Localization and Mapping of Cheap Educational Robot with Low Bandwidth Noisy IR Sensors

Muhammad Habib Mahmood[1,2] and Pere Ridao Rodriquez[2]
[1]*Department of Electrical Engineering, Air University, Islamabad, Pakistan*
[2]*Computer Vision and Robotics (ViCOROB) Institute, University of Girona, Girona, Spain*
habib@mail.au.edu.pk, mhabib@eia.udg.edu

Keywords: Localization, Mapping, Kalman Filter, Particle Filter, Occupancy Grid Mapping, Monte-Carlo Localization.

Abstract: The advancements in robotics has given rise to the manufacturing of affordable educational mobile robots. Due to their size and cost, they possess limited global localization and mapping capability. The purpose of producing these robots is not fully materialized if advance algorithms cannot be demonstrated on them. In this paper, we address this limitation by just using dead-reckoning and low bandwidth noisy infrared sensors for localization in an unknown environment. We demonstrate Extended Kalman Filter implementation, produce a map of the unknown environment by Occupancy grid mapping and based on this map, perform particle filtering to do Monte-Carlo Localization. In our implementation, we use the low cost e-puck mobile robot, which performs these tasks. We also putforth an empirical evaluation of the results, which shows convergence. The presented results provide a base to further build on the navigation and path-planning problems.

1 INTRODUCTION

Localization in robotics is considered an old problem. Many methods have been explored to formulate a solution to this problem but as it is dependent on the surrounding environment, all the solutions differ under varying environmental conditions. Moreover, besides the method, the result of localization is also dependent on the sensors. While, the use of expensive sensors with sophisticated hardware is necessary for industrial use, the approach changes completely, if the purpose is to use the robots for educational purposes.

Recently, researchers have shown particular interest in low cost educational robots. Recent works (S. Wang and Magnenat, 2016; S. Bazeille and Filliat, 2015) show that their has been a growing trend of trying to accommodate accessible low budget robots in the academic environment. This approach has been tried for mapping and navigation (S. Bazeille and Filliat, 2015), multi-robot systems (J. McLurkin and Bilstein, 2013), collaborative robots (A. Prorok and Martinoli, 2012) etc. This trend finds its roots from the idea that complex algorithms can be implemented in low budget machines.

The robots used in education are build to be inexpensive, containing low cost sensors with repeatable motor drives. The sensors mounted on these machines are highly susceptible to environmental noise. The purpose of these robots is to visually demonstrate theoretical concepts. If these robots are used in environments where the effect of ambient sources adds to the noise in sensing, then their repeatable utilization suffers. On top of it, if advance robotics' concepts are tried to be tested, the robots tend to fail.

In this paper, we approach the problem of robustly implementing complex localization and mapping algorithms on an inexpensive e-puck robot with low bandwidth infrared (IR) sensors in a probabilistic framework. Utilizing dead-reckoning and overhead camera, we perform Extended Kalman-Filter Localization. Then, by using the low bandwidth IR noisy sensor, we devise a map of an unknown environment by the Occupancy Grid Mapping algorithm. The acquired map is then utilized in the implementation of particle filtering by Monte-Carlo Localization. A detailed evaluation of the quantitative results of these algorithms with convergence are also shown. It can also be seen that the error in localization for all implementations remains with in 3σ bound at all times.

2 RELATED WORK

The advent of robotics has given rise to several offshoots. Among these, recent trends by researchers

Mahmood, M. and Rodriguez, P.
Localization and Mapping of Cheap Educational Robot with Low Bandwidth Noisy IR Sensors.
DOI: 10.5220/0006204305830590
In *Proceedings of the 6th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2017)*, pages 583-590
ISBN: 978-989-758-222-6

suggests that there has been a growing interest in using educational robots to implement and demonstrate complex robotics algorithms. In an implementation, a mapping robot using RP Lidar scanner was used (M. Markom and Shakaff, 2015). Though, the implementation is for an indoor robot, the Lidar sensor size makes its usage difficult in small sized educational robots. Motion and pose tracking for mobile robots was also proposed using open-source off-the-shelf hardware and software (A.D. Cucci and Matteucci, 2016), along with an experimental benchmark protocol for indoor navigation (C. Sprunk and Jalobeanu, 2016). These implementations demand high computational overheads, which might again be difficult to sustain in a small educational robot.

On a different note, a localization approach using fixed remote surveillance cameras was also proposed (Shim and Cho, 2016). However, this implementation can only be tested on a high cost design. Another localization and mapping implementation, with unknown noise characteristics, presented an application of Simultaneous Localization and Mapping (SLAM) (Ahmad and Namerikawa, 2010). This implementation was also computationally extensive, unsuitable for educational robots.

Several dedicated educational robots have also been presented. e-puck was presented as a multi-sensor table-top small mobile robot (F. Mondada and Martinoli, 2009). Another one was Roomba, which was a uniquitous mobile robot platform (Tribelhorn and Dodds, 2007). Similar frameworks for low cost robot implementations with different approaches were presented as collaborative localization (A. Prorok and Martinoli, 2012), multi-robot system (J. McLurkin and Bilstein, 2013), mapping and navigation (S. Bazeille and Filliat, 2015) and localization (S. Wang and Magnenat, 2016).

A repeatable implementation of these advance algorithms in low cost robots is a limitation, which persists in these implementations. To overcome this, we propose the implementation of EKF, OGM and MCL in e-puck mobile robot. In the remaining sections, section 3 explains the localization and mapping problem, section 4 describes the Experimental setup and results, followed by a discussion in section 5 and conclusion in section 6.

3 LOCALIZATION AND MAPPING

The localization and mapping algorithms were implemented in the e-puck mobile robot. Initially, the localization problem was addressed by Extended Kalman Filter Localization (EKF Localization). Then, by using the noisy IR sensors, Occupancy Grid Mapping of the unknown arena was performed. Finally, the map was used in a particle filter framework to perform Monte-Carlo Localization.

3.1 EKF Localization

Extended Kalman Filter is an implementation of Bayes Filter, which allows formulating the localization problem as a Bayesian estimation problem. The Bayes filter is derived from the Bayes Theorem and it relates the conditional probabilities and the prior probabilities of two events. EKF Localization algorithm presented in (S. Thrun and Fox, 2005) is an extension of the Kalman Filter (KF). KF's assumption of linearity is rarely fulfilled in practice and EKF overcomes this assumption of linearity by considering that state transition probability and/or the measurement probability can be non linear functions. The posterior for true belief was estimated by approximating a Gaussian, which was calculated using linearization approximation. This got rid of transformation matrices A_k and B_k and resulted in Jacobians G_k and H_k as described in (S. Thrun and Fox, 2005).

Algorithm 1: EKF Localization.

procedure EKF($\hat{x}_{k-1}$,P_{k-1},u_k,z_k)
  $\hat{x}_k^- = f(\hat{x}_{k-1}, u_k, 0)$
  $P_k^- = A_k P_{k-1} A_k^T + W_k Q_k W_k^T$
  $K_k = P_k^- H_k^T (H_k P_k^- H_k^T + V_k R_k V_k^T)^{-1}$
  $\hat{x}_k^- + K_k(z_k - h(\hat{x}_k^-, 0))$
  $P_k = (I - K_k H_k) P_k^-$
  return (x_k, P_k)

In Figure 1, the robot moved from Pose 1 in frame $\{R-1\}$ to Pose 2 in frame $\{R\}$. The robot moved by the vector x_{k-1}^k given in robot frame $\{R-1\}$. This vector had to be compounded with x_{k-1}^B to get x_k^B. The compounding function was a non linear function, which also had noise affect in it. This effect was catered for by the inclusion of w_k, which was the noise associated with odometry (dead-reckoning), having zero mean and a covariance matrix Q_k. As the cross covariance of the state variable was zero, the Q_k matrix became a diagonal matrix. In the measurement step, the robot pose with respect to the global frame was estimated with the help of an overhead camera z_k.

The update step was performed by using the mean and covariance generated in the prediction step in conjunction with the measurement z_k and its noise v_k. In the update step, first the Kalman Gain K_k was calculated, using the value of V_k. Then, the innovation

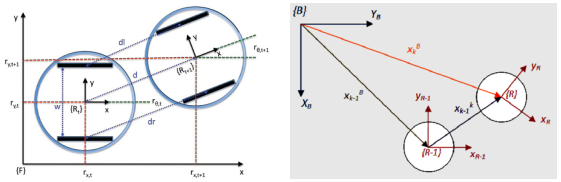

Figure 1: **Left:** The inexpensive e-puck mobile robot used in the experiments. **Middle:** The model for odometry computations in e-puck. Here, motion between two consecutive time instants is show. **Right:** The model for rigid body transformations for e-puck, based on position and heading direction in two consecutive time instants.

step, the difference between the measured global position of the robot and the estimated global position of the robot, $z_k - z_k^B$, was calculated. By these calculations, the mean x_k and covariance P_k were updated.

3.2 Occupancy Grid Mapping

Occupancy Grid Mapping (OGM) has become one of the dominant paradigms for environmental modeling in mobile robotics (P. Schmuck and Zell, 2016; Colleens and Colleens, 2007; D. Kortenkamp and Murphy, 1998). OGM addresses the mapping problem under the assumption that the robot pose is known. A 2D floor plan OGM map describes a 2D slice of the 3D world. The basic idea was to represent the map as a field of random variables known as *cells*, arranged in an evenly spaced grid. Each cell in the 2D grid contained quantitative information regarding the environment. OGM algorithm updates the posterior over each grid cell as being occupied, empty or unknown (Elfes, 1989) based on approximate posterior estimation. The implemented mapping algorithm updated the cells based on inverse sensor model algorithm (S. Thrun and Fox, 2005).

Algorithm 2: Occupancy Grid Mapping.

procedure OGM($\{l_{t-1,i}\}, x_t, z_t$)
  for all cells m_i do
    if m_i in perceptual field of z_i then
      $l_{t,i} = l_{t-1,i}$+IRSM $(m_i, x_t, z_t) - l_0$
    else
      $l_{t,i} = l_{t-1,i}$
    endif
  endfor
  return $(l_{t,i})$

OGM used log odds for the representation of occupancy. The odds of a state x were defined as the ratio of the probability of that event divided by the probability of its complement. The advantage of log-odds over the probability representation was that they can avoid numerical instabilities for probabilities near

Algorithm 3: inverse_range_sensor_ model.

procedure IRSM(i, x_t, z_t)
  let x_i, y_i be the center of mass of m_i
  $r = \sqrt{(x_i - x)^2 + (y_i - y)^2}$
  $\phi = atan2(y_i - y, x_i - x) - \theta$
  $k = argmin_j |\phi - \theta_{j,sens}|$
  if $r > min(z_{max}, z_t^k + \alpha/2)$ or $|\phi - \theta_{j,sens}| > \beta/2$ then
    return l_0
  if $z_k < z_{max} and |r - z_{max}| < \alpha/2$
    return l_{occ}
  if $r \leq z_t^k$
    return l_{free}
  endif

zero and one. The algorithm had previous log-odds l_{t-1}, current robot pose x_t and measurement z_t as input. It looped through all grid cells and updated those cells, which fall into the sensor cone of the measured z_t. The update of the occupancy log-odds value was done by using the inverse range sensor model algorithm. The constant l_0 is the prior of occupancy represented as a log-odds ratio.

The inverse sensor model used for IR range sensors assigned an occupancy value l_{occ} to all the cells inside the measurement cone. The width of the occupied region was controlled by the value α and the opening angle of the sensor beam was given by β. The inverse model first calculated the beam index k and the range r for the center of mass of the cell m_i. The prior for occupancy in log-odds form l_0 was returned if the cell was outside the measurement cone or if it lied more than $\alpha/2$ distance behind the detected range z_t. It returned l_{occ} if the range of the cell was within $\alpha/2$ of the detected range z_t. It returned l_{free} range of the cell was shorter than the measurement range by more than $\alpha/2$.

3.3 Monte-Carlo Localization

Monte Carlo Localization (MCL) represents the posterior using a particle set, which makes it a non parametric implementation of the Bayes filter. Being non-parametric, it can represent a much broader space of distributions. Each particle is an instance of the state at time t, which is a hypothesis of the actual state. The measurement model assigns weights the estimated particles hypotheses making sure of the survival of the particles closer to the true state. The selection of an appropriate measurement model is an important step.

The MCL algorithm was obtained by substituting the appropriate probabilistic motion model and measurement model in the particle filter algorithm. The

Algorithm 4: Monte-Carlo Localization.

procedure MCL(χ_{t-1}, u_t, z_t, m)
  $\bar{\chi}_t = \chi_t = \emptyset$
  for $m = 1$ to M do
    $x_t^{[m]} =$ sample_motion_model$(u_t, x_{t-1}^{[m]})$
    $w_i^{[m]} =$ measurement_model$(z_t, x_t^{[m]}, m)$
    $\bar{\chi}_t = \bar{\chi}_t + < x_t^{[m]}, w_t^{[m]} >$
  endfor
  for $m = 1$ to M do
    draw i with probability $\propto w_t^{[i]}$
    add $x_t^{[i]} to \chi_t$
  endfor
  return χ_t

inputs of the algorithm were a particle set χ_{t-1}, odometry u_t, the most recent measurement z_t and an occupancy grid map m of the environment. The algorithm represents the belief $bel(x_t)$ as a set of M particles $\chi_t = x_t^{[1]}, x_t^{[2]}, ..., x_t^{[M]}$. Each particle x_t was an instance of the state at time t, which was a hypothesis of the actual state. The state transition probability density $p(x_t|x_{t-1}, u_t)$ was updated by using a sample motion model. Then, the importance factor, $w_t^{[m]}$ for each particle $x_t^{[m]}$ was calculated. The importance factor incorporates the measurement z_t in the particle set χ_t. It calculated the probability of measurement z_t under the updated particle set χ_t. The predicted particle set and the normalized measurement weights were then used to resample the particle set through importance resampling. Here, the probability of drawing each particle was given by its importance weight. The three most important steps of the algorithm were sample motion model, measurement model and the importance resampling algorithm. The sample motion model incorporated the motion u_t of every particle in the particle set. Each particle was updated with some different random noise to model the uncertainty of the environment. The importance resampling step made use of the low variance sampler (S. Thrun and Fox, 2005). The measurement motion model used in this application was the correlation based sensor models.

4 EXPERIMENTAL SETUP

To conduct the experiments of the above mentioned algorithms, we used an 800mm by 600mm 2D wooden field, an overhead camera covering the whole field in its field of view (FOV) and an educational mobile robot, e-puck.

The 2D rectangular field was placed in the laboratory where ample ambient light was present in the

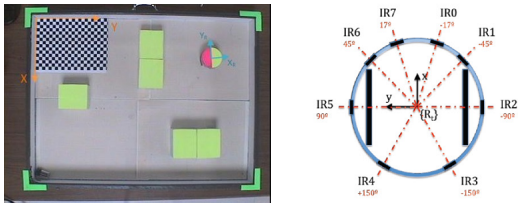

Figure 2: **Left:** A 2D planar field with e-puck mobile robot. Global coordinate frame in Orange and Robot coordinate frame in Blue. The checkers board calibration patten for extrinsic parameters in on the arena as well. **Middle:** The orientation of IR sensors on the e-puck is shown. **Right:** The e-puck robot in the arena with its IR rays extended upto 6cms.

surroundings, shown in Figure. 2. The area encompassed by the field can be identified with four 'L' shaped dark green markers pasted on each corner of the field. While moving around the field, the robot could encounter two types of walls. The wall of obstacles (OW), which had green markers on top and the boundary walls of the field (FW). Both types of walls were painted white so that maximum Infrared rays could get reflected off them. The size of obstacles was greater than the size of the robot. The field was relatively small and movable.

The overhead camera was a *SONY SSC-DC198P* and it was fixed on top of the field. The calibration of the camera was done by using Faugeras-Tosconi pinhole model, implemented in Matlab's Camera Calibration Toolbox. In total, 18 checker board images in various orientations were used to get the intrinsic parameters. The extrinsic matrix was calculated by using the already calculated intrinsic parameters and the checker board's image placed beside the global origin of the field as shown in Figure. 2. As the field was movable, the extrinsic matrix had to be re-calculated before the start of operation each time.

The mobile robot used for experimentation was e-puck, which was a small, cheap and resourceful (F. Mondada and Martinoli, 2009). Overall three basic utilities of e-puck were used: motors for motion, encoders for odometry and Infrared Range (IR) sensors for range sensing. Throughout the experiment, the two stepper motors of the robot were moved with constant linear and angular velocity. The encoder pulse count of each wheel, when calculated with the wheel circumference and axle length, gave an estimate of the robot pose in the robot's coordinate frame. This estimation of pose is known as odometry, u_k. The 8 IR sensors, orientations shown in Figure. 2, present on e-puck were used to calculate the distance of obstacles from the robot. The effective range of these IR sensors was a maximum of 6 cm with a very narrow opening angle. This meant that an IR sensor

detects an obstacle only when the IR beam was almost perpendicular to the obstacle.

4.1 Results

Dataset. To be able to work offline, a dataset of e-puck's motion around the field was created storing all the desired variables. The path followed by the robot covered the whole field many times so that there was enough data for validation as well. For the implementation of each algorithm, the desired variables were taken from the dataset and evaluated.

EKF Localization. The center of the field was taken to be the initial belief with uncertainty in the covariance matrix covering the whole arena shown in Figure. 3. The prediction step took into account the odometry control input. The mean of belief was updated by compounding the computed odometry with the previous belief. The compounding function was a non linear function, which also had noise affect catered in it.

For every iteration in the measurement step, an image was taken by the camera and processed using the calibration parameters to identify the robot pose with respect to the global coordinate frame. The algorithm performed color segmentation and then calculated the centroid of the robot. This estimated robot pose, with its associated covariance matrix, was then used to update the robot position mean and covariance. Colored semi-circular markers were used to determine the orientation (heading direction), of the robot. In this case, where an overhead camera was used for measurement, the measurement equation was formulated by a linear equation. The update step was performed by using the mean and covariance generated in the prediction step in conjunction with the measurement and measurement noise. The update strengthens the robot belief around the true robot pose shown in Figure. 3. Three iterations 1st, 16th and 76th are shown in the figure. It can be seen that as the number of iteration increases, the robot belief strengthens around its true location. It can be visually verified that with the increase in the number of iterations, the Gaussian of robot belief grows narrower, showing that the robot is confident about its estimated pose. If at any time the update is stopped, the robot belief still follows the robot around but the uncertainty region starts expanding with respect to the standard deviation in odometry.

Occupancy Grid Mapping. Occupancy grid map of the environment inside the field was created by using the IR sensors on the mobile robot e-puck. The image overlay of the map is shown on top of the images captured by the overhead camera. The measured range from each IR sensor was mapped by using the OGM algorithm to retrieve a local map. In the implemented algorithm, the IR sensor readings were taken to be deterministic, which means the measured light falling on the receptor of the sensor was converted into distance and was taken as the true value of distance. This was an extraordinary assumption and should be taken care of in the future.

In the presented result, the grid cell size in the software map was 1mm and the real size of the field was 600 mm x 800 mm. The images representing the results had the same pixel ratio, with walls of the field extending from each side. That meant that a pixel in the image corresponds to 1mm in distance. When evaluating the sharpness of edges in the map, this measure is very important. The robot was made to move beside the obstacles in the free space around the field several times in the collected dataset so that enough data is available for the mapping step. The result showed that not all the walls of obstacles and the field are updated, there are discontinuities present in them. As the grid cell size was too small, the noise in measurement affected the grid cell update a lot. This will not be the case if the resolution is higher, as then the measurement to noise ratio will differ.

In Figure. 5, the back converted log odds map, into probability map, is shown. All the values in the map fall between 0 and 1. There are three colors in gray scale, which identify the different type of grid cells. These can be identified visually, as gray color showing the grid cells, which were not updated and are unknowns, the white colored grid cells representing the obstacles and the black colored grid cells representing the free space.

The result of OGM is shown on top of the ground truth. In Figure. 5, left image, only the grid pixels identified as occupied are plotted in white on top of the image with one to one correspondence in millimetre scale. It can be visually verified that the image fits perfectly on top of the obstacles and the boundary walls. In addition to the occupied cells, the cells which were classified as unknowns are also pasted on top of the ground truth in black color. Although most of the cells were correctly classified, there were other cells, which are actually free but are classified as unknowns. This happens because of the small opening angle of the IR beam. It can easily be removed in a post processing step, where all these cells in the field except for obstacles, would be classified as free under the condition that a single clot of unknown cells is smaller than the size of the robot. This can only be done if the size of the

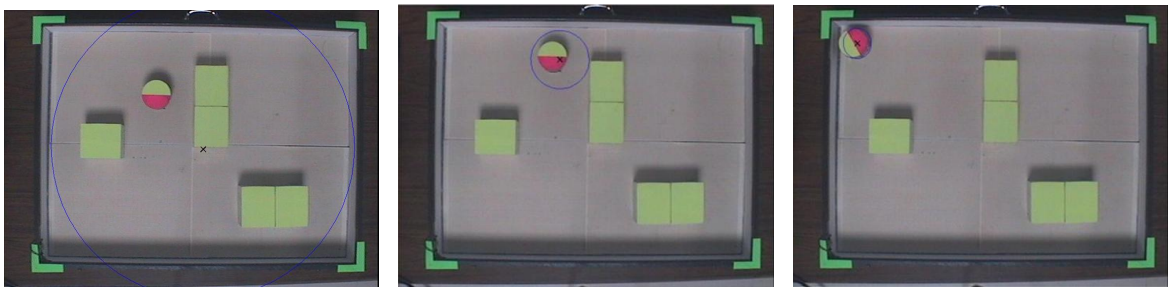

Figure 3: **Left:** The 1st iteration. The mean of initial belief of EKF is in the center of the field represented with a cross 'x' and the covariance ellipse covers the whole field shown in 'blue circle'. **Middle:** The 16th iteration. The mean in the updated belief is converging towards the true robot pose in subsequent iterations. **Right:**The 76th iteration. The belief grows even more strong in the subsequent iterations. The covariance ellipse is smaller, which shows that the algorithm has converged.

obstacle is assumed to be greater than the size of the robot, which was the case in our implementation. The presented results are a good approximation of the actual map.

Monte-Carlo Localization. Monte-Carlo localization was implemented by using the occupancy grid map generated by OGM and the IR sensors of the e-puck. The inputs of the algorithm were a particle set χ_{t-1}, odometry control input u_t, the most recent measurement z_t and an occupancy grid map m of the environment. It first samples the motion model, then calculates weights by measurement model and then resamples the particle set.

In the resampling step, the low variance sampler generates a single random number and selects samples according to this number but, still, with a probability proportional to the sample weight. The result is a resampled particle set, with the density of particle presence increased in the areas, where the particles had more weight. A drawback of resampling is that if it is done too often the diversity in the particle set vanishes and if it is too infrequent, then the particle samples are wasted in regions of low probabilities. A rule of thumb to solve this problem is to check the particle weight covariance, *Neff*, and resample, if it is bigger than a threshold, *Nth* taken to be M/2.

In the presented results, the number of particles M was 1000 so *Nth* was 500. In the first experiment, resampling was done in every step and Neff was not used. After just the first resampling step, it can be seen in Figure. 4 that the particles beside the FW and OW, whose local map intersected with obstacles on the global map, got minimum weight and hence were not resampled. This happens because in measurement z_k no obstacle was detected.

In the subsequent iterations, the robot kept on moving along its path with particles oriented randomly in the free space until an obstacle was detected. In this case, after resampling, only the particles which were oriented along FW or OW remain as in Figure. 4. The robot moved along with the FW on its right side. All the particles in the field moving in a pattern similar to robot motion got equivalent weightage. As the robot arrived closer to the wall perpendicular to it, this act was another feature for localization. The particles, which do not agree with robot motion, start losing weight and the accumulation of particles near the true robot pose continued. Thus, eventually only one belief is left which was in agreement with the true robot pose, as can be seen by the covariance ellipse of the particle cluster in Figure. 4. In all the subsequent robot motion, the mean, x, of particle cluster remained close to the true robot belief with the error being inside the 3σ bound as depicted in Figure. 7.

In the second type of results presented, *Neff* based resampling was performed. This enabled diversity in the particle set, without adding too many particles of low probability. The algorithm was initialized in the same manner with uniformly distributed random particles. When the weight covariance was less than a threshold, the algorithm instead of resampling multiplied the recently calculated weights with the weights of the previous step. This made sure that the particles with more plausible pose get more weight and at the same time the particles with low probability should not vanish in just one iteration. When the robot approached the FW, the robot belief around the true robot position strengthened. At the same time, other plausible beliefs were also kept. This was the major difference in *Neff* based MCL that it keeps a track of all the possible beliefs till very late, shown in Figure. 6. As the robot approached further unique positions near both, the FW or the OW, the belief around the true robot position grew until all the other hypotheses disappear, and all the particles converge to the true robot belief shown in Figure. 6.

It should be noticed that in Figure. 6, it is the 201 iteration of the robot motion sequence. It is very different in comparison with results presented for the

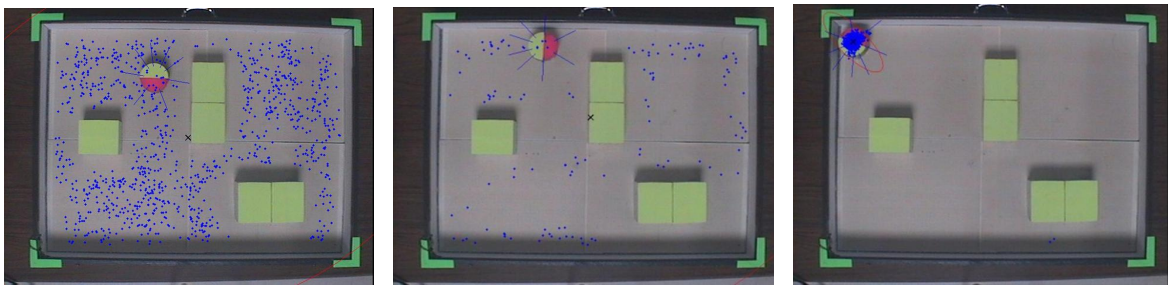

Figure 4: These results are without Neff. **Left:** MCL result in 1st iteration. MCl is initialized with particles randomly oriented. **Middle:** The 21st iteration. The particles resample. The particles are not converging because of the self similariy of the arena. **Right:** The 76th iteration. MCL is converging, as along the walls and then around the corner, the self similarity is minimum and the particles converge.

Figure 5: **Left:** Occupancy Grid Map of the environment shown in grayscale, with white showing the obstacles, black representing the free cells and gray color showing the unknown parts of the field. **Right:** Occupancy Grid Map registered on top of the ground truth with obstacles in white and unknowns in black. It is a 1 to 1 scale mapping between the arena area and the IR sensed distances, in millimeters.

first experiment, where the robot had converged even in the 80^{th} iteration. This result gives a good idea as to how important it to consider weight covariance for making a decision about resampling. It makes sure that all the different hypotheses generated by the particle set are given their due importance until it substantially establishes the most suitable one. This kind of particle diversity is important in self similar environments like the one being considered for these experiments. Here again in all the subsequent robot motion, the mean, x, of particle cluster remains close to the true robot belief with the error being inside the 3σ bound as depicted in Figure. 7.

The error in x and y coordinates of the ground truth against the particle cluster mean is shown in Figure. 7. One set of results is without *Neff* and one with it. It can be seen that when *Neff* is used, the particles follow the ground truth more steadily, with less error in each axis. In both cases, the error in estimation is bounded by 3σ range but the change in standard deviation is not as rapid in *Neff* as it is without it, which makes the use of *Neff* more desirable.

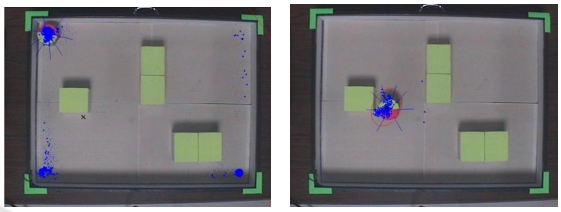

Figure 6: These results are with Neff. **Left:** MCL result with *Neff* in 80^{th} iteration, which is very different from the one shown earlier in Figure. 4. **Right:** MCL result with Neff in 201st iteration. This is when the particle set has converged to the true robot belief, much later than the one without Neff.

5 DISCUSSION

The presented results were attained with the limited resources of the educational e-puck robot. The robot had two limitations regarding range sensing. The first is that the field of view in range sensing was limited as IR sensors have a very narrow beam of measurement and there were no other range sensors available. The second limitation was that the accuracy in IR range sensing is not good because it is easily tempered by ambient light. This makes the conversion of light reading into distance inconsistent, which makes the whole mapping and localization problem unreliable. Another drawback is that IR sensors only detect an obstacle when the obstacle wall is almost perpendicular to the IR beam. This means that even when an obstacle is present in the IR sensor measurement range, it does not detect it because the beam is not perpendicular. This makes the sensor measurement more unreliable and makes the localization problem that much more difficult. Nonetheless, it can be seen that even in the presence of noise, and low cost IR sensors, the robot was able to localize itself in an unknown environment. It is definitely a leap in terms of the development of educational robots.

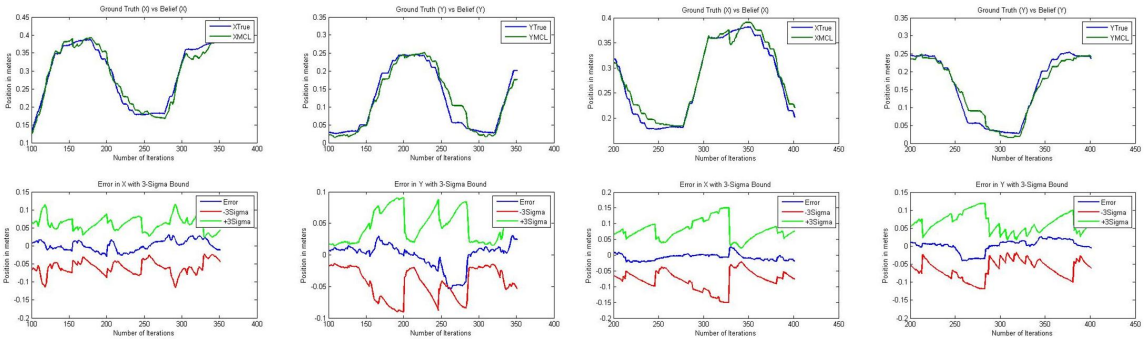

Figure 7: Each column pair contains (x, y) position of the robot. **Left two columns:** Error with 3σ bound without *Neff*. **Right two Column:** Error with 3σ bound with *Neff*.

6 CONCLUSIONS

The work presented in this paper showed an inexpensive educational robots with low cost noisy sensors being used to demonstrate complex localization and mapping concepts. We were able to implement and demonstrate different techniques of solving two of the major problems of mobile robotics i.e. *Localization and Mapping*. The methods implemented were EKF Localization, Occupancy Grid Mapping and MCL. In the paper, it was demonstrated that these complex algorithms can be implemented in a real environment with a cheap robot. A natural progression of our work is the implementation of the Simultaneous Localization and Mapping (SLAM) problem on such platforms. More specifically, the compiled dataset and the implementation results acquired in this paper are very useful to be used in more advanced algorithms, i.e. Occupancy Grid based FastSLAM or DP-SLAM among others. In general, our contribution enhances the usage of small educational robots to demonstrate complex robotics concepts.

REFERENCES

A. Prorok, A. B. and Martinoli, A. (2012). Low-cost collaborative localization for large-scale multi-robot systems. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE.

A.D. Cucci, M. Migliavacca, A. B. and Matteucci, M. (2016). Development of mobile robots using off-the-shelf open-source hardware and software components for motion and pose tracking. In *Intelligent Autonomous Systems*. SPRINGER.

Ahmad, H. and Namerikawa, T. (2010). Robot localization and mapping problem with unknown noise characteristics. In *2010 IEEE International Conference on Control Applications*. IEEE.

C. Sprunk, J. Röwekämper, G. P. L. S. G. T. W. B. and Jalobeanu, M. (2016). An experimental protocol for benchmarking robotic indoor navigation. In *Experimental Robotics*. SPRINGER.

Colleens, T. and Colleens, J. (2007). Occupancy grid mapping: An empirical evaluation. In *Mediterranean Conference on Control & Automation*. IEEE.

D. Kortenkamp, R. B. and Murphy, R. (1998). Ai-based mobile robots: Case studies of successful robot systems. CAMBRIDGE.

Elfes, A. (1989). Occupancy grids: A probabilistic framework for robot perception and navigation. CMU.

F. Mondada, M. Bonani, X. R. J. P. C. C. A. K. S. M. J. Z. D. F. and Martinoli, A. (2009). The e-puck, a robot designed for education in engineering. In *conference on autonomous robot systems and competitions*. IPCB.

J. McLurkin, A. Lynch, J. A. S. R. T. B. A. C. K. F. and Bilstein, S. (2013). A low-cost multi-robot system for research, teaching, and outreach. In *Distributed Autonomous Robotic Systems*. SPRINGER.

M. Markom, A. Adom, E. T. S. S. A. R. and Shakaff, M. (2015). A mapping mobile robot using rp lidar scanner. In *International Symposium on Robotics and Intelligent Sensors*. IEEE.

P. Schmuck, S. S. and Zell, A. (2016). Hybrid metric-topological 3d occupancy grid maps for large-scale mapping. In *IFAC-PapersOnLine*. ELSEVIER.

S. Bazeille, E. B. and Filliat, D. (2015). A light visual mapping and navigation framework for low-cost robots. In *Journal of Intelligent Systems*. DE GRUYTER.

S. Thrun, W. B. and Fox, D. (2005). *Probabilistic Robotics*. MIT Press, USA, 2nd edition.

S. Wang, F.Colas, M. L. F. M. and Magnenat, S. (2016). Localization of inexpensive robots with low-bandwidth sensors. In *13th International Symposium on Distributed Autonomous Robotic Systems*. INRIA.

Shim, J. and Cho, Y. (2016). A mobile robot localization via indoor fixed remote surveillance cameras. In *Sensors*. MDPI.

Tribelhorn, B. and Dodds, Z. (2007). Evaluating the roomba: A low-cost, ubiquitous platform for robotics research and education. In *International Conference on Robotics and Automation*. IEEE.