Vision-based Mobile Robot Map Building and Environment Fuzzy Learning

Khaled Al Muteb
College of Computer and Information Sciences
King Saud University
P.O. Box 51178
Kingdom of Saudi Arabia
muteb@ksu.edu.sa

Abstract—This article describes an AI related procedure to learn the environment and the navigation map for the KSU-IMR (King Saud University-Intelligent Mobile Robotic) system. Beforehand, navigation was routed based wholly on the mobile visual sensory inputs (3D vision). After learning, the mobile robot undertakes the most suitable action while navigating, this is after it found the most suitable route after learning the navigation map. Successful computer routines are used to serve such purposes, resulting on learning maps of occupancy grids while relying on stereo vision SLAM based navigation.

Keywords- Neuro-fuzzy; SLAM; Visual Navigation; KSU-IMR

I. INTRODUCTION

A. Visual Navigation

Visual perception is an important integral for mobile robots, where lots of efforts are put during the design phase of mobile robots. There are considerable amount of efforts to let mobile robots to function within unknown, unstructured environments. In order to achieve such goals, a mobile robot is be capable to observe the surrounding environment. In addition, it should allow a mobile robot to function safely within that environment. In reference to Nickerson et al. [1], Dudek et. al. [2], Brooks et. al. [3], Brooks, [4], typically, almost mobile robots with navigation capabilities within unconstrained environments, do make use of sonar transducers or laser range sensors for their spatial sensing. Recently, mobile robots do make use of advanced computer vision. However, such vision system is usually for feature tracking, or even landmark sensing, where this does not often include occupancy grid mapping or obstacle detection. On the other hand, the work initiated by Moravec and Elfes, [5], and Elfes, [6], which were associated to the occupancy grid mapping, is the most widely used mobile robot mapping method. Both Matthies and Grandjean [7], in their research have provided a very much harsh and depth analysis of range sensing techniques with stereo vision. Their study has shown that, for disparity estimates, and for Gaussian distributed random errors with standard deviations reaching a minimum value of 0.05 pixels. It was also found that, such standard deviations measures were constant, even for different resolutions of images. Matthies and Grandjean [7], have also reported that, “while it is true that they used a different comparison score (sum of squared differences), cameras and calibration, their results show the magnitude of accuracy that can be achieved through careful correlation stereo vision”.

Moravec and Elfes, as presented by Moravec and Elfes [5], has pioneered the work and research of the occupancy grids. Further referring were also reported by Elfes [6], Konolige [8], Murray and James [9], and the formulated in the Carnegie Mellon University by Martin and Moravec [10] as an approach to figure out paradigm for internal representation of stationary defined environment, with an evenly spaced grids. Apart from being used directly for sensory data fusion, there also exist interesting variations of evidence grids, such as place-centric grids as in Youngblood et. al. [11], histogram grids as in Koren and Borenstein, [12] and response grids as in Howard and Kitchen [13]. Vision-Based Mobile Robotic Architecture.

In this sense, this main contribution of this research, is toward the use of vision generated maps for mobile robot navigation. Learned maps are therefore used in making the right motion decisions while navigating.

II. KSU-IMR SYSTEM ARCHITECTURE

System architecture is shown in Fig. (1). The main purpose of this research has been entirely focused towards the development of Neuro-Vision system for mobile robots path planning. Range finders capture very slight properties of the real environment where a mobile robotic system is to move.
A. Navigation Methodology

The followed approach is as follows: Tasks; Building intelligence for mobile robot path planning. This is be achieved by creating navigation intelligence capabilities while robot is in motion. The main intelligent technique that will be used in this research are based on neural net architectures. These techniques will be developed and implemented on real time bases for an enhanced mobile robot navigation. The KSU-IMR is achieved while incorporating the a number of interrelated functionalities, as in Fig. (1).

B. System Software and Low Level Coding.

The main tasks performed by the computers are the 3D depth measurement of the environment, visual analysis, and real-time mobile path planning. Another unique characteristics to be acquainted by the robot, is its visual 3D perception that uses both depth from focus defocus and the 3D binocular stereo. Building intelligence for mobile robot path planning. This be achieved by creating navigation intelligence capabilities while robot in motion. This will be rather based on intelligent path planning techniques. The four important stages are (Localization, Map building and updates, search optimal path planning, and motion control routine). The adopted neural learning architecture is shown in Fig. (2) and Fig. (3).

C. Path Searching, A*.

Along its way A* search algorithm passes through the mapping graph, hence it surveys for the best path of the lowest known cost, while updating a “sorted priority queue” of different path divisions. In a continuous search till a final goal is found, during a traversing of a mobile detected map, a section of a path being traversed would be given a higher cost than another encountered path segment. It leaves the higher-cost path segment and continues to search for lower-cost path sections. A "heuristic estimate" for the distance to the mobile desired posture, as named by $h(x)$. The $h(x)$ term of the $f(x)$ function, must show an admissible heuristic; that is, it must not overestimate the distance to the goal. For a use of the $A^*$ in mobile robot routing, the function $h(x)$ might characterize by a straight-line distance to the final mobile goal. If the heuristic $h(x)$ satisfies the additional condition ($h(x) \leq d(x,y)+h(y)$) for every edge $(x,y)$ of the graph (where $d$ denotes the length of that edge), $h(x)$ is known to be consistent. $A^*$ can assume an effect implementation, i.e. no node needs to be processed more than once, $A^*$ is equivalent to running Dijkstra's algorithm, however, with the reduced cost:

$$d'(x,y) = (d(x,y) - h(x) + h(y))$$

A main concern of the $A^*$ search, is related to time complexity. However, for a tree traversing search, this is a polynomial. Since there is only a single goal posture, the mobile must move to, hence the heuristic function $h(x)$ is to satisfy the following condition:

$$\|h(x) - \hat{h}(x)\| = O(\log h^*(x))$$

In (2), $\hat{h}(x)$ is defined as the OPTIMAL HEURISTIC, the exact cost to get from $x$ to the final mobile robot target position.

D. Mapping Technique with Occupancy Grids.

An occupancy grid splits a space where the mobile robot to move, into small discrete grids. It then assign each grid location a numerical value. Such numerical value is associated with the probability that the location is occupied or not by an obstacle. Before the mobile robot starts a maneuver, all assigned grid values are set to a medial value. Murray and James [9] has indicated that, “our own experiments with sub-pixel interpolation indicate that the Triclops stereo vision module produces results with standard deviations well below one pixel”. We can still approximate a model for the mobile robot stereo vision by the adopting the following approximated relation of (3):

$$P(d|\beta) = 1 \quad \text{for} \quad \beta = \beta(d+0.5) \rightarrow \beta(d-0.5)$$

$$P(d|\beta) = 0 \quad \text{otherwise}$$

This work is supported by NPST program by King Saud University (Project No. 08-ELE300-02).

Figure 2. A five layers Neuro-fuzzy architecture. Environment maps learning.

Figure 3. Mobile robot fuzzy decision making.
Typically, for the employed stereo vision system, while aligning the optical axes, i.e., with a focus at infinity, we defined a relation of disparity \( d \) to depth \( \beta \) as:

\[
\beta(d) = \left(\frac{f_y}{d}\right) \beta \tag{4}
\]

It resembles an elliptical area used as a model in Matthies and Shafer [14]. In reference to the geometry of triangulation Guilherme et al. [15], it is need to evaluate (for a given pixel and disparity) the region of uncertainty.

### E. Mobile SLAM

In this regard, for all within this paper, we shall use the nonlinear mobile robot model, in addition to nonlinear observation models. This includes the system state, such as position and orientation of the mobile, in addition to the position of all landmarks. We shall denote the state of the mobile robot as \( X_i(k) \). Dynamical motion of the mobile robot is modeled by linear discrete-time state transition as in (5):

\[
X_i(k+1) = F_i(k) x_i(k) + u_i(k+1) + V_i(k+1) \tag{5}
\]

In which \( F_i(k) \) is the well-known state transition matrix, \( u_i(k) \) is the control inputs, and \( V_i(k) \) is the uncorrelated mobile dynamics noise errors, as expressed with zero mean and covariance \( Q_i(k) \). We shall name the locality of an \( i \)th feature (landmark), by \( p_i \). Defining, a state transition matrix of an \( i \)th observed feature is expressed as

\[
p_i(k+1) = p_i(k) = p_i \tag{6}
\]

For the time being, we shall let the number of features (landmarks) of a size \( N \) vector, as they are stationary features. Vector of all \( N \) landmarks is denoted:

\[
p = (p_1^T \ p_2^T \ \cdots \ p_N^T)^T \tag{7}
\]

We shall extend \( x(k) \), the state vector to contain also the state vector of the features of localities, and is denoted by:

\[
x(k) = (x_i^T \ p_1^T \ \cdots \ p_{N+1}^T)^T \tag{8}
\]

This leads to the following entire mobile robot and features state transition model:

\[
\begin{bmatrix}
x(k+1) \\
p_i(k+1) \\
p_{i+1} \\
p_N \\
\end{bmatrix} = 
\begin{bmatrix}
F_i(k) & 0 & \ldots & 0 \\
0 & I_i & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & I_{N+1} \\
\end{bmatrix} 
\begin{bmatrix}
x(k) \\
p_i \\
p_{i+1} \\
p_N \\
\end{bmatrix} + 
\begin{bmatrix}
u_i(k+1) \\
O_i \\
O_{i+1} \\
O_N \\
\end{bmatrix} + 
\begin{bmatrix}
y_i(k+1) \\
O_i \\
O_{i+1} \\
O_N \\
\end{bmatrix} \tag{9}
\]

In (9), \( I_i \) is an \( \Re^{(p_i \times p_i)} \) as put as identity matrix. In addition, the matrix \( O_i \) is the \( \dim(p_i) \) null vector.

### F. Mobile Robot Motion Predication Phase

In reference mobile body allocated frames, and for the case of ‘C as a vector of coordinates of an \( i \)th defined landmark, where there are \( k \) landmarks, as state vector is hence defined by: \( \{x_i(k), y_i(k), \theta_i(k), e_1, \ldots, e_k\}^T \). For the PowerRob mobile robot system, it is equipped with \( (\theta_{k}, \Delta r_k) \), i.e., differential drive in which are the right and left angular displacement of the respective wheels is known (measured). For instance, if the mobile robot wheels rates is considered contact during one sampling period, hence we can decide the mobile motion kinematics geometric models. This is expressed as:

\[
\begin{bmatrix}
x(\tau) \\
y(\tau) \\
\theta(\tau) \\
e(\tau) \\
\end{bmatrix} =
\begin{bmatrix}
x_{k-1} \\
y_{k-1} \\
\theta_{k-1} \\
e_{k-1} \\
\end{bmatrix} +
\begin{bmatrix}
\Delta x(\theta_{k-1} + \Delta \theta_{k-1}) - \sin(\theta_{k-1}) \\
\Delta y(\theta_{k-1} + \Delta \theta_{k-1}) - \cos(\theta_{k-1}) \\
\Delta \theta_{k-1} \\
\Delta e_{k-1} \\
\end{bmatrix} \tag{10}
\]

In (10), the terms \( (\Delta x, \Delta \theta) \) are representing the “linear” and “angular” displacement of the mobile robot. In terms of mobile physical parameters:

\[
\begin{align}
\Delta x &= \frac{(\Delta \theta_{k-1} r_k + \Delta \theta_{k-1} r_k)}{2} \\
\Delta \theta &= \frac{(\Delta \theta_{k-1} r_k + \Delta \theta_{k-1} r_k)}{2} \tag{11}
\end{align}
\]

### G. MAP Model Updating

While building a MAP, this entirely dependent on voxels. However, this needs to updated the state of each voxel by adopting \( K_{i,j} : 0 \leftrightarrow 1 \) a credibility value. In this sense, such a CREDIBILITY MEASURE defines a measure to trust a given observation \( O_i(V_i) \) of the voxel \( (i) \) calculated based on the stereo pair taken at a time instant \( (\tau) \). Such a map update is computed based on time index consideration, i.e., for a defined time \( t \), and for an updated occupancy observation \( O_{i,j}(V_i) \), the corresponding voxel (state) is updated by:

\[
\begin{bmatrix}
S_{i,j}(V_i) \\
S_{i,j}(V_i) \\
0 \\
\end{bmatrix} = \begin{bmatrix}
(t - k_{i,j})S_{i,j}(V_i) + k_{i,j}O_{i,j} \\
(t - k_{i,j})S_{i,j}(V_i) + k_{i,j}O_{i,j} \\
0 \\
\end{bmatrix} \tag{12}
\]

\( K_{i,j} \) is depending on a number of time varying terms. Such a dependency does include determined voxel neighborhood homogeneity, quantity of preceding measurements, the period of last observation. Already occupied voxel is not likely to be found in an otherwise empty environment. Therefore, measurements demonstrating homogeneous
sections, are further expected to be (credible) and we don't want to trust the very first measurements and over aged measurement at a point too much. If $O(i)(V_i')$ is designated as neighborhood HOMOGENEITY of an observation, hence $O(i)(V_i')$ is found through a use of a set of $(N)$ voxel within neighborhood of $(V_i')$. To achieve such a theme, therefore, (13) expresses $H_{(i)}$, the homogeneity of an observation at a voxel $(V_i')$ at a time instant $t$.

$$H_{(i)} = \left\{ \frac{\left( \sum_{j=1}^{N} o_i(v - o_i(v')) \right)}{|N|} \right\}$$  \hspace{1cm} (13)

$k_{(i)}$ in (13) is the credibility measure is expressed in terms of homogeneity of an observation $H_{(i)}$, by (14):

$$k_{(i)} = \left\{ \frac{N_{(i)}(t - H_{(i)})}{\sqrt{2\pi}} \right\} \exp\left\{ -\frac{(t - H_{(i)})^2}{2\sigma^2} \right\}$$  \hspace{1cm} (14)

In (14), the term $N_{(i)}$, is representing counts of preceding observations. $N_{(i)}$ is evaluated for $V_i'$, the voxels, where such calculations continuous over the time $(t)$. Furthermore, in (14), $(t_{\text{last}})$ is the time of the last observation, as calculated for the voxel $V_i'$, and finally, $\sigma$ is representing a constant for the age scaling. Meta information (previous observations age, prior observations counts), are important data to be updated, such updates are stored in each voxel.

The goal is then to recursively compute at each time step $k$ the set of samples $S$, that is drawn from $p(x_k | Z^{k-1})$. A particularly elegant algorithm to accomplish this has recently been suggested independently by various authors. It is known alternatively as the bootstrap filter [16], the Monte-Carlo filter [17] or the Condensation algorithm [18],[19],[20]. These methods are generically known as particle filters, and an overview and discussion of their properties can be found in Doucet [22]. In analogy with the formal filtering problem outlined in Section 2, the algorithm proceeds in two phases. FIRST: The PREDICTION PHASE: In the first phase we start from the set of particles $S_{(k-1)}$ computed in the previous iteration, and apply the motion model to each particle $S'_{(k-1)}$ by sampling from the density $p(x_k | s_{(k-1)}, u_{(k-1)})$: (i) for each particle $S'_{(k-1)}$: draw one sample $s_{(k)}$ from $p(x_k | s_{(k-1)}, u_{(k-1)})$.

In doing so a new set $S_k$ is obtained that approximates a random sample from the predictive density $p(x_k | Z^{k-1})$. The prime in $S_k$ indicates that we have not yet incorporated any sensor measurement at time $k$. In reference to [23], we use the motion model and the set of particles $S_{(k-1)}$ in order to build an empirical predictive density function of:

$$p(x_k | Z^{k-1}) = \sum_{t=1}^{N} p(x_k | s_{(k-1)}, u_{(k-1)})$$  \hspace{1cm} (15)

In (15), we describe a blended density approximation to $p(x_k | Z^{k-1})$.

III. Fuzzy Models For Navigation Maps

A. Fuzzy Models

Fuzzy models can also be applied to systems that are well understood but due to the nonlinearities untraceable with standard linear methods. Rule-based structure of fuzzy models allows for integrating heuristic knowledge with information obtained from process measurements. Fuzzy sets are used to define the process operating conditions such that fuzzy dynamic model of a nonlinear process can be described in the following way, Fig. (4):

$$R_i: \text{If operating condition}$$

Then

$$\hat{y}_i(k) = \sum_{j=0}^{n} a_j y(k-j) + \sum_{j=1}^{r} b_j u(k-j) \hspace{1cm} (i = 1, 2, \ldots, r)$$  \hspace{1cm} (16)

The final model output is obtained through the center of gravity defuzzification as in (17):

$$\hat{y}(k) = \frac{\sum_{i=1}^{r} \mu_i \hat{y}_i(k)}{\sum_{i=1}^{r} \mu_i}$$  \hspace{1cm} (17)

In (16) and (17), $\hat{y}(k)$ is the process output, $u$ is the process input, $\hat{y}_i$ is the prediction of process output in the $i^{th}$
operating region, \( r \) is the number of fuzzy operating regions, \((i)\) and \((o)\) are the time lags in the input and the output, respectively, \( \mu_i \) is the membership function for the \( i^{th} \) model, and finally, \( a_{ij} \) and \( b_{ij} \) are the ARMAX model parameters.

The membership function for an operating region is constructed in a number of ways. One approach to calculate its membership function as follows:

\[
\mu_i = \min(\mu_k(x), \mu_m(y)) \\
\mu_i = \mu_b(x), \mu_m(y)
\]

In (18), \( \mu_i \) is a membership function in \( i^{th} \) operating region, \( \mu_k(x) \) is membership function of \( x \) being ‘low’, and \( \mu_m(y) \) is the membership function of \( y \) being ‘medium’. In (17), we calculate the gradient required for gradient based network training.

B. Neuro-Fuzzy Models.

Fuzzy model described in section (A) can be represented by a special type of network topology which is termed here a neuro-fuzzy. Fuzzy reasoning is capable of handling uncertain and imprecise information while a neural network is capable of learning from examples. Neuro-fuzzy intend to combine the advantages of both fuzzy reasoning and neural networks. For simplicity, we assume the fuzzy inference system under consideration has two inputs \( x(k-l) \) and \( y(k-L) \) one output \( y(t) \). For instance, if the rule base contains two fuzzy if-then rules of Takagi and Sugeno’s type, then a rule can be written as:

**Rule 1:** If \( x(k-l) \) is \( A_i \) and \( y(k-2) \) is \( B_j \), then \( f_s = p_x x(k-l) + q_i y(k-2) + r_i \)

**Rule 2:** If \( x(k-l) \) is \( A_2 \) and \( y(k-2) \) is \( B_j \), then \( f_s = p_x x(k-l) + q_i y(k-2) + r_s \)

where \( p, q, \) and \( r \) are constants and called parameter set. That is, the if parts of the rules are same as in the ordinary fuzzy if-then rules, then parts are linear combinations of the input variables. The employed neuro-fuzzy architecture is shown in Fig. (2), where node functions in the layers are described. **First layer:** Each \( i^{th} \) node in this layer is a square node with a node function

\[
O_1 = \mu A_{i} x(k-l)
\]

where \( x(k-l) \) is the input to \( i^{th} \) node, and \( A_i \) is the linguistic label \( \text{small}, \text{large}, \text{.. etc.} \) associated with this node function. In other words, \( O_1 \) is the membership function of \( A_i \) and it specifies the degree to which the given \( x \) satisfies the quantifier \( A_i \). Usually we choose \( \mu A_{i} x(k-l) \) to be bell-shaped with maximum equal to UNITY and minimum equal to ZERO, such as:

\[
\mu A_{i} x((k-l)) = \frac{1}{1+ \left( \frac{x(k-l) - c_i}{a_i} \right)^2} \\
\mu A_{i} x((k-l)) = \exp \left( \frac{-x(k-l)^2}{a_i} \right)
\]

where \( \{a_i, b_i, c_i\} \) is the parameter set. As values of these parameters change, membership shaped functions vary accordingly, thus exhibiting various forms of membership functions on the linguistic label \( A_i \). **Second layer:** Every node in this layer is a circle node which multiplies incoming signals and sends their product out. For instance,

\[
y_i = \mu A_{i} x(k-l) \times \mu B_{j} y(k-2) \quad i = 1, 2
\]

Each node output represents the firing strength of a rule. **Third layer:** Every node in this layer is a circle node. The \( i^{th} \) node calculates the ratio of the \( i^{th} \) rule's firing strength to the sum of all rules' firing strengths:

\[
f_i = \frac{y_i}{y_1 + y_2 + \ldots + y_i} \quad i = 1, 2
\]

**Fourth Layer:** Every node \( i \) in this layer is a square node with a node function

\[
O^4_i = \bar{y}_i f_i = \bar{y}_i (p_x x(k-l) + q_i y(k-2) + r_i)
\]

where \( \bar{y}_i \) is the output of layer 3, and \( \{p_i, q_i, r_i\} \) is the parameter set. Parameters in this layer will be referred to as consequent parameters. **Layer 5:** The node in this layer is a circle node. It computes the overall output as the summation of all incoming signals, i.e.,

\[
O^5 = \text{overall output} = \sum_i \bar{y}_i f_i = \sum_i \frac{y_i f_i}{\sum_i y_i}
\]

From the designed neuro-fuzzy architecture shown in Fig. (2), it is observed that, given values of premise parameters, the entire output is expressed as a linear combinations of the consequent parameters. More precisely, output \( \hat{y} \) in Fig. (2) can be rewritten as:

\[
\hat{y}_m = \tilde{y}_m + \tilde{y}_m + \ldots + \tilde{y}_{m-1}
\]

where \( \tilde{y}_m = (\tilde{y}_m x(k-l)) p_1 + (\tilde{y}_2 y(k-2)) q_1 + (\tilde{y}_3 r_1)

which is linear in the consequent parameters \( (p_1, q_1, r_1, p_2, q_2, \text{.. etc.)} \). The consequent parameters thus identified are optimal (in the consequent parameter space) under the condition that the premise parameters are fixed.
A grid-based mapping approach was selected for the implementation of SLAM algorithm. The implementation ensured that real-time execution, map-accuracy via modeling un-certainty and loop-closure detection is well-implemented within the mapping module. The size of the map currently stands at \((900m^2)\). The mapping updates and initializations are restricted to the area under observation, which roughly equals \((15m^2)\), thus any increase in map area will not trigger map-wide initialization or update. The mobile robot has successfully reached its target \((x,y)\) location using the planned paths in an autonomous approach. We submit a set of goals to our system, hence the system plans path in sections for each of the goal. Once an obstacle scenario is significantly changed so much, in such a way that it affects the planned path, a algorithm is used to re-plan the path to the goal. Specifically, some of the path planning examples and runs are shown in Fig. (5).

**Figure 5.** Successful path planning and learning occupancy grip clues.

**V. CONCLUSIONS**

This article has presented an ongoing research effort to learn a mobile robotic system navigation maps. Maps have been created while relying on results coming SLAM routines. The learning system was a five layers Neuro-fuzzy architecture that has the ability to be trained for a number of navigation situations. Initial results are showing excellent convergences of training, thus an ability learn the environments even at some critical situations. The next phase of such experimental work is to verify the leaning abilities of the adopted five layers learning system, even for various navigation scenarios.

**REFERENCES**