Link Graph and Feature Chain based Robust Online SLAM for Fully Autonomous Mobile Robot Navigation System using Sonar Sensors

Amit Kumar Pandey
akpandey@research.iiit.ac.in

K. Madhava Krishna
mkrishna@iiit.ac.in

Robotics Research Center, International Institute of Information Technology (IIIT-H), Hyderabad, India

***Abstract*— Local localization of a fully autonomous mobile robot in a partial map is an important aspect from the view point of accurate map building and safe path planning. The problem of correcting the location of a robot in a partial map worsens when sonar sensors are used. When a mobile robot is exploring the environment autonomously, it is rare to get the consistent pair of features or readings from two different positions using sonar sensors. So the approaches, which rely on readings or features matching, are prone to fail without exhaustive mathematical calculations of sonar modeling and environment modeling. This paper introduces link graph based robust two step feature chain based localization for achieving online SLAM (Simultaneous Localization And Mapping) using sonar data only. Instead of relying completely on matching of feature to feature or point to point, our approach finds possible associations between features to localize. The link graph based approach removes many false associations enhancing the SLAM process. We also map features onto Occupancy Grid (OG) framework taking advantage of its dense representation of the world. Combining features onto OG overcomes many of its limitations such as the independence assumption between cells and provides for better modeling of the sonar providing more accurate maps.**

I. Introduction

In this paper we will focus on issue of continuous localization of a mobile robot, which is inevitable requirement for any fully autonomous mobile system to navigate safely in a partially known environment and to explore and build accurate and safe map of the environment. During the process of exploration, robot takes decisions to move to different locations to get a new 360 scan of the unknown portion of the environment to increment the map and plans a safe path within the known region to reach there. During the process of scanning and moving, the robot looses track of its 3D position (x,y,θ) due to unavoidable odometric error, which needs to be corrected before any further exploration action has to be taken. This Simultaneous Localization and Mapping problem (SLAM) is considered as chicken and egg problem in mobile robotics; because for building an accurate map and planning a safe and correct path robot need to know its exact position, which requires localization, but to localize the robot need to have the map. Figure 1 shows a map built by an autonomous robot, which is actually the map of a straight corridor, but due to the localization error the straight parallel walls are appearing as curved walls.

Offline SLAM processes tend to delay the localization process as they await the detection of features such as corners and edges or more data. But in an exploration process such delays can become unacceptable and requires for correction of robot pose at frequent intervals. In sonar based exploration due to very limited number of accurately detected features in one scan frequent pose correction is difficult. This is because edges and corners are highly unreliable (sometimes may not be detected also) and wall segments are the only reliable features. Our approach creates *feature chain* having a root feature and other link features using wall segments, which is used for localization. And the *link graph* approach removes the false associations between wall segments and corner and edge readings and gives a reliable set of associations suitable for SLAM. This it does by doing angle fit and distance fit analysis over several possible association patterns and chooses the best set of associations.

There are basically two broad approaches for SLAM. One is based on feature matching and another is based on raw data matching. But these approaches are basically dependent on the assumption that the inputs from the sensors are reliable and consistent. Generally a combination of sensors like vision and sonar [1] are used or a laser sensor [2, 3] is used or camera [4] is used. But very few authors [5, 6] have addressed the SLAM problem in indoor environment with only sonar sensors. The reason is that, by using sonar sensors, the situation rapidly deteriorates since sonar

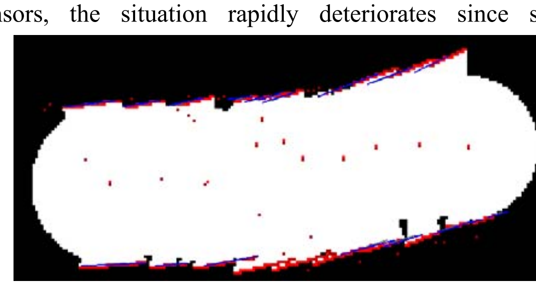


Fig. 1: Map of a straight corridor without correcting localization error during map building process. Straights walls are appearing as a curved.

We would like to thank Ms Sabita Pandey for her help and support in completing this work.

readings are susceptible to high degree of uncertainty especially due to angular and radial errors along with specular reflection problems. In fact one cannot get a pair of consistent readings for a particular world location from two different positions.

As shown in the fig. 2, the world point P_w has been hit by two sonar beams, from positions P_{t1} and P_{t2} of the robot at successive time instants. One may expect to obtain readings r_1 & r_2 as shown by solid lines. But practically one gets instead some unpredicted readings like r_{1extd} & r_{2extd}. The algorithms which assume that these inconsistencies in the reading are due to robot's odometric error will not work, since for a given environment it is very difficult to predict whether the inconsistency in the reading is due to sonar error or due to error in robot pose. This problem is not seen however with the laser sensor.

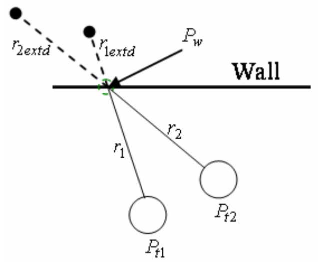


Fig. 2: From two position for sonar sensor instead of getting readings shown by solid line, we will get some unpredictable readings shown by extended dotted line, for a same world point Pw.

In exact feature matching based SLAM; once again there are problems with sonar sensors. The features which we can extract from sonar sensor are wall segments, corners and edges [7, 8]. Each of such features has a sparse representation and a robot scan from two proximal places does not guarantee matching between features.

If we assume for a corridor like simple environment, with only array of sonar sensors, wall segments will be the feature, which can be detect. And as shown in fig. 3a, in the best case it can detect two different wall segments with some overlapping parts and in worst case we will get non overlapping segments as shown in fig 3b. Hence not only matching of features between successive scans is difficult, the usual image registration algorithms that try to solve for the rotational and translational displacement between successive scans do not work since overlap between features is minimal and it is also difficult to predict how much the actual overlapping is. While these problems do not exist with laser range finders and vision based sensors, sonar is still an attractive proposition due to its low cost. Also Extended Kalman Filter (EKF) based online SLAM approaches [9,10] which use edges, corners and other point like features as landmarks to localize will not work for the corridor like environment (figure 1) or the environment where edges and corners are sparse and most of the features detected are line segments. Further the EKF approach tends to work well the less ambiguous the point landmarks are. For this reason, EKF SLAM requires significant engineering of feature detectors, sometimes using artificial beacons as features [9]. So, either the robot has to visit the previously seen edge or corners to localize or it has to use some manually put landmarks to localize or it has to delay localization until it detects some edges, corners or other landmarks. And the situation becomes worst with sonar sensors because assuring edges and corners like point features from an individual 360 scan by sonar are very difficult whereas getting line segment like feature is easy.

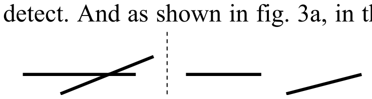


Fig. 3a Fig. 3b

Fig. 3a, 3b: By sonar sensor for two instants even from nearly same positions we can get two different wall segment features, which can be overlapping but need not to be the same (fig. 3a) or non-overlapping (fig. 3b).

In order to overcome the above problems and to achieve SLAM with sonar sensor only, we have proposed a novel ***Feature Chain*** based two-step localization. In fact with sonar, even edges, corners or circular objects can get classified as line segment features [8]. This obviously results into false association with an actual wall and hence deteriorates the localization accuracy. So to remove these false associations and to achieve robustness a ***Link Graph*** based global association analysis approach has been introduced. The resulting maps show that with the presented approach the robot is accurately able to complete the loop autonomously without any external path guiding.

II. BACKGROUND WORK

In [7], we have presented a novel approach to get a safe and more accurate map using sonar sensors only, based on feature detection and mapping onto occupancy grid framework. We will use that approach for updating the map. For the continuity, the basics of the approach will be briefly discussed here.

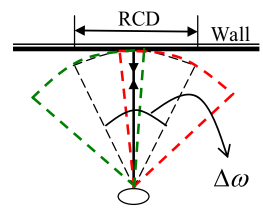


Fig. 4.: Concept of RCD. For all the three sonar beams shown as different colors, the perpendicular reading shown as middle solid arrowed line will return to the sonar sensor.

In general the reading from sonar is reliable only when at least one ray of the sonar beam hits normally to the surface. As shown in fig. 4, three different sonar beams are hitting the walls (shown as different colors) with different axis angles. But the distance returned by all the three beams will be the perpendicular distance shown by the middle solid line, because only that ray will return back to the sonar sensor. Hence we will get a set of approximately same reading (within a threshold) for all the sonar beams which have been fired within the angular range $\Delta\omega$, where $\Delta\omega$ is beam width of one sonar beam. This region is called the ***Region of Constant Depth (RCD),*** readings of which are the most reliable readings in the case of sonar sensors. There are special patterns of RCDs for walls and corners for a 360 degree scan of the environment by sonar sensor. In [7] we have utilized these properties to extract the features like walls and corners and able to build the accurate and safe map of the environment. In [7], a detailed description of a

Bayesian framework based approach for mapping is presented. Here we will assume that features like wall segments and corners have been extracted in each 360 degree scan by sonar sensors, and we will directly use this information for our present work.

III. METHODOLOGY

As explained in section I, it is rare to get consistent pair of features or readings by sonar from two different positions. So, instead of relying on feature matching, present method relies on feature association and creates chains of associated features.

A. Seed Chain Creation

Robot makes a first 360 degree scan of the environment from its current position. Instead of updating the map instantly with each reading, features are extracted from the raw range data [7]. Each wall segment feature will serve as a root of a new *feature chain* and will be assigned a probability P(root_of_first_scan) = 1. Initially all chains are having only one feature which is the root itself. Any feature which will be detected in future scans will be either associated with any chain or will become a new root for a new feature chain. If associated, then that feature will be a part of the feature chain and called as *link*. In that fashion a feature chain will grow. Also a chain can grow in both the directions.

B. Feature Association

During the process of exploration robot moves to a new location and takes a new 360 scan. Most of the time each such scan will result into some new wall segment features. These new features will be first tried to associate with any link or root of any existing feature chain. For associating overlapping wall segments, a mean distance *d1* is calculated, which is the average distances from both end points and middle point of a wall segment extracted in the current scan onto other existing wall segments which are link or root of feature chains. Also the angle *a1* between the two wall segments is calculated. If they are within some threshold, then both are associated. For non overlapping walls one more criterion is tested, the distance *d2* between the nearest end points of both the walls. If this is also within some threshold only then both features are being associated. In our case we have experimentally fixed *d1* as 20 cm, *d2* as 50 cm and *a1* as 20 degrees. Note that it is not necessary that features will be associated with the root of a chain; they can be associated with any other link feature of an existing chain. And it may also be the case that a feature will not be associated at all, which has been addressed later in this paper.

For better understanding we will use figure 5 which is a typical scenario from sonar scan. Red segments in figure 5 show new features extracted in a particular scan and green segments show the links of an existing feature chain to which the current feature has been associated based on above mentioned criteria. Other links of the chains have been shown in black. The corresponding root of each chain has been marked separately. Note that in some cases current feature has been associated with the root of the chain and in others with link of the chain.

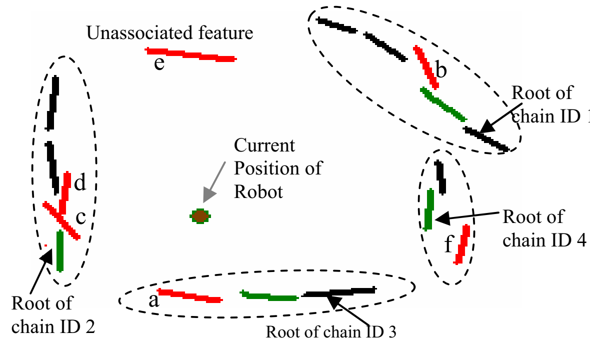


Fig. 5: 4 different feature chains (encircled), Features extracted in current scan (Red Segments), Links of existing chains to which current features has been associated (Green Segments).

Using the initial set of associated features obtained as above; we create a two-layered graph, which we are calling as *link graph*. Vertices at top layer of the link graph show the IDs of the feature chains to which at least one feature extracted in current scan has been associated. Vertices at the bottom layer show the current associated features ID. Edges show the association. Figure 6 shows the initial link graph of associations shown in figure 5.

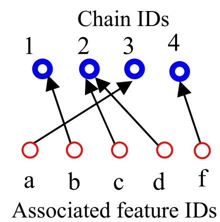


Fig. 6: Initial link graph of associations of fig 5.

C. Achieving Robustness

The initial link graph, constructed as above may contain some false associations, which if used for correcting the pose (x, y,θ) may result in incorrect localization. In fact the association till now is loosely coupled in the sense they are based on individual feature association. So to detect false association we have to analyze the associations from a global perspective. For this, a two-step individual feature fit based analysis, with root of corresponding chain, is done.

First angle fit analysis is done. For each associated feature of initial link graph, robot is virtually oriented in such a way that the angle difference between that particular feature and its associated root becomes zero. And then with this new orientation of robot, the orientation and position of all other features are being calculated. Then for this particular fit, the angle difference of each associated feature present in initial link graph with its root is calculated. Note that this angle difference is calculated with the root of the chain whereas for association, the angle difference was checked with each link of the different chains. Then another link graph is created with those associations whose angle differences with root are within a window of + *th* to – *th*. In our case we have chosen *th* as (Avg_ang + ε), where Avg_ang is average angle difference between all the initially

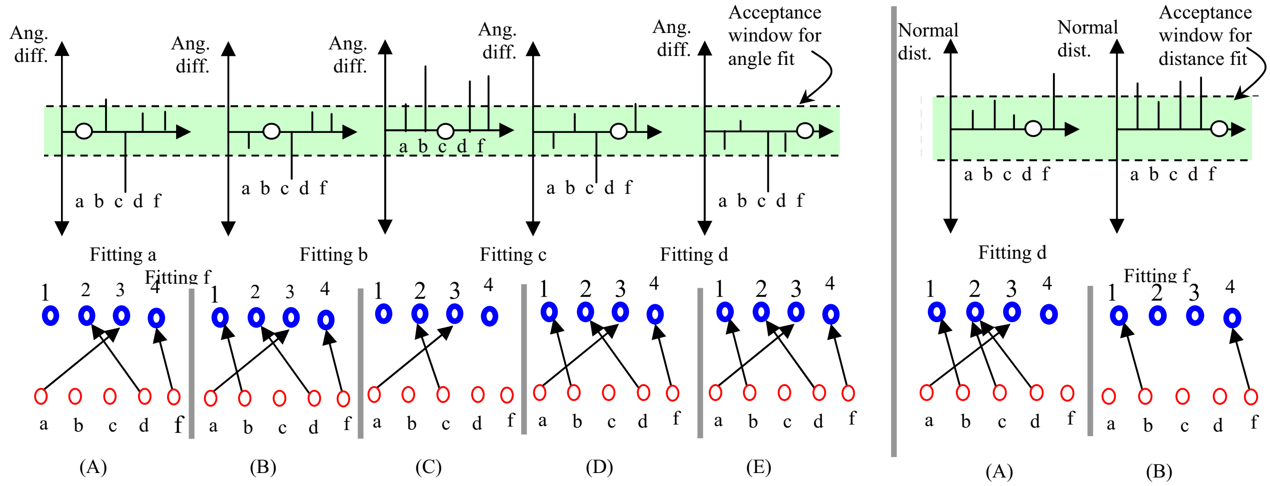


Figure 7

Figure 8

Figure 7 & 8 : Top row of Fig. 7: Angle difference between feature and the root after exactly fitting one feature at a time by making angle difference 0. Bottom row of fig 7: Corresponding link graphs with the links which are within the green window of the accepted angle difference. Top row of Fig. 8: Normal distance from middle of feature and the root after exactly fitting one feature at a time by making normal distance 0. Bottom row of fig. 8 : Corresponding link graphs with the links which are within the green window of the accepted normal distance.

associated features and the root of the associated chains; ε is error limit calculated experimentally, in our case ε is 2 degree. Figure 7 shows link graphs by fitting angle of each associated feature of figure 5 one at a time. The link graph having maximum number of links will be the winner link graph from angle fit analysis. In our example the link graphs 7(B), 7(D) and 7(E) are having maximum number of links, so they will be the winner from angle fit analysis and also will be a candidate for final winner link graph.

The rotational error between two wall RCDs of the same wall are generally due to the rotational error of the robot between the two locations. However the same can not be said of corner and edge RCDs [8]. One may think it is easy to discard such corner and edge RCDs from the association process but many a time such corner and edge RCDs are wrongly classified as walls. This in turn leads to wrongly classified walls getting associated with previous walls. But the errors of such false associations will not be globally consistent with the errors of other true associations. If we will take these types of false associations for correcting location and orientation we end up with incorrect localization. There exist several other sources of false associations whose errors are not globally consistent with errors due to true associations. The angle fit analysis explained above takes advantage of this inconsistency to remove these false associations. The other sources of such errors are pillar coming out of the wall, circular objects close to wall and so on.

The winner link graphs from angle fit obtained as above may contain some associations, which are not agreeing globally from the point of view of perpendicular distance between feature and the corresponding root. As shown in the figure 5, the current feature f has been associated with feature chain 4 and it is also present in the winner link graph of angle fit. But it may be the case that the associated chain is of an open door, which is parallel to a wall and feature f is a new feature of that wall but falsely associated with feature chain of door. These types of false association can occur in other cases where there is some step like structure in the environment. To remove those associations we will generate a set of a different kind of link graph based on the distance fit as explained below.

Firstly, all the associated features will be made parallel to their root features by virtually rotating them individually with respect to robot. Then the average normal distance from middle point of each feature to the line formed by the associated roots will be found. Let us say it as actual average normal dist, Act_Avg_Norl_Dist. Now each associated feature will be made parallel to the root feature one at a time and the normal distance from middle point of the feature to root line segment will be found. This normal distance will be broken into Δx and Δy components. All features will be shifted by this (Δx, Δy). It is basically fitting one feature exactly based on the normal distance. Then other feature will be again made parallel to their corresponding roots with respect to robot from their new shifted position and again the individual normal distance from the middle point of each feature to the corresponding root line will be found. All those association for which this new normal distance is within some threshold –th2 to +th2 will form an edge in the link graph corresponding to that particular fit. We have taken th2 as (Act_Avg_Norl_Dist + ε2), ε2 is calculated practically, in our case it is 5 cm. Figure 8 shows link graph made by distance fit. Figure 8(A) shows the link graph when feature d has been exactly fit, we will get similar link graphs for fitting features a, b and c. Figure 8(B) shows the link graph by fitting feature f. So, obviously the winner link graph for distance fit is 8(A). Now to decide the final winner

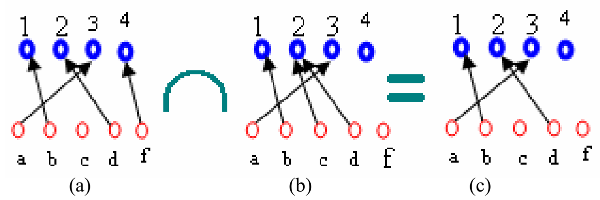
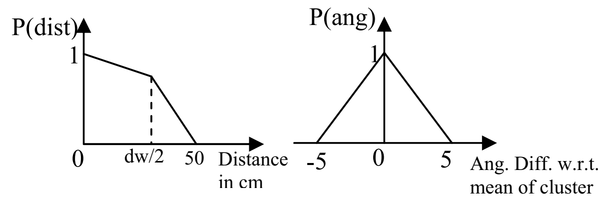


Figure 9(a)(b)(c) : 9(a) : one of the winner graph by angle fit, 9(b) : one of the winner graph by distance fit, 9(c) : Final winner link graph by taking common edges.



Figures 10(a)-10(b): (a) graph for calculating probability wrt. Nearest end point distance; (b) wrt angle difference from mean of the cluster.

link graph, various pairs of winner graph of angle fit and distance fit will be made by taking one from each set at a time. For each pair, another link graph is created by taking the common edges from both the graphs. The final winner will be the link graph from the pair, which will be having maximum number of common edges. And the common edges will be the final edges of the winner link graph. In our example we will get various pairs having same number of common edges, the pair {7(B), 8(A)} is one of them. So we will choose the common part of this pair as the final winner link graph as shown in figure 9.

The features for which an edge is present in the final link graph will be inserted as new links in the corresponding feature chains. And for other features, a new chain will be created by making them as the root of the chain.

This approach is able to remove false associations, which are features c and f of figure 5 in our example. In practice we can get this type of features, which can easily be associated falsely with existing features due to various reasons mentioned earlier.

D. Calculating Probability of Association

After having a reliable set of association which will be used for correcting pose (x, y,θ) of the robot, we will assign some measure of certainty of association to a particular feature with the corresponding associated chain. We will assign higher weight to those associations, which are more agreeing in a global sense. Because it may be the case that angle difference of a particular feature with its root is less but for 3 other associated features it is falling in a higher range within some limits. So those 3 associations are more agreeing about the amount of orientation error even when their angle differences are more. Hence we find clusters of features based on the angle difference with corresponding roots. The cluster having maximum number of features will be given highest weight. A probability is assigned to each cluster by equation 1.

$$P(cluster) = \frac{Number\_of\_element\_in\_the\_cluster}{Total\_number\_of\_elements} \quad \textbf{(1)}$$

A second type of probability is calculated based on the distance of the associated feature to the nearest end of the associated link (not necessarily the root) in the chain. This probability P(dist) is calculated by the graph shown in figure 10(a). Here dw is the average width of the commonly existing doors. If the difference is less than half of the door width then probability will be higher to ascertain more certainly that the current feature as the part of the same wall. A third type of probability is calculated based on angle difference from the mean of the cluster. Note that cluster itself has been found by angle difference of a feature from the corresponding root. Graph in figure 10(b) shows how this probability P(ang) is calculated.

And the final probability of association is computed as a union of these probabilities as in equation 2:

$$P(association) = P(Cluster) \cup P(dist) \cup P(ang) \quad \textbf{(2)}$$

E. Creating New Chains from Unassociated features

It is common and also encouraging to have some unassociated features in a particular scan. It is an indication of some new unseen portion or structure of the environment. Each unassociated feature will create a new chain by making itself as the root of that chain. But this time the probability of that root will not be 1 as in the case of first scan. P(root) is basically the certainty of location of that root feature itself and since robot has moved from its initial position so some error must have been introduced. There are two sub-cases: (a) some other features of the current scan have been associated with some existing feature chain. (b) None of the features of the current scan could be associated with any of the existing chains. For the first case the probability to root will be assigned by equation (3) :

$$P(new\_root) = \frac{\sum_{i=1}^{n} P(root_{i^{th}}\_associated\_chain)}{n} \quad \textbf{(3)}$$

Where n is the total number of features of the current scan which has been associated to any of the existing chain. For the case (b), the P(new_root) will be half of the probability which has been calculated for the most recent case (a) by equation (3). And it will be assigned a tag of being a *dangling chain*. This will be the region where uncertainty in localization will be the most. Our exploration strategy should be such that it should encourage getting some unassociated features but of type (a), discourage getting dangling chains and also it should try to associate dangling chains with some normal chains by finding the missing link if possible. A special case has been discussed at subsection G.

F. *Local Localization*

After having a reliable feature association with association probabilities, a two step localization process has been introduced to correct the position (x, y) and orientation θ of the robot. First the orientation of the robot is corrected followed by (x, y) correction.

1) *Correcting Orientation*

For correcting the angle assumption of the robot we take a weighted sum of angle difference from the roots (not with the associated links), and find the robot's orientation, which will minimize equation (4).

$$\theta = \arg\_\min \left(\frac{\sum_{i=1}^{n} ((w_1 + w_2 + w_3) * ADR(asso_i))}{n} \right) \qquad \textbf{(4)}$$

Where arg_min returns the value of argument θ which will minimize the quantity inside the bracket, n is the number of associated features' instances. $ADR(asso_i)$ is the angle difference of the feature i with the root of the associated feature chain. It is important to note that the association has been done with link of the chain but the correction will be done with the root of the chain, because the roots are more reliable then any other link of the same chain. Weights w1 is simply the probability of root, P(root), of the associated chain, calculated by equation (3); w2 is half of the probability of association P(association) calculated by equation (2). If a feature has been detected far from the robot's current position the accuracy of orientation and position of feature compared to the nearby-detected features will be less. So another weight w3 takes care of this fact. Figure 11 shows the graph to calculate w3 with respect to distance from the robot. 20 and 200 are the minimum and maximum range of getting reliable reading by our sonar sensor.

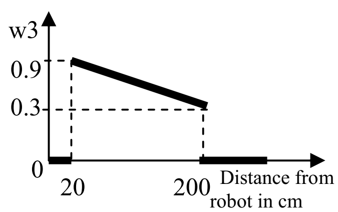


Figure 11: Graph for calculating weight w3 based on distance to feature

2) *Correcting Position (x,y)*

For correcting the (x,y) co-ordinate of the robot, we have to align each individual associated feature completely with its associated link in the chain. Then the normal distance from the middle point of the current feature onto the associated link feature will be calculated as shown in figure 12. For each valid association this distance will be decomposed into (Δx, Δy). This decomposition is required, because if we directly use the normal distance for searching the correct position and if all the associated features are parallel to either x or y axis then we will end up with a line segment as the possible correct position instead of getting a point. But by using maximum values of Δx and Δy from this decomposition as a window for our search in x and y direction, if features are parallel to any of the axis, as may be the case for corridor, one of the components will nearly equal to 0 for all the associated features. Hence even if robot will not be able to correct the position along one axis, it will also not increase the error along that axis by avoiding the ambiguity.

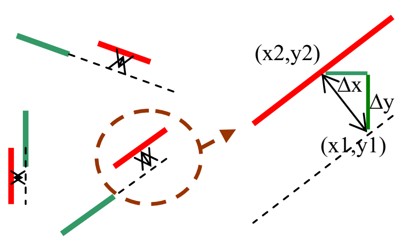


Figure 12 : Finding normal distance and decomposing the shift into (x,y) component.

Now to get the correct (x,y), we search in a window of –maximum of Δx and –maximum of Δy to +maximum of Δx and +maximum of Δy, from the current assumed position of the robot. For each possible position of the robot in that window, new position of the middle points (which has been found earlier by making all the associated features virtually parallel to root with respect to robot) is calculated. The shift in x and y, which will minimize the weighted sum of the normal distances given by equation (5), will be the final shift to be added in the current position to get the correct position of robot.

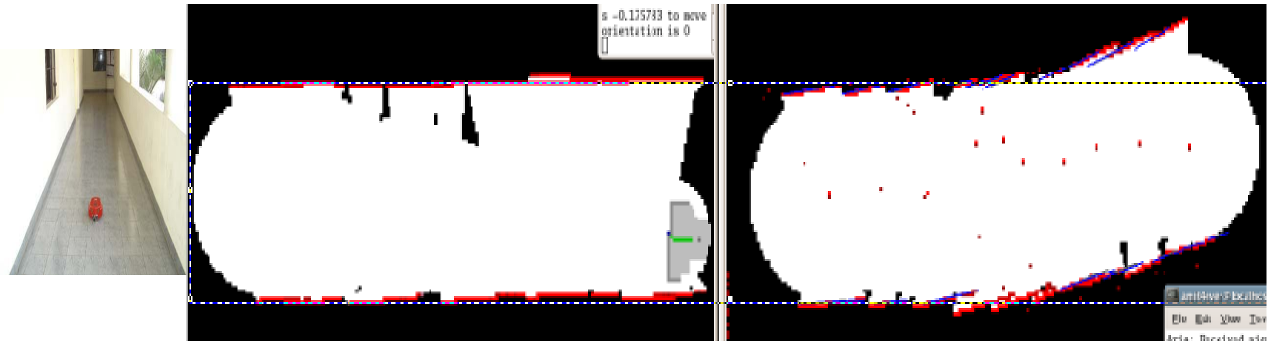
$$(\nabla x, \nabla y) = \arg\_\min \left(\frac{\sum_{i=1}^{n} ((w_1 + w_3) * NDR(asso_i))}{n} \right) \qquad \textbf{(5)}$$

Where w1 and w3 are the weights calculated in equation (4) and $NDR(asso_i)$ is the normal distance from the middle of the feature (made virtually parallel) to the associated root.

After getting corrected (x,y,θ), all the readings and the features are processed to make them consistent with the corrected location and orientation of the robot. And then map is incremented using our map building approach presented in [7].

G. *Handling Duel Association*

A special case can arise when a same current feature has been initially associated with two different feature chains. Then each association will be treated as separate instance and all the analysis for removing false associations will be done as usual. If even after false association removal, that feature is still being associated with both the chains; it indicates either the loop completion or the missing link of a feature due to which same wall is being treated as two different features so far. Also one chain may be the dangling chain mentioned in the section III E. In all the cases the two chains will be merged and the new root will be the root which is having higher probability. Also if one chain is dangling chain then that tag will be removed from that chain, because it has found its base chain. So the exploration strategy will not try to find its base chain any more to reduce the region of uncertainty.



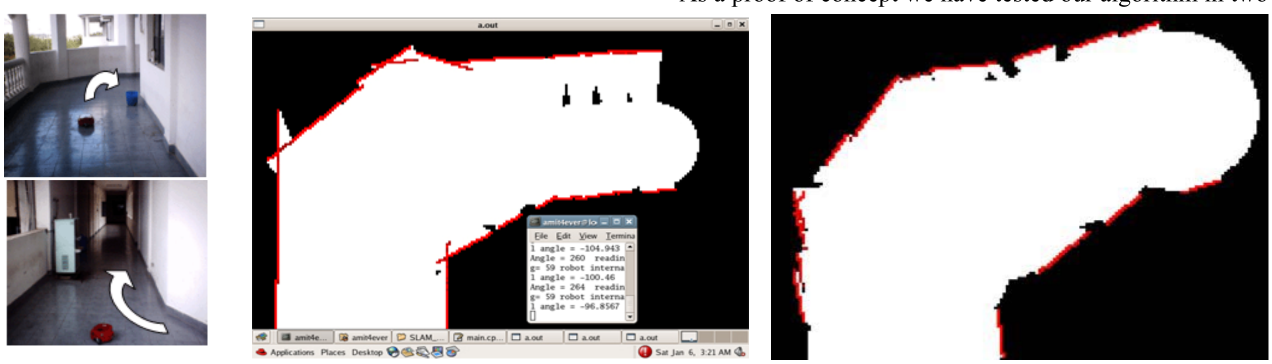
Figures 13(left) 14(a)(middle), 14(b)(right): a corridor environment; figures 14(a), 14(b): Comparison of both the maps obtained with SLAM and without SLAM, with an overlaid dotted rectangle resembling actual environment.

IV. Experimental Results and Analysis

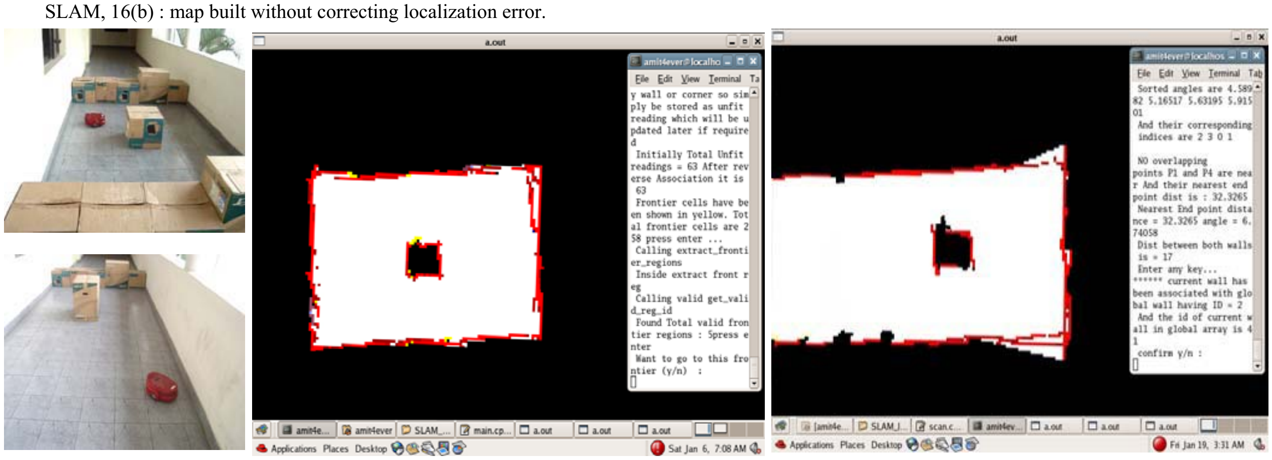
We have tested our algorithm in different real environments. We are using Amigobot equipped with 8 sonar sensors fixed at certain angular intervals. For taking a 360 scan we are rotating the robot with 4 degree interval and storing the readings from only one sensor. Remaining sonar sensors are used during path planning. Figure 13 shows the corridor environment. Figure 14(a) shows the map build by using present approach of SLAM in a fully autonomous fashion. Figure 14(b) shows the map built without correcting the localization error. For comparing both the maps, a dotted rectangle has been overlaid on them. The shift of the walls from the sides of the rectangle is much in 14(b), which is without SLAM and the shift in 14(a) is negligible showing the accuracy of map closer to the ground truth.

Fig 15(a) and fig. 15(b) show two portions of a corridor as it bends around a corner, which is not at right angles. Figure 16(a) shows the map built using the present method of SLAM. Figure 16(b) shows the map obtained by robot of the same environment but without robot's position being corrected. The enhancement is visible. In fig. 16(b) the robot's position and orientation becomes more inaccurate after some scans resulting in a map that is shifted and tilted. Using the present approach however the map in figure 16(a) is closer to ground truth results.

As a proof of concept we have tested our algorithm in two



Figures 15(a)(b)(left column), Figure 16(a)(middle), 16(b)(right): 15(a),(b) :Two portion of a corridor; 16(a): map built by using present approach of SLAM, 16(b) : map built without correcting localization error.



Figures 17(a),(b)(left column): Two environments which requires the robot to complete the loop, 17(b) is having one end open; Figures 18(a)(middle) : Map built of fig 17(a); figure 18(b)(right) : map of 17(b), in both case loop has been completed successfully.

Figure 19 (left) & 20 (right) : fig. 19 :Another environment having rooms and corridor; fig. 20: map built by using present approach of SLAM. Without correcting the localization error robot was even not able to come out of the room autonomously.

different environments, which are having loops. Figure 17(a) and 17(b) shows two such environmental setups. 17(b) is having one side open, so that after completing the loop robot should be able to explore and build the map of the open portion of the corridor. Figure 18(a) shows map of 17(a) build by using present approach and figure 18(b) shows map of 17(b) using present approach. It is evident from these maps that in both the cases the robot was able to successfully complete the loop by exploring the environment autonomously without any path guiding, and the maps are also near to ground truth. Without correcting localization error robot was even not able to explore the loop environment completely without external path guidance. Figure 19 shows another environment having corridor and rooms. Figure 20 shows the map built by present approach of SLAM. Robot had started from the first room and then autonomously came out of the room by exploring and building the map with safe path planning. Then by building the map of corridor it entered into another room to explore and build map. During the entire process due to various small structures in the environment robot was getting lots of false associations. Our approach was able to detect and remove those false associations to localize properly. Also when we tested without correcting the localization error, robot was even not able to come out from the first room autonomously to explore the corridor and other rooms.

V. Conclusion

We have presented a robust *feature chain* based approach for online SLAM by using data from sonar sensors alone. The readings from sonar sensors are unpredictable beyond some minimum angle of incidence of the sonar beam onto the obstacle and hence getting consistent pair of features or readings from two different positions is rare. The proposed feature chain based approach is able to handle various issues with sonar data and successfully localizes and builds a more accurate map. One of the benefits of maintaining the feature chains having roots and other links is that the robot does not need to re-visit the previously detected features each time it has to localize. It can use the information of earlier features to localize from the current position itself even if those features have not been detected in the present scan. Furthermore for associating a current feature the closest link of the chain is used but for correcting the localization error we traverse back to the root of the chain and then the position and orientation of the root is used for minimizing the cost functions, thus enhancing the accuracy. Also the developed two-step global consistency association analysis based on link graphs of angle fit and distance fit is able to remove false associations; these false associations are practically unavoidable when using sonar data. The present approach enables the robot to explore the environment autonomously without any external aid. The robot is able to build maps involving loops, demonstrating its efficacy.

References

[1] Jinwoo Choi, Sunghwan Ahn, Minyong Choi, Wan Kyun Chung, “Metric SLAM in home hnvironment with visual objects and sonar features,” IROS,2006.

[2] Jens-Steffen Gutmann, Christian Schlegel, “AMOS: Comparision of scan matching approaches for self localization in indoor environment,”1996.

[3] Cyrill Stachniss, Dirk Hahnel, “Exploration with active loop closing for FastSLAM,” IROS, 2004.

[4] Li Maohai, Hong Bingrong, “Novel mobile robot simultaneous localization and mapping using rao-blackwellised particle filter,” International Journal of Advance Robotics System, 2006.

[5] Juan D. Tardos, Jose Neira, Paul M. Newman, John J Leonard, “Robust mapping and localization in indoor environment using sonar data,” International Journal of Robotics Research, 2002.

[6] Olle Wijk, Henrik I.Christensen, “Triangulation based fusion of sonar data with application in robot pose tracking,” IEEE Transactions on Robotics and Automation,2000.

[7] Amit Kumar Pandey, K. Madhava Krishna, Mainak Nath “Feature based occupancy grid maps for sonar based safe mapping,” *IJCAI* – 2007, pp. 2172-2177 [online]. Available: http://www.ijcai.org/papers07/contents.php.

[8] Simon Lacroix, Gregory Dudek, “On the identification of sonar features,” IROS, 1997.

[9] Sebastian Thrun, Wolfram Burgard, Dieter Fox, “Probabilistic Robotics,” The MIT Press, Cambridge, 2005, pp 312-316, 324-332.

[10] Bertrand Douillard, “Design and implementation of an SLAM algorithm on an Amigobot,” M.S. Thesis, Dept. of Mechanical and Aerospace engineering, State university of New York at Buffelo, 2005.