Available online at www.sciencedirect.com
ScienceDirect
Robotics and Autonomous Systems 54 (2006) 740–749
Robotics and Autonomous Systems
www.elsevier.com/locate/robot

Loop closure detection in SLAM by combining visual and spatial appearance

Kin Leong Ho, Paul Newman*

Oxford Robotics Research Group, Parks Road, Oxford OX1 3PJ, United Kingdom

Received 19 December 2005; received in revised form 9 March 2006; accepted 6 April 2006
Available online 27 July 2006

Abstract

In this paper we describe a system for use on a mobile robot that detects potential loop closures using both visual and spatial appearance of local scenes. Loop closing is the act of correctly asserting that a vehicle has returned to a previously visited location. Current approaches rely heavily on vehicle pose estimates to prompt loop closure. Paradoxically, these approaches are least reliable when the need for accurate loop closure detection is the greatest. Our underlying approach relies instead upon matching distinctive 'signatures' of individual local scenes to prompt loop closure. A key advantage of this method is that it is entirely independent of the navigation and or mapping process and so is entirely unaffected by gross errors in pose estimation. Another advantage, which is explored in this paper, is the possibility to enhance robustness of loop closure detection by incorporating heterogeneous sensory observations. We show how a description of local spatial appearance (using laser rangefinder data) can be combined with visual descriptions to form multi-sensory signatures of local scenes which enhance loop-closure detection.

Keywords: SLAM; Loop closing; Visual appearance; Spatial appearance

1. Introduction and motivation

SLAM (simultaneous localization and mapping) is a core information engineering problem in mobile robotics and has received much attention in past years especially regarding its estimation theoretic aspects [1,17,25]. Good progress has been made but SLAM is still far from being an established and reliable technology. A big problem is a lack of robustness. This is markedly manifested during what has become known as loop closing. It is common practice to use estimates produced by a SLAM algorithm itself to detect loop closure. The naive approach adopted in early SLAM work simply performs a nearest neighbor statistical gate on the likelihood of the current measurements given map and pose estimates. If the pose estimate is in gross error (as is often the case following a transit around a long loop), while in reality the vehicle is in an already mapped area, the likelihood of measurements being explained by the pose and map estimate is vanishingly small. The consequence of this is that loop closure is not detected. Previously visited areas are re-mapped, but in the wrong global location, error accumulates without bound.

Fig. 1 shows an obvious case of poor loop closing. Liberalization and perception errors have lead to a gross error in vehicle location estimate — so bad that the true location lies outside the three-sigma bound on vehicle uncertainty. The problem here is that the likelihood used is not independent of vehicle pose. More sophisticated techniques offer some degree of robustness against global vehicle error. For example, by looking at the relationship between features in the local area [18] or continually trying to relocate in a bounded set of sub-maps [1] that are expected to have some non-empty intersection with the true local area. However these methods still struggle when the estimated vehicle position is in gross error.

In our approach, an environment explored by the robot is broken up into a sequence of individual local scenes. Temporal relationships between the local scenes are established from sequence order. Each local scene is described by a 'distinctive signature'. Adjacent local scenes have partial overlap of the

* Corresponding author.
E-mail addresses: klh@robots.ox.ac.uk (K.L. Ho), pnewman@robots.ox.ac.uk (P. Newman).
URLs: http://www.robots.ox.ac.uk/~klh (K.L. Ho), http://www.robots.ox.ac.uk/~pnewman (P. Newman).

0921-8890/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.robot.2006.04.016

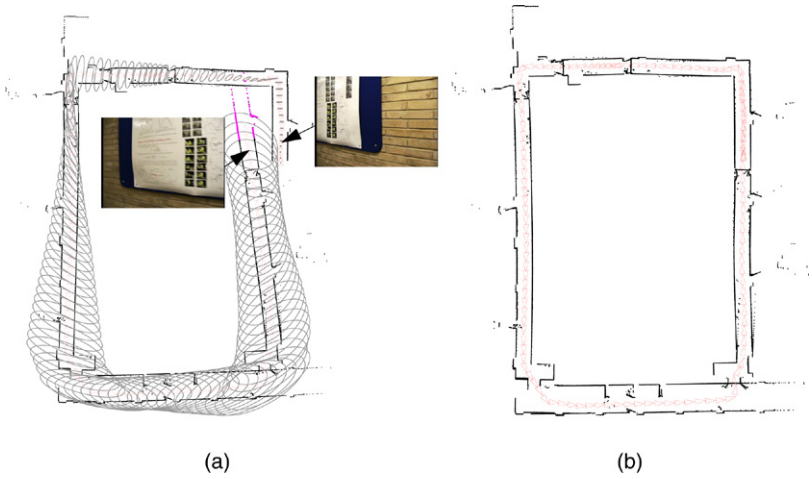

Fig. 1. (a) Shows a snapshot of our SLAM algorithm just before loop closing takes place. The vehicle poses stored in the state vector are shown as red triangles. The performance of the SLAM algorithm is just as would be expected. Global uncertainty (gray ellipses) increases as the length of the excursion from the start location increases. A poor scan match at the bottom right introduced a small angular error which leads to a gross error in pose estimate when in reality the vehicle has returned to near its starting locations (top right). The inset images are the two camera views used in the loop-closing process. The left hand image is the query image and the right hand one the retrieved, matching image. The poses that correspond closest in time to the two images are indicated with arrows. (b) Shows the final map after applying the loop closing constraint. As expected the marginal covariances on each vehicle pose decrease and a crisp map results — as would be the case for any choice of SLAM algorithm. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

environment. When comparing a newly captured signature against a database of signatures, loop closure will be prompted if a match is found. By relying on vehicle pose estimates, relative or global map feature estimates, current approaches are susceptible to unbounded accumulation of perception and liberalization errors. Errors in our approach, on the other hand, are bounded to errors from relative comparison of each individual local scene.

The focus of this paper is to combine visual and spatial appearances of a local scene into a distinctive signature that can be used to prompt loop closure. Our motivation is two-fold. Firstly multi-modal sensing naturally leads to richer and more discriminative descriptions. Our second motivation is to address shortcomings in our solely visual-appearance system. The underlying assumption of our approach is if two local scenes have similar signatures, they are likely to correspond to the same location. This assumption is not always valid. We found that many urban environments possessed repetitious visual features which produced false positives. A query image, taken from a current location, would be matched to the contents of one or more previous images stored in a large database. This would eventually lead to the erroneous declaration of loop-closure events in (to a human) ludicrous situations.

For example, fire escape notices, multi-paned windows and occasional wall patterns were repetitious visual events in our test environments. However they did not occur in similar spatial settings. An example is shown in Fig. 2. This paper shows that by describing the local spatial appearance of the image capture locale, false visual matches can be successfully discriminated against. The operation at the heart of the spatial discrimination component is the comparison of two 2D laser images (not necessarily a single scan and more likely to be a scan-patch in the terminology of [9]) of the locales of two camera images. One picture-laser pair will be a query-pair — encapsulating the spatial and visual appearance of the robot's current location. This pair will typically be compared to one of many possible candidate pairs in a database — a set of picture-laser pairs built over the vehicle's past trajectory.

2. Derivation of visual signature

We begin by describing how to derive a visual signature of a local scene. Periodically, an image of the surrounding environment will be captured. This image is unique to the specific pose of the robot with respect to the environment. The goal is to reduce this image into a set of descriptors. Every image set of descriptors is compared against each other to determine similarity between local scenes. Many authors have successfully used visual landmarks in SLAM, for example [20,4]. In this paper we also use a camera to extract visual landmarks. However we do not use them as geometric features within the SLAM algorithm. Instead, we focus on the photometric information contained with the descriptors and the wide-baseline stability of the descriptors for the purpose of scene comparison

2.1. Detection of image features

Previously in [19], a system was developed that was able to close loops with visually salient features. Fig. 1 shows a typical result in which two, automatically detected visually salient images were used to close a loop. Two detection algorithms were employed in that work, namely the scale saliency algorithm [8] and the "maximally stable extremal regions" (MSER) algorithm [14]. The scale saliency algorithm measures saliency of regions within images as a function of local image complexity weighted by a measure of self-similarity across scale space. The MSER algorithm finds "maximally stable extremal regions" which offers significant invariance under affine transformations. The reason for the

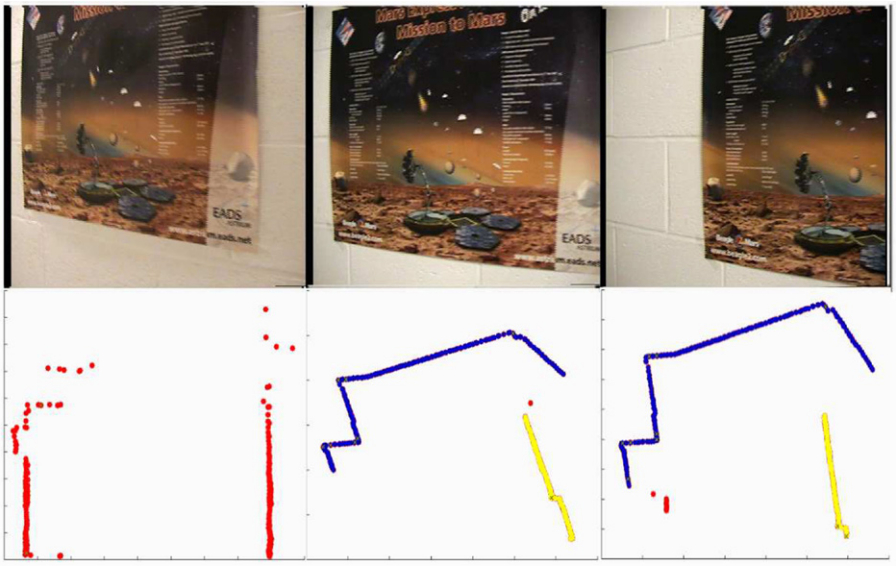

Fig. 2. The top row shows three visually similar images stored in an image database. The query image is shown in the center. Using photometric information alone, it is extremely difficult to discern whether the right or left image is the correct match. A false loop closing event might be signalled (albeit a visually correct match) with the image on the left. However, taking into account of spatial information will discredit the left image match while confirming an alternative visual match shown on the right hand side.

wide-baseline stability of the technique lies in the fact that connectivity (which is essentially what is detected) is preserved under reasonable affine transformations (<70° in the plane). In this work, we adopt a different detection algorithm, namely the Harris-affine detection algorithm [15]. The primary reason for selecting the Harris-affine algorithm is because it produces a richer and more diverse range of descriptors than other detection algorithms given the same image database. A wider range of descriptors helps in differentiating image similarity. Also, it was demonstrated in [16] that Harris-affine regions enjoy comparable wide-baseline performance as MSERs.

2.2. Description of image features

Having found image regions, we encode them in a way that is both compact, to allow swift comparisons with other regions, and rich enough to allow these comparisons to be highly discriminatory. The descriptor chosen is the SIFT descriptor [13] which has become immensely popular in computer vision applications [24] and used with good effect in SLAM in [12]. The SIFT description algorithm transform each image region into a $4X4$ array of histograms with 8 orientation bins in each. Consequently, each SIFT descriptor is a 128 dimension feature vector. The high dimensionality of the SIFT descriptor plays a critical role in matching accuracy.

2.3. Assignment of weights to descriptor

The vector space model [24] which has been successfully used in text retrieval is employed in this work. Each image can be considered as a document consisting of visual words. In this case, each SIFT descriptor is a visual word. Construction of a visual vocabulary is achieved by clustering similar SIFT descriptors (in terms of Euclidean distance) into visual words that can be used for inverted file indexing. An agglomerative clustering algorithm is used. Weights, W_i, are assigned to each SIFT descriptor, D_i, (word) according to the frequency of the occurrence of the visual word in the image database. This is based on the inverse document frequency [22] formulation: $W_i = \log_{10}(N/n_f)$ where N is the number of images stored in the image database and n_f is the number of images containing the visual word, D_i. The collection of images is represented by an inverted index for efficient retrieval. To further enhance the retrieval speed, we employ a k-d tree to search for the visual words.

2.4. Similarity scoring function

To measure the similarity between two images, I_u and I_v, we employ the cosine similarity method. Since each image is represented as a vector of words with different weights, we can measure their cosine similarity by the inner product of the two image vectors as shown in Eq. (1). The scoring for a match of a term is based on the weights from the inverse document frequency. If the images have different numbers of visual words, imaginary visual words with zero weights are inserted into the smaller image vector so that the sizes of both image vectors are equal

$$S(I_u, I_v) = \frac{\sum_{i=1}^{n} u_i . v_i}{\left(\sum_{i=1}^{n} u_i^2\right)^{1/2} . \left(\sum_{i=1}^{n} v_i^2\right)^{1/2}} \quad (1)$$

where $I_u = [u_1 \cdots u_n]$, $I_v = [v_1 \cdots v_n]$ and u_i and v_i are visual words from the respective images.

The left-hand column of Fig. 3 illustrates an anomaly where the matched visual scene is visually similar to the query but the robot is actually at a different location. This is an example where the visual image matching system is working as hoped

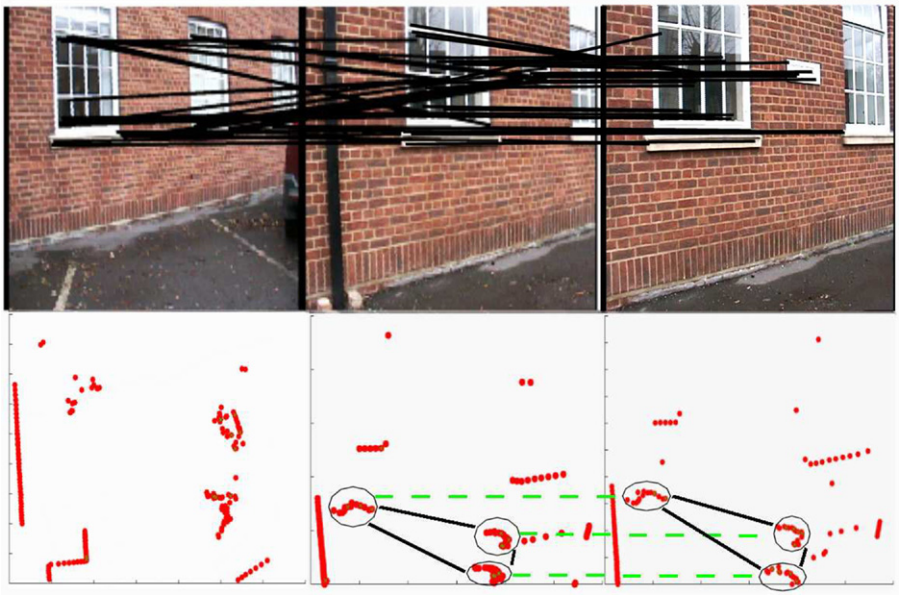

Fig. 3. Results in an outdoor environment. As shown above, a false positive loop closure is signalled w.r.t. the left-hand image when only visual information is taken into consideration. This is discounted when spatial descriptors are used in addition. The query image and patch is in the middle and a correct match (spatially and visually) is shown on the right. There are actually four visual correspondences (but mutually occluding when drawn) between the central query image and each of the candidate images.

yet it incorrectly suggests a loop-closure event. It is true that some of the false image feature correspondences can be removed through the enforcement of epipolar constraints. However, matching of image features on repetitive artifacts will still occur. On the other hand, the geometry of the local environments are truly different and can help to differentiate the two local scenes. The spatial appearance of the immediate environment must be taken into consideration. Accordingly, the rest of the paper is devoted to describing one approach to this task.

3. Derivation of spatial signature

A laser scan can be considered to be top-view image of the geometric structure of the environment. Though most efforts have concentrated on extracting shape descriptors of 2D objects in images [26,2,10] have applied their shape similarity system to the problem of robot localization and mapping in recognition of the similarity in these two problems. We begin by describing how a complete laser patch is passed through a pipeline of processes resulting in a set of descriptors that encode the shape and spatial saliency of local regions. We then discuss how these descriptors can be compared with one another before bringing the descriptor generation and comparison functionality together to build a discriminative system.

3.1. Initial segmentation

The laser scan is divided into smaller but sizeable "segments". These segments are formed using a standard nearest neighbor clustering algorithm [23]. A new segment is formed whenever there is a significant break along a contour.[1] These breaks are due to both occlusions and the true structure of

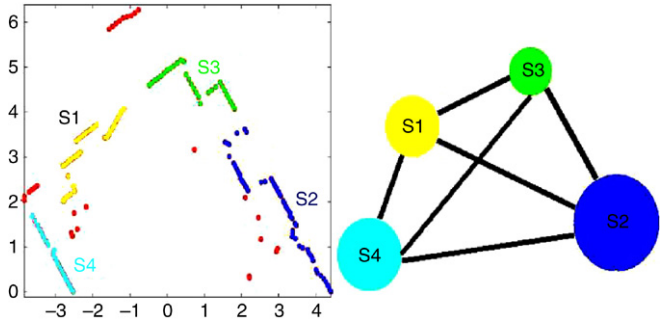

Fig. 4. This shows a typical geometry patch after segmentation and a graph depiction of the way we encapsulate the information contained. Each node is a segment and contains the CAF function, its entropy measure and a list of critical points. The edges represent a known spatial relationship between segments.

the environment. Fig. 4 shows a typical segmentation; the laser patch on the left is broken up into four segments. We represent each segment as a node on the graph on the right. The spatial relationships *between* the segments are encoded into edges that connect the nodes. The generation of these edge descriptors will be discussed after considering how the segments themselves are described.

3.2. Segment descriptors

The segmentation completed, we now desire to describe each segment. The generated descriptors will be the values of each node in Fig. 4. Each node is described using a cumulative angular function, its entropy value and a set of "critical points" along the segment's boundary. The motivation behind and method employed in these steps are as follows:

3.2.1. The cumulative angular function

Each segment is described by the "cumulative angular" or "turning" function [5,11] as illustrated by Fig. 5. The turning

[1] Note we do not require a convex scan patch.

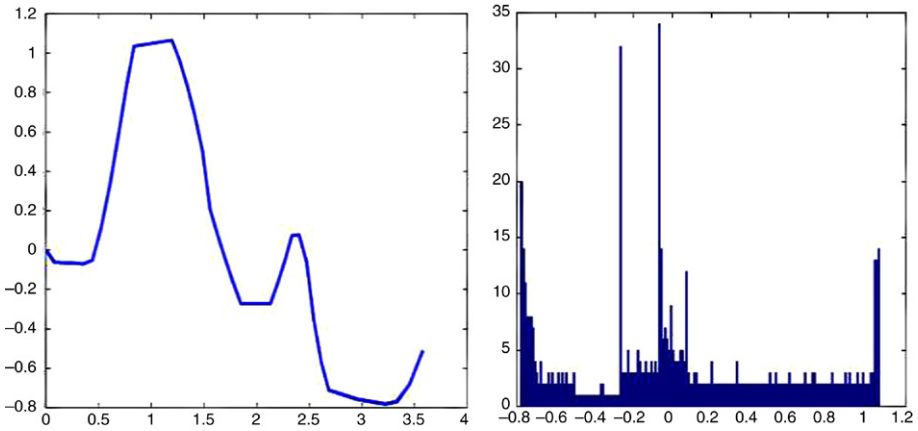

Fig. 5. The cumulative angular function is transformed into a histogram of angular values. Each bin contains the number of points along the cumulative angular function that have angular values that fall within the bin value. Using this histogram of bin values, the entropy of the cumulative angular function can be calculated.

function is a plot of the cumulative change in turning angle ϕ versus the arc-length S of the segment. To illustrate, the turning function maps straight lines $ax + by + c = 0$ to $\phi = 0$, circles to $\phi = \alpha S$ and squares to a "staircase" function in ϕ. A key characteristic of the cumulative angular function is that it is rotational and translational invariant.

3.2.2. Entropy

We wish to measure the *complexity* of a segment so that we can prefer matches between "complex" shapes to matches with "simple" shapes. This is motivated by reasoning that a positive match between two complex shapes is more likely to be a true positive than a match between one simple, one complex and two simple shapes. A natural way to encode complexity is via entropy. In this case we may write an expression for entropy as

$$\mathcal{S}_{\mathcal{D}} = -\int_{i \in D} P_{D_i} \log_2 P_{D_i} d_i \tag{2}$$

where P_{D_i} is the probability of descriptor D_i takes on values in D the set of all descriptor values. The descriptor values in this case are the angular values along the cumulative angular function.

The integral is calculated from a histogram of the cumulative angle function. Each histogram bin contains the number of points along the cumulative angular function that have angular values that fall within the bin value. The entropy follows from Eq. (2). A distinctive segment will have a cumulative angular function with multiple peaks and troughs while a simple segment will have a relatively flat cumulative angular function. In deriving shape descriptors, emphasis (via thresholding) is placed on encoding segments with high entropy as they are more distinctive.

3.3. Inter-segment descriptors

Given shapes of individual segments have been encoded, we are now interested in describing the spatial configuration of segments within a laser patch, in a manner that is rotationally and translational invariant.

3.3.1. Critical points

We wish to encode the spatial configuration between segments, which will form the inter-segment descriptors of the laser scan (the edges of the graph in Fig. 4). We do this by first extracting points of high curvature along the segments. We call these "critical points" after [27]. Critical points are sharp changes in the cumulative angle function and they are marked as crosses in the laser patches shown in Fig. 6. Repeatability of extraction of these critical points is an important consideration. The thresholding on CAF (cumulative angle function) entropy selects in favor of segments possessing strong critical points — regions of high curvature likely to be visible over a range of vantage points.

3.3.2. Segment configurations

In contrast to [27] where distances between all "anchor points" are used to match laser scans, ownership of critical points by individual segments is considered when matching of segments between laser patches. The distance and relative orientation between critical points form the links (the lines joining the two segments shown in Fig. 6) that lock two segments in a fixed configuration. To determine the relative orientation between the critical points, we first have to determine the orientation of the segment. This is simply done using the largest eigenvector of the segment.

4. Descriptor comparison

4.1. Segment descriptor comparison

We now describe how two segment descriptors generated according to Section 3.2 can be compared to one another. Each segment (a node in the graph of Fig. 4) contains the CAF function, its entropy measure and a list of critical points. Considering two such nodes we use three disparity measures based on their properties.

4.1.1. Angular function disparity

By representing a 2-D patch segment as a 1-D shape descriptor, finding the best fit between two segments reduces

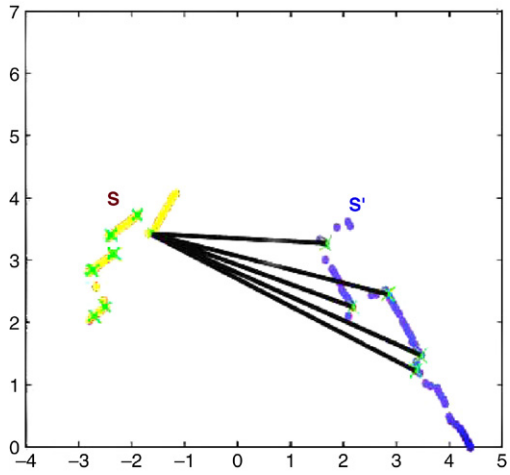

Fig. 6. The way in which the relationship (in this case an SE2 transformation) between segments is encoded. The two segments S and S' contain nc and nc' critical points respectively. For each critical point in S we form a"bundle" of links to all nc' critical points in S'. In all there will be nc bundles and $nc \times nc'$ links in total but only one bundle is shown here. Each bundle (so long as it contains more than one link) defines a rigid transformation between a critical point in S and the entire segment S'. We define each edge in the graph of figure 2 to be the set of all bundles from S to S'. This is by intent a redundant way to store the relationship between the two segments.

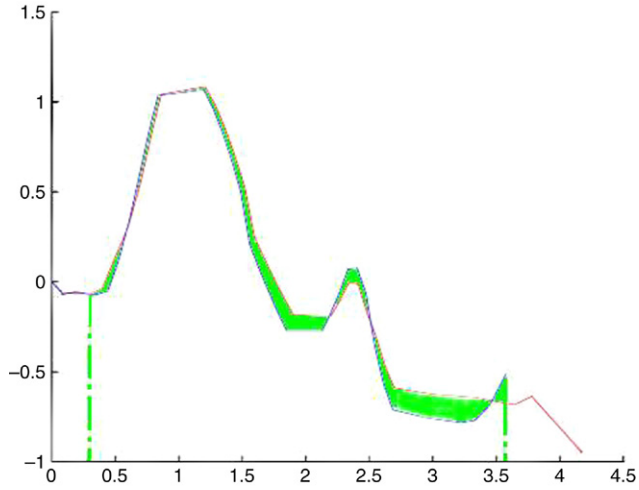

Fig. 7. Fig. 7 shows the disparity between two CAFs of two segments, S and S'. CAF is the one-dimensional representation of 2-D segments, which encodes the structure of the points within the segment by the change in tangential angles between consecutive points. The difference between the two CAFs is the area between the two curves. Segments that are similar to each other will have similar angular functions and correspondingly, the disparity between the two angular functions will be small.

from a 3-D search space $[x, y, \theta]$ problem into a 2-D search space [3]. This is a search problem in the position–rotation space (β, γ) since scale is fixed in our application. The query curve is translated vertically and shifted horizontally to find the minimum error between the query curve and the pattern curve, see Fig. 7. This approach is similar to the method employed by [3], except that their search problem is in scale–position–rotation space. The difference, $e(\beta, \gamma)$, between CAFs is calculated as

$$e(\beta, \gamma) = \int_0^l (T_1(s) - T_2(s + \beta) + \gamma)^2 \mathrm{d}s \tag{3}$$

where the two cumulative angular functions are denoted by T_1 and T_2, the position–rotation search space is parameterized by (β, γ) and s parameterizes arc-length around the segment. A scalar similarity measure η_1 lying in [0, 1] is then calculated as $\eta_1 = \frac{1}{1+e}$.

4.1.2. Match length disparity

A second scalar η_2 is calculated as the matched length to total length ratio: $\eta_2 = \frac{l(m)}{l(T)}$ where $l(m)$ is the length of the matched segment portion and $l(T)$ is the total length of the query segment. In Fig. 7, the matched length is the portion of the abscissa where there is overlap between the two cumulative angular functions and total length is the length of the query cumulative angular function. The larger the portion of the segment that is matched (based on η_1), the more similar the segments are.

4.1.3. Entropy disparity

We use relative entropy to measure the similarity between segments. The relative entropy, or the Kullback–Leibler distance, is given by:

$$K(f \| f') = \sum_{i=1}^{m} f_i \times \ln \frac{f_i}{f'_i} \tag{4}$$

where m is the number of bins and f and f' are the probability distributions approximated by the angle histograms an example of which is illustrated in Fig. 5. The smaller the relative entropy, the more similar the distribution of the two histograms. When both distributions are equivalent $K(f \| f') = 0$. The relative entropy is normalized to lie within [0, 1] to produce a third scalar η_3.

We only calculate η_3 (and hence compare segments) when both have large S_D. The concept is that it is less likely for segments with high entropy to mismatch compared to segments with low entropy. Consider a laser scan of a long, straight corridor represented by two straight line segments; these straight line segments will match easily with straight line segments from any other laser scans taken at other portions of the corridor. The above three similarity scalars are stacked in vector $\eta_{S,S'} = [\eta_1, \eta_2, \eta_3]^{\mathrm{T}}$. That describes the degree of similarity between S and S'. If S and S' are identical segments $\eta_{S,S'}$ will be $[1, 1, 1]^{\mathrm{T}}$.

4.2. Edge comparison

As well as comparing the shape characteristics of segments, the matching technique described in the next section will ask if the relationship *between* segments within a patch are similar to those in a test case. As suggested in [27], we determine the similarity between the segment–segment links by matching arrays of distances and relative orientations of the segment–segment edges. In Fig. 8, the segment–segment relationships for two laser scans are shown. Due to occlusions, a minority of the critical points found in one laser scan are not

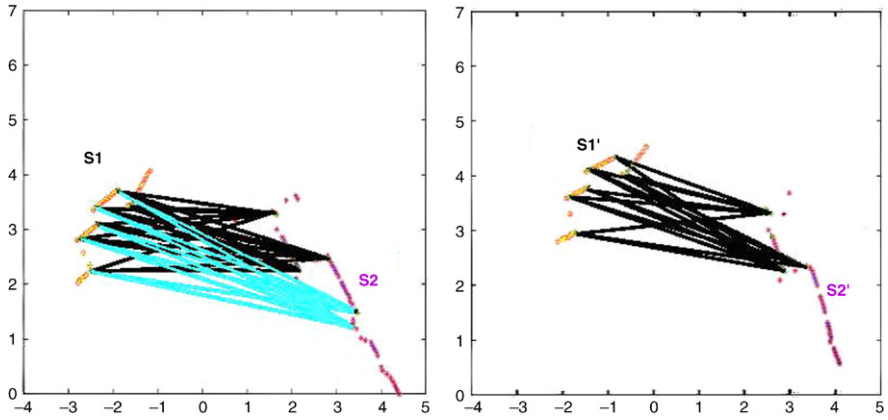

Fig. 8. A method of comparing inter-segment relationships (edges). In our determination of similarity between edges, the bundles of links that comprise the edges are compared against each other using distance and relative angle criteria. The dark links represent those that have been successfully matched with links of the other edge. The links marked as light toned lines represent links that have not been matched with links of the other edge. This can be due to occlusions or the segmentation process.

seen in the other. The links that are successfully matched are highlighted in a darker tone. The quality of the match between the edges is determined by the ratio of matched links versus the total number of links: $q_m = \frac{n_m}{n_r}$ where n_m is the number of matched links and n_r is the number of links between segments in query scan.

5. Matching of spatial descriptors

Our shape similarity metric comprises of two parts: the shape similarity between two segments and the spatial similarity between segments. The quality of match between segment S_i from the query scan and segment S'_j from the reference scan is defined by $Q_{ij} = \lambda \eta_{S_i,S'_j} + (1-\lambda) \times q_m$, where $Q_{ij} \in [0, 1]$ and the parameter $\lambda \in [0, 1]$, determines the relative importance attached to the matching of the shape of segments and the links between segments. It was determined experimentally that the value of 0.3 for λ produced the best matching performance for the particular environment in our experiment.

Given the set of matching score Q_{ij} between all pairs of segments $S_{i\ldots n}$ on the query laser scan and $S'_{j\ldots m}$ on the reference laser scan, we want to maximize the total matching score of matching subject to the constraint that the matching must be one to one. We solve this by using the Hungarian method [21] which finds the optimal combination of segments matching that gives the highest matching score possible between the query and reference laser scans.

6. Experimental results

To examine and demonstrate the effectiveness of our approach, we tested our algorithm in an outdoor environment. The ATRV-Jnr mobile robot was driven around a car park in front of a building. The vehicle camera kept a constant orientation in vehicle coordinates — looking forward and slightly to the left. Every two seconds an image was grabbed and written to disk. The vehicle was equipped with a standard SICK laser, the output of which was also logged along with the odometry from the wheel encoders. Each image was time stamped, processed and finally entered into a database as a collection of feature descriptors. Using the image's time stamp, the corresponding laser scan is retrieved, processed and entered alongside the visual information as a collection of spatial descriptors. Here, a database[2] of 155 images and laser scans was collected — see Figs. 9 and 10 — where the ground is relatively flat.

6.1. Retrieval system results

Firstly, we demonstrate the effectiveness of the image retrieval system through some examples. In Fig. 9, the top row represents the query images. Down each column are the corresponding images most similar to the query images in descending order of similarity. Notice that the image on the second row, first column, is actually a false positive. This is what we have expected. Even though the image is visually similar to the query image, the image was captured from a different location from the query image. This is an instance where the assumption "similar appearance, same location" does not hold. On the positive side, the retrieval system demonstrates robustness to dynamic objects. Looking into the second column and the last column, it is observed that the retrieval system is able to select images from nearby locations despite the presences of dynamic objects such as vehicles and humans.

Next, we demonstrate the effectiveness of the laser scan retrieval system. Similarly, the top row in Fig. 10 represents the query laser scans. Down each column are the corresponding laser scans deemed most similar to the query laser scans in descending order of similarity. The observation is the complexity of the laser patches makes it difficult to describe laser patches adequately with only geometric primitives such as lines and corners. The results from column two demonstrate the rotational invariance property of our description and matching technique.

[2] The database of images and laser scans collected from the urban environment can be viewed at http://www.robots.ox.ac.uk/~klh/AclandImageDB.htm and http://www.robots.ox.ac.uk/~klh/AclandLSDB.htm respectively.

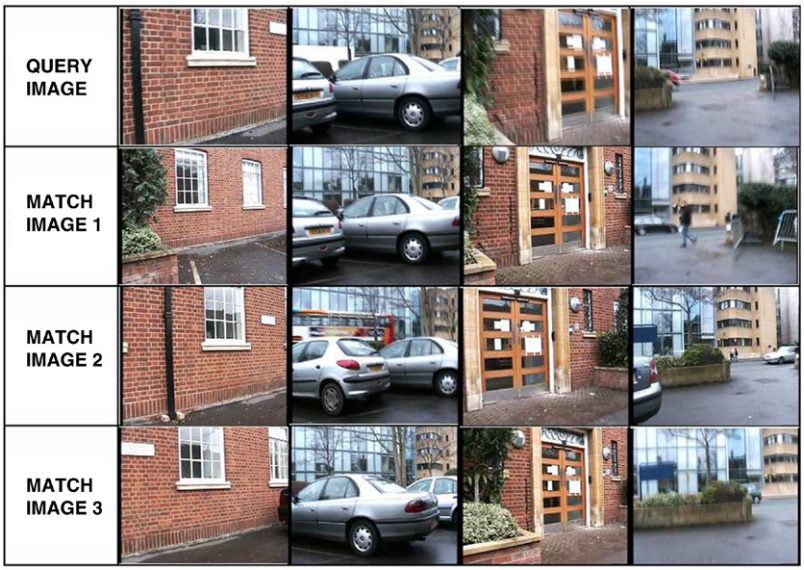

Fig. 9. Experimental results from our image query system. The top row contains the query images. Down the column are the corresponding matches in descending order of similarity. Notice the robustness of our image matching system against dynamic objects such as human and moving vehicles. However, the image in row 2, column 1 is an example of how visual match alone will produce wrong loop detection in environments where there are repetitive visual artifacts.

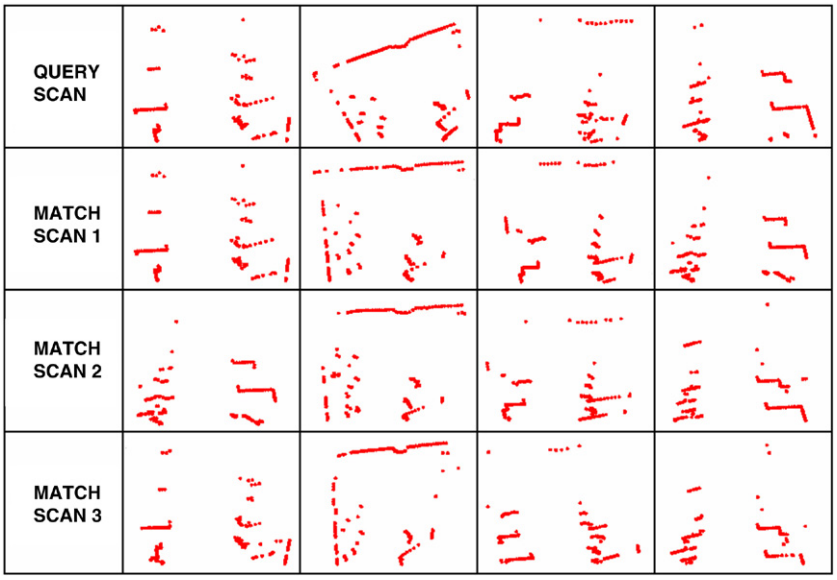

Fig. 10. Experimental results from our laser scan shape-based query system. The top row contains the query laser scans. Down the column are the corresponding matches in descending order of similarity.

6.2. *Similarity matrix*

We encode the similarity between all possible pairings of scenes in a "similarity matrix". Each element $M_{i,j}$ of the similarity matrix is the similarity score between signature i and signature j from the data sequence. The main diagonal consist of a bright red line. It is the result of matching every local scene against itself and consequently a perfect match. When there is a loop closure, there will be a connected sequence of off-diagonal elements with high similarity scores found within the similarity matrix. This is shown by the off-diagonal bright lines.

The data in Fig. 11 was taken over two loops around the outside of a large building. A visual similarity matrix is constructed. It is composed of similarity scores calculated from matching each image in the database against every image in the database. A spatial similarity matrix is also constructed by matching each laser scan in the database against every laser scan with the technique described in Section 5. To construct the combined visual and spatial similarity matrix, we simply add the normalized similarity scores from the visual and spatial similarity matrix in a similar fashion as [7]. Cells with high similarity scores are shaded in a bright tone while cells with low similarity scores are shaded in a dark tone. From these three similarity matrices as shown in Fig. 11, we can compare the loop closure detection performance of using the image

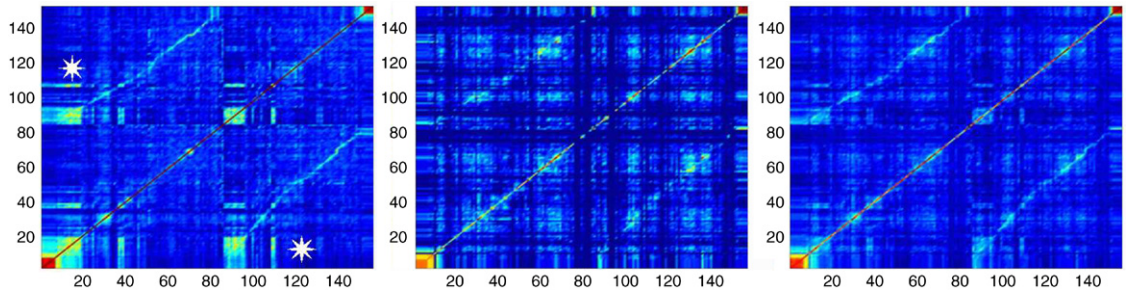

Fig. 11. Similarity matrices. The left matrix is the image similarity matrix. The middle matrix is the laser scan similarity matrix. The right matrix is the combined image–laser scan pair similarity matrix. Improvement in loop closure detection can be observed by the improved definition of the off-diagonals of the combined image and laser scan pair similarity matrix over the off-diagonals of the other similarity matrices.

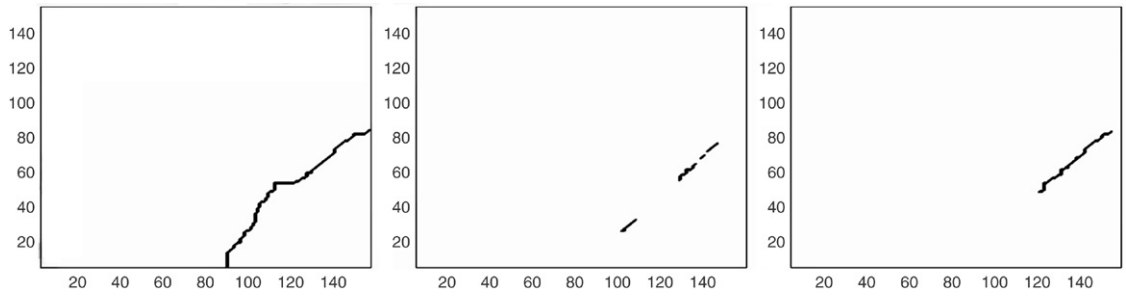

Fig. 12. From left to right: Sequence detection in visual similarity matrix, spatial similarity matrix and combined similarity matrix. The dark lines represents sequences of matching pairs of signatures.

matching system, the laser scan matching system and the image–laser scan pair matching system.

As soon as the second loop begins (around the time image 85 was taken) we expect to see off-diagonals appearing indicating matches with earlier data. In the visual similarity matrix, (L.H.S. of Fig. 11), the off-diagonals are, for the most part, well defined. However there are some large blurred off-diagonal patches which stem from highly visually ambiguous areas. Considering the central figure, the spatial similarity matrix, we also see the off-diagonals although they are less defined reflecting the diminished certainty in matches coming from less discriminative (relative to the visual images) data.

Finally the R.H.S. figure shows the similarity matrix resulting from the combination of visual *and* spatial descriptions in the matching process. Importantly the blurred off-diagonal regions present in the visual matrix have been reduced in magnitude leaving a clear well defined off-diagonal trail of first loop to second loop correspondences. In particular note how the false positive visual match highlighted with an asterisk is down weighted when considered in conjunction with the local spatial appearance. The dark bands appearing in all three matrices are when one image (laser or visual) has so few descriptors that it cannot be reliably matched to anything in the database. This typically occurs when the vehicle drives very close to an object — a wall or parked car in our case.

6.3. Sequence detection in similarity matrix

Given a similarity matrix, we then pose the loop closing problem as the task of extracting statistically significant *sequences* of similar scenes from this matrix. Instead of relying on a single image match or image–laser scan match, [6] proposed a novel algorithm to detect loop closure from similarity matrices. It exploits topological links between neighboring local scenes to detect sequences of matching images and laser scans that indicate loop closure. The exact details of this algorithm can be found in [6].

Fig. 12 shows the results from applying the sequence detection algorithm onto the three similarity matrices, namely the visual similarity matrix, the spatial similarity matrix and the combined similarity matrix. It is noted that the second loop closure occurred from the 85th pose to 155th pose — a sequence of 71 poses. For the visual similarity matrix, a sequence of 104 matching pairs of images was detected. For the spatial similarity matrix, the top three most significant sequences are illustrated, with the longest sequence consisting of 8 matching pairs of laser scans. For the combined similarity matrix, the most significant sequence of 46 matching pairs of image–laser scans is shown.

The performances of sequence detection for the three similarity matrices are varied. A long sequence of matching pairs of images is detected in the visual similarity matrix but the sequence consists of a substantial amount of false positive matches. Although sequences detected from the spatial similarity matrix are very short, they do not contain any false positive matches. The sequence detected from the combined similarity matrix is significantly longer and, again, does not contain any false positive matches. In the light of a policy of preferring Type II errors over Type I errors (tolerating missed detections over false positives) the combined spatial/visual approach resulting in long substantial strings of positive matches is advantageous for loop-closure detection. In this

exemplar case detection occurs later for when using combined similarity matrix than when using the visual similarity and spatial similarity matrices.

7. Conclusions

We have developed a system which uses both spatial and visual appearance to guide and aid the detection of loop-closure events. We described how spatial shape information may be encoded and compared using entropy and relative entropy respectively. The spatial matching process is designed to be robust to occlusion and viewpoint changes. It uses a redundant number of transformations between salient features on segment boundaries. Finally, overall spatial similarity between two laser patches is determined by comparing both the shape of segments within patches and their mutual spatial arrangements. The folding in of spatial information has improved performance and has resulted in a promising and robust system. It is clear that folding in temporal (sequences) spatial (laser) and visual (images) terms is advantageous to the loop-closure detection problem. Although, by intent, we have not made any recourse to the SLAM p.d.f.s whose estimation we wish to support, we do not exclude the possibility of couching our approach in probabilistic terms. This would most likely require off-line learning of likelihoods and priors over vast hand-labelled data sets and this is an interesting area of current research.

References

[1] M. Bosse, P. Newman, J.J. Leonard, S. Teller, SLAM in large-scale cyclic environments using the atlas framework, International Journal of Robotics Research 23 (2004) 1113–1139.

[2] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (4) (2002) 509–521.

[3] S. Cohen, L. Guibas, Partial matching of planar polylines under similarity transformation, in: Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, January 1997, pp. 777–786.

[4] A. Davison, D. Murray, Simultaneous localization and map-building using active vision, Pattern Analysis Machine Intelligence 24 (7) (2002) 865–880.

[5] R. Hinkel, T. Knieriemen, Environment perception with a laser radar in a fast moving robot, in: Proceedings of Symposium on Robot Control, Karlsruhe, Germany, October 1988, pp. 68.1–68.7.

[6] K. Ho, P. Newman, Multiple map intersection detection using visual appearance, in: International Conference on Computational Intelligence, Robotics and Autonomous Systems, 2005.

[7] G. Jones, J. Foote, K. Sparck Jones, S. Young, Retrieving spoken documents by combining multiple index sources, Research and Development in Information Retrieval (1996) 30–38.

[8] T. Kadir, M. Brady, Saliency, scale and image description, International Journal Computer Vision 45 (2) (2001) 83–105.

[9] K. Konolige, Large-Scale Map-Making, in: Proceedings of the National Conference on AI (AAAI), San Jose, CA, 2004.

[10] L. Latecki, R. Lakämper, D. Wolter, Shape similarity and visual parts, in: International Conference on Discrete Geometry for Computer Imagery, 2003.

[11] L. Latecki, R. Lakämper, Application of planar shape comparison to object retrieval in image databases, Pattern Recognition 35 (1) (2002) 15–29.

[12] D.G. Lowe, S. Se, J. Little, Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks, International Journal of Robotics Research 21 (8) (2002) 735–758.

[13] D.G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2) (2004) 91–110.

[14] J. Matas, O. Chum, M. Urban, T. Pajdla, Robust wide baseline stereo from maximally stable extremal regions, in: Proceedings of the British Machine Vision Conference, 2002.

[15] C. Mikolajczyk, C. Schmid, Scale and affine invariant interest point detectors, International Journal of Computer Vision 60 (1) (2004) 63–86.

[16] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, L. Van Gool, A Comparison of affine region detectors, International Journal of Computer Vision 65 (1–2) (2005) 43–72.

[17] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit, FastSLAM: A factored solution to the simultaneous localization and mapping problem, in: Proceedings of the AAAI National Conference on Artificial Intelligence, 2002.

[18] J. Neira, J.D. Tardós, Data association in stochastic mapping using the joint compatibility test, International Transactions on Robotics and Automation 17 (6) (2001) 890–897.

[19] P. Newman, K. Ho, SLAM - Loop closing with visually salient features, in: International Conference on Robotics and Automation, 18–22 April, 2005.

[20] D. Ortin, J. Neira, J.M.M. Montiel, Relocation using laser and vision, in: International Conference on Robotics and Automation, 2004.

[21] C. Papadimitiou, K. Stieglitz, Combinatorial Optimization: Algorithms and Complexity, Prentice Hall, 1982.

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

[23] R. Sibson, SLINK: an optimally efficient algorithm for the single-link cluster method, The Computer Journal 16 (1) (1973) 30–34.

[24] J. Sivic, A. Zisserman, Video Google: A text retrieval approach to object matching in videos, in: Proceedings of the International Conference on Computer Vision, October, 2003.

[25] S. Thrun, Y. Liu, D. Koller, A.Y. Ng, Z. Ghahramani, H. Durrant-Whyte, Simultaneous localization and mapping with sparse extended information filters, International Journal of Robotics Research 23 (7–8) (2004) 693–716.

[26] H. Wolfson, On curve matching, IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (5) (1990) 483–489.

[27] J. Weber, K. Jörg, E. Puttkamer, APR-global scan matching using anchor point relationships, in: The 6th International Conference on Intelligent Autonomous Systems, IAS-6, Venice, Italy, July, 2000, pp. 471–478.

Kin Leong Ho received his B.Sc. in Systems Engineering with Distinction from the United States Naval Academy in 2003. He is a DPhil student at Oxford University Robotics Research Group under the sponsorship of the Rhodes Scholarship. Currently, he is serving as a naval combat officer in the Republic of Singapore Navy. His research interests are in the area of mobile robotics, namely vision based navigation, cooperative robotics and simultaneous localization and mapping.

Paul Newman obtained an M.Eng. in Engineering Science from Oxford University in 1995. After a brief sojourn in the telecommunications industry in 1996 he undertook a Ph.D. in autonomous navigation at the University of Sydney, Australia. In 1999 he returned to the United Kingdom to work in the commercial sub-sea navigation industry. In late 2000 he joined the Department of Ocean Engineering at M.I.T. where as a post-doc and later a Research Scientist, he worked on algorithms and software for robust autonomous navigation for both land and sub-sea agents. In early 2003 he was appointed to a Departmental, and in 2005, University Lectureship in Information Engineering at the Department of Engineering Science, Oxford University. He heads the Mobile Robotics Research group and has research interests in pretty much anything to do with autonomous navigation.