Place Recognition-based Fixed-Lag Smoothing for Environments with Unreliable GPS

Roozbeh Mottaghi, Michael Kaess, Ananth Ranganathan, Richard Roberts and Frank Dellaert
College of Computing, Georgia Institute of Technology
{roozbehm, kaess, ananth, richard, dellaert}@cc.gatech.edu

***Abstract*— Pose estimation of outdoor robots presents some distinct challenges due to the various uncertainties in the robot sensing and action. In particular, global positioning sensors of outdoor robots do not always work perfectly, causing large drift in the location estimate of the robot. To overcome this common problem, we propose a new approach for global localization using place recognition. First, we learn the location of some arbitrary key places using odometry measurements and GPS measurements only at the start and the end of the robot trajectory. In subsequent runs, when the robot perceives a key place, our fixed-lag smoother fuses odometry measurements with the relative location to the key place to improve its pose estimate. Outdoor mobile robot experiments show that place recognition measurements significantly improve the estimate of the smoother in the absence of GPS measurements.**

I. INTRODUCTION

A popular approach for pose estimation of robots is place recognition. For instance, GPS-equipped outdoor robots which traverse rough terrain usually suffer from bad localization. If a robot is asked to move on the same path and repeat the task it previously did, it will not usually be able to do so due to pose inaccuracies. One useful application of place recognition is that the robot can determine its offset from the correct path and adjust its position accordingly. A similar scenario involves a robot sharing its own map with a convoy of robots on the same terrain. Place recognition can also help a robot in path planning. Consider a robot that plans a path from a starting point to a goal. If the robot traverses a loop while exploring the terrain and recognizes the closing point of the loop, the next time it would not plan any path through the loop, minimizing the time that is needed to reach the goal.

There are some factors that make the task of place recognition challenging. Finding an exact match for a previously visited location using a robot is not trivial since the sensors are noisy and the environment may also change. In addition, changes in the robot's plan or the uncertainties in the robot's motion often result in visiting a place from a different angle, which makes the task of place recognition even more difficult. In this paper, we propose a novel solution that tackles some of these issues.

There is extensive literature on place and scene recognition methods, some of which we briefly mention here. Torralba et al. [1] present a place recognition method for a navigation system that uses Hidden Markov Models and a low-dimensional global representation of an image to recognize a scene and perform localization. They also consider the result of place recognition as a good prior for object recognition in an environment. Mozos et al. [2] describe a method that uses semantic recognition of places in an office environment and use that method to localize an indoor mobile robot.

Some authors have tackled the more challenging task of outdoor place recognition for robots. Morita et al. [3] apply a support vector machine to learn image features in key images and perform localization using learning-based frame matching. Bradley et al. [4] propose another example of outdoor topological localization based on place recognition. In [5], a database of panoramic images is built and a pose is assigned to each image according to GPS. Unlike topological methods, their work can approximate the orientation and location of the robot with respect to the key place.

The major drawback of these methods is that their results are either high-level and inaccurate or they assume perfect localization during the learning phase of key images. For example, the topological approaches can only recognize that the robot is inside a particular room but they cannot give accurate information about the exact location of the robot.

In this paper, we investigate a location estimation method for an outdoor robot using place recognition. Our assumption is that there will be no GPS signal available and the robot should correct its belief about its location when it recognizes a place whose position has been already learned by the robot. In the rest of the paper, we describe the place recognition phase, localization using a fixed-lag smoother, followed by an example of a key place learning approach, and finally real robot results.

Recently, Royer et al. [6] have presented a similar approach to ours to localize an outdoor robot using a sequence of learned keyframes. Our method has a few advantages over theirs: we only store a sparse set of key frames. Additionally, our fixed-lag smoother uses the future information to correct the trajectory of the robot while they have to enter the length of the path manually to set their scale factor.

II. APPROACH

We focus on an application where we learn a set of key frames beforehand. The robot should recognize the key places when re-visiting that location by checking its database of stored images. The robot then incorporates the place recognition measurements into a fixed-lag smoother in order to localize itself. Our method of stereo matching for recognizing the key places is described below.

A. Place Recognition

Assuming we already have a database of key places, as obtained from the learning phase, the robot can correct its location estimate by recognizing when it gets near such key places. To recognize a key place, the current observations are compared with the database of key frames, which contains images and features for key places. The features that we use are stereo Harris features. When we capture a new image during place recognition, we find potentially matching features (putatives) between the current observation and the images that we have stored in the learning run, by using the sum of the squared differences between the image patches that we consider around each feature location. Then, we use RANSAC on the putatives to prune the outliers. If the proportion of the inliers is greater than some threshold, we conclude that the robot's current place matches the image in the database. It should be noted that each key frame consists of the left and right stereo images.

Finally, a rigid transformation is calculated between the current camera pose and the pose of the camera when it captured the image of the matched key place. By back-projecting the inlier features to the world frame, we find the relative location of each pair of matched features (inliers only), but the relative location is not unique for all of the inliers. By performing an optimization, we find a transformation that minimizes the error of the odometry for all pairs of inliers [8]. Since the described method is computationally demanding, we do not perform the matching procedure for every single image in the database, but only for the key places whose associated pose is close to the current location estimate. Fig. 1 shows the inlier features between the left image of one of the key frame images and the left image of the current observation of the robot.

As the result of this phase, we obtain the mean of the relative pose to a key place and a covariance matrix that captures the uncertainty of the place matching.

B. Pose Estimation Using a Fixed-Lag Smoother

Knowledge about the relative pose to a key place is used to improve the robot's pose estimate. We treat the result of place recognition as a new sensor that gives us a global position measurement for the robot. Our assumption is that the learning phase provides us with global information about the location of key places. And together with the relative location of the current position of the robot to the matched key place, we therefore obtain a global measurement.

At time t, we use a fixed-lag smoother similar to [9] to find the posterior for the last p robot poses (p is the length of the lag) given all of the global and relative measurements. The Bayes net in Fig. 2 represents the pose estimation problem where the relative locations in the network are conditioned both on the location of the robot and the location of key places. $X_p = x_{t-p+1:t}$ represents the set of the last p states

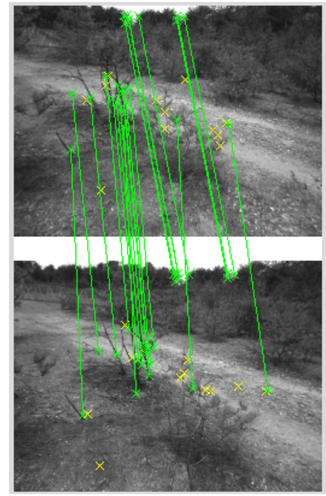

Fig. 1. A snapshot from the image matching process. Putatives are shown in yellow and the inliers are shown in green. There are 156 features in the key image. 35 putatives were found, of which 23 were identified as inliers.

namely, the robot poses. The set of states from the start location up to the lag is denoted by $X_f = x_{0:t-p}$.

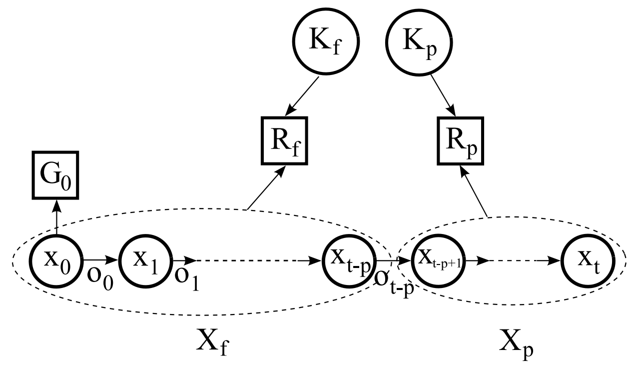

Fig. 2. A graphical model that represents our smoothing problem. x_t is the robot pose at time t, p is the length of the lag, o_t is the odometry reading, $\mathcal{K}_f$ and $\mathcal{K}_p$ are key places and R_f and R_p are the relative odometry to the matched keyplaces.

As the location of key places is only known within some uncertainty, we marginalize over the key place locations, obtaining an estimate X_p^* for the current lag states:

$$X_p^* = \arg\max_{X_p} \int_{X_f \mathcal{K}_f \mathcal{K}_p} P(X_f, X_p, \mathcal{K}_f, \mathcal{K}_p | G_0, R_f, R_p) \tag{1}$$

where R_f and R_p are the set of relative transformations obtained in the recognition phase before and in the lag, respectively. $\mathcal{K}_f$ and $\mathcal{K}_p$ are the set of key places visited before the current lag and in the lag, respectively. For simplicity, we assume that the keyplaces are independent variables. As before, we have assumed that there is no global position information available to the robot after leaving the start location and we only have one global measurement at the beginning (G_0, eg. GPS measurements). Using Bayes

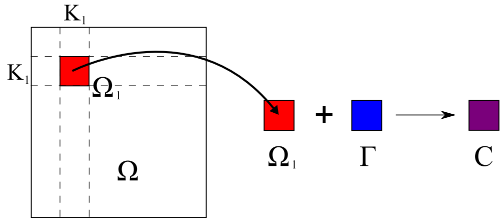

Fig. 3. Calculating a new covariance matrix for marginalizing over key place information. K_l is the matched key place, and Ω and Γ are covariance matrices that show the uncertainty in the location of key places and stereo matching, respectively.

rule, the integral of (1) is proportional to:

$$\int_{\mathcal{K}_p} P(R_p|X_p, \mathcal{K}_p)P(\mathcal{K}_p) \int_{x_{t-p}} P(x_{t-p+1:t}|x_{t-p}, o_{t-p:t-1}) \quad P(x_{t-p}|G_0, R_f, o_{1:t-p-1}). \tag{2}$$

Performing the optimization to find the estimates for the lag states requires solving the integrals. The solution to the integrals in (2) can be written as follows:

$$\prod_{i=t-p+1}^{t} P(x_i|x_{i-1}, o_{i-1}) \cdot \prod_{k} P(R_{j_k}|x_k, K_l)P(K_{l_k}) \quad P(G_0|x_0) \cdot P(x_{t-p}|G_0, R_f, o_{1:t-p-1}) \tag{3}$$

where R_j is a relative measurement to the l^{th} keyplace K_l. The first factor in (3) is the motion model, the second factor is obtained as the result of place recognition, the third factor $P(G_0|x_0)$ is a global location measurement which is conditioned on the initial pose, and the last factor is the prior on the state immediately before the fixed lag.

To convert the problem to a least square optimization we need the covariances for all of the factors in Eq. 3. The covariance matrices for the second product term are obtained as follows: In the learning phase, we calculate the joint covariance of all of the keyplaces. As mentioned before, we assume the key places are independent and we approximate the covariance matrix by a block diagonal matrix, where each block contains information about one of the key places. We also obtain a covariance matrix in a similar way from the place recognition phase which is denoted by Γ_j for the j^{th} measurement.

To calculate the covariance for the second term of (3), we add Γ_j to a block from Ω that is related to the matched place and is denoted by Ω_l for the l^{th} place, as shown in Fig. 3. The new covariance matrix is denoted by C_{lj} and is equal to $(\Omega_l + \Gamma_j)$. The mean of the measurement is equal to $K_l + R_j$. When we marginalize out the key places, the mean of the measurement is obtained by simply adding (considering rotations) the relative location of the robot to the key place and the pose of the key place [11]. Hence, we have introduced a new global location measurement with a specified mean and covariance, assuming that both factors in the second product term are normal distributions.

Now, the goal is to find an optimal estimate for states of the robot in the lag (unknowns) based on the different measurements that we have. As described in [10], the problem can be formulated as a non-linear least-square optimization problem. Since the terms in (3) are assumed to be normal distributions, our optimization problem from (1) is converted to the following error minimization problem:

$$X_p^* = \arg\min_{X_p} \{ \sum_{i=t-p+1}^{t} \|f_i - x_i\|^2_{\Lambda_i} + \|g_0(x_0) - z_0\|^2_{\Sigma} + \sum_k \|h_k(R_{j_k}, K_{l_k}) - x_k\|^2_{C_{lj}} + \|x_{t-p} - \mu_{t-p}\|^2_{P_{t-p}} \} \tag{4}$$

where we use $\|.\|_\Sigma$ to denote the Mahalanobis distance and $f_i = f(x_{i-1}, o_{i-1})$ is the function that predicts the next location of the robot based on the current location x_i and the odometry o_i, augmented with Gaussian noise with covariance Λ. Also, $r_k = h_k(R_{j_k}, K_{l_k}) + v_k$ is the term for the absolute measurement which is calculated according to the location of a key place and the relative location of the robot to that key place. v_k is the measurement noise for the place recognition phase and its covariance C_{lj} is calculated according to Fig. 3. Finally, $\mathcal{N}(\mu_{t-p}; P_{t-p})$ is a Gaussian prior for the robot's state at time $t-p$, denoted by x_{t-p}. The fixed-lag smoother adds a new state for each new measurement and considers the first state in the lag as the new prior state.

We linearize these functions by using the first terms of their Taylor expansion [10] and the problem becomes a linear least squares formulation, $J^*\delta X - b = 0$, where J^* is the measurement Jacobian matrix and is obtained by assembling the Jacobians of individual measurements (shown in Fig. 4), δX is our unknown (the p states in the lag) and b is the difference between the predictions and the real measurements.

We solve the estimation problem by QR-factorization of J^*, followed by back-substitution to obtain δX. Since the R factor is a sparse matrix, the state estimation for the lag can be computed very efficiently. The details of the fixed-lag smoother and handling out-of-sequence measurements can be found in [9].

III. Place Learning

In this section, we describe an example method for learning the key places and the joint covariance matrix. We assume that either a human driver or a more capable robot marks some arbitrary key places, or that they are extracted automatically as mentioned in [6]. Fig. 5 represents this idea when a robot traverses a path in a dense forest environment where there is no GPS signal available except at the beginning of the experiment and at the goal location. The small circles represent the special places that have been specified by the human driver.

Since the odometry measurements are subject to noise and other sources of uncertainty, we apply learning to obtain a better location estimate of the special places. Therefore, we find the location of a set of places that maximize the probability distribution:

$$P(\{K_1, K_2, ..., K_N\} | O_{0:T}, G_0, G_T) \tag{5}$$

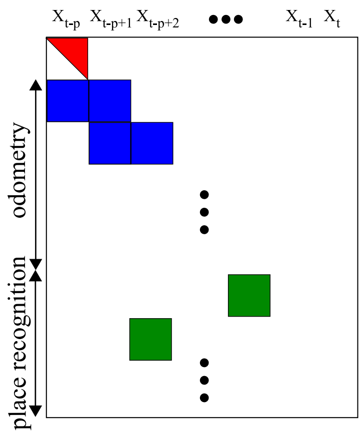

Fig. 4. The Jacobian matrix which is obtained by assembling the individual measurement Jacobians. The x_{t-p} block corresponds to the prior for the state before the lag as remembered from previous factorizations.

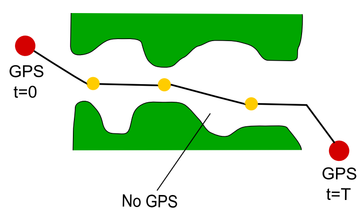

Fig. 5. Some key places are marked in the areas where no GPS is available. Larger circles represent the points at which GPS is available and the small circles are key places.

where N is the number of special places, $O_{0:T}$ is a set of odometry measurements, and G_0 and G_T are GPS measurements at the beginning and at the goal location, respectively.

We use a least-squares optimization technique to find a better estimate for the location of key places. A Gaussian noise model is assumed for the places so our distribution for each place is as follows: $P(K_i) = |2\pi\Sigma|^{-0.5} exp(-\frac{1}{2}\left\|K_i - \overline{K_i}\right\|^2_{\Sigma})$, where $\overline{K_i}$ is the inaccurate measurement of the robot. We formulate the problem as minimizing an energy function E, which is the logarithm of the joint distribution of the places. Since the logarithm is a monotonic function, minimizing the log function is equivalent to minimizing its argument. We do not have a sensor to measure the location of a place and the only measurements that we have are a set of odometry measurements which represent the relative location of two consecutive places. Therefore the energy function E is defined as:

$$\begin{aligned} E &= \|K_1 - G_0\|^2_{\Sigma_G} \\ &+ \sum_{i=1}^{N-1} \|odo_{i\rightarrow i+1} - (K_{i+1} \ominus K_i)\|^2_{\Sigma_O} \\ &+ \|K_N - G_T\|^2_{\Sigma_G} \end{aligned} \quad (6)$$

where Σ_O and Σ_G are the covariances that we consider for the odometry and GPS measurements, respectively and G_0 and G_T are our GPS readings at the beginning and at the goal location. We also assume that K_1 and K_N are the places for which we have a GPS measurement (both ends

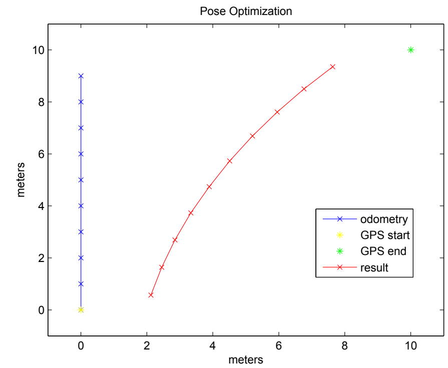

Fig. 6. Result of LM optimization in the case that GPS readings and odometry measurements do not coincide. Ten special places are shown by cross signs.

of the trajectory). Each key place K_i is a six-dimensional variable that is defined as $(x_i, y_i, z_i, \psi_i, \theta_i, \phi_i)$, which are the coordinates of a place in a global frame, with ψ, θ and ϕ representing the yaw, pitch and roll angles. Since the normalization constants do not have any effect on the minimization, we have removed them from the energy function. Also, $odo_{i\rightarrow i+1}$ is the odometry measurement between place number i and $i+1$, and $\ominus$ is an operator that measures the odometry between two places, as defined in (7) below. Note that the odometry is not simply a vector subtraction but takes into account the rotation of the robot as well. Therefore, $K_{i+1} \ominus K_i$ is defined as:

$$\begin{pmatrix} x_d(c_\theta + c_\psi) + y_d(c_\theta + s_\psi) - z_d s_\theta \\ x_d(s_\phi s_\theta c_\psi - c_\phi s_\psi) + y_d(s_\phi s_\theta s_\psi + c_\phi c_\psi) + z_d(s_\phi c_\theta) \\ x_d(c_\phi s_\theta c_\psi + s_\phi s_\psi) + y_d(c_\phi s_\theta s_\psi - s_\phi c_\psi) + z_d(c_\phi c_\theta) \\ \psi_{i+1} - \psi_i \\ \theta_{i+1} - \theta_i \\ \phi_{i+1} - \phi_i \end{pmatrix} \quad (7)$$

where, for example, x_d is the difference between the x coordinate of place $i+1$ and place i and s_ϕ is equal to $sin(\phi)$ for place i.

The next step is to find the variable values that minimize the function. We use the Levenberg-Marquardt nonlinear optimization [7] algorithm which is a combination of gradient descent and Gauss-Newton methods. We have $6N$ variables in our problem since we have N six-dimensional places. There are also $6(N-1) + 2 \cdot 6 = 6N + 6$ residuals for the optimization method since we have $N-1$ odometry measurements and two GPS measurements.

We need the covariance matrix of the keyplaces for the calculations of the previous section. The covariance matrix for the key places, Ω, is a $6N \times 6N$ matrix, and is equal to $(\mathcal{J}^T\mathcal{J})^{-1}$, where $\mathcal{J}$ is the Jacobian matrix that we obtain from Eq. 6.

Fig. 6 shows the result of our place learning optimization technique for a simulated run. The six-dimensional poses have been projected onto the x-y plane. The straight line shows the odometry of the robot between the ten key places (each place is shown by an ‘x’ symbol in the graph). It is assumed that odometry measurements are one meter each in the direction that the robot is facing. Note that the GPS measurement at the goal does not coincide with the robot's odometry (GPS measurements are shown with star symbols). The red curve is the result of the optimization, where the odometry measurements are stretched and the robot trajectory is directed toward the points of GPS readings. The reason for stretching odometry measurements is that the original belief of the robot was that it had traversed 9 meters while the GPS readings show about 14 meters difference between the source and the goal location.

IV. Experimental Results

In this section, we investigate if place recognition provides any significant improvement for the pose estimation in the absence of any global positioning device. We performed outdoor mobile robot experiments on the LAGR robot platform, which is equipped with IMU, wheel encoders, a GPS receiver and two stereo camera pairs. For efficiency, we only use one pair of the cameras for image matching. As discussed, we discard GPS measurements after leaving the start location. Our state vector is an 8-tuple which consists of x, y, z, yaw, pitch, roll, linear velocity of the robot and angular velocity in yaw. All results were obtained using a lag of length 25.

We first drove the robot on a rectangular path and captured key frames at the four corners of the rectangle, and performed the learning described in Section III to get a better estimate for the locations of the four key places. Then, we reset the robot position and started the place matching and pose estimation algorithm. To have a better comparison with the ground truth, we drove the robot manually again and recorded three sets of data for pose estimation: smoothed pose with place recognition, smoothed pose with complete GPS and odometry measurements and finally poses based on wheel odometry and IMU. The length of the path is roughly 80 meters, but the path is not a perfect rectangle. In addition, it is almost impossible for a human operator to drive the robot in a straight line. The result of one run is shown in Fig. 7. The plotted trajectories are the projection of the 6D trajectories onto the $x-y$ plane. The origin of the robot trajectory is the bottom left corner. The cyan trajectory shows integrated wheel odometry and IMU measurements, which naturally has a large drift. The blue trajectory is the result of the smoother that incorporates place recognition measurements and discards GPS measurements and the black trajectory is our approximate ground truth. The ground truth is obtained by a very well-tuned Kalman filter based pose estimator that is provided with the robot. It incorporates all of the measurements except place recognition including GPS along the trajectory and for our experiments resulted in an almost perfect estimate (its result is visually good and it closes the loops). As shown in the figure, our smoothed pose is almost identical to the ground truth.

Fig. 7(a) shows the locations of key frames and the locations of the robot when it visited key frames. The reason that the green crosses sometimes deviate from the path is that some of the features are not in the effective stereo range, therefore, the rigid transformation between the current location and the key frame location has a large covariance. As shown in Fig. 8 the ground in front of the robot is fairly featureless and some of the features are detected at far buildings and trees.

Fig. 8. Four corners of the rectangle that were used as key frames in the experiments. One of the stereo images is shown for each key place.

TABLE I

T-test result for the squared error distribution - comparison with complete smoothing

	p-value	max error (m)	mean error (m)	reject null hyp.?
Place recognition	0.95	5.11	1.6	no
Odometry + IMU	0.01	12.85	2.95	yes

To check if our place recognition provides a significant improvement over odometry, we repeat the experiment on the same path four times and run t-test on the error distribution to see if the errors are from distributions of the same mean or not. We recorded the output of the pose estimator for *local sensors* only (wheel odometry + IMU), smoothing using place recognition and odometry measurements (*place recognition smoothing*), smoothing using complete GPS and odometry data (*complete smoothing*) and the approximate ground truth. The number of data points is roughly 5000. The number is slightly different for each type of data since some of the processes are faster and output more pose estimates in a run. We assign a time stamp to each data point and define the error as the distance between a data point and a point on the ground truth that has the closest time stamp. Since we are mainly interested in $x-y$ plane pose estimates, the distance is calculated on that plane.

To provide a statistical comparison between place recognition and local sensors, we perform a t-test on the weighted histograms of the error. Since both the frequency and the magnitude of an error is important to us, we assign a weight to each bin in the histogram and define the weight to be the center of that bin. We used histograms of 40 equally spaced bins. Table I shows the result of t-test for the squared error. The hypothesis is that the error histograms arise from a distribution of equal means and the result for accepting or rejecting the null hypothesis at 5% significance level are

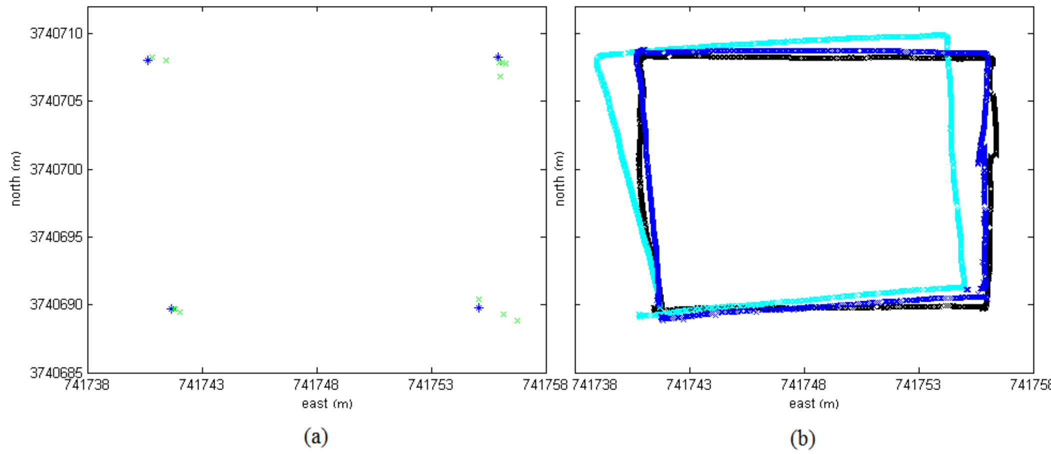

Fig. 7. (a) The positions of key places are shown as blue stars. The green crosses are the positions of the robot when it sees a key place. (b) The cyan trajectory is based on wheel odometry and IMU, the smoothed poses with place recognition are shown in blue and our approximate ground truth in black.

shown in the table. *Place recognition smoothing* and *local sensors* have been compared against *complete smoothing*. The results show the error for local sensors is significantly different from the error of complete smoothing but there is not a significant difference between place recognition smoothing and complete smoothing on the rectangular path.

V. Conclusion and Future Work

We proposed a new approach for place recognition by relying only on noisy odometry and stereo matchings with the assumption of non-reliable global positioning information. First, some special places are marked manually and their location estimates are improved using a least-squares optimization technique. Then the robot starts the autonomous run in which it searches for correspondences between the current observation and the special places in its database. The result of place recognition is treated as a new six dimensional global measurement with a given mean and covariance. We incorporated this new measurement in a fixed-lag smoother to improve pose estimation in environments where no GPS signal is available.

The current method has some shortcomings. The features that we use are not affine invariant, so if a key place is visited from a completely different view point, the robot is not able to recognize the place. Finding a computationally efficient method for 3D modeling is part of our ongoing research.

Dynamic environments are considered a challenging case for this application. We prune the outlier features which result from some moving objects in the environment, but in general the recognition task is very dependent on the structure of the scene and the size and motion of the dynamic entities. In addition, we use a fixed-lag smoother, so the false matches that are far from the current location estimation of the robot do not have large impact on the pose estimate.

Currently, our smoother only estimates the robot poses. A possible extension is to consider the location of key places as unknowns and solve the optimization for both, key place and robot poses, so that we calculate a better estimate for the location of key places as well. It is also apparent that if the robot does not see a key place on its path, the error accumulates over time and there will be no improvement in the pose estimation.

Acknowledgments

This work was funded by DARPA under the IPTO/LAGR program (contract #FA8650-04-C-7131). The authors would like to thank the Georgia Tech LAGR team for providing the research platform and the first author gratefully acknowledges the generous support of Dr. Tucker Balch during the project.

References

[1] A. Torralba, K. Murphy, W. Freeman, and M. Rubin "Context-based Vision System for Place and Object Recognition." In *Intl. Conf. on Computer Vision (ICCV)*, pp. 273–280, 2003.

[2] O. Martínez Mozos, C. Stachniss, and W. Burgard "Supervised Learning of Places from Range Data using Adaboost." In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pp. 1742–1747, 2005.

[3] H. Morita, M. Hild, J. Miura, and Y. Shirai "View-Based Localization in Outdoor Environments Based on Support Vector Learning." In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pp. 2965–2970, 2005.

[4] D. Bradley, R. Patel, N. Vandapel, and S. Thayer "Real-Time Image-Based Topological Localization in Large Outdoor Environments." In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pp. 3670–3677, 2005.

[5] J.-J. Gonzalez-Barbosa and S. Lacroix "Rover Localization in Natural Environments by Indexing Panoramic Images." In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pp. 1365–1370, 2002.

[6] E. Royer, M. Lhuillier, M. Dhome, and J. Lavest "Monocular Vision for Mobile Robot Localization and Autonomous Navigation," *IJCV/IJRR joint issue - International Journal of Computer Vision and International Journal of Robotics*, 74(3), pp. 237–260, 2007.

[7] J. Nocedal and S.J. Wright, Numerical Optimization, Springer Series in Operations Research. Springer Verlag, 1999.

[8] K. Ni and F. Dellaert "Stereo Tracking and Three-Point/One-Point Algorithms - A Robust Approach in Visual Odometry," In *Intl. Conf. on Image Processing (ICIP)*, pp. 2777–2780, 2006.

[9] A. Ranganathan, M. Kaess, and F. Dellaert "Fast 3D Pose Estimation with Out-of-Sequence Measurements," In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2007.

[10] F. Dellaert and M. Kaess "Square Root SAM: Simultaneous Location and Mapping via Square Root Information Smoothing," *Intl. J. of Robotics Research*, 25(12), pp. 1181–1203, 2006.

[11] J. Larsen "Gaussian Integrals," Technical Report, Informatics and Mathematical Modelling, DTU, 2003.