Evaluation of grid-map sensor fusion mapping algorithms

Conference Paper · November 2007

DOI: 10.1109/ICSMC.2007.4413893 · Source: IEEE Xplore

IADIS International Journal on Computer Science and Information Systems
Vol. 2, No. 2, pp. 13-32
ISSN: 1646-3692

EVALUATION OF GRID-MAP SENSOR FUSION MAPPING ALGORITHMS

Keren Kapach and Yael Edan *Department of Industrial Engineering and Management*
Ben-Gurion University of the Negev, P.O.Box 653, Beer Sheva 84105, ISRAEL

kapach, yael @bgu.ac.il

ABSTRACT

This paper presents a thorough evaluation of grid map based sensor fusion algorithms for mapping the environment of a mobile robot. Three physical sensors were used for creating the grid maps: a CCD camera, a set of ultrasonic sensors and a laser rangefinder. An adaptive fuzzy logic algorithm was compared to three logical sensor fusion algorithms. Results indicate the superiority of the adaptive fuzzy logic algorithm with a 0.075 confidence level.

KEYWORDS

sensor fusion, algorithms, robotics and autonomous mobile robots.

1. INTRODUCTION

The basic task of a mobile robot is to explore an unknown environment in order to perform complex tasks, *e.g.*, navigation [Daniel *et al.,* 2005], lawn mowing [2][Arkin *et al.,* 2000], target tracking [Zhen *et al.,* 2005], motion planning and execution [Luo and Kay, 1989]. Performing these tasks requires building accurate maps that describe the robot's surroundings [24][Stepan *et al.,* 2005].

The first step in building a map is to choose the appropriate representation model. There are several different techniques for representing the environment of a mobile robot, including configuration space [Lozano-Perez, 1981]; generalized cones [4][Brooks, 1982]; spherical octree [Chen, 1987]; and occupancy grid maps [Stepan, 2005]. In this research the grid map paradigm was used since it is a simple and fast technique, In the grid map model the environment is divided into discrete cells, each containing a binary value that indicates whether the area represented by the cell is *Occupy* ('1') or *Empty* ('0').

An autonomous mobile robot must be equipped with several sensors in order to sense its surroundings. To complete the mapping mission, it is necessary to choose a method for handling the multitude of sensors. In multi-sensor systems, the logical sensor paradigm is commonly used [19][Moravec and Elfes, 1985]. A logical sensor is an abstract definition of a sensor that can be used to provide a uniform framework for multisensory integration [Moravec and Elfes, 1985]. This approach enables to add sensors to the system without changing its whole concept. In this work several logical sensors were implemented.

The next step towards an accurate environment mapping is to choose the appropriate fusion level and the desired sensor fusion algorithm. To enhance the accuracy of the maps the environmental information received from multiple sensors must be merged. Sensor fusion deals with synergetic merging of information from several different physical sensors [Adibi and Gonzales, 1992]. The fusion of the data or information from multiple sensors can take place at different levels of representation [Adibi and Gonzales, 1992] using different fusion algorithms. *Signal level* fusion refers to the combination of the signals of sensors to a signal that is usually of the same form as the original signal but of greater quality. *Pixel level* fusion can be used to increase the information content associated with each pixel in an image formed through a combination of multiple images, *e.g.,* the fusion of a range image with a two-dimensional intensity image adds depth information to each pixel in the intensity image. This can be useful in the subsequent processing of the image. *Feature level* fusion can be used to both increase the likelihood that a feature extracted from the information provided by a sensor actually corresponds to an important aspect of the environment, and as a means of creating additional composite features for the system to use. *Symbol level* fusion allows the information from multiple sensors to be effectively used together at the highest level of abstraction. Symbol level fusion may be the only means by which sensory information can be fused if the sensors are very dissimilar or refer to different regions of the environment.

Several fusion systems involve the use of feedback [Srinivasan, 1986]. In these systems, the fusion algorithm feeds the decision to each of the logical sensors, but the sensors do not consider their previous values. In fusion systems with memory, the logical sensors consider their previous values [6][Cohen, 2005].

Several sensor fusion algorithms have been previously developed [18][Luo *et al.,* 2001; Najjar and Bonnifait, 2005; Smith and Cheeseman, 1986]. The adaptive fuzzy logic (AFL [Cohen and Edan, 2004]) uses symbol level fusions and is considered as an algorithm that has memory and feedback. The AFL uses on-line logical sensory performance measures to determine the quality of the fused map.

This paper presents a thorough evaluation of a system that uses an adaptive fuzzy algorithm developed by [Cohen and Edan, 2004] and compares it to logical fusion algorithms. To choose the best performing algorithms, a previously developed statistical evaluation method was used [Cohen *et al.,* 2005]. The method is based on performance measures that measure the difference between the fused map and the environment truth map. The method first checks if the statistical difference between algorithms; then isolates the two best performing algorithms and finally picks the best performing algorithm.

2. SENSOR FUSION SYSTEM

2.1 General

The sensor fusion system is an expansion of previous work [Cohen, 2005] and includes three physical sensors: ultrasonic sensors, a CCD camera and a Laser rangefinder. The sensor fusion system uses three basic concepts: *logical sensors*, a binary *grid map* and *performance measures*. The *logical sensor* paradigm used to provide a uniform framework for multisensory integration [Henderson and Shilcrat, 1984[11]]. This approach enables to add sensors to the system without changing its whole concept. In this work, two additional logical sensors were added to the system easily due to the use of logical sensors. The *grid map* paradigm was chosen to present the environment perception due to its simple implementation and use [Moravec and Elfes, 1985][19]. Using the grid map representation, the environment is divided into a fixed size discrete grid. Each grid cell is assigned a binary value that indicates if that location is occupied by an obstacle or not. A value '0' represents an 'Empty' Cell, and a value '1' represents an 'Occupy' cell. Each logical sensor represents the environment using a unique *grid map*, and the different grid maps from all the logical sensors were fused into one map using fusion algorithms. The *performance measures* quantify the difference between two grid maps [Cohen, 2005]. The logical sensor performance measures were used in the fusion process and the sensor fusion performance measures were used in the algorithms' evaluation process.

2.2 Information Flow

The system includes N logical sensors representing k physical sensors. The logical sensors work asynchronously. The schematic description of the information flow is presented in Figure 1. At each time step t, the i[th] logical sensor maps the environment using the physical sensor readings and creates a local observation grid map (LOGM), denoted by y_i^t $i = 1, 2, \cdots, N$. Let c_i and d_i be the local observation grid map dimensions. The map $y_i^t \in [0,1]^{c_i \times d_i}$ contains binary values each cell of that map. The values indicate whether the cell is '*Occupy*' or '*Empty*'.

The system transfers each sensor's LOGM into a local binary grid map (LGM), denoted by $u_i^t, i = 1,2,\cdots, N$. Let c and d be the local grid map dimension. The map $u_i^t \in [0,1]^{c \times d}$ contains binary values for each cell of that map. The LGM dimensions are identical for all logical sensors.

There are two types of sensor algorithms: **logical** and **adaptive**. The algorithms differ in the memory and feedback properties; the adaptive algorithms use performance measures in the fusion process while the logical algorithms do not.

For **logical** algorithms (Figure 1):

The LGM reaches the fusion center, where it yields the fused grid map (FGM) $u_0^t \in [0,1]^{c\times d}$, based on all the LGM $u^t, u^t = (u_1^t, u_2^t, \cdots, u_N^t)$, using the fusion rule $f(\cdot)$ as follows [Cohen, 2005]:

$$u_0^t = f(\boldsymbol{u^t}) \tag{1}$$

For **Adaptive** algorithms (Figure 1):

The performance measures of the logical sensors are calculated based on the previous local grid maps $u^{t-1}, u^{t-1} = (u_1^{t-1}, u_2^{t-1}, \cdots, u_N^{t-1})$, of the logical sensors and the previous fused grid map defined as u_0^{t-1}. The performance measures are denoted as p_i^{t-1}, where $i = 1,2,\cdots,N$. The calculation of the performance measure depends on the fusion rule in the fusion center. A detailed description on the performance measures calculation process can be found in [Kapach, 2007]. An average value of the logical sensor performance measures $p^{t-1,t-2}, \boldsymbol{p^{t-1,t-2}} = (p_1^{t-1,t-2}, p_2^{t-1,t-2}, ..., p_N^{t-1,t-2})$ is calculated based on $\boldsymbol{p^{t-1}}$ and $\boldsymbol{p^{t-2}}$, where $p_i^{t-1,t-2} = \dfrac{p_i^{t-1} + p_i^{t-2}}{2}$, $i = 1,2,...,N$ [Cohen, 2005].

Both the local grid maps u^t and an average value of the logical sensor performance measures $p^{t-1,t-2}$ are transmitted to the fusion center. At the fusion center, based on all local grid maps u^t and the average value of the logical sensor performance measures $p^{t-1,t-2}$ the sensor fusion algorithm yields the fused grid map u_0^t at time step *t*, using the decision rule $f(.)$ as follows:

$$u_0^t = f(u^t, p^{t-1,t-2}) \tag{2}$$

The fused grid map, u_0^t is fed back to all logical sensors (u^t) to calculate the new performance measures (p^t) [Cohen, 2005].

The following information flow is identical to both logical and adaptive algorithms.

At each time step *t*, a virtual global grid map (VGGM), denoted by $Z_0^t, Z_0^t \in [0,1]^{a\times b}$, expands the size of the fused grid map u_0^t from $c \times d$ to $a \times b$, which is the full size. This is done by assigning zero values to all cells of the virtual global grid map Z_0^t, except those which appear in u_0^t (their values are as in the u_0^t map).

All the VGGM's are placed in a new map, the global grid map (GGM), denoted by Z, $Z \in [0,1]^{a\times b}$. The VGGM's are places in the GGM according to the robot's new position along the path. The GGM is a $a \times b$ matrix and is the output of the entire mapping process that represents the whole environment mapping along the robot's path.

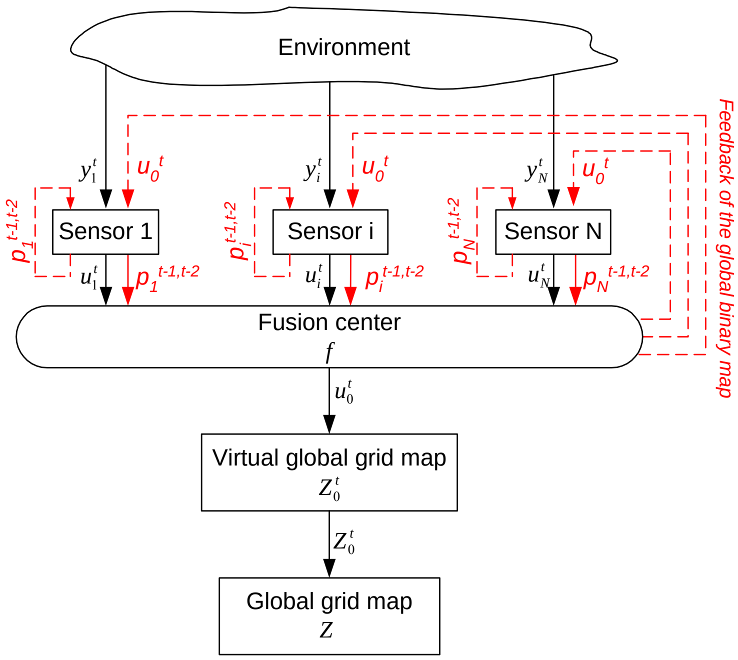

Figure 1. Information flow at time t – logical and adaptive algorithms
(Adapted from Cohen, 2005)
Data in black represents the logical algorithms information flow
Data in red with the data in black represent the adaptive algorithms information flow

2.3 Performance Measures

These performance measures use the binary decisions about the cell's condition in the grid maps. Since the cell's condition is a binary value (a positive value indicates *'Occupy'* and '0' indicates *'Empty'*), there are four logical conditions for the difference between the two maps.

the performance measures are defined as the summation over all cells of the four logical conditions: *Occupy – Occupy*, *Empty – Empty*, *Occupy – Empty* and *Empty – Occupy*.

Performance measures are used in two cases. In each case the calculation process is slightly different. In the first case, they are defined as 'logical sensor performance measures' and are used in the AFL algorithm, to quantify the difference between the logical sensor's maps and the fused map received as an output from the fusion algorithm. In the second case, they are defined as 'sensor fusion algorithm performance measures' and used in the sensor fusion evaluation process, to quantify the difference between the sensor fusion's map and the original truth map.

2.3.1 Logical Sensor Performance Measures

Four performance measures were defined to quantify the difference between each logical sensor's map and the fused map. The values of each of the 4 pixels in both maps can be either *'True=1'* or *'False=0'*. Therefore, four possible options exist when conducting a comparison of an indexed pixel in the maps, resulting in four performance measures [Cohen, 2005]: *OO* – Number of '1' in both maps divided by the number of '1' in the fused map, *EE* – Number of '0' in both maps divided by the number of '0' in the fused map, *OE* – Number of '0' in the logical sensor's map but '1' in the fused map divided by the number of '0' in the fused map, *EO* – Number of '1' in the logical sensor's map but '0' in the fused map divided by the number of '1' in the fused map. Ideally, *OO* and *EE* should be 1 and *OE* and *EO* should by 0. The following definitions are adapted from [Cohen, 2005]:

Let:

$$
\begin{aligned}
LGM_{jk} &= \begin{cases} 0 & Empty\ jk\ cell\ in\ LGM \\ >1 & Occupied\ jk\ cell\ in\ LGM \end{cases} \\
FGM_{jk} &= \begin{cases} 0 & Empty\ jk\ cell\ in\ FGM \\ >1 & Occupied\ jk\ cell\ in\ FGM \end{cases}
\end{aligned}
\qquad [3]
$$

Where:

$LGM^t(i)_{jk}$ are cells in the sensor's local grid map (u_i^t) and

FGM^t_{jk} are cells in the fused grid map (u_0^t) corresponding to the j row and k column

Then:

$$OO^{t}(i)=\begin{cases}\dfrac{\sum_j\sum_k\left(LGM^{t}(i)_{jk}\cdot FGM^{t}_{jk}\right)}{\sum_j\sum_k FGM^{t}_{jk}} & \sum_j\sum_k FGM^{t}_{jk}>0\\ E_{LGM}\,E^{t}_{FGM}(i) & else\end{cases}\qquad[4]$$

$$EE^{t}(i)=\begin{cases}\dfrac{\sum_j\sum_k\left(\left(1-LGM^{t}(i)_{jk}\right)\cdot\left(1-FGM^{t}_{jk}\right)\right)}{\sum_j\sum_k\left(1-FGM^{t}_{jk}\right)} & \sum_j\sum_k\left(1-FGM^{t}_{jk}\right)>0\\ O_{LGM}\,O^{t}_{FGM}(i) & else\end{cases}\qquad[5]$$

$$OE^{t}(i)=\begin{cases}\dfrac{\sum_j\sum_k\left(LGM^{t}(i)_{jk}\cdot\left(1-FGM^{t}_{jk}\right)\right)}{\sum_j\sum_k\left(1-FGM^{t}_{jk}\right)} & \sum_j\sum_k\left(1-FGM^{t}_{jk}\right)>0\\ 1-O_{LGM}\,O^{t}_{FGM}(i) & else\end{cases}\qquad[6]$$

$$EO^{t}(i)=\begin{cases}\dfrac{\sum_j\sum_k\left(\left(1-LGM^{t}(i)_{jk}\right)\cdot FGM^{t}_{jk}\right)}{\sum_j\sum_k FGM^{t}_{jk}} & \sum_j\sum_k FGM^{t}_{jk}>0\\ 1-E_{LGM}\,E^{t}_{FGM}(i) & else\end{cases}\qquad[7]$$

2.3.2 Sensor Fusion Performance Measures

The sensor fusion performance measures are calculated by comparing each cell of the original truth map ($ORG, ORG \in [0,1]^{a\times b}$), with the corresponding cell on the global grid map (*GGM*) which is defined as Z in the information flow. The performance measures use the binary decisions about the cell's condition in the grid maps. The values of the sensor performance measures OO, EE, OE and EO were calculated by multiplying the relevant variables by the coefficients as detailed in Table 1.

Table 1. Coefficients for calculating the sensor fusion performance measures

Occupy

$\mathbf{Occupy_{Coefficient}}$

$$\begin{aligned}
&If\ \left(Occupy_{ORG} = Occupy_{GGM} = 0\right)\ Then\\
&\quad Occupy_{Coefficient} = Empty_{Coefficient}\\
&elseif\quad 0 \le \frac{Occupy_{GGM}}{Occupy_{ORG}} \le 1\ Then\\
&\quad Occupy_{Coefficient} = \frac{Occupy_{GGM}}{Occupy_{ORG}}\\
&else\\
&\quad Occupy_{Coefficient} = \frac{Occupy_{ORG}}{Occupy_{GGM}}
\end{aligned}$$

$\mathbf{Empty_{Coefficient}}$

$$\begin{aligned}
&If\ \left(Occupy_{ORG} = Occupy_{GGM} = a\cdot b\right)\ Then\\
&\quad Empty_{Coefficient} = Occupy_{Coefficient}\\
&elseif\quad 0 \le \frac{a\cdot b - Occupy_{GGM}}{a\cdot b - Occupy_{ORG}} \le 1\ Then\\
&\quad Empty_{Coefficient} = \frac{a\cdot b - Occupy_{GGM}}{a\cdot b - Occupy_{ORG}}\\
&else\\
&\quad Empty_{Coefficient} = \frac{a\cdot b - Occupy_{ORG}}{a\cdot b - Occupy_{GGM}}
\end{aligned}$$

Where a and b are defined as the global grid map's dimensions.

Let:

$$GGM_{jk} = \begin{cases} 0 & Empty\ jk\ cell\ in\ GGM \\ >0 & Occupied\ jk\ cell\ in\ GGM \end{cases}$$
$$ORG_{jk} = \begin{cases} 0 & Empty\ jk\ cell\ in\ ORG \\ >0 & Occupied\ jk\ cell\ in\ ORG \end{cases} \quad [8]$$

Where:
GGM_{jk} are cells in the global grid map (Z) and
ORG_{jk} are cells in the original map (ORG) corresponding to the j row and k column

Then:

$$O_{GGM}O_{ORG} = \begin{cases} \dfrac{\sum_j \sum_k \left(GGM_{jk} \cdot ORG_{jk}\right)}{\sum_j \sum_k ORG_{jk}} & \sum_j \sum_k ORG_{jk} > 0 \\ E_{GGM}E_{ORG} & else \end{cases} \quad [9]$$

$$E_{GGM}E_{ORG} = \begin{cases} \dfrac{\sum_j \sum_k \left(\left(1-GGM_{jk}\right)\cdot\left(1-ORG_{jk}\right)\right)}{\sum_j \sum_k \left(1-ORG_{jk}\right)} & \sum_j \sum_k \left(1-ORG_{jk}\right) > 0 \\ O_{GGM}O_{ORG} & else \end{cases} \quad [10]$$

$$O_{GGM}E_{ORG} = \begin{cases} \dfrac{\sum_j \sum_k \left(GGM_{jk} \cdot \left(1-ORG_{jk}\right)\right)}{\sum_j \sum_k \left(1-ORG_{jk}\right)} & \sum_j \sum_k \left(1-ORG_{jk}\right) > 0 \\ 1-O_{GGM}O_{ORG} & else \end{cases} \quad [11]$$

$$E_{GGM}O_{ORG} = \begin{cases} \dfrac{\sum_j \sum_k \left(\left(1-GGM_{jk}\right)\cdot ORG_{jk}\right)}{\sum_j \sum_k ORG_{jk}} & \sum_j \sum_k ORG_{jk} > 0 \\ 1-E_{GGM}E_{ORG} & else \end{cases} \quad [12]$$

And

$$OO = Occupy_{Coefficient} \cdot \left[O_{GGM} \; O_{ORG} \right] \qquad [13]$$

$$EE = Empty_{Coefficient} \cdot \left[E_{GGM} \; E_{ORG} \right] \qquad [14]$$

$$OE = \left(1 - Empty_{Coefficient} \right) \cdot \left[O_{GGM} \; E_{ORG} \right] \qquad [15]$$

$$EO = \left(1 - Occupy_{Coefficient} \right) \cdot \left[E_{GGM} \; O_{ORG} \right] \qquad [16]$$

2.4 Sensor Fusion Algorithms

Three logical algorithms were compared to an adaptive algorithm. In the logical algorithms the logical sensor distinguishes between two basic states, *Occupy* and *Empty*. These algorithms present different versions of *Identify the obstacle by at least n logical sensors*: Logical OR (n=1), MOST (n>N/2) and logical AND (n=N), where *N* is the total number of logical sensors in the system [Cohen, 2005; Blum *et al.*, 1997; Klein, 1993].

In the adaptive algorithm, each time step *t*, the ith logical sensor creates its local grid map ($i.e., u_i^t$). The fused map ($i.e., u_0^t$) is built using the average value of the performance measures [Cohen, 2005] which are recalculated online. An adaptive fuzzy logic algorithm was selected due to its superior performance as indicated in previous analyses [Cohen, 2005] The AFL algorithm receives on-line the four performance measures of each logical sensor and each logical sensor local grid map as inputs and calculates the binary fused map using fuzzy logic. The AFL algorithm that was evaluated was the algorithm that achieved best performances according to.

The AFL algorithm uses logical sensors performance measures as fuzzy variables with three fuzzy sets: High, Average and Low. Each fuzzy set member is associated with a trapezoid membership function. The membership function evaluates the degree of membership of each variable value of the respective fuzzy set member [Cohen, 2005]. The fuzzy sets values and membership function of the fuzzy variables of the best performing algorithm according to Cohen's evaluation (denoted as 1010 in [Cohen, 2005]) are presented in Table 2.

Table 2. Fuzzy set values of the fuzzy variables [Cohen, 2005]

Fuzzy variable	**Fuzzy sets**		
	Low	**Avg.**	**High**
$O_{LGM}O_{FGM}{}^{t-1,t-2}(i)$	0,0,0.3,0.45	0.4,0.45,0.55,0.6	0.55,0.7,1, 1
$E_{LGM}E_{FGM}{}^{t-1,t-2}(i)$	0,0,0.3,0.45	0.4,0.45,0.55,0.6	0.55,0.7,1, 1
$O_{LGM}E_{FGM}{}^{t-1,t-2}(i)$	0,0,0.3,0.45	0.4,0.45,0.55,0.6	0.55,0.7,1, 1
$E_{LGM}O_{FGM}{}^{t-1,t-2}(i)$	0,0,0.3,0.45	0.4,0.45,0.55,0.6	0.55,0.7,1, 1

For each logical sensor at every time stamp *t*, two fuzzy output variables are calculated: $Occupy_i^t$ and $Empty_i^t$, where $i = 1,2,\cdots,N$ and N is the total number of the logical sensors. These output fuzzy variables also have three fuzzy sets: High, Average and Low. Each fuzzy set member is associated with a trapezoid membership function. The fuzzy sets values of the output fuzzy variables are presented in Table 3.

Table 3. Fuzzy sets values of the fuzzy output variables [Cohen, 2005]

Fuzzy variable	**Fuzzy sets**		
	Low	**Avg.**	**High**
Occupy	0,0,0.3,0.45	0.4,0.45,0.55,0.6	0.55,0.7,1,1
Empty	0,0,0.3,0.45	0.4,0.45,0.55,0.6	0.55,0.7,1,1

The fuzzy output variables are calculated using twelve **If-Then** rules presented in Table 4.

Table 4. If-Then rules [Cohen, 2005]

	Fuzzy variables input				**Fuzzy variables output**	
Rule	$O_{LGM}O_{FGM}^{t-1,t-2}(i)$	$E_{LGM}E_{FGM}^{t-1,t-2}(i)$	$O_{LGM}E_{FGM}^{t-1,t-2}(i)$	$E_{LGM}O_{FGM}^{t-1,t-2}(i)$	Occupy	Empty
1	High				High	
2	Avg.				Avg.	
3	Low				Low	
4			High		Low	
5			Avg.		Avg.	
6			Low		High	
7		High				High
8		Avg.				Avg.
9		Low				Low
10				High		Low
11				Avg.		Avg.
12				Low		High

The rules are defuzzified using the Mamdani inference with centroid method [Mamdani and Assilian, 1975] and are evaluated to determine the final value of the $Occupy_i^t$ and $Empty_i^t$ final value [Cohen, 2005].

The fused map cells are binary, where '1' indicates that the cell is 'Occupy' and '0' indicated that the cell is 'Empty'. The decision rule for the fused map cells is based on the summation of the logical sensor's $Occupy_i^t$ and $Empty_i^t$ final values. For all the corresponding cells in the logical sensor's map that are '0', their values are summed. For all the corresponding cells in the logical sensor's map that are '0', their $Occupy_i^t$ values are summed. If the *Occupy* sum is greater than the *Empty* sum, the cell in the fused map is set to '1', otherwise – '0'. The pseudo-code for fused map decision rule is presented in Table 5:

Table 5. Pseudo code for fused map decision rule

Adaptive fuzzy logic decision rule
$for\ x = 1 : MapSizeX$ $\quad for\ y = 1 : MapSizeY$ $\quad\quad for\ i = 1 : N$ $\quad\quad\quad if\ u_i^t(x, y) = 0$ $\quad\quad\quad\quad Empty^t = Empty^t + Empty_i^t$ $\quad\quad\quad else$ $\quad\quad\quad\quad Occupy^t = Occupy^t + Occupy_i^t\}$ $\quad\quad if\ Occupy^t > Empty^t$ $\quad\quad\quad u_0^t(x, y) = 1$ $\quad\quad else$ $\quad\quad\quad u_0^t(x, y) = 0$

3. EXPERIMENTS

3.1 Experimental Setup

Figure 2. Pioneer 2-AT

The experiment consisted of a mobile robot (Active Media Pioneer 2-AT) equipped with two sets of eight ultrasonic sensors in front and on the back, a SICK laser scanner mounted on top of the robot and a SONY CCD camera mounted on the laser scanner. Only the six front ultrasonic sensors were used, so all the sensors scan the area in front of the robot.

The experiment took place in a controlled laboratory environment where a special setup was constructed (Figure 2). The experiment setup consisted of a 5m long black path (width 2.5m). Five similar red cylindrical obstacles were placed along the path (Ø25 cm, 50 cm height). The logical sensors created different maps, because not all obstacles were always noticeable due to different properties of the physical sensors. These differences caused the logical sensors to disagree. To increase disagreement between logical sensors two types of decoys were set alongside the robot's path [Cohen, 2005].

The decoys were made of light brown rug. The first type of decoy was less than 6 cm. in width and length; the size of the second type was around 30 cm. (Figure 3). The decoys location was randomly change between repetitions.

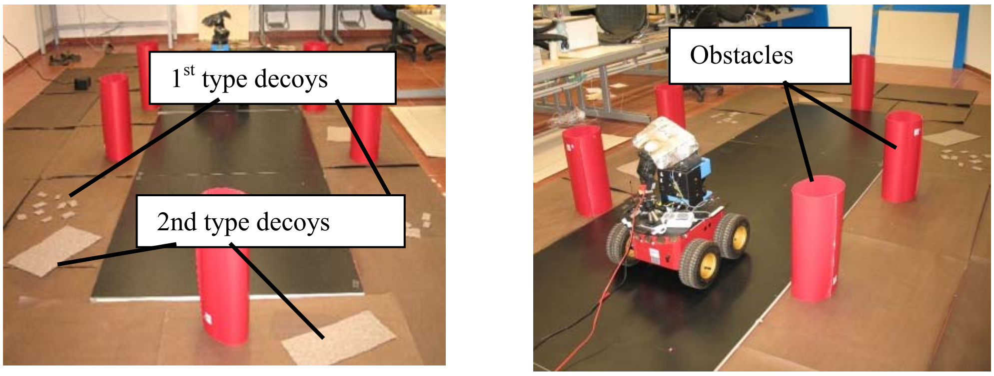

Figure 3. Experimental setup photographs

3.2 Logical Sensors Implementation

Total of seven logical sensors were used in the experiment. Two logical sensors were generated using the ultrasonic data denoted as US1 and US2 using the logical OR algorithm and Probabilistic-Approach algorithm respectively [Ribo and Pinz, 2001]. Two logical sensors were generated using the laser data denoted as LASER1 and LASER2. The first one used the entire 180° laser scanning range, while the second one used only every third angle reading. LASER1 and LASER2 were implemented using geometrical transformations from the data received from the laser sensor and the robot's location received from the robots encoders. Three logical sensors were generated from the image grabbed by the camera denoted as CAM1, CAM2 and CAM3. Each of the logical sensors aimed to identify different types of objects [Cohen, 2005]. CAM1 aimed to identify the obstacles, CAM2 to identify the obstacles and the first type of decoys and CAM3 to identify the obstacles and the second type of decoys. The camera's logical sensors were implemented using image processing algorithms [Kapach, 2007]. However, the algorithms were not optimized and their performances highly depended on the lighting conditions, which varied along the path due to external conditions (*e.g.,* shadows from ceilings and from obstacles in the room). Figure 4 presents the mapping results from the seven logical sensors in the controlled laboratory conditions with the real world map as the reference. Mapping results from all experiments and repetitions can be found in [Kapach, 2007].

Real world map	US1	US2	LASER1	LASER2	CAM1	CAM2	CAM3

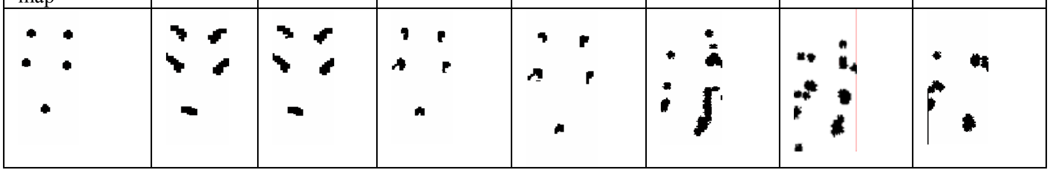

Figure 4. An example of logical sensors mapping and the real world map

3.3 Experimental Procedure

During the experiment, the robot moved forward at constant velocity (0.1 m/sec) and scanned the area in from of it using three physical sensors. The robot traveled 400 ± 5 cm and performed 38 scanning cycles. To eliminate influence of the robot localization problem [Lin *et al.*, 2003] the robot moved only forward. To ensure that the robot traveled straight, the robot was placed at the beginning of the path and a laser pointer mounted on top of the robot marked the starting point on a calibration board placed at the end of the path. The robot's exact location was changed until the point on the calibration board matched the exact beginning point. At the end of the experiment the robot's location was measured again using the laser and the calibration board and if the robot diverged more than 4cm the repetition was not considered in the analysis. To enhance image processing performance, the only light source was a 300W spot placed behind the camera and a sheet of aluminum foil was placed in the back of the spot to prevent light reflection. The experimental software was written using a Visual C++ compiler and a dedicated API (ARIA 2.4 from Active Media [Kapach, 2007]).

3.4 Experimental Design

Seven different experiments were conducted (Table 1). The experiments differed in the environmental and logical sensors conditions. Malfunctions were created artificially by setting logical sensors to empty, full and shifting positions by a constant value. Lighting conditions were changed in the third and seventh experiment. Each experiment was repeated seven times. Experiment's repetitions were performed under the same conditions with natural variations such as lightning conditions, shadows and time differences. The motivation for choosing the different conditions for experiments is explained in part III. In each experiment, environment mapping was achieved using the four different sensor fusion algorithms, resulting in a total of 196 environmental mappings (4 sensor fusion algorithms X 7 Experiments X 7 repetitions). Figure 5 presents the algorithms' map results from experiment 1, first repetition and the corresponding real world map. Four sensor fusion performance measures were calculated from every experiment and for each fusion algorithm map (using equations [13]-[16]).

Table 6. Experimental design for statistical evaluation experiment

<table>
<tr><th>Exp.</th><th>US1</th><th>US2</th><th>LASER1</th><th>LASER2</th><th>CAM1</th><th>CAM2</th><th>CAM3</th><th>Comments</th></tr>
<tr><td>1</td><td>Empty</td><td rowspan="2">Regular Algorithm</td><td>Full</td><td>Regular Algorithm</td><td>Regular Algorithm</td><td>Shift:
X=X+40cm
Y=Y+40cm</td><td>Shift:
X=X-40cm
Y=Y-40cm</td><td></td></tr>
<tr><td>2</td><td>Full</td><td>Empty</td><td>Regular Algorithm</td><td>Regular Algorithm</td><td>Shift:
X=X-40cm
Y=Y-40cm</td><td>Empty</td><td></td></tr>
<tr><td>3</td><td rowspan="5">Regular Algorithm</td><td>Empty</td><td rowspan="5">Regular Algorithm</td><td>Full</td><td>Regular Algorithm</td><td>Regular Algorithm</td><td>Regular Algorithm</td><td>Lights off for cycles 15-end</td></tr>
<tr><td>4</td><td>Full</td><td>Empty</td><td>Regular Algorithm</td><td>Regular Algorithm</td><td>Full</td><td></td></tr>
<tr><td>5</td><td>Full</td><td>Empty</td><td>Shift:
X=X+20cm
Y=Y-40cm</td><td>Full</td><td>Shift:
X=X-40cm
Y=Y+60cm</td><td></td></tr>
<tr><td>6</td><td>Empty</td><td>Full</td><td>Regular Algorithm</td><td>Shift:
X=X+60cm
Y=Y+60cm</td><td>Full</td><td></td></tr>
<tr><td>7</td><td>Regular Algorithm</td><td>Regular Algorithm</td><td>Regular Algorithm</td><td>Regular Algorithm</td><td>Regular Algorithm</td><td>Lights off for cycles 15-end</td></tr>
</table>

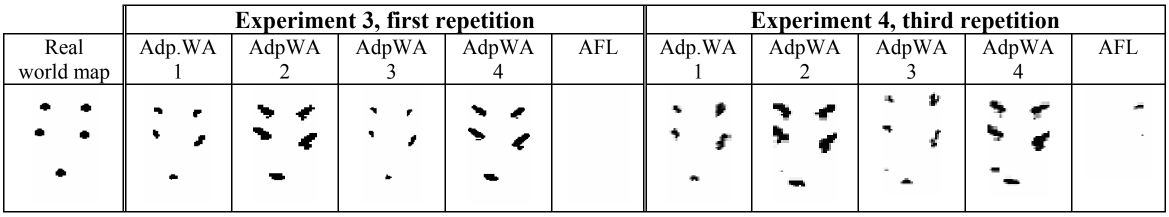

Figure 5. Map results from different experiments

4. ANALYSIS AND RESULTS

The evaluation method [Cohen *et al.,* 2005] aimed to: 1) ensure that the experiments are different enough in order to guarantee that the results regarding the best algorithm are not specific for the specific dataset, 2) confirm that enough repetitions were done in each experiment and 3) select the best performing algorithm.

4.1 Experimental Design

To ensure the results are not specific for the specific dataset the following steps were conducted to ensure that experiments are different and that there are a sufficient number of different repetitions [Cohen *et al.,* 2005].

4.1.1 Different Experiments

For each experiment between every two repetitions, each logical sensor's map from one repetition is subtracted from all other logical sensor maps from the other repetitions and saved as an absolute value [Cohen *et al.,* 2005]. For example, LS1 map from experiment 1, repetition 1 is compared to LS1 map from experiment 2 and all its repetitions, experiment 3 and all its repetitions and so on. The number of cells different than '0' (signed cells) is saved for each comparison. For 7 LS, 7 repetitions and 7 experiments, this results in 7,203 subtracted maps. For each comparison, the worst difference of all logical sensors is saved [Cohen *et al.,* 2005].

4.1.2 Different Repetitions

For each experiment, each logical sensor's map is subtracted from all its repetitions in pairs and saved as an absolute value [Cohen *et al.,* 2005]. For example, LS1 map from the experiment 1 is subtracted from LS1 maps from all other repetitions. This is conducted for all LS, and the number of cells different than '0' (signed cells) is saved for each comparison. For 7 LS, 7 repetitions and 7 experiments, this results in 1,029 comparisons. For each comparison, the worst difference is saved, *e.g.*, the maximum number of signed cells [Cohen *et al.,* 2005].

4.1.3 Volume of Overlap Region

This measure is an indicator that the experiments and repetitions are indeed different. This measure evaluates the overlap of two populations (*e.g.*, experiments and repetitions) and should be as negative as possible [Tin and Mitra, 2002]. If the volume is not negative, the experiments are not different enough and more experiments need to be performed. The volume is calculated using the minimum and maximum number of signed cells from all the comparisons between the experiments and repetitions. The volume is negative as equation [17] shows, implying that the experiments are different and repetitions are similar:

$$VOLR = \frac{MIN(5136, 756) - MAX(877, 78)}{MAX(5136, 756) - MIN(877, 78)} = -0.0243 \qquad [17]$$

4.1.4 Number of repetitions

The number of repetitions is based on a t-test detailed in [Cohen *et al.,* 2005] and is calculated for α=0.05 and β=0.2. For each performance measure, the number of repetitions was calculated, and the final number was taken as the maximum number from all the performance measures. S (standard deviation) for each performance measure was taken as the upper bound of the standard deviation for this performance measure from all the experiments. Δ is chosen to be 20% from the average upper bound and was taken for each performance measure separately too. Based on the results (Table 7), the largest R is for the OO measure; this results in six necessary repetitions. Since each experiment has already seven repetitions, no additional repetitions were required.

Table 7. R calculations for each performance measure

	S	Δ	R
OO	0.0371	0.041	**6**
EE	0.0221	0.1459	1
OE	0.0011	0.1471	1
EO	0.0625	0.15	2

4.2 Statistical Analysis

To find the best performing algorithm a statistical analysis that includes three non-parametric tests was conducted [Cohen *et al.,* 2005]. The first test is the *Friedman's test* [Hollander and Wolfe, 1973], that checks whether the algorithms performances are considered different. Friedman's test is performed separately for each performance measure in every experiment. In this test, the algorithms are ranked from the least (rank=1) to the largest (rank=4) for every repetition. The test statistic uses the rank differences. The second step is the *multiple comparison's procedure* [Hollander d Wolfe, 1973] that picks the best performing couple of algorithms. The multiple comparisons' procedure uses the sum of ranks for each algorithm to divide the algorithms into homogenous subgroups. Two algorithms belong to the same subgroup if the difference between the sums of their ranks does not exceed a predefined critical value. The critical value is taken from table A.17 in [Hollander and Wolfe, 1973]. The significant value for this test is derived from the number of repetitions and the number of the compared algorithms, and appears in the same table. The third and final step is the *sign test* [Hollander and Wolfe, 1973] that picks the best performing algorithm. The sign test checks the significance of difference between the medians of the two algorithms. If the p-value of this test is smaller than the desired significance level, this proves that one algorithm is superior to the other.

4.2.1 Friedman's test

Friedman's test [Hollander and Wolfe, 1973] was performed to check if there is a significant difference between the algorithms. The null hypothesis is that the algorithms perform similar in terms of the median, and there is no difference between them. Friedman's test was performed for each performance measure for every experiment separately. In this test, the algorithms are ranked from the least (rank=1) to the largest (rank=4) for every repetition. The test statistic uses the rank differences. P-values for all 7 experiments for all seven experiments are presented in Table 8. The very small p-values imply a difference between algorithms.

Table 8. Friedman's test results

Exp.	**Sensor fusion performance measures**	**p - value**
1.	OO	0.0002
	EE	0.0001
	OE	0.0003
	EO	0.0002
2.	OO	0.0001
	EE	0.0001
	OE	0.0001
	EO	0.0002
3.	OO	0.0004
	EE	0.0001
	OE	0.0002
	EO	0.0005
4.	OO	0.0003
	EE	0.0006
	OE	0.0005
	EO	0.0005
5.	OO	0.0002
	EE	0.0001
	OE	0.0003
	EO	0.0004
6.	OO	0.0005
	EE	0.0001
	OE	0.0002
	EO	0.0002
7.	OO	0.0001
	EE	0.0003
	OE	0.0004
	EO	0.0001

4.2.2 Multiple-comparison procedure

Friedman's *multiple-comparison procedure* [Hollander and Wolfe, 1973] uses the sum of ranks for each algorithm to divide the algorithms into homogenous subgroups. Two algorithms belong to the same subgroup if the difference between the sums of their ranks does not exceed a predefined critical value. The critical value is taken from table A.17 in [Hollander and Wolfe, 1973]. The significance value for this test is derived from the number of repetitions and the number of the compared algorithms, and appears in the same table. For four algorithms and seven repetitions, algorithms that have any difference smaller than 14 between their sums of ranks belongs to the same subgroup. A close look at the results indicates that in most cases MOST and AFL algorithms belong to the same best subgroup and thus they are considered the two best performing algorithms.

4.2.3 Sign test

The last step is to choose the best performing algorithm between MOST and AFL. This is done using the *sign test* [Hollander and Wolfe, 1973]. The sign test checks the significance of difference between the medians of the two algorithms. If the p-value of this test is smaller than the desired significance level, this proves that one algorithm is superior to the other. AFL was better than MOST in 12 out of 28, in 4 cases MOST outperformed AFL and in 12 cases there was no difference between the algorithms (Table 9). The p-value corresponding 12 out of 28 cases is 0.075, implying that the AFL algorithm is the best, as presented in Table 10, which was generated using SPSS software for windows release 12.0.0. This evaluation corresponds to a qualitative evaluation achieved from using visual representation of the generated maps for the different sensor fusion algorithms.

Table 9. Sign test data

Experiment environmental conditions	**Sensor fusion performance measures**							
	OO		EE		OE		EO	
	MOST	**AFL**	**MOST**	**AFL**	**MOST**	**AFL**	**MOST**	**AFL**
1.	Ties	Ties	Ties	Ties	Ties	Ties	Ties	Ties
2.	0	7	0	7	0	3	0	7
3.	0	7	0	7	3	4	0	7
4.	Ties	Ties	Ties	Ties	Ties	Ties	Ties	Ties
5.	7	0	7	0	7	0	7	0
6.	Ties	Ties	Ties	Ties	Ties	Ties	Ties	Ties
7.	0	7	1	6	1	6	1	6
Total	**1**	**3**	**1**	3	**1**	**3**	**1**	**3**
Note: The values in this table indicate the number of times each algorithm outperforms the opponent.								

Table 10. Sign test results

Frequencies

		N
MOST - AFL	Negative Differences(a)	12
	Positive Differences(b)	4
	Ties(c)	12
	Total	28

a MOST < AFL
b MOST > AFL
c MOST = AFL

Test Statistics(b)

	MOST - AFL
Exact Sig. (2-tailed)	.077(a)

a Binomial distribution used.
b Sign Test

5. CONCLUSIONS

This paper presents a thorough evaluation of three logical sensor fusion algorithms and an adaptive fuzzy logic algorithm in an experimental setup including three different logical sensors. Results indicate that the best performing algorithm is the adaptive fuzzy logic with 0.075 confidence level corresponding to previous results [Cohen *et al.,* 2005]. Future research is aimed at evaluating the performances of a new sensor fusion algorithm that uses non-binary grid maps [Kapach, 2007].

ACKNOWLEDGEMENTS

This research was partially supported by the Paul Ivanier Center for Robotics Research & Production Management, and by the Rabbi W. Gunther Plaut Chair in Manufacturing Engineering, Ben-Gurion University of the Negev.

REFERENCES

[1] Abidi M. A. and Gonzales R. C. 1992. *Data fusion in robotics and machine intelligence*, Academic Press, San Diego, CA.
[2] Arkin E. M. *et al.,* 2000. Approximation algorithms for lawn mowing and milling, *Computational Geometry*, 17(1-2): 25-50.
[3] Blum R. S. *et al.,* 1997. Distributed detection with multiple sensors: Part II – advanced topics, *Proceedings of the IEEE*, 85(1): 64-79.
[4] Brooks R. A. 1982. Solving the find-edge problem is good representation of free space, *Proceedings AAAI-82*: 381-387.
[5] Chen S. 1987. Multisensor fusion and navigation of mobile robots, *International Journal of Intelligent Systems*, 2(2): 227-251.
[6] Cohen O. 2005. *Grid-map Based Sensor Fusion for Autonomous Mobile Robots*, Ph.D. thesis, Ben-Gurion University of the Negev, Beer Sheva, Israel.
[7] Cohen O. and Edan Y. 2004. Adaptive fuzzy logic algorithms for sensor fusion mapping, *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, The Hague, The Netherlands, October 10-13: 2326-2331.
[8] Cohen O. *et al.,* 2005. Statistical evaluation method for comparing grid-map based sensor fusion algorithms, *The International Journal of Robotics Research*, 25(2):117-133.
[9] Daniel F. *et al.,* 2005. Controlling the Navigation of a Mobile Robot in a Corridor with Redundant Controllers, *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, Barcelona, Spain, April 2005: 3855-3860.
[10] Mamdani E. and Assilian S., 1975, an Experiment in Linguistic Synthesis with a Fuzzy Logic Controller. *International Journal of Man-Machine Studies*, 7(1): 1-13.
[11] Henderson T. and Shilcrat E. 1984. Logical sensor system, *Journal of Robotic Systems*, 1(2): 169-193.
[12] Hollander M. and Wolfe D. A. 1973. *Nonparametric statistical methods*, John Wiley & Sons Inc, New York, NY.

[13] Kapach K. 2007. *Evaluation of grid map based sensor fusion algorithms*, M.Sc. thesis, Dept. of Industrial Engineering and Management, Ben-Gurion University of the Negev, Beer Sheva, Israel.

[14] Klein L. A. 1993. A Boolean algebra approach to multiple sensor voting fusion, *IEEE Transactions on Aerospace and Electronic Systems*, 29(2): 317-327.

[15] Lin H. H. *et al.,* 2003. Ultrasonic self localization and pose tracking of an autonomous mobile robot via fuzzy adaptive extended information filtering, *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*: 1283-1290.

[16] Lozano-Perez T. 1981. Automatic planning of manipulator transfer movements, *IEEE Transactions on Systems Man and Cybernetics SMC*-11(10): 781-798.

[17] Luo R. C. and Kay M. G. 1989. Multisensor integration and fusion in intelligent systems, *IEEE Transactions on Systems, Man. and Cybernetics*, 19(5): 901-931.

[18] Luo R. C. *et al.,* 2001. Multilevel multisensor based decision fusion for intelligent animal robot, *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*: 4226-4231.

[19] Moravec H. P. and Elfes A. E. 1985. High resolution maps for wide-angle sonar, *Proceedings of the 1985 IEEE International Conference on Robotics and Automation*: 116-121.

[20] Najjar M. and Bonnifait P. 2005. Towards an Estimate of Confidence in a Road-Matched Location, *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, Barcelona, Spain, April 2005: 2229-2234.

[21] Ribo M. and Pinz A. 2001. A comparison of three uncertainty calculi for building sonar based occupancy grids, *International Journal of Robotics and Automation Systems* 35: 201-209.

[22] Smith, R. C., and Cheeseman, P. 1986. On the representation and estimation of spatial uncertainty. *International Journal of Robotics Research*(4), 56-58

[23] Srinivasan R. 1986. Distributed radar detection theory, *IEEE Proceedings*, 133 pt. F: 55-60.

[24] Stepan P. *et al.,* 2005. Robust Data Fusion With Occupancy Grid, *IEEE Transactions on Systems, Man and Cybernetics* 35(3): 106-115

[25] Tin K. H. and Mitra B. 2002. Complexity measures of supervised classification problems, *IEEE Transactions on Pattern analysis and Machine Intelligence* 24(3): 289-300.

[26] Ye C. and Borenstein J. 2002. Characterization of a 2-D Laser scanner for mobile robot obstacle negotiation, *Proceedings of the IEEE International Conference on Robotics & Automation*, Washington, DC, May: 2512-2518.

[27] Zhen J., Balasuriya A. and Challa S. 2005. Sensor fusion based 3D Target Visual Tracking for Autonomous Vehicles with IMM, *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, Barcelona, Spain, April 2005: 1841-1846.