

Table-Based Matching Algorithm for Localization and Orientation Estimation of Multi-Robot System

Ola A. Hasan
Electrical Engineering Depart.
Basrah University
Basrah, Iraq
Loliastar91@gmail.com

Abdumuttalib T. Rashid
Electrical Engineering Depart.
Basrah University
Basrah, Iraq
abdturky@gmail.com

Ramzy S. Ali
Electrical Engineering Depart.
Basrah University
Basrah, Iraq
rsawaily@gmail.com

Abstract In this paper, a new algorithm called table-based matching for multi-robot (node) that used for localization and orientation are suggested. The environment is provided with two distance sensors fixed on two beacons at the bottom corners of the frame. These beacons have the ability to scan the environment and estimate the location and orientation of the visible nodes and save the result in matrices which are used later to construct a visible node table. This table is used for matching with visible-robot table which is constructed from the result of each robot scanning to its neighbors with a distance sensor that rotates at 360° ; at this point, the location and identity of all visible nodes are known. The localization and orientation of invisible robots rely on the matching of other tables obtained from the information of visible robots. Several simulations implementation are experienced on a different number of nodes to submit the performance of this introduced algorithm.

Index Terms— Localization, Beacons, multi-robot.

I. INTRODUCTION

Due to a high interest to the multi-robot system, the lack of localization and path planning have increased, especially in indoor systems, such as libraries, supermarkets, underground parking, etc [1-3]. The position determination (Localization) is a vital issue in multi-node or multi-robot systems. The gathered information from sensor nodes should include the locations of them to provide a more accurate view of the observed sensor environment. Also, in the applications that are used for objects tracking, the location determination has a vital role [4].

By location information, the nodes in the multi-node system can be classified to beacons (anchors), unknown location nodes and localized nodes [5]. The locations of anchors can be obtained by using GPS or by placing them at known coordinates points; the other nodes can be localized using localization protocols [6, 7].

Localization protocols are divided into two main categories: range-based and range-free [8]. With the range-based localization protocols, the exact location information of nodes is required and by using the ranging techniques, the remaining nodes

can compute their distance or angle to the known location beacons. Depending on this information and by using the localization techniques, the unknown location nodes can estimate their locations [9]. The range-free localization protocols depend on the way that the anchors are distributed, and they also depend on the connectivity information between nodes. The range-free localization protocols have lower accuracy as compared with the range-based localization protocols, and they have no need for any additional hardware techniques because they do not use the ranging techniques [11].

The ranging techniques classified according to the way they are calculating distance or angle into four main classes: Received Signal Strength (RSS), Time Of Arrival (TOA), Time Of Difference Arrival (TDOA) and Angle Of Arrival (AOA) [12]. In RSS ranging technique, the distance estimation is based on the strength of received signal and received power [13]. In TOA, the distance is estimated by measuring the time to transmit and receive the signal between the two nodes. So, these measurements are based on the speed of the signal and the propagation delay [14]. In TDOA, the distance can be estimated by using the difference in the receiving time between

two separate signals [15]. Finally, AOA uses the direction of the received signal (the angle) instead of distance [16]. The RSS technique has accepted as the most practical and the TOA as the most accurate.

After the measurement of distance or angle, the range-based protocols apply localization techniques to estimate the location for a node. The localization techniques include trilateration, triangulation, and multilateration [17]. Trilateration uses, at least, three neighbors. Each of them has a known location and uses the measured distance from each of them to the unknown location node to be the radius of the circle that has to draw around each of the three known location nodes. The intersection of these circles is the location of the unknown location node [18]. On the other hand, triangulation uses AOA ranging technique, and it needs the location information of two nodes to draw a triangle, and it uses the trigonometric relations to localize the unknown node [19]. Multilateration is just like trilateration, the only difference is that it uses the distance measurements of multiple neighbors, and it is used in the case of noisy distance measurements to provide accurate position estimation [20].

According to the computation model, the multi-node systems can be categorized to distributed and centralized architectures. In distributed architecture, each node can calculate its own position by communicating with its neighbors, the drawback of this architecture it needs additional hardware at each node for position computation [21]. The centralized architecture has a base station (BS) which are collected all the received data, and it does most of the computations. This method has the main drawback that is any BS failure causes the system breaking down [22].

This paper aims to deal with the localization and orientation problems of the multi-node system. The system will consist of two beacons placed at the two bottom corners of a fixed frame. Each beacon has a distance sensor that scans the environment and returns the result in its dedicated matrix. Based on this information a table called the visible -node table is constructed which contains the angles between each successive neighbors for all the seen robots in the environment, until this moment the locations of

the seen robots are known but the identities of them are not. The identities will be known by matching the angles in the first table with the angles of another table called the visible -robot table; this table is constructed with the help of reading obtained from a distance sensor placed on each robot. The details of this algorithm are discussed in section 2. Section 3 describes the results of this algorithm, and finally section 4 describes the conclusions.

II. THE TABLE-BASED MATCHING ALGORITHM

This algorithm is used to determine the location and orientation of robots in some environment by the matching of two tables: the first table contains the angles between two next neighbors for all visible robots. This is achieved by using the information obtained from scanning the environment using two beacons. In this part of the algorithm, we will refer to each visible robot by the node, this is because the beacons are capable of estimating the coordinates of the visible robots, but they cannot recognize the identity of them. The second table contains the angles between two next neighbors for all robots depending on the reading provided by a distance sensor installed on each robot. So, by matching these two tables, the location, and identity of each robot are known. Fig.1 shows the block diagram for this algorithm.

A. Visible nodes localization using two beacons

This section includes two steps to determine the location of nodes on two beacons which are located in the bottom corners of the environment. Each beacon equipped with a distance sensor to scans the environment and for computing the positions of all visible nodes.

Step 1: the first beacon (beacon 1) examines the environment. In fact, some nodes are visible to beacon 1 and others are not, as shown in Fig.2. The red nodes are visible to beacon 1 and the gray ones are not. The nodes which are represented by the visible nodes can be manipulated according to the sub-steps mentioned below:

1. The angle α_{b1}^i and distance d_{b1}^i of each visible node i to beacon1 are estimated by the scanning.

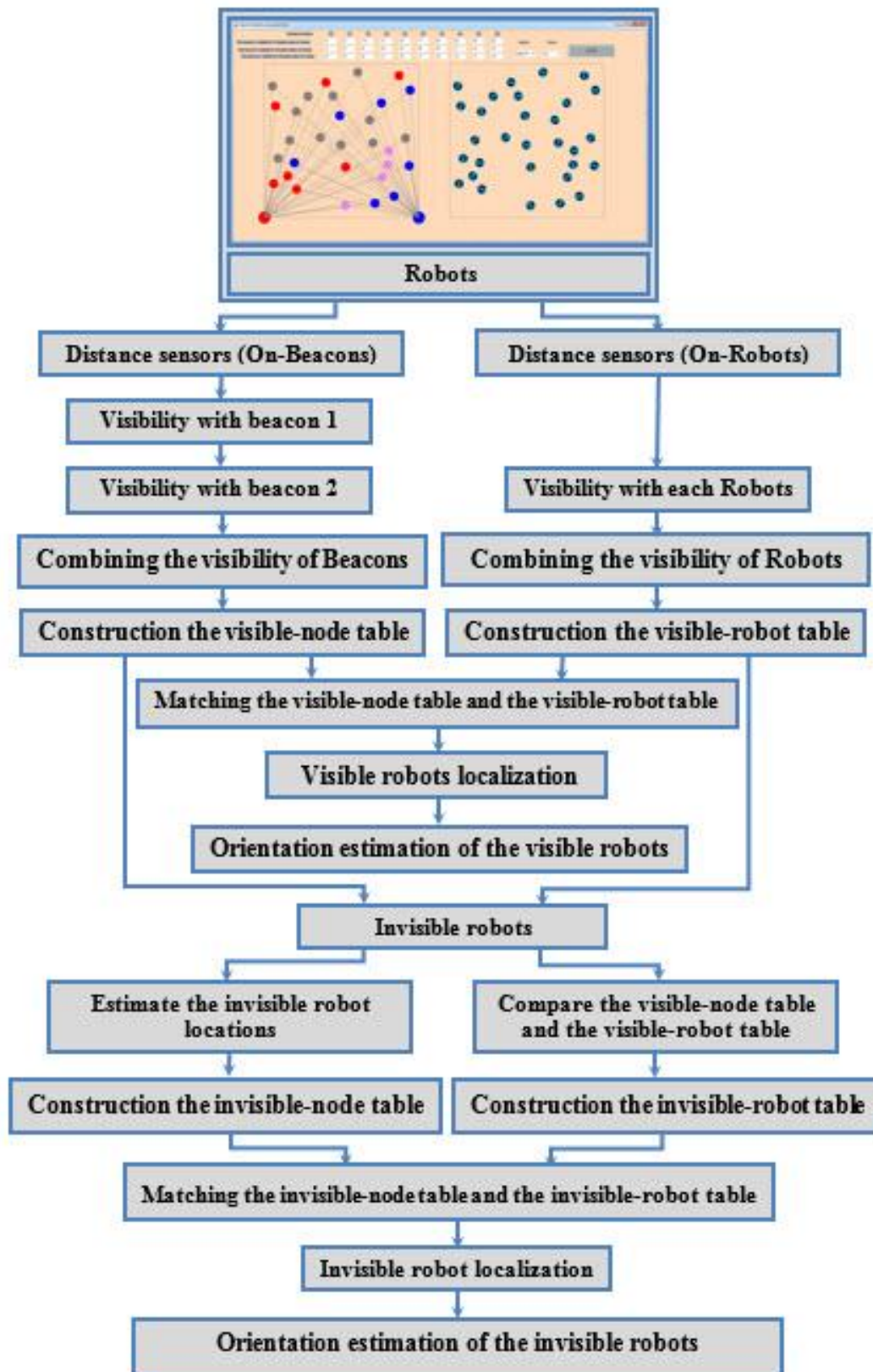


Fig.1. Block diagram of the table-based matching algorithm.

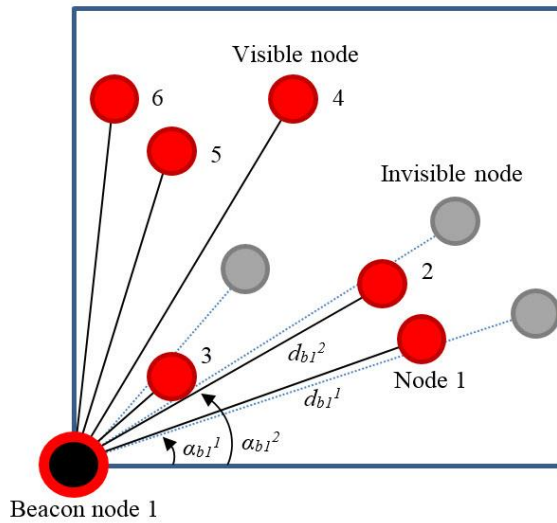


Fig.2. Scanning the environment with beacon 1.

2. The position of visible node i to beacon1 (x_{b1}^i, y_{b1}^i) is computed according to the following equations:

$$x_{b1}^i = x_{sb1} + d_{b1}^i \cos \alpha_{b1}^i \quad (1)$$

$$y_{b1}^i = y_{sb1} + d_{b1}^i \sin \alpha_{b1}^i$$

Where (x_{sb1}, y_{sb1}) is the position of beacon 1.

3. All the gained information from beacon 1 are stored in the matrix A_1

$$A_1 = \begin{bmatrix} d_{b1}^1 & \alpha_{b1}^1 & x_{b1}^1 & y_{b1}^1 \\ d_{b1}^2 & \alpha_{b1}^2 & x_{b1}^2 & y_{b1}^2 \\ \vdots & \vdots & \vdots & \vdots \\ d_{b1}^r & \alpha_{b1}^r & x_{b1}^r & y_{b1}^r \end{bmatrix} \quad (2)$$

Where r is the number of visible nodes to beacon 1.

Step 2: beacon 2 scans the environment to estimate the positions of nodes that are found within its visibility field. As shown in Fig.3, some nodes are visible to beacon 2 (blue nodes), and others are not (gray nodes). The nodes are manipulated basis on some sub-steps that are similar to the sub-step mentioned in step 1 as follow:

1. Scan the environment with beacon 2 to estimate the angle α_{b2}^i and distance d_{b2}^i of each visible node.

2. Compute the position of each visible node i to beacon 2 (x_{b2}^i, y_{b2}^i) according to the following equations:

$$x_{b2}^i = x_{sb2} + d_{b2}^i \cos \alpha_{b2}^i \quad (3)$$

$$y_{b2}^i = y_{sb2} + d_{b2}^i \sin \alpha_{b2}^i$$

Where (x_{sb2}, y_{sb2}) is the position of beacon 2.

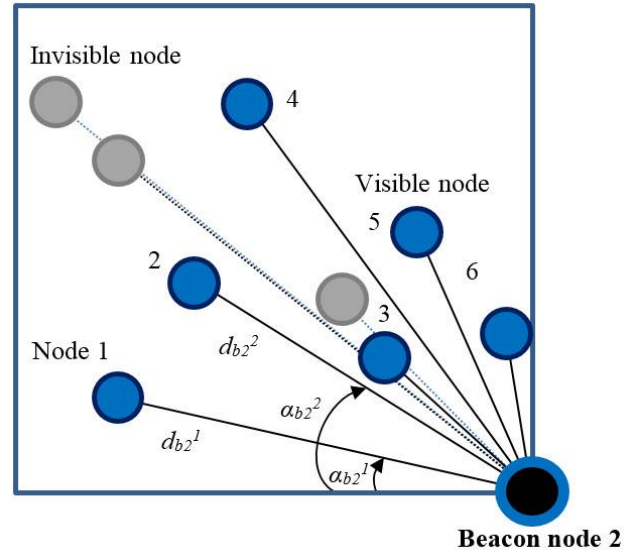


Fig.3. Scanning the environment with beacon 2.

1. Store all the information obtained from beacon 2 in the matrix A_2

$$A_2 = \begin{bmatrix} d_{b2}^1 & \alpha_{b2}^1 & x_{b2}^1 & y_{b2}^1 \\ d_{b2}^2 & \alpha_{b2}^2 & x_{b2}^2 & y_{b2}^2 \\ \vdots & \vdots & \vdots & \vdots \\ d_{b2}^v & \alpha_{b2}^v & x_{b2}^v & y_{b2}^v \end{bmatrix} \quad (4)$$

Where v is the number of visible nodes to beacon 2.

B. Combining the visibility of the two beacons

This section describes the matching between the scanning of beacon 1 and the scanning of beacon 2 to construct the complete visibility of both beacons. The coordinates of nodes that are visible by both beacons or by one of them are stored in a new matrix called A_3 and give them new names such that the pair (x_1, y_1) which represents the node that is stored in the first row of matrix A_3 . As shown in Fig. 4, the node that is seen by both beacons will be colored in purple, on the other hand, the node that is only seen by beacon 1 will be red, and the node that is

only seen by beacon 2 will be blue. The matching is done according to the following steps:

Step 1: starting from the first row in matrix A_1 , we take the coordinates (x_{b1}^1, y_{b1}^1) and scan matrix A_2 for coordinates that matching or close to them.

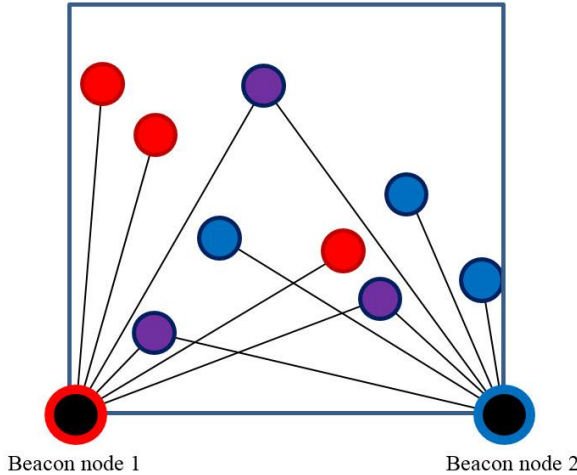


Fig.4. Combining the scanning of the two beacons.

Step 2: If matching occurs, the node will be colored in purple; this means that both beacons see the node and its coordinates will be stored in the first row of matrix A_3 . Else if the coordinates are close to each other, we choose the nearer one (less distance) to the beacons. As an example, from Fig. 5 we find that the coordinates (x_{b1}^1, y_{b1}^1) and (x_{b2}^3, y_{b2}^3) represent the coordinates of the same node but on beacon 1 and beacon 2. We choose the coordinates (x_{b2}^3, y_{b2}^3) because the distance to beacon 2 (d_{b2}^3) is nearer than the distance to beacon 1 (d_{b1}^1).

It is worth to mention here, that less distance to the beacon means less error in location estimation where the error is represented by the distance between the node center and the closest one of the beacon's visibility paths. In Fig. 6, we notice that error e_1 for the measured distance 1 is less than error e_2 for the measured distance 2. Where e_1 is the distance between the node's position at distance 1 and the closest visibility path of the beacon and e_2 is the distance between the node's position at distance 2 and the nearest beacon's visibility path.

Step 3: Replace the coordinates, distance and angle of the matched node in matrices A_1 , A_2 with zero to be excluded later during the rest of matching.

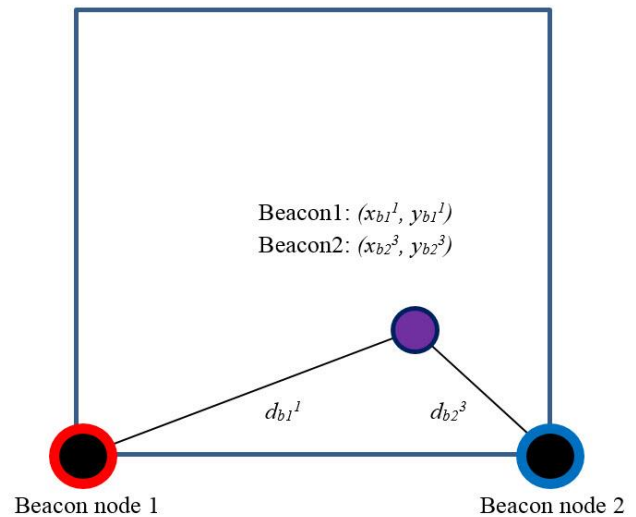


Fig. 5. The distances from both beacons to the same node.

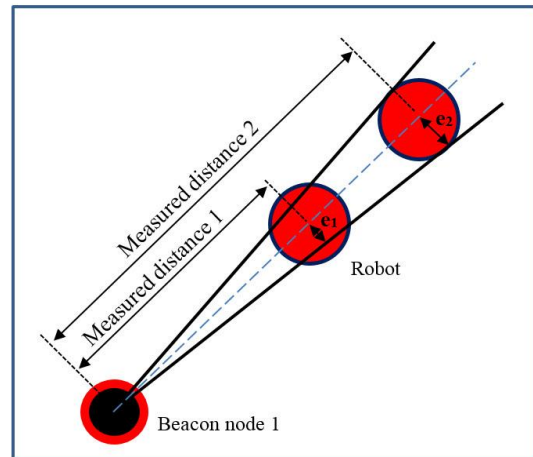


Fig. 6. Illustration of location estimation error

Step 4: Repeat steps 2 and 3 until finishing all the purple nodes.

Step 5: If the coordinates in matrix A_1 are not matched or even close to any coordinates in matrix A_2 (the row which has an angle of non-zero value), the node which belongs to those coordinates will be colored in red to indicate that this node is being seen by beacon 1 only. The same coordinates will be used in matrix A_3 after changing their names to be suitable with the names used by matrix A_3 .

Step 6: After finishing all the rows in matrix A_1 , we check the angles in matrix A_2 . If any angle has a value instead of being zero, the node will be colored with blue which means that the node is being seen by beacon 2 only, and we change the names of coordinates for this node to be suitable to store in matrix A_3 .

$$A_3 = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_q & y_q \end{bmatrix} \quad (5)$$

Where q is the number of visible nodes to beacon 1 or beacon 2.

C. Construction of the visible node table

At the finishing of section B, all the visible nodes are included in matrix A_3 . But we do not know which robot belongs to which node because the beacons locate the robots, but they did not recognize them. Depending on the information in matrix A_3 , we construct the visible node table according to the following steps:

Step 1: Each row of the visible node table must have information about one visible node and its visible neighbors. The principle that is used to check the visibility of any neighbor is achieved by drawing a straight line between each visible node i and all other nodes in the environment and then, for each line check if any other node intersects it. The intersection means that the furthest node on this line is not visible by the visible node i and this case occurs when the distance h between the center of the other node and the straight line from node i to node j is less than the radius of the other node k . Fig. 7 schematically depicts the approach used in computing distance h as shown in following equations.

$$a = (y_i - y_j) / (x_i - x_j) \quad (6)$$

where (x_i, y_i) and (x_j, y_j) are the coordinates of the visible node i and the node j .

$$b = -1 \quad (7)$$

$$c = y_j - a * x_j \quad (8)$$

$$h = \text{Abs}(a * x_k + b * y_k + c) / \text{Sqrt}(a^2 + b^2) \quad (9)$$

where (x_k, y_k) is the coordinates of the other node k .

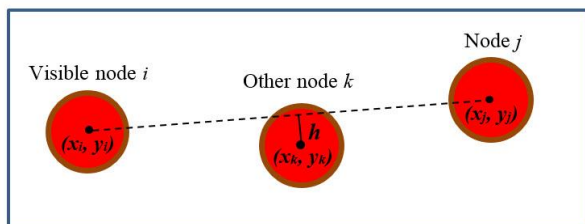


Fig. 7. The distance between node k and the path from the visible node i to the node j .

Step 2: Form a one-row matrix of angles for each visible node. These angles represent the edge between each node and its visible neighbors as shown in Fig.8. Store the number of visible nodes for each node i in the variable $nb(i)$. If the node is recognized as invisible in step 1, the angle will be infinity. Otherwise, the angle α_i^j between the node i and each of its neighbor j is calculated according to the following equation:

$$\alpha_i^j = \tan^{-1} (y_j - y_i) / (x_j - x_i) \quad (10)$$

The matrix A_{vn}^i for each visible node i will be as follow:

$$A_{vn}^i = [\alpha_i^1 \quad \alpha_i^2 \quad \dots \quad \alpha_i^q] \quad (11)$$

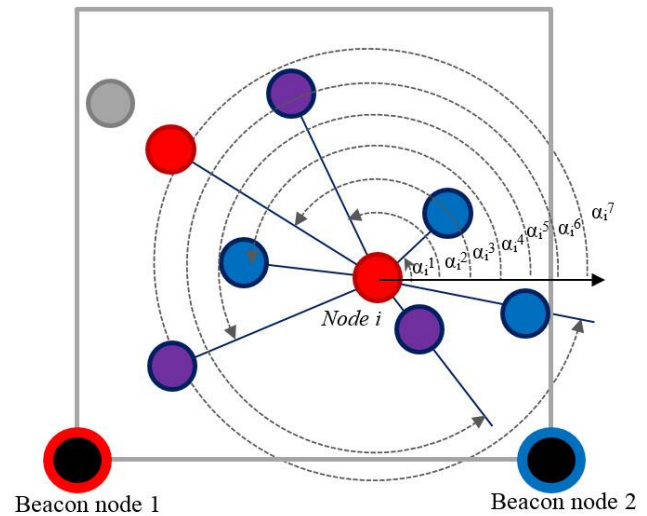


Fig.8. The angles of paths from node i to all its neighboring nodes.

Step 3: For each node i , calculate the angle β_i^j between each of its successive neighbors (j and $j-1$), as shown in Fig.9. At first, the angles in matrix A_{vn}^i is sorted in ascending order and then each angle β_i^j is calculated by subtracting the corresponding successive angles except the infinity ones will be excluded from the computation of β_i^j as in equation 12.

$$\beta_i^j = \alpha_i^j - \alpha_i^{j-1} \quad (12)$$

Where the angle $\alpha_i^0 = 0^\circ$ and the last angle $\beta_i^{nb(i)+1}$ is calculated by using equation 13.

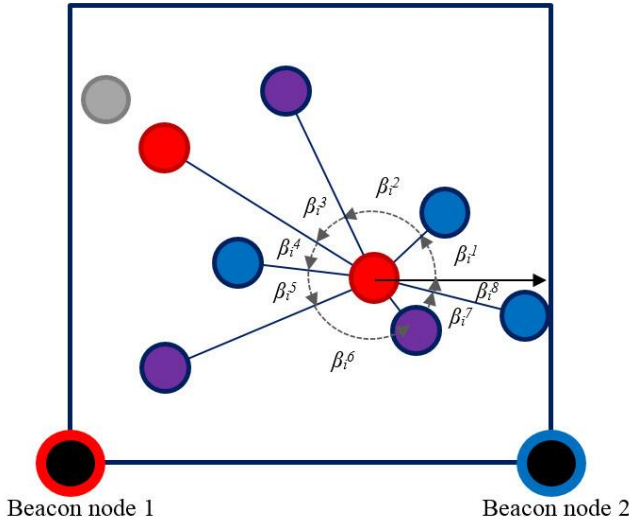


Fig.9. The angles between each next neighbors of node i .

$$\beta_i^{nb(i)+1} = 360 - \alpha_i^{nb(i)} \quad (13)$$

The first angle β_i^1 is adjusted by adding the value of the last angle to its value as in equation 14.

$$\beta_i^1 = \beta_i^1 + \beta_i^{nb(i)+1} = \alpha_i^j + 360 - \alpha_i^{nb(i)} \quad (14)$$

Step 4: Store all the calculated values of the angles β_i^j in visible-node table as shown in table 1. The values of the invisible nodes' angles are represented by ∞ value.

Table 1. The visible-node table

	Neighbor node 1	Neighbor node 2	Neighbor node 3	- -	Neighbor node q
Visible node 1	β_1^1	β_1^2	β_1^3	- -	β_1^q
Visible node 2	β_2^1	β_2^2	β_2^3	- -	β_2^q
Visible node 3	β_3^1	β_3^2	β_3^3	- -	β_3^q
-	-	-	-	- -	-
Visible node q	β_q^1	β_q^2	β_q^3	- -	β_q^q

D. Construction the visible-robot table

Each robot is provided with a distance sensor which rotates at 360° and scans the environment. Depending on the readings of this sensor, we construct the visible-robot table according to the following steps:

Step 1: For each robot i , we form a matrix which contains the angles of detecting other neighboring robots. The angles measurement is started from the motion direction of the robot i , as shown in Fig.10. Use the value ∞ for the angles of invisible robots and store the number of visible robots to each robot i in the variable $rb(i)$.

The matrix for each robot i will be as follow:

$$A_{vr}^i = [\gamma_i^1 \quad \gamma_i^2 \quad \dots \quad \gamma_i^t] \quad (15)$$

Where t is the total number of robots in the environment.

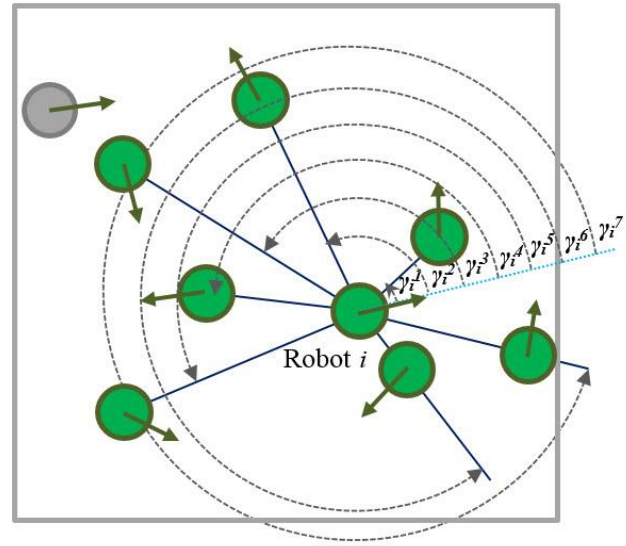


Fig.10. The illustration of the angle between robot 1 and its neighbor robots.

Step 2: For each robot i , use equation 16 to calculate the angle θ_i^j by subtracting the corresponding successive angles between each of its successive neighbor robots (j and $j-1$), as shown in Fig.11.

$$\theta_i^j = \gamma_i^j - \gamma_i^{j-1} \quad (16)$$

Where the angle $\gamma_i^0 = 0^\circ$ and the last angle $\theta_i^{rb(i)+1}$ is calculated by using equation 17.

$$\theta_i^{rb(i)+1} = 360 - \gamma_i^{rb(i)} \quad (17)$$

Where the first angle θ_i^1 is adjusted by adding the value of the last angle to its value as shown in equation 18.

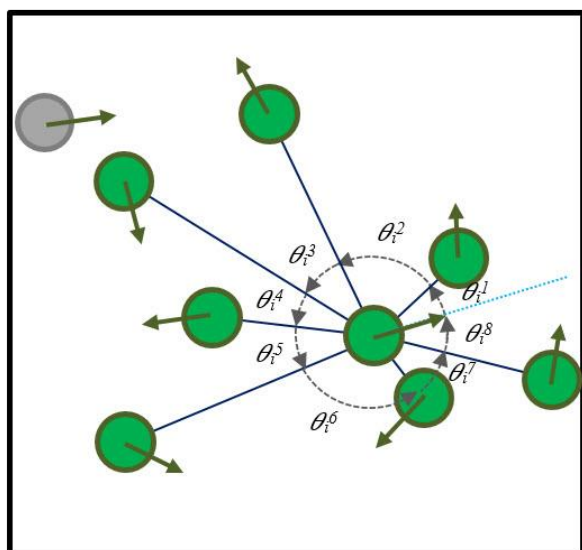


Fig.11. The angles between each next neighbors of robot i .

$$\theta_i^1 = \theta_i^1 + \theta_i^{rb(i)+1} = \gamma_i^j + 360 - \gamma_i^{rb(i)} \quad (18)$$

Step 3: Use the visible-robot table to store all the calculated values of the angles θ_i^j as shown in table 2. The values of invisible robots' angles are represented by ∞ value.

E. Visible robots localization

As mentioned before, the visible node table contains angles which are calculated with the help of information found in matrix A_3 which contains the absolute locations of robots, but it does not recognize the identity of each robot.

Table 2. The visible-robot table

	Neighbor robot 1	Neighbor robot 2	Neighbor robot 3	- -	Neighbor robot t
Visible robot 1	θ_1^1	θ_1^2	θ_1^3	- -	θ_1^t
Visible robot 2	θ_2^1	θ_2^2	θ_2^3	- -	θ_2^t
Visible robot 3	θ_3^1	θ_3^2	θ_3^3	- -	θ_3^t
-	-	-	-	- -	-
Visible robot t	θ_t^1	θ_t^2	θ_t^3	- -	θ_t^t

Also, the visible-robot table provides similar angles to those found in the visible node table, but these angles are calculated from the information gained from the detecting angles of the distance sensor that is installed on each robot. Invisible robot table, the identity of each robot is

known, but the locations of them are not. The matching operation is used for visible-robot localization by taking a row in visible-robot table and scan visible-node table for matched angles. The identity and the location of this robot will be known if the matching occurs. The steps for localization are achieved as follow:

Step 1: Starting from the first row of the visible-robot table and scan all the rows of the visible node table for a node that has the same number of neighbors.

Step 2: When we find a row with a node that has the same number of neighbors then we shall check for matching between the angles in this row and the angles found in the first row of the visible-robot table. If matching occurs, the identity and the location of this robot will be known as shown in Fig.12.

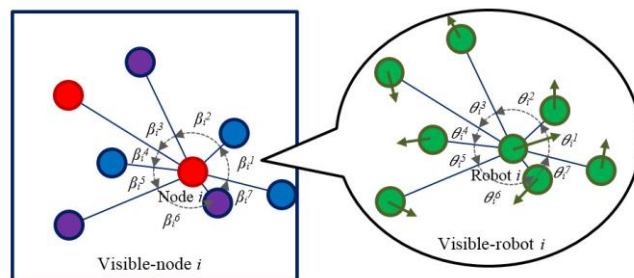


Fig.12. Matching between visible node i and visible robot i .

Step 3: If we did not find any row in the visible node table that contains the same number of neighbors. Then this leads to the fact that some nodes are not found in the visible node table because the beacons have not seen them, and those nodes will be treated as invisible nodes as shown in Fig. 13. On another hand, the same nodes appear in the visible-robot table because they have been detected by the distance sensor placed on the robot. In this case, we scan the visible node table for a row that has a close number of neighbors to that found in the visible robot table and then, we check for matching among the angles in this row with angles in the row of the visible robot table. The unmatched angle in the visible node table must be equal to the sum of the unmatched angles in the visible robot table. If the matching occurs, then the identity and the location of the robot in this row are known as shown in Fig.13.

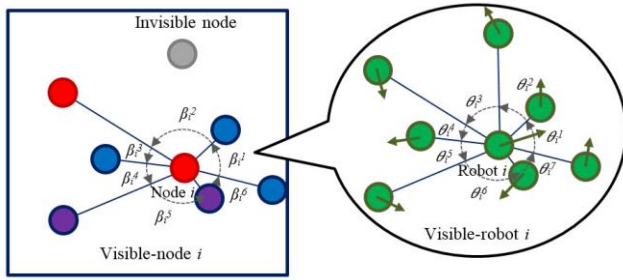


Fig.13. The effect of the invisible node in the matching process.

Step 4: This operation is continued for the other rows in the visible robot table until the end of all rows in the visible node table (the identity and location of robots that matched with the nodes in the visible node table are known).

F. Orientation estimation of the visible robots

This section describes how to calculate the direction of each visible robot. As shown in Fig14, the orientation can be estimated according to the following steps:

Step 1: The angles α_{b1}^i and α_{b1}^j are estimated by scanning, where α_{b1}^i represent the detecting angle of node i and α_{b1}^j represent the detecting angle of node j invisible node environment.

Step 2: The angle between these two robots is computed as below:

$$\omega = \alpha_{b1}^i - \alpha_{b1}^j \quad (19)$$

Step 3: The distance between node i and node j is computed by using equation 20

$$t = ((y_{b1}^j - y_{b1}^i)^2 / (x_{b1}^j - x_{b1}^i)^2)^{0.5} \quad (20)$$

Step 4: The law of sine is used to determine the value of angle σ

$$\frac{t}{\sin \omega} = \frac{d_{b1}^i}{\sin \sigma} \quad (21)$$

Step 5: Compute the value of angle ϕ which is the complementary of the angle σ

$$\phi = 180 - \sigma \quad (22)$$

Step 6: Depending on the scanning of the distance sensor, the angle γ_j^i which represents the angle between the robot and its first neighbor on the left is calculated.

Step 7: The orientation of robot j is calculated as follow:

$$\phi = \lambda + \alpha_{b1}^j - \gamma_j^i \quad (23)$$

G. Invisible robot localization

Some of the robots are invisible to the beacons, the average of these unknown location robots increases as the robots increase in the environment. So, in this section, we suggest a way to estimate the locations and the identities of these robots based on the previous information obtained for the visible robots. The following are the steps used for this purpose:

Step 1: After the matching of table 1 and table 2 had finished, we found that some rows in table 2 are not found in table 1. These rows which represent the invisible robots by the beacons are transferred to a new table could the invisible robot table, as shown in Table 3.

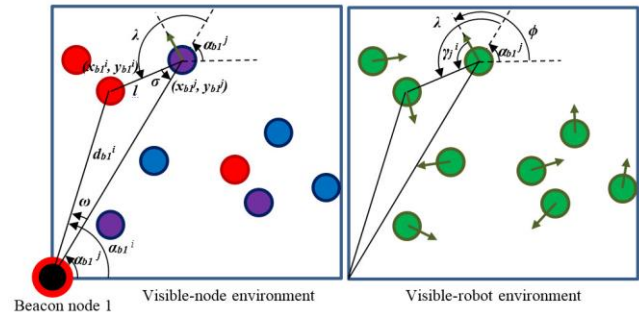


Fig.14. The orientation of visible robot j .

Table 3. The invisible-robot table

	Neighbor robot 1	Neighbor robot 2	Neighbor robot 3	- -	Neighbor robot t
Invisible robot 1	θ_1^1	θ_1^2	θ_1^3	- -	θ_1^t
Invisible robot 2	θ_2^1	θ_2^2	θ_2^3	- -	θ_2^t
Invisible robot 3	θ_3^1	θ_3^2	θ_3^3	- -	θ_3^t
-	-	-	-	- -	-
Invisible robot (t-q)	$\theta_{(t-q)}^1$	$\theta_{(t-q)}^2$	$\theta_{(t-q)}^3$	- -	$\theta_{(t-q)}^t$

Step 2: Each visible robot i has two matrices (A_{vr}^i and A_{vn}^i). Matrix A_{vr}^i contains the angles between robot i and each of its visible neighboring robots which are computed by the distance sensor equipped on every robot and matrix A_{vn}^i contains the angles between robot i and each of its visible neighboring robots which are calculated by the sensor found on beacons. So that, some of the neighboring robots in matrix A_{vn}^i are invisible to the beacons and to estimate the

locations and identities of those robots, we shall match contents of both matrices to detect the angles of the invisible robots. Fig. 15 shows an example of an invisible robot. In the visible node scheme, we found that node 3 is invisible to the beacons and in the visible robot scheme we found the same robot that indicates as robot 3 is visible by robot 1 and has angle γ_1^2 . The computation of the coordinates for this robot is achieved according to the following sub-steps:

1. Use the orientation ϕ of robot 1 and the angle γ_1^2 of robot 3 with respect to robot 1 to compute the angle ψ_1^2 .

$$\psi_1^2 = \phi + \gamma_1^2 \quad (24)$$

2. By using the visible-robot table, we can compute the angle ρ .

$$\rho = 180 - \theta_1^2 - \theta_2^2 \quad (25)$$

3. Compute the distance τ_1 between robot 1 and robot 2.

$$\tau_1 = ((y_2 - y_1)^2 + (x_2 - x_1)^2)^{0.5} \quad (26)$$

4. Use the law of sine to compute the distance τ_2 .

$$\frac{\tau_1}{\sin \rho} = \frac{\tau_2}{\sin \theta_2^2} \quad (27)$$

5. The coordinates of robot 3 are computed using equation 28.

$$\begin{aligned} x_3 &= \tau_2 * \cos \psi_1^2 \\ y_3 &= \tau_2 * \sin \psi_1^2 \end{aligned} \quad (28)$$

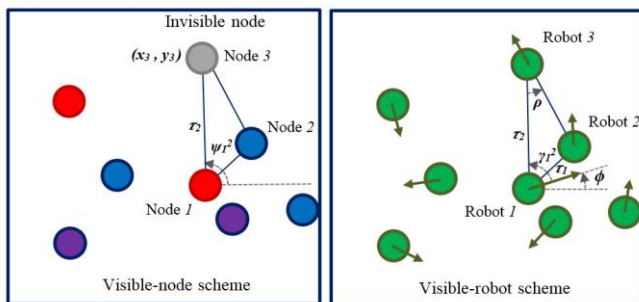


Fig.15. Illustration of a location estimation of invisible robots.

Step 3: The coordinates of each invisible robot are computed in step 2. To calculate the identities of these robots, we need to construct a new table (Table 4) for the invisible nodes as in section 2.3. Each row in this table has the angles between each two successive neighboring robots to one of the invisible node. The identity of each invisible robot is known in table 3 and the coordinates of the same robots are known in table 4. By matching both tables, we can assign the coordinates to the exactly invisible robot. The procedure used to build table 4 is similar to the procedure used to create table 1 but the difference is that table 4 is used for the invisible nodes. Also, the method used to localize the invisible robots is the same as the method used for localizing the visible robots in section 2.5.

Step 4: The orientation estimation for all the invisible robots are achieved with the same procedure as in section 2.6 that is used for the orientation estimation of the visible robots.

Table 4. The invisible-node table

	Neighbor node 1	Neighbor node 2	Neighbor node 3	- -	Neighbor node t
Invisible node 1	β_1^1	β_1^2	β_1^3	- -	β_1^t
Invisible node 2	β_2^1	β_2^2	β_2^3	- -	β_2^t
Invisible node 3	β_3^1	β_3^2	β_3^3	- -	β_3^t
-	-	-	-	- -	-
Invisible node (t-q)	$\beta_{(t-q)}^1$	$\beta_{(t-q)}^2$	$\beta_{(t-q)}^3$	- -	$\beta_{(t-q)}^t$

III. THE SIMULATION RESULTS

The implementation of the table-based matching algorithm is done using visual basic 2012. Simulations are repeated 100 times for different sizes of networks, starting from 5 to 50 robots and for different radiuses (10, 15 and 20 pixels). The robots are placed on an area of 500*500 pixels. The system parameters are:

1. The radius of the robot (R): Changing the radius of the robot will change the opacity and visibilities averages.

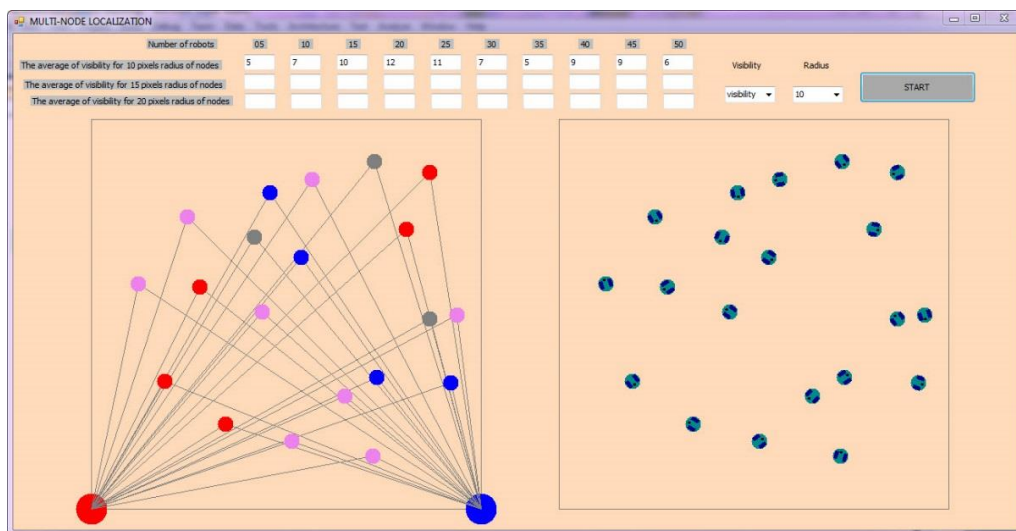
2. The size of the network (N): It refers to the number of robots. Increasing or decreasing the size of the network will increase or decrease the density of nodes in the environment.

3. The distance between beacons and nodes (D): This parameter affects the accuracy of localization estimation i.e. less distance leads to more accurate estimation.

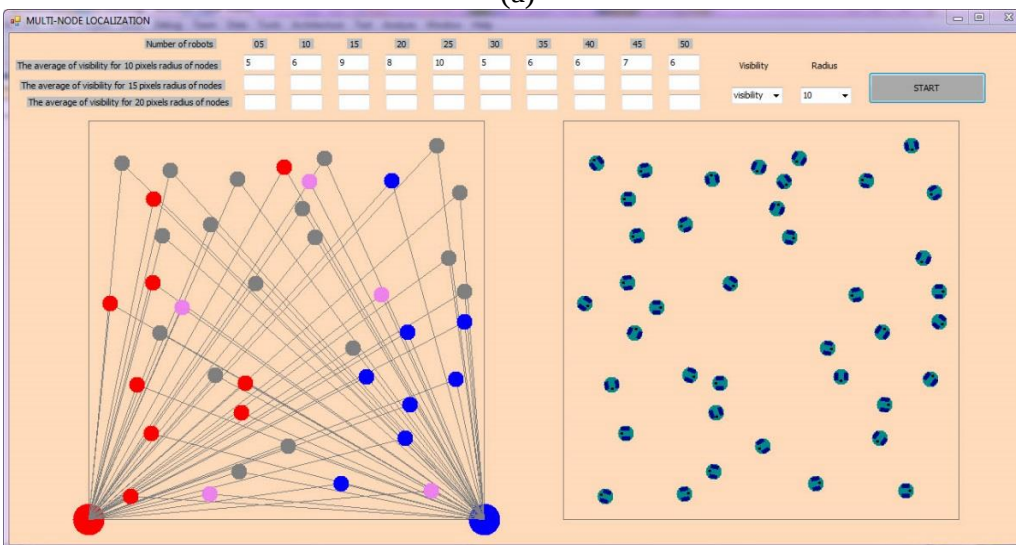
4. The rotating angle (RA): This parameter also affects the localization estimation accuracy, in other words, less angle degree means less error.

We start by studying the process of construction of the visible node table and visible-robot table through three examples represented in Fig. 16, Fig.17 and Fig.18. Fig. 16(a) shows the effects of 20 robots with a 10 pixels radius on the environment visibility. Fig. 16(b) shows the effects on the same environment but with 40 robots. Fig 17(a) and (b) study the effects of 15 pixels radius robot on the visibility of environments with 20 and 40 robots respectively. Fig. 18 is analogous to Fig. 16 and Fig. 17 with 20 pixels radius of robots.

Further simulation examples as shown in Fig. 19, Fig.20 and Fig. 21 are proposed to display the effects of the beacon's rotating angle, the robots' size and the distance between the beacon and each of the robots on the accuracy of localization. For better measurement we choose an environment with 20 robots on a 500×500 pixels area. Fig. 19(a) shows an environment with 20 robots each has 10 pixels radius and a beacon equipped with a distance sensor rotates with 1-degree angle. Fig. 19(b) and (c) represent the same environment but with a distance sensor rotates at an angle of 2 and 3 degrees. Fig. 20 and Fig. 21 represent the analogous of Fig. 19 with robots of 15 and 20 pixels radius respectively.



(a)



(b)

Fig.16. Visible-node table and visible-robot table construction for robots with $R=10$ pixels: (a) $N = 20$ robots ;(b) $N = 40$ robots.

We will use the following performance average metrics for a quantitative analysis of the table-based matching algorithm:

1. The average of visible robots by using beacon 1: This metric test the average of robots which are seen by a distance sensor equipped with the first beacon which is fixed to the bottom left corner of the environment.

2. The average of visible robots by using beacon 2: The average of robots which are seen by the second beacon which has a distance sensor placed at the bottom right corner is tested by this metric.

3. The average of visible nodes by using both beacons: It is metric that tests the average of visibility by both beacon 1 and beacon 2.

4. The average of visible nodes by using either beacons or one of them: It is a metric that tests the average of visible nodes in the environment. The above four metrics affect the accuracy of position and orientation estimation i.e. the accuracy of estimation increases as the visibility of nodes increases.

5. The average of opacity: This metric computes the average of robots which have no available information from beacon nodes. As a solution for opacity, we can use section 2.7 to estimate the location of invisible nodes.

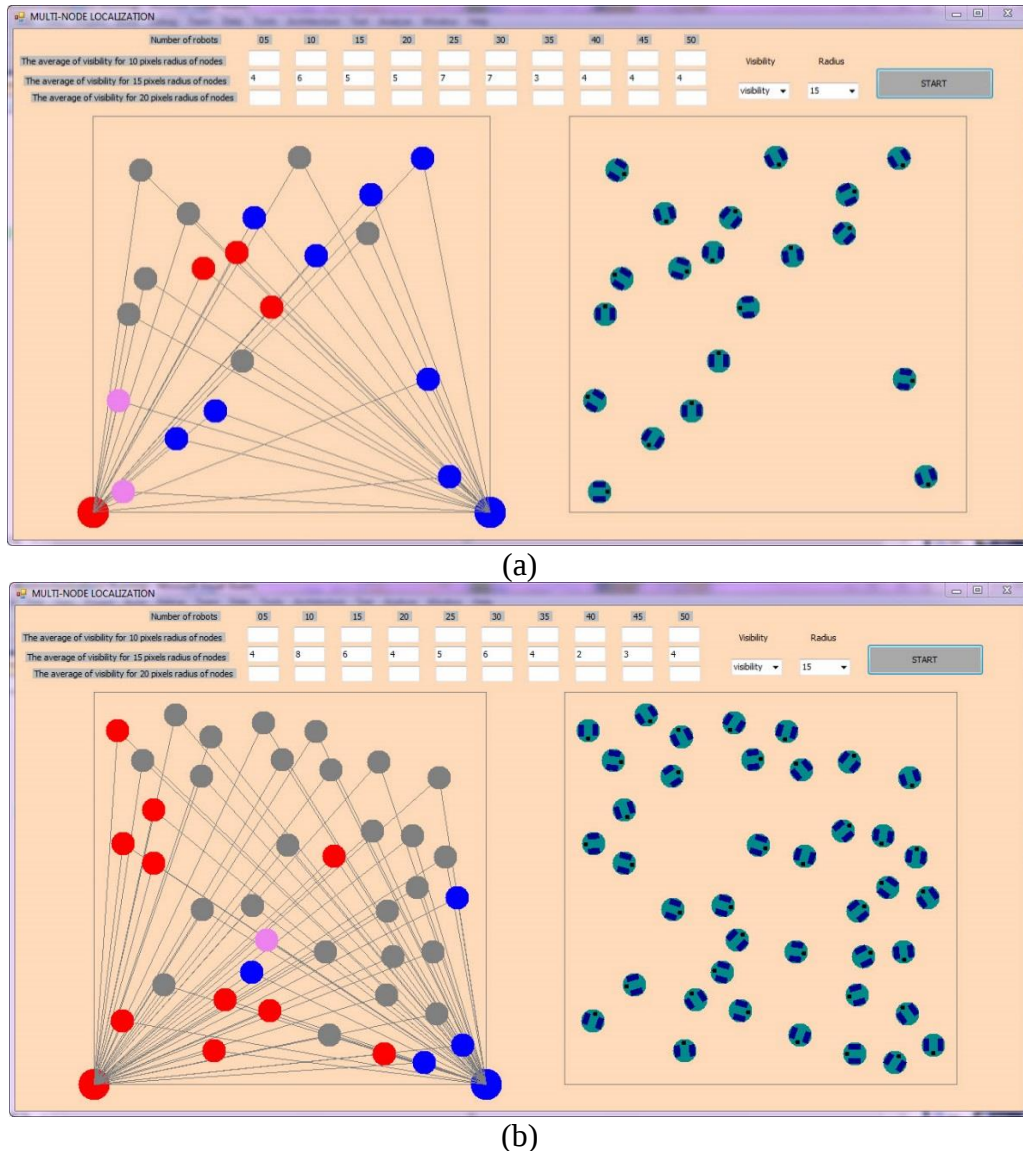


Fig. 17. Visible-nodes table and visible-robot table construction for robots with R=15 pixels: (a) N = 20 robots ;(b) N = 40 robots.

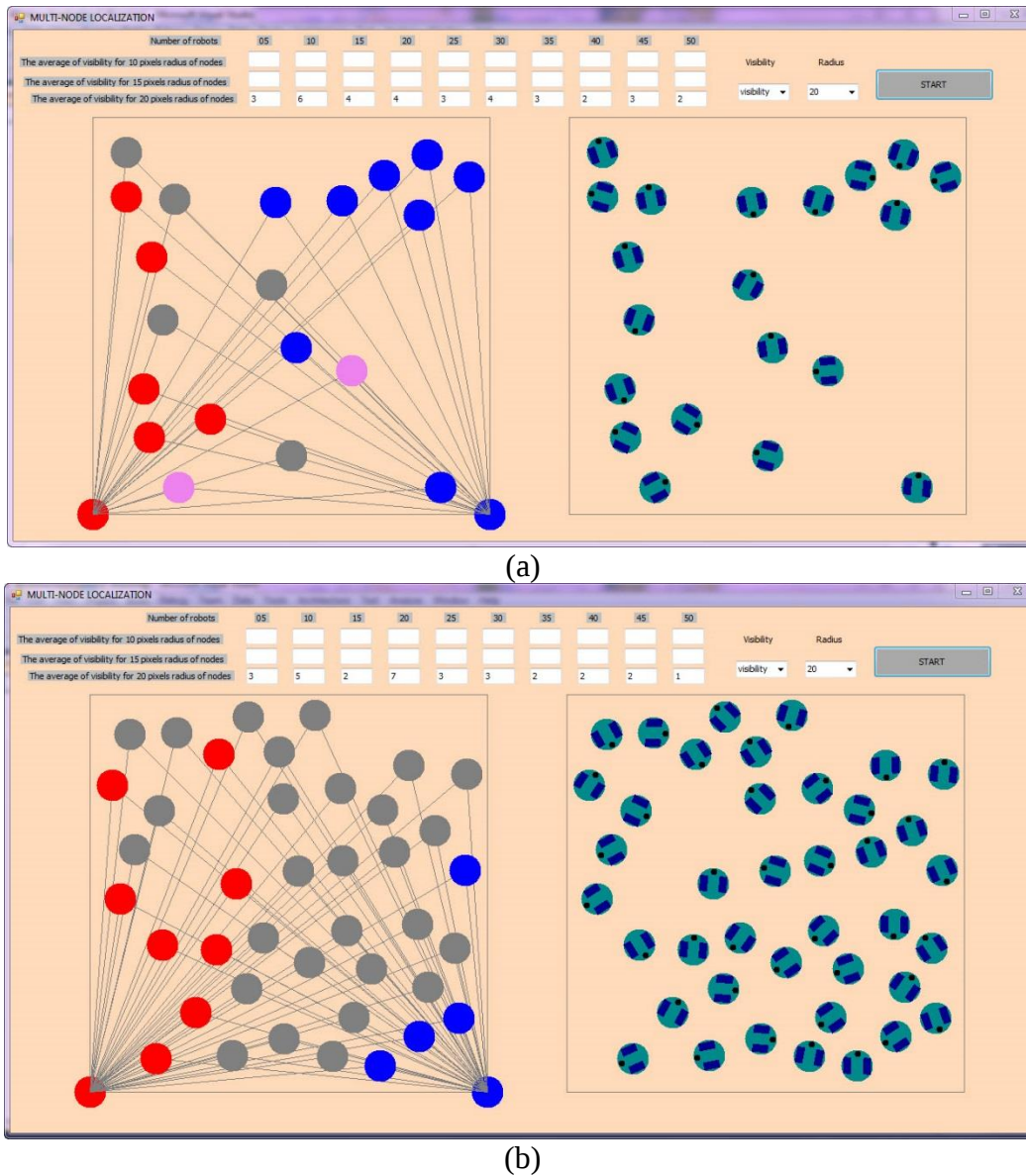


Fig. 18. Visible-nodes table and visible-robot table construction for robots with $R=20$ pixels: (a) $N = 20$ robots ;(b) $N = 40$ robots.

6. The median error: The effects of distance, rotation angle and radius of robots on the accuracy of localization estimation are computed by this metric.

Fig. 22 is used to determine the percentage of node visibility versus the number of nodes in the environment. This study is implemented on nodes with 10, 15 and 20 radiuses on beacon1. This figure shows that the visibility decreases as the number of nodes increases and also as the radius of nodes increases. A similar study for the visibility by beacon 1 is done for beacon 2 as shown in Fig.23.

The red nodes in Fig. 17 are seen by beacon 1; the blue nodes in the same figure are seen by beacon 2 and finally, the purple nodes are seen by both beacons which their visibility percentage is indicated in Fig. 24. The visibility of these nodes is less than the visibility that is shown in Fig. 22 and Fig.23 because this visibility represents the intersection of the two beacon visibilities. Fig. 25 displays the visibility percentage of nodes that are seen by any one of the beacons or by both and for that reason, it is greater than the visibility percentages shown in the last three figures.

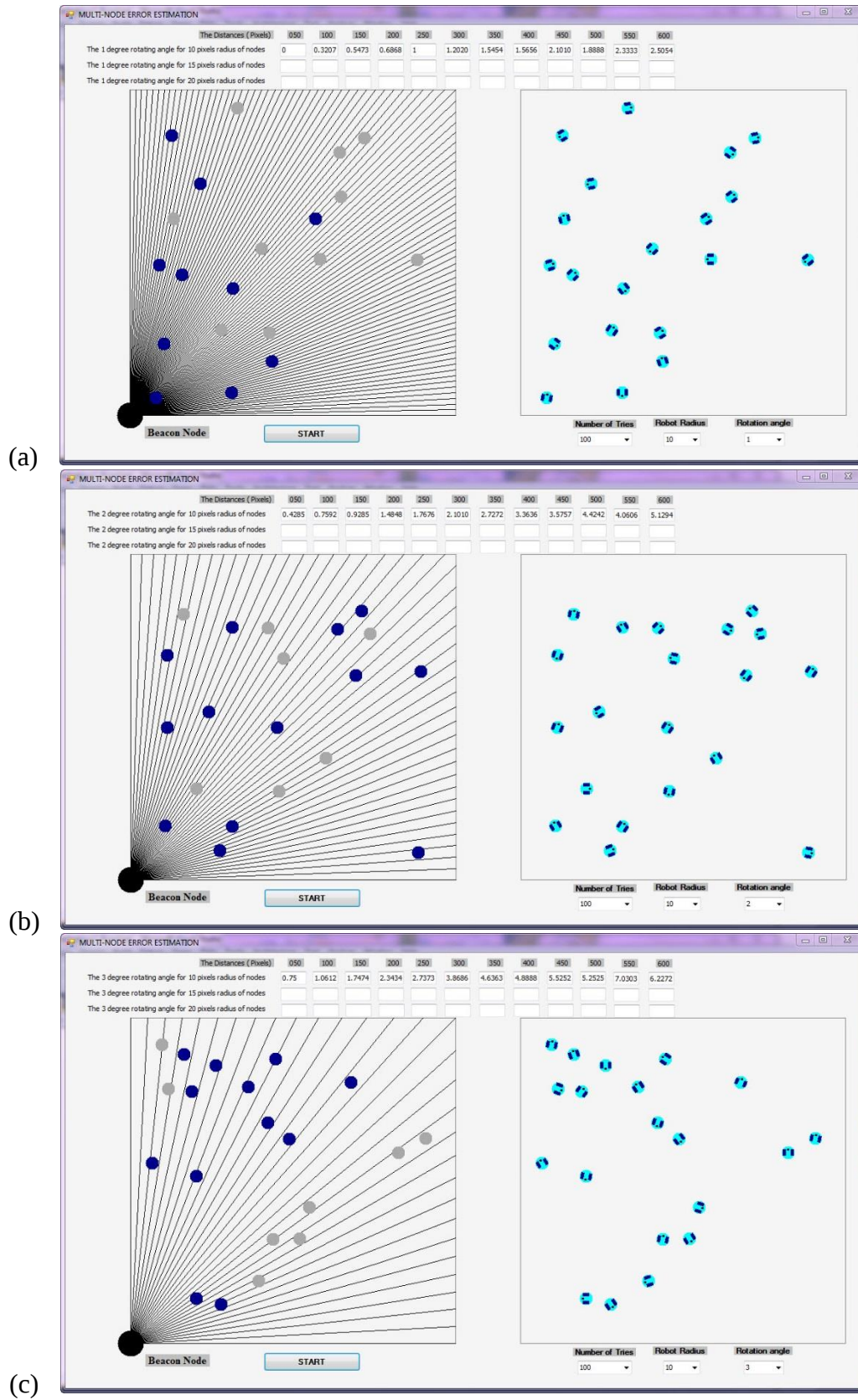


Fig. 19. Median error estimation for robots with $R=10$ pixels: (a) $RA = 1$ degree ;(b) $RA = 2$ degree ; (c) $RA = 3$ degree.

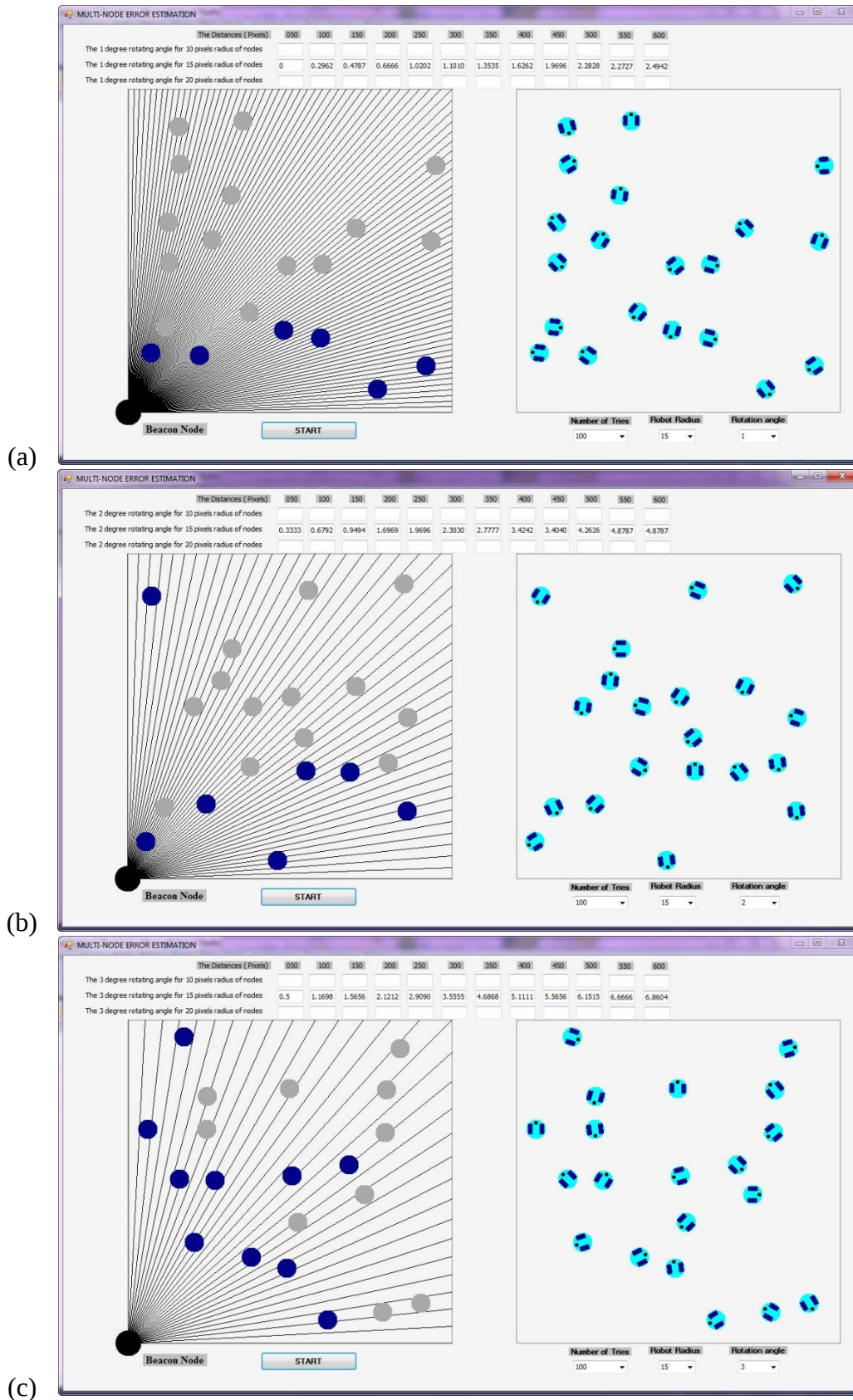


Fig. 20. Median error estimation for robots with $R=15$ pixels: (a) $RA = 1$ degree ;(b) $RA = 2$ degree ; (c) $RA = 3$ degree.

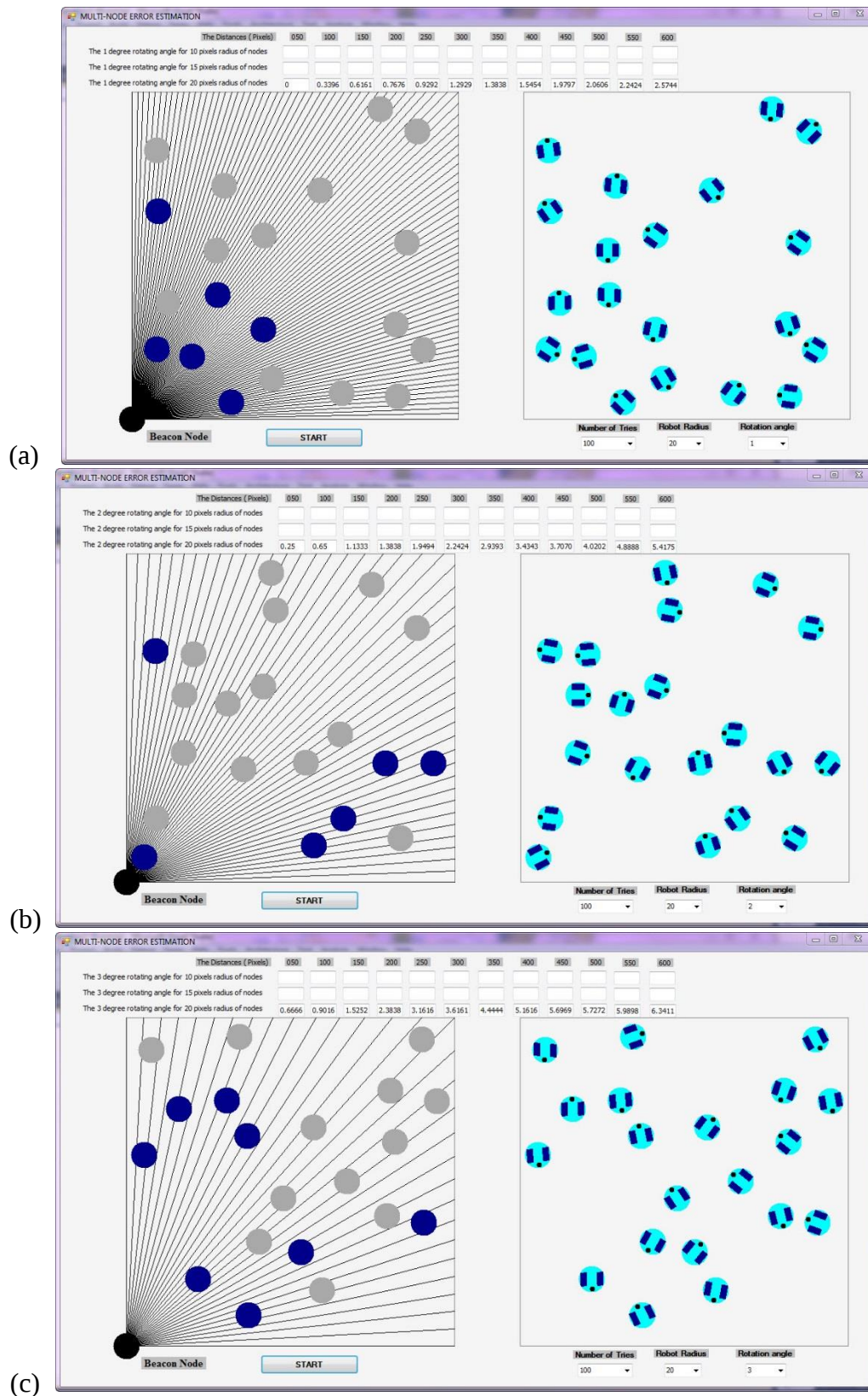


Fig. 21. Median error estimation for robots with $R=20$ pixels: (a) $RA = 1$ degree ;(b) $RA = 2$ degree ; (c) $RA = 3$ degree.

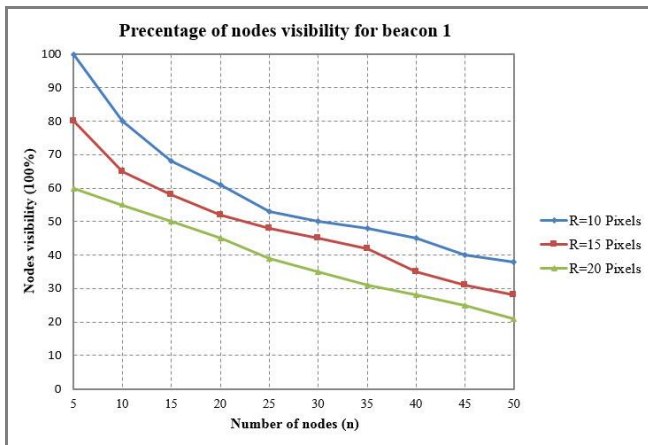


Fig. 22. The percentage of node visibility with respect to beacon 1 vs. n for different values of R.

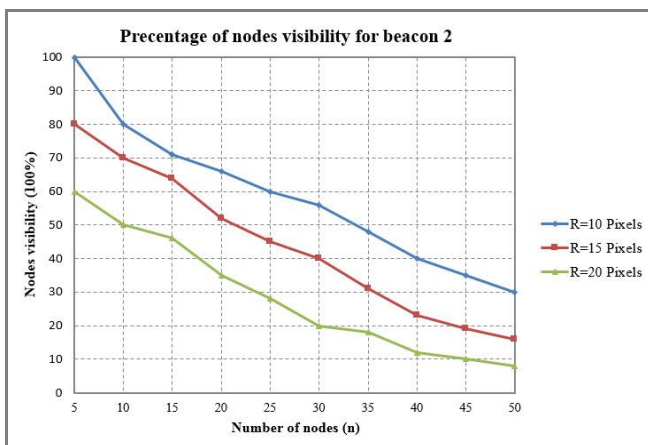


Fig. 23. The percentage of node visibility with respect to beacon 2 vs. n for different values of R.

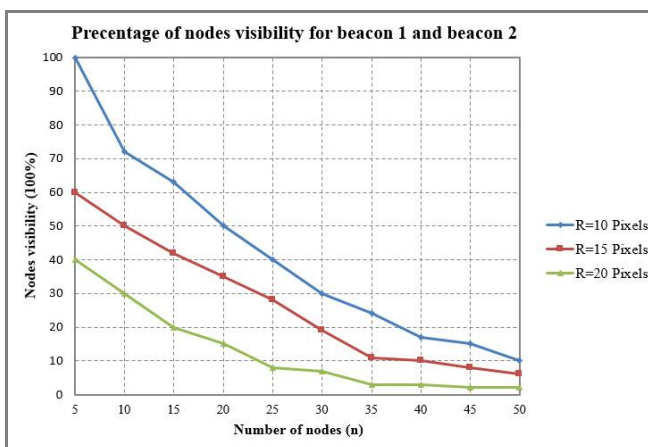


Fig. 24. The percentage of node visibility with respect to beacon 1 and beacon 2 vs. n for different values of R.

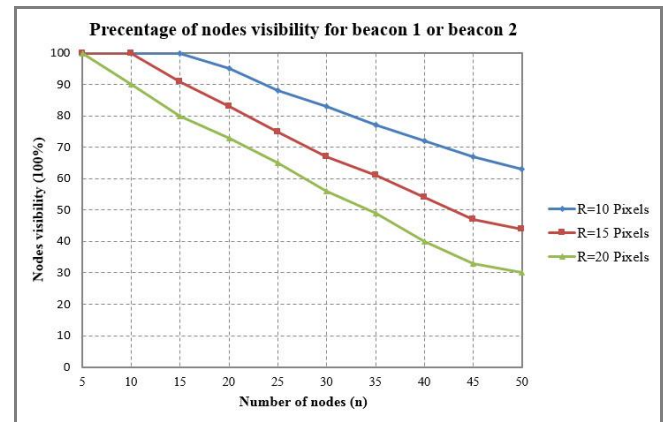


Fig. 25. The percentage of node visibility on beacon 1 or 2 vs. n for different values of R.

In Fig.26, it is clear that the average of invisible nodes increases when both of R and N increase or one of them increases.

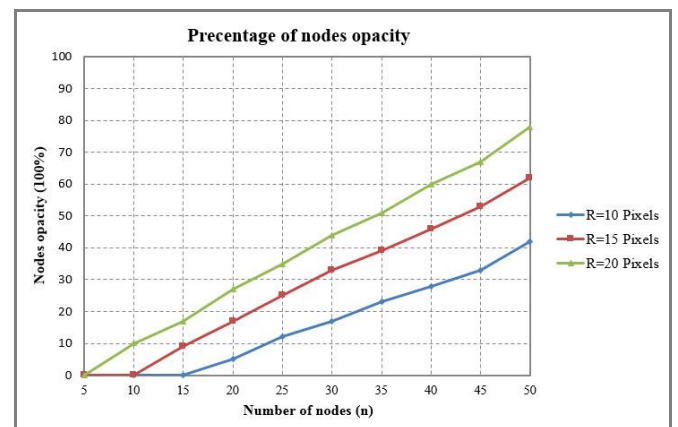


Fig. 26. The percentage of node opacity on beacon 1 and beacon 2 vs. n for different values of R.

Figures (27, 28 and 29) are used to study the average of error in location estimation with respect to D and RA. Increasing the rotation angle will increase the error and reduce the accuracy. Also, increasing the distance between beacons and nodes has the same effect on the error average. On the other hand, the radius of the node has no effect on the average of error because the error represents the perpendicular distance from the node's center to the detection line.

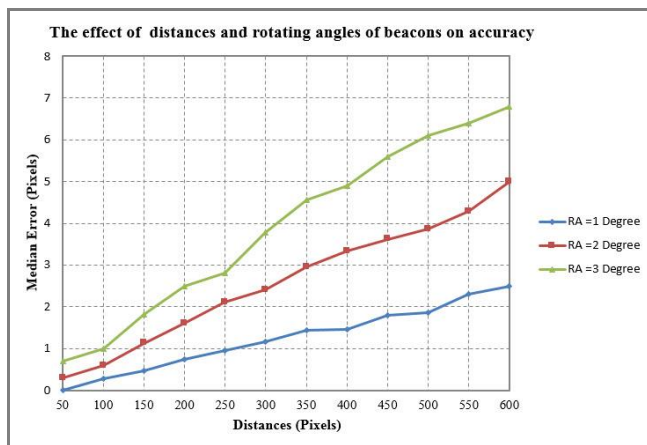


Fig. 27. Median error estimation of the robot position with 10 pixels radius vs. distances, for several values of the rotating angle of beacons.

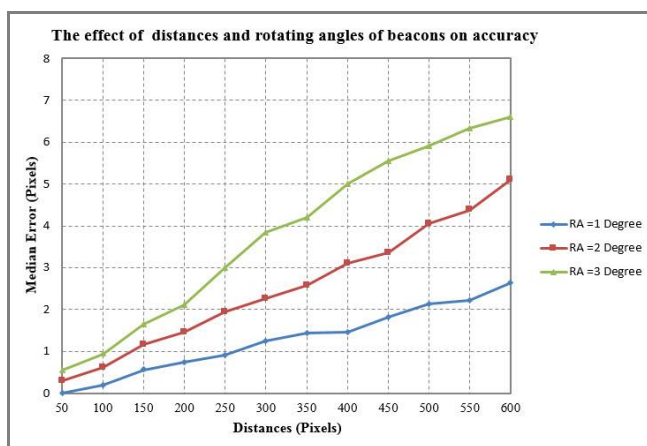


Fig. 28. Median error estimation of the robot position with 15 pixels radius vs. distances, for several values of the rotating angle of beacons.

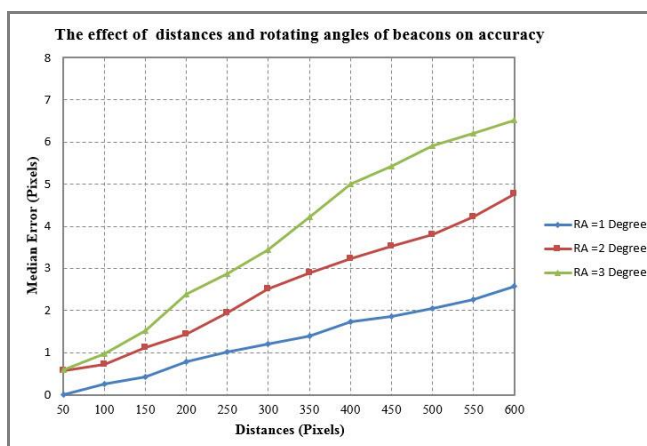


Fig. 29. Median error estimate of the robot position with 20 pixels radius vs. distances, for several values of the rotating angle of beacons.

IV. CONCLUSIONS

This paper introduces a new algorithm for multi-robot (node) localization and orientation estimation. The idea of this algorithm is based on matching the information between two tables (the visible node table and the visible-robot table). The information in the visible node table has been collected depending on distance sensors installed on the beacons at the left and right corners of the frame. On the other hand, the information found in the visible-robot table is collected with the help of a distance sensor fixed on every robot in the environment.

This algorithm has the property of precise calculations for robots' locations and orientations because it uses very accurate distance sensors. Also, this algorithm as compared with other algorithms has an increased visibility average because of the using of two beacons in the system. Another feature of this algorithm is the ability to address all the robots in the system with any parameter such as the number of robots, the radius of robots, the beacons' rotation angle, etc.

The simulation results for our algorithm show that the visibility average is good with respect to the number of robots in the environment but there is an inverse relation between them and to improve the visibility standard in the system we should reduce the radius of robots. The system has a small error average and in the worst case, the error percentage doesn't exceed 8 pixels. It is worth to mention that the error average has a corresponding relation with the beacons' rotation angle and an inverse relation with the distance between the robots and the beacons. The robots in this system are fixed during the calculation of their locations and orientations. So, we suggest improving the system to involve even the mobile robots, and we also recommend using more than two beacons to increase the visibility average.

References

- [1] A. T. Rashid, M. Frasca, A. A. Ali, A. Rizzo, L. Fortuna, "Multi-robot localization and orientation estimation using robotic cluster matching algorithm", Elsevier / Robotics and Autonomous Systems, vol.63, Part 1, pp. 108-121, 2015.

- [2] Z. Y. Ibrahim , A. T. Rashid, and A. F. Marhoon, " An algorithm for Path planning with polygon obstacles avoidance based on the virtual circle tangents", Iraq J. Electrical and Electronic Engineering, Vol. 12, No. 2, pp. 221-234 , 2016.
- [3] Z. Y. Ibrahim , A. T. Rashid, and A. F. Marhoon, " Prediction-Based Path Planning with Obstacle Avoidance in Dynamic Target Environment ", Basrah Journal for Engineering Sciences, Vol. 16, No. 2, pp. 48 – 60, 2017.
- [4] O. A. Hasan, A. T. Rashid, R. S. Ali, J. Kosha, "A practical performance analysis of low-cost sensors for indoor localization of multi-node systems", International Conference on Internet Technologies & Applications, pp. 284-287, 2017.
- [5] K. K. Almuzaini, A. Gulliver, Range-based localization in wireless networks using density-based outlier detection, Wireless Sensor Network. 2, pp. 807-814, 2010.
- [6] A. Kumar, N. Chand, V. Kumar and V. Kumar, Range free localization schemes for wireless sensor networks, International journal of computer networks & communications (IJCNC). Vol. 3, No.6, pp. 115-129, 2011.
- [7] A. T. Rashid, W. H. Zayer and M. T. Rashid, "Design and Implementation of Locations Matching Algorithm for Multi-Object Recognition and Localization "Iraq J. Electrical and Electronic Engineering, Vol.14 No.1, pp. 10-21, 2018.
- [8] N. Dhopre, Sh. Y. Gaikwad, Localization of wireless sensor networks with ranging quality in woods, International journal of innovative research in computer and communication engineering. Vol. 3, No. 4, pp. 2893-2896, 2015.
- [9] A. Cornejo, R. Nagpal, Distributed Range-Based Relative Localization of Robot Swarms, Algorithmic Foundations of Robotics XI, the series Springer Tracts in Advanced Robotics, Vol. 107, pp. 91-107, 2015.
- [10] R. Khadim, M. Erritali, A. Maaden, Range-free localization schemes for wireless sensor networks, International journal of computer, electrical, automation, control and information Engineering. Vol. 9, No. 7, pp. 1452-1459, 2015.
- [11] S. P. Singh, S. C. Sharma, Range free localization techniques in wireless sensor networks: A review, 3rd international conference on recent trends in computing. pp. 7-16, 2015.
- [12] Ch. Zhao, Y. Xu, H. Huang, Sparse localization with a mobile beacon based on lu decomposition in wireless sensor networks, Radio Engineering. Vol. 24, No. 3, pp. 739-748, 2015.
- [13] C. Ning, R. Li, K. Li, Outdoor Location Estimation Using Received Signal Strength-Based Fingerprinting, Wireless Personal Communications, Springer, Vol. 89, No. 2, pp. 365-384, 2016.
- [14] Y. Y. Fan, R. E. Kalaba, J. E. Moore, Arriving on Time, Journal of Optimization Theory and Applications, springer, Vol. 127, No. 3, pp. 497-513, 2005.
- [15] M. Ilyas and I. Mahgoub, Handbook of sensor networks: Algorithms and architectures, New York: Wiley, 2005.
- [16] I. Amundson, J. Sallai, X. Koutsoukos, A. Ledeczi, Radio Interferometry Angle of Arrival Estimation, the series Lecture Notes in Computer Science, springer, 5970, pp. 1-16, 2010.
- [17] Sh. Nasrani, A. Singla, Localization enhancement of wireless sensor networks by using membrane computing, IJITKM, Vol. 8, No. 2, pp. 27-32, 2015 .
- [18] K. Bouchart, D. F. Simard, S. Gaboury, B. Bouchard, A. Bouzouane, Accurate trilateration for Passive RFID Localization in Smart Homes, International Journal of Wireless Information Networks, springer, Vol. 21, No. 1, pp. 32-47, 2014.
- [19] S. Ahmad, R. Lu and M. Ziaullah, Bluetooth an optimal solution for personal asset tracking: A comparison of bluetooth, RFID and miscellaneous anti-lost tracking technologies, International journal of u- and e- service, science and technology. Vol. 8, No. 3, pp. 179-188, 2015.
- [20] J. N. S., M. R. Mundada, Localization techniques for wireless sensor networks, International journal of engineering and technical research (IJETR), Vol.3, No. 1, pp. 30-36, 2015.
- [21] O. A. Hasan, A. T. Rashid, R. S. Ali, "Hybrid approach for multi-node localization and Identification", Basrah Journal for Engineering Sciences, vol.16, No 2, pp. 11-20, 2016.
- [22] O. A. Hasan, R. S. Ali, A. T. Rashid, "Centralized approach for multi-node localization and identification", Iraq J. Electrical and Electronic Engineering, vol.12, No 2, pp. 178-187, 2016.