Probabilistic outlier removal for robust landmark identification in stereo vision based SLAM

Wikus Brink[1], Corné E. van Daalen[1] and Willie Brink[2]

***Abstract*— We consider the problem of performing simultaneous localisation and mapping (SLAM) with a stereo vision sensor, where image features are matched and triangulated for use as 3D landmarks. We explain how we obtain 3D landmark measurements and derive a Gaussian noise model for these measurements. We then argue that the classic way of removing outliers from stereo image features, by estimating fundamental matrices, has limitations. We propose instead the use of a probabilistic measure in determining consensus sets of hypotheses generated in a RANSAC framework. In order to test the performance of this approach we incorporate it into an EKF SLAM system, which is notorious for its sensitivity to landmark mismatches. We measure the accuracy achieved on outdoor datasets, using DGPS as ground truth, and compare it to two other standard SLAM algorithms. We find that the proposed system outperforms the others significantly.**

I. Introduction

Many mobile robotic systems rely on accurate simultaneous localisation and mapping (SLAM). It is a technique used by a robot to build a map of some unknown environment while simultaneously tracking its own motion in this map. In developing such a technique one faces a chicken-and-egg situation because an accurate map is necessary for localisation, and accurate localisation is necessary to build a map. The inter-dependency between the estimates of the robot location and the map of the environment makes SLAM an interesting and challenging research problem.

Although SLAM is considered to be solved at a theoretical and conceptual level [1][2], practical implementation highlighted some issues that still need to be resolved, the most prominent of these being sensor related. Many SLAM systems build a probabilistic map by filling it with landmarks as they are observed by the robot's sensors. A measurement of visible landmarks relative to the robot is made at a particular time step. The task of SLAM is then to optimally combine this measurement with the current control inputs, thereby updating the robot's state and the locations of landmarks. If landmarks cannot be accurately identified and tracked over time, practical SLAM will be impossible.

In recent years vision systems have increased in popularity as a sensor for mobile robotics. Cameras are not only generally much less expensive than alternative sensors such as laser range finders and radar systems, but they also contain more information per sample. However, reliable extraction of useful information from images can be difficult.

The use of two synchronised calibrated cameras presents the possibility to extract 3D information from images captured at every time step. Feature detectors such as SIFT [3] or SURF [4] are commonly used to extract points of interests or features from images. These algorithms are designed to identify and track features over multiple images. When matched between stereo images, these features can be converted to 3D coordinates. This enables us to track features as 3D landmarks for SLAM using well-known methods.

Although much work has been done to increase the accuracy of feature matching, mismatches (outliers) may still occur on a regular basis. One such mismatch can create severe errors in the state estimates of a SLAM algorithm. In order to identify outliers, random sample consensus (RANSAC) [5] with the fundamental matrix [6] is commonly used. This approach does, however, have some disadvantages concerning execution time and robustness.

Several researchers have considered outlier removal in vision based SLAM. Vedaldi et al. [7] and Civera et al. [8], for example, both use the measurement update step of an EKF in a RANSAC framework to find outliers in a measurement. A limitation of these methods is that they require the EKF for state estimation, and therefore cannot be used with any other system (such as FastSLAM or GraphSLAM).

In this paper we present a new method of removing outliers that uses probabilistic estimates for the 3D locations of features. Our method is similar to one by Dubbelman et al. [9] but, where they use either RANSAC or expectation maximisation with the Bhattacharyya distance, we use a measurement noise model of triangulated image features to derive a new consensus measure for RANSAC. We fit the RANSAC model to the 3D location estimates of feature matches, rather than their image coordinates, and argue that this model is better suited for SLAM which uses the same 3D location estimates. Our approach has an additional advantage over the classic fundamental matrix based one in that far fewer iterations are needed.

In order to evaluate our method we incorporate it into an adaptation of the EKF SLAM algorithm, which is notoriously prone to erratic behaviour in the presence of mismatched landmarks. We compare results with FastSLAM, which typically performs better in the presence of mismatched landmarks, using location data from a differential GPS (DGPS) as ground truth. Note that although we test our method with EKF SLAM, it can be used in conjunction with any other SLAM algorithm.

[1]Wikus Brink and Corné van Daalen are part of the Electronic Systems Laboratory, Electrical and Electronic Engineering, Stellenbosch University, Stellenbosch, South Africa. wikusbrink@ieee.org, cvdaalen@sun.ac.za

[2]Willie Brink is at Applied Mathematics, Department of Mathematical Sciences, Stellenbosch University, Stellenbosch, South Africa. wbrink@sun.ac.za

II. Image features and stereo geometry

We start by briefly discussing how we find features in images, triangulate these features for use as 3D landmarks and approximate the noise associated with each measurement of a landmark. This characterisation is similar to our previous work [10].

A. Feature detection and matching

Although there are many algorithms available for identifying and matching features in images we believe the performance of our method to be largely independent of the feature detector used, due to the improved outlier removal scheme. In our implementations, we opted for speeded-up robust features (SURF) [4] because of the good compromise between execution time and accuracy. This algorithm performs the task of finding salient points in an image and calculating a descriptor for each that is invariant under scale, rotation and moderate affine transformation. Once we have identified points in two images, a nearest neighbour search can be performed on the descriptors to find matches.

For every new synchronised stereo image pair, captured at a particular time step, we follow this detection and matching procedure to obtain a measurement of the features as a set of pairs of image coordinates. We model each pair as a measurement with Gaussian noise:

$$x = \begin{bmatrix} x_L \\ y_L \\ x_R \\ y_R \end{bmatrix} + \mathcal{N}(0, N_t), \tag{1}$$

where (x_L, y_L) and (x_R, y_R) are the image coordinates of the feature in the left and right images. By $\mathcal{N}(0, N_t)$ we mean a sample drawn from the normal distribution with mean 0 and covariance matrix N_t (the same notation is used throughout the rest of this section). We choose to describe the noise covariance in Equation 1 by

$$N_t = \text{diag}(\sigma_{x_L}^2, \sigma_{y_L}^2, \sigma_{x_R}^2, \sigma_{y_R}^2), \tag{2}$$

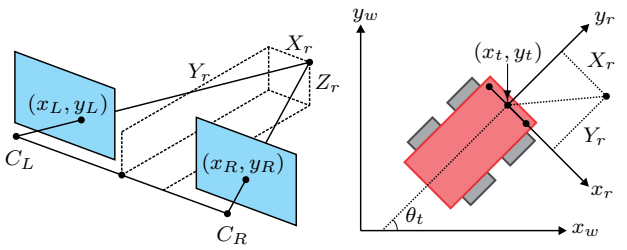
with σ_{x_L}, σ_{y_L}, σ_{x_R} and σ_{y_R} the standard deviations in pixels of the match measurement. We can then match the descriptors of a new measurement with the descriptors of features already found at previous time steps, to arrive at putative correspondences.

B. Stereo geometry of calibrated images

Now that we have stereo image features that can be tracked over time, we convert them into 3D landmarks.

Standard off-line calibration can be performed on the stereo camera pair. We rectify each incoming stereo image pair by projecting the epipoles to infinity so that every correspondence in these two images will have the same vertical coordinates [6]. Some matching errors can also be removed by enforcing this epipolar constraint.

Figure 1(a) depicts the geometry of a pair of stereo cameras with camera centres at C_L and C_R, where the image planes have been rectified, and a landmark $\begin{bmatrix} X_r & Y_r & Z_r \end{bmatrix}^T$



(a) camera geometry (b) robot geometry

Fig. 1. The geometry of our system.

observed at image coordinates (x_L, y_L) in the left image and (x_R, y_R) in the right image.

With the geometry of the stereo camera pair, the landmark location in metres can be calculated in robot coordinates as

$$\begin{bmatrix} X_r \\ Y_r \\ Z_r \end{bmatrix} = \begin{bmatrix} \frac{(x_L - p_x)B}{x_L - x_R} - \frac{B}{2} \\ \frac{fB}{x_L - x_R} \\ \frac{(p_y - 0.5(y_L + y_R))B}{x_L - x_R} \end{bmatrix} + \mathcal{N}(0, Q_t), \tag{3}$$

where B is the baseline (distance between C_L and C_R), f the focal length and p_x and p_y the x- and y-offset of the principal point, all obtained from the calibration process. Q_t is the covariance matrix of the measurement.

Note that we differentiate between robot coordinates (subscript r) and world coordinates (subscript w) as indicated in Figure 1(b), and x_t, y_t and θ_t are the robot's position and orientation in world coordinates at time t.

We know that a transformation from N_t to Q_t is possible if we have a linear system and, since Equation 3 is not linear, we use a first order Taylor approximation to find the transformation matrix

$$W_t = \begin{bmatrix} \frac{\partial X_r}{\partial x_L} & \frac{\partial X_r}{\partial y_L} & \frac{\partial X_r}{\partial x_R} & \frac{\partial X_r}{\partial y_R} \\ \frac{\partial Y_r}{\partial x_L} & \frac{\partial Y_r}{\partial y_L} & \frac{\partial Y_r}{\partial x_R} & \frac{\partial Y_r}{\partial y_R} \\ \frac{\partial Z_r}{\partial x_L} & \frac{\partial Z_r}{\partial y_L} & \frac{\partial Z_r}{\partial x_R} & \frac{\partial Z_r}{\partial y_R} \end{bmatrix}. \tag{4}$$

It then follows that Q_t can be approximated as

$$Q_t = W_t N_t W_t^T. \tag{5}$$

This approximation is performed to maintain a Gaussian noise model, which is necessary for the SLAM algorithms and our method of identifying incorrect matches.

In order to see the effect of the linearisation and evaluate the accuracy of our assumptions, we measured the noise on typical landmarks. Figure 2 depicts, for two of the several landmarks tested, 500 measurements in robot coordinates with a confidence ellipse obtained from a Gaussian fit. We compared the measured noise in the landmarks with the noise model in Equation 5, and found them to be consistently similar. These tests enabled us to calibrate the values of σ_{x_L}, σ_{y_L}, σ_{x_R} and σ_{y_R} and confirm that our approximations are valid.

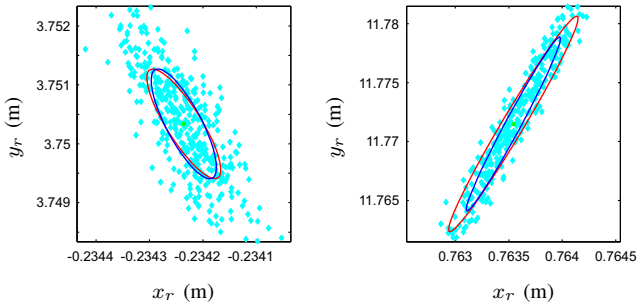


Fig. 2. Comparing noise of measured landmarks (cyan dots with fitted confidence ellipses in blue) and calculated covariance (red confidence ellipses).

III. Outlier removal

Matching feature descriptors can be sufficient for finding putative landmark correspondences. However, there will usually be some mismatches that can adversely affect the accuracy of a SLAM system. RANSAC is a popular technique for identifying and removing such errors. In this section we give an overview of the algorithm and explain how it is commonly used with the fundamental matrix. We then propose a new way of using it that is more suited for SLAM.

A. RANSAC

RANSAC [5] is a technique for estimating model parameters from data potentially containing outliers. Its basic structure can be expressed as two steps repeated k times:

1) draw a random sample from the data and fit model parameters to the sample;
2) find a consensus set from the data that correspond to the proposed model within a set threshold.

Upon completion, inliers are identified as the largest consensus set found. The model parameters can be re-estimated using the entire set of inliers.

The number of iterations, k, is commonly calculated as

$$k = \frac{\log(1-p)}{\log(1-w^n)}, \tag{6}$$

with p the probability that the algorithm will produce a good result, w the probability of choosing an inlier when a single point is drawn from the data and n the size of the sample.

If the data can be arranged from best to worst in terms of the inlier likelihood, the RANSAC algorithm can find an optimal consensus set within fewer iterations. This extension is referred to as progressive sample consensus (PROSAC) [11]. In our case the data would be sorted by the error between descriptors from the matching process. The sample used to obtain the model parameters for RANSAC is drawn from a growing subset of the data, the subset being the best matched features. This increases the likelihood of selecting better feature matches for estimating the model parameters at every iteration. PROSAC performs at least as good as RANSAC but, as shown by Chum and Matas [11], usually requires far fewer iterations to find an optimal solution.

A standard approach to RANSAC with image features is to use the fundamental matrix as a model [12]. The fundamental matrix is a 3×3 matrix that encapsulates the epipolar geometry between two camera views [6]. The Sampson distance can be used to determine consensus of data points to the proposed model. This approach can be highly effective in some cases, but it is not ideal for SLAM applications. Its first limitation is the use of image coordinates and not 3D world coordinates as those found in a typical SLAM map. Secondly, at least 7 correspondences are needed in order to calculate a fundamental matrix and, from Equation 6, this could mean that a large number of iterations would be required. The third limitation is the possibility that mismatches can be classified as inliers. If mismatched features are by chance close to corresponding epipolar lines, they will agree with the model and will not be identified as outliers.

We choose to rather work with the 3D coordinates of the measurement, in an effort to overcome these limitations. For every new measurement, there exists a 3D translation vector and rotation matrix that relates the measured landmarks to corresponding landmarks in the map already built from previous measurements. This rigid transformation can be calculated easily, using the least squares method proposed by Umeyama [13], and used as a RANSAC model. Only three 3D point correspondences are needed and this reduced sample size results in a smaller value for k and therefore a faster execution time.

A naive way to establish consensus of a 3D point correspondence to the proposed model would be to calculate the Euclidean distance between the two points after one has been translated and rotated with the model parameters. This approach, however, produces very poor results due to the nature of the measurement noise. We found the likelihood that two measurements are of the same point, explained next, to be a far better measure of consensus.

B. Improved consensus measure

Consider measurement y_1 with covariance C_1 of the point r_1 measured at time t_n, and y_2 with covariance C_2 of the point r_2 measured at t_{n+1}. We wish to calculate the probability that the two points are in fact the same point that has been measured twice (at different time steps), i.e. we want

$$P_c \triangleq P[r_1 = r_2 \,|\, y_1, y_2]. \tag{7}$$

We first define a volume $V = \{x \in \mathbb{R}^3 : ||x||_2 \leq b\}$, where b is a constant, and consider the set

$$A = \{(r_1, r_2) \in V \times V : r_1 = r_2\}. \tag{8}$$

Here $V \times V$ indicates the Cartesian cross product. If $r_1 = r_2$, then r_1 and r_2 will be contained within a volume of zero size and, if we let $\{\Delta V_1, \Delta V_2, \ldots\}$ be an infinite partition on V, we have

$$A = \lim_{a \to 0} \left(\bigcup_{i=1}^{\infty} \{(r_1, r_2) \in \Delta V_i \times \Delta V_i : \|\Delta V_i\| \leq a\} \right). \tag{9}$$

It follows that $P_c = P[A \mid y_1, y_2]$ and, since the volumes $\Delta V_1, \Delta V_2, \ldots$ do not overlap,

$$P_c = \sum_{i=1}^{\infty} \lim_{\|\Delta V_i\| \to 0} P[r_1 \in \Delta V_i \cap r_2 \in \Delta V_i \mid y_1, y_2] \quad (10)$$

$$= \sum_{i=1}^{\infty} \lim_{\|\Delta V_i\| \to 0} \int_{\Delta V_i} \int_{\Delta V_i} p(r_1, r_2 \mid y_1, y_2)\, dr_1\, dr_2, \quad (11)$$

where p indicates the probability density function. We manipulate the above using Bayes' rule so that

$$P_c = \sum_{i=1}^{\infty} \lim_{\|\Delta V_i\| \to 0} \int_{\Delta V_i} \int_{\Delta V_i} \frac{p(y_1, y_2 \mid r_1, r_2)\, p(r_1, r_2)}{p(y_1, y_2)}\, dr_1\, dr_2. \quad (12)$$

We consider the prior probability $p(r_1, r_2)$ to be uniform over V, because we do not have any prior information of the points measured, and $p(y_1, y_2)$ is a constant.

Integrating an absolutely continuous function over an infinitesimal volume is equal to the product of the integrand and the size of the volume, therefore

$$P_c = K_1 \sum_{i=1}^{\infty} \lim_{\|\Delta V_i\| \to 0} \|\Delta V_i\| \int_{\Delta V_i} p(y_1, y_2 \mid r_1, r_2)\, dr_2, \quad (13)$$

where $r_1 \in \Delta V_i$ and $K_1 = p(r_1, r_2)/p(y_1, y_2)$.

In the limit, as $\|\Delta V_i\|$ goes to zero, $r_1 = r_2 = r$ and the sizes of the volumes all become equal, say to $\|\Delta V\|$. Furthermore we assume the two measurements y_1 and y_2 to be statistically independent since they are made at different time steps. If we now use a Gaussian noise model for each measurement, as described in section II-A, we can say

$$P_c = K_2 \int_V p(y_1 \mid r)\, p(y_2 \mid r)\, dr \quad (14)$$

$$= K_2 \int_V \prod_{i=1}^{2} |C_i^{-1}|^{\frac{1}{2}}\, e^{-\frac{1}{2}(r-y_i)^T C_i^{-1} (r-y_i)}\, dr, \quad (15)$$

with $K_2 = (2\pi)^{-3} K_1 \|\Delta V\|$.

We observe that P_c can now be written in terms of an integral over a single Gaussian distribution, as

$$P_c = (2\pi)^{\frac{3}{2}} K_2 \frac{|C_1^{-1}|^{\frac{1}{2}} |C_2^{-1}|^{\frac{1}{2}}}{|C|^{\frac{1}{2}}}\, e^{-\frac{1}{2}z} \int_V f(r)\, dr, \quad (16)$$

where $C = C_1^{-1} + C_2^{-1}$,

$$z = y_1^T C_1^{-1} y_1 + y_2^T C_2^{-1} y_2 - (y_1^T C_1^{-1} + y_2^T C_2^{-1})\, C^{-1} (C_1^{-1} y_1 + C_2^{-1} y_2), \quad (17)$$

and, with $g = C^{-1}(C_1^{-1} y_1 + C_2^{-1} y_2)$,

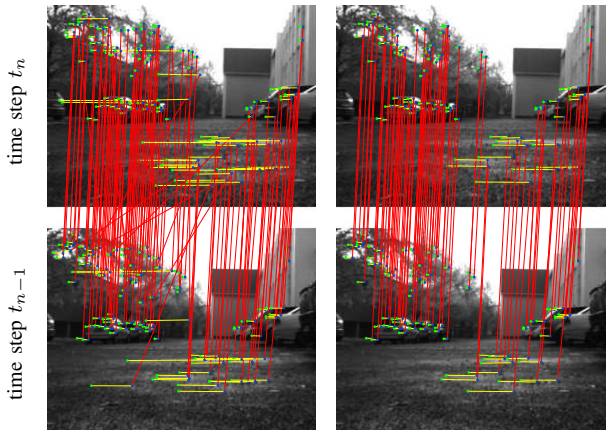
$$f(r) = \frac{|C|^{\frac{1}{2}}}{(2\pi)^{\frac{3}{2}}}\, e^{(r-g)^T C\, (r-g)}. \quad (18)$$

When $b \to \infty$, i.e. when the size of V goes to infinity, the integral part of Equation 16 evaluates to 1, yielding

$$P[r_1 = r_2 \mid y_1, y_2] \propto \frac{|C_1^{-1}|^{\frac{1}{2}} |C_2^{-1}|^{\frac{1}{2}}}{|C_1^{-1} + C_2^{-1}|^{\frac{1}{2}}}\, e^{-\frac{1}{2}z}, \quad (19)$$

with z stated in Equation 17. The expression on the righthand side of Equation 19 can now be used as a consensus measure with RANSAC as explained in Section III-A.

Figure 3 depicts images of a typical outdoor scene captured at two consecutive time steps. SURF features are detected and matched between the left and right images captured at time t_n. Detected features in the left image are then matched with features in the previous left image (of time t_{n-1}). A similar matching is done between the two right images. Those that are consistent in both the left and right image pairs are kept, and shown in (a). This set is then subjected to RANSAC with our proposed model, consensus measure and PROSAC extension. The resulting inliers are shown in (b).



(a) putative matches (b) inlier matches

Fig. 3. Left images of a typical outdoor scene taken at two time steps with feature matches (red) before and after outlier removal. Image coordinates of features in the left images (blue) and their matches in the right images (green) are also shown. The disparity (yellow) between left and right image feature coordinates is a good indication of distance from the cameras (distance is proportional to the inverse of disparity).

IV. Application: SLAM

In order to test our approach to outlier removal, we incorporate it into the classic EKF SLAM algorithm. A well-known problem with EKF SLAM is its tendency to become erratic and produce very large localisation errors if even a single landmark mismatch occurs [10]. This problem makes it an ideal candidate for our proposed method of removing outliers. We also describe a FastSLAM implementation that is used for comparison. We make some minor modifications to the original algorithms, as described by Thrun et al. [1] and Durrant-Whyte et al. [2], to better suit our requirements.

A. EKF SLAM

The EKF SLAM algorithm performs its task by maintaining a state vector comprising the location and orientation of the robot as well as the locations of all the measured landmarks. A linearisation of the robot motion and measurement equations enables us to use the Kalman filter equations to update the estimates at every time step.

Speed of execution depends mostly on the size of the map (number of landmarks) being maintained. The use of image features as landmarks can be a heavy computational burden, considering the fact that a single measurement can quite easily contain hundreds of features. For this reason we choose to maintain features in the map only temporarily, while they are observed. This eliminates the possibility of loop-closure, a trade-off that may or may not be acceptable depending on the application. Only those features that are seen in the current time step are maintained in the state vector, while all others (i.e. those that were previously, but not currently, observed) are discarded.

Our implementation of the EKF SLAM algorithm performs the following at each time step:

1) extract features from stereo images;
2) match features to previous measurement using feature descriptors;
3) flag all new landmarks, i.e. those not yet in the state vector, to be included at the next time step if they are observed again;
4) remove outlier matches with our robust RANSAC based scheme;
5) include landmarks in the state vector that were flagged at the previous time step and that match with landmarks in the current measurement;
6) perform an EKF control update with the robot's motion model;
7) perform an EKF measurement update with the observed landmark matches from the current measurement.

Steps 3 and 5 indicate that we add features only if we can confirm that they have been observed more than once. This ensures that no unnecessary landmarks are included in the map, and greatly decreases execution time.

B. FastSLAM

The FastSLAM algorithm, that we use for comparison, is based on the Rao-Blackwellised particle filter [14]. Some of the states are estimated using particles and some using EKFs. In this case the robot's states are estimated with particles and for every particle a map is maintained using an EKF for every landmark. This may seem computationally expensive, but since each EKF estimates only one landmark the matrices used in the update equations are small and computations are executed quickly.

An important feature of FastSLAM is that landmarks not observed at the current time step are not used in the update. This means that the map being maintained can grow very large without influencing execution time much, and is limited only by the memory of the computer used. Moreover, FastSLAM typically remains stable when mismatched landmarks are present. These two reasons make FastSLAM an ideal algorithm for use with image features as landmarks, and a good comparison to evaluate our EKF SLAM implementation.

V. Experimental results

In order to evaluate and compare results of the different SLAM systems we captured some outdoor datasets with two Point Grey Firefly MV cameras mounted on a Mobile Robots Pioneer 3-AT platform. The cameras were synchronised to capture images at 4 Hz and the robot was controlled by human input. Ground truth data was recorded with a DGPS (accurate to about 5cm) mounted on the robot. Note that this ground truth data is not used in any of our SLAM systems, but employed merely for comparing results and measuring accuracy afterwards.

We followed an approach similar to those in [8] and [9] to quantify the robot location error in the output of an EKF SLAM system with our proposed method of removing outliers. Table I lists for each of two separate experiments the total length of the particular trajectory, the mean and maximum error over the experiment, and the relative error with respect to the trajectory.

TABLE I
Error in position estimates, measured over two experiments.

trajectory length (m)	mean error (m)	maximum error (m)	% mean error over the trajectory
45	0.23	0.51	0.51
71	0.23	0.50	0.33

We also compared results obtained with the following systems, all implemented in MATLAB:

- naive localisation using only robot control inputs;
- EKF SLAM where outliers are removed with a RANSAC based estimation of the fundamental matrix;
- EKF SLAM where outliers are removed with our proposed probabilistic approach;
- FastSLAM with no outlier removal after initial matching (but enforcing e.g. epipolar constraints).

Figure 4 depicts the route, viewed from above, given as output by each of these four systems along with the DGPS data for the 71 m dataset. For this experiment the robot was controlled to drive forward, do a slow turn, drive straight, make a three point turn (which includes some reversing) and drive back. A subset of the features used by the EKF SLAM system with probabilistic outlier removal is visible as dots in the figure.

Figure 5 depicts the Euclidean error (in metres) in every estimated route, as measured over time against the ground truth. The FastSLAM system made a small orientation error during the first turn, which resulted in some drift, but fared quite well for the rest of the run. Thresholds in the fundamental matrix EKF SLAM system had to be set very strict to reduce the risk of outliers being misclassified as inliers. However, since this resulted in a reduced set of inliers used at every step, drift accumulated throughout the sequence. Our proposed method of removing outliers outperformed this classic method, and quite significantly so.

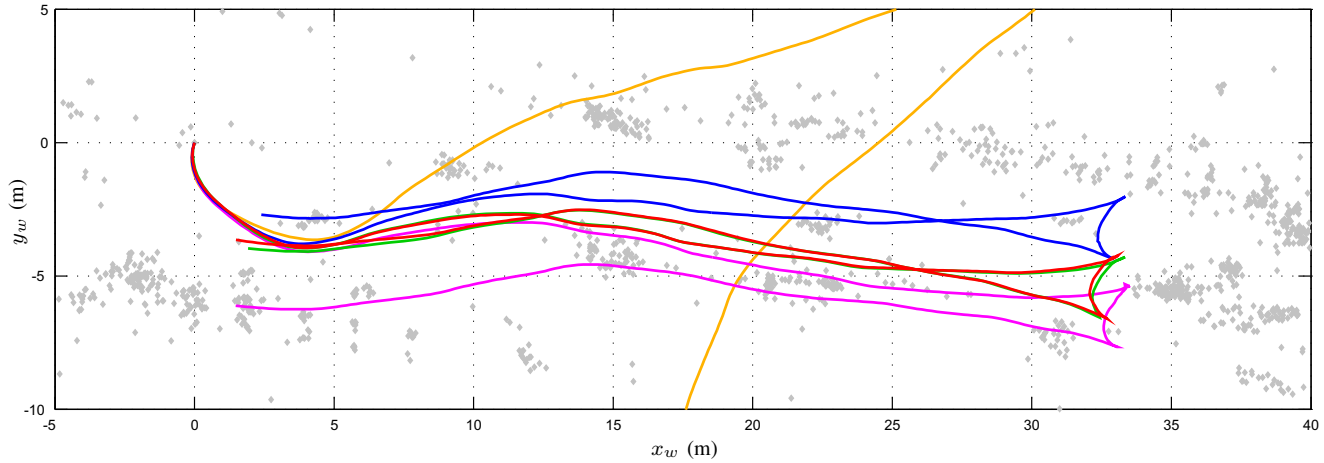


Fig. 4. Routes (starting at the origin) as estimated by robot odometry (yellow), FastSLAM (blue), EKF SLAM with fundamental matrix outlier removal (magenta), and EKF SLAM with our probabilistic 3D outlier removal (red). The ground truth measured by a DGPS is shown in green, and part of the map as built by the system with probabilistic outlier removal is shown as grey dots.

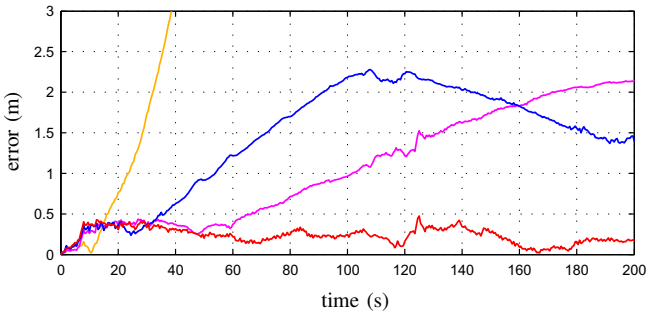


Fig. 5. Location error over time, as measured against the DGPS data, made by the four systems. The colours correspond to those used in Figure 4.

VI. Conclusions

In this paper we considered the SLAM problem and specifically the problem of performing SLAM with a stereo camera pair as the only sensor.

We characterised this sensor by deriving a Gaussian noise model for the 3D locations of triangulated image features. We then argued that the classic way of removing outliers from stereo features, by estimating a fundamental matrix in 2D image space, is not particularly well suited for SLAM. Instead, our approach estimates a model of the robot's motion and removes outliers by taking uncertainties in the 3D locations of features into account and solving that problem in a probabilistic framework.

Because of its sensitivity to outliers, EKF SLAM was picked as a platform for our outlier removal scheme. We compared it to the classic fundamental matrix EKF SLAM as well as to a FastSLAM implementation, and measured localisation errors against DGPS data. The proposed system shows significant improvement in accuracy over the others.

The EKF approach to SLAM is attractive because of its speed and simplicity and, with our way of removing outliers, it now appears to be a robust and practically viable option.

We currently use the probability in Equation 19 as a consensus measure in RANSAC, after a hypothesis has been created as a least-squares fit over the sample, and future work may include rather generating this hypothesis by maximising its probability.

References

[1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2006.

[2] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping (SLAM): Part I," *IEEE Robotics and Automation Magazine*, vol. 13, no. 2, pp. 99–110, 2006.

[3] D. Lowe, "Object recognition from local scale invariant features," *IEEE International Conference on Computer Vision*, pp. 1150–1157, 1999.

[4] H. Bay, A. Ess, T. Tuytelaars, and L. van Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[5] M. Fischler and R. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[6] R. Hartley and A. Zisserman, *Mutiple View Geometry*, 2nd ed. Cambridge University Press, 2003.

[7] A. Vedaldi, H. Jin, P. Favaro, and S. Soatto, "KALMANSAC: robust filtering by consensus," *IEEE International Conference of Computer Vision*, pp. 633–640, 2005.

[8] J. Civera, O. Grasa, A. Davison, and J. Montiel, "1-point RANSAC for EKF-based structure from motion," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3498–3504, 2009.

[9] G. Dubbelman, W. van der Mark, and F. Groen, "Accurate and robust ego-motion estimation using expectation maximization," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3914–3920, 2008.

[10] W. Brink, C. van Daalen, and W. Brink, "Stereo vision as a sensor for EKF SLAM," *22nd Annual Symposium of the Pattern Recognition Association of South Africa*, pp. 19–24, 2011.

[11] O. Chum and J. Matas, "Matching with PROSAC – progressive sample consensus," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 220–226, 2005.

[12] W. Zhang and J. Kosecka, "Image based localization in urban environments," *3D Data Processing, Visualization and Transmission*, pp. 33–40, 2006.

[13] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 4, pp. 376–380, 1991.

[14] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with Rao-Blackwellized particle filters," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, 2007.