A Framework for Vision Based Bearing Only 3D SLAM

P. Jensfelt, D. Kragic, J. Folkesson and M. Björkman
Centre for Autonomous System
Royal Institute of Technology
SE-100 44 Stockholm, Sweden
[patric,danik,johnf,celle]@nada.kth.se

Abstract— **This paper presents a framework for 3D vision based bearing only SLAM using a single camera, an interesting setup for many real applications due to its low cost. The focus in is on the management of the features to achieve real-time performance in extraction, matching and loop detection. For matching image features to map landmarks a modified, rotationally variant SIFT descriptor is used in combination with a Harris-Laplace detector. To reduce the complexity in the map estimation while maintaining matching performance only a few, high quality, image features are used for map landmarks. The rest of the features are used for matching.**

The framework has been combined with an EKF implementation for SLAM. Experiments performed in indoor environments are presented. These experiments demonstrate the validity and effectiveness of the approach. In particular they show how the robot is able to successfully match current image features to the map when revisiting an area.

I. Introduction

One key competence for a fully autonomous mobile robot system is the ability to build a map of the environment from sensor data and use it to localize. Natural landmark detection and incremental building of consistent maps for SLAM purposes have been a center point of robotic research for the last several years, [1], [2], [3], [4], [5]. For large scale and complex environments especially regarding full 3D, the problem is still an open research topic. Solving the SLAM problem with vision as the only external sensor is now the goal of much of the effort in the area [6], [7], [8], [9], [10]. Monocular vision is especially interesting as it offers a highly affordable solution in terms hardware.

In this paper, we present a framework for the management of visual features in SLAM. When designing this framework the following properties were desirable; i) produce few but stable landmarks, ii) robust matching of features, iii) means for finding the distance to landmarks for initialization in the map, iv) fast and robust detection of loop closing situations.

The first of these requirements is important when working for example in an EKF setting where the computational complexity grows quadratically with the number of features. We also want to keep the number of outliers low since most on-line SLAM methods handle such measurements poorly. The emphasize of the paper is not the actual estimation of the map but rather the management of the measurement to make such estimation possible. We show experimental results where the output from the framework is fed into an EKF SLAM implementation.

The main contributions in this paper are: a method for vision landmark initialization, the use of Harris-Laplace for feature detection together with a modified SIFT descriptor that is rotationally *variant* for robust data association and a representation that allows for fast and reliable matching between the current camera frame and landmarks in the map.

We distinguish between recognition features and location features. A single location feature will be associated with several recognition features. The recognition features' descriptors then give robustness to the match between the location features and the features in the current image. The location features are used to build a 3D SLAM map of the environment suitable for robot localization.

II. Related Work

Single camera SLAM is an instance of bearing only SLAM. Each image in itself does not contain enough information to determine the location of a certain landmark. Solving for the location requires that images from multiple view points are combined. This approach is similar to what in the computer vision society if referred to as the structure-from-motion problem (SFM). The major difference is that the SFM methods are commonly run off-line and consider batch processing of all the images acquired in the sequence while SLAM requires incremental and computationally tractable approaches suitable for on-line and real-time processing. Furthermore, the SFM methods do not assume feedback from information sources such as odometry that are commonly used in SLAM. The fact that a landmark cannot be initialized from a single frame means that a solution to bearing only SLAM must explicitly address this problem. Different solutions for initial state estimation in bearing only SLAM have been proposed.

A combination of bundle adjustment, commonly used in regular structure-from-motion approaches, and Kalman filter has been proposed in [11]. It has been shown that even if the method is less optimal than a regular Kalman filter approach, it gives better reconstruction results. In [8], a framework for vSLAM is presented based on a structure-from-motion approach from multiple views. It is mentioned that for cases when the robot performs only translational motion along the optical axis, the 3D triangulation is significantly uncertain due to very little or no disparity between matched features. Similarly to our approach, the reconstruction is performed using multiple images.

In [6], a particle filter approach is used to represent the unknown initial depth of features. In principle the initial distribution of particles would need to cover all possible range values for a feature. The convergence of the particle filters depends on the camera movements and may not occur at all. In [12] where the initial state is approximated using a Gaussian Sum Filter (GSM) for which computational load grows exponentially. In [13], an approximation to GSM approach is taken that performs undelayed initialization with an additive growth in the problem size. Features are extracted using a Harris point detector. This commonly leads to a high number of features, which is a problem for SLAM. According to [13] the features have to be (and are) pruned but no details of how this is done are provided.

Recently, a SLAM system using stereo vision and Rao-Blackwellised particle filter was presented in [9]. Visual features are detected using Difference-of-Gaussians and matched using SIFT descriptor. Two important problems were raised but not solved in this work. The first is related to the large number of detected features which make the approach inappropriate for large-scale and textured environments. One of the contributions of our work is that we deal with this problem by using a feature detector that gives raise to fewer features (presented in more detail in Section IV). In our work we keep the number of features low enough to allow for an EKF to be used in real-time. The second problem raised in [9] relates to the management and correspondence of SIFT features where matching is performed by one-to-all comparison. It is mentioned that better strategy would be to implement KD-trees using a limited number of features. This is the approach taken in our work which is an additional contribution. We use Harris corner features across different scales represented by a Laplacian pyramid for feature detection. For feature matching, we take a combination of a modified SIFT descriptor and a KD-tree.

Finally, as mentioned by [8], [9] and many others, feature matching becomes even more difficult when only a narrow field of view camera is used. We show that even in this case, our approach gives good matching results.

A single SIFT descriptor is not discriminative enough in itself to solve the data association problem. Especially in man-made environments where structures like corners give raise to many SIFT points with very similar descriptors. When used for object recognition [14] it is a combination of descriptors extracted from the object that provide the discriminative strength. This idea is used in the vSLAM approach [8] where the SIFT points are used to recognize places. In [15] "chunks" of SIFT points are used to present landmarks in an outdoor environment. In this work we let the position of the landmark be defined by a series of single modified SIFT points from different view points but let each such point be accompanied with a chunk of descriptor that make the matching robust.

Detecting that a loop has been closed is one of the more challenging problem in SLAM. In [16] a portion of the map of laser scans near the current robot pose is correlated with older parts of the map every few scans to detect loops. In [10] visually salient so called "maximally stable extremal regions" or MSERs are encoded using SIFT descriptors. Images are taken every few meters or seconds and compared to a database to detect loop closing events. As we will see later our framework also allows us to detected loop closing situations in an effective way.

III. Landmark Initialization

In this work we have adopted a delayed approach to SLAM, but we go about it differently then the above mentioned delayed approaches. The idea is to let the SLAM estimation lag behind *N* frames and use these *N* frames to i) determine which points make good landmarks and ii) find an estimate of their 3D location. This way the landmarks can be initialized with an estimate of the depth immediately in the SLAM process. This allows linearization directly without the need to apply multiple hypotheses [13] or particle filtering [6] techniques to estimate the depth. Landmarks for which the depth cannot be determined in *N* frames are not passed on to the SLAM process. Analysis over multiple frames makes it possible to determine landmark quality as well. Every time a new landmark is passed on to the SLAM module it is accompanied with an estimate of its depth which allows for it to be fully initialized directly. Figure 1 shows the information flow in the system.

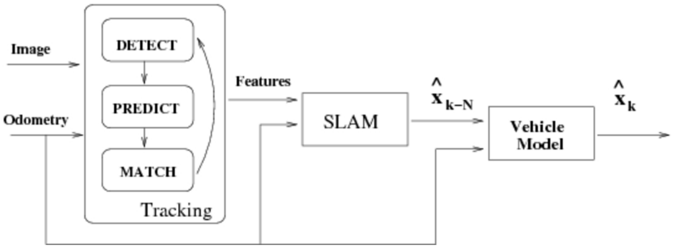

Fig. 1. The flow of data in the system. The image and odometry information is processed in the tracking module where matches are found between consecutive frames. The output is delayed *N* frames to the SLAM module. If an estimate of the current robot pose it desired one can be calculated by predicting forward the pose from the SLAM module using odometry or other dead-reckoning sensors.

As the input to the SLAM process is lagging *N* frames so is its output. In applications where an estimate of the current position of the robot is needed, for example for controlling the motion of the robot, we can use odometry and other dead-reckoning sensors to predict forward from the last pose estimated by SLAM. For typical values for *N*, the addition to the prediction error is small. This additional error is the price that we be pay for being able to initialize landmarks using bearing-only information and to be able to apply additional checks on feature quality.

The location features are those image features that are selected for use as map landmarks for SLAM. When selecting location features the following criteria are considered:

- Detection in more than some minimum number of frames
- Determination of the 3D landmark position by triangulation.
- The resulting 3D point is stable over time in the image.

The first requirement removes the noise and dynamic features. The second removes the features which the robot motion makes impractical. The baseline for the triangulation depends on the landmark location and the trajectory followed by the robot. The third requirement removes features that lack sharp positions in all images. Two reasons that a feature may lack sharp image positions are parallax or a lack of a strong maximum in scale space. The stability test is a threshold on the maximum perpendicular distance from the triangulated point to the bearings.

IV. Feature Description

In a recent study, Mikolajczyk and Schmid [17] analyzed a large number of interest point descriptors and their behaviors under changes, such as scale and illumination. The descriptor that turned out to be most robust in this study was the SIFT descriptor originally proposed in [14]. It was also concluded that the point detector used was less significant.

Primarily for reasons of low computational cost, the original version of the SIFT descriptor uses feature points determined by the peaks of a series of Difference of Gaussians on varying scales. In our implementation, we instead use the so called Harris-Laplace features, [18]. In the ordinary version of Harris corner detector, corners are detected in points where the product of the ellipse axis lengths defined by eigenvalues of the second order gradient matrix reaches a local maximum. Features detected this way are rotationally and translationally invariant. However, in SLAM applications, the scale of a point will change significantly and it is therefore important for the feature representation to be scale invariant.

Therefore peaks are found both spatially as well as in scale using Laplacian pyramids, thus making scale invariance possible. The reason for choosing Harris-Laplace was the fact that they respond to regions of high curvature, instead of blob-like image structures obtained by series of Difference of Gaussians. This leads to features more accurately localized spatially, which is essential when features are used for reconstruction and localization, instead of just recognition.

In a sparse, indoor environment many of the detected features come from corners features. The original SIFT descriptor assigns canonical orientation at the peak of smoothed gradient histogram. This means that similar corners but with a significant rotation difference may be matched to each other: a false match is shown on the left in Fig. 2[1]. To avoid this an extra matching step has to be added that compares the orientation of the two points. Given that the robot is assumed to move on a flat surface, which means that the camera undergoes planar motion, rotational invariance is not as important as in cases where a full camera pose change is present. In fact, it leads to a more difficult matching problem. Therefore, we have implemented a rotationally 'variant' SIFT descriptor where we avoid the canonical orientation at the peak of smoothed gradient histogram and leave the gradient histogram as it is. Another problem related to corner-like features is that it is difficult to estimate their dominant scale since they tend to be equally strong across different scales. Thus the location of these features in the images can move due to scale changes.

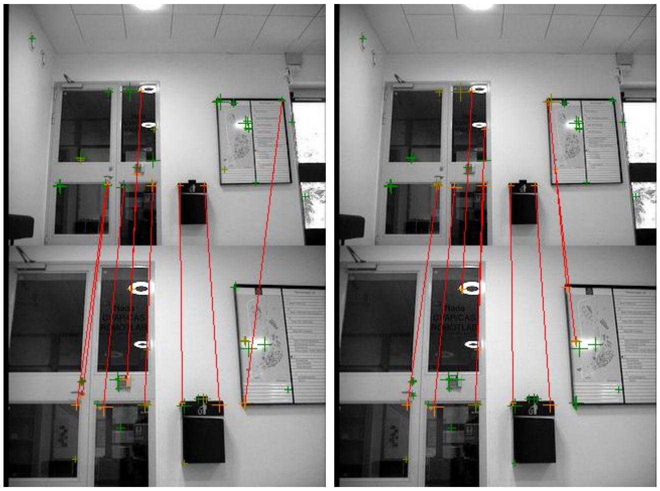

Fig. 2. Feature detection using Harris-Laplace approach and matching using left) orientation independent and right) orientation dependent SIFT descriptor.

V. Feature Tracking

We are now able to extract interest points from an image frame. The SIFT descriptor of a certain location changes as the robot moves through the environment. However, features will not move far between frames and the descriptors will not change significantly so we can track them and update the descriptor as we go along. This way each landmark/point in the world, p_i, will have a set of descriptors d_j associated with it. These descriptors describe the landmark from different vantage points.

A buffer with the points extracted from the last N frames is stored in memory. To manage the matching between frames, lists with associated points are maintained. Figure 3 shows the basic organization of this frame memory. Note that the association list on the right hand side ideally corresponds to different landmarks in the world. These lists can be analyzed to judge the quality of the corresponding landmark candidate. The output from the tracking module is a selection of the points in the oldest frame. The output consists of the points that correspond to already initialized landmarks or meet the criteria listed at the end of Section III for the first time in this frame and thus can be initialized.

A. Feature Motion Estimation and Matching

Provided that we have an estimate for how the camera has moved between frames (here we use odometry) we can predict the position of features in a new frame using standard optical

[1] In Figure 2 we have pruned features found at low scales (up to 4th order) to make the image less cluttered.

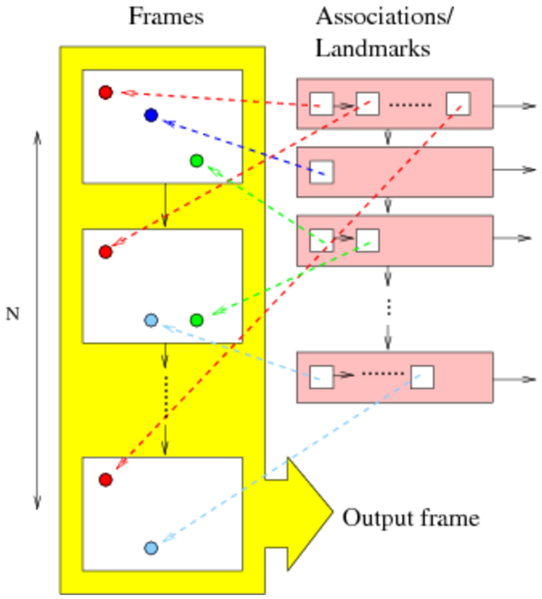

Fig. 3. A schematic view of the frame memory that stores the points in the N last frames and the associations between points.

flow equations. The prediction allows us to reduce the search windows which increases the speed of matching and can help eliminate some "impossible" matches.

Detected features that do not match the predicted positions of landmarks currently in the frame memory are matched against a database with initialized landmarks. This allows the detection of loop closing situations. As described in the next section this can be done fast and for the experiments we have not put any limitations on the database search. In a truly large scale setting one could direct the search by, for example, only querying the part of the database corresponding to the same floor of the building. The first time the location of a landmark has been established through triangulation and it has met the other criteria listed in Section III and thus passed on to SLAM estimation it is added to the database as a new location feature discussed further in Section VI.

VI. Database Management

The frame memory described in Section V deals with the tracking of points in the image. However, as soon as the camera turns away from a scene the points will drop out of the tracking window after not being seen for N frames. An important task in SLAM is to be able to detect when the robot is revisiting an area. For this purpose it is common to use a database of old points which can be searched for matches.

Here we let each landmark in our database have a set of descriptors that describe the landmark from different vantage points. Each descriptor describes the appearance of the landmark in some image. These different descriptors are provided by the frame memory that matches the landmark frame for frame. We add a new descriptor for a landmark when it is further than some threshold away from any of the other descriptors as measured in the descriptor space. Figure 4 shows the structure of the database where the landmarks are denoted with $F_1, F_2, \ldots, F_N$. The dashed box contains the descriptors for each of the landmarks. We use a KD-tree representation and a Best-Bin-First [19] search strategy for the data base. This allows very fast descriptor matching even with a very large number of points. In [20] 200,000 points are matched in real-time.

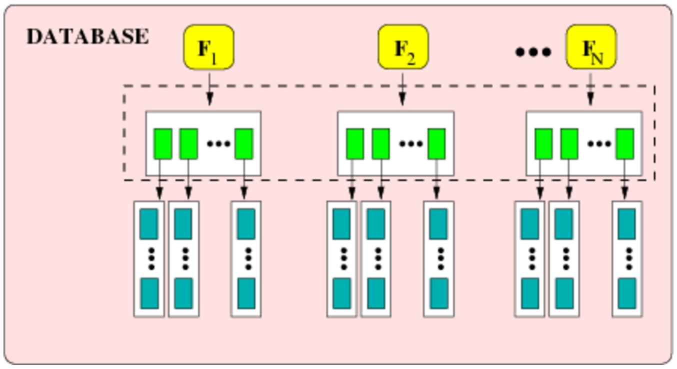

Fig. 4. Each landmark in the database has a set of descriptors that corresponds to location features seen from different vantage points. To validate a match, each of these descriptors keeps a list of the other descriptors found in the same frame. We refer to these as recognition descriptors. These provide the ability to "recognize" it again.

The SIFT descriptor is only locally unique. In man-made indoor environment there will be many structures with similar descriptors. For this reason it is not enough to match a single landmark descriptor against the database. However, by including all the descriptors from the scene where the descriptor in question was found the matching becomes very robust [8]. We refer to these other descriptors as recognition descriptors and the corresponding image features as recognition features. That is, when matching a feature to the database we first look for matches between its location descriptor and the descriptors in the database. Then, we verify the match using the corresponding two sets of recognition descriptors.

As a final confirmation for a database match we confirm that the displacement in image coordinates for the descriptor is consistent with the transformation between the two frames estimated from the matched recognition descriptors from above. The relative positions of the location and recognition features should be similar in the two frames. In the current implementation we simplify the calculation and only look at the 2D point displacement in the image. This final confirmation eliminates matches that are close in the environment and thus share recognition descriptors. As an example we observe that the SIFT descriptor for the upper left corner of the mail box in Figures 2 is very similar to the one for the upper left corner of the piece of glass to the left and they will have the similar recognition features when extracted from this frame.

To summarize the matching has the following steps

1) Match with the set of descriptors for the location features (dashed box in Figure 4) to get matching candidates.
2) Validate the candidates by matching using all extracted

descriptors from the current frame and the recognition descriptors associated with the location feature descriptor from the last step.

3) Confirm by checking the motion, in the images, of all the matches from the above two steps.

These steps have allowed very robust matching against the database as will be demonstrate in Section VIII.

VII. Simultaneous Localization and Mapping

The framework presented above handles the detection, selection, initialization and matching but not the position estimation. In this paper we use an EKF implementation for SLAM, however the framework can also be combined with other SLAM methods. Each point in the map is represented by its three Cartesian coordinates (x, y, z).

As in [6] we look for new features to add to the map only when we do not have enough previously initialized features in the current view. We divide the image into three regions. As long as there are at least two landmarks in a regions we do not initialize any new landmarks there. This reduces the number of landmarks and thus increases computational speed. The idea with the region is that we want to have a certain spread of the points so that we can handle motion in either direction without the risk of loosing all the points from one frame to the next as they the move in the image.

VIII. Experimental Evaluation

The experimental evaluation has been carried out on a PowerBot platform from ActivMedia. It has a non-holonomic differential drive base with two rear caster wheels. The camera used in the experiments is a Canon VC-C4 pan-tilt-zoom CCD camera. An image resolution of 320x240 pixels was used. Images are grabbed at 10Hz. To make sure that the images in the buffer in fact produce a baseline for triangulation a lower threshold for the camera movement between images are used. In the experiments images were discarded if the camera had not moved more than 3cm or turned more than 1°.

In the first series of tests we took the robot to an atrium with many possible loops. Figure 5 shows the map with the visual 3D landmarks marked with red (dark) boxes. A map built previously using a laser scanner is overlayed for comparison. The trajectory followed by the robot in the experiment is also shown in the figure. The total time for the experiments was roughly 9 minutes during which the robot completed 3 loops for a total of about 100m distance traveled. The time to process the data was about 7 minutes on a 1.8GHZ computer. This shows that we are able to run the system in real-time even though we are matching every unmatched features to the database in every used frame. In total 2611 images, roughly half of the acquired, were used.

The time to perform the tracking over frame has constant complexity. The matching to the database uses the KD-tree in the first step which makes this first step fast. This typically results in a few possible matching candidates. Depending on the scene, a typical frame has between 40-100 points. Out of these, as many as half often do not match any of the old features in the frame memory and are thus matched to the database. A typical landmark in the database has around 10 descriptors acquired from different viewing angles.

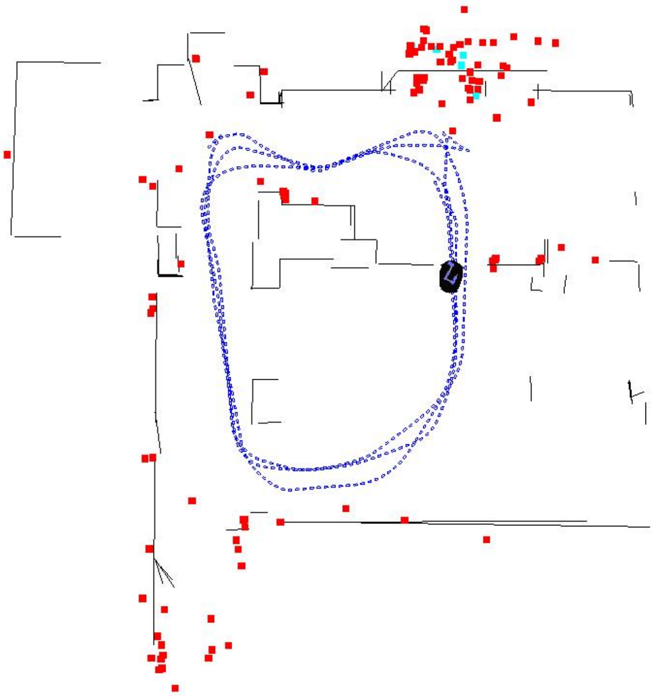

Fig. 5. Map built using the framework presented in this paper combined with an EKF implementation for SLAM. The red squares show the location of landmarks. Superimposed on this map is a map made previously with a laser scanner.

It can be seen that there are not so many features as typically seen in similar work using SIFT like features [9], [21]. This shows a clear benefit of using only the most stable point features for SLAM and the rest for recognition/matching as in our work.

From the figure it can also be seen that most features lie on walls or very close to one. Note also that some of the landmarks are detected fairly high up on the walls, in the ceiling or even on a lamp hanging down in the upper left region at the turn of the trajectory in Figure 5. The region in the upper right corner warrants some comments. The bottom row of images in Figure 6 show this part of the environment. There are many objects in this region at different depths. Also the walls shown in the 3D map are extensions of the 2D laser scans to constant height walls. Here the laser scan is at the height of the bench and not the wall making the points on the actual wall appear behind the virtual one. Some of the spread in depth for the landmarks is also due to not having seen the landmarks from different angles as the robot was always approaching this section straight on and not passing by it, producing a large baseline, like in other cases along the trajectory. This provides uncertain information about the depth. This is a problem often brought up in regular computer vision applications. The error is inversely proportional to the baseline and increases with the square of the depth [22].

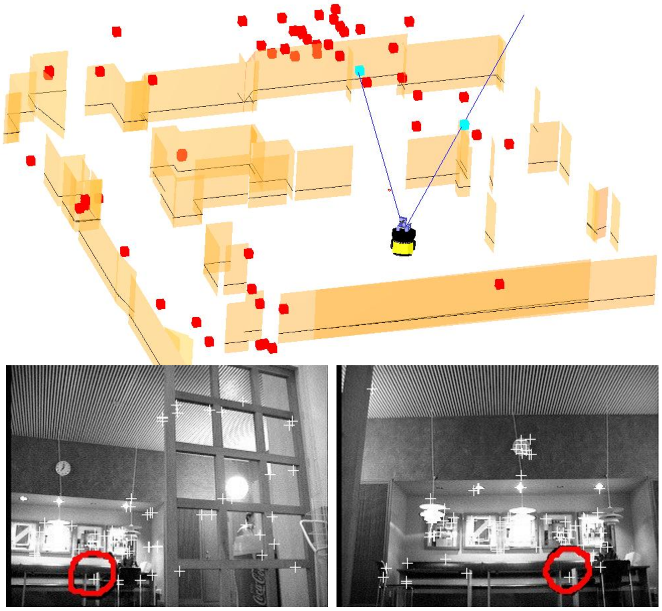

Fig. 6. The situation when the robot is closing the loop for the first time by re-observing a feature toward the back. The observed features are marked in cyan (light) in the upper part. The matched pair of features are circled in the lower two images. The image on the right is from the first time the robot was here.

Figure 6 shows the situation as the robot is just closing the loop for the first time by re-observing one of the earliest detected landmarks. The two lines protruding from the robot show the bearing vectors defined by the observations. It is the landmark furthest away from the robot, toward the wall in the back, which is re-observed. The two images in figure 6 show the image from the first time it was detected (right) and the image at which loop closing takes place (left). The landmark in question is marked with a red circle in the images[2].

Notice that we do not use the information from the SLAM process at all in the experiments when performing the matching with the database. We believe this is a great advantage and shows the robustness of the matching. Not relying on a position estimate to narrow down the matching means for example that this method could be used to perform global localization, that is, finding the robot position without prior knowledge which we plan to show in the future.

Figure 7 shows two other loop closing situations from two other experiments. The first one is from the atrium area as well and the second is from an office type environment

Summarizing the experiments, it has been shown that we are able to extract few and stable landmarks for SLAM using a modified SIFT descriptor and Harris-Laplace for feature detection. We have also shown that we can perform robust matching, initialize the position of landmark using a narrow field of view camera and perform loop closing detection in real-time.

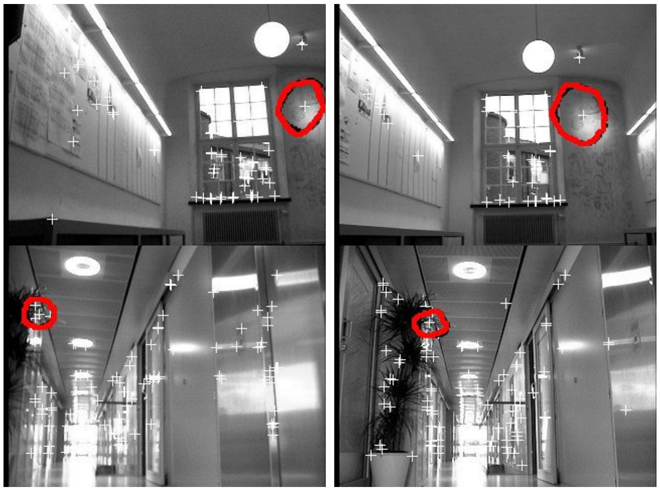

Fig. 7. Two other loop closing situations, one from the atrium area and one from an office type environment. After returning to an area the robot successfully matches landmarks from the previous time it was there. The circles show the matched pair of features. The right hand side shows the image from the data base.

It is worth pointing out once again that we are in essence able to detect previously visited areas in every frame in real-time by matching unmatched landmarks to the database. We perform this loop detection completely without any input from the SLAM estimation process or any other form of position information. In fact, our system could as well be used for place recognition completely separate from SLAM. This is a topic of future research.

IX. Conclusion

In this paper, we have presented a framework for selecting vision features for SLAM and for managing a database of them for robust matching. The robustness of matching is achieved by using many scale invariant features to define each match. Then having confirmed the match using many features we use only one of them for SLAM. The selection is based on its usefulness for SLAM as opposed to the usefulness for matching. Thus only well localized features are used for the SLAM estimation.

The features are extracted from the images using Harris-Laplace corner detection combined with a scale space maximization. This leads to rotationally variant features that have relatively well defined image locations. The rotational variant features are more suitable to SLAM problems where the camera is not rotating around its optical axis. Rotational invariance in this case just adds false matches.

The selection of the features to use for SLAM estimation is based on three criteria. First the persistence of the feature over many frames eliminates spurious and dynamic features. Second the motion of the robot must allow the range to the feature to be determined through triangulation. This will allow us to linearize the bearing measurements in our SLAM algorithm. The third criteria is the stability over time of the bearing to the feature. Thus the feature must give consistent bearings over a number of frames. Some vision features that

[2] Notice that the images are stored only for debugging and illustration purposes. All that is need to be stored for the actual processing is the descriptors extracted from the images.

lack good localization in the image will tend to drift about and not give good stability across frames. If the best fit triangulation point is too far from some bearing vector the feature is rejected.

We have validated our ideas using a robot moving in indoor environments. We have shown that we can build a map of points found using a single narrow field of view camera. We also showed that we could match the features again when returning to the same location by using the image information only.

References

[1] S. Thrun, D. Fox., and W. Burgard, "A probalistic approach to concurrent mapping and localization for mobile robots.," *Autonomous Robots*, vol. 5, pp. 253–271, 1998.

[2] J. A. Castellanos and J. D. Tardós, *Mobile Robot Localization and Map Building: A Multisensor Fusion Approach*. Kluwer Academic Publishers, 1999.

[3] M. G. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Corba, "A solution to the simultaneous localization and map building (slam) problem," *IEEE Transactions on Robotics and Automation*, vol. 17, pp. 229–241, June 2001.

[4] J. Tardós, J. Neira, P. Newman, and J. Leonard, "Robust mapping and localization in indoor environments using sonar data," *International Journal of Robotics Research,*, vol. 4, 2002.

[5] S. Thrun, Y. Liu, D. Koller, A. Ng, Z.Ghahramani, and H. Durrant-White, "Simultaneous localization and mapping with sparse extended information filters," *International Journal of Robotics Research*, vol. 23, no. 8, pp. 690–717, 2004.

[6] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *Proc. of the ICCV*, oct 2003.

[7] J. Folkesson, P. Jensfelt, and H. I. Christensen, "Vision slam in the measurement subspace," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA05)*, 2005.

[8] L. Goncavles, E. di Bernardo, D. Benson, M. Svedman, J. Ostrovski, N. Karlsson, and P. Pirjanian, "A visual front-end for simultaneous localization and mapping," in *Proc. IEEE International Conference on Robotics and Automation (ICRA 2005)*, pp. 44–49, apr 2005.

[9] R. Sim, P. Elinas, M. Griffin, and J. J. Little, "Vision-based slam using the rao-blackwellised particle filter," in *Proc. IJCAI Workshop on Reasoning with Uncertainty in Robotics, Edinburgh, Scotland*, July 2005.

[10] P. Newman and K. Ho, "SLAM-loop closing with visually salient features," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA'05)*, (Barcelona, Spain), pp. 644–651, Apr. 2005.

[11] M. Deans and M. Hebert, "Experimental comparison of techniques for localization and mapping using a bearings only sensor," in *Proc. of the ISER '00 Seventh International Symposium on Experimental Robotics*, Dec. 2000.

[12] N. M. Kwok, G. Dissanayake, and Q. Ha, "Bearing only SLAM using a SPRT based gaussian sum filter," in *Proc. of the IEEE International Conference on Robotics and Automation, (ICRA05)*, Apr. 2005.

[13] T. Lemaire, S. Lacroix, and J. Solà, "A practical 3D bearing-only SLAM algorithm," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'05)*, pp. 2757–2762, Aug. 2005.

[14] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. of the International Conference on Computer Vision (ICCV 1999)*, (Corfu, Greece), pp. 1150–57, Sept. 1999.

[15] R. H. Luke, J. M. Keller, M. Skubic, and S. Senger, "Acquiring and maintaining abstract landmark chunks for cognitive robot navigation," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'05)*, pp. 3770–3775, Aug. 2005.

[16] J. Gutmann and K. Konolige, "Incremental mapping of large cyclic environments," in *Proc. of the 1999 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, vol. 1, pp. 318–325, 1999.

[17] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition, CVPR03*, pp. 257–263, jun 2003.

[18] K. Mikolajczyk and C. Schmid, "Indexing based on scale invariant interest points," in *Proc. IEEE International Conference on Computer Vision (ICCV 01)*, pp. 525–531, jul 2001.

[19] J. S. Beis and D. G. Lowe, "Shape indexing using approximate nearest-neighbour search in high-dimensional spaces," in *Proc. of the Conference on Computer Vision and Pattern Recognition (CVPR'97)*, (Puerto Rico), pp. 1000–1006, June 1997.

[20] T. D. Barfoot, "Online visual motion estimation using FastSLAM with sift features," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'05)*, pp. 3076–3082, Aug. 2005.

[21] S. Se, D. G. Lowe, and J. Little, "Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks," *International Journal of Robotics Research*, vol. 21, no. 8, pp. 735–58, 2002.

[22] R. Hartley and A. Zisserman, eds., *Multiple View Geometry in Computer Vision*. New York, NY: Cambridge University Press, 2000.