Combining Visual and Spatial Appearance for Loop Closure Detection in SLAM

K. Ho, P. Newman
Oxford University Robotics Research Group

***Abstract*— In this paper we describe a system for use on a mobile robot that detects potential loop closures using both the visual and spatial appearance of the local scene. Loop closing is the act of correctly asserting that a vehicle has returned to a previously visited location. It is an important component in the search to make SLAM (Simultaneous Localization and Mapping) the reliable technology it should be. Paradoxically, it is hardest in the presence of substantial errors in vehicle pose estimates which is exactly when it is needed most. The contribution of this paper is to show how a principled and robust description of local spatial appearance (using laser rangefinder data) can be combined with a purely camera based system to produce superior performance. Individual spatial components (segments) of the local structure are described using a rotationally invariant shape descriptor and salient aspects thereof, and entropy as measure of their innate complexity. Comparisons between scenes are made using relative entropy and by examining the mutual arrangement of groups of segments. We show the inclusion of spatial information allows the resolution of ambiguities stemming from repetitive visual artifacts in urban settings. Importantly the method we present is entirely independent of the navigation and or mapping process and so is entirely unaffected by gross errors in pose estimation.**

***Index Terms*— Mobile Robotics, SLAM, Loop Closing, Saliency, Visual Features, Spatial Descriptions, Data Association.**

I. Introduction and Motivation

SLAM (simultaneous localisation and mapping) is a core information engineering problem in mobile robotics and has received much attention in past years especially regarding estimation theoretic aspects. 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 markedly manifested during what has become known as loop closing — the act of correctly asserting that the vehicle has returned to a previously visited location. Loop closing is hardest in the presence of substantial errors in vehicle pose estimates — exactly when it is needed most.

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. This method fails catastrophically just when it is needed most. 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 and the robot is, for all intents and purposes, lost. This is bad.

Paul Newman and Kin Leong Ho are at the Oxford University Robotics Research Group email: {pnewman,klh}@robots.ox.ac.uk

Figure 1 shows an obvious case of poor loop closing, linearisation 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. (“so bad” was caused by an earlier error of a fraction of a degree) The situation depicted in figure 1 might be avoided by adjusting noise parameters, employing a different SLAM algorithm entirely or simply adding bespoke error checks here and there. Such changes may allow successful loop closing in this and perhaps several other similar cases but the central problem still remains - it is an unstable equilibrium to use SLAM estimates in the loop closing process when the statistics themselves may be at best biased and at worst inconsistent.

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 [15] or continually trying to relocate in a bounded set of sub-maps [2] 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.

As argued in [18], the hard part about loop closing is not handling the presence of a loop but detecting when loop closure is even a possibility. To do this one needs to decide when and where to look. Searching only in the neighborhood of the vehicle is not robust in the face of gross vehicle error.

In [6] hyper-priors are learnt off-line typifying the geometric and topological structure of regions (corridors and intersections) commonly found in indoor settings. As the robot(s) moves through its/their environment, local scene observations are combined with the initial hyper-prior to produce a modified posterior. This distribution is used as a generative model for the observations of the local scene and used to calculate the probability of new measurements being “in or out-of-map”. Although this method does offer substantially improved robustness and performance its success is predicated upon good structural priors which are applicable to the entire workspace.

It is possible to integrate out the dependence on vehicle pose and search over all possible vehicle poses - essentially repetitively solving the kidnapped robot problem [16]. Another very attractive proposal is to eschew the need to make hard and fast, one-time-only data association decisions and instead use a mechanism that allows past decisions to be revoked or changed and their effect to be vanquished from the state estimates. [7][19]. While this policy takes the sting out of making the wrong decisions and would undoubtedly have a substantial effect on the overall reliability of SLAM systems, it does not negate or depreciate the advantages in making better decisions in the first place.

Many authors have successfully used visual landmarks in SLAM, for example [20], [3], [13], [5]. 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 carefully choose them so that their saliency and wide-baseline visibility allow detection of loop closure events independently of vehicle estimates.

The extraction and use of these "salient" features is described in [18] and briefly summarized in the next section. The contribution of this paper is to build on this work and to extend a visual appearance based system into a visual *and* spatial appearance based system. In this work the spatial appearance will be measured by a conventional, ubiquitous laser range finder.

Our motivation for this extension is two-fold. Firstly multi-modal sensing naturally leads to richer and more discriminative descriptions - just as shifting from 2D laser scans to 3D laser scans does. Our second and perhaps more tangible motivation is to address shortcomings in our solely visual-appearance system. 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. This paper shows that by describing the local spatial appearance of the image capture locale, false visual matches can be successfully discriminated against. This increases the reliability of loop closure detection.

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 [10]) 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.

A. Related Work

We propose a new method of comparing laser patches based on their mutual spatial similarity irrespective of beliefs in their global location. The conventional correlation based scan matching technique (which was used in the SLAM algorithm producing Figure 1) was discounted because it was inefficient for 3D scan matching and, in the absence of a prior on the relative pose between patches, produced false matches especially in the presence of substantial occlusion and capture locations.

The description of shape is central to our system. We use shape as a key quality of the laser data. Shape descriptors of boundary contours of 2D objects are widely adopted in a multitude of applications, particularly in shape-based object retrieval in image databases. Here, we highlight the techniques on which we rely and build. Hinkel [8] developed the angle histogram method which measures the relative angle between any two adjacent laser range points. Weiss [23] employed the angle histogram to match rangefinder scans from different locations and hence compute the translational and rotational displacement of a mobile robot. Importantly the authors point out that an angle histogram is largely rotationally translationally invariant . Weber [22] attempts to find matches to a query laser scan by graphs constructed of anchor points — reproducible object feature positions that correspond to sharp edges in the angle function.

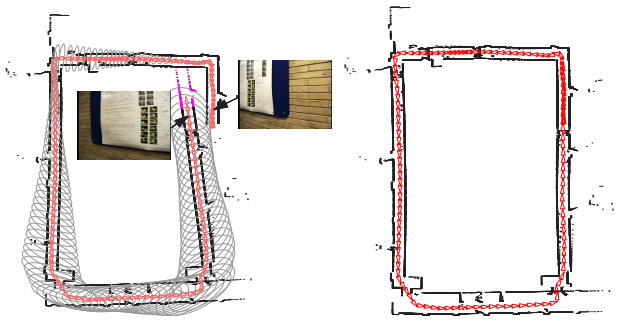
A laser scan can be deemed as an top-view image of the geometric structure of the environment and though most efforts have concentrated on extracting shape descriptors of 2D objects in images [24], [1]. Latecki et. al [11] have applied their shape similarity system to the problem of robot localization and mapping in recognition of the similarity in these two problems.

II. An Image-Based Retrieval System

In [18], a system was developed able to close loops with visually salient features. Figure 1 shows a typical result in which two, automatically detected visually salient images were used to close a loop.

Visual saliency is a broad term that refers to the idea that certain parts of a scene are "pre-attentively distinctive" [17]. The Scale Saliency algorithm that is used in [18] was proposed by Kadir and Brady [9] in which salient regions within images are defined as a function of local image complexity weighted by a measure of self-similarity across scale space.

In addition to being salient we wish to detect image features that are robust to changes in view point. The motivation for this is as follows. The vehicle camera is unlikely to have the same pose when the host vehicle revisits an area as it did when it first encountered it. We adopt one such detector [14], which finds "maximally stable extremal regions" or "MSERs" and offers significant invariance under affine transformations. The reason for the wide-baseline stability of the technique lies in the fact that connectivity (which is essentially what is detected) is



(a) (b)

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 in red. 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.

preserved under reasonable affine transformations ($< 70^0$ in the plane).

Having found image regions that fulfil the above two criteria (salient *and* wide-baseline stable) 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 [12] which has become immensely popular in computer vision applications [21] and used with good effect in SLAM in [13].

A. A Failure Condition

The left-hand column of Figure 10 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 yet it incorrectly suggests a loop closure event. The geometry of the local environments are truly different. 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.

III. Spatial Descriptors

We begin by describing how a complete laser patch is passed through a simple 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 highly discriminative system.

A. Initial Segmentation

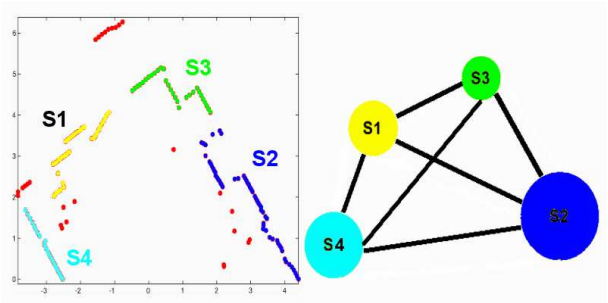


Fig. 2. Above shows a typical geometry patch after segmentation and a graph depiction of the way we encapsulate the information contained. Each node is a segment and contains the CAF function, its entropy measure and a list of critical points. The edges represent a known spatial relationship between segments.

The laser scan is divided into smaller but sizeable "segments". These segments are formed using a standard nearest neighbor clustering algorithm. 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 the environment. Figure 2 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.

B. Segment Descriptors

The segmentation completed, we now desire to describe each segment. The generated descriptors will be the values of each node in Figure 2. 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:

1) The Cumulative Angular Function: Each segment is described by the "cumulative angular" or "turning" function [23], [4] as illustrated by Figure 3. The turning function is a plot of the cumulative change in turning angle ϕ versus the arc-length of the segment. To illustrate, the turning function maps straight lines $ax + by + c = 0$ to $\phi = 0$, circles to $\phi = \alpha\phi$ and squares to a "staircase" function in ϕ. A key characteristics of the cumulative angular function is that it is rotational and translational invariant. Before describing the segment, the points in the segment are passed through a low-pass filter to filter the effects of small amplified high frequency noise from uncertainties in range measurements.

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

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

shapes is far 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.

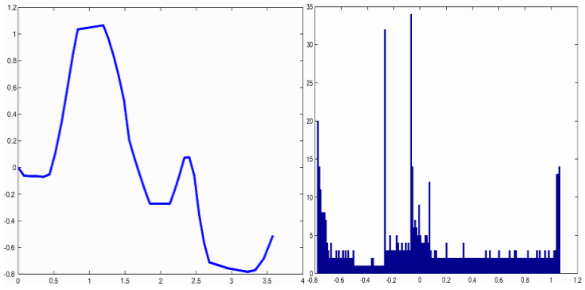


Fig. 3. The cumulative angular function is transformed into a histogram of angular values. Each bin contains the number of points along the cumulative angular function that have angular values that fall within the bin value. Using this histogram of bin values, the entropy of the cumulative angular function can be calculated.

In this case we may write an expression for entropy as

$$\mathcal{S}_{\mathcal{D}} = \int_{i \in D} P_{D_i} \log_2 P_{D_i} d_i \qquad (1)$$

where P_{D_i} is the probability of descriptor i and 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 1.

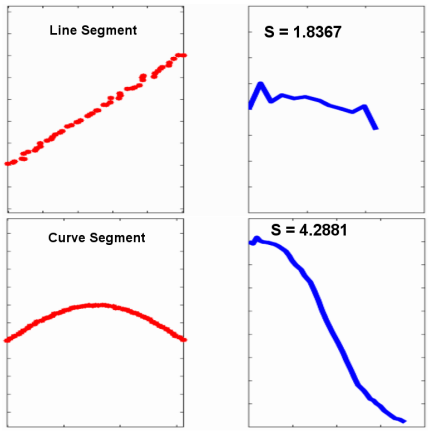


Fig. 4. Segments of laser data (right) are mapped to cumulative angle functions (left). S_D is calculated from Equation 1.

Figure 4 illustrates the process. First cumulative angle functions are calculated and then binned and integrated according to 1. Note how as expected the line segment has a smaller entropy value (1.8) while the more interesting curve segment has a larger entropy value (4.3). 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.

C. Inter-Segment Descriptors

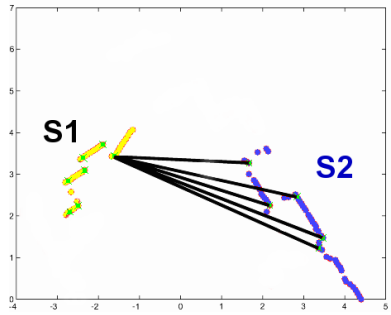


Fig. 5. The way in which the relationship (in this case an SE2 transformation) between segments is encoded. The two segments S and S' contain nc and nc' critical points respectively. For each critical point in S we form a"bundle" of links to all nc' critical points in S'. In all there will be nc bundles 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 B 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.

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 2. We do this by first extracting points of high curvature along the segments. We call these "critical points". Critical points are sharp changes in the cumulative angle function and they are marked as crosses in the laser patches shown in Figure 5.

Repeatability of extraction of these critical points is an important consideration. The thresholding on C.A.F (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.

2) Segment Configurations: The distance and relative orientation between critical points form the links (the lines joining the two segments shown in figure 5 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 done using simply the largest eigenvector of the segment.

IV. Descriptor Comparison

The previous section described the generation of descriptors from a laser patch. This section discusses how these descriptors can be compared to one another.

A. Segment Descriptor Comparison

We now describe how two segment descriptors generated according to Section III-B can be compared to one another. Each segment (a node in the graph of Figure 2) 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.

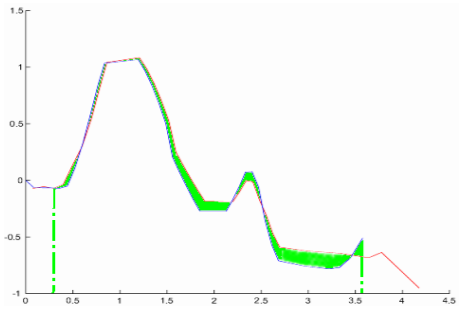


Fig. 6. Figure 6 shows the disparity between two CAFs of two segments, S and S'. CAF is the one-dimensional representation of 2-D segments, which encodes the structure of the points within the segment by the change in tangential angles between consecutive points. The difference between the two CAFs is 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.

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 from a 3-D search space $[x, y, \theta]$ problem into a 2-D search space [4]. 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 slide horizontally to find the minimum error between the query curve and the pattern curve, see Figure 6. This approach is similar to the method employed by [4], except that their search problem is in scale-position-rotation space. The difference, $e(\beta, \gamma)$, between CAFs is calculated as

$$e(\beta, \gamma) = \int_0^l (T_1(s) - T_2(s + \beta) + \gamma)^2 ds \qquad (2)$$

where the two cumulative angular functions 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

$$\eta_1 = \frac{1}{1 + e} \qquad (3)$$

.

2) Match Length Disparity: A second scalar η_2 is calculated as the matched length to total length ratio:

$$\eta_2 = \frac{l(m)}{l(T)} \qquad (4)$$

where $l(m)$ is the length of the matched segment portion and $l(T)$ is the total length of the query segment.

In figure 6, 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.

3) Entropy Disparity: We use relative entropy to measure the similarity between segments. The relative entropy , or the Kullback-Leibler distance, is given by:

$$K(f||f') = \sum_{i=1}^{m} f_i \times \ln \frac{f_i}{f'_i} \qquad (5)$$

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 Figure 3. 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.

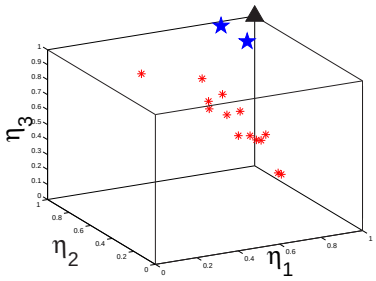


Fig. 7. Segment to Segment matching using the similarity vector η. The triangle represents a perfect match(Identical segments). The figure shows the similarity between a query segment, Sq and all segments from two other scans S' and S''. Two close matches are found which are depicted as stars. The axes are angular function similarity measure, matched length ratio measure and entropy similarity measure.

The above three similarity scalars are stacked in vector $\eta_{S,S'} = [\eta_1, \eta_2, \eta_3]^T$. That describes the degree of similarity between S and S'. If S and S' are identical segments $\eta_{S,S'}$ will be $[1, 1, 1]^T$ In figure 7, the position of every segment is displayed in η-space. The triangle represents the position of a perfect match with the query segment in all three similarity measures. The stars correspond to the segments that are most similar to the query segment. The asterisks are the positions of other segments η-space.

B. 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 [22], we determine the similarity between the segment-segment links by matching arrays of distances and relative orientations of the segment-segment edges. In Figure 8, the segment-segment relationships for two laser scans are shown. Due to occlusions, a minority

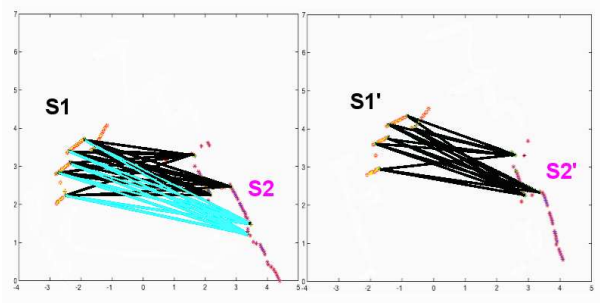


Fig. 8. A method of comparing inter-segment relationships (edges). In our determination of similarity between edges, the bundles of links that comprise the edges are compared against each other using distance and relative angle criteria. The dark links represent those that have been successfully matched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

of the critical points found in one laser scan are not seen in the other. The links that are successfully matched are highlighted in a darker tone. To establish correspondence of spatial configuration between the segments, a 50 percent paring is required.

V. Matching

We now consider the task of deciding whether a particular scan $\mathcal{S}$ matches another scan $\mathcal{S}'$. We are not willing to tolerate reliance on scans containing straight lines or 'point features'. We wish to be robust to both occlusion and gross changes in view point. Importantly, for the reasons argued in section I we must not require any prior on the likely transformation between $\mathcal{S}$ and $\mathcal{S}'$. We progress by discussing segment to segment matching (node to node in the graph of Figure 2) and then move on to segment group to segment group matching.

We begin by randomly selecting a random segment i from the n_s segments in $\mathcal{S}$. We then find n'_s similarity vectors $\eta_{i,1:n'_s}$ between i and the n'_s segments in $\mathcal{S}'$ as described in section IV-A. The best pairing $< i, j' >$ is chosen such that $j' = \arg\min_j(|| \mathbf{1} - \eta_{i,j} ||)$.

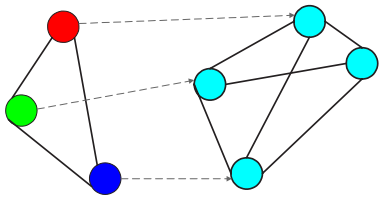


Fig. 9. The alignment process aims to find a match between two graphs. A positive match only occurs if the individual nodes match based on the process described in subsection III-B and the individual edges match based on the process described in subsection IV-B.

The final step of our method seeks to match the graph representations of laser patches and is similar to that used in [22]. It tries to match both the shape and saliency properties of the segments and their relationships to neighboring clusters in the presence of substantial occlusion and partial observations. In essence we are matching segment shape and segment-cluster shape across two scans - this is akin to finding a largest subgraph of $G(S)$ in $G(S')$ and $G(S')$ in $G(S)$. However, presence of occlusion and partial observability means nodes and edges are likely to missing. We seed the search with the best pairing $< i, j' >$. Each of the neighbors of segment i in $\mathcal{S}$ is compared with the neighbors of segment $j*$ in $\mathcal{S}'$. If a new segment-segment correspondence is found using the methods of Section IV and the edges between segments also pass the similarity test the algorithm continues to examine and compare the neighbors of the new connected segments. In this way a connected, matching subgraph is built from the two most similar segments in the two patches. We currently employ an empirical heuristic to declare two patches similar — if the cardinality of the matching subgraph is greater or equal to two a positive match is signalled. This is fairly stringent condition as it implies shape finding strong similarities between three or more substantial, complex shapes and their mutual arrangement being similar in both patches.

VI. Experimental Results

To examine and demonstrate the effectiveness of our approach, we tested our algorithm in both indoor and outdoor environments. In the indoor environment, a small ATRV-Jnr mobile robot was driven up and down in a loop along the stretch of corridor in a building. We note that this is by no means a large loop or an extremely challenging environment for contemporary SLAM algorithms but it does posses some visually confusing properties.

The vehicle camera kept a constant orientation in vehicle coordinates –looking forward and slightly to the right. 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 along side the visual information as a collection of spatial descriptors. In this indoor, experiment, a database of a little under 200 images was collected. Figure 10 shows the rare but troublesome failure condition occurring when only visual appearance is used. Replicas of the same poster were found in different locations along a corridor. The query image is shown in the center and a false loop closing event is signalled (albeit a visually correct match) with the image on the left. However the spatial processing we have described discredits this match while confirming the second match shown on the right hand side.

The same equipment and software was used in an outdoor experiment. The ATRV-Jnr mobile robot was driven around a car park in front of a building. Here, a database of 148 pictures was collected — see Figure 11 — where the ground is relatively flat. As is common in urban environments, there were many replicated small-scale objects, for example the white-framed windows are repeated along the length of the building. Once again the combined processing

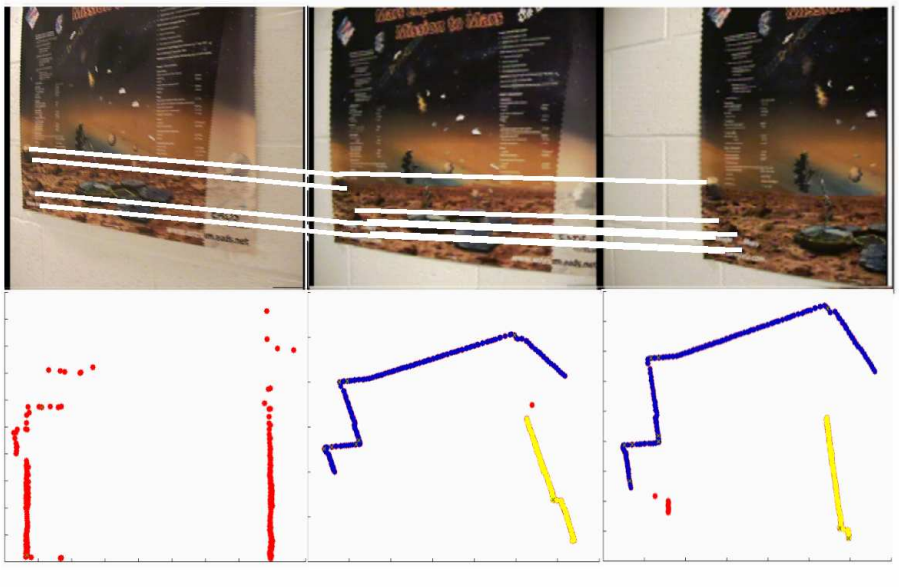


Fig. 10. Image matching using Maximally stable extremal regions and scale saliency regions. The query image is shown in the center and a false loop closing event is signalled (albeit a visually correct match) with the image on the left. However, the spatial processing we have described in this paper discredits this match while confirming an alternative visual match shown on the right hand side.

of visual and spatial appearance measurements allows a visually ambiguous situation to be resolved. In these two modest experiments we saw no false positives when using both spatial and visual information. We are currently in the process of collecting a more substantial data set with an order of magnitude more image/laser-patch pairs.

VII. Conclusions, Issues and Future Work

Although the results presented here only use 2D spatial descriptions it is important to note that the ideas presented here can be equally applied to 3D laser data — something that is a current area of research as an natural extension to this work. Segments become patches, critical points become points of locally-maximum Gaussian curvature, segment length becomes patch area and the entropy metrics remain unchanged while the graph edges simply become 6DOF transformations rather than in-plane shifts.

The work we have presented here uses several threshold values. We are currently looking at learning suitable values from large data set rather than using the experimentally chosen values we use at the moment. We also believe that the the work would benefit from estimating and using probability of spatial and visual matches as a function of the similarity metrics. This is work underway.

We have developed a system which uses both spatial and visual appearance to guide and aid the detection of loop closure events. Spatial shape information is encoded and compared in a principled way using entropy and relative entropy respectively. The spatial matching process is designed to be robust to occlusion and view point 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 markedly improved performance and has resulted in a robust, useful system.

References

[1] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligience*, 24(24):509–521, April 2002.

[2] M. Bosse, P. Newman, J. J. Leonard, and S. Teller. SLAM in Large-scale Cyclic Environments using the Atlas Framework. *International Journal of Robotics Research*, 23:1113–1139, 2004.

[3] J. A. Castellanos, M. Devy, and J. D. Tardós. Towards a topological representation of indoor environments: A landmark-based approach. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems*, pages 23–28, 1998.

[4] S. Cohen and L. Guibas. Partial matching of planar polylines under similarity transformations. *Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 777–786, January 1997.

[5] A. J. Davison and D. W. Murray. Simultaneous localization and map-building using active vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):865–880, 2002.

[6] D. Fox, J. Ko, K. Konolige, and B. Stewart. A hierarchical bayesian approach to the revisiting problem in mobile robot map building. In *Proceedings of International Symposium on Robotics Research*, 2003.

[7] D. Hahnel, W. Burgard, B. Wegbreit, and S. Thrun. Towards lazy data association in slam. *11th International Symposium of Robotics Research, Sienna*, 2003.

[8] R. Hinkel and T. Knieriemen. Environment perception with a laser radar in a fast moving robot. *In Proceedings of Symposium on Robot Control, Karlsruhe, Germany*, pages 68.1–68.7, October 1988.

[9] Timor Kadir and Michael Brady. Saliency, scale and image description. *International Journal Computer Vision*, pages 83–105, 2001.

[10] K. Konolige. Large-scale map-making. *Proceedings of the National Conference on AI (AAAI), San Jose, CA*, 2004.

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

[12] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

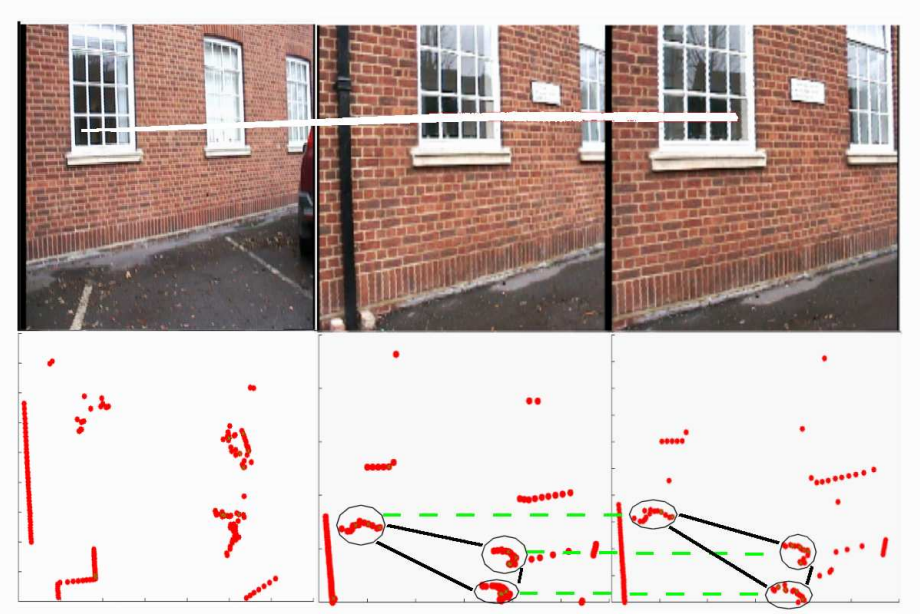


Fig. 11. Results in an outdoor environment. As in the case shown in Figure 10, a false positive loop closure is signalled w.r.t. the left-hand image when only visual information is taken into consideration. This is discounted when spatial descriptors are used in addition. The query image and patch is 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.

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

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

[15] J. Neira and J. D. Tardós. Data association in stochastic mapping using the joint compatibility test. *IEEE Trans. Robotics and Automation*, 17(6):890–897, 2001.

[16] J. Neira, J. D. Tardós, and J. A. Castellanos. Linear time vehicle relocation in slam. *IEEE Transactions on Robotics and Automation*, 2003.

[17] U. Neisser. Visual search. *Scientific American*, pages 94–102, 1964.

[18] P. Newman and K. Ho. SLAM - Loop Closing with Visually Salient Features. *To be published in IEEE International Conference on Robotics and Automation*, 18-22 April 2005.

[19] P. M. Newman and H. F. Durrant-Whyte. An efficient solution to the SLAM problem using geometric projections. *Proceedings of the November 2001 SPIE conference Boston, USA*, 2001.

[20] D. Ortin, J. Neira, and J.M.M. Montiel. Relocation using laser and vision. *Proceedings of the IEEE International Conference on Robotics and Automation*, 2004.

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

[22] J. Weber, K. Jörg, and E. Puttkamer. APR-Global scan matching using anchor point relationships. *The 6th International Conference on Intellgient Autonomous Systems (IAS-6), Venice, Italy*, pages 471–478, July 2000.

[23] G. Weiβ, C. Wetzler, and Puttkamer E. Keeping track of position and orientation of moving indoor systems by correlation of range-finder scans. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1994.

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