SIFT Based Graphical SLAM on a Packbot

John Folkesson, Henrik Christensen

SIFT Based Graphical SLAM on a Packbot

John Folkesson
Dept. of Mech. Engineering
Massachusetts Institute of Technology
Cambridge, MA 02139
Email: johnfolk@mit.edu

Henrik Christensen
Center for Robotics and Intelligent Machines
Georgia Institute of Technology,
Atlanta, GA 30332
Email hic@cc.gatech.edu

Abstract

We present an implementation of Simultaneous Localization and Mapping ($SLAM$) that uses infrared (IR) camera images collected at 10 Hz from a Packbot robot. The Packbot has a number of challenging characteristics with regard to vision based SLAM. The robot travels on tracks which causes the odometry to be poor especially while turning. The IMU is of relatively low quality as well making the drift in the motion prediction greater than on conventional robots. In addition, the very low placement of the camera and its fixed orientation looking forward is not ideal for estimating motion from the images. Several novel ideas are tested here. Harris corners are extracted from every 5^{th} frame and used as image features for our SLAM. Scale Invariant Feature Transform, $SIFT$, descriptors are formed from each of these. These are used to match image features over these 5 frame intervals. Lucas-Kanade tracking is done to find corresponding pixels in the frames between the SIFT frames. This allows a substantial computational savings over doing SIFT matching every frame. The epipolar constraints between all these matches that are implied by the dead-reckoning are used to further test the matches and eliminate poor features. Finally, the features are initialized on the map at once using an inverse depth parameterization which eliminates the delay in initialization of the 3D point features.

I. Introduction

The goal of much research over the last decade has been to give robots a sense of location and direction comparable to that of humans. Humans have very limited sense of their ego motion with eyes closed. With eyes opened one can move with much more confidence. Judgements of distance and orientation are greatly improved. Comparable automatic robot vision capabilities have not yet been demonstrated. Much progress has been made [1], [2], [3], [4], [5],[6].

The problem has been given the name of vision based simultaneous localization and mapping SLAM. An issue in making maps from features seen in images is the lack of depth information in a single image [7]. Most of the SLAM methodologies estimate a maximum likelihood solution given the motion and feature measurements. This estimate typically assumes that the map and robot pose can be described by Gaussian distributions. Features seen in a single camera image correspond to real 3D objects. The probability distribution of such an object's location given the image has the shape of an infinite cone in Cartesian 3 space. This causes considerable difficulty for SLAM methods that use xyz as the feature representation. This problem has been cleverly solved [8] by utilizing a representation in terms of the bearing from the pose of the camera at the first observation and the inverse depth in that frame. The 6 parameters of the camera pose for this frame also become part of the parameterization. The result is a 9 parameter representation of the 3D point whose distribution can be well approximated by a Gaussian.

Another issue in vision based SLAM is the extraction of pixel level features from images [9], [10] and matching these with images of the same objects from different vantage points. Scale Invariant Feature Transform, SIFT, features have been use for SLAM [11], [12]. SIFT features have relatively weak descriptors associated with them. These descriptors have the advantage of invariance with respect to scale. They are also fairly similar for changes in viewing angle up to about 15-20 degrees. The descriptors alone can not be used to match features however and must be combined with other evidence such as epipolar constraints or the relative locations of other matching points.

II. Packbot Robot

Figure 1 shows the Packbot robot. This robot is designed to reach inaccessible places. It can travel up and down stairs, on its back and even withstand drops of several meters. It communicates over a wireless Ethernet connection.

To help it stay localized, it has GPS, an inexpensive IMU and encoders on its two tracks. The GPS is of no value indoors and the robot is meant to be used in GPS deprived areas. The encoders on the tracks give reasonably good information on distance traveled while the robot moves straight on level ground with good traction. Obviously there is much error when slippage occurs on turns and poor surfaces. The IMU helps reduce this somewhat but can not correct all of the errors.

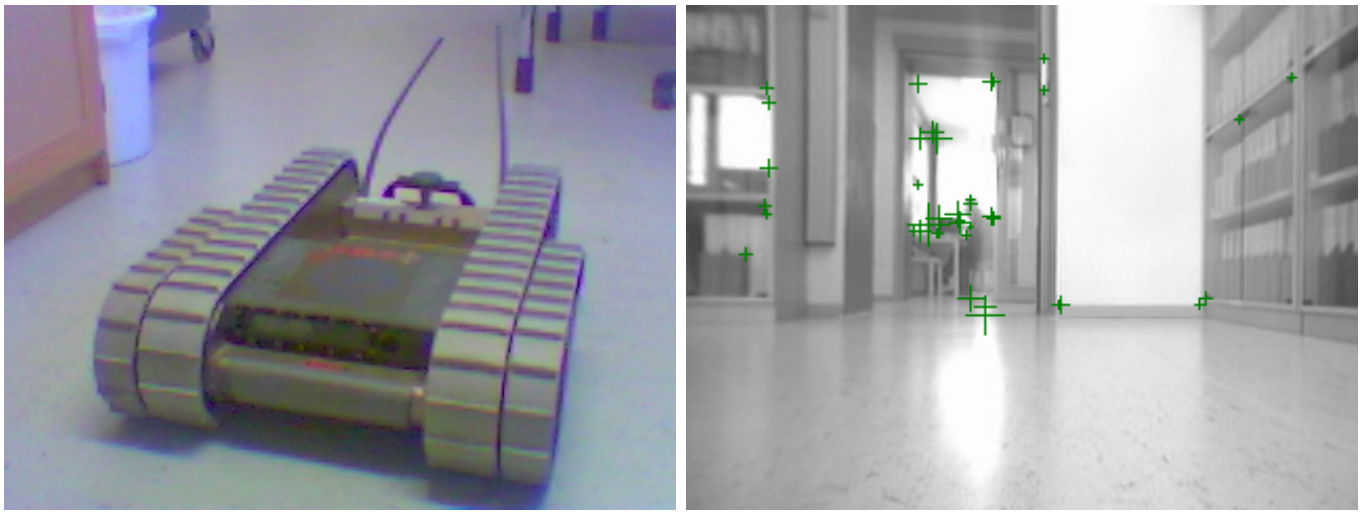

Fig. 1. The Packbot robot has a rectangular window in the front. Behind this sits an IR camera and a wide angle camera. Left is an example of one of the IR images from the Packbot. The green + signs are detected Harris corners. The size of theses + signs indicates the scale of the corresponding SIFT descriptor.

The robot is also equipped with two cameras, one a wide angle camera and the other an IR camera. We have used the IR camera to improve the motion estimate by doing SLAM on Harris corners extracted from the images matched using SIFT descriptors. Figure 1 shows a typical image from the IR camera. As can be seen in this image Harris corners in the image do not always correspond to well defined objects in 3 dimensional space. Notice the reflection of the window in the glass door of the bookcase on the left of the image. Also notice the intersection of objects inside the far room with the edge of the doorway that give rise to Harris corner image features. Distinguishing these type of 2D image features from ones that are better defined in the real 3D world is the challenge.

III. Graphical SLAM method

The Graphical SLAM method has been presented in [13] and [14]. We refer to those papers for the mathematical details. The measurements of the motion over some interval of time from the encoders and the IMU give us some probability distribution over the robot poses at the beginning and end of the interval. This is the probability of the measurements given the state (times a normalization factor that does not depend on the state). The negative log of this is the energy contribution of the dead reckoning over the interval. Thus the tradeoff between different measurements of the motion can be made based on a minimization principle on a graph of robot pose nodes.

The camera images give us addition information on the motion and additional states corresponding to features seen in the images. These additional feature states become feature nodes in our graph. The measurements of them by the camera create energy terms relating them to the robot poses. So that the graph consists of state nodes for the 6 dof[1] robot poses at the instant in time of the camera frames and the feature nodes of the 3D points that gave rise to image features in those frames. The connections between these state nodes is through energy nodes that correspond to measurements. The energy nodes can be connected to one or more state node[2].

The main advantage of working with this graph construction is that information can be gathered more flexibly and the linearization errors are greatly reduced. The difference is that we do not marginalize out pose states from

[1]degrees of freedom: $(x, y, z, \theta, \phi, \psi)$.

[2]Gravity or compass measurements give rise to energy nodes attached to only one state node.

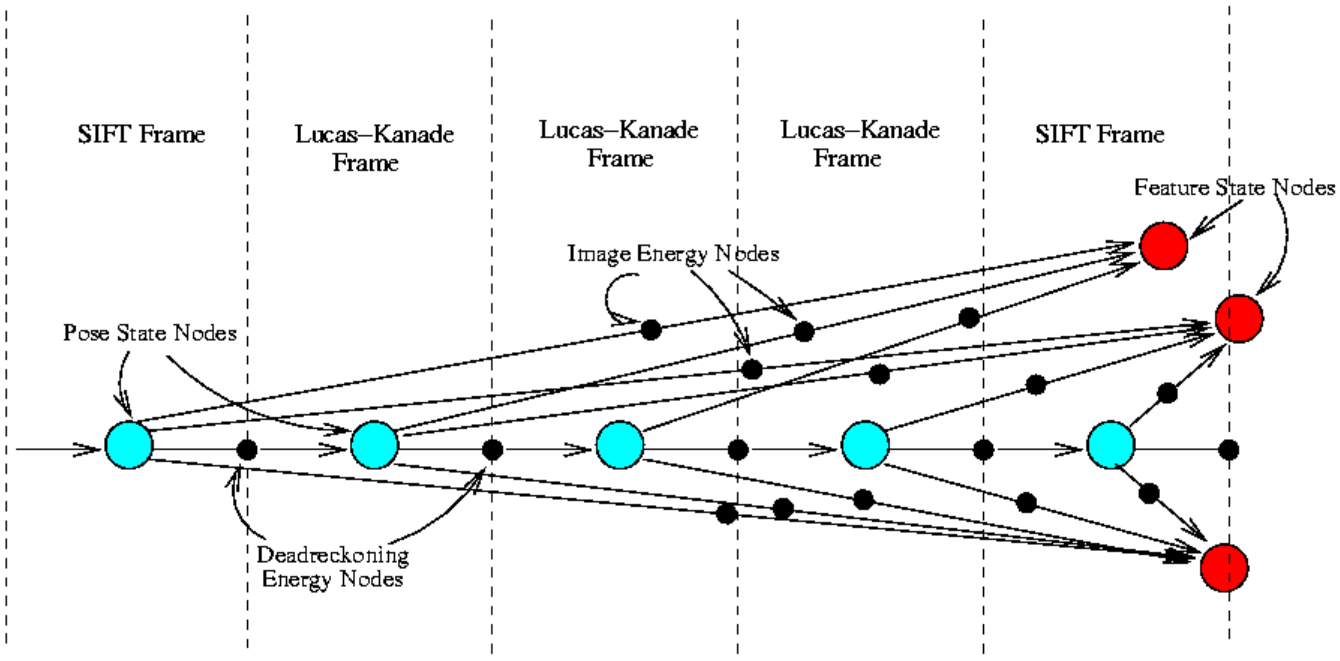

Fig. 2. Here we show a 5 frame segment of the graph. There are 3 Lucas-Kanade frames in this illustration. We used 4 such frames in our experiments.

the system. Having the pose states allows one to add and remove measurements from the solution. One can change data associations after testing how the energy would be changed. As a general principle one can say that it is easier to find representations that fit well to Gaussian distributions by increasing the dimension of the state space. This is true for the robot pose and, as we will see, for the feature part of the state as well.

The down side of increasing the dimensions is higher computation cost. In fact the full graphical solution gets unmanageable very quickly. However two escapes can be found. First the growth of the graph by adding more measurements does not change the solution for states far removed from the added measurements. That is unless the measurements include some new global constraints such as closing of a loop. So as long as the new measurements are not matched to older features one can simple relax the end of the graph, a constant time update. This leaves out the global localization problem completely. One is concerned with local accuracy only and the global solution is not important.

The other escape is to marginalize out part of the state from mature sections of the graph. This will allow global matching with reasonable complexity while maintaining local accuracy of the high dimensional representation. This is done by forming what we call star nodes. Star nodes are simple an energy node that replaces a set of energy nodes with a linear approximation of their energies. Having linearized the energy, one can then marginalize out the pose nodes that only connect to this one star node. By doing this in a local frame one obtains invariance with respect to large transformations of sections of the map. Thus once can still rotate the solution by say 20 degrees in order to close a loop without invalidating the linearizations.

One thus maintains the full state for the part of the graph connected to currently observed features. Then for the older section one forms a star node that marginalizes out a section of the robot path. The star node is allowed to grow until it has eliminated a predetermined number of pose nodes or has a maximum number of features. Then a new star node is begun.

IV. The Feature Tracking

The SLAM algorithm is only one part of an implementation. The results will also depend on the tracking/matching of features in the sequence of image frames. Harris corners have an advantage over other image features[3] in that they have well defined pixel locations in the image. Those locations may or may not correspond to well defined 3D locations in the real world.

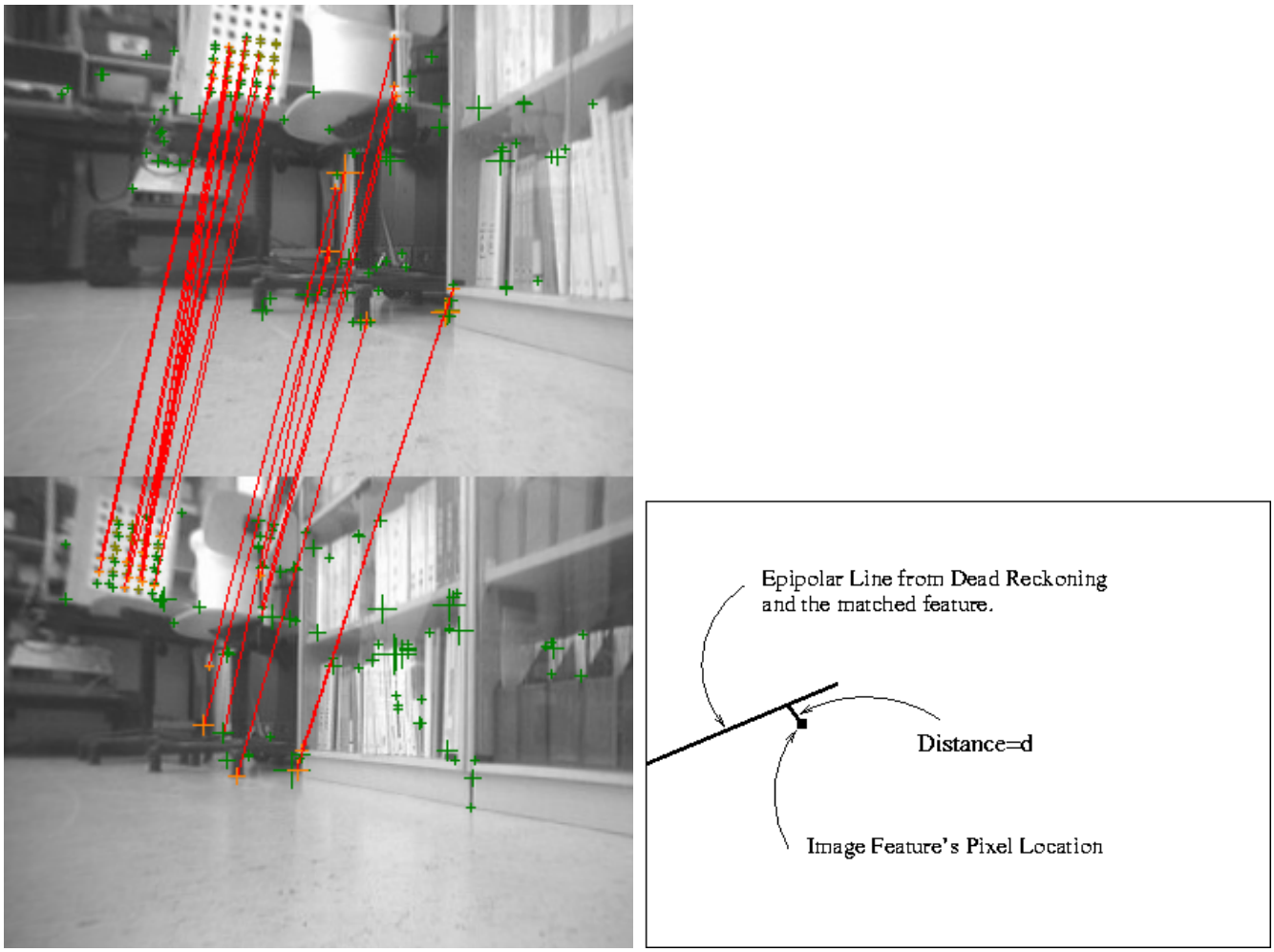

Fig. 3. The left is one example of poor SIFT matching between frames. Here the robot motion between the images leads to significant change in the images. The nature of the scene is leading to similar SIFT descriptors for different parts of the image. Still many of the points match correctly. Notice however that the calibration grid that happens to be in the frame is in fact confusing the SIFT descriptor matching. The repetitive pattern is the problem. On the right we show the geometry of the constraint energy nodes. We have two bearings, $\mathbf{b}_1$ and $\mathbf{b}_2$ to the point feature and an estimate of $\mathbf{Dx}$ the transformation between the camera frames. The energy of this constraint would be $\theta^2/(2C)$, where C, the measurement variance in θ, is found from the errors in the bearing angles. The two ranges r_1 and r_2 are calculated to minimize d. In the case of poorly defined ranges as when the bearings are in the $\mathbf{Dx}$ direction we simply use a default distance of 5 m for r_1.

The matching of the SIFT descriptors normally gives a good indication of pixel level correspondences between frames. The difficulty in this implementation was that it took about 100 msec per image to extract the Harris corners and calculate the SIFT descriptors for them. We would like to use a 10 Hz frame rate as the images can change rapidly as the robot moves about. We can only hope to achieve cpu utilizations of around 50-70% in the real time application. Add to this the need to do SLAM calculations and one realizes that SIFT descriptors can not be extracted from every frame.

[3]such as for instance, difference of Gaussians.

The solution we chose was to do Harris/SIFT extraction on every 5^{th} frame. Then for the intervening frames we interpolate the SIFT matches on each side using Lucas-Kanade tracking [15]. This tracking maximizes the correlation to a templet of the pixels around our feature to achieve sub pixel tracking accuracy. When the tracking failed to join the two bordering SIFT points smoothly the tracked pixel matches were not used. In this way we could reduce the utilization by a factor of 5 giving us a margin for the other computations that needed to be done.

Having tracked an image feature over 6 frames we apply the epipolar constraint to the motion in the image sequence. This we do in by utilizing our graph of pose nodes. We form temporary constraint energy nodes connecting two poses. The energy of these constraint nodes is defined in figure 3. There we have shown the two camera frames as having no relative rotation. The actual deadreckoning estimate is for the full 6 dof transform between frames including any rotation. This is then used to find the constraint. One advantage of this definition of the constraint energy is that the epipolar constraint can be applied for features seen in the direction of the motion. Such features have no epipolar line but the constraint energy still makes sense.

These constraints are one dimensional and three of them can specify the rotation beween two frames given the translation. The scale of the the translation can not be found but the energy is a function of the translation direction as well. We can differentiate the constraint energy with respect the pose states. This will give us a gradient and hessian to relax the graph with.

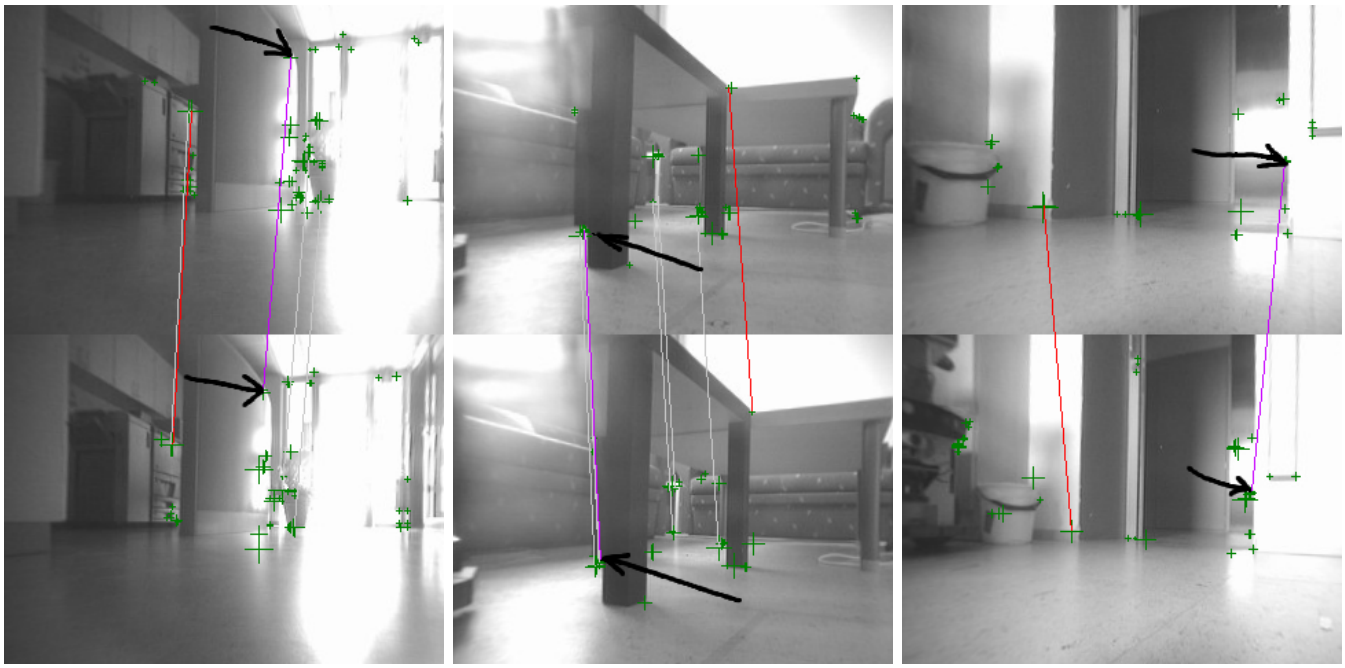

Fig. 4. Here the dark (red) lines connect the matches that became fully initialized Cartesian points. The light gray lines connect matches that were used with the inverse depth parameterization to create constraints between frames. The purple lines show features that have been identified as 'bad' (indicted also by the arrows). These image features were formed by two objects at differing depths or reflected light on a curved wall (furthest left).

We add such constraint nodes for all the new correspondences that we found by SIFT matching and Lucas-Kanade tracking. Then we relax the pose states attached to these constraint nodes. The resulting energy of each constraint is then tested. If any are above some threshold we mark the image feature as bad and remove all its energy nodes. Image features marked as bad will be kept to be matched to again and any future image features that match to bad features will not be used. It is hoped that this will eliminate some of the occlusion type features, see figure 4. We then relax the graph again until no constraint energy is over the threshold.

Having tested the epipolar constraints we remove the temporary constraint nodes. By modeling the 3D points that gave rise to the image features we can form two dimensional constraints between each pose and feature node. These are then stronger constraints than the epipolar one dimensional constraints between two pose nodes. They

require forming 3D point feature state nodes.

We do this and then add energy nodes to the 3D point feature nodes. These energy nodes have an energy based on the innovation in the predicted pixel locations in the images. We do this without any initialization delay by using a special inverse depth representation as explained in the next section. We accumulate the information on each feature in this way. If the standard deviation in the estimated depth becomes less than 30% of the depth we initialize a full Cartesian represented point feature. Only such features will be added to a global matching database. This global database entry will include the surrounding SIFT points as part of the context used in matching. This global database was formed but not matched to here. Global matching is the next stage of this research.

Figure 4 shows some examples of the constraints used to improve the odometry. In these example there were some initialized Cartesian points as well as a some inverse depth type constraints. There were also features labled as bad. These were image features that did not correspond to well defined 3D points.

The benefit of features to the motion estimate decreases rapidly after about 3 or 4 matches are found between frames. On the other hand, matching to poor features or miss matching can cause the estimate to worsen. We therefore only keep the best matches (those that have the lowest constraint energy) plus any matches from any initialized Cartesian points. When features were scarce in the images we relaxed our threshold to allow some matches.

V. The Feature Representation

We did not begin by implementing the inverse depth parameterized features. Instead we began by implementing a conventional Cartesian coordinated representation for the features. This meant that we had to wait to initialize the point features until we had enough lateral motion to triangulate the feature location into good Gaussian ellipsiods in the Cartesian feature space. For features that did not resolve the depth sufficiently we simply left the epipolar constraint nodes on the graph. These pairwise constraints between poses are weaker than the constraints of forming a 3D feature but they do not require any depth. The results were an improvement over dead-reckoning alone but not nearly as good as that which we obtained by switching to the inverse depth parameterization.

The idea of the inverse depth parameterization is to use the pose the feature was first seen from as part of its parameterization. Then one takes the bearing angles or pixel locations in that frame as two other parameters. This eliminates the problem of the ever widening cone of possible feature location that a Cartesian xyz representation would have. By using the bearing angles as parameters the covariance is a well defined ellipse in the bearing space.

For the ninth and last parameter of the features one chooses the inverse of the depth of the feature. The reason for this is that the depth itself could initially be anything from a few cm to infinity. This range will be transformed to a finite range for the inverse depth. The feature can then be initialized at once without needing to accumulate information on it. The distribution in this parameter space is well approximated by a Gaussian from the start.

This parameterization was rather easy to implement as we already had the pose as part of our state.

VI. Experiments

The experiments were carried out in out lab. We collected the images and telemetry data transmitted from the robot on a laptop computer connected via the wireless Ethernet. The images were directly saved in the 320X240 grey-scale jpg format. The first stage of the feature extraction was an undistortion of the images using the openCV library.

The first path was a simple circle around a table in the middle of the room. By carefully marking the floor we were able to return to exactly the same pose at the end of the test. This gave us a relatively easy data set to work with and the ground truth of returning to the same pose. Note that global matching of the SIFT features was not done so the program would not recognize the features from the end as being the same as those at the beginning. This global loop closing is future work and not done here. Our goal here is to try and minimize the drift of the estimate, a sort of visual odometry.

Figure 5 shows how the map looks in 3D just after the first few points have been added. The results are shown in figure 6. As you can see there were no major problems in this small test and the estimate is nearly prefect. In table I we summarize the error correction around the loop.

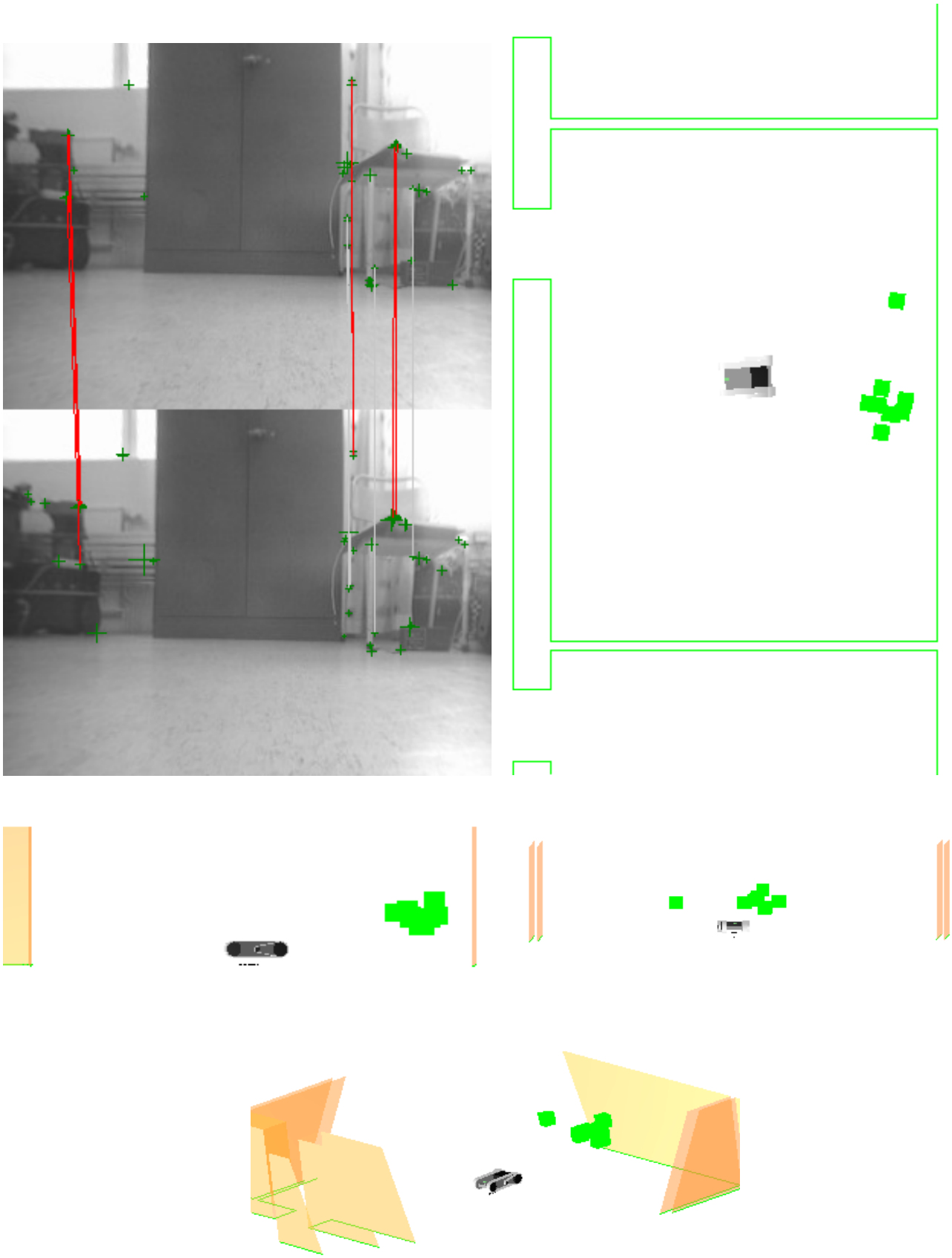

Fig. 5. Here we show how the matched image features get mapped in 3D. The early map is shown along with one of the early matches that shows some of the first features to be initialized. The map is shown from 4 angles top, side back and from a downward angle.

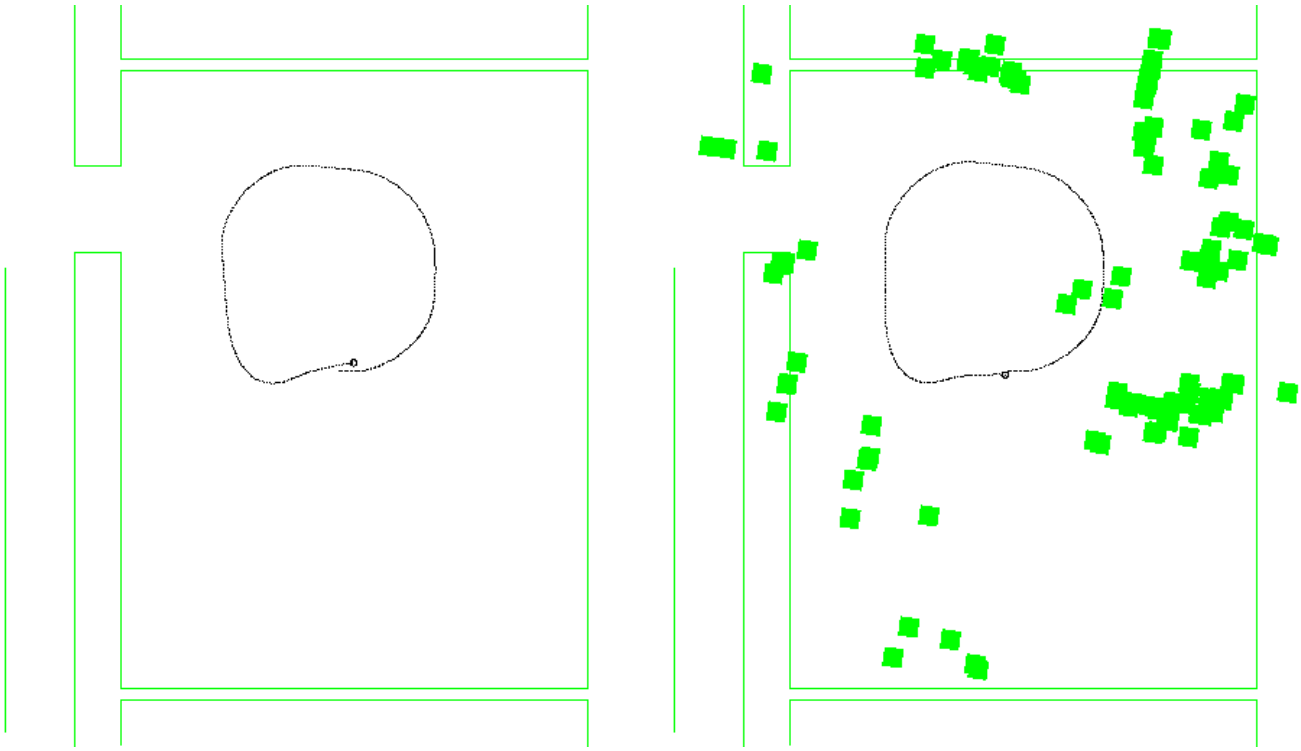

Fig. 6. We show on the left the dead-reckoning estimate of the robot path around a small loop. The corrected path and the map of point features (shown as squares) is shown on the right. The field of view is forward the whole time so the robot does not see the whole room as it moves. Also there is no attempt to match features seen at the start with those seen at the end.

Test	x (m)	y (m)	z (m)	θ (rads)	ϕ (rads)	ψ (rads)
True	0.000	0.000	0.085	-1.4300	0.0000	0.0000
Dead-reckoning	0.104	-0.148	0.085	-1.3088	-0.0012	-0.0012
Inverse Depth Features	0.042	0.001	0.087	-1.4531	-0.0006	-0.0008

TABLE I

THIS SHOWS THE ESTIMATED ENDING POSE FOR THE ROBOT AROUND THE SMALL LOOP. THE ENDING POSITION FOR THE DEAD-RECKONING IS OFF BY ABOUT 18 CM AND 0.13 RADS. BY USING THE IR CAMERA AND THE INVERSE DEPTH PARAMETERIZED FEATURES WE GET ABOUT 4 CM ERROR AND .02 RADS ERROR. THE ERROR IN POSITIONING THE ROBOT TO LINE UP WITH THE START POINT WAS LESS THAN 1 CM AND .01 RADS. 117 ENTRIES WERE MADE IN THE GLOBAL DATABASE, (INITIALIZED CARTESIAN POINTS) AND THERE WERE 1230 FRAMES.

The second test was a much longer path which lead out into the corridor and into two other rooms before returning to the exact starting location. The results are shown in figure 7. Here you see that the tuning of the filter had a significant impact on the results. A perfect loop closing without any global matching being done was not possible although we were able to get close on some occasions. The problem here as compared to the first case was not so much the length of the path but rather the variety of environments encountered. In some parts of the path there were simple no features found in the images. This a weakness that can not be overcome. In other cases occlusion type features could not be entirely filtered out of the data by the epipolar constraints.

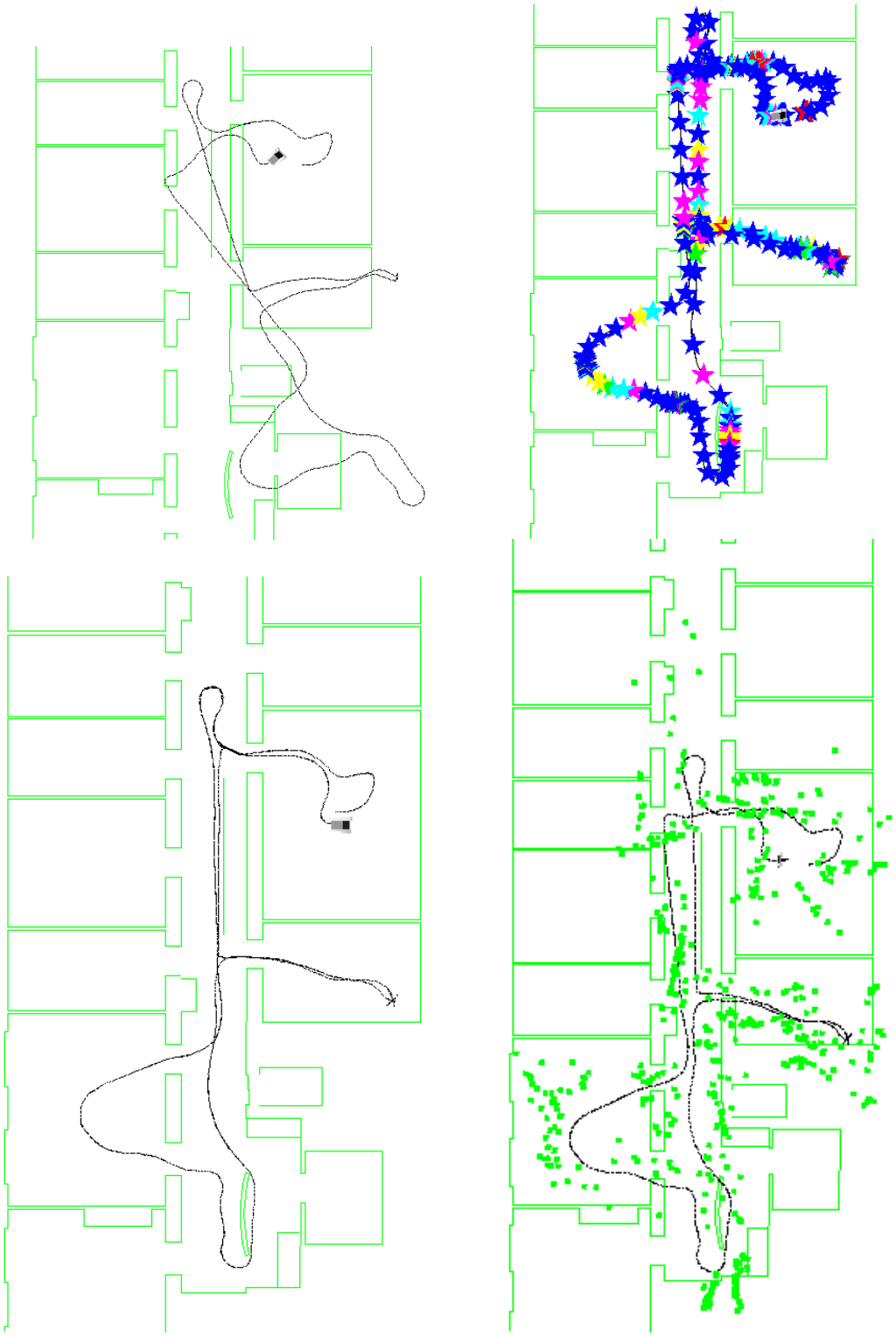

Fig. 7. On the top left we show the dead-reckoning estimate of the path for our second test. As one can see there are larger errors each time the robot turns. On the bottom left we show the best corrected path we were able to obtain using the image data. One can see the motion estimate can be rather good. This map was the result of excessive parameter tuning but represents the ideal result we would like to get more robustly. On the bottom right we have a more typical good result of the corrected path that one obtains from using the image data. There we show the features as well. Generally one always gets some improvement regardless of the tuning but the amount varies from a little better to very good depending on the tuning and the images. The top right shows another typical result where we have displayed the star nodes that form the local map patches in our graph. The two maps on the right have been rotated about the start position, (3 and 5 degrees), in order to line the average path up with the building. better. These maps had about 650 fully initialized Cartesian features and 8,600 frames.

VII. Conclusion

Despite some unavoidable short comings we saw a consistent improvement over dead-reckoning and for areas that had many good features the results were near perfect corrections. This should give us a good base on which to build global matching and constraints on. The SIFT descriptors are after all designed specifically for global matching and should facilitate the next phase of this research.

The results indicated clearly that the inverse depth parameterization does solve the problem of initialization features seen in a single camera frame. This allowed us to enforce two dimensional constraints on the motion between two frames as compared to the one dimensional epipolar constraint. That in turn lead to better visual odometry estimates.

The use of Harris corners in combination with SIFT descriptors gave mixed results. We would like to have better criteria for selecting features in the images that would give features that were not only salient in the images but also gave bearings to true well defined 3D points. Our tracking method based on the epipolar constraint energy nodes did eliminate most of the bad image features.

The fundamental problem we encountered was that of lack of any features at all in many of the images. This was a problem when the field of view was a blank wall or contained a bright window that blinded the camera. The only way to recover from such a situation is to recognize some part of the environment later on so that loop closing can be done to correct the path.

Acknowledgment

This work was supported by the Swedish defense agency FMV. This support is gratefully acknowledged.

References

[1] A. Davison, "Real-time simultaneous localization and mapping with a single camera," in *Proc. of the IEEE International Conference on Computer Vison*, 2003.

[2] A. Davison, Y. G. Cid, and N. Kita, "Real-time 3D SLAM with awide-angle vision," in *Intelligent Autonomous Vehicles (IAV-2004), M. I. Ribeiro and J. Santos-Victor, Eds., IFAC/EURON, IFAC/Elsevier*, 2004.

[3] D. Burschka and G. D. Hager, "V-GPS(SLAM): Vision-based inertial system for mobile robots," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA04)*, vol. 1, 2004, pp. 409–415.

[4] P. Newman, D. Cole, and K. Ho, "Outdoor SLAM using visual appearance and laser ranging," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA06)*, vol. 1, 2006, pp. 1180–1188.

[5] 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.

[6] P. Jensfelt, D. Kragic, J. Folkesson, and M. Björkman, "A framework for vision based bearing only 3D SLAM," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA'06)*, Orlando, FL, May 2006.

[7] J. Sola, A. Monin, M. Devy, and T. Lemaire, "Undelayed initialization in bearing only slam," in *Proc. of the IEEE/RJS International Conference on Intelligent Robots and Systems, (IROS 2005)*, vol. 1, 2005, pp. 2499–2504.

[8] J. Montiel, J. Civera, and A. J. Davison, "Unified inverse depth parametrization for monocular SLAM," in *Proc. of the Robotics Science and Systems Conference (RSS06)*, vol. 1, 2006.

[9] C. J. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. 4th Alvey Vision Conference*, Manchester, 1988, pp. 147–151.

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

[11] 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.

[12] T. D. Barfoot, "Online visual motion estimation using fastslam with sift features," in *Proc. of the IEEE International Conference on Intelligent Robots and Systems (IROS2005)*, vol. 1. IEEE/RSJ, 2005, pp. 579–585.

[13] J. Folkesson, P. Jensfelt, and H. I. Christensen, "Graphical SLAM using vision and the measurement subspace," in *Proc. of the IEEE/RSJ International Conference on Intellegent Robots and Systems (IROS05)*, 2005.

[14] J. Folkesson and H. I. Christensen, "Graphical SLAM for outdoor applications," *To Appear: Journal of Field Service Robotics*, 2006.

[15] B. D. Lucas and T. Kanande, "An iterative image registration technique with and application to stereo vision," in *Proc. of the International Joint Conference on Artificial intelligence*, 1981.