

# **Important Copyright Notice:**

The provision of this paper in an electronic form in this site is only for scholarly study purposes and any other use of this material is prohibited. What appears here is a near-publication draft of the final paper as appeared in the journal or conference proceedings. This is subject to the copyrights of the publishers. Please observe their copyrights.

# Improved Pose Estimation for Mobile Robots by Fusion of Odometry Data and Environment Map

REZA HOSEINNEZHAD and BEHZAD MOSHIRI

*Department of Electrical and Computer Engineering, Faculty of Engineering, University of Tehran, Tehran, Iran; e-mail: rhnezhad; moshiri@ut.ac.ir*

MOHAMMAD REZA ASHARIF

*Department of Information Engineering, Faculty of Engineering, University of the Ryukyus, Okinawa, Japan; e-mail: asharif@ie.u-ryukyu.ac.jp, and Department of Electrical and Computer Engineering, Faculty of Engineering, University of Tehran, Tehran, Iran*

(Received: 30 March 2001; in final form: 10 April 2002)

**Abstract.** This paper proposes a new approach for calibration of dead reckoning process. Using the well-known UMBmark (University of Michigan Benchmark) is not sufficient for a desirable calibration of dead reckoning. Besides, existing calibration methods, usually require explicit measurement of actual motion of the robot. Some recent methods, use the smart encoder trailer or long range finder sensors such as ultrasonic or laser range finders for automatic calibration. Manual measurement is necessary in the case of the robots that are not equipped with long range detectors or such smart encoder trailer. Our proposed approach, uses an environment map that is created by fusion of proximity data, in order to calibrate the odometry error automatically. In the new approach, the systematic part of the error is adaptively estimated and compensated by an efficient and incremental maximum likelihood algorithm. Actually, environment map data are fused with the odometry and current sensory data in order to acquire the maximum likelihood estimation. The advantages of the proposed approach are demonstrated in some experiments with Khepera robot. It is shown that the amount of pose estimation error is reduced by a percentage of more than 80%.

**Key words:** pose estimation, dead reckoning calibration, occupancy grids, maximum likelihood, map building, sensor data fusion.

## 1. Introduction

Almost, sufficient knowledge about the robot's pose (consisting of its position and the direction angle of its motion) is essential in every mobile robotic application. Usually the wheels of the robot are equipped with some motion encoders and a pose estimation is achieved by the data, provided by these encoders. But this estimation requires some calibration because there are several sources of errors in the robot's pose, estimated by wheel encoders. For example, wear and tear can change the diameter of wheels or loosen belts may cause odometry error and so on. Such effects can introduce significant systematic errors into the robot's odometry. The requirement for such calibration is as old as the field of robotics itself and the

literature of methods for calibrating robots (for example, see (An et al., 1988; Cox and Wilfong, 1990; Borenstein et al., 1996; Crowley, 1995; Thrun et al., 1998). Well-nigh, all of the existing calibration methods have certain disadvantages. Many existing calibration methods call for human intercession, i.e., in order to calibrate a mobile robot's odometry, a person (or some external agent or device) has to measure the exact position and direction angle of the robot, and deduce the physical model from these measurements. Because of two reasons, such approaches are unsuitable. Firstly a considerable amount of endeavor is involved in calibration process of the robot that normally cuts it off from its continuous operation. Secondly and more importantly, the physical characteristics of the mobile robot and the environment around it change. For robot arms, robotic platforms and many other stationary devices, the environment is mainly static. In addition, the odometry error of the robot arm joint is strictly internal to the joint and is not deviated due to most changes of the environment.

On the other side, a mobile robot's odometry depends on the kind of surface that the robot is traveling on it. If the surface varies (e.g., from carpet to tile), then the calibration parameters change. Therefore, calibration process for such a robot, must be adaptive and in addition to pose estimation by dead reckoning, mobile robot must fuse odometry data with a feedback from the environment. Normally, such a feedback is generated by sensory data. Existence of a rich chronicle of using multi sensor data fusion methods in mobile robotic applications (see (Abidi and Gonzalez, 1992; Dasarathy, 1994; Richardson and Marsh, 1988; Van Dam, 1998) as some examples) enthruses us to fuse the sensory data with odometry data in order to compensate the errors and predict parameter variations. In such a localization process, robot's pose is estimated by utilizing both uncalibrated odometry and sensory data, e.g., from a laser range finder sensor. Thus, the demand for a model of the odometry error is extinguished (MacKenzie and Dudek, 1994). Thrun et al. implemented a maximum likelihood based method for pose estimation on a mobile robot, that was equipped with long range finder sensors (Thrun et al., 1998).

In our map building and navigation experiments, we used a simple Khepera robot. This robot is equipped only with infra-red proximity detectors (K-Team, 1998). The data, provided with such sensors, are appropriate for short distances only. Besides, there is no exact model to convert the proximity values to distance values and applying the ray tracing approach. For example, we cannot use the mathematical sensor model that is based on ray tracing (i.e. the likelihood of a "hit" depends on the occupancy probability of the grid cell that is being traced (Foley et al., 1990)). In this paper, a new approach is introduced for compensation of the systematic part of the error that exists in odometry data. In our approach, a new maximum likelihood method is applied to estimate the parameters. In the process of approximating the likelihood ratio, the local maps that are extracted from a previously created global map, are compared with the local map that is developed by the current values of the infra-red sensors of the robot. Actually, the extracted local map, depends on both the global map data and the estimation of

the robot's pose. Thus, corresponding to each dead reckoning parameter values, there is a different extracted map. In our method, we seek for the parameter values, corresponding to an extracted local map, that is the best fitted with the local map, generated from current sensory data.

The proposed method will be most useful for the cases, where the sensory data are not informative enough (e.g., there only infra red range finders, giving proximity data). Sometimes, position estimation is done for a robot, equipped with long-range sensors (such as ultrasonic sensors) or CCD cameras, etc., which are capable of giving more informative data about the environment. For such robots, we have better use other methods like the ray-tracing technique (Thrun et al., 1998), because our proposed method includes a higher level of computational load with respect to the other methods of dead-reckoning error calibration. Nevertheless, it can be said that the simple and commercial robots, that are designed to do simple tasks in home environment, a store, a hospital or even in a factory, are mostly equipped with chip and simple sensors like IR proximity detectors, not with more informative but more expensive sensors like cameras. This fact makes our positioning method more valuable.

In the second section, probabilistic formulation of pose estimation by dead reckoning (using odometry data) is briefly reviewed. Then, the new method for maximum likelihood estimation, is introduced in the third section. Experimental results will be given in the fourth section. Finally, we will give conclusions in the last section.

## 2. Odometry Parameters: A Brief Review

Robot motion is probabilistically modeled in our calibration method. More distinctly, let  $\pi = (x, y, \theta)^T$  denote the robot's pose in a two-dimensional space ( $\theta$  is the robot's heading direction). Robot motion is modeled by the conditional probability distribution  $P(\pi' | \pi, d)$ , where  $\pi$  is the robot's pose before executing an action,  $d$  is the displacement measured by the robot's odometry, and  $\pi'$  is the pose after executing the action.

Assume that  $d_{\text{trans}}(k)$  and  $d_{\text{rot}}(k)$  are the estimated values of translational and rotational displacements in  $k$ th iteration and  $D_{\text{trans}}(k)$  and  $D_{\text{rot}}(k)$  are their real values, respectively. If  $|l|$  is the distance, traveled between the two iterations, then our probabilistic model will be expressed as below:

$$\begin{aligned} D_{\text{trans}}(k) &= d_{\text{trans}}(k) + \alpha_{\text{trans}} \times |l| + e_{\text{trans}}, \\ D_{\text{rot}}(k) &= d_{\text{rot}}(k) + \alpha_{\text{rot}} \times |l| + e_{\text{rot}}, \end{aligned} \tag{1}$$

where the terms  $\alpha_{\text{trans}} \times |l|$  and  $\alpha_{\text{rot}} \times |l|$  stand for the *systematic* error and  $e_{\text{trans}}$  and  $e_{\text{rot}}$  are zero-mean random variables and stand for the *nonsystematic* error (refer to (Borenstein et al., 1996) for more details on systematic and nonsystematic errors of odometry data). Calibration of the robot's pose estimation is defined by estimating  $\alpha_{\text{trans}}$  and  $\alpha_{\text{rot}}$ .

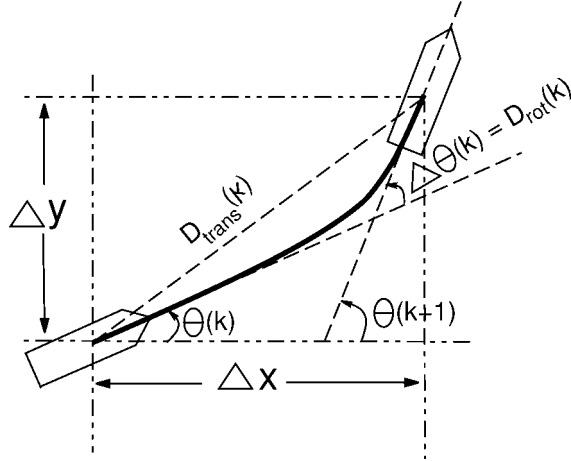


Figure 1. Robot's kinematics, based on the values of rotational and translational displacements, estimated by dead reckoning.

As (1) indicates, our model presumes that both errors increase linearly with the distance traveled. In practice, it has been found that this model is superior over other various choices, including models with more parameters (Thrun et al., 1998). Figure 1 shows one of the models for kinematics of a robot. This model is called unicycle. There are several kinematics models for mobile robots (e.g., a four-wheels car model). These models can be converted to each other and the unicycle model is the simplest model (Hwang and Ahuja, 1992). As it is demonstrated in the figure, robot's pose state transition can be expressed by the following equation:

$$\begin{cases} \theta(k+1) = \theta(k) + D_{\text{rot}}(k), \\ x(k+1) = x(k) + D_{\text{trans}}(k) \times \cos(\theta(k) + D_{\text{rot}}(k)), \\ y(k+1) = y(k) + D_{\text{trans}}(k) \times \sin(\theta(k) + D_{\text{rot}}(k)). \end{cases} \quad (2)$$

### 3. Parameter Estimation

We estimate the parameters  $\alpha_{\text{trans}}$  and  $\alpha_{\text{rot}}$  by using the sensory data and odometry data that have been gradually gathered during robot motion. Actually, it is a maximum likelihood estimation problem and we search for the parameters that appear most plausible under the existing sensory and odometry data. This estimation is formulated as below:

$$(\alpha_{\text{trans}}^*, \alpha_{\text{rot}}^*)^T = \underset{\alpha_{\text{trans}}, \alpha_{\text{rot}}}{\operatorname{argmax}} P(\alpha_{\text{trans}}, \alpha_{\text{rot}} \mid Q_k, S_{k+1}) \quad (3)$$

where  $(\alpha_{\text{trans}}^*, \alpha_{\text{rot}}^*)^T$  is the vector of optimal estimated values for  $\alpha_{\text{trans}}$  and  $\alpha_{\text{rot}}$  parameters.  $S_{k+1}$  means the sensory data (e.g., infra-red proximity or laser or ultrasonic range finder sensors or any other source of information related to environment perception) in the  $k$ th iteration and  $Q_k$  is the collection of the whole odometry

and sensory information that is gathered in  $k$ th iteration or was stored before. More specifically, it can be defined by

$$Q_k = \{S_1, O_1, S_2, O_2, \dots, S_k, O_k\}$$

where  $O_i$  is the odometry data that is/was gathered in  $i$ th iteration, i.e., the erroneous measured values of translational and rotational displacements  $d_{\text{trans}}(i)$  and  $d_{\text{rot}}(i)$ .

If the data set is large, then the maximum likelihood estimation problem is mathematically intractable (see (Thrun et al., 1998)). We have made the whole algorithm simpler and adaptive. In order to consider the variations of parameters and contribute adaptation to the estimation algorithm, parameter values are estimated in each iteration by a local maximum likelihood estimator as below:

$$(\alpha_{\text{trans}}^{*(i)}, \alpha_{\text{rot}}^{*(i)})^T = \underset{\alpha_{\text{trans}}, \alpha_{\text{rot}}}{\operatorname{argmax}} P(\alpha_{\text{trans}}, \alpha_{\text{rot}} \mid Q_i, S_{k+1}) \quad (4)$$

and then, they are adapted by the following rule:

$$\gamma \times \begin{pmatrix} \alpha_{\text{trans}}^* \\ \alpha_{\text{rot}}^* \end{pmatrix} + (1 - \gamma) \times \begin{pmatrix} \alpha_{\text{trans}}^{*(i)} \\ \alpha_{\text{rot}}^{*(i)} \end{pmatrix} \rightarrow \begin{pmatrix} \alpha_{\text{trans}}^* \\ \alpha_{\text{rot}}^* \end{pmatrix}. \quad (5)$$

Here,  $\gamma \leq 1$  is an exponential forgetting factor, that decays the weight of measurements over time. It is usually selected near to 1 and was 0.9 in our experiments. Equations (4) and (5) were used for parameter estimation, for simplicity and adaptation purposes. The simplicity property of this formulation is more important and it is achieved by iterative estimation instead of global estimation which is much more difficult. It is important noting that both the adaptability of this formulation and the adaptability of the environment mapping process, guarantee that pose estimation is adaptive to the slow enough variations of the robot environment. More details about this matter is brought in Section 5.

### 3.1. LIKELIHOOD FUNCTION

The only thing that is remained to be computed or approximated, is the parameter likelihood function  $P(\alpha_{\text{trans}}, \alpha_{\text{rot}} \mid Q_i, S_{i+1})$  in (4). According to Bayesian rule, this value can be expressed as follows:

$$\begin{aligned} & P(\alpha_{\text{trans}}, \alpha_{\text{rot}} \mid Q_i, S_{i+1}) \\ &= \zeta \times P(S_{i+1} \mid Q_i, \alpha_{\text{trans}}, \alpha_{\text{rot}}) \times P(\alpha_{\text{trans}}, \alpha_{\text{rot}} \mid Q_i) \end{aligned} \quad (6)$$

where  $\zeta = (P[S_{i+1} \mid Q_i])^{-1}$  is a normalizing factor and can be ignored during the maximization process. The knowledge in  $Q_i$  without knowing about  $S_{i+1}$ , contains no information related to  $\alpha_{\text{trans}}^{*(i)}$  and  $\alpha_{\text{rot}}^{*(i)}$ . So,  $P(\alpha_{\text{trans}}, \alpha_{\text{rot}} \mid Q_i) = P(\alpha_{\text{trans}}, \alpha_{\text{rot}})$  and this is a priori probability that may also be discarded while maximization process. It remains to calculate or approximate the term  $P(S_{i+1} \mid Q_i, \alpha_{\text{trans}}, \alpha_{\text{rot}})$ . We call this term *sensation likelihood*.

### 3.2. ESTIMATION OF THE SENSATION LIKELIHOOD

Assume that  $W$  is the world and  $\Delta\pi$  is the relative displacement between the robot's poses  $\pi_{i+1}$  and  $\pi_i$ . According to the theorem of total probability, sensation likelihood can be calculated by:

$$P(S_{i+1} | Q_i, \alpha_{\text{trans}}, \alpha_{\text{rot}}) = \iint P(S_{i+1} | W, \Delta\pi, Q_i, \alpha_{\text{trans}}, \alpha_{\text{rot}}) \times P(W, \Delta\pi | Q_i, \alpha_{\text{trans}}, \alpha_{\text{rot}}) dW d\Delta\pi \quad (7)$$

Since the sensor data  $S_{i+1}$  is independent from  $Q_i$  and  $\alpha_{\text{trans}}, \alpha_{\text{rot}}$ , and the world  $W$  and the displacement data  $\Delta\pi$  are independent from each other, (7) can be expressed as below:

$$P(S_{i+1} | Q_i, \alpha_{\text{trans}}, \alpha_{\text{rot}}) = \iint P(S_{i+1} | W, \Delta\pi) \times P(W | Q_i, \alpha_{\text{trans}}, \alpha_{\text{rot}}) \times P(\Delta\pi | Q_i, \alpha_{\text{trans}}, \alpha_{\text{rot}}) dW d\Delta\pi. \quad (8)$$

Besides, we know that  $W$  is independent of the motion data and parameters. Also  $\Delta\pi$  depends only on the recent motion data  $O_i$  and the  $\alpha$  parameters. Thus (8) can be more simplified as follows:

$$P(S_{i+1} | Q_i, \alpha_{\text{trans}}, \alpha_{\text{rot}}) = \iint P(S_{i+1} | W, \Delta\pi) \times P(W | S_1, S_2, \dots, S_i) \times P(\Delta\pi | O_i, \alpha_{\text{trans}}, \alpha_{\text{rot}}) dW d\Delta\pi. \quad (9)$$

Of course, integrating over all possible worlds  $W$  and all displacements  $\Delta\pi$  is infeasible. We can approximate the sensation likelihood by substituting the integrals in (9) with their expected values. They are much easier to calculate. Finally, the following expression, is proposed as a close approximation for the sensation likelihood:

$$P(S_{i+1} | Q_i, \alpha_{\text{trans}}, \alpha_{\text{rot}}) \cong P(S_{i+1} | W = E[W | S_1, S_2, \dots, S_i] \wedge \Delta\pi = E[\Delta\pi | O_i, \alpha_{\text{trans}}, \alpha_{\text{rot}}]) \quad (10)$$

where  $E[\cdot]$  is an indication of the conditional expected quantity of a random variable.

### 3.3. CALCULATION OF EXPECTATIONS AND APPROXIMATED LIKELIHOOD

In this subsection, calculation of the two expected values and the approximated likelihood in (10) are explained. The first term,  $E[W | S_1, S_2, \dots, S_i]$  is considered as a global occupancy grids map of the environment, that has been generated up to the  $i$ th iteration. There is a large set of methods for creation of such a map. It can be generated by Bayesian fusion of sensory data (Elfes, 1989, 1987) or by a more intelligent neural-Bayesian approach (Van Dam, 1998) or by our new method, called

*pseudo information fusion* (Asharif et al., 2000, 2001). Although there are several approaches for environment mapping in mobile robotics, occupancy grids has been selected (see (Thrun, 1998, 1995) for more information about mapping methods and map learning). That is because a grid-based map can be easily created, handled and applied to the likelihood estimation, as it will be shown later in this subsection. The second term,  $E[\Delta\pi \mid O_i, \alpha_{\text{trans}}, \alpha_{\text{rot}}]$  is an expected displacement vector, denoted by  $E[(\Delta x, \Delta y, \Delta\theta)^T \mid O_i, \alpha_{\text{trans}}, \alpha_{\text{rot}}]$  where  $O_i = (d_{\text{trans}}, d_{\text{rot}}, |l|)^T$  is the recent movement information that is obtained by using the wheel encoders data. Equations (1) and (2) can be utilized to calculate the expected value of  $\Delta\pi$ , knowing the above information. But the non-systematic error term in (1) is ignored in our formulation because it covers a minor part of the total error and also it depends on many stochastic processes, so there are many parameters if it is to be estimated and this estimation will be difficult. This error occurs because of the non-flatness of the floor surface, random slipping of the robot’s wheels, etc. Actually the non-systematic error depends mainly on the environmental characteristics. Thus, if we would like to model and estimate it, some stochastic model must be used with parameters that mainly depend on environmental characteristics. By this approach, we might be able to model and estimate this error through passing many times the same and well-known environment, but when the environment changes, this modeling and estimation must be repeated.

The only term that must be estimated, is the sensation likelihood, i.e.,  $P(S_{i+1} \mid W; \Delta\pi)$ . This is the likelihood of the scan, recorded in the final position. If the sensors are long distance range finders (e.g., laser or ultrasonic sensors), then the likelihood value may be obtained by a simple ray tracing. In that case, the likelihood of a “hit” depends on the occupancy of the grid cell that is being traced. Consequently, sensor measurements that are more fitted to the occupancy grids map, will get a higher likelihood value, while measurements that contradict the map, will be assigned a lower likelihood. See (Thrun et al., 1998; Foley et al., 1990) for more details.

If the robot is not equipped with long range finder sensors, but only with some proximity range detectors (e.g., infra-red proximity detectors in the case of a Khepera miniature robot), then ray tracing cannot be applied to the likelihood estimation problem. In such a case, we may use the idea of *local map matching*. It is based on the fact that any sensor scan is uniquely corresponded to a local map around the robot. More specifically, there is a transformation  $\mathbf{T}$  that transform a sensor scan  $S_{i+1}$  to a local map  $\Gamma$ , i.e.,  $\Gamma = \mathbf{T}(S_{i+1})$ . In our experiments with Khepera, this transformation has been implemented by using a feed forward neural network (multi-layered perceptron). We will discuss it in the next section. It is assumed that this transformation is a one-to-one correspondence (i.e., two different sensor measurements are transformed to two different local maps around the robot). This assumption is practically valid. So, the sensation likelihood may be expressed as below:

$$P(S_{i+1} \mid W, \Delta\pi) = K \times P(\Gamma \mid W, \Delta\pi) \quad (11)$$

where  $K$  is a factor that can be ignored during maximization. Knowing a global map of the environment (or the world model  $W$ ) and an estimated location of the robot in this map (calculated by the information in  $\Delta\pi$ ), one can easily extract a previously-known local map around the robot called  $\Gamma'$ . Both of the local maps  $\Gamma'$  and  $\Gamma$  are two matrices of real numbers in  $[0, 1]$  (occupancy probabilities) with the same size. Actually if a local map is extracted from a real map of the environment, it will only have items of 0 or 1, in an ideal case. But the local map  $\Gamma$  is generated by a neural network\* that its inputs are sensory data. Because of the errors and uncertainties existing in sensory information, the output of the neural network is not restricted to be only 0 or 1 and it is in  $[0, 1]$ . It is interpreted as a probability value. Thus if two sensory measurements are different, it is almost impossible that they are projected to the same local maps with the same sets of all probabilities. That is why, we practically assume that this projection is one-to-one.

We interpret the conditional probability  $P(\Gamma | W, \Delta\pi)$  as a judgment about the existing fitness between these two local maps. This fitness is formulated by the following equation:

$$f(\Gamma, \Gamma') = 1 - \frac{\|\Gamma - \Gamma'\|}{\|\Gamma - \frac{1}{2}I\| + \|\Gamma' - \frac{1}{2}I\|} \quad (12)$$

where  $I$  is a matrix of the same size as  $\Gamma$  and  $\Gamma'$  with all elements equal to 1 and  $\|\cdot\|$  is matrix norm, defined by

$$\|A\| = \sum_i \sum_j |\alpha_{ij}|. \quad (13)$$

This fitness measure, is normalized to  $[0, 1]$  interval. In an ideal case, where  $\Gamma$  and  $\Gamma'$  are absolutely matched, this factor becomes 1. In the worst case, where the two maps are completely contradicting, i.e.,  $\forall i, j; \gamma_{ij} + \gamma'_{ij} = 1$ , the fraction part of the expression will have a numerator equal to its denominator and so, this factor will be zero.

Actually, only if there is sufficient occupied area around the robot, then (12) is a valid measure for the fitness between the two local maps. A threshold,  $\mu^{\min}$ , is introduced for the total existing occupancy in both of the local maps. Parameter calibration by maximum likelihood will take place if the following occupancy condition is true:

$$\|\Gamma\| + \|\Gamma'\| \geq \mu^{\min}. \quad (14)$$

As a simple example, two typical local maps around the robot are shown in Figures 2 and 3. They have been extracted from a global map, corresponding with the same location of the robot while it acquires a sensor scan. The difference is that no calibration has been applied to pose estimation in Figure 2 while utilization

---

\* The neural network has been introduced in this work, for local map generation. In other cases, another method may be used for performing this task.

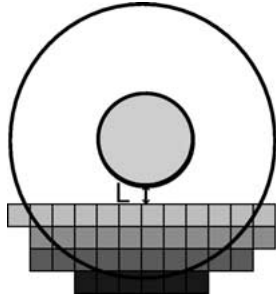


Figure 2. A local map extracted from a global occupancy grids map. No calibration has been applied to the robot's pose estimation process in this case.

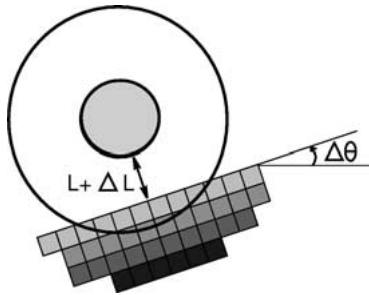


Figure 3. A local map extracted from a global occupancy grids map. Calibration of pose estimation, has led to some rotational and translational displacement in the extracted map.

of the adapted values of  $\alpha_{\text{trans}}$  and  $\alpha_{\text{rot}}$  in calibration of dead reckoning in (1) has led to some rotational and translational displacements  $\Delta L$  and  $\Delta\theta$  in the extracted local map in Figure 3. By the way, any of the two maps, can play the role of  $\Gamma'$  in (12) and be compared to the local map, generated from the recent sensor scan.

### 3.4. ALGORITHM DESCRIPTION

Figure 4 demonstrates a step by step algorithm of our calibrated pose estimation method while the robot is exploring and mapping the environment around itself. It acquires a sensor scan  $S_i$  at the  $i$ th iteration. By using the recently obtained sensory data it updates a global occupancy grids map, that has been gradually created by employing a map building method, e.g., Bayesian method (Elfes, 1989, 1987), Dempster–Shafer reasoning (Murphy, 1998) method or pseudo information method (Asharif et al., 2000, 2001). The robot moves to a new location in the next step ( $(i + 1)$ th iteration). It acquires the odometry data  $O_i$  and gets a new sensor scan  $S_{i+1}$ . The robot is able to create a local map, merely by the new sensory data (without requiring any positional information). On the other hand, it estimates its new location, by making use of the odometry data and calibrating its estimate by the current dead reckoning parameters  $\alpha_{\text{trans}}$  and  $\alpha_{\text{rot}}$ . It uses Equation (1) without the non-systematic error terms and Equation (2) in this step. Now a local map can

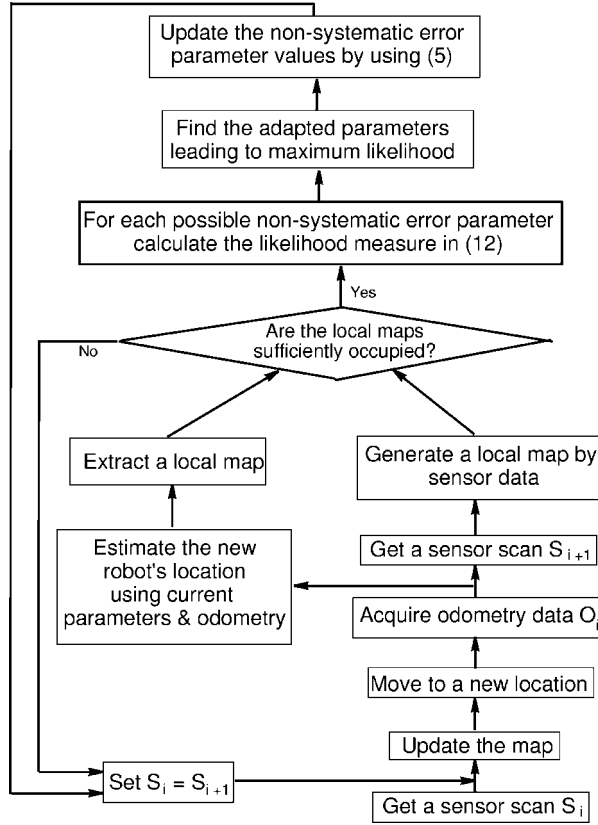


Figure 4. Our proposed step by step algorithm of calibrated pose estimation for a mobile robot, while it is exploring and mapping the environment around itself.

be extracted from the occupancy grids that is centered at the estimated position of the robot. This map is calculated in the same size of the local map that is generated from sensory data. In the next step, the two maps are checked if they satisfy the sufficient occupancy criterion. If they do not, algorithm skips to the next iteration by setting  $S_i = S_{i+1}$  and updating the map, etc. Otherwise, for each possible  $(\alpha_{\text{trans}}^{*(i)}, \alpha_{\text{rot}}^{*(i)})^T$  values, the new robot's pose is re-estimated and the corresponding local map is re-extracted and the fitness measure  $f(\Gamma, \Gamma')$  is calculated by (12). The parameter values which lead to maximum fitness and hence to maximum likelihood, are chosen as  $\alpha_{\text{trans}}^{*(i)}, \alpha_{\text{rot}}^{*(i)}$  and applied to Equation (5) to adapt  $\alpha_{\text{trans}}$  and  $\alpha_{\text{rot}}$  values in the next step. Finally, the next iteration is started by replacing  $S_i$  with  $S_{i+1}$  and updating the map, etc.

#### 4. Experiments

Khepera miniature mobile robot has been applied to exploration, localization and map building experiments. In these experiments, the data provided by 8 infra-

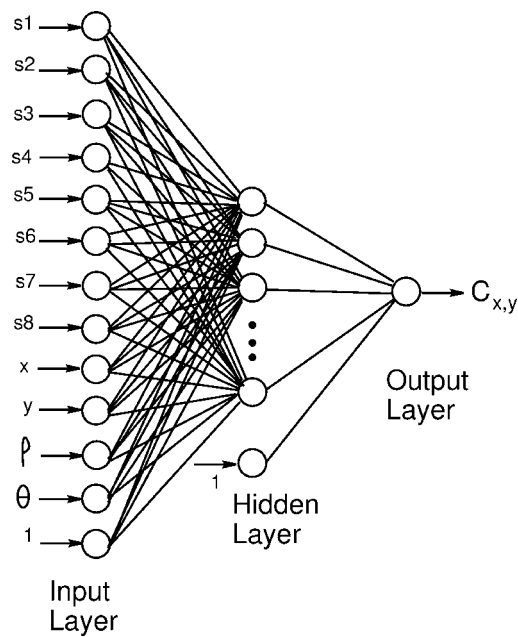


Figure 5. The feed-forward perceptron that was trained for local map building, by using proximity data, provided by Khepera's infra-red sensors.

red sensors around Khepera, were the only sources of information. (See (K-Team, 1998) or (Mondada et al., 1993) for more details on Khepera sensors and structure.) There is no accurate inverse model for the infra-red proximity detectors, like the model for ultrasonic range finders (Oriolo et al., 1997). We trained a feed-forward multi-layered perceptron to implement an inverse model for the sensors. The inputs of the network are the eight proximity values, and the local coordinates of a cell in the occupancy grids map around the robot. The output of the network is the occupancy probability value of the cell. The architecture of the neural network is shown in Figure 5. This output value is fused with the associated occupancy probability value of the same cell in a global map of the environment. This global map has been gradually calculated based upon previous sensory information.

Finally, the resulting probability is applied to improve the global map. In other words, during exploration of the environment by the reactive obstacle avoidance method of Braitenberg (Braitenberg, 1984),\* Khepera creates a local occupancy grids map in every sensing iteration. These local maps are integrated with an initially blank global map, by using the pseudo information fusion method (Asharif et al., 2000, 2001) and it improves gradually. In the figures of this section, the routes of the robot motion are plotted inside an occupancy grids map of the environment

\* Another more intelligent alternative for obstacle avoidance is our newly introduced method of fuzzy rule-based command fusion (Moshiri et al., 1999).

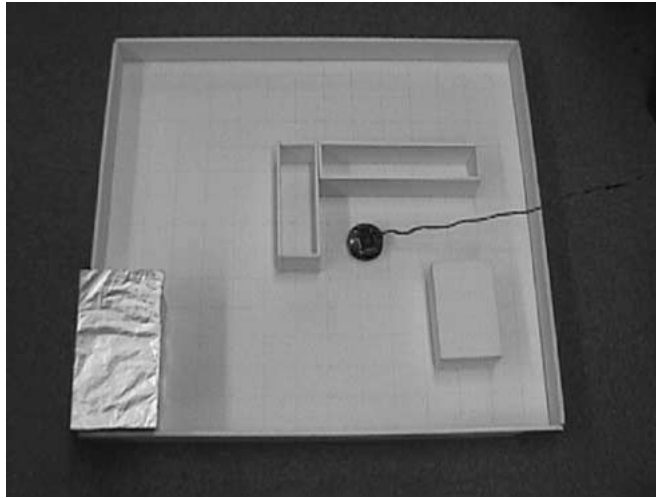


Figure 6. A photo of Khepera and the environment, in our experiments.

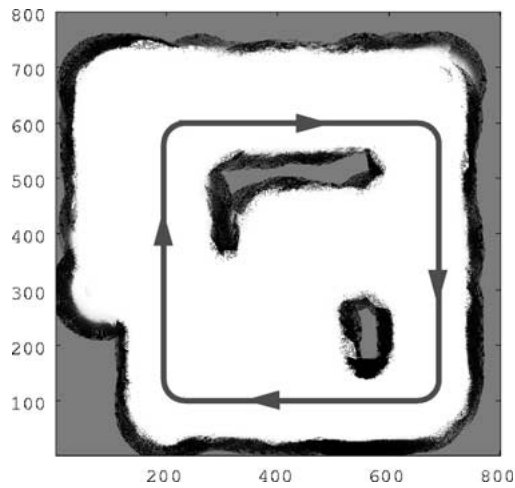


Figure 7. The square path, traveled by the robot in our UMBmark experiment.

that had been already generated by the same robot. This was done to implicitly show the potential of concurrent mapping and positioning for the mobile robot, while using the proposed self-localization method.

Figure 6 shows the typical environment in which our experiments have been done. The well-known UMBmark (University of Michigan Benchmark) was tried in the first experiment. The robot travels a square path with a perimeter of 200 cm length, 16 times in clockwise and 16 times in counterclockwise direction. Figure 7 shows the occupancy grids map, created by pseudo information fusion, and the square path inside it. Practically, the robot does not travel the exact square path in each round, because the motion control commands are created based on the robot's

Table I. Details of  $\Delta x$  and  $\Delta y$  values (in millimeters) and the relative errors, resulted in each round of the first trial of UMBmark experiment. In this trial there is no calibration and that is why the error values are undesirable

Round number	Clockwise rounds			Counter-clockwise rounds		
	$\Delta x$	$\Delta y$	Relative error %	$\Delta x$	$\Delta y$	Relative error %
1	42.2	27.5	25.2	41.6	22.6	23.7
2	41.0	32.6	26.2	32.2	20.7	19.1
3	30.0	26.3	19.9	42.1	32.1	26.5
4	30.3	25.6	19.8	43.8	23.5	24.9
5	42.3	32.3	26.6	34.7	19.8	20.0
6	43.4	25.3	25.1	31.2	32.3	22.5
7	34.1	32.0	23.4	37.8	20.3	21.5
8	33.9	24.3	20.9	30.7	28.0	20.8
9	30.6	29.5	21.3	34.7	27.8	22.2
10	36.3	28.6	23.1	40.8	29.2	25.1
11	35.9	22.8	21.3	33.4	32.1	23.2
12	27.0	20.0	16.8	39.1	31.5	25.1
13	40.9	25.9	24.2	29.8	29.1	20.8
14	41.2	20.5	23.0	43.0	24.2	24.7
15	29.1	27.4	20.0	35.0	23.1	21.0
16	29.8	20.2	18.0	18.1	25.0	15.4

pose information, acquired from the erroneous odometry data. It causes the robot not to return to its initial starting point in each round. This deviation of the location of the robot with respect to the starting point was manually measured. In the first trial of 32 rounds, their mean values were  $|\overline{\Delta x}| = 35.5$  mm and  $|\overline{\Delta y}| = 26.3$  mm, equivalent to a relative error of 22.1%. Table I shows the detailed values of  $\Delta x$  and  $\Delta y$  and the relative errors. This relative error is defined and calculated by

$$\text{Relative error} = \frac{\sqrt{\Delta x^2 + \Delta y^2}}{4 \times L} \times 100\% \quad (15)$$

where  $L = 50$  mm is the side-length of the square path. The UMBmark calibration parameters were calculated to be  $E_d = 0.9991$  and  $E_b = 0.9953$ . They are some correction factors for wheel diameter and wheel base, respectively. They were computed, based on the results that were obtained in the first trial, in order to decrease the systematic odometry error.\*

In the second trial, the calibrated values of the wheel diameter and base, were applied to odometry and motion command generation process. This calibration

\* Refer to (Borenstein et al., 1996) for more details on UMBmark basics and formulation.

*Table II.* Details of  $\Delta x$  and  $\Delta y$  values (in millimeters) and the relative errors, resulted in each round of the second trial of UMBmark experiment. In this trial, calibration of odometry errors is done by using the parameters  $E_d$  and  $E_b$  that were estimated from the results of the first trial by Borenstein’s method. The error values have been effectively decreased with respect to the case of the first trial where there is no calibration

Round number	Clockwise rounds			Counter-clockwise rounds		
	$\Delta x$	$\Delta y$	Relative error %	$\Delta x$	$\Delta y$	Relative error %
1	15.6	10.0	9.26	11.8	9.7	7.64
2	13.7	12.0	9.11	17.4	15.8	11.75
3	13.4	14.9	10.02	15.1	13.0	9.96
4	15.0	13.4	10.06	13.2	11.1	8.62
5	14.7	12.5	9.65	19.0	12.9	11.48
6	15.7	9.8	9.25	16.1	11.7	9.95
7	17.0	15.7	11.57	14.3	13.6	9.87
8	18.0	10.7	10.44	17.5	12.4	10.72
9	11.8	13.6	9.00	16.1	13.3	10.44
10	18.3	14.2	11.58	13.1	19.7	11.83
11	12.9	13.8	10.81	16.3	13.6	10.61
12	17.2	13.1	10.82	18.8	10.8	10.84
13	12.9	13.6	9.37	16.2	10.0	9.52
14	16.3	12.0	10.12	11.6	15.6	9.72
15	14.2	15.0	10.33	14.5	11.0	9.10
16	18.1	9.8	10.29	13.8	11.3	8.92

caused a lower average deviation of  $|\overline{\Delta x}| = 15.3$  mm and  $|\overline{\Delta y}| = 12.8$  mm, or in other words a relative error of 9.97%. Table II shows the detailed values of  $\Delta x$  and  $\Delta y$  and the relative errors in this second trial.

In the third trial, we applied our proposed calibration method to the pose estimation and motion commands generation. It was observed that the robot traverses the square path more accurately and the average position deviation was measured as  $|\overline{\Delta x}| = 3.2$  mm and  $|\overline{\Delta y}| = 1.6$  mm. This is equivalent to a relative error of 1.8%. It is seen that relative estimation error is decreased by 82%. Table III shows the detailed values of  $\Delta x$  and  $\Delta y$  and the relative errors in the third trial. Table IV shows the average summary of the detailed data, given in Tables I–III.

In the second experiment, the robot was moved from a start point to an end point in a specific route. Two routes are shown in Figure 8. The actual path and the path that was estimated to be traversed by the robot, are depicted by solid and dash line respectively. It is observed that the estimated route gradually deviates. Finally, the end point where was actually reached by the robot is located in a far distance from the estimated end point and following errors were measured:

*Table III.* Details of  $\Delta x$  and  $\Delta y$  values (in millimeters) and the relative errors, resulted in each round of the third trial of UMBmark experiment. In this trial, calibration of odometry errors is done by using Borenstein’s method, along with the method proposed in this paper. The error values have been effectively decreased with respect to the case of the second trial where there is only Borenstein calibration

Round number	Clockwise rounds			Counter-clockwise rounds		
	$\Delta x$	$\Delta y$	Relative error %	$\Delta x$	$\Delta y$	Relative error %
1	3.60	1.70	1.99	2.52	1.66	1.51
2	3.33	1.93	1.91	2.75	1.49	1.56
3	3.38	1.83	1.92	2.92	1.28	1.59
4	3.87	1.44	2.06	3.20	1.87	1.85
5	3.93	1.69	2.14	3.09	1.67	1.76
6	3.65	1.81	2.04	3.32	1.48	1.82
7	3.59	1.39	1.92	3.28	1.90	1.89
8	2.85	1.40	1.59	3.08	1.29	1.67
9	2.42	1.81	1.51	3.41	1.49	1.85
10	3.36	1.62	1.86	2.59	1.70	1.55
11	2.77	1.95	1.69	3.56	1.55	1.94
12	3.38	1.86	1.93	3.02	1.38	1.66
13	3.05	1.57	1.71	3.10	1.45	1.71
14	3.42	1.23	1.82	3.60	1.49	1.95
15	2.76	1.98	1.70	3.64	1.52	1.97
16	2.93	1.92	1.75	3.03	0.88	1.58

*Table IV.* Summary of pose estimation results in the UMBmark experiments. Actually, this table shows the mean values of the data given in Tables I–III

Pose estimation method	$\overline{ \Delta x }$ mm	$\overline{ \Delta y }$ mm	Relative error %
Dead reckoning without calibration	35.5	26.3	22.1
Dead reckoning with UMBmark calibration	15.3	12.8	9.97
Dead reckoning with the proposed calibration method	3.2	1.6	1.8

$$|\Delta x| = |x_{\text{End}} - x_{\text{Estimated end}}| = 269 \text{ mm},$$

$$|\Delta y| = |y_{\text{End}} - y_{\text{Estimated end}}| = 168 \text{ mm}.$$

In the same experiment, the same odometry data were applied to our proposed calibrated version of dead reckoning method and another estimation for the path

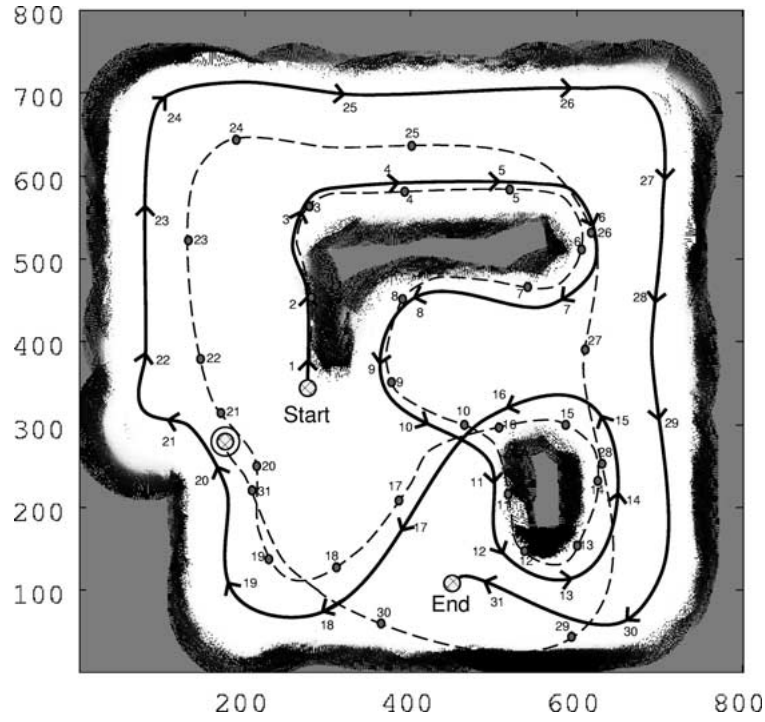


Figure 8. The real and the estimated paths, traversed by the robot are shown by solid and dash lines, respectively. 31 points on the real and estimated paths have been marked by their numbers.

was obtained. Both the actual and the new estimated paths are depicted in Figure 9. In contrast with the previous case, it is observed that the estimated path is much closer to the real one. That is because of the efficient calibration of systematic error of odometry data by using the adapting parameters  $\alpha_{\text{trans}}^*$  and  $\alpha_{\text{rot}}^*$ . Indeed, end position estimation errors were measured as below:

$$|\Delta x'| = |x_{\text{End}} - x_{\text{Estimated end}}| = 25 \text{ mm},$$

$$|\Delta y'| = |y_{\text{End}} - y_{\text{Estimated end}}| = 38 \text{ mm}.$$

Thus, estimation error has been reduced by

$$\left( 1 - \frac{\sqrt{|\Delta x'^2| + |\Delta y'^2|}}{\sqrt{|\Delta x^2| + |\Delta y^2|}} \right) \times 100\% = 85.66\%.$$

In Figures 8 and 9, 31 points on the real path and their estimated location on the estimated path are specified and marked by their numbers. The error values  $|\Delta x|$ ,  $|\Delta y|$  and  $\Delta r$  have also been calculated. Table V shows the detailed results for all of these points, including the start and end points.

Table V. Details of the error distance between each of the points on the real and estimated paths in Figure 8 (pose estimation, using UMBmark method calibration) and Figure 9 (pose estimation, using the proposed method). The term  $\Delta l$  means the absolute distance between the points on the paths. It is calculated by:  $\Delta l = \sqrt{\Delta x^2 + \Delta y^2}$

Point number	UMBmark calibration			Proposed method		
	$ \Delta x $ (mm)	$ \Delta y $ (mm)	$\Delta l$ (mm)	$ \Delta x $ (mm)	$ \Delta y $ (mm)	$\Delta l$ (mm)
Start point	0	0	0	0	0	0
1	0.0	0.0	0.0	0.0	0.0	0.0
2	4.8	1.4	5.0	4.5	2.4	5.1
3	10.5	8.6	13.6	8.9	18.8	20.8
4	10.5	10.0	14.5	8.9	23.8	25.4
5	16.2	10.0	19.1	18.8	23.8	30.3
6	13.3	31.9	34.6	12.9	26.8	29.7
7	43.3	13.3	45.4	15.9	4.0	16.4
8	15.7	1.9	15.8	16.8	1.0	16.8
9	13.8	23.8	27.5	5.9	19.8	20.7
10	46.2	2.8	46.3	13.9	15.8	21.0
11	16.7	15.2	22.6	0.0	16.8	16.8
12	26.7	1.4	26.8	17.8	1.0	17.9
13	10.0	40.0	41.2	9.9	16.8	19.5
14	24.3	16.7	30.7	7.9	11.9	14.3
15	43.8	9.0	44.7	11.9	15.8	19.8
16	12.4	25.7	28.5	20.8	4.9	21.4
17	5.2	34.3	34.7	18.8	4.0	19.2
18	15.1	52.2	54.4	17.8	33.7	38.1
19	48.1	30.9	57.2	8.9	43.6	44.5
20	46.7	3.4	46.8	7.9	39.6	40.4
21	60.4	8.9	61.1	16.8	34.7	38.5
22	67.4	4.0	67.5	12.9	17.8	22.0
23	52.5	39.6	65.8	14.9	11.9	19.1
24	87.2	50.5	100.8	20.8	0.0	20.8
25	85.2	63.4	106.2	16.8	7.9	18.6
26	26.8	175.4	177.4	15.8	3.0	16.1
27	95.1	209.1	229.7	0.0	35.7	35.7
28	64.4	198.2	208.4	12.9	10.9	16.8
29	105.0	265.6	285.6	10.9	35.7	37.3
30	298.8	5.9	299.3	53.5	5.9	53.8
31	284.4	112.9	306.0	20.8	20.8	29.5
End point	269.0	168.0	317.1	25.0	38.0	45.5

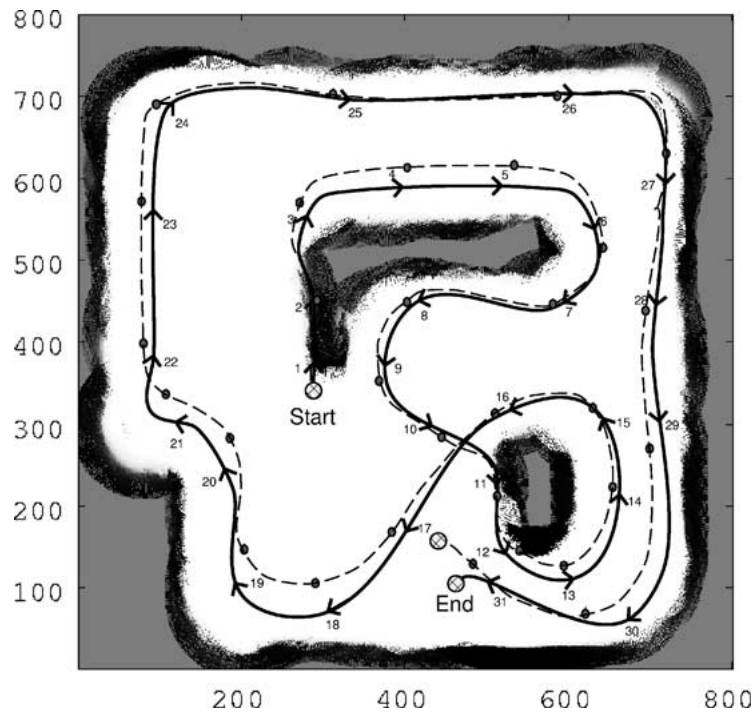


Figure 9. The real and the estimated paths, traversed by the robot are shown by solid and dash lines respectively. In this case, our proposed calibration method has been applied to the pose estimation to generate the estimated path. That is why it is much closer to the actual path. 31 points on the real and estimated paths have been marked by their numbers.

## 5. Conclusions

A new approach for calibration of dead reckoning process in pose estimation of mobile robots was proposed in this paper. We attempted to model the systematic part of the error, existing in odometry data, by introducing two calibration parameters and estimating them adaptively. It was suggested to fuse the current sensory and odometry information with the information that had been obtained previously. The local map, generated by current sensory data, is integrated with a local map that is extracted from a global occupancy grids map of environment. Actually, the most desired parameters are those which lead to a mostly fitted version of an extracted local map. It was shown that the proposed approach gives a maximum likelihood estimation of the parameters. The most important advantages of the method are its simplicity and its applicability to the cases where the robot is equipped with short range distance sensors (proximity detectors). In our experiments, Khepera robot was utilized to examine the performance of the method for reduction of pose estimation error. The proximity data, provided by the infra-red sensors of the robot and the odometry data, provided by wheel encoders, were the only sources of information in these experiments. Results show that our

approach causes a reduction of more than 80% in pose estimation error. Besides, the proposed approach is applicable in an online case. It means that the robot may calibrate its position by this method, while it is exploring the environment around itself and mapping it. In this case, the important difference is the fact that there is no previously created global map for extraction of local maps. But it seems very probable that a partially generated map can be useful to be applied in our method. Meanwhile, although the environment was static in our experiments, the proposed method will be useful for calibration of odometry errors in a dynamic environment, if the environment does not change rapidly. That is because when the environment changes, the instantaneous sensory data change. So, the local map generated by the neural network from the current sensory information changes. On the other hand, the global map which has been upgraded by the recent sensory information (obtained in the recent previous iteration of sensing and local map generation), changes. In our calibration method, the fitness of the local map generated from the current sensory data and the local map, extracted from the global map, is the main quantity applied for error calibration by parameter estimation. Consequently, if the environment variation is slow enough in such a way that it does not effectively change during the time interval between two successive iterations, our method will also work in a satisfying manner for dead-reckoning error calibration in dynamic environments.

### **Acknowledgements**

This research work was partially supported by the School of Intelligent Systems (SIS) (IPM, Tehran, Iran) under contract no. 0180-3 and also financially supported by the ministry of industries and mines under contract no. 782015168. It was also supported through mutual research program between University of Tehran, Faculty of Engineering, Department of Electrical and Computer Engineering (Iran) and University of the Ryukyus, Faculty of Engineering, Department of Information Engineering (Japan) by Monbusho scholarship scheme.

### **References**

- Abidi, A. and Gonzalez, R. C.: 1992, *Data Fusion in Robotic and Machine Intelligence*, Academic Press, New York.
- An, C. H., Atkeson, C. G., and Hollerbach, J. M.: 1988, *Model Based Control of a Robot Manipulator*, MIT Press, Cambridge, MA.
- Asharif, M. R., Moshiri, B., and HoseinNezhad, R.: 2000, Pseudo information measure: A new concept for sensor data fusion, applied in map building for mobile robots, in: *Proc. of Internat. Conf. on Signal Processing Applications and Technology (ICSPAT 2000)*, Dallas, TX, USA, 16–19 October.
- Asharif, M. R., Moshiri, B., and HoseinNezhad, R.: 2001, Environment mapping for Khepera robot: A new method by fusion of pseudo-information measures, in: *Proc. of the 6th Internat. Symposium on Artificial Life and Robotics (AROB 06)*, Tokyo, Japan, 15–17 January, pp. 305–308.

- Borenstein, J., Everett, H. R., and Feng, L.: 1996, Where am I? Sensors and methods for mobile robot positioning, Technical Report, University of Michigan, <ftp://ftp.eecs.umich.edu/people/johannb/pos96rep.pdf>.
- Braitenberg, V.: 1984, *Vehicles*, Kluwer Academic, Dordrecht.
- Cox, I. J. and Wilfong, G. T.: 1990, *Autonomous Robot Vehicles*, Springer, Berlin.
- Crowley, J. L.: 1995, Mathematical Foundations of navigation and perception for an autonomous mobile robot, in: L. Dorst, M. Van Lambalgen, and F. Voordraak (eds), *Reasoning with Uncertainty in Robotics*, Lecture Notes in Artificial Intelligence, Vol. 1093, Springer, Berlin, pp. 9–51.
- Dasarathy, B.: 1994, *Decision Fusion*, IEEE Press, New York.
- Elfes, A.: 1987, Sonar-based real-world mapping and navigation, *IEEE J. Robotics Automat.*, 249–265.
- Elfes, A.: 1989, Using occupancy grids for mobile robot perception and navigation, *Computer* **22**(6), 249–265.
- Foley, J., Van Dam, A., Feiner, S., and Hughes, J.: 1990, *Interactive Computer Graphics: Principles and Practice*, Addison-Wesley, Reading, MA.
- Hwang, Y. K. and Ahuja, N.: 1992, Gross motion planning – A survey, *ACM Computing Surveys* **24**(3), 219–291.
- K-Team, S. A.: 1998, Khepera User Manual (5.0 ed.), Lausanne, Switzerland.
- MacKenzie, P. and Dudek, G.: 1994, Precise positioning using model-based maps, in: *Proc. of IEEE Internat. Conf. on Robotics and Automation*, San Diego, CA, May, pp. 1615–1621.
- Mondada, F., Franzi, E., and lenne, P.: 1993, Mobile robot miniturization: A tool for investigation in control algorithms, in: *Proc. of the 3rd Internat. Symposium on Experimental Robotics*, Japan, October, pp. 501–513.
- Moshiri, B., Eydgahi, A., Najafi, M., and HoseinNezhad, R.: 1999, Multi-sensor data fusion used in intelligent autonomous navigation, in: *IASTED – CA '99 (Control Applications)*, Banff, Canada, 25–29 July, pp. 515–520.
- Murphy, R.: 1998, Dempster–Shafer theory for sensor fusion in autonomous mobile robots, *IEEE Trans. Robotics Automat.* **14**(2), 197–206.
- Oriolo, G., Ulivi, G., and Vendittelli, M.: 1997, Fuzzy maps: A new tool for mobile robot perception and planning, *J. Robotic Systems* **14**(3), 179–197.
- Richardson, J. M. and Marsh, K. M.: 1988, Fusion of multisensor data, *Internat. J. Robotics* **7**(6), 78–96.
- Thrun, S.: 1995, An approach to learning mobile robot navigation, *Robotics Automat. Systems* **15**, 301–319.
- Thrun, S.: 1998, Learning metric-topological maps for indoor mobile robot navigation, *Artificial Intell.* **99**(1), 21–71.
- Thrun, S., Burgard, W., and Fox, D.: 1998, A probabilistic approach to concurrent mapping and localization for mobile robots, *Machine Learning* **31**, 29–53.
- Van Dam, J. W. M.: 1998, Environment modeling for mobile robot: Neural learning for sensor fusion, PhD Thesis, University of Amsterdam.