignment scores is chosen. To determine which images have contributed to the maximal alignment score, the algorithm back-traces through the contributing moves. The i-components produce $\mathcal{A}$ and the j-components produce $\mathcal{B}$.

We applied this algorithm to the uncluttered similarity matrix shown in Fig. 3(a). Figure 4(a) shows the lower triangular matrix. Figure 4(b) shows the significant match sequences extracted and Fig. 5 shows the actual images involved.

5. Ambiguity Management

The algorithm described in Section 4 works well in environments with few visually ambiguous or repetitive regions. The question is how the algorithm will perform in a more "confusing" environment such as the one resulting in the visual similarity matrix shown in Fig. 6(a). The astute observer will notice an off-diagonal dark line starting at around image 450. This is the start of the genuine loop closure. However, there are also numerous dark (mutually similar) off-diagonal regions. These are typically caused by repetitive imaging of architectural features like windows, long brick walls or broadly homogenous foliage. The concern now is that the visually ambiguous regions will prompt incorrect loop closures. False loop closures are a real disaster for SLAM systems, leading to catastrophic map damage and "lost" vehicles. Figure 7 shows an erroneous loop closure sequence extracted from the matrix shown in Fig. 6(a) before the ambiguity management steps this section describes are applied.

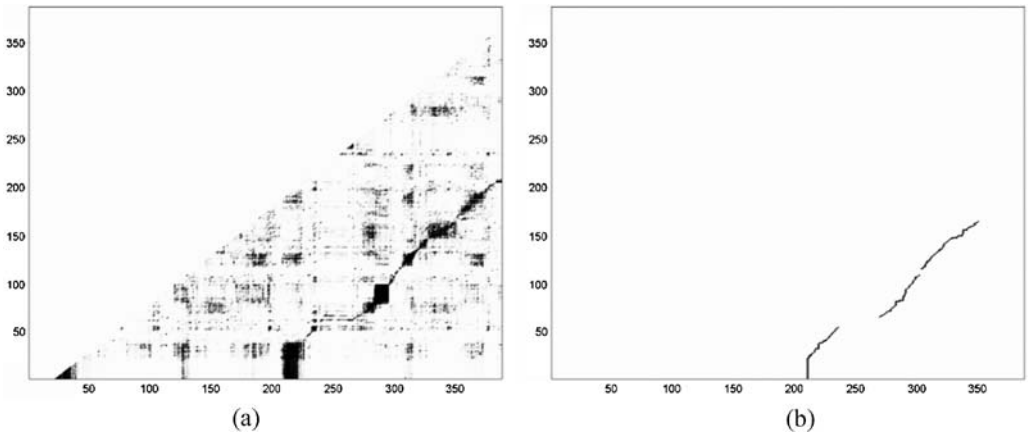

Figure 4. (a) A lower triangular matrix of a visual similarity matrix after loop closure has occurred. (b) The result of applying the modified Smith-Waterman algorithm to find significant local alignments.

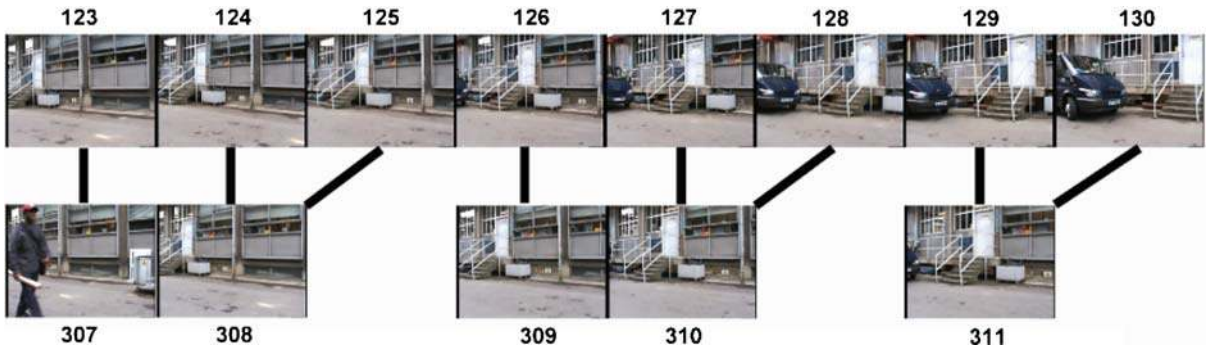

Figure 5. Two matching sequences $\mathcal{A}$ and $\mathcal{B}$ annotated with capture time. Note the occasional one-to-two pairings which correspond to vertical and horizontal moves through the similarity matrix.

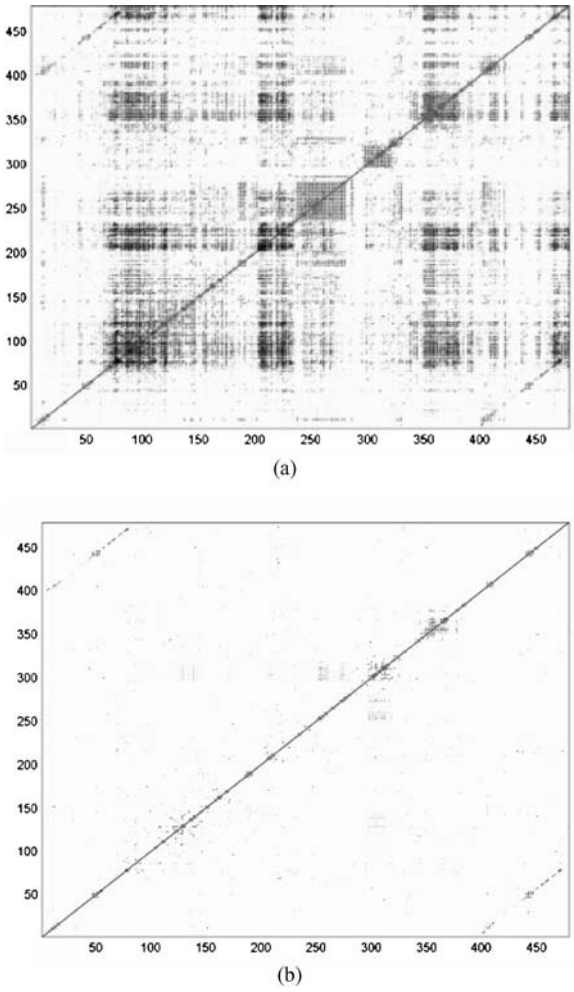

Figure 6. (a) shows a visual similarity matrix constructed from images collected from an exploration run (around the path shown in Fig. 17). (b) shows the visual similarity matrix after rank reduction. Note that the off-diagonal dark line (which signifies the true loop closure) has not been affected by the rank reduction whereas visually ambiguous regions within the similarity matrix have been removed.

5.1. Spectral Decomposition

We will now discuss how decomposition of M can remove the effects of these visually ambiguous regions in loop closure detection. $\mathtt{M}$ is a symmetric real $n \times n$ matrix. There is an orthogonal matrix V and a diagonal D such that $\mathtt{M} = VDV^T$. The columns of V are eigenvectors, $v_1 \ldots v_n$ of $\mathtt{M}$ and diagonal entries of D are its eigenvalues $\lambda_1 \ldots \lambda_n$.

$$M = [v_1 \ldots v_n] \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{bmatrix} \begin{bmatrix} v_1^T \\ \vdots \\ v_n^T \end{bmatrix}$$
$$= \sum_{i=1}^{n} v_i \lambda_i v_i^T$$

where v_i is the ith column vector of V, λ_i is the ith diagonal entry of D, $\lambda_1 \geq \lambda_2 \geq \cdots \lambda_k \geq 0$ and the rank of $\mathtt{M}$ is equal to n. The outer product expansion form for the eigenvalue decomposition, $\mathtt{M} = \sum v_i \lambda_i v_i^T$, expresses $\mathtt{M}$ as a sum of rank one matrices $M_i^{\ominus} = v_i \lambda_i v_i^T$. Figure 8(a) is the first ($i = 1$) rank one matrix of the visual similarity matrix shown in Fig. 6(a). Structurally, these two matrices are very similar. Figure 8(b) shows nine different images associated with high scoring cells in this outer product matrix. All of these images have significant amounts of vegetation within the images. The decomposition has extracted the dominant theme within this particular environment. Indeed, there is vegetation scattered throughout this particular environment explored by the robot witnessed by the distribution of high scoring cells along the main diagonal. Continuing, Fig. 9 shows the rank one matrix associated with the second largest eigenvalue and its associated set of images. High scoring cells in this matrix are concentrated in a smaller area while low scoring cells are spread throughout the matrix. The nine images suggests that images with rectangular structures such as bricks and windows are generally associated with this particular eigenvector. All of these images are images

Figure 7. An erroneous loop closure detection in the presence of ambiguity. This is one of the sequence matches that results by applying the sequence detection algorithm directly to the matrix in Fig. 6. Although the two sequences are indeed similar, repeating visual entities such as wall patterns, window styles, vegetation, results in a Type I (false positive) error.

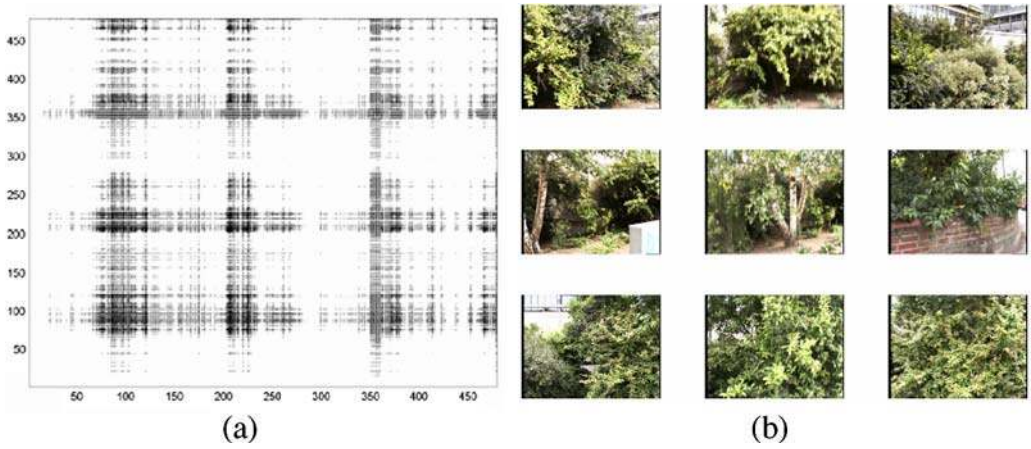

Figure 8. On the left is the first rank one approximation ($M_1^{\ominus} = \lambda_1 v_1 v_1^T$) of the visual similarity matrix shown in Fig. 6. On the right are nine images with the highest scores in the matrix. These images are mostly images of vegetation which, because they spread across the matrix as dark colored cells, constitute a common theme throughout the database.

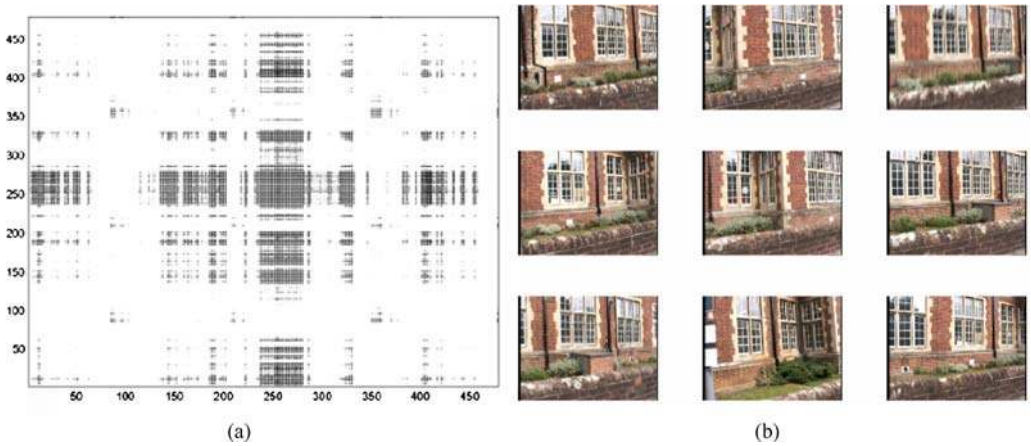

Figure 9. On the left is the second $M_2^{\ominus}$ rank one approximation of the visual similarity matrix shown in Fig. 6(a). On the right are nine images with the highest scores in the matrix. These images are mostly images with rectangular shaped entities such as windows and bricks. Given that the robot was exploring in an urban environment, it is not surprising that such features are common.

captured around the circumference of a building comprising red brick walls interlaced with white, Georgian windows. Many buildings have similar facades. We should be concerned that a false loop closure could be triggered when the robot traverses to the rear of the building, which appears similar to its front.

These few examples suggest that the principal eigenvectors of M associate with "themes" which permeate a particular environment. While these themes, which capture common similarity between many images, are useful for summarizing an environment they are detrimental when detecting loop closure. Consider the images in Fig. 9. The images are all of a single building presenting a distinctive Victorian architecture on all of its facades (the building is actually at the northern apex of the aerial photograph in Fig. 17). When naively passing the matrix of Fig. 6(a) through the sequence extraction routine of Section 4 false positives result—for example the one shown in Fig. 7. The top row is of the north eastern side of the northern apex while the bottom row is of the western side (camera looks right then left). Although these two sequences do look similar, the ambiguity resulting from the repetitive nature of the building and foliage on the far side of the road causes a false positive detection. In Section 5.2 we shall describe how this problem can be addressed.

5.2. *Removing Common Mode Similarity*

Generally, images with visually ambiguous artifacts are detrimental to our needs because they appear similar to many different scenes. We can expect to enhance and increase robustness in loop closure detection by removing the effects of such artifacts.

By decomposing the similarity matrix into a sum of outer products we are able to remove the *effects* of common mode similarity without removing the images themselves. This is an important point—an image can contain both visually ambiguous artifacts and globally salient artifacts. A similar approach was used in Alter et al. (2000) when removing principal "eigengenes" and "eigenarrays". The relative magnitude of λ_i is a measure of the degree to which of the dyad $M_i^{\ominus} = v_i \lambda_i v_i^T$ expresses the *overall* structure of the similarity matrix M.

If themes are responsible for the dominant structure in M then, because $\sum_{i=1}^{r} \mathbf{v}_i \lambda_i \mathbf{v}_i^T$ is the best rank-r approximation to M under the Frobenius norm, we should expect their effect in M to be captured in the dominant eigenvalues/vectors. Thus, we can diminish the effect of visual ambiguity/repetitive scene structure by reconstructing M by omitting the first r terms of the summation in Eq. 8. We shall now discuss how to choose r.

Figure 10(a) shows a distribution of eigenvalues obtained from the EVD of the visual similarity matrix in Fig. 6(a). The magnitudes of the eigenvalues initially drop dramatically before the rate of decline levels off—there are a few principal eigenvectors that go a long way to describing the entire matrix.

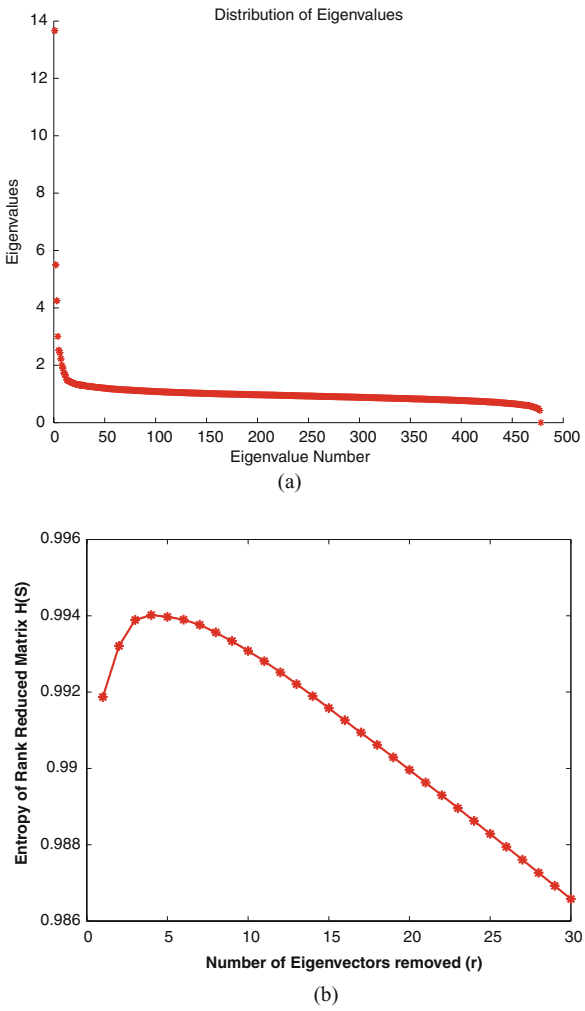

Figure 10. (a) shows a typical distribution of eigenvalues of the visual similarity matrix in Fig. 6 (a). (b) shows how, for this M, $H(\mathtt{M}, r)$ varies with r, the number of rank reductions applied.

For an $n \times n$ M, we define the relative significance, $\rho(i, r)$ of λ_i to the last $n - r$ eigenvalues as

$$\rho(i, r) = \lambda_i / \sum_{k=r}^{n} \lambda_k \tag{8}$$

Using this we can measure the complexity of decomposition of M as an entropy

$$H(\mathtt{M}, r) = \frac{-1}{\log(n)} \sum_{k=r}^{n} \rho(k, r) \log(\rho(k, r)). \tag{9}$$

Equation (9) measures the complexity of the composition of M with first $r - 1$ dyads removed. $H(\mathtt{M}, r) = 0$

corresponds to an ordered and redundant $\mathtt{M}$ which can be represented by a single eigenvector. $H(\mathtt{M}, r) = 1$ corresponds to a disordered similarity matrix where all eigenvectors are equally expressive. Our approach is to sequentially remove outer-products from $\mathtt{M}$ until $H(\mathtt{M}, r)$ is maximised leaving a similarity matrix in which no one single theme dominates. We may replace $\mathtt{M}$ with a rank reduced version

$$\mathtt{M}' = \sum_{i=r^\star}^{n} \mathbf{v}_i \lambda_i \mathbf{v}_i^T \qquad r^\star = \arg\max_r H(\mathtt{M}, r) \tag{10}$$

Figure 10(b) depicts $H(\mathtt{M}, r)$ as a function of r (using the data set whose eigenvalues are shown in Fig. 10(a). For this particular case, the maxima was reached after removing the first four outer products. Figure 6(b) shows the final rank reduced matrix.

The rank reduction technique successfully removes visually ambiguous regions without removing the important loop-closing off-diagonals—see for example Fig. 11(a). Applying the modified Smith-Waterman algorithm on $\mathtt{M}'$ we can successfully detect loop closure as shown in Fig. 11(b). Both of these figures correspond to the matrix shown in Fig. 6(a) which has substantial intrinsic ambiguity. Figure 12 shows the images which constitute the final (correct, in contrast to Fig. 7) loop closure sequences $\mathcal{A}$ and $\mathcal{B}$.

Of course, applying the procedure to simpler data sets with less ambiguity also works. For example, Fig. 3 showed a similarity matrix created from images taken while driving around the exterior of a 1970s tower block. The rank reduced version and extracted loop closure sequences are shown in Fig. 13.

5.3. *Sequence Significance*

We wish to evaluate our confidence in the pairing between $\mathcal{A}$ and $\mathcal{B}$. Is it really due to genuine loop closure? We need to be convinced that this score is due to the temporal ordering of the revisited scenes and that a random ordering of the images would not yield a similar maximal alignment. Maxima resulting from a randomized population can be well described by the Gumbel or Extreme Value Distribution (E.V.D.) (Gumbel, 1958):

$$p(\eta_{\mathcal{A},\mathcal{B}}) = \frac{1}{\beta} \exp^{-z} \exp^{-\exp^{-z}} \tag{11}$$

where

$$z = \frac{\eta_{\mathcal{A},\mathcal{B}} - \mu}{\beta} \tag{12}$$

where $\eta_{\mathcal{A},\mathcal{B}}$ is the maximal alignment score, μ is the mean of the distribution and β is the standard deviation of the

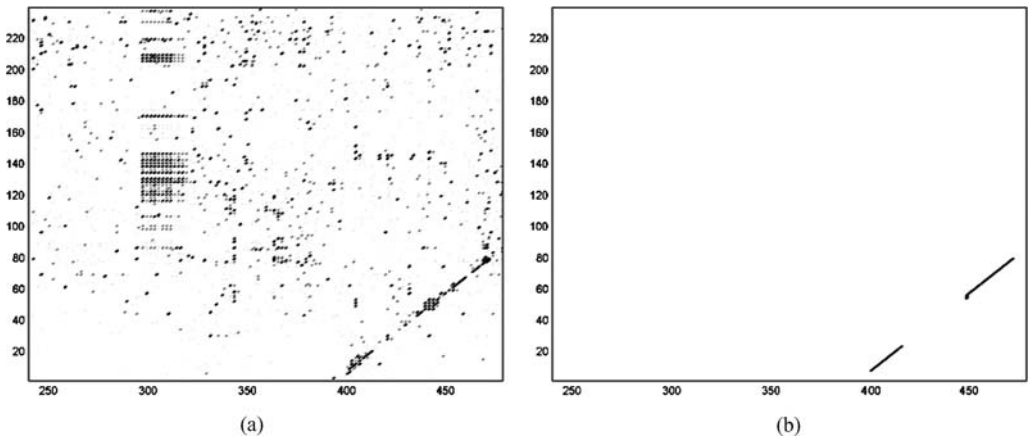

Figure 11. (a) shows a zoomed in portion of a lower triangular matrix of a visual similarity matrix after loop closure has occurred. The visually ambiguous regions has been removed through rank reduction. (b) shows the result of applying a local sequence alignment algorithm to find the most significant local alignment.

Figure 12. Positive, correct loop closure detection in an ambiguous setting. The top row is sequence $\mathcal{A}$ and the lower row sequence $\mathcal{B}$. The original similarity matrix is shown in Fig. 6.

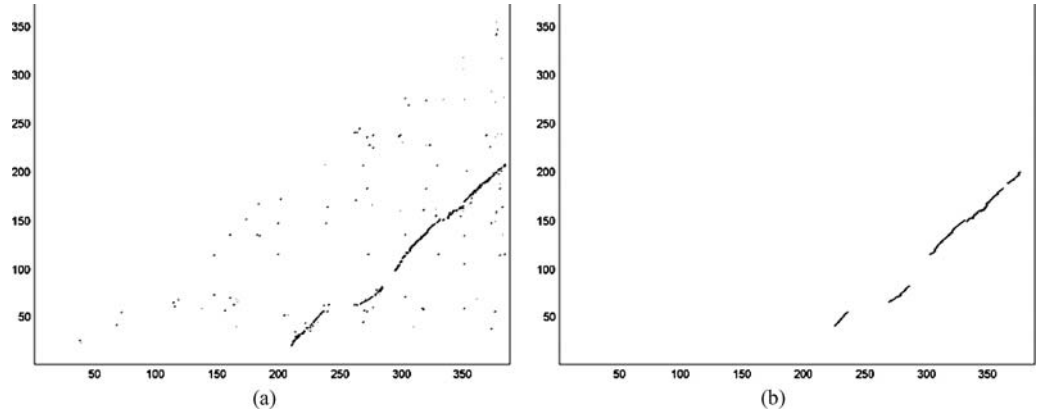

Figure 13. On the left is a lower triangular matrix of the largely unambiguous similarity matrix shown in Fig. 3 after rank reduction. Contrast this figure with Fig. 4(a). On the right is the result of applying the modified Smith-Waterman algorithm to find matching sequences $\mathcal{A}$ and $\mathcal{B}$.

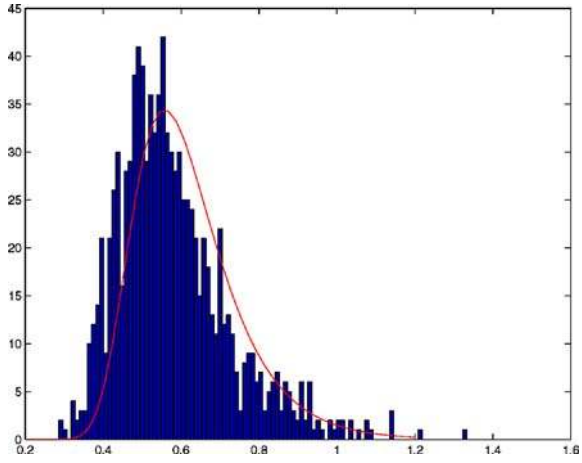

Figure 14. Typical Gumbel distribution of maximal segment scores ($\eta_{\mathcal{A},\mathcal{B}}$) from 1000 shuffles of the similarity matrix.

distribution. This p.d.f can be used to judge the significance of the maximal alignment score for the pair of matching image subsequences. Adopting the approach suggested in Altschul and Erickson (1985), we randomly shuffle the visual similarity matrix and obtain the maximal alignment score each time. This results in a distribution similar to that shown in Fig. 14 (1000 shuffles). The distribution parameters μ and β are estimated (we fit the parameters using Levenburg Marquardt) from the histogram of maximal alignment scores.

Equipped with estimates $\hat{\mu}$ and $\hat{\beta}$ and the closed form c.d.f of the distribution we can evaluate the probability of scores greater than or equal to $\eta_{\mathcal{A},\mathcal{B}}$ conditioned on all N images:

$$P(\eta \geq \eta_{\mathcal{A},\mathcal{B}} \mid \mathtt{M}) = 1 - \exp^{-\exp^{-z}} \tag{13}$$

Equation (13) allows the evaluation of the probability that an extracted sequence of image matches, $\langle \mathcal{A}, \mathcal{B} \rangle$, with score $\eta_{\mathcal{A},\mathcal{B}}$, could have been generated at random from $\mathtt{M}$. The differences between the sequence score $\eta_{\mathcal{A},\mathcal{B}}$ obtained from the original, temporally ordered $\mathtt{M}$ and those obtained from the randomly shuffled versions are solely attributable to the topology or connectedness of the spatial locations at which the vehicle captured the images. Thus Eq. (13) can be used to evaluate the probability, conditioned on all previous scene appearances, that the detected sequence does indeed indicate a bona-fide loop closure.

An important distinction between global localisation and loop-closing is that in the former it is often known *a-priori* that a correspondence between a vehicle's local scene and a stored representation of the workspace exists. In the case of loop-closing this is not the case—the vehicle may never revisit the same location. It is also possible that within the totality of images, $\mathcal{I}$, multiple loop closure events are captured. The probabilistic formulation in Section 5.3 allows for both these situations. Sequences can be extracted from $\mathtt{M}$ in decreasing order of alignment score, $\eta_{\mathcal{A},\mathcal{B}}$, until the probability of false positives associated with $\eta_{\mathcal{A},\mathcal{B}}$ becomes excessive. We typically set a threshold of 0.5%.

The entire loop-closure detection process can now be summarised:

1. From n images build a $n \times n$ similarity matrix $\mathtt{M}$ as described in Section 3.
2. Remove Common mode similarity via rank reduction as described in Section 5.2.
3. Estimate Gumbell distribution parameters from the rank reduced similarity matrix, $\mathtt{M}'$ as described in Section. 5.3.
4. Extract highest scoring sequence from M'.
5. Test significance of the alignment score using Eq. (13), if acceptable, advise loop closure, go to 4.
6. End

Table 2. Description and references to four data sets.

Data set	Purpose	Discussion & Illustration
Thom	Benign workspace, producing a clean similarity matrix with little ambiguity	Section 3. Figures 3 and 5.
Jenkin	Medium sized loop around a set of buildings. Used to showcase combination of loop closure detection with SLAM system	Section 6.1. Figures 16 and 12.
New College	Larger data set with combination of visual themes. again a noisy data similarity matrix results with hard to discern loop closures.	Section 6.2. Figures 19 and 18.
Cloisters	Repetitive visual structure producing a noisy similarity matrix	Section 6.3. Figures 26 and 28.

6. Results

Data sets of images captured from various exploration runs can be found in http://www.robots.ox. ac.uk/∼klh/dataset.htm. They are divided into four sets: "Thom", "Jenkin", "New College" and "Cloister". The qualities of the data sets are summarised in Table 2. The following sections examine the performance of our loop-closure detection system when applied to these data sets while 6.4 discusses the execution times.

6.1. Scenario I

To further illustrate the effectiveness of our approach in supporting SLAM we consider a 3D, laser-based SLAM scenario in an outdoor environment. The SLAM algorithm was described in Section 2. For every 0.5 m the robot traverses and for every 30 degrees change in heading, an image is captured. The camera orientation toggles after capturing an image between 60 degrees left and 60 degrees right. Every image and laser scan captured is time stamped. The robot travelled a distance of just over 370 meters before returning back to a previously visited location. The similarity matrix constructed for this environment is that shown earlier in Fig. 6(a). Figure 11(a) shows a portion of the same matrix following rank reduction and (b) highlights the loop closing sequence extracted. The matching sequences $\mathcal{A}$ and $\mathcal{B}$ are shown in Fig. 12. Figure 15 shows the trajectory of robot poses maintained by the SLAM system (Newman et al., 2006) before loop closure. Note that the estimated position of the robot is more than 100 meters off its actual position when it has completed a loop—well outside the 5-sigma bound of the vehicle marginal. Many standard loop closure detection techniques based on pose estimates will fail under such gross errors.

6.1.1. Loop Closure Geometry. We have a system that, given a sequence of time-stamped views, can detect proximity to a previously visited location. Now however, to execute the loop closure, we need to know the geometry of the loop closure—the euclidean transformation

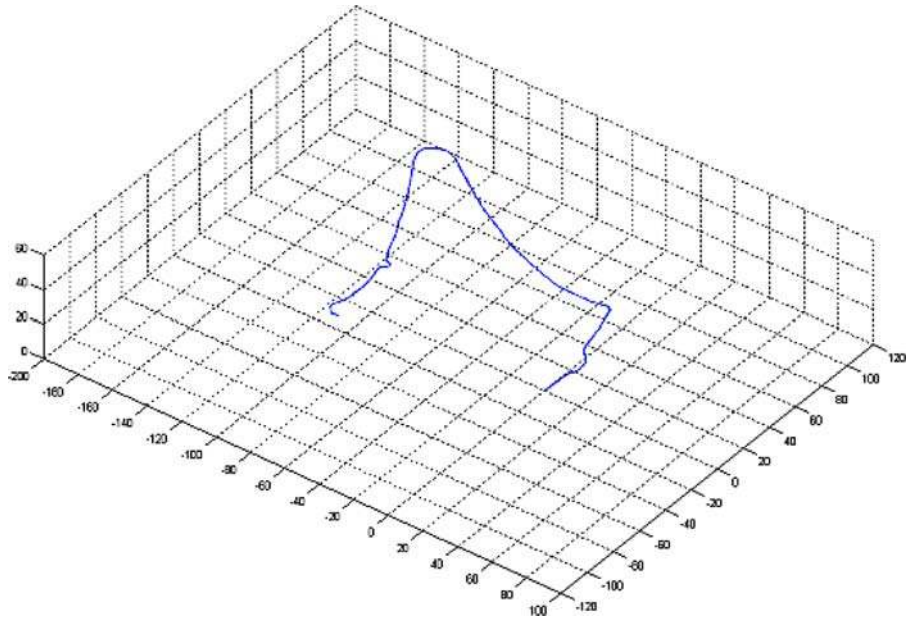

Figure 15. The pre-loop closing estimated trajectory of the vehicle. The sharp turn compounded with long gently curved traversals leads to a gross error in position estimate. Each cell is 20 by 20 m.

between recent and past views that constitutes the loop closure.

One option would be to use image time to index into the pose state **x** vector described in Section 2, which is, after all, a sequence of past poses, to find which previous pose i occupied the scene we are now revisiting at time j. However, this approach has problems when it comes to undertaking a laser scan match to deduce *precise* estimate of the interpose transformation $z_{i,j}$ – without a reliable prior or "seed solution" the iterative scan matching method we adopt is prone to converge to an incorrect minima. At the same time, exhaustive search in 6D is prohibitively slow. Instead, we shall use the sequences $\mathcal{A}$ and $\mathcal{B}$ and laser range data to estimate $T_{i,j}$.

Consider the following common projective model of two identical cameras with projection matrices P and P' (Hartley and Zisserman, 2000). A homogenous 3D image scene point $X = [X, Y, Z, 1]^T$ is imaged at $x = PX$ for the first camera and $x' = P'X$ for the second camera. Without loss of generality the origin can be fixed at the center of the first camera, and, if the second camera center is parameterized by a rotation matrix R and a translation t with respect to the origin, then P and P' can be written $K[I \mid 0]$ and $K[R \mid t]$ respectively. Here K is the matrix of intrinsic camera parameters. In the case of calibrated cameras (K known) the image points, x and x', are related by the "Essential matrix" E such that $x'^T Ex = 0$.

The determination of relative camera poses via decomposition of the essential matrix has been used to good effect in robot localization (Kŏsecká and Yang, 2004; Royer et al., 2004) and SLAM navigation (Eustice et al., 2004). Given two image views of the same scene, five points of correspondence are selected for use in an implementation of the algorithm presented in Nister (2004). This "Five Point Method" is capable of dealing with planar degeneracy if the matrix of intrinsic camera parameters is known.

The essential matrix has a convenient structure. It can be written in terms of R and t as $[t]_\times R$ where $[t]_\times$ denotes the 3×3 skew symmetric (cross product) matrix constructed from t. Given the elements of E this decomposition yields four possible solutions for R and t. The correct solution is selected by application of a final cheirality constraint; scene points must be in front of the cameras. The two images used in the geometry estimation are those with the greatest similarity score within the image sequence returned from the loop-closure detector. The resulting t is correct only up to scale and we look to range information from the laser data to perform the metric upgrade.

Given the instantaneous rigid transformation between the laser scanner and the camera, 3D laser data can be expressed in the 3D coordinate frame of the camera and projected onto the image plane. For each of the five visual features, the nearest projected laser-range points are found. Out of the five image features, the image feature which has the closest projected laser range point is selected. The 3D position of this image feature is now known allowing the final scale ambiguity to be removed yielding metric R and t.

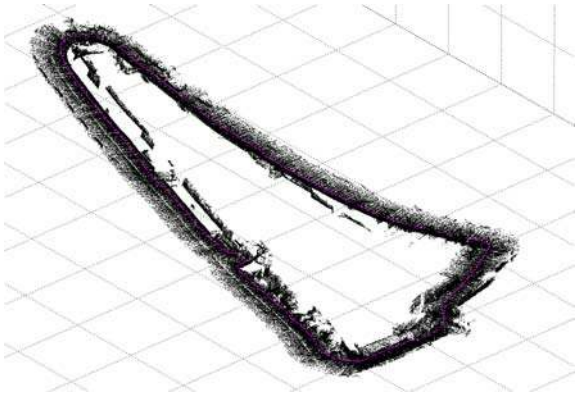

Figure 16. Using loop closure detection. A 3D, post loop-closing, SLAM map. The total length of this loop is 371 m.

Given estimates of R and t the iterative laser scan matching can proceed, starting with these estimates as an initial solution to $\mathbf{T}_{i,j}$. The scan matcher further aligns the two scans, refining estimates of $\mathbf{T}_{i,j}$ to use as a measurement on the SLAM state vector.

Figure 16 shows a 3D laser map after loop closure detection using our technique. The huge discrepancy in pose estimates of the robot does not affect the performance of our loop closure detection technique. The actual loop closure was achieved via constrained nonlinear optimization in a manner similar to that described in Estrada et al. (2005) and Cole and Newman (2006). Figure 17 is a plan view of the final estimated vehicle trajectory superimposed over an aerial photograph of the workspace. A metric grid has been placed over the area of interest—each grid box is 20 m by 20 m.

6.2. *Scenario II*

We now consider a more challenging scenario moving around through both gardens and buildings at different elevations. This setting is beyond the capabilities of our current SLAM system and so we use a GPS sensor to provide ground truth. The experiment proceeded as before: for every metre travelled and for every 30 degrees change in heading, an image is captured. The camera toggles from 60 degrees left and right. Each image captured is time-stamped. Throughout the experiment, GPS NMEA strings are logged.

Figure 18 shows an aerial image of the environment where the experiment was conducted. GPS estimates of the robot's position are plotted onto the image as white

Figure 17. An aerial image of the workspace with the final estimated vehicle trajectory and metric grid superimposed upon it.

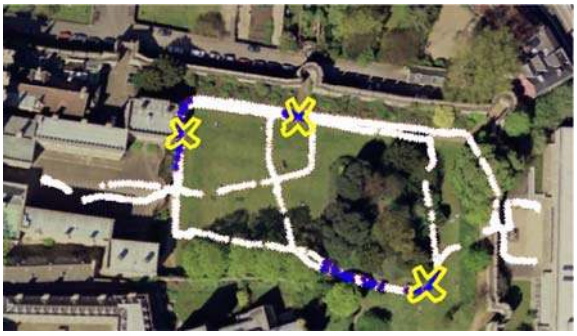

Figure 18. An aerial image of the test scenario—New College Gardens. The position of the mobile robot is measured by a GPS receiver with positional uncertainty of around 5m. The path taken by a mobile robot is marked with white crosses. There are various breaks in GPS signal reception. The large yellow crosses mark the positions where loop closure is detected.

crosses. Due to intermittent GPS reception, certain portions of the robot's trajectory are missing.

In all, 568 images were collected during this experiment. The visual similarity matrix for this environment is shown in Fig. 19(a). This is a highly visually confusing environment as there were many repetitive patterns. From the visual similarity matrix, it can be observed that there are plenty of visually ambiguous regions, marked by the dark squares. A rank reduced matrix is shown in Fig. 19(b) after removing the top five eigenvalues and corresponding outer products. Due to multiple loop closure at different locations, there are multiple dark off-diagonals.

Figure 20(b) shows nine different images associated with high scoring cells in $v_1\lambda_1 v_1^T$. All of these images have significant portions of vegetation within the images. The decomposition has extracted the dominant 'theme' within this particular environment. The distribution of the vegetation in the environment can be seen from the distribution of high scoring cells along the main diagonal of $v_1\lambda_1 v_1^T$ shown in Fig. 20(a). The white portion corresponds to images of the building, images of the courtyard and images taken in the middle of the open field. Figure 21(b) shows nine different images associated with high scoring cells in $v_2\lambda_2 v_2^T$. It is harder to discern the category of images—perhaps a bias towards textured material like leaves and grainy walls. Figure 22(b) shows nine different images associated with high valued cells in $v_3\lambda_3 v_3^T$. All of the photos contain images of the wall encircling the park—witnessed by the spread of dark cells throughout the matrix. Finally, Fig. 23(b) shows nine different images associated with $v_4\lambda_4 v_4^T$. These images are mostly images of a building seen only at the start and end of the experiment. The matching sequences $\mathcal{A}$ and $\mathcal{B}$ are shown in Fig. 24. Although this is a particularly challenging environment in which to detect loop closure, the system is able to extract suitable loop closure evidence. In the third column along the sequence, the matched images look very different due to a wide difference in viewpoint. Nevertheless the overall scene similarity accumulated along a trajectory has enough statistical significance to imply a loop-closure event.

6.2.1. Analysis. This environment is a particularly challenging one with the structure of the initial $\mathtt{M}$ being indicative of a general lack of distinctive images. Three main loops were successfully detected as depicted in Figs. 25(a)–(c) as large crosses. The detection of the second loop (b) is an excellent example of the kind of challenging detection that our architecture enables. We note that the northernmost east-west border remains ambiguous even after $H(\mathtt{M}, r)$ is maximised (in this case by rank reduction by 5).

Analyzing the Type II errors (missed positives) provides more insight into our approach. Starting with the extreme right of the aerial image shown in Fig. 25(d), a small loop (marked by a yellow ellipse) was not detected. This is because there is not enough overlap between the first pass and the second pass. In fact, there is only one point of intersection between them at the entrance of the courtyard. Consequently, there should only be one correct image match from that loop closure. However, the single image did not score significantly enough to trigger a loop closure. This is a limitation of our approach—a certain amount of overlap between the first pass and second pass must occur before a statistically significant alignment

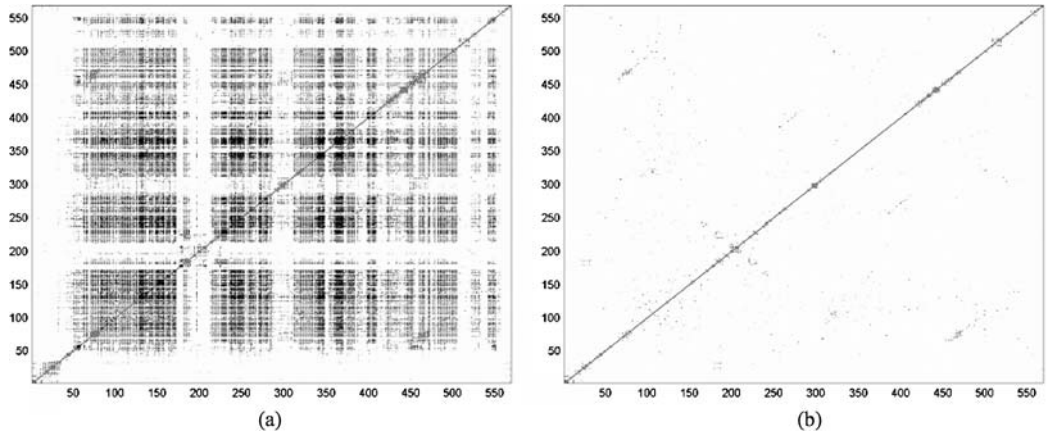

Figure 19. (a) shows the highly ambiguous visual similarity matrix constructed from images from the "New College" data set. (b) shows the visual similarity matrix after rank reduction. Note that the off-diagonal dark line (which signifies the loop closure) has not been affected by the rank reduction whereas visually ambiguous regions within the similarity matrix have been removed. Due to multiple loop closure at different locations, there are multiple dark off-diagonals.

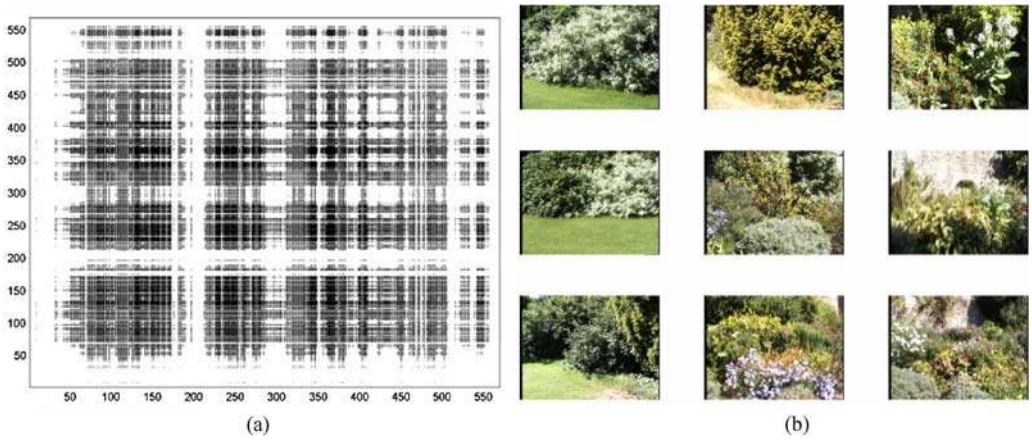

Figure 20. (a) shows the first rank one approximation of the visual similarity matrix shown in Fig. 19. (b) shows nine images with the highest scores in $v_1\lambda_1 v_1^T$. These images are mostly images of vegetation.

Figure 21. (a) The matrix $v_2\lambda_2 v_2^T$ extracted from M shown matrix shown in Fig. 19. (b) shows nine images with the highest scores.

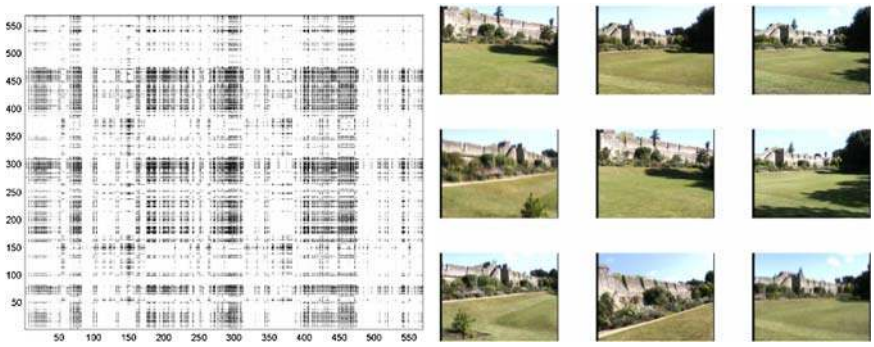

Figure 22. (a) shows the third approximation, $v_3\lambda_3 v_3^T$, of the matrix shown in Fig. 19. (b) shows nine images associated with the largest elements.

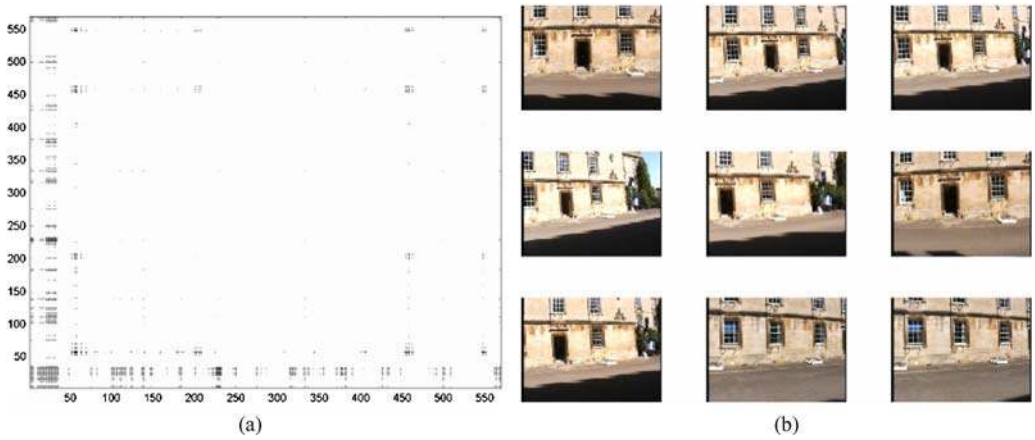

Figure 23. (a) $v_4\lambda_4 v_4^T$ (b) images associated with the nine largest element values. These images are mostly images of a building seen only at the start and end of the experiment.

Figure 24. Two matching sequences of images ($\mathcal{A}$, $\mathcal{B}$). These images correspond to the bottom right loop closure in Fig. 25(b). It is not immediately apparent that this is indeed a loop-closure. In the third column along the sequence, the two images look very different due to a substantial difference in viewpoint.

score, $\eta_{\mathcal{A},\mathcal{B}}$, can accumulate. The precise amount of overlap required to acquire significance depends on the environment itself and the numerical similarity between individual images as described in Section 5.3. Naturally, more ambiguous scenes require longer sequences.

The middle yellow ellipse in Fig. 25(d) marks another potential loop closure that was not detected. Again, there was only a small area of trajectory coincidence. A bigger problem here is that in the center of the park all the scene diversity (the borders) is in the far field and coupled with a 90° difference in heading this leads to utterly different images. This is a strong motivator to use an omni-cam instead of a standard pan/tilt/zoom camera.

Finally, the yellow ellipse on the extreme left highlights the last potential loop closure that was not detected. The reason for this failure can be seen in Fig. 23. The subtraction of $v_4\lambda_4 v_4^T$ removed the elements of $\mathtt{M}$ indicating similarity between images of the building facades. Essentially this loop was not detected because of a high likelihood of false loop closure caused by the repetitive architecture of the building. Our policy, to support SLAM, is to strongly prefer Type II errors over Type I errors.

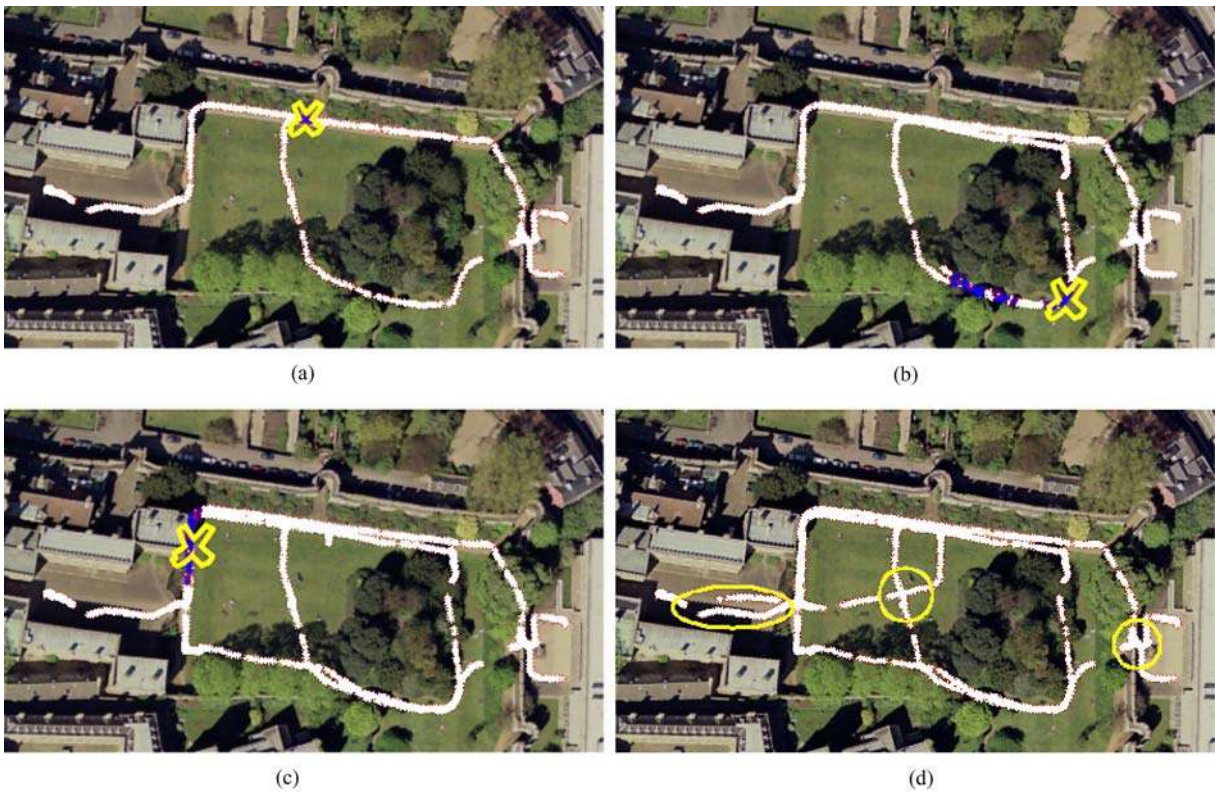

Figure 25. (a) shows the first loop closure event correctly detected. Loop closure was detected right at the start of the loop. Interestingly, none of the ensuing eastern leg was detected as a return to a previous location. The next detection is shown in (b) when the vehicle rejoins a previous trajectory following a excursion into a courtyard. (c) shows the third loop closure event. The first half of the loop was well detected by our loop closure but, once again, something about the east-west botanical border inhibited loop closure detection. Finally, (d) highlights loop closure events that were not detected with yellow ellipses.

6.3. Scenario III

In our final loop closing experiment, our algorithm is put to test in an environment where, by design, every local scene is visually similar. The question is whether our loop closure detection will fail where there are no obvious globally distinct scenes. This experiment took place in the cloister of a college, see Fig. 26. The control parameter settings for the camera were the same as the previous experiments. A sequence of 268 images were collected. The similarity matrix is shown in Fig. 27(a). The cleaned $\mathtt{M}'$, matrix is shown in Fig. 27(b).

Figure 26. A view from inside the cloisters which by intention present a repetitive and ambiguous architectural theme.

6.3.1. Analysis. The two matching sequences are shown in Fig. 28. It can be confirmed that this is indeed a correct match by observing the presence of plaques and the occasional statue. Note that given the background of common mode similarity between images it took twelve images to accumulate enough evidence to render the alignment score $\eta_{\mathcal{A},\mathcal{B}}$ "significant" and trigger a loop closure. A SLAM map resulting from the loop-closure is shown in Fig. 31.

This environment can be summarized as a continuous stretch of arched windows and a continuous stretch of wall. Only two principal eigenvectors were selected to be removed by our entropy maximization method. The two outer products and corresponding images are shown in Figs. 29 and 30. As expected these correspond to the two aforementioned dominant themes.

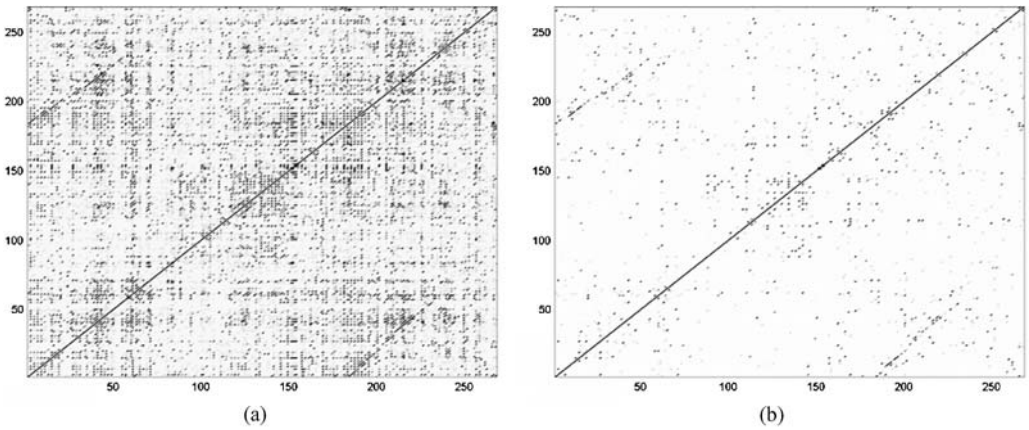

Figure 27. (a) shows a visual similarity matrix constructed from images collected from an exploration run around the cloister shown in Fig. 26. (b) shows the matrix after rank reduction.

Figure 28. Loop closure detection in the cloisters. The above sequence of twelve images triggered the loop closure in this repetitive environment. The validity of the detection can be verified by noting the presence of plaques and other antiquities across paired images.

6.4. Timing

Table 3 shows the execution times for the various components of the loop closure detection scheme described in this paper. Results are shown for the four vision data sets used throughout the paper. The code was written in C++ and run on a 2 GHz Centrino Processor (Samsung X50 Laptop) with 1 GB of RAM.

The nature of the SLAM algorithm we use means that loop closures can be applied between any two poses, past or present, and so the loop closure detection process need not run in step with the state estimation. We don't expect loop closure events to be common-place so we can afford run-times of minutes. However for truly large data sets we run the risk of falling more and more behind. For example running with two thousand images the procedure takes over 10 minutes. As stated earlier in Section 3, it is not necessary to build a new vocabulary for every run—it can be a one-shot off-line task from a large training data set. Indeed, as can be seen from

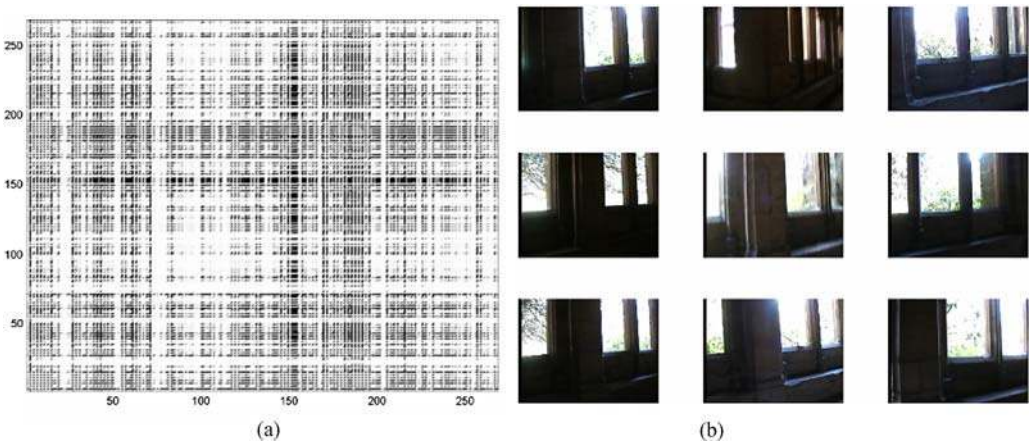

Figure 29. (a) shows the first rank one approximation of the visual similarity matrix shown in Fig. 27. (b) shows nine images with the greatest cell values—all are images of the windows.

Table 3. Run times for the four different data sets described in Table 2.

		DataSet			
# Scenes		Cloisters 212	Thom 387	New College 510	Jenkin 485
$\mathcal{V}$ Creation	Pass 1 (s)	7	27	29	134
	Pass 2 (s)	4	8	10	20
	$\lvert\mathcal{V}\rvert$	2603	3822	3138	5982
Detection	M Creation (ms)	6	200	400	900
	Rank Reduction (s)	0	1	3	3
	EVD Estimation (s)	2	10	18	16
	Loop Extraction (ms)	4	80	90	100
	Total (s)	2	11	21	20

Figure 30. (a) shows the second rank one approximation to the similarity matrix in Fig. 27 and (b) shows nine images with the highest cell values. All of the photos are images of the wall which is always in view hence dark cells are spread throughout the matrix.

the table, this opportunity becomes important as both the number of scenes grows and the number of descriptors per image increases (the Jenkin data set timings have , on average, just under three times as many features per image as the other three data sets). As the number of scenes increases the cost of the rank reduction becomes more significant and for truly huge data sets (e.g 5000 images) it dominates. We are working on a way to perform the rank reduction incrementally so the entire procedure need not be repeated every time a new scene is captured.

6.5. *Using Laser Images*

Our approach is not limited to similarity matrices for visual images. It is equally applicable to any similarity matrix which is formed by an appropriate similarity function that compares sensor observations from different local scenes. We demonstrate the applicability of our approach when using raw 2D, 180° laser scans. The details of the similarity function $\mathcal{S}(L_i, L_j)$ which compares two scans L_i and L_j can be found in Ho and Newman (2005b). In summary the method considers the spatial appearance of the two scans and compares them in terms of their entropy, shape and interest points. As required, the function returns a similarity score between zero and one. Figure 32(a) shows a spatial similarity matrix (SSM) constructed from 2D laser scans collected from an exploration run around a building. We can see the off-diagonals for the laser similarity matrix are less defined, reflecting the diminished certainty in matches coming from less discriminative (relative to the visual images) data. Figure 32(b) shows the SSM after rank reduction.

Figure 33(a) shows a lower triangular matrix of the SSM shown in Fig. 32(b). Figure 33(b) shows the result of applying the local sequence alignment algorithm. Figures 34, 35 and 36 illustrate the rank one matrices based on the top three eigenvalues and their associated laser scans. Our technique has successfully detected loop closure events using laser scans despite their reduced descriptive power. We believe that our approach could be applied to other sensing modalities for which a suitable scene similarity function $S(I_i, I_j)$ can be defined where I_i, I_j are scenes captured in the native sensor modality.

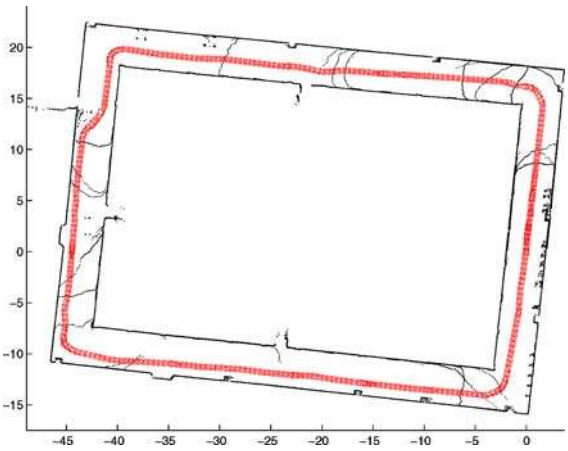

Figure 31. A post loop closure 2D SLAM map of the cloister with robot trajectory. The map is built with a 2D laser scanner and the southern, east-west traversal is as shown in Fig. 26. The curved structures appearing to cross the cloister corridors are artifacts stemming from the uneven floor pitching the vehicle and causing the laser scans tp intersect the floor before the walls.

7. Application to Multiple Vehicles

The approaches of current collaborative multi-robot map building algorithms can be broadly classified into three main categories: (1) merging sensory data from multiple robots with known data association between features in local maps built by different robots (Fenwick et al., 2002); (2) detecting other robots to determine relative position and orientation between local maps (Fox et al., 2000; Konolige et al., 2003) or assuming relative poses between robots are known (Thrun, 2001); (3) deriving the transformation between robots' coordinate systems through the matching of landmarks (Dedeoglu and Sukhatme, 2000; Thrun and Liu, 2003). Generally, algorithms with strong assumptions about known data association or relative poses have been limited to simulatation or highly engineered experiments. The algorithms that have worked with real world data with weaker assumptions have been limited to those that rely on detection of other robots. This approach means that, in an exploration task, the robots might duplicate each other's work by exploring the same environment without being aware of each other's past accomplishments. Alternatively, the robots may hypothesize their relative positions and try to congregate at a hypothesized meeting point. This allows the robots to determine accurately each other's relative poses but distracts them from the task of exploration (Konolige et al., 2003). A more exploration-efficient way of joining local maps is to detect map intersections, independently of coordinate frames, and then align the maps.

Map intersection detection can be considered to be a loop closing problem; where one robot "closes the loop" of the map built by another (Ho and Newman, 2005a). Here we use visual appearance to detect intersections between local maps built by multiple robots. These common intersections can be used to align the maps.

7.1. *Map Joining Techniques*

Data association is an infrequently considered problem in multi-robot mapping. In Thrun and Liu (2003) this is addressed by introducing an algorithm that aligned local maps into a global map by a tree-based algorithm for searching similar looking landmark configurations. The landmark configuration consists of relative distances and angle between a triplet of adjacent landmarks. Another landmark-based algorithm for map matching was described in Dedeoglu and Sukhatme (2000), which combined topological maps of indoor environments. Landmarks such as corners, T-junctions, ends-of-corridor and closed doors were stored in the search space for

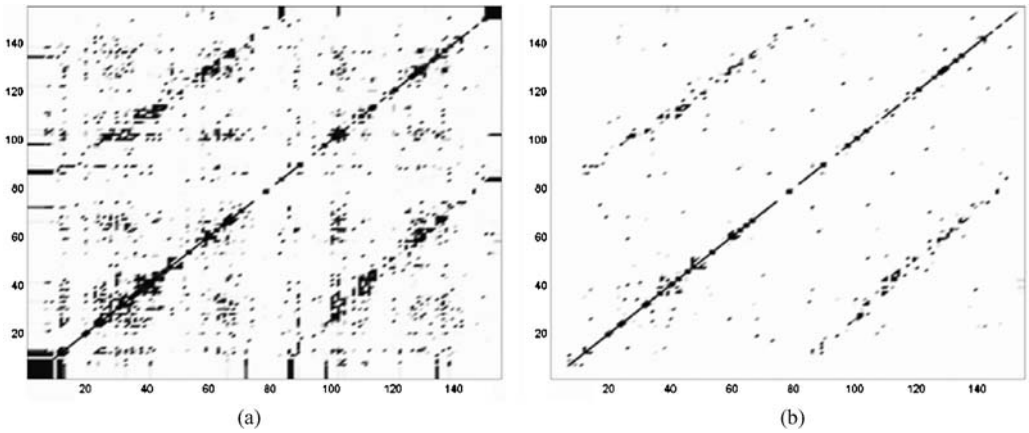

Figure 32. (a) shows a spatial similarity matrix (SSM) constructed from laser scans collected from an exploration run around a building. (b) shows the similarity matrix after rank reduction.

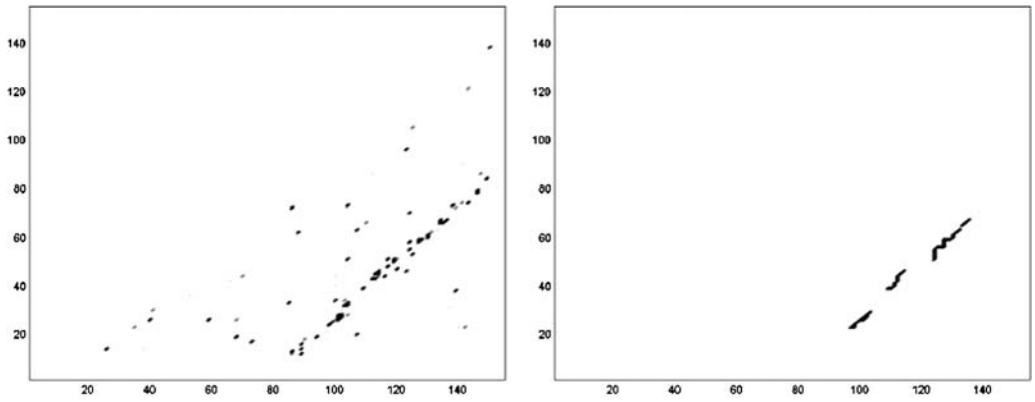

Figure 33. On the left is a lower triangular matrix of a spatial similarity matrix (SSM) after loop closure has occurred. On the right is the result of applying the sequence extraction algorithm.

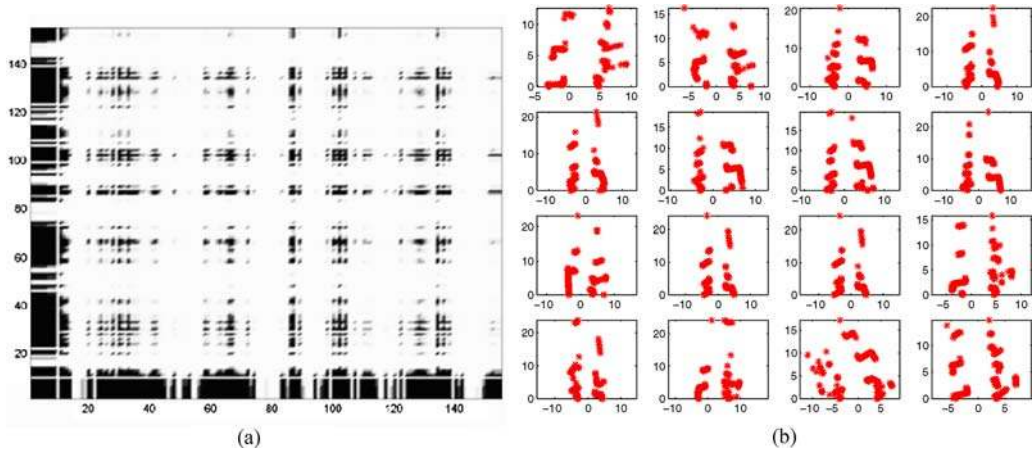

Figure 34. On the left is the first rank one approximation of the SSM shown in Fig. 32. On the right are 16 laser scans with the element values.

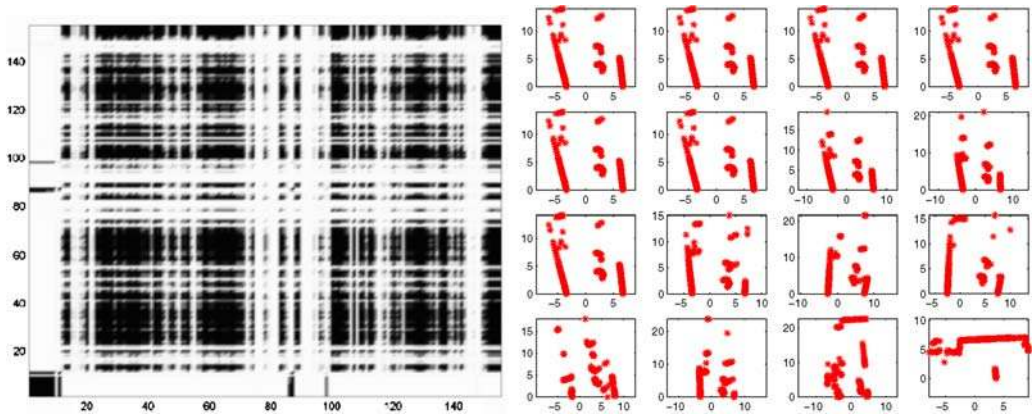

Figure 35. On the left is the $v_2 \lambda_2 v_2^T$ approximation of the spatial similarity matrix shown in Fig. 32. On the right are 16 laser scans with the highest scores in the matrix. These laser scans can be broadly defined as having parallel lines.

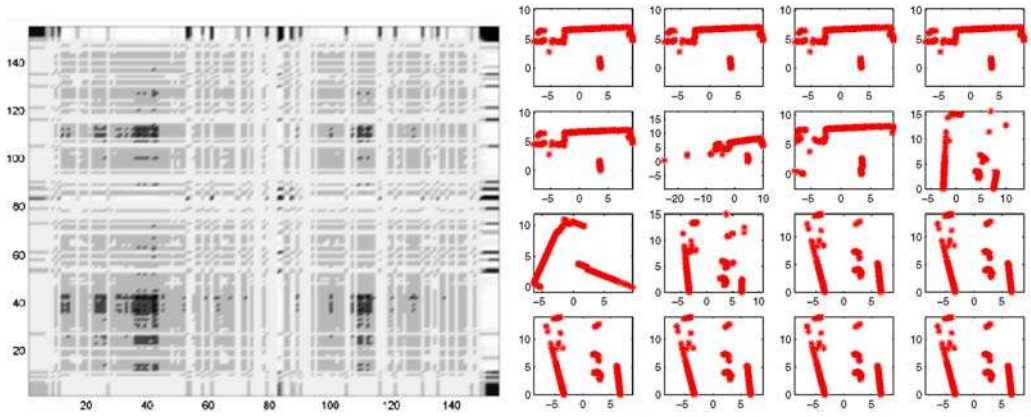

Figure 36. On the left is the third rank one approximation of the SSM shown in Fig. 32. On the right are 16, broadly T-shaped, laser scans with the highest scores in the matrix.

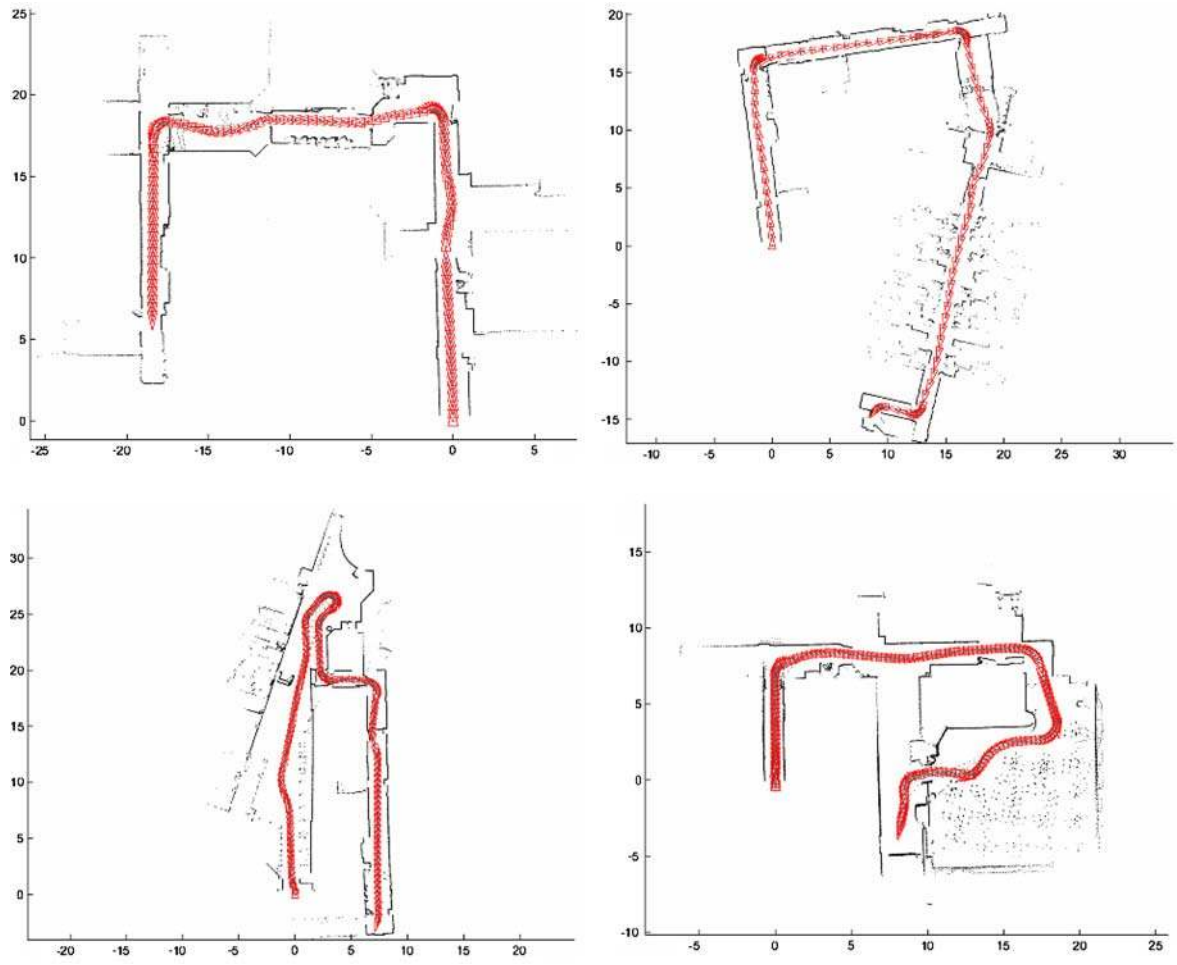

Figure 37. Local maps of different parts of the same building built by different robots. There is an overlap between each of the maps but it is not easy to discern from the laser patches alone.

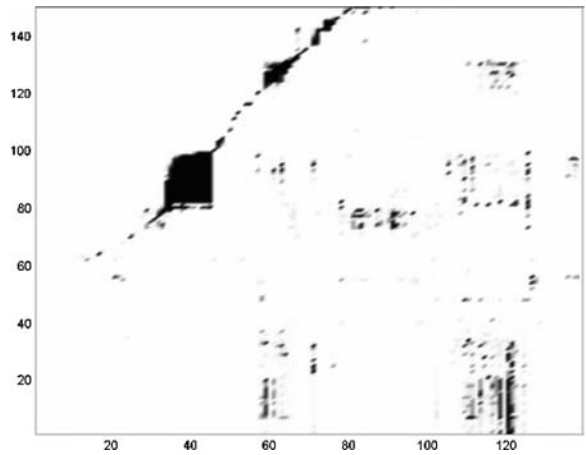

Figure 38. A similarity matrix constructed from comparison between image sequences collected by two robots. The dark line highlights the sequence of images that are similar to each other—indicating that there is an overlap in the two environments explored. Note there is no similarity along the diagonal as in all previous similar figures because axes i and j correspond to images captured by different robots.

correspondences. However, spatial configuration of three landmarks or simple geometric primitives are not very discriminative features.

A vision-based approach was used in Hajjdiab and Laganiere (2004) to combine maps built by a team of robots in the same worksite. Images described by color histograms are compared against each other to find the best matching image pairs. In the experimental setup, only images of planar surfaces are captured. Therefore, an inter-image homography can be calculated for selected image pairs. If the homography is supported by a sufficiently high number of corners, intersection is found and robot paths can be registered with respect to one another. However, the use of a single image pair for matching is prone to false positives (hence our motivation for using sequences). Importantly, none of the algorithms described above have any mechanism to determine that two local maps have *no* common overlap. They simply find the 'best' alignment possible between the two.

Figure 37 shows that the intersections between planar maps (ubiquitous in contemporary mobile robotics) may be far from obvious. However, application of the image-based techniques described in the paper can be of great help.

7.2. *Map Alignment*

A visual similarity matrix is constructed for each pair of robots. Each element $M_{A,B}(i, j)$ is the similarity between image i from robot A and image j from robot B. Every image from robot A is compared with all images from robot B. When there is an overlap between the local maps of the robots, there will be a connected sequence of elements with high similarity scores found within the visual similarity matrix. This is shown by the dark line in Fig. 38.

7.3. *Map Joining Results*

In our experiment, four robots start exploring from different locations of the same building. Each robot builds its

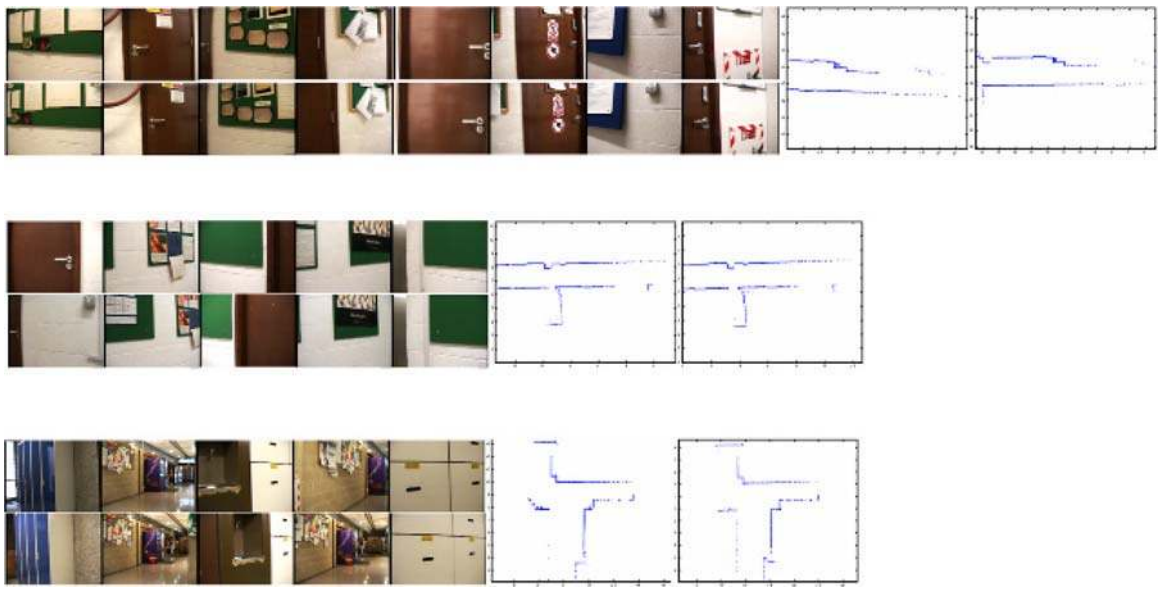

Figure 39. Matching subsequences between image streams gathered by different robots. The local regions in each map are shown to the right of each pairing—the first for the top sequence and the second (far right) for the second (lower) sequence.

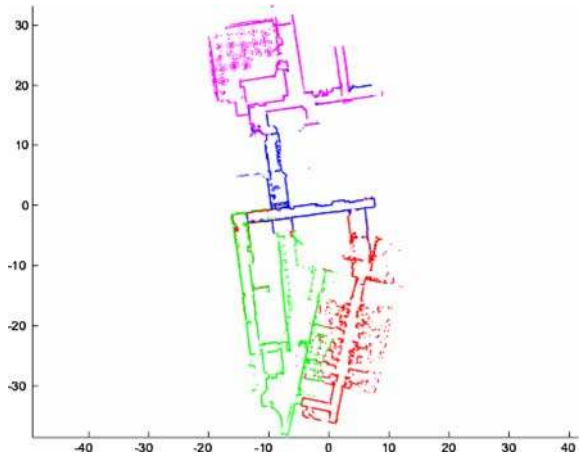

Figure 40. A fused map formed by detecting intersections between and aligning the four local maps shown in Fig. 37.

own local map as shown in Fig. 37. By comparing the image sequences collected by robot A and robot B, a 114 by 146 similarity matrix is constructed. The time complexity of comparing an image from one sequence against all the images in the other sequence is $O(log(p))$ where p is the number of visual words stored in the database. The average time to compare one image against the other sequence of 145 images is 0.269 seconds. The time complexity of the sequence extraction algorithm is $O(nm)$ where n and m are the lengths of the respective sequences. For this particular similarity matrix, sequence extraction took less than 0.3 seconds. All figures are for a Pentium 4, 2.40 GHz CPU. Figure 39 shows typical pairs of image subsequences found by the sequence extraction algorithm. Since each image and laser scan is time-stamped, we can extract the portion of the local maps that correspond to regions in which the images were captured. An estimated 2D transformation alignment between maps can then be calculated (using principal moment alignment for example) and used to bring the two geometric maps into close proximity. From here, scan matching produces accurate map-to-map transformations, allowing the four maps to be fused resulting in the map shown in Fig. 40. In the case for 3D laser scan matching, an initial transformation estimate can be obtained using the approach described in Section 6.1.1.

8. Conclusion

We have introduced a novel technique to robustly detect loop closure events. We do this by detecting sequences of similar scenes. Searching for matching scene image sequences implicitly exploits the temporal and spatial proximity of image acquisition locations and allows evidence to be integrated over the vehicle trajectory. Through suitable decomposition of a similarity matrix, we are able to remove the effects of ambiguous artifacts. This procedure is driven by consideration of the distribution of information throughout the matrix. This enhances the robustness of our technique in environments that can be markedly visually and geometrically confusing. Furthermore, we are able to test the statistical significance of sequences detected to further reduce the chances of committing Type I (false positive) errors. We have provided extensive experimental results over a range of realistic and challenging outdoor settings using visual images and in each case analyzed performance in the context of the SLAM problem. Where appropriate, we have used a combination of metric and visual information to not only detect but also execute loop closure. The image sets used are available on-line. We showed how our technique can be equally well applied to detecting loop closure without the benefit of a vision system when using laser images alone. Finally, we posed the multi-robot map-joining problem as a special case of loop-closure detection. We used our technique to find the work-space intersection of four SLAM maps further emphasizing the value of this appearance based techniques in the SLAM domain. The work presented addresses a central problem in mobile robotics and SLAM research. It offers a promising way to proceed that, importantly, is independent of the SLAM techniques it supports.

Acknowledgment

The authors would like to thank Josef Sivic, Frederik Schaffalitzky and Andrew Zisserman for their useful comments and Krystian Mikolajczyk for providing the software for extracting Harris-Affine descriptors. Thanks also to Dave Cole who was central in creating the 3D Laser map.

Note

1. Throughout this paper similarity matrices are displayed with artificially increased contrast so their fine structure survives the reproduction process.

References

1. Alter, O., Brown, P., and Botstein, D. 2000. Singular value decomposition for genome-wide expression data processing and modelling. In *Proceedings of National Academy of Science*, 97(18).
2. Altschul, S. and Erickson, B. 1985. Significance of nucleotide sequence alignments: A method for random sequence permutation that preserves Dinuclotide and Codon usage. *Molecular Biology and Evolution*, 2:526–538.
3. Bar-Shalom, Y. 1987. *Tracking and Data Association*. Academic Press Professional, Inc. San Diego, CA, USA.
4. Bosse, M., Newman, P., Leonard, J.J., and Teller, S. 2004. SLAM in large-scale cyclic environments using the atlas framework. *International Journal of Robotics Research*, 23:1113–1139.

5. Cole, D. and Newman, P. 2006. Using laser range data for 3D SLAM in outdoor environments. In *Proceedings of International Conference on Robotics and Automation*, Florida.
6. Dedeoglu, G. and Sukhatme, G. 2000. Landmark-based matching algorithm for cooperative mapping mapping by autonomous robots. In *Proceedings of the Fifth International Symposium on Distribution Autonomous Robotics Systems*.
7. Estrada, C., Neira, J., and Tardos, J. D. 2005. Heirarchical SLAM: Real-time accurate mapping or large environments. *IEEE Transactions on Robotics Research*, 8(4):588–597.
8. Davison, A.J. and Murray, D.W. 2002. Simultaneous localization and map-building using active vision. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 24(7):865–880.
9. Davison, A.J. 2003. Real-time simultaneous localisation and mapping with a single camera. In *Proceedings of International Conference on Computer Vision*.
10. Duda, R., Hart, P., and Stork, D. 2001. *Pattern Classification*. Wiley, New York: Chichester.
11. Eustice, R., Pizarro, O., and Singh, H. 2004. Visually augmented navigation in an unstructured environment using a delayed state history. In *Proceedings of International Conference on Robotics and Automation*.
12. Eustice, R., Singh, H., and Leonard, J. 2005. Exactly sparse delayed-state filters. In *Proceedings of International Conference on Robotics and Automation*.
13. Fenwick, J., Newman, P., and Leonard, J. 2002. Cooperative concurrent mapping and localization. In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, pp. 1810–1817.
14. Fox, D., Burgard, W., Kruppa, H., and Thrun, S. 2000. A probabilistic approach to collaborative multi-robot localization. *Autonomous Robots*, 8(3).
15. Gumbel, E.J. 1958. *Statistics of Extremes*. Columbia Univeristy Press: New York, NY.
16. Fitzgibbon, A. 2001. Robust registration of 2D and 3D point sets. In *Proceedings of the British Machine Vision Conference*.
17. Gutmann, J. and Konolige, K. 1999. Incremental mapping of large cyclic environment. In *Proceedings of the Conference on Intelligent Robots and Applications (CIRA)*, Monterey, CA.
18. Hajjdiab, H. and Laganiere, R. 2004. Vision-based multi-robot simultaneous localization and mapping. In *Canadian Conference on Computer and Robot Vision*, pp. 155–162.
19. Hartley, R. and Zisserman, A. 2000. *Multiple View Geometry in Computer Vision*. Cambridge University Press: Cambridge.
20. Ho, K. and Newman, P. 2005. Multiple map intersection detection using visual appearance. In *International Conference on Computational Intelligence, Robotics and Autonomous Systems*.
21. Ho, K. and Newman, P. 2005. Combining visual and spatial appearance for loop closure detection. In *Proceedings of European Conference on Mobile Robotics*.
22. Kŏsecká, J. and Yang, X. 2004. Global localization and relative pose estimation based on scale-invariant features. In *Proceedings of International Conference on Pattern Recognition*.
23. Kosecka, J., Li, F., and Yang, X. 2005. Global localization and relative positioning based on scale-invariant keypoints. *Robotics and Autonomous Systems*, 52(1).
24. Konolige, K., Fox, D., Limketkai, B., Ko, J., and Stewart, B. 2003. *Map Merging for Distributed Robot Navigation Proceedings of International Conference on Intelligent Robots and Systems*.
25. Konolige, K. 2004. Large-scale map-making. In *Proceedings of the National Conference on AI (AAAI)*, San Jose, CA.
26. Leonard, J.J. and Newman, P. 2003. Consistent, convergent, and constant-time SLAM. In *Proceedings of International Joint Conference on Artificial Intelligence*.
27. Levin, A. and Szeliski, R. 2004. Visual odometry and map correlation. *IEEE Conference on Computer Vision and Pattern Recognition*.
28. Lowe, D.G. 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.
29. Lu, F. and Milios, E. 1997. Robot pose estimation in unknown environments by matching 2D range scans. *Journal of Intelligent and Robotic Systems*, 18:249–275.
30. Matas, J., Chum, O., Urban, M., and Pajdla, T. 2002. Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of British Machine Vision Conference*.
31. Mikolajczyk, C. and Schmid, C. 2004. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86.
32. Neira, J. and Tardōs, J. D. 2001. Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions of Robotics and Automation*, 17(6):890–897.
33. Newman, P. and Ho, K. 2005. SLAM—Loop closing with visually salient features. In *Proceedings of International Conference on Robotics and Automation*.
34. Newman, P., Cole, D. and Ho, K. 2006. Outdoor SLAM using visual appearance and laser ranging. In *Proceedings of International Conference on Robotics and Automation*, Florida.
35. Nister, D. 2004. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):756–770.
36. Ranganathan, A., Menegatti, E., and Dellaert, F. 2006. *IEEE Transactions on Robotics*, 22(1):92–107.
37. Royer, E., Lhuiller, M., Dhome, M., and Chateau, T. 2004. Towards an alternative GPS sensor in dense urban environment from visual memory. In *Proceedings of British Machine Vision Conference*.
38. Se, S., Lowe, D.G., and Little, J. 2002. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *International Journal of Robotics Research*, 21(8):735–758.
39. Se, S., Lowe, D.G., and Little, J. 2005. Vision based global localisation and mapping for mobile robots. *IEEE Transactions on Robotics*, 21(3):364–375.
40. Silpa-Anan, C. and Hartley, R. 2005. Visual localization and loopback detection with a high resolution omnidirectional camera. *Workshop on Omnidirectional Vision*.
41. Sparck Jones, K. 1972. Exhaustivity and specificity. *Journal of Documentation*, 28(1):11–21.
42. Smith, R., Self, M., and Cheeseman, P. 1987. A stochastic map for uncertain spatial relationships. In *4th International Symposium on Robotics Research*.
43. Smith, T.F. and Waterman, M.S. 1981. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197.
44. Sivic, J. and Zisserman, A. 2003. Visual Google: A text retrieval approach to object matching in videos. In *Proceedings of the International Conference on Computer Vision*.
45. Thrun, S. 2001. A probabilistic online mapping algorithm for teams of mobile robots. *International Journal of Robotics Research*, 20(5):335–363.
46. Thrun, S. and Liu, Y. 2003. Multi-robot SLAM with sparse extended information filers. In *Proceedings of the 11th International Symposium of Robotics Research*.
47. Torralba, A., Murphy, K., Freeman, W., and Rubin, M. 2003. Context-based vision system for place and object recognition. In *Proceedings of International Conference on Computer Vision*.
48. Wang, J., Cipolla, R., and Zha, H. 2005. Vision-based global localization using a visual vocabulary. In *Proceedings of International Conference on Robotics and Automation*.
49. Wolf, J., Burgard, W., and Burkhardt, H. 2005. Robust vision-based localization by combining an image-retrieval system with monte carlo localization. *IEEE Transactions on Robotics*, 21(2):208–216.