Motion Estimation with Cooperatively Working Multiple Robots

Guleser K. Demir*, Richard M. Voyles†, and Amy C. Larson†
*Department of Electrical and Electronics Engineering, Dokuz Eylul University, Izmir, Turkey
Email: kalayci@eee.deu.edu.tr
†Department of Computer Science, University of Minnesota, Minneapolis, MN
Email: larson@,voyles@cs.umn.edu

***Abstract*— We have investigated the performance of simultaneously estimating the 3D motion and structure for navigation when the scale information is obtained by utilizing the cooperative efforts of multiple robots. The method determines the relative positions of robots by tracking a specific geometric feature that is part of their structure, and then uses the Extended Kalman Filter to estimate the motion and structure. For implementation we used two CRAWLER Scouts, and performed several experiments to explore the effects of cooperative running of robots on the motion estimation.**

I. Introduction

The coordinates of a feature in an image plane of a camera depend mainly on three distinct factors: its position in the real world, the relative 3D motion of the camera and the scene, and the internal geometry of the camera. Tracking of these features for several consecutive images enables us to estimate all three factors [7], [1]. However, it is well known that the 3D orientation of the features (i.e. structure) and translational motion of the camera can only be determined up to an unknown scale factor. For example, moving a camera around a soda can (10 cm tall) and a grain silo (30 m tall) can produce the same feature motion. This problem does not occur in estimating the rotational changes and the focal length of the camera since they are independent of the scale factor.

The actual values of both structure and motion are important for robotic applications in unknown terrains, such as search-and-rescue, planetary exploration, or surveillance. Motion estimation is essential for general navigation through a space, but it can also be used to adapt locomotion. In other work ([10]), we demonstrated that different gait motions affect different forward progress depending on the terrain underfoot. Motion estimation provides a performance measure of gait motion, thus can be used to adapt the gait to a specific terrain to improve efficiency.

In this work we propose a new method to determine the 3D motion of the robot and the structure surrounding the robot via the recovery of the scale information through the cooperative efforts of multiple robots. The main idea for recovering the scale is to consider each robot as a special geometric feature that can be accurately described in terms of a concise 3D parameterization. Each of the robots repeats move-and-stop actions and serves as a special feature for the other robots. A given robot, equipped with a camera, tracks the feature points in the environment and estimates its own motion and structure by processing the information from the other robots.

In the following section we outline the method of motion and structure estimation via the cooperative working of multiple robots. Then in Section III we give the experimental results of the method. In Section IV we conclude the work.

II. The Method

Cooperative localization was first proposed in [9], whereby two robot teams alternately moved while the other team remained stationary, acting as landmarks. In contrast to shape from motion techniques, this was posed as a deterministic triangulation. Several variations on this theme have since been developed (e.g. [4], [5], [11], to name a few), which all show improvements in motion estimation using cooperative techniques. One key element for each approach is robot recognition, which is achieved in a variety of ways including laser rangefinders, sonar, and specialized vision sensors. A theoretical analysis of the expected improvement using multirobot motion estimation is presented in [12]. Our approach uses visual information only. It estimates the relative position of one robot with respect to a stationary (landmark) robot, then uses this pose estimate along with feature tracking in an Extended Kalman filter to estimate the motion of the robot and the structure of the unknown environment. Our technique is particularly applicable to systems with no odometric or range information.

In this work we have used two robots to implement cooperative motion estimation, however, it can be generalized to multi-robot systems in which the performance and efficiency can increase via minor modifications. The two robots move in such a way that they can maintain line-of-sight, in which one robot remains stationary while the other robot moves. While seeing each other in each frame increase the performance, it is not necessary for each frame. The information from the stationary robot can be used in initializing the motion estimation and locking the scale, thus freeing the other robot(s) to actively complete their own tasks, periodically re-establishing contact.

Our testing platform is CRAWLER Scout (also known as TerminatorBot, see Figure 1), which is a small-scale robot designed for applications of search-and-rescue, planetary exploration, and surveillance. CRAWLER Scout locomotes

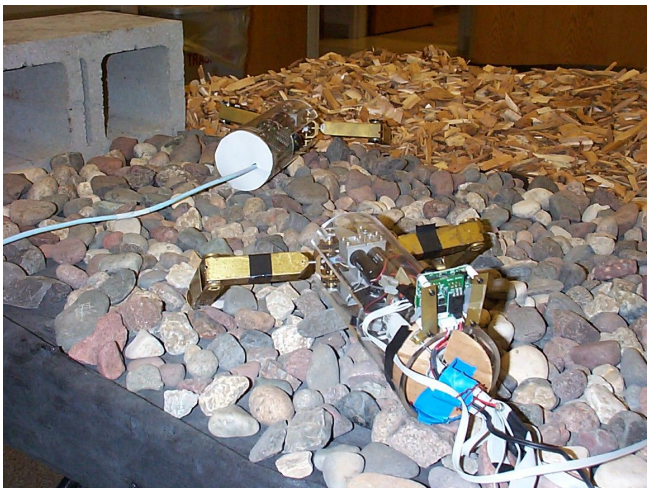

Fig. 1. *Two CRAWLER Scouts in position for cooperative localization. The front robot is equipped with the new, streamlined tether*

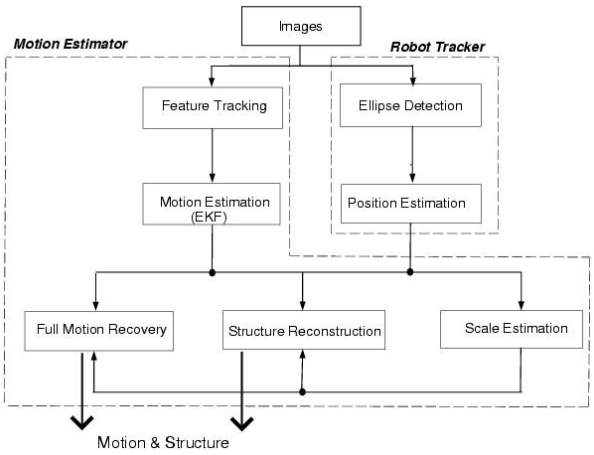

Fig. 2. General framework of the algorithm

by dragging its cylindrical body along the ground with its arms, much like cold-blooded reptiles. The versatility in arm motion allows for a variety of gaits, as well as variation within a gait. Our previous work provides details of the mechanism and developed gaits of TerminatorBot [17].

The proposed method can be considered to contain two stages: robot tracker and motion estimator. Here the robot tracker determines the relative positions of the moving and stationary robots and tracks the centroid of the stationary robot. On the other hand, the motion estimator estimates the motion and 3D locations of the feature points using the Extended Kalman Filter method. (The feature points include the centroid and the selected features from environment.) The tracker and estimator hand-shake since the motion estimator uses the information fed by the robot tracker.

The robot tracker uses visual matching of the known 3D model of the robot to the scene. For simplicity, we reduced the model of the cylindrical robot to a disk and located the back end of the cylinder. The choice of the circular disk has certain advantages: (i) its perspective projection in any arbitrary orientation is always an ellipse; (ii) it has the property of high image location accuracy; and (iii) the complete boundary or an arc of a projected circular feature can be used for 3D location estimation without knowing the exact point correspondence [13]. This approach, commonly known as 'feature-based 3D location of objects', has been extensively studied in the literature [16], [6].

To improve robustness, it is possible to track the full 3D shape of the robots, as opposed to the 2D cylindrical base. Yesin and Nelson [19] demonstrated real-time tracking of complex 3D shapes with occlusion and partial views. Using the full shape of the robot divorces the method presented here from the specific shape of our specific robot platform.

The center coordinates of the circle in the images are fed to the motion estimator as an additional feature point. Then the motion estimator uses this feature point and other feature points from the scene to estimate motion and to scale the results via the position information from the robot tracker. (Scale information is obtained using the fact that the coordinates of one of the feature points in the motion estimator should equal the position estimation from the robot tracker.) Figure 2 shows the general framework of the method. Below we describe the methods used in the motion estimator and the robot tracker in detail.

A. Robot Tracker

In order to determine the position relative to the stationary robot, we first detect the circular base of the robot whose projection on the image plane is an ellipse. Then we use a closed-form mathematical solution to solve the circular-feature-based 3D location estimation.

1) Detection of Ellipse: Here we use the color signature of the CRAWLER Scout, especially its circular base, and perform a statistical analysis to determine its color properties composed of the means and the standard deviations for each of the respective RGB color channels. Then we use color thresholding to convert the image into black-white and after median filtering we locate the ellipse. Then we apply a sobel edge detector that returns edges at those points where the gradient of the input image is at a maximum. As the last calculation in this step, we determine the five elliptical parameters of the best-fit ellipse (the center coordinates, the radii, and the orientation of the ellipse) to 2D edge points. These parameters are obtained by using a least-squares fitting technique [3].

2) Estimation of Position from Ellipse : The problem of determining the position of the circle can be done by two consecutive steps: first find a plane that intersects the general cone to form a circular curve, and then find the center of this curve [13].

Let $ax^2+bxy+cy^2+dx+ey+f=0$ be the equation of the ellipse in the x-y image plane and assume all distances expressed in units of the focal length (i.e. the focal length is set to unity). Then

$$ax^2 + bxy + cy^2 + dxz + eyz + fz^2 = P^T C P = 0\,, \quad (1)$$

where $P = [x, y, z]^T$ and C is a real, symmetric matrix of the ellipse. A number of coordinate transformations are needed to find the pose from this general cone equation. The first coordinate transformation performs a rotation R_1^T and then a translation T to obtain the right elliptic cone of the form $\lambda_1 X^2 + \lambda_2 Y^2 + \lambda_3 Z^2 = 0$. Here, rotation is determined simply by diagonalizing C. If $\lambda_1, \lambda_2, \lambda_3$ are the eigenvalues of C, with $\lambda_1 < \lambda_2 < \lambda_3$, and $\mathbf{e_1}$, $\mathbf{e_2}$, $\mathbf{e_3}$ the corresponding eigenvectors, the rotation matrix is $R_1 = [\mathbf{e_1}|\mathbf{e_2}|\mathbf{e_3}]$. The translation T can be found by balancing the related coefficients of the two equations – the equation obtained by applying rotation R_1^T and the equation of the right elliptic cone.

The next step is to apply a new transformation to find a plane such that the intersection with the right elliptical cone is a circle. This transformation, R_2, makes the new Z axis, Z', normal to the plane $lX+mY+nZ = P$. From the equation of the intersecting curve, by imposing the equality of the coefficients of $X^{'2}$ and $Y^{'2}$ and by nullity of the coefficient of $X^{'}Y^{'}$, the rotation matrix can be found as

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} \frac{-m}{\sqrt{l^2+m^2}} & \frac{-ln}{\sqrt{l^2+m^2}} & l \\ \frac{l}{\sqrt{l^2+m^2}} & \frac{-mn}{\sqrt{l^2+m^2}} & m \\ 0 & \sqrt{l^2+m^2} & n \end{pmatrix} \begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} \quad (2)$$

where

$$n = \sqrt{\frac{\lambda_2 - \lambda_1}{\lambda_3 - \lambda_1}}, m = 0 \text{ , and } l = \pm\sqrt{\frac{\lambda_3 - \lambda_2}{\lambda_3 - \lambda_1}}. \quad (3)$$

Notice that the method yields two plausible solutions in the absence of further constraints. Thus, by applying R_1 and R_2, the coefficients of the equation of the desired plane can be estimated, and from this equation, the direction cosines of the surface normals can be estimated as well.

Having found a plane which gives a circle when intersecting with the cone, now we find the position of the circle's center. Since we know the radius of the circle, r, (which is equal to the radius of the cylindrical body of CRAWLER Scout) we calculate where on the Z' axis the intersection plane should be located. Applying R_2 to the right elliptic cone, we obtain a general cone equation. We know that the actual shape is circlular, hence $a = c$ and $b = 0$ then from this resulting equation the coordinates of the center, $[X_0^{'}, Y_0^{'}, Z_0^{'}]$, are given by

$$X_0^{'} = -\frac{Z_0^{'}B}{A} \ , \ Y_0^{'} = -\frac{Z_0^{'}C}{A} \ ,$$
$$Z_0^{'} = \pm\frac{Ar}{\sqrt{B^2 + C^2 - AD}} \quad (4)$$

where

$$\begin{aligned} A &\equiv (\lambda_1 R_{1_{11}}^2 + \lambda_2 R_{1_{12}}^2 + \lambda_3 R_{1_{13}}^2), \\ B &\equiv (\lambda_1 R_{1_{11}} R_{1_{31}} + \lambda_2 R_{1_{12}} R_{1_{32}} + \lambda_3 R_{1_{13}} R_{1_{33}}), \\ C &\equiv (\lambda_1 R_{1_{21}} R_{1_{31}} + \lambda_2 R_{1_{22}} R_{1_{32}} + \lambda_3 R_{1_{23}} R_{1_{33}}) \\ D &\equiv (\lambda_1 R_{1_{31}}^2 + \lambda_2 R_{1_{32}}^2 + \lambda_3 R_{1_{33}}^2). \end{aligned}$$

B. Motion Estimator

For the motion estimator, we first detect and track appropriate features in the scene and then use an Extended Kalman Filter to estimate motion and structure.

1) Feature Detection and Tracking: Extraction of good features from the image and then tracking them across multiple images is the first step in the motion estimation problem. A general definition of a good feature is a textured image window with high intensity variation in both x and y directions, which can be tracked well across multiple images. We use a KLT (Kanade-Lucas-Tomasi) Tracker [14], which has proven useful for feature detection and tracking among several extensive studies in this area.

In this method, one examines if a small image window, W, containing N pixels centered on the pixel **x**=(x_1,x_2), is distinctive enough to be considered as a feature. Denote the image intensity function by $I(x_1, x_2)$ and consider the local intensity variation matrix

$$G = \sum_W \begin{pmatrix} g_1^2 & g_1 g_2 \\ g_1 g_2 & g_2^2 \end{pmatrix} . \quad (5)$$

with $g_i = \partial I / \partial x_i$, $i = 1, 2$. If both eigenvalues of G, λ_1 and λ_2, exceed a predefined threshold λ: min(λ_1, λ_2) $> \lambda$ the image pixel is accepted as a candidate feature.

To locate the detected features in subsequent images, which is called feature tracking, KLT seeks to minimize the intensity difference ϵ between two images I and J

$$\epsilon = \sum_{k=1}^{N} [J^k(\mathbf{x} + \mathbf{d}) - I^k(\mathbf{x})]^2 \quad (6)$$

where **d** is a displacement vector. After equalizing the derivatives with respect to **d** to zero and linearizing the system by using first-order Taylor expansion, we obtain the linear system

$$G\mathbf{d} = \mathbf{e} \quad (7)$$

where **e** is

$$\mathbf{e} = \begin{pmatrix} \sum_{k=1}^{N} [I^k(\mathbf{x}) - J^k(\mathbf{x})] g_1 \\ \sum_{k=1}^{N} [I^k(\mathbf{x}) - J^k(\mathbf{x})] g_2 \end{pmatrix}$$

for the feature motion **d**. For a given pair of successive image frames the solution of Equation 7 is found using Newton-Raphson iterative scheme.

2) Motion Estimation Using Extended Kalman Filter: The Kalman filter (KF) ([18], [20]) is a recursive estimator which gives optimal estimates of a linearly dynamic x_k given a sequence of observations z_k linearly related to the state and assuming Gaussian noise statistics. If we consider the dynamical system

$$x_k = Fx_{k-1} + w_k \quad (8)$$
$$z_k = Hx_k + v_k \quad (9)$$

where F and H are linear functions and w_k and v_k are zero mean Gaussian processes with covariances Q and R, respectively, then the time and update cycle of the Kalman

filter is given by

$$\begin{aligned} \textit{Time Update} \quad \hat{x}_k^- &= F\hat{x}_{k-1}^- \\ F_k^- &= FP_{k-1}F^T + Q \\ \textit{Measurement Update} \quad K_k &= P_k^- H^T (HP_k^- H^T + R)^{-1} \\ \hat{x}_k &= \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \\ P_k &= (I - K_k H)P_k^- \end{aligned} \tag{10}$$

for the given initial estimates. If either the state or observation equations are nonlinear then an Extended Kalman filter ([20], [18]), EKF, is used in which the RHS of Equations 8 and 9 are linearized at each time step.

In our study the observations are the 2D image coordinates of n features which are concatenated into an observation vector. Although some of the examples in the literature (e.g. [2] and [8]) directly calculated higher derivatives of motion by including both position and translational velocity with the rotation and rotational velocity in the state being estimated, we chose the simpler state of the system used by [1] (but not its structure representation). In this study, the state of the system, x, contains the scene's 3D structure, the relative 3D motion between the camera and scene and the focal length of the camera. In the representation of rotation, we use a quaternion representation instead of Euler angles. Then we formulate the rotation in terms of the incremental rotation quaternion:

$$\delta\mathbf{q} = (\sqrt{1-\epsilon}, \omega_x/2, \omega_y/2, \omega_z/2) \tag{11}$$

$$\epsilon = (\omega_x^2 + \omega_y^2 + \omega_z^2)/4\,. \tag{12}$$

From the unit incremental rotation quaternion $\delta\mathbf{q}$, the global rotation matrix R can be computed as described in [1]. In the state vector, each feature point is represented by three points, as opposed to the single state variable per feature point proposed by [1].

The feature points have a globally referenced 3D location, $I = (X, Y, Z)$, that we estimate given the measurements from multiple camera locations. Given this globally referenced location of a point, its location in the camera centered frame can be represented by

$$\begin{pmatrix} X_c \\ Y_c \\ Z_c\beta \end{pmatrix} = \begin{pmatrix} T_x \\ T_y \\ T_z\beta \end{pmatrix} \begin{pmatrix} 1 & & \\ & 1 & \\ & & \beta \end{pmatrix} R \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \tag{13}$$

where the 3x3 matrix R defines the rotation of the camera and $T = (T_x, T_y, T_z\beta)$ defines its 3D translation. In this representation the depth and camera translation in the Z-direction are not calculated directly (to avoid numerical instability in the case of long focal lengths) instead, the product $Z_c\beta$ and $T_z\beta$ are estimated.

Using the representation discussed so far we form the state vector as given by

$$x = [T_x, T_y, T_z\beta, w_x, w_y, w_z, \beta, X_1, Y_1, Z_1, \cdots, X_n, Y_n, Z_n]$$

Here, the first six parameters represent the camera translation and incremental rotation and the seventh is for the focal length of the camera. The remainder represent the positions of the n feature points in the global frame of reference.

In this study we use a central projection model in which the projection of a scene point $\mathbf{X}$= (X_c, Y_c, Z_c) onto an image plane is given by

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} X_c \\ Y_c \end{pmatrix} \frac{1}{1 + Z_c\beta} \tag{14}$$

where $\beta = 1/f$ is the inverse focal length and (u, v) is the image location of the point ([15],[1]). Indeed, geometrically, this model is identical to the usual model;

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} X_c \\ Y_c \end{pmatrix} \frac{f}{Z_c}\,, \tag{15}$$

but this formulation has the advantage of enabling estimation of the imaging geometry via the focal length f independent of the depth term Z_c over the traditional projection model, which irreparably couples the depth and the focal length estimate, making f merely a scaling factor. In addition, this model makes the transformation more stable numerically at long focal lengths (such as the orthographic projection case, where $\beta = 0$). The relation between state and observations is obtained by combining Equations 13 and 14, and since this relationship is nonlinear, an EKF is utilized to estimate the state. We assume that the motion of the camera is not known *a priori*, and thus a dynamic model in Equation 8 is chosen as the identity transform plus noise.

III. Experiments

For a quantitative discussion of the method mentioned above, we perform an experiment with 12 frames of a real image sequence. The scene contains another robot, with diameter 3.8cm starting at a distance of 83cm, and the background, which contains the KLT-tracked features, is starting at a distance of 260cm away from the moving robot.

For the initial state vector, x_0,

$$x_0 = [T_{x_0}, T_{y_0}, T_{z\beta_0}, w_{x_0}, w_{y_0}, w_{z_0}, \beta_0, X_{1_0}, Y_{1_0}, Z_{1_0}, \cdots, X_{n_0}, Y_{n_0}, Z_{n_0}]$$

we use following settings. Since we assume that the robot starts its motion at the origin of coordinates, we set the first 6 entries of x_0 as $[0, 0, -1, 0, 0, 0]$. The first feature point represents the special feature point which is the center point of the circular back of the stationary robot. Therefore, the position information coming from the robot tracker is used to set the entries of the state vector related with the special feature point ([X_{1_0}, Y_{1_0}, Z_{1_0}]). The remaining feature points, which are the KLT-tracked feature points, are all assumed to be at the same position. For the error covariance matrix we set the entries as follows. The entries related to the translational and rotational motion are set to zero (since we are sure that the robot is at the origin at the beginning). The entries related to the KLT-tracked feature points are set to high values because of the high uncertainty in the initial values of the feature points whereas those related to

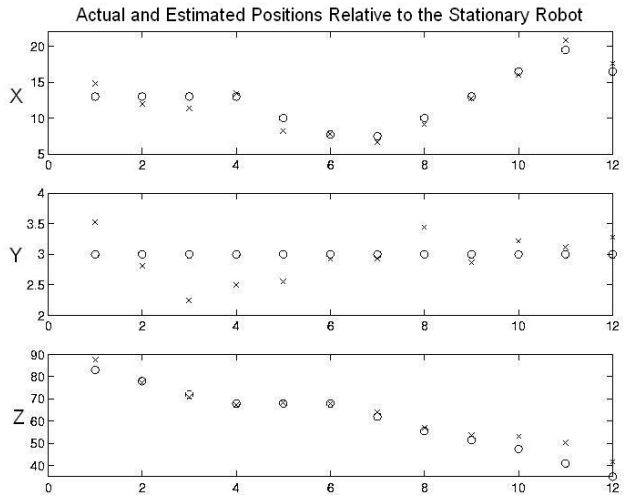

Fig. 3. The actual (o) and estimated (x) positions of the moving robot relative to the stationary robot. Estimations were obtained using our robot tracker method.

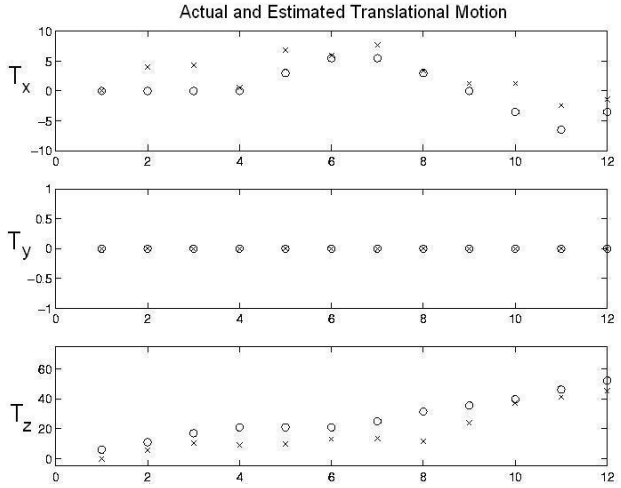

Fig. 4. The actual (o) and estimated (x) translational motion of the robot returned by robot tracker.

the feature point obtained by the robot tracker (the central coordinates of the ellipse) are set in inverse-proportion with the accuracy of robot tracker; higher the accuracy smaller the relevant entries of initial covariance matrix. Finally, for the measurement noise covariance matrix, the entries related to KLT are set to 1, whereas those related to the robot tracker are set to 3 because the latter is expected to have relatively large tracking error.

We start by illustrating the performance of the robot tracker. Depicted in Figure 3 are the actual (o) versus the estimated (x) values of the robot's position. In other words these are the relative 3D coordinates of the special feature point placed on the stationary robot. Clearly, these coordinates are used to set the related initial values of the EKF and to obtain scale information. The errors in X, Y, and Z directions are in the range of $\mp 2cm$, $\mp 0.8cm$ and $\mp 9cm$, respectively.

Next we discuss the actual and estimated translational motion of the robot using only the central coordinates of the ellipse returned by the robot tracker. Since the tracked robot is stationary during the measurements, relative changes in the centroid (in X, Y and Z directions) between successive frames is solely attributed to the translational motion of the moving robot. The experimental results are shown in Figure 4 where the actual (o) and estimated (x) translations of the robot are illustrated. Since the robot tracker has a certain amount of error, one has to arrange the related entries of the initial covariance matrix of EKF with appropriate proportion to that error.

Although we obtain motion estimation and relative position between robots from robot tracker we use EKF to improve the motion estimation and to explore the environment using different feature points from the scene. In Figure 5, we show the results of EKF estimation with 10 feature points. As noted before the entries of the initial error covariance matrix show the initial uncertainty about corresponding state variables. In this experiment we show the estimation errors with different uncertainty settings. The results shown by (*) correspond to the case where the entries related with the special feature point of the initial error covariance are set considering the robot tracker performance properly. On the other hand, the results shown by (■) correspond to the case where it is assumed that the robot tracker is working ideally (there is no error in obtained information, so related initial error covariance entries are all zero). Finally, the results shown by (◇) correspond to the case where it is assumed that the information coming from the robot tracker is too noisy. A comparison of the results in Figure 4 with the ones in Figure 5(*) prove that the use of EKF increases the estimation performance, especially in Z direction. Additionally, Figure 5 shows that setting the initial entries should be done carefully as setting larger values than the proper ones increases estimation error. Interestingly enough, although the estimation in the T_z direction is worse than the proper setting when it is assumed the tracker is working ideally (a false assumption), the estimation in T_x is better (Figure 5 (■)). We are not sure about the possible reasons for this situation.

In Figure 6, we show the absolute error in T_x and T_z values with varying number of feature points for the purpose of testing the effects of feature points on the motion estimator. Specifically, the absolute errors are shown by '*' for $N = 1$, by '■' for $N = 2$, by '◇' for $N = 10$ and by '·' for $N = 20$. Here $N = 1$ means only the feature point returned by the robot tracker is available and KLT is tracking nothing, $N = 2$ stands for two feature points, one from KLT one from the robot tracker, $N = 10$ means one point from the robot tracker and nine points from KLT, and similarly for $N = 20$. A short glance at the figure shows that for T_x, small N values give better estimation results, whereas for T_z, the estimation error tends to improve as N increases.

IV. Conclusion

In this work we proposed a new method which estimates the 3D motion and structure by utilizing the extended

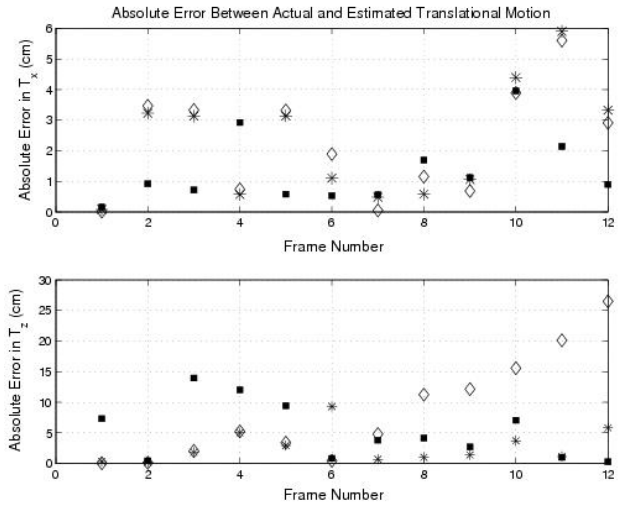

Fig. 5. The absolute error between actual and estimated translational motion when the related entries of initial error covariance matrix are set properly (*), are set assuming the accuracy of the robot tracker is perfect(■) and are set assuming the accuracy of the robot tracker is worse than actual (◇).

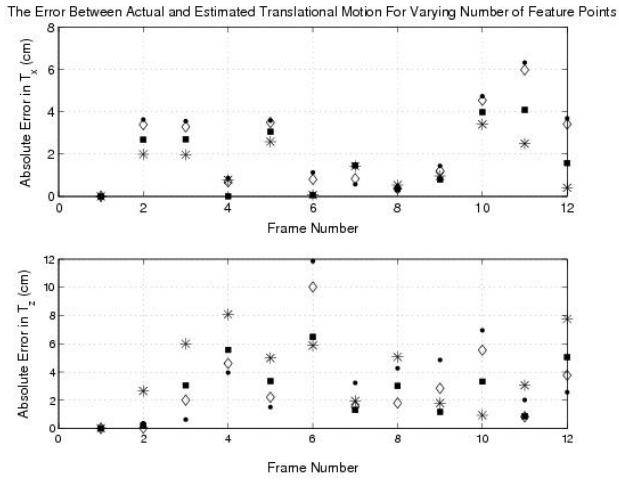

Fig. 6. The absolute error in T_x and T_Z when the number of feature points is $N = 1$ (*), $N = 2$ (■), $N = 10$ (◇) and $N = 20$ (·).

Kalman filter, where the scale information is obtained by the use of a multi-robot system. The main steps in the method can be summarized as follows: (i) track the known shape of the robot to determine centroid and pose of the body (we simplified this to tracking a disk), (ii) track several feature points in the image, (iii) set the inital parameters of EKF using the data from robot tracker, (iv) apply an EKF to find motion and 3D coordinates of the features and the robot and find scale information.

In this work, we used two CRAWLER Scouts in implementing the method and chose the circular base of its body to simplify the model matching. After using color information and edge detection, we found five elliptical parameters that the circular feature projects onto the image plane. Then from these five parameters and from the known radius of the circle, we obtained the position information by applying a closed-form analytical approach to the problem. In the experiments, we used the inter-robot distance obtained above as the initial 3D location estimation of the special feature in the EKF implementation.

The results show that the number of feature points effects the estimation performance, the use of multiple robots increases the performance of estimation and multiple robots can be used to obtain scale information.

ACKNOWLEDGMENT

We wish to thank Stergios Roumeliotis for the very helpful discussions, and Monica LaPoint and Berk Yesin for their vision code.

REFERENCES

[1] T. Jebara and A. Azarbayejani and A. Pentland, *3D Structure from 2D motion*, IEEE Sinal Processing Magazine. 16 (3), 1999.

[2] T.J. Broida, S. Chandrashekhar, R. Chellappa, *Recursive 3-D Motion Estimation from a Monocular Image Sequence*, IEEE Transactions on Aerospace and Electronic Systems, 26(4), 639-656, July 1990.

[3] A. Fitzgibbon, M. Pilu and B. Fisher *Direct Least Squares Fitting of Ellipses*, IEEE T-PAMI, 21(5), 476-480, May 1999.

[4] D. Fox, W. Burgard, H. Kruppa, and S. Thrun. *A probabilistic approach to collaborative multi-robot localization*, Autonomous Robots, 8(3):325–344, Jun. 2000.

[5] R. Grabowski, L.E. Navarro-Serment, C.J.J. Paredis, and P.K. Khosla. *Heterogeneous teams of modular robots for mapping and exploration*, Autonomous Robots, 8(3):293–308, Jun. 2000.

[6] R. M. Haralick and J. Hyonam, *2D-3D Pose Estimation* , Proc. of the IEEE 9th Int. Conf. Pattern Recognition, 385-391, November 1988.

[7] T. S. Huang and A. R. Netravali, *Motion and Structure from Feature Correspondences: A Review*, Proceedings of the IEEE, 82(2), 252-268, February 1994.

[8] Ratnam V.R. Kumar, Arun Tirumalai, and Ramesh C. Jain, *A Non-linear Optimization Algorithm for the Estimation of Structure and Motion Parameters*, Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 136-143, June 1989.

[9] R. Kurazaume, S. Nagata, and S. Hirose, *Cooperative Positioning with Multiple Robots*, Proc. of IEEE Int'l Conf. in Robotics and Automation, pp. 1250–1257, 1994.

[10] A.C. Larson, R.M. Voyles, and G.K. Demir, *Terrain Classification Through Weakly-Structured Vehicle/Terrain Interaction*, Proc. IEEE Int'l Conf. on Robotics and Automation, 218-224, 2004.

[11] I.M. Rekleitis, G. Dudek, and E. Milios. *Multi-robot Cooperative Localization: A Study of Trade-offs Between Efficiency and Accuracy*, Proc. of IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems, 2002.

[12] S.I. Roumeliotis and I.M. Rekleitis. *Analysis of Multirobot Localization Uncertainty Propogation*, Proc of IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems, pp. 1763–1770, 2003.

[13] R. Safaee-Rad, I. Tchoukanov, K.C. Smith and B. Benhabib, *Three-Dimensional Location Estimation of Circular Features for Machine Vision*, Transactions on Robotics and Automotion 8(5), 1992.

[14] J. Shi and C. Tomasi,*Good Features to Track*, Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR'94), 593-600, 1994.

[15] Richard Szeliski and Sing Bing Kang, *Recovering 3D shape and motion from image streams using non-linear least squares*, CRL Technical Report 93/3, Digital Equipment Corp., Cambridge Research Labs, March 1993.

[16] E. Trucco and A. Verri, *Introductory Techniques for 3D Computer Vision*, New Jersey, Prentice Hall, 1998

[17] R. M. Voyles, A.C. Larson, *Terminatorbot: A Novel Robot with Dual-Use Mechanism for Locomotion and Manipulation*, IEEE/ASME Transactions on Mechatronics, *to appear*.

[18] G. Welch and G. Bishop, *An Introduction to the Kalman Filter*, Technical Report TR 95-041. Department of Computer Science, University of North Carolina at Chapel Hill, 1995.

[19] K.B. Yesin and B.J. Nelson, *CAD-Model Based Visual Tracking of MEMS Components for 3D Microassembly in Microfactory*, Third International Workshop on Microfactories, ed. R. Hollis and B.J. Nelson, pp. 81–84, 2002.

[20] P. Zarchan and H. Musoff, *Fundamentals of Kalman Filtering: A Practical Approach*, AIAA, 2000.