2011 IEEE/RSJ International Conference on
Intelligent Robots and Systems
September 25-30, 2011. San Francisco, CA, USA

RS-SLAM: RANSAC Sampling for Visual FastSLAM

Gim Hee Lee, Friedrich Fraundorfer, and Marc Pollefeys
Computer Vision and Geometry Laboratory, Department of Computer Science, ETH Zürich, Switzerland
glee@student.ethz.ch, fraundorfer@inf.ethz.ch, marc.pollefeys@inf.ethz.ch

Abstract— **In this paper, we present our RS-SLAM algorithm for monocular camera where the proposal distribution is derived from the 5-point RANSAC algorithm and image feature measurement uncertainties instead of using the easily violated constant velocity model. We propose to do another RANSAC sampling within all the inliers that have the best RANSAC score to check for inlier misclassifications in the original correspondences and use all the hypotheses generated from these consensus sets in the proposal distribution. This is to mitigate data association errors (inlier misclassifications) caused by the observation that the consensus set from RANSAC that yields the highest score might not, in practice, contain all the true inliers due to noise on the feature measurements. Hypotheses which are less probable will eventually be eliminated in the particle filter resampling process. We also show in this paper that our monocular approach can be easily extended for stereo camera. Experimental results validate the potential of our approach.**

I. Introduction

In the recent years, we have seen much success in applying many well established Simultaneous Localization and Mapping (SLAM) algorithms [1], [2] on robots with a single camera as the sole sensor. Two most notable works are Davison's Extended Kalman Filter (EKF) monocular SLAM [3] and Ethan's scalable monocular SLAM [4] which made used of the particle filter. Both algorithms are Bayesian approaches that require a sufficiently accurate motion model to make predictions on the robot pose before projecting the image features from the previous frame onto the current frame based on these predictions for efficient feature correspondence search in the measurement update. In the absence of other interospective sensors such as the wheel encoder, the constant velocity model was chosen as the motion model and this works very well as long as the camera motion stays within the bound of the error covariance from the constant velocity model.

The constant velocity model which has brought significant success to monocular SLAM would however be easily violated in many practical robotics applications and this violation usually causes the SLAM algorithms to fail catastrophically [5]. This happens when the robot performs an erratic motion where the movement is larger than the prediction from the constant velocity model, or when the robot is moving away from the predicted direction. Failure to predict sufficiently accurate relative robot motion often leads to failure to search for the correct feature correspondences in the measurement update and consequently causing huge errors in the pose and map estimates. Klein observed the same problem and therefore, in addition to the constant velocity model, he introduced a relocalizer in his PTAM framework [6] which relocalizes the camera during localization failures.

In this paper, we present our RS-SLAM which is a new monocular SLAM framework that combines the 5-point RANSAC [7], [8] and FastSLAM [9] algorithms. We propose a visual FastSLAM based framework which makes use of the 5-point RANSAC algorithm and image feature measurement uncertainties as the proposal distribution for the particles during the prediction step instead of using the easily violated constant velocity model. It was observed in [10] that the consensus set from RANSAC that yields the highest score might not, in practice, contain all the true inliers (thus not being the best solution) due to noise on the feature measurements. To make our algorithm less susceptible to inlier misclassifications, we propose to do another RANSAC sampling within all the inliers that have the best RANSAC score to check for inlier misclassifications in the original correspondences and use all the hypotheses generated from these consensus sets in the proposal distribution. We prevent overconfident estimates by keeping all the hypotheses where inconsistent hypotheses will eventually be eliminated by the particle filter resampling process.

Image features such as SIFT [11] and SURF [12] could be used where the descriptors for these features can be used for finding correspondences. Our algorithm follows the FastSLAM framework. During the prediction step, particle samples are drawn from the proposal distribution generated from the 5-point RANSAC algorithm and image feature uncertainties. During the update step, 3D map points are updated according to the sampled poses from the prediction step. We use vocabulary tree [13], [14] to detect loop closure opportunities, and loop closures are done within the FastSLAM framework. An important advantage of our algorithm over many Structure from Motion techniques is that we do not have to apply a separate bundle adjustment [15] process for loop closures. We also show in this paper that our monocular RS-SLAM could be easily extended to stereo camera by replacing the 5-point RANSAC with the 3-point RANSAC [16] algorithm.

In Section II, we briefly explain the 5-point RANSAC, 3-point RANSAC and FastSLAM algorithms which are essential in understanding our RS-SLAM algorithm. In Section III, we describe our monocular RS-SLAM in detail. In Section IV, we show the extension of our RS-SLAM for stereo camera. Lastly, in Section V, we show results from the experiments done to validate our algorithm.

978-1-61284-456-5/11/$26.00 ©2011 IEEE

II. BACKGROUND

In this section, we give a brief description of the 5-point RANSAC, 3-point RANSAC and FastSLAM algorithms which are essential for the derivation of our RS-SLAM. More detailed explanations of the respective algorithms could be found in [7]–[9], [16].

A. 5-Point RANSAC Algorithm

The 5-point algorithm [7] computes the relative pose between 2 monocular camera frames. As the name implies, it uses 5 pairs of matched features from two images to compute the Essential matrix E that relates the camera poses which these images are taken. The relative pose R and t between the two cameras could then be extracted from E. More details on how to extract R and t from E could be found in [17]. It is however important to note that the 5-point algorithm would perform badly with poor feature matches. The poor feature matches are known as outliers and the correctly matched features are known as inliers. The goal is then to make use of the RANSAC algorithm [8] to find the inliers set for the 5-point algorithm.

The 5-point RANSAC algorithm starts by randomly selecting 5 pairs of matched features from all the matched features across the two images. This is followed by computing the Essential matrix E from this 5 pairs of matched features using the 5-point algorithm. E is then used to compute the reprojection errors [17] for all the other matched features. All the matched features that yields a reprojection error below a certain threshold γ are counted as inliers. These inliers are known as the "consensus set". The whole process is repeated for N times, and the largest consensus set is selected as the inliers set. The number of trials N is determined by Equation (1) and is computed at the end of every trial.

$$N = \frac{\log(1-p)}{\log(1-\alpha^s)} \tag{1}$$

s is the number of features needed (s=5 in this case), p is the probability that all selected features are inliers (p is usually assigned as 0.99), and α is the probability that any selected feature is an inlier. α needs to be recomputed at every trial and is given by the number of inliers over the total number of matched features.

After getting the largest consensus set of inliers from the 5-point RANSAC algorithm, the features measurement errors become the only source of error. The next step is to minimize these errors to get an optimal estimate of the Essential matrix E with the 5-point algorithm using an augmentation of all the matches from the largest consensus set. The final relative pose R and t is then obtained from E. This is similar to the Direct Linear Transformation algorithm described in [17].

B. 3-Point RANSAC Algorithm

The 3-point algorithm computes the relative pose R and t between 2 stereo camera frames. This is because a stereo camera conveniently provides the disparity map which is used to compute the 3D coordinates of the feature points with respect to the camera. As such, a minimum of 3 point correspondences between the 3D points in the current and previous frames is needed to compute the relative pose R and t by minimizing the following cost function

$$\underset{R,t}{\operatorname{argmin}} \sum_{k}^{n} \|X_k - (RY_k + t)\|^2 \tag{2}$$

where X_k and Y_k are 3D points from the previous and current stereo frames. The absolute orientation [16] algorithm gives the solution to the minimization in closed-form where $R = VU^T$ and $t = \overline{X} - R\overline{Y}$. $\overline{X}$ and $\overline{Y}$ are the centroids of the 3D points from the previous and current stereo frames. U and V are the left and right singular vectors from the singular value decomposition of H given by

$$H = \frac{1}{N}\sum_{k}^{n}(Y_k - \overline{Y})(X_k - \overline{X})^T \tag{3}$$

After getting the inliers from the 3-point RANSAC, an optimal estimate of R and t can then be obtained by minimizing the cost function in Equation 2 with all the inliers.

C. FastSLAM

The full SLAM problem asks if a robot is able to incrementally build a consistent map M of an unknown environment and simultaneously determines its entire motion $x_{1:t}$ within this map given all the measurements $z_{1:t}$ and controls $u_{1:t}$. The SLAM problem is denoted probabilistically as

$$p(x_{1:t}, M | z_{1:t}, u_{1:t}) \tag{4}$$

and could be factorized into

$$p(x_{1:t}|z_{1:t}, u_{1:t}) \prod_{k=1}^{K} p(M_k|x_{1:t}, z_{1:t}, u_{1:t}) \tag{5}$$

This factorization means that the K features in the map M become independent of each other if the entire robot motion $x_{1:t}$ is known.

Montermerlo [9] took advantage of this observation and proposed the FastSLAM. FastSLAM uses the Rao-Blackwellized particle filter that consists of three main steps: Prediction, Update and Resampling steps. In the prediction step, particles samples are drawn from the proposal distribution given by the current robot controls u_t. The particles set now represents the distribution of plausible robot poses. In the update step, the map features are updated with the current sensor measurements z_t. Each particle holds an individual map that is built conditioned upon the robot pose represented by that particle. The map features are stored in a balanced tree hence FastSLAM has an update cost of $O(logN)$. Following the observation from Equation (5), the map features are independent of each other and its uncertainties could be represented by individual Gaussian computed from the sensor measurements. These Gaussians are then used in the assignment of weights for the particles. A particle with higher weight means lower uncertainties in

the map features and more likely to get selected in the resampling step. A critical requirement for FastSLAM to work is that the proposal distributions must be as close to the posterior distribution as possible. This requirement is satisfied by taking into account the latest sensor measurement z_t in the proposal distribution as suggested by the FastSLAM 2.0 algorithm [9].

III. Our Monocular RS-SLAM Algorithm

In this section, we describe our RS-SLAM algorithm in detail. First, we show how the proposal distribution for the FastSLAM algorithm is obtained from the 5-point RANSAC algorithm and image feature measurement uncertainties. Next, we describe how the map is updated in each of the particle. Then we describe how weights are assigned to particles for resampling and how loop closures are done. Lastly, we describe the initialization process for our RS-SLAM algorithm.

A. Proposal Distribution from 5-point RANSAC

The main role of the RANSAC algorithm mentioned earlier in Section II-A is to ensure correct data association between the image feature correspondences. It was mentioned earlier that however inliers chosen from consensus sets perturbed by noise are susceptible to errors. To alleviate this problem, we propose to first compute all the inliers from the consensus set with the highest RANSAC score followed by another RANSAC within these inliers to check for inlier misclassifications in the original correspondences and use all the hypotheses in the proposal distribution. We prevent overconfident estimates by keeping all the hypotheses where inconsistent hypotheses will eventually be eliminated by the particle filter resampling process.

The hypothesis of relative pose R and t are computed from the 5-point algorithm using each of the consensus set generated in RANSAC. This means that the feature measurement uncertainties are transformed from the image space into the space of R and t with the 5-point algorithm model. The error distribution for R and t forms the proposal distribution for our RS-SLAM, but unfortunately a closed-form formulation of this error distribution is not trivial. Hence, we propose to draw samples of the proposal distribution directly from the image space. First, we assumed that every feature measurement is corrupted by Gaussian noise with zero mean and σ_{pixel} standard deviation. A sample that represents the "true" feature position can then be obtained by subtracting a random sample drawn from the Gaussian distribution from the measured feature position. We do this for all the feature correspondences within a RANSAC consensus set. Next, we compute the "true" R and t from the 5-point RANSAC with all the "noise-free" features and this forms a sample from our proposal distribution. This sampling step is done for every particle during the prediction step in FastSLAM. Figure 1 shows a plot of the relative translations t from an example proposal distribution for our RS-SLAM.

There are two main advantages from our proposal distribution. First, every sample drawn from the proposal distribution

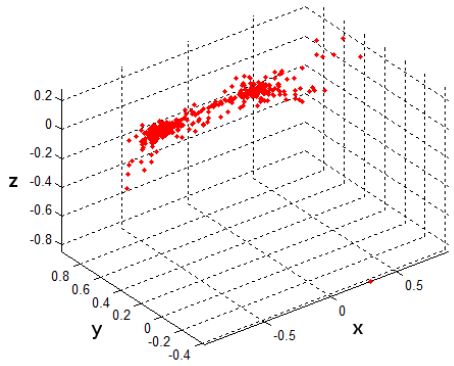

Fig. 1. An example of proposal distribution for our monocular RS-SLAM.

is computed using the 5-point algorithm and we do not make any assumptions about the robot motion. Hence, our approach is able to handle more erratic camera motions as compared to the constant velocity model. Second, we prevent wrong data association and overconfident estimates by maintaining a distribution of probable relative robot motions generated from RANSAC and the feature measurement uncertainties within our FastSLAM based framework. Particles that select correct hypotheses from the proposal distribution will build more consistent maps and therefore more likely to be selected and duplicated during the resampling process (see Section III-C for more details). As a result, we do not have to make use of a separate bundle adjustment process [15] to correct the mapping process since we are maintaining a distribution of plausible robot motions with our FastSLAM framework.

B. Map Update

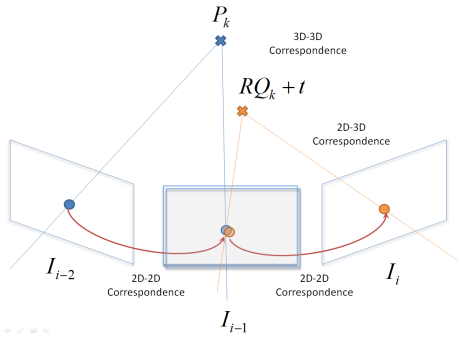

Fig. 2. Illustration of 2D-2D, 2D-3D and 3D-3D correspondences. The 3D-3D correspondence $RQ_k + t$ and P_k do not coincide because of the unknown scale factor s.

After the prediction step, each of the particle holds a predicted relative robot motion R and t sampled from the posterior distribution mentioned in the previous section. During the update step, R and t, and 2D image features are used to build a map which consists of 3D points of the environment. We first get the 3D points from R and t, and 2D image features via triangulation [17]. However, these 3D points could not be added into the map directly because the

relative scale s of R and t with respect to the 3D map is not known from the 5-point algorithm. The relative scale s has to be computed by minimizing the cost function in Equation (6), where Q_k and P_k are 3D points from the current and previous poses, and they are 3D-3D correspondences in the map. Figure 2 shows an illustration of the relationship of P_k and Q_k. P_k and Q_k are 3D-3D correspondence if their 2D-2D correspondence could be found over the current and previous 2 images I_i, I_{i-1} and I_{i-2}. R and t is the relative motion between the current and previous robot poses which transforms Q_k into the coordinate frame of P_k. $d(.)$ is the Euclidean distance between the 3D points Q_k and P_k. A minimum number of one 3D-3D point correspondence is sufficient to find s since it is a scalar number. The minimization can be done in close form as shown in Equation (7) since it involves only one single parameter s. $\langle\cdot\rangle$ refers to the dot product of the vectors $P_k \in \Re^3$ and $\widetilde{Q}_k \in \Re^3$, and $\widetilde{Q}_k$ is Q_k transformed with R and t. Another similar method to find the relative scale could be found in [18].

$$\underset{s}{\operatorname{argmin}} \sum_k d\left(s(RQ_k + t) - P_k\right) \tag{6}$$

$$s = \frac{\sum_k \widetilde{Q}_k . P_k}{\sum_k \widetilde{Q}_k . \widetilde{Q}_k} \tag{7}$$

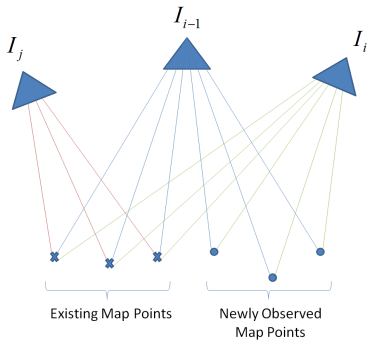

Fig. 3. Illustration of newly observed 3D points and existing map points which are observed in the current view. I_j can be any image with $1 \leq j < i-1$.

Once the scale s is known, we add newly observed 3D points into the map and update the existing map points which are observed in the current view. Figure 3 shows an illustration of newly observed and existing map points which are observed in the current view. The newly observed map points are those seen only by the current I_i and previous I_{i-1} image, and the existing map points which are observed in the current view are those seen by the current I_i image and any image I_j as long as $1 \leq j < i-1$. Since the 3D map points are independent from each other in FastSLAM (see Section II-C), we separately do a triangulation for each of the newly observed points with its feature correspondence and relative pose R and t, and do a re-triangulation for each of the existing points with all its feature correspondences and relative poses across all images that observed the points.

To make the estimates of our 3D map points more robust, we do an iterative refinement of the coordinates of the 3D map points seen in the current image by minimizing its reprojection errors. Equation (8) shows the cost function for the iterative refinement, where x_j are all the robot poses which observes the 3D map point M_k, m_{jk} is the 2D image feature that corresponds to M_k in x_j and $\Phi(.)$ is the reprojection function. Notice that we do not optimize the robot pose because it is assumed to be known after the prediction step in FastSLAM (see Section II-C). The minimization is done using the Levenberg-Marquardt algorithm [19] and the Jacobian associated to each of the 3D point are used as its error covariance. This minimization process could be done in approximately constant time because the number of observed features in the image is almost constant at any instance of time.

$$\underset{M_k}{\operatorname{argmin}} \sum_j \sum_k \left(\Phi\left(x_j, M_k\right) - m_{jk}\right) \tag{8}$$

C. Weight Update, Resampling and Loop Closure

The next step after map update is to assign the weight for every particle. The weight of each particle is defined by

$$w_t^i = |2\pi\sigma_t| \exp\left\{-\frac{1}{2}(z_t - \overline{z}_t)^T \sigma_t^{-1} (z_t - \overline{z}_t)\right\} \tag{9}$$

where z_t is the measured features locations in the current frame, $\overline{z}_t$ is the locations of the reprojections of existing 3D points in the map into the current frame which are correspondences with z_t and σ_t is the error covariances $\overline{z}_t$. σ_t is obtained from the transformation of the error covariances of the 3D map points into the image space with the standard pinhole camera model.

We build a vocabulary tree [13], [14] for the detection of any loop closure opportunity. This is done by matching the visual words extracted from the current image with the vocabulary tree, and a loop closure opportunity arises when there is a high score for the match between the visual words from the current image and a previously seen image stored in the vocabulary tree. Every particle uses the same vocabulary tree for loop closure detection. We also do a geometric check to verify from our RS-SLAM estimates that the estimated current robot pose where the current image is taken is physically close to the location where the loop closure image was taken to prevent false positive. Feature correspondences are computed for the current image and loop closure image once a loop closure opportunity arises and these correspondences are used to compute the particle weights using Equation (10).

The final step of our RS-SLAM is to do resampling where a new generation of particles are drawn from the current generation of particles with replacement. The chance of a particle getting selected is directly proportional to its weight.

D. Map Initialization

The relative scale factor s described in Section III-B could only be found if some 3D map points already exist. This

means that a separate initialization process has to be done to create new maps. We initialize the same map for all particles with the first two images from the SLAM image sequence. We adopt a standard Structure from Motion approach for the initialization. First, we compute the Essential matrix E that relates this pair of images from the 5-point algorithm. Next, the relative robot pose R and t is extracted from the Essential matrix E. We make use of the 5-point RANSAC algorithm mentioned in Section II-A for a more robust estimation of the Essential matrix E. Finally, with the knowledge of the relative robot pose R and t, we do triangulation of the image features to get the 3D map points. An arbitrary scale of 1 is set for relative robot pose R and t, and 3D map points since the absolute scale could not be obtained for monocular camera.

IV. Our RS-SLAM for Stereo Camera

In a similar vein as the monocular case, the proposal distribution of the stereo RS-SLAM also consists of the hypotheses generated from the 3-point RANSAC algorithm discussed in Section II-B, and the error distribution of each hypothesis of R and t. Unlike the monocular case, the error distribution of R and t are sample from the 3D points and its measurement uncertainties since the 3D points are directly available from a stereo camera. Figure 4 shows an example of the proposal distribution for our stereo RS-SLAM.

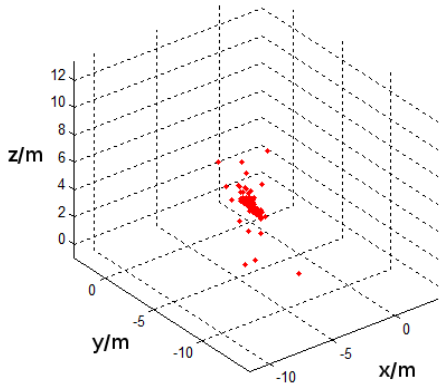

Fig. 4. An example of proposal distribution for our stereo RS-SLAM.

Also similar to the monocular case, map update for our stereo RS-SLAM is done with iterative refinements of the 3D map points seen in the current frame by minimizing its reprojection errors according to Equation 8. Note that all reprojections are done on the left camera (reference frame) of the stereo setup. Finally, the weight update of each particle from our stereo RS-SLAM is given by

$$w_t^i = |2\pi\Sigma_t| \exp\left\{-\frac{1}{2}(Z_t - \overline{Z}_t)^T \Sigma_t^{-1}(Z_t - \overline{Z}_t)\right\} \quad (10)$$

where Z_t is the 3D points from the current frame, $\overline{Z}_t$ is the existing 3D map points which are correspondences with Z_t and Σ_t is the error covariances of $\overline{Z}_t$.

V. Experiments Setup and Results

A. Monocular RS-SLAM

Fig. 5. Pixhawk quadrotor used for data collection.

We validated our monocular RS-SLAM with images taken from the Pixhawk quadrotor [20] built in our laboratory shown in Figure 5. The quadrotor is equipped with a Pointgrey Firefly MV [1] USB camera looking downward on the ground plane, and an Intel CORE 2 DUO high-speed computer for control of the quadrotor and data capture. Camera images are taken at a framerate of 30 fps, and the resolution of the images is 640 x 480. We flew the quadrotor manually over a square trajectory that forms a closed loop and spans a total distance of approximately 20m. Images were collected from the camera looking at the ground plane.

We do not use all the images from the captured image sequence for performing our RS-SLAM algorithm. This is because the 5-point algorithm for estimating the relative motions R and t works best when there is sufficient movements between the camera where two images are taken. As such, only keyframes with sufficient movements are selected for feature extraction and correspondences. An image frame is chosen as a keyframe if the average pixel movements of all the feature correspondences exceed a threshold value.

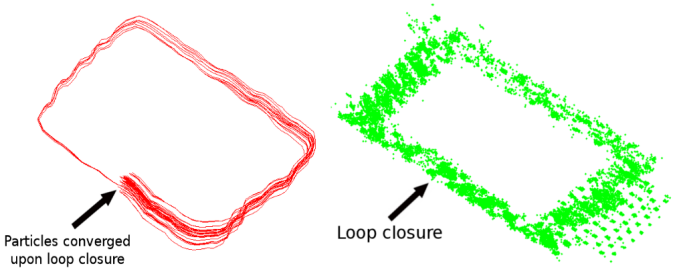

Fig. 6. (Left) Particles trajectories during loop closure. (Right) 3D map generated from the particle with the highest weight.

Figures 6 show the results from our implementation of the RS-SLAM algorithm using the image sequence collected from the quadrotor. The left plot shows the full trajectories of all the particles after loop closure. A loop closure opportunity has been detected and our monocular RS-SLAM successfully closed the loop. The right plot shows the 3D points map generated from the particle with the highest weight.

[1] http://www.ptgrey.com/

B. Stereo RS-SLAM

We apply our stereo RS-SLAM on the stereo images from the New College Dataset [21] collected from a bumblebee stereo camera mounted on a Segbot traversing through a trajectory of approximately 130m that forms a closed loop. Figure 7 shows the trajectories of the particles before (left) and after (right) loop closure. The particles diverged without loop closure (left) and converged upon loop closure (right). Figure 8 shows the map and trajectory from the particle with the highest weight after loop closure. We overlaid our result on a satellite image as a groundtruth and the nice fit verifies the accuracy of our implementation.

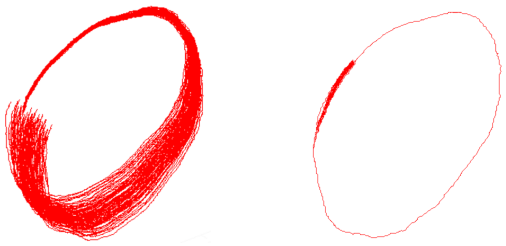

Fig. 7. (Left) Particles trajectory just before loop closure. (Right) Particles trajectory after loop closure. The particles converged upon loop closure.

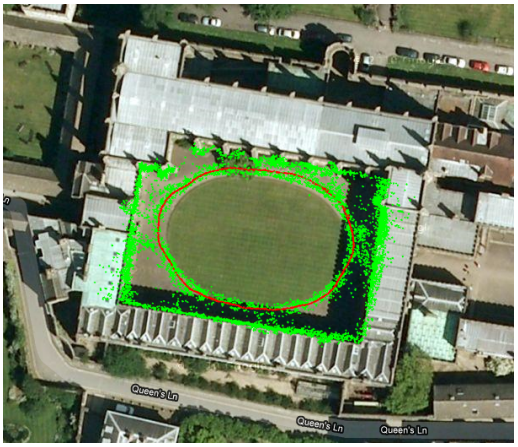

Fig. 8. Map and trajectory of the particle with the highest weight from our stereo RS-SLAM overlaid with satellite image as groundtruth.

VI. Conclusion

In this paper, we have proposed the monocular RS-SLAM algorithm. We suggested a more robust method of using the hypotheses from the 5-point RANSAC algorithm and image feature measurement uncertainties instead of the commonly used constant velocity model as the proposal distribution for the particles from our FastSLAM based framework. We also showed that the RS-SLAM algorithm can be easily extended to stereo camera. We have successfully validated both our monocular and stereo RS-SLAM algorithms with experiments and the results are shown in this paper. In our future work, we would like to also estimate the relative scale estimation for the monocular RS-SLAM probabilistically in the FastSLAM framework instead of estimating it based on the current optimization process.

Acknowledgments

This work was supported in part by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant #231855 (sFly) and by the Swiss National Science Foundation (SNF) under grant #200021-125017.

References

[1] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part i," *IEEE Robotics and Automation Magazine*, June 2006.

[2] ——, "Simultaneous localization and mapping: Part ii," *IEEE Robotics and Automation Magazine*, June 2006.

[3] A. J. Davison, "Real-time simultaneous localization and mapping with a single camera," *International Conference on Computer Vision (ICCV)*, 2003.

[4] E. Eade and T. Drummond, "Scalable monocular slam," *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2006.

[5] D. Chekhlov, M. Pupilli, W. Mayol-Cuevas, and A. Calway, "Real-time and robust monocular slam using predictive multi-resolution descriptors," *2nd International Symposium on Visual Computing*, November 2006.

[6] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," *International Symposium on Mixed and Augmented Reality (ISMAR)*, 2007.

[7] D. Nister, "An efficient solution to the five-point relative pose problem," *IEEE Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. 195–202, 2003.

[8] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Readings in computer vision: issues, problems, principles, and paradigms*, pp. 726–740, 1987.

[9] M. Montemerlo and S. Thrun, *FastSLAM: A Scalable Method for the Simultaneous Localization and Mapping Problem in Robotics*. Springer Tracts in Advanced Robotics, 2007.

[10] O. Chum, J. Matas, and J. Kittler, "Locally optimized ransac," *DAGM-Symposium*, p. 236 243, 2003.

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

[12] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," *Computer Vision and Image Understanding (CVIU)*, vol. 110, no. 3, pp. 346–359, 2008.

[13] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2006.

[14] F. Fraundorfer, C. Wu, J.-M. Frahm, and M. Pollefeys, "Visual word based location recognition in 3d models using distance augmented weighting," *International Symposium on 3D Data Processing, Visualization and Transmission*, 2008.

[15] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, "Bundle adjustment – a modern synthesis," *Vision Algorithms: Theory and Practice, Lecture Notes in Computer Science*, pp. 298–372, 2000.

[16] S. N. Berthold K. P. Horn, Hugh M. Hilden, "Closed-form solution of absolute orientation using unit quaternions," *Journal of the Optical Society of America A*, 1987.

[17] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

[18] Z. Wang and G. Dissanayake, "Map-aided 6-dof relative pose estimation for monocular slam using sparse information filters," *International Conference on Control, Automation, Robotics and Vision*, December 2010.

[19] D. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *SIAM Journal on Applied Mathematics*, p. 431441, 1963.

[20] L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys, "Pixhawk: A system for autonomous flight using onboard computer vision," *IEEE International Conference on Robotics and Automation*, May 2011. [Online]. Available: http://pixhawk.ethz.ch/

[21] M. Smith, I. Baldwin, W. Churchill, R. Paul, and P. Newman, "The new college vision and laser data set," *The International Journal of Robotics Research*, vol. 28, no. 5, pp. 595–599, May 2009. [Online]. Available: http://www.robots.ox.ac.uk/NewCollegeData/