Proceedings of the 2005 IEEE
International Conference on Robotics and Automation
Barcelona, Spain, April 2005

SLAM- Loop Closing with Visually Salient Features

Paul Newman and Kin Ho
Oxford University Robotics Research Group
Department of Engineering Science
University of Oxford, OX1 3PJ
{*pnewman,klh*}*@robots.ox.ac.uk*

***Abstract*— Within the context of Simultaneous Localisation and Mapping (SLAM), "loop closing" is the task of deciding whether or not a vehicle has, after an excursion of arbitrary length, returned to a previously visited area. Reliable loop closing is both essential and hard. It is without doubt one of the greatest impediments to long term, robust SLAM.**

This paper illustrates how visual features, used in conjunction with scanning laser data, can be used to a great advantage. We use the notion of visual saliency to focus the selection of suitable (affine invariant) image-feature descriptors for storage in a database. When queried with a recently taken image the database returns the capture time of matching images. This time information is used to discover loop closing events. Crucially this is achieved independently of estimated map and vehicle location.

We integrate the above technique into a SLAM algorithm using delayed vehicle states and scan matching to form interpose geometric constraints. We present initial results using this system to close loops (around 100m) in an indoor environment.

***Index Terms*— Mobile Robotics, SLAM, Loop Closing, Saliency, Visual Features**

I. Introduction and Motivation

The SLAM research community has made good progress in the past years on the SLAM-estimation problem. We now have a set of methods capable of simultaneously estimating vehicle location and building a workspace map — itself used in the localization task. Particular progress [25], [17], [31], [8] has been made with regard to the scaling problem — how to prevent the runaway computational cost with workspace size seen in early SLAM algorithms [33], [7]. Yet with all this progress we still do not have the kind of SLAM-enabled systems hoped for. This issue is clear - SLAM systems lack robustness. The core problem is Data Association — the task of placing measurements into one of the following categories:

- associated with as yet unknown regions of workspace
- associated with already known (mapped) region
- association Pending - not enough evidence to make a decision
- spurious.

It is common practice to use the estimates produced by the SLAM algorithm itself to aid this decision. The naive approach adopted in early SLAM work [7] 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 — probably for good.

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 [26] or continually trying to relocate in a bounded set of sub-maps [3] 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.

It seems hard, yet important, to step out of the dubious, self-referential circle of using a potentially erroneous estimate (pose) to make a decision regarding the fusion of measurements — the outcome of which will affect the very estimates that are being used to make the decision in the first place.

The hard part about loop closing is not asserting 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. Note that a gross vehicle pose error does not imply a gross error of judgement was made. A small heading error over long linear traversals quickly leads to substantial position errors.

It is possible to integrate out the dependence on vehicle pose and search over all possible vehicle poses - essentially solving the kidnapped robot problem as often as possible. For example in [27] a solution to the kidnapped robot problem is proposed that is linear in the number of mapped features. 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. [13][29]. 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.

In [9] 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.

In this paper we argue that if SLAM is to become the robust tool it should be, then we should not rely on using the same source of geometric measurements for mapping, localizing and data association — something else is needed. We should look to use sensorial information that is outside the central SLAM estimation loop. The use of a camera to provide this out-of loop information is an obvious choice. Cameras are cheap, ubiquitous, passive and information rich. Given this enviable list of properties one might question why cameras should not be the central sensor in a SLAM system. Unfortunately, it remains disproportionately hard to use a camera (a bearing only sensor) for SLAM on a mobile vehicle. Even though great progress is being made in realtime, single-camera SLAM (for example [6][15][21]), the quality of maps and efficacy of algorithms obtained using scanning lasers over very large areas still surpasses the state-of-the-art in vision based SLAM. Nevertheless, we show in this paper how a judicious and selective use of visual information can greatly improve the performance of a laser-based SLAM system.

In [14], a camera was used to capture images of assumed-planar quadrilaterals and transformed under a homography to remove perspective distortion. These images were matched using Harris interest points matching and used for navigation on a mobile robot. In [30] camera and laser information are used in combination to localize and build a map consisting of planar facets. In this paper, however, we place no constraint on the geometry of the vehicles environment.

II. Visual Feature Processing

We propose that for visual features to aid the Loop Closing task they must be salient, wide-baseline-stable and descriptive. This section discusses motivation for each of these requirements and the methods we employ to meet them.

A. Saliency

This criterion is central to our case and is a key difference between the work presented here and [21]. We seek image regions that are locally distinct and "stand out" from their immediate background.

Commonly in contemporary SLAM work, each measurement coming from "geometric-range" sensors like scanning laser, radar and sonar is either stored with only minimal further processing (scan-matching techniques [18]) or is tested against one of a set of geometric models in the map describing the workspace (feature-based SLAM [1]). Rarely is enough consideration given to the distinctiveness of the feature and/or aspects of the local region (however [4] presents an interesting case of using topological saliency).

Visual saliency is a broad term that refers to the idea that certain parts of a scene are "pre-attentively distinctive"[28]. The Scale Saliency algorithm we use here was proposed by Kadir and Brady [16] and was based on earlier work by Gilles [11]. Salient regions within images are defined as a function of local image complexity weighted by a measure of self-similarity across scale space. Entropy,H, is a natural choice to measure image complexity. Consider a region D containing n pixels described as $(d_0 \dots d_n)$. At some scale s we can write the entropy H_D of the region D around a pixel at position $\vec{x}$ as a function of s and $\vec{x}$:

$$H_D(s, \vec{x}) = -\int_{i \in D} P_D(s, \vec{x}) \log_2 P_D(s, \vec{x}).d_i \qquad (1)$$

Here $P_D(s, \vec{x})$ is a pdf built from the image data in the region (parameterized by s) D surrounding a pixel at $\vec{x}$ which encodes the probability of descriptor d_i within D. For a given $\vec{x}$ particular choices of s, call them $S = \{s_0 \dots s_k\}$, which cause H_D to peak are interesting.

Sharp peaks imply a rapid change in entropy around a given scale whereas shallow peaks imply a large degree of self similarity and so are less interesting. This preference for "interesting scales" is implemented by weighting the entropy according to the rate of change of the statistics of $P_D(S, \vec{x})$ with s.

Hence by defining an entropy vector $H_D(S, \vec{x})$ with one element for each element of S a saliency metric $\mathcal{Y}(\vec{x}, S)$ can be written as

$$\mathcal{Y}_D(S, \vec{x}) = H_D(S, \vec{x}) \times \mathcal{W}_D(S, \vec{x}) \qquad (2)$$

and the weighting function for an element of S, $\mathcal{W}_D(s, \vec{x})$ is given by:

$$\mathcal{W}_D(s, \vec{x}) = s. \int_{i \in D} |\frac{\partial}{\partial_s} P_D(s, \vec{x})|.d_i \qquad (3)$$

The term $\mathcal{Y}_D(S, \vec{x})$ is calculated for all pixels in the image resulting in a cloud of points in $\Re^3$ –(x, y, s). Finally these points are clustered into groups with similar x and y positions.

The left hand column of figure 1 shows some results of running this algorithm over two images taken from a mobile robot as it moves down a corridor and passes an opening into an office.

This form of saliency detection was also suggested in [10] but without demonstration of a successful implementation in a SLAM algorithm. In contrast to this work we also actively seek regions which are likely to be wide-baseline visible. This will now be discussed.

B. Wide-Baseline Stability

The saliency detector just described selects image regions that are interesting in the context of a single image. In addition to being salient we wish to detect image features

Fig. 1. Two indoor images grabbed two seconds apart from a mobile robot trundling down a corridor. The first column of images highlight the salient regions detected scale saliency algorithm described in Section II-A with light circles. The middle column illustrates maximally stable extremal regions described in Section II-B. The last column of images highlights the matching of salient, MSER, interest points described with SIFT descriptors. It should be noted that the matching lines are not parallel because the interest points are not found on a planar surface. Under a variation in viewpoint, the interests point undergo differing translations in the image plane.

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. However, if the same world-entity is being observed albeit from a different position and angle it is clearly advantageous to be able to re-detect it. The task is to find a detector that offers such wide-baseline performance. One such detector [23] finds "maximally stable extremal regions" or "MSERs" which offer significant invariance under affine transformations.

Consider an image consisting of pixels taking on values in the range $D = \{d_{max} \ldots d_0\}$(for example 8-bit intensity in the range [0:255]). Set an index i to 0 and for simplicity assume only one pixel, q has value d_{max} ($D[0]$) , this pixel is placed in a set R. The method proceeds by incrementing i and examining all connected neighbors of R (which at this point contains only q) and adding them to R if their value is $D[i]$. The algorithm then iterates once more, incrementing i and this time testing all neighbors of the enlarged R. The set R is classified as an MSER when its size remains constant w.r.t i - in other words the region has stopped growing and there is a discontinuity of pixel values all around its perimeter. The use of the "union-find" algorithm allows fast implementation of the set operations involved (for example when two regions merge). For further detail the reader is referred to [5], [23]. The reason for the wide-baseline stability of the technique lies in the fact that connectivity (which is essentially what is detected) is preserved under reasonable affine transformations.

Fig. 2. The combination of MSER regions and SIFT descriptors leads to good wide-base line matching (This is particular true for planar surfaces). Here lines indicate the correspondences found between features in very different views of a poster.

C. Feature Descriptions

Having found image regions that fulfil the above two criteria (wide-baseline stable *and* falling within a salient region) we need to 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. A sensible choice here is the SIFT descriptor [20] which has become immensely popular in computer vision applications [32] and used with good effect in SLAM in [21]. To summarize the approach, we take the salient MSE regions and place a 4×4 grid over them. For each of the 16 cells in the grid, a pixel gradient magnitude is calculated at 45^o intervals. This yields a $4 \times 4 \times 8 = 128$ dimensional descriptor vector for each processed region. Figure 2 shows the typical wide baseline performance we achieved on planar surfaces.

III. Application to Loop Closing

We are now in a position to use the above three techniques to close loops in a SLAM problem. As the

vehicle moves around its environment and explores new areas it occasionally (every few seconds or meters of driven path) takes a picture through an onboard camera. These pictures are passed through a saliency-MSER-descriptor pipeline and incrementally a database of descriptors is built. Each image will produce a whole set of descriptors which are stored alongside the time at which the image was captured. The data base can be queried every time a new image is acquired (before adding it to the database) or asynchronously. The mechanism we employ to perform the query is simple but has a complexity linear in database size (this is something we propose to improve upon in future work). The query image I_Q generates n_q descriptor vectors V_q. For each stored candidate image I_C in the database with n_c descriptors V_c a $n_q \times n_c$ adjacency matrix $M_{q,c}$ is created where the $(i,j)^{th}$ entry $M_{q,c}(i,j)$ is the $\mathcal{L}_2$ norm $\mid\mid V_q(i) - V_c(j) \mid\mid$. These distances are thresholded resulting in n_{qc} matched descriptors between the query image and the candidate image in the database. [1] When all images have been compared those candidates producing the largest number of feature matches $n_{q,c}$ are selected as wide-baseline matches. In particular the times at which the image was captured can be used for loop closing.

Consider the case when a new image is added to the database and a correspondence is found between that and an image taken much earlier in the SLAM session. One field of the query result contains the time t_m at which the earlier image was taken. Under reasonable assumptions (which will be discussed later) this match makes a strong assertion that the vehicle is now once more close to where it was at time t_m. Note that the database stores only visual and temporal data. We do not store the estimate of the position of the vehicle in the database because by the time a match is found other loop closing or estimation events may have rendered this estimate invalid. By keeping an external journal of position and time (which must be updated if the SLAM algorithm employed makes substantive changes to old position estimates) a search can be initiated to relocate the vehicle near where it was at time t_m or to make a concerted effort to associate current measurements with components of the map built earlier at t_m. Note we are not using the estimated state to make decisions about when loop closing should occur. We only use it to process the event.

IV. Experimental Results

A. A SLAM Implementation

To illustrate the effectiveness of our approach we choose to employ a simple, single coordinate frame, non-constant time, laser based scan matching SLAM algorithm. This choice is made entirely without prejudice - any SLAM algorithm could have been used. We choose to use this particular method because it is simple to explain and offers good performance in our chosen environment. The SLAM technique described below is in spirit close to [22],[18] uses the delayed state ideas in [19][24] and is similar to one of the SLAM schemes employed in [3] although here we use a different scan matching technique.

The estimated quantity is a state vector $\mathbf{x}(i|j)$ which initially contains a single vehicle x, y, θ pose $\mathbf{x}_v(0|0)$. Associated with it is a covariance matrix $\mathbf{P}(0|0)$. Here we are adopting the common notation that the quantity $\mathbf{x}(i|j)$ is the estimate of the true state $\mathbf{x}$ at time i given measurement 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.

$$\mathbf{x}_v(k+1|k) = \mathbf{x}_v(k|k) \oplus \mathbf{u}(k+1) \tag{4}$$

Where $\oplus$ is the transformation composition operator as used originally in [33] which has the following two jacobians associated with it:

$$\mathbf{J}_1(\mathbf{x}_1, \mathbf{x}_2) \triangleq \frac{\partial(\mathbf{x}_1 \oplus \mathbf{x}_2)}{\partial \mathbf{x}1}$$
$$\mathbf{J}_2(\mathbf{x}_1, \mathbf{x}_2) \triangleq \frac{\partial(\mathbf{x}_1 \oplus \mathbf{x}_2)}{\partial \mathbf{x}2}$$

These allow the second order statistics of $\mathbf{x}$ (k+1—k) following a control input to be written as

$$\mathbf{P}_v(k+1|k) = \mathbf{J}_1(\mathbf{x}_v, \mathbf{u})\, \mathbf{P}(k|k)\, \mathbf{J}_1(\mathbf{x}_v, \mathbf{u})^T + \mathbf{J}_2(\mathbf{x}_v, \mathbf{u})\, \mathbf{U}\, \mathbf{J}_2(\mathbf{x}_v, \mathbf{u})^T$$

where the $(k|k)$ and $(k+1)$ indices have been dropped from $\mathbf{v}$ and $\mathbf{u}$ respectively for clarity and $\mathbf{U}$ is the covariance of the noise process in control u.

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

$$\mathbf{x}(k+1|k) = \begin{bmatrix} \mathbf{x}(k|k) \\ \mathbf{x}_v(k|k) \oplus \mathbf{u}(k+1) \end{bmatrix} \tag{5}$$

$$= \begin{bmatrix} \mathbf{x}_v(0|0) \\ \vdots \\ \mathbf{x}_v(k|k) \\ \mathbf{x}_v(k+1|k) \end{bmatrix} \tag{6}$$

The state vector is simply a vector of previous vehicle poses. Similarly the augmented covariance matrix $\mathbf{P}$ can be written as:

$$\mathbf{P}(k+1|k) = \begin{bmatrix} \mathbf{P}(k|k) & \mathbf{P}_{vp}(k+1|k) \\ \mathbf{P}_{vp}(k+1|k)^T & \mathbf{P}_v(k+1|k) \end{bmatrix} \tag{7}$$

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 (for example in our implementation around 50cm). This overall, compounded transformation becomes $u(k)$ and the k is incremented and the above described state project step undertaken. In this way the state vector grows linearly with the exploration path length and not with time.

The scan-matching part of the algorithm works as follows. Consider two poses at times i and j. Each pose has

[1] Each descriptor is a 128D vector and so we had little trouble in selecting a threshold that worked in a variety of scenarios, however the automatic setting of this threshold is a topic of research.

an associated laser scan L_i and L_j each containing n_i and n_j set of x, y points in the vehicle frame of reference. The scan-matching algorithm works on the assumption that there is a large overlap between the surfaces sampled in these two scans. It finds a transformation T parameterized by the vector $\mathbf{z}_{ij} = (x, y, \theta)$ such that

$$\eta = \sum_{k=1:n_j} \Phi(L_i, T(L_j^k, \mathbf{z}_{ij})) \tag{8}$$

is minimized. The function $\Phi(L_i, T(L_j^k, \mathbf{z}_{ij}))$ returns the unsigned distance between the k^{th} point in scan j transformed by $\mathbf{z}_{ij}$, and all of scan i. Note that we are not performing point to point associations as is common in ICP [2] like algorithms. In our implementation Φ uses the distance transform of L_i and uses the coordinates of the transformed points of L_j to look up 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 [22] 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:

$$\mathbf{x}_v(k+1|k) = \mathbf{x}_v(k|k) \oplus \mathbf{z}_{k,k+1} \tag{9}$$

There are several ways to use this observation. It could simply be stored and used (in linearized form) as an observation in a sparse bundle adjustment as proposed in [18]. 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 two pose 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).

B. Loop Closing

This choice of SLAM algorithm described in Section IV-A makes loop closing events particularly easy to handle. Imagine an oracle provides an observation $\mathbf{z}_{i,k}$ where $i << k$ — i.e. relates the current pose (end of state vector) to a pose dropped a long time ago (perhaps somewhere in the middle). This may well be a loop-closure event. All we need do to use the measurement is rewrite the measurement equation 9 in terms of pose states k and i and proceed as before with a standard EKF update ($\mathcal{O}(n^2)$).

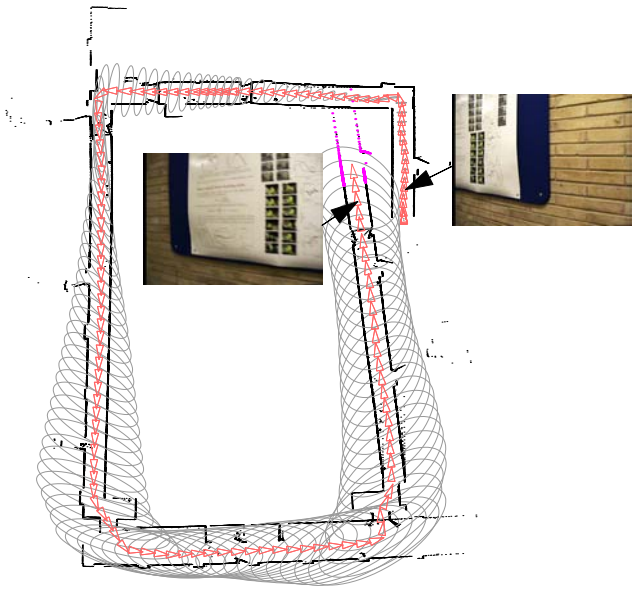

Fig. 3. A snapshot of the SLAM algorithm just before loop closing takes place. The vehicle poses stored in the state vector are shown in red. The performance of the SLAM algorithm is just as would be expected. Global uncertainty (gray ellipses) increases as the length of the excursion from the start 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.

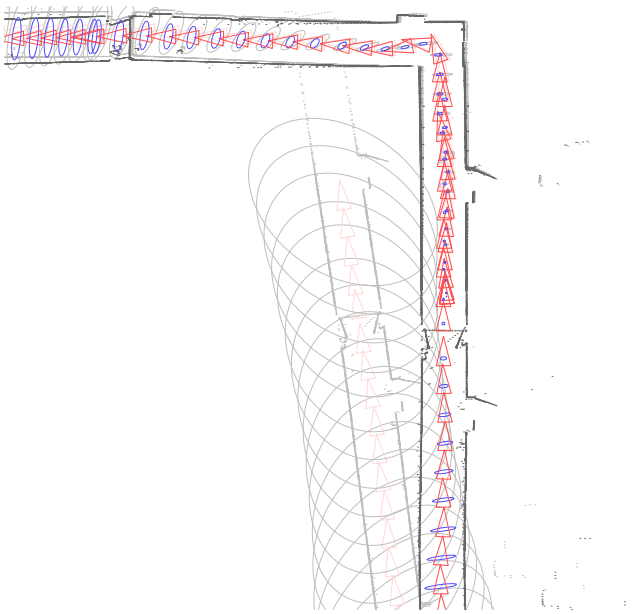

Fig. 4. A close up of the region close to the point of loop-closure. The pre-closure trajectory and uncertainties are shown in faint ink. Note how, as expected, the insertion of a loop-closing constraint between two poses that are temporally very distant causes a marked reduction in uncertainty (blue ellipses) in the recent poses.

Because the whole trajectory is stored in the state vector, all previous pose states will be adjusted in proportion to their uncertainty in order to accommodate the loop closure assertion.

The question now is how do we build such an oracle? We propose to use the database of visually salient features described in section II. If the current view from the camera is matched to a previous view, and we have confidence that the matching process is highly discriminative, as is certainly the case with the visual saliency scheme in use here, then it is highly likely that the vehicle is in the neighborhood of the earlier pose. The scan matcher can be run (with an initial seed zero transformation) to find the transformation between the two proximate poses.

The algorithm proceeds as follows. We do not look for loop closing between the most recent pose, k, and some historical pose. Rather we use a pose $q = k - n/2$ where n is some small number such that the set of recent poses and attached scans from $[(k-n) : k]$ represents a scan patch as proposed in [12]. If the query image at pose time q matches an image taken at the time associated with pose m, then another scan patch is produced around m. The query and match scan patches are described with respect to the pose frames q and m respectively. The motivation for the use of the patches - essentially simulating a multi-viewpoint scanner - is, as suggested in [12], to decrease the interpose ambiguity during the scan matching process. Finally the scan-matcher is run to produce a transformation between pose m and q. An estimate of the uncertainty in this match is derived by fitting a quadric to the error surface near the optimized transformation. From this quadric the hessian can be derived and hence a suitable covariance matrix.

C. Experimental Scenario

A small ATRV-Jnr mobile robot was driven around a building containing a large loop of around 100m length. We note that this is by no means a large loop or an extremely challenging environment for contemporary SLAM algorithms. However the accumulated spatial error is significant and serves to highlight the effectiveness of using salient, wide-base-line image patches to close loops without recourse to geometry.

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. The simple SLAM algorithm described in section IV-A was run using only the raw laser data and odometry. Figure 3 shows and describes the state of the algorithm just before the first loop closing event occurred. Figure 6 shows the system state just after the loop-closure constraint has been applied.

The top row of Figure 5 shows the correspondences found between the loop-closing images [2]. Note how most of the lines are parallel but not all. This apparent mismatching is a peculiarity in the scene — by chance, the images are of a poster which itself contains multiple, small highly self-similar pictures (of a remarkable luminous green, gloved hand). Nevertheless we remain confident that the probability of a false positive is low given the number of correspondences found. The lower two images in Figure 5

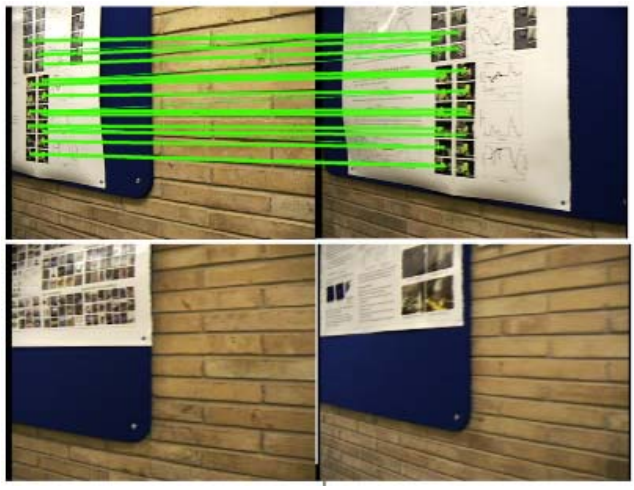

Fig. 5. View of the feature to feature correspondences found between the two images (top right and top left) used in the loop closing event shown in figure 3. The lower two images show similar images in the database that were successfully discriminated against.

[2]In our initial implementation it takes around one second to process a 640 x 480 image

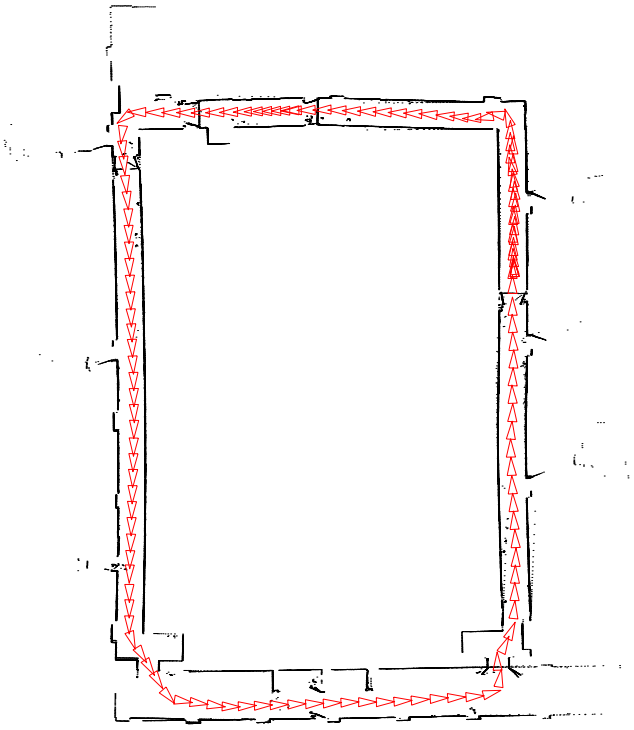

Fig. 6. A complete map of the test area just after loop closing

Fig. 7. Threshold setting is arbitrary (and unsatisfactory). Here the salient white plate is matched to a similar plate in a different location in a remarkably similar scene. This false positive was removed by requiring (for the results presented in this paper) at least 3 descriptor matches in an image.

show two other similar images in the database for which no correspondences were found. In all, the database comprised 250 images and the vehicle drove twice round the loop shown in Figure 6. In this initial test no false-positive matches were produced.

Finally, Figure 6 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. Although it is incidental to the focus of this paper, it is worth noting that the multiple-pose formulation used here does have the problem of not being able to refine the map over multiple passes without dropping more and more vehicle states. A pure feature-based approach does not have this issue, however it falls short of fully utilizing the richness of the laser data by limiting the map to a collection of often restrictive geometric primitives.

V. Conclusions, Issues and Future Work

This paper has presented some initial results concerning the use of salient image features in laser based SLAM work — in particular in detecting possible loop closure events in a manner which is independent of estimated vehicle pose. We suggest that this last point is central in achieving robustness — measurement decisions should be made independently of internal SLAM states.

While this initial fusion of ideas from the robotics and vision literature appears successful, we now identify and discuss some areas which should be either extended or improved upon.

Firstly, we use two image capture times to seed a scan-match correspondence search under the assumption that the two images were grabbed from spatially close poses. This is a reasonable assumption in the majority of indoor environments. However the scale invariance inherent in the saliency and SIFT feature descriptors means that in outdoor environments this assumption is not valid — the same feature occurring at markedly different scales in two images implies they were taken at very different locations. To remedy this we anticipate augmenting the image data base with local area laser scans to disambiguate scale.

Secondly, the setting of the number of feature matches required for a positive image to image association is arbitrary, and as shown in Figure 7, setting it too low can result in false positives. This is a catastrophe if it results in erroneous loop closing in a SLAM algorithm without an "undo" option. We are currently investigating ways in which this parameter could be learnt for a given workspace — initially with supervised learning. In a similar vein we are working towards anchoring the decision process in a probabilistic framework in which we calculate a probability of loop closure given previous images and a current view. We see this as an important goal.

We intend to take more of an active vision approach to camera control and use saliency detection to initialize a track on a region of space. This will allow active testing of wide-baseline visibility of image features *before* entering them in the visual database — again with the aim of increasing overall robustness.

We conclude that augmenting laser-based systems with vision systems can lead to a marked increase in performance. The techniques we have presented here are well suited for achieving robust loop closing – a key requirement in SLAM-enabled systems.

References

[1] K. O. Arras, J. A. Castellanos, M. Schilt, and R. Siegwart. Feature-based multi-hypothesis localization and tracking using geometric constraints. *Robotics and Autonomous Systems, 1056*, pages 1–13, 2003.

[2] P. J. Besl and H. D. Mckay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2), 1992.

[3] 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(12):1113–1139, Dec 2004.

[4] H. Choset and K. Nagatani. Topological simultaneous localization and mapping (slam): toward exact localization without explicit localization. *IEEE Transactions on Robotics and Automation*, 17(2), 2001.

[5] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms (Second Edition)*. MIT Press and McGraw-Hill, 2002.

[6] A. J. Davison and D. Murray. Simultaneous localisation and map-building using active vision. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002.

[7] M.W.M.G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (slam) problem. *IEEE Transactions on Robotics and Automation*, pages 229–241, 2001.

[8] A. Eliazar and R. Parr. DP-SLAM: Fast, robust simulataneous localizatoin and mapping without predetermined landmarks. *Proceedings of the Intl Joint Conf. on Artificial Intelligence (IJCAI-03). Acupulco*.

[9] 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.

[10] F. Fraundorfer. A map for mobile robots consisting of a 3d model with augmented salient image features. *26th Workshop of the Austrian Association for Pattern Recognition (AGM/AAPR) 2002*, pages 249–256, 2002.

[11] S. Gilles. *Robust Description and Matching of Images*. PhD thesis, University of Oxford, 1998.

[12] J. Gutmann and K. Konolige. Incremental mapping of large cyclic environment. *Proceedings of the Conference on Intelligent Robots and Applications (CIRA), Monterey, CA*, 1999.

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

[14] J. B. Hayet, C. Esteves, M. Devy, and F. Lerasle. Visual landmarks detection and recognition for mobile robot navigation. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR'03)*, 2003.

[15] E. Hygounenc, I-K. Jung, P. Soueres, and S. Lacroix. The autonomous blimp project at laas/cnrs: achievements in flight control and terrain mapping. *International Journal of Robotics Research*, 2003.

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

[17] J. Knight, A. Davison, and I. Reid. Towards constant time slam using postponement. *Proc. IEEE/RSJ Int'l Conf on Intelligent Robots and Systems, Maui, 29th October - 3rd November*, pages 406–412, 2001.

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

[19] J. J. Leonard, P. M. Newman, and R. J. Rikoski. Towards robust data association and feature modeling for concurrent mapping and localization. *Proceedings of the Tenth International Symposium on Robotics Research*, 2001.

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

[21] 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.

[22] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4(4):333–349, 1997.

[23] 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.

[24] P. F. McLauchlan. A batch/recursive algorithm for 3d scene reconstruction. *In International Conference on Computer Vision and Pattern Recognition, Hilton Head, SC, USA*, 2:738–743, 2000.

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

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

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

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

[29] 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.

[30] D. Ortin, J. M. M. Montiel, and A. Zisserman. Automated multisensor polyhedral model acquisition. *IEEE Transactions on Robotics and Automation*, pages 1007–1012, 2003.

[31] M. Paskin. Thin junction tree filters for simultaneous localization and mapping. *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pages 1157–1164, 2003.

[32] 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.

[33] R. Smith, M. Self, and P. Cheeseman. A stochastic map for uncertain spatial relationships. In *4th International Symposium on Robotics Research*, 1987.