

# Constructing Dependable Certainty Grids from Unreliable Sensor Data

Wilfried Elmenreich

Lakeside Labs, Mobile Systems Group,  
Institute of Networked and Embedded Systems  
University of Klagenfurt\*,  
Klagenfurt, Austria  
Email: [wilfried.elmenreich@uni-klu.ac.at](mailto:wilfried.elmenreich@uni-klu.ac.at)

**Abstract** — *Measurements from sensors as they are used for robotic grid map applications typically show behavior like degradation or discalibration over time, which affects the quality of the generated maps. This paper presents two novel algorithms for the generation of certainty grids dealing with this behavior.*

*The first algorithm named Fault-Tolerant Certainty Grid (FTCG) performs voting over multiple sensor readings. This approach removes up to  $(n-1)/2$  faulty measurements for grid cells that are updated by  $n$  independent sensors, however requires that each grid cell is covered by at least three different independent sensors.*

*The second algorithm named Robust Certainty Grid (RCG) uses a sensor validation method that detects abnormal sensor measurements and adjusts a confidence value for each sensor. This method supports also reintegration of recovered sensors from transient faults and sensor maintenance by providing a measurement for the operability of a sensor. The RCG algorithm works with at least three sensors with a partially overlapping sensing range and needs fewer sensor inputs and less memory than the FTFCG approach.*

*Results from simulation and an experimental evaluation on an autonomous mobile robot show that under the presence of unreliable sensor data, both algorithms perform better than the Bayesian approach typically used for certainty grids.*

## 1 Introduction

An important task for autonomous mobile robots is building accurate maps of their environment based on information produced by onboard sensors [1]. Such maps can be topological (describing the connectivity of different places) or metric (capturing geometric properties in grid-like structures).

A certainty grid is a metric method to map different sensor readings into a tessellated map describing the surroundings of a mobile robot, being a simple and intuitive representation of distributed spatial sensor information [2]. For mobile robots, usually the grid is modeled as a two-dimensional or, especially for non-flat outdoor environments, a three-dimensional map of the robot's surroundings. Based on the contents of the

grid, the mapping, navigation, and path planning systems of the robot decide on the robot's actions.

In the literature, the term occupancy grid is also used. In some cases, an occupancy grid refers to a structure with binary values for each cell, i.e., occupied or empty. Depending on the application, the grid is initially assumed to be empty [3, 4] or occupied [5]. However, in many cases the term occupancy grid is also used for a metric structure where each cell is assigned a probability of being occupied by an obstacle. Either way, grid-like structures are considered the most successful data structure for mobile robots for the representation of the environment [6].

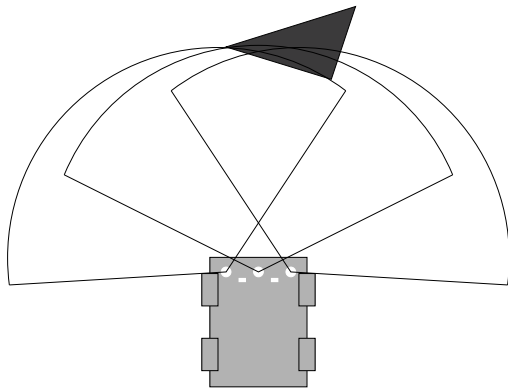
For an autonomous robot, the certainty grid represents a critical data structure that may not become erroneous due to a single sensor failure. Single sensors are affected by several sources of error, like sensor deprivation, limited spatial or temporal coverage, imprecision, cross-sensitivity, and uncertainty [7, 8, 9]. In order to overcome these problems, the inputs from several sensors are combined to form a dependable representation of the environment.

In the original certainty grid by Elfes [2], transient sensor errors are tolerable since typically a cell is updated by several measurements. However, when employing sensors with frequent measurement errors special treatment for fusing measurements into the grid is required in order to prevent the validity of the certainty grid. Analysis of sensors as they are typically used for robotic map applications show that most sensors actually follow the latter scenario, that is they show degradation due to wear-out, staining, and discalibration. In spite of these problems it is necessary to extend the treatment of the sensor measurements in order to construct a dependable map from unreliable sensor data.

In this paper we present two approaches for tolerating sensor faults. The Fault-Tolerant Certainty Grid (MSE) method employs triple coverage of the sensing area and uses voting in order to cope with sensor failures. The Robust Certainty Grid (MSE) approach employs a robust implementation of the certainty grid that uses grid cells which are covered by multiple sensors for a mutual evaluation of the sensor's reliability. This evaluation is then used to determine the future contribution of the evaluated sensor's measurements

---

\*This work was done while the author was with the Vienna University of Technology, Austria.



(a) Obstacle in front of robot

.0	.0	.0	.0	.0	.0	.5	.5	.5	.5	.5	.0	.0	.0	.0
.0	.0	.0	.0	.0	.0	1	1	1	.5	.5	.0	.0	.0	.0
.0	.0	.0	.0	.0	.0	.0	.0	.0	1	.0	.0	.0	.0	.0
.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
.5	.5	.5	.5	.5	.5	.5	.5	.5	.5	.5	.5	.5	.5	.5

(b) Representation in certainty grid

Figure 1: Representation of the robot environment in a certainty grid

to the grid.

The rest of the paper is organized as follows: Section 2 describes the problem statement and the assumptions on sensor failures. Section 3 briefly describes the commonly used certainty grid fusion algorithm and discusses possible shortcomings. Sections 4 and 5 describe the MSE and the MSE approach. An experimental evaluation of all three algorithms is presented in Section 6; the results are discussed in Section 7. Related work to this topic is discussed in Section 8 until Section 9 concludes the paper.

## 2 Problem Statement

We assume a robot using multiple sensors to scan an area of interest for obstacles. One sensor must cover a particular area, this can be achieved with a camera or a distance sensor that is swiveled over a particular sector. The goal is to get a map of the robot’s environment containing obstacles and free space.

Figure 1 a) depicts a schematic four-wheeled robot and three partially overlapping detection areas created by three sensors. In Figure 1 b), an example for the corresponding certainty grid is depicted. Note that for areas not covered by the sensors and for occluded areas (e.g., behind the obstacle) the probability for an obstacle is assumed with 50 % (maximum entropy assumption [10]).

Note that from the view of hardware architecture it is almost impossible to mount sensors in a way that the viewpoint angle from a sensor to an object is perfectly identical to the angle of the replicated sensor. A replicated sensor will thus always be located slightly offside, thus viewing objects from different angles. Even if two sensors are working correctly, they may produce different results. Figure 2 depicts an example for an object that yields ambiguous sensor readings for two correctly operating distance sensors. Although both sensors are working correctly according to their specification, sensor B detects an object for the given region while sensor A detects free space for this particular cell.

### 2.1 Fault Hypothesis

We consider the sensors to form independent fault containment regions, thus a failure in one of the sensors does not cause a failure in another sensor.

We have conducted experiments with several sets of infrared distance sensors for the purpose of analyzing common failure modes of sensors in [11]. According to the results it depends on the configuration of the sensor set if sensor errors are correlated. We assume to have a sensor configuration where sensors have independent failure modes.

A failing sensor delivers either wrong measurements or no measurements. We distinguish the following three types: A *transient sensor failure* is triggered by some random event that causes the sensor to deliver a singular degraded or false measurement instead of a correct one. Martin and Moravec considered the effect of transient sensor failures neglectable in the certainty grid [12]. An *intermittent sensor failure* consists of a sequence of transient sensor failures. After some duration the sensor may recover from the failure. An intermittent sensor failure could be caused by a stained sensor lens that recovers when the dirt is removed. A sensor with a *permanent failure* has broken down abidingly and requires an explicit repair action in order to

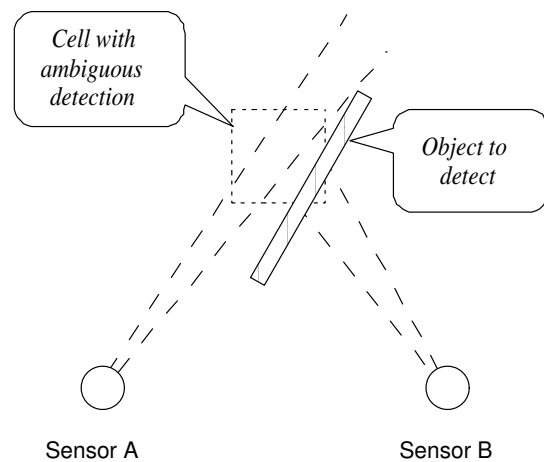


Figure 2: Discrepancy between sensor A and sensor B due to object shape

re-obtain its nominal performance. From a set of  $n$  sensors we assume that at most  $\lfloor (n-1)/2 \rfloor$  sensors deliver incorrect values due to transient, intermittent, or permanent failures.

The sensors are considered the only source of faults in the regarded system. Other failures like communication and node failures are orthogonal to the described sensor failures and can be handled by a dependable system architecture as for example the Time-Triggered Architecture (TTA) [13].

### 3 Certainty Grid Approach using Bayes' Rule

Matthies and Elfes [14] propose a uniform method for integrating various sensor types. Each sensor is assigned a spatial interpretation model, which is developed for each kind of sensor, that maps the sensor measurement into corresponding cells. When sensor uncertainties are taken into account, we arrive at a probabilistic sensor model.

The calculation of new grid values is usually done by Bayesian inference. Suppose that two sensors  $S_1$  and  $S_2$  give two occupancy probability values for a particular grid element *cell*. Assuming *conditional independence* and *maximum entropy* [10], the updated probability of a grid cell being occupied by an obstacle can be calculated using Bayes' rule:

$$\frac{1}{\mathbb{P}(\text{cell.occ}|S_1, \dots, S_n)} - 1 = \prod_{i=1}^n \left( \frac{1}{\mathbb{P}(\text{cell.occ}|S_i)} - 1 \right) \quad (1)$$

where  $\mathbb{P}(\text{cell.occ}|S_i)$  depicts the probability of *cell* being occupied given the measurement of sensor  $S_i$ .

Equation 1 shows fully associative and commutative behavior. Thus, the order of processing does not influence the result.

Basically, it is assumed that the certainty grid application has no *a priori* knowledge on the geometry of its environment and that objects in this environment are static or change with a neglectable rate. The effect of transient sensor errors can usually be neglected, because they have little effect on the grid [12]. However, this approach does not provide means to handle transient or permanent sensor faults which would cause errors in the grid that significantly affect navigation or map-building systems.

### 4 Fault-Tolerant Certainty Grid Approach

The MSE approach uses a separate entry of an occupancy value for each sensor. Thus, each sensor produces its own grid independently of the other sensors. The concise world model is built by fusing all sensor

grids. The proposed fault tolerant algorithm is similar to the Fault-Tolerant Average algorithm for clock synchronization [15].

The fault-tolerant fusion is performed as follows: First the input data from all sensors for each grid cell are gathered. If the sensors do not update the grid cells simultaneously, the fusion of the measurements into the grid cell is delayed until all sensor data has been gathered for a particular grid cell. It is assumed that the environment does not change significantly during the measurement phase.

Then from the set of proposed certainty values for each grid cell the median value is selected to update the particular cell. Each cell requires updates from at least three independent sensors in order to tolerate a faulty measurement. If the measurement is lower or higher than all the other measurements, it is voted out by the median selection. If the measurement is between two correct measurements, this benign value will be used, but not distort the certainty grid.

As long as there is at most one fault out of three measurements (or  $\lfloor \frac{n-1}{2} \rfloor$  faulty measurements out of  $n$  measurements) for a particular cell, the faulty measurement does not cause a wrong estimation for this grid cell. However, this fault-tolerant approach comes with two drawbacks. First, it potentially hinders a lot of correct sensor measurements from contributing to the grid. Second, the method requires at least triple coverage to be fault tolerant. Typically, the sensors' field of view is expected not to full overlap due to geometrical properties of robots. For the regions not covered by at least three sensors, the median selection is not fault-tolerant.

In order to overcome these problems we present a sophisticated algorithm for handling sensor faults and use the fault-tolerant certainty grid approach as a reference for its performance.

## 5 Robust Certainty Grid

The MSE has been designed in order to support robustness against sensor failures while not requiring a full triple redundancy for the whole sensor system.

### 5.1 Robust Certainty Grid Algorithm

In contrast to the MSE approach, the MSE algorithm fuses all sensor inputs immediately into a single certainty grid. Together with the grid occupancy value *occ* each cell stores information about the so-called *owner* of the current *occ* value within the cell. Typically, the owner is the sensor that updated this cell most recently. This property of each cell is named the current *owner* of the cell:

$$\text{cell.owner} = \begin{cases} 0 & \text{unknown} \\ 1 & \text{sensor 1} \\ \vdots & \\ n_{\text{sensors}} & \text{sensor n} \end{cases}$$

---

```

procedure AddToGrid( sensor, cell )
begin
  if (cell.owner = unknown) or (cell.owner = sensor) then
    cell.occ := sensor.measurement;
    cell.owner := sensor;
  else
    comparison := 4*(cell.occ-0.5)*(sensor.measurement-0.5);
    weight1 := abs(cell.occ-0.5)*cell.owner.conf;
    weight2 := abs(sensor.measurement-0.5)*sensor.conf;
    if weight1 = weight2 then
      cell.occ := (cell.occ+sensor.measurement) / 2;
    else
      cell.occ := (cell.occ*weight1+sensor.measurement*weight2)
        / (weight1 + weight2);
    if comparison > CONFIRMATIONTHRESHOLD then
      inc(cell.owner.conf);
      inc(sensor.conf);
    if comparison < CONTRADICTIONTHRESHOLD then
      dec(cell.owner.conf);
      dec(sensor.conf);
    contribution := 4*(cell.occ-0.5)*(sensor.measurement-0.5);
    if contribution > CONTRIBUTIONTHRESHOLD then
      cell.owner := sensor;
    else
      cell.owner := unknown;
end

```

---

Figure 3: AddToGrid algorithm

All grid cells are initialized with  $cell.occ = 0.5$  and  $cell.owner = unknown$ . When a new measurement has to be added to the grid, the following *AddToGrid* algorithm is executed (Figure 3 lists the algorithm in pseudocode):

- If the particular grid cell has an empty owner field or the cell owner is identical with the contributing sensor, the measurement of the sensor is taken as is and the cell stores the index of the sensor as new owner.
- If there is a different contributor than the current owner, the measurement is first compared to the cell value  $cell.occ$  by calculating a value named *comparison*.

The value of *comparison* is higher if the measurement is similar to the cell’s occupancy value and lower if the measurement contradicts the cell’s current occupancy value. If *comparison* is above a particular *confirmation threshold*, we speak of a *confirmation* of cell value and new measurement. If *comparison* is below a particular *contradiction threshold*, we speak of a *contradiction* of the cell value and the new measurement. In case of a confirmation, the confidence values of the new sensor and the sensor named in the owner field are both increased. In case of a contradiction, the confidence values of the new sensor and the sensor named in the owner field are both decreased. If *comparison* is not significant, it does neither yield a confirmation nor a contradiction.

- The new occupancy value of the cell is calculated as a weighted average between old value and actual measurement. The weights are derived from the respective confidence values and the *significance* of the measurement. A measurement is more significant if it has a greater absolute distance to the *uncertain* state (0.5).

- The *contribution* is a measurement of the consistency of the sensor measurement with the new  $cell.occ$  value. If the *contribution* is above a certain threshold, the contributing sensor becomes the new owner of the cell. Otherwise the  $cell.owner$  value is reset to *unknown*.

## 6 Experimental Evaluation

We used two different test environments to evaluate the proposed algorithms, a real mobile robot and a simulation environment. While the experiments on real hardware provide a better understanding on the applicability of the approach, some evaluations have been done in a simulated environment with controlled reproducible sensor inputs.

### 6.1 Mobile Robot Experiment

The mobile robot comprises a model car called Smart Car equipped with a suit of pivoted distance sensors, two ultrasonic sensors pointing straight forward, an electric drive, and a steering unit (see Figure 4).

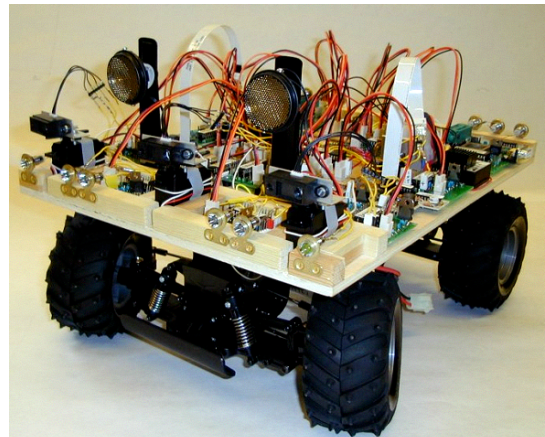
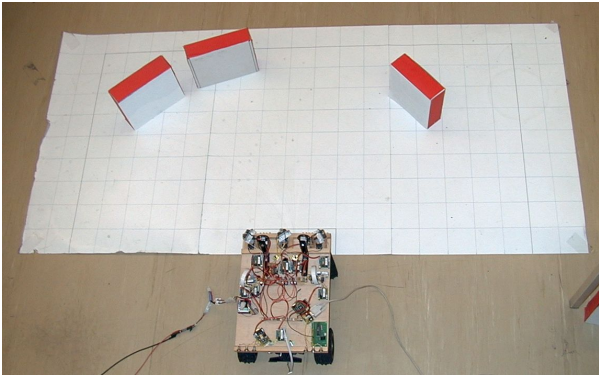


Figure 4: Smart Car: Autonomous mobile robot with pivoting sensors

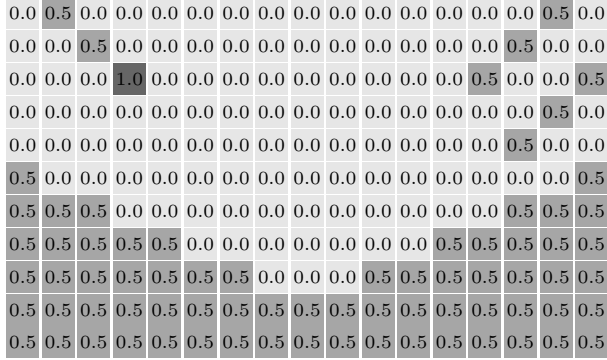
The certainty grid is built from the input of three Sharp GP2D02 infrared sensors. These type of sensors provide a rather narrow sensor beam, however they suffer from a significant amount of transient sensor errors. Environmental influences such as a change in lighting may cause also intermittent failures.

The statistical behavior of the infrared sensor errors has been analyzed in [16]. Apart from a limited accuracy, it is difficult to distinguish a correct measurement from the returned measurement signal when no object is within detection range (80 cm). The behavior of the sensors has been improved by an algorithm that tries to predict the correct interpretation based on measurement fluctuation, which works well for particular sensors, but has a remaining probability of failure for others [16].

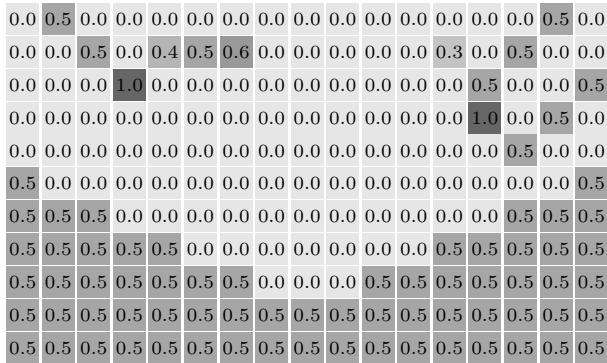
We have analyzed several samples of GP2D02 sensors by comparing the calibrated sensor measurement



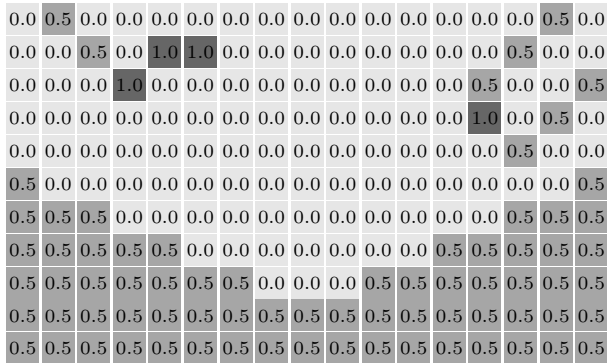
(a) Experiment setup



(b) Bayesian fusion



(c) Robust certainty grid method



(d) Fault-tolerant median method

Figure 5: Evaluation of algorithms on real data

to the real distance to an object. Figure 6 depicts the measured distribution of sensor errors for three different sensors.

The results obtained from the experiment with the Smart Car are depicted in Figure 5. Figure 5(a) shows a photograph of the test environment, while Fig-

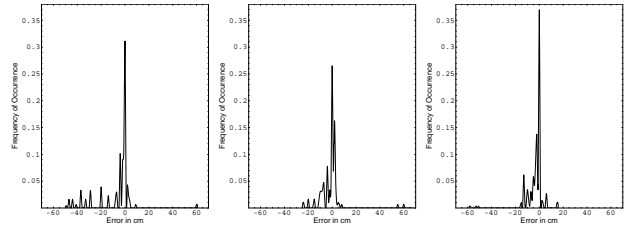


Figure 6: Error histograms for three Sharp GP2D02 infrared sensors (from [16])

	accuracy	$P_{failed\_inf}$	$P_{false\_inf}$
Sensor A	6%	16%	4%
Sensor B	10%	18%	8%
Sensor C	7%	< 1%	< 1%
Sensor D	12%	18%	9%
Sensor E	3%	< 1%	< 1%

Table 1: Measured sensor properties

ures 5(b–d) depict the certainty grids generated from the sensor data using different algorithms.

The probability value of a grid cell has been implemented as an 8-bit integer value. Thus, a value of 0 corresponds to the *free* state ( $p = 0$ ), 128 means the *uncertain* state ( $p = 0.5$ ) while 255 is used to express the *occupied* state ( $p=1.0$ ) of a grid cell. The certainty grid had a size of 17 times 11 cells whereas each cell corresponds to a 10 cm times 10 cm square.

## 6.2 Evaluation by Simulation

The MSE and the MSE algorithms have been evaluated together with Bayesian fusion. For the evaluation we have implemented a simulation program that emulates an arbitrary number of sensors that are swept around in order to map a given artificial environment.

We have simulated systems with 3, 4, and 5 sensors. For each sensor we emulated the measuring behavior of a corresponding real GP2D02 sensor. The sensors showed a great variance in their accuracy and the ability to detect situations without objects in detection range. Table 1 shows the sensor properties for five different GP2D02 sensors that have been used for the simulation. The value  $p_{failed\_inf}$  determines the percentage of fails in detecting this situation, while  $p_{false\_inf}$  determines the percentage of false alarms. Using this system, we have simulated the measurement and processing of 1560 sensor measurements for each of the three fusion methods.

	MAE·10 <sup>-3</sup>	MSE·10 <sup>-3</sup>
Bayesian fusion	101	69
Robust certainty grid	22	12
Fault-tolerant certainty grid	9	6

Table 2: Performance of certainty grid algorithms by Mean Average Error (MAE) and Mean Squared Error (MSE)

Table 2 shows a comparison of the average results from the simulation. The grid has been built from simulated sensor measurements for different obstacle configurations using the MSE, Bayesian fusion, and the MSE algorithm. The evaluation criterium is the difference between a grid generated by a particular algorithm to a certainty grid generated by perfect virtual sensors. Thus, the value of each grid cell has been compared yielding a Mean Average Error (MSE) and a Mean Squared Error (MSE) for each method. The MSE algorithm achieved the lowest deviation, i. e., best performance. Both, the MSE and the MSE approach achieved a better performance than the standard Bayesian fusion.

### 6.3 Simulation of Intermittent Sensor Failures

The MSE uses a sensor validation method that detects abnormal sensor measurements and adjusts a confidence value for each sensor. This method supports the detection and reintegration of recovered sensors from intermittent faults. This self-healing behavior is related to the concept of self-stabilization as introduced by Dijkstra in [17].

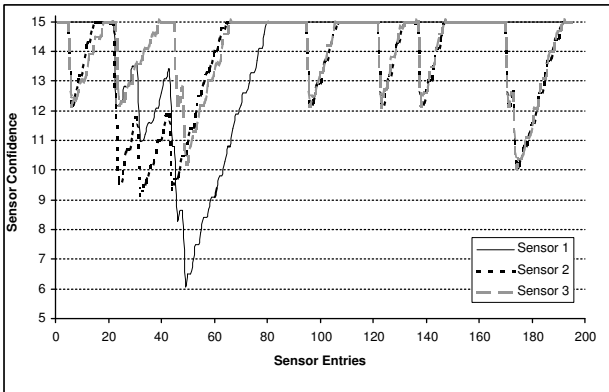


Figure 7: Recovery of sensor confidence after a transient failure in the RCG algorithm

Figure 7 depicts the confidence values of a set of three sensors from a simulation of about 200 sensor readings using the MSE algorithm. All sensors had been initialized with the highest confidence, but sensor 1 suffered from a transient failure during its first 50 readings. The simulated failure added a noise signal of 20% of the sensor’s range to the output signal. While that failure persisted, the confidence of sensor 1 decreased. The confidence values of sensor 2 and 3 were also fluctuating due to inconsistencies in their readings and contradictions with sensor 1. As shown in the figure, the confidence value of sensor 1 recovered within about 30 sensor readings after removal of the transient failure.

## 7 Discussion

Examination of the infrared distance sensors showed that, due to wear-out, staining, and discalibration, real sensors typically show degradation behavior over time. Using such sensors as inputs to grid mapping applications leads to errors in the resulting certainty grid. Thus, the generated map deviates from the real environment, which has been shown in real experiments featuring a mobile robot and in simulation experiments.

The experiments have been conducted for a robot in a static environment but apply as well to a moving robot that updates its position in the map via a positioning service (that could be also derived from map and sensor data following a Simultaneous Localization And Mapping (MSE) approach)). We assumed that the environment is mostly static or changes slowly so that subsequent sensor measurements observe a consistent environment. This requirement can be overcome if object movements can be tracked and the position of an object can be predicted using previous positions. This scenarios, however, are considered extensions outside the scope of the paper.

The approaches presented in this paper need at least three independent sensors in order to compensate a single faulty measurement. In contrast to the MSE algorithm, the MSE algorithm gains extra sensor space, because the sensor views may overlap only partially. When considering implementations on small embedded devices such as a Microcontroller with few kilobytes of on-chip RAM, the memory requirements for both approaches are of interest. The MSE algorithm needs to store a certainty grid for each sensor separately (the values of the resulting grid may be calculated on demand in order to save memory) ( $n_{sensors} \cdot gridheight \cdot gridwidth$  memory elements), whereas the MSE needs  $gridheight \cdot gridwidth$  memory elements for the certainty values and  $\frac{\lceil \log_2(n_{sensors}+1) \rceil}{8} \cdot gridheight \cdot gridwidth$  extra bytes of memory for the storage of the owner values. The memory requirements for the confidence values can usually be neglected, since the number of sensors normally is remarkably lower than the total number of cells in the grid. Thus, the memory requirements of the MSE algorithm are considerable lower than the memory consumption of the MSE approach.

In contrast to Bayesian fusion and the MSE algorithm, the *AddToGrid* procedure (as shown in Figure 3) of the MSE is sensitive to the ordering of measurements. Thus, when a grid cell is updated by subsequent measurements, the order of updates makes a difference in the result. In order to achieve deterministic behavior, a consistent ordering of measurement updates is required. In time-triggered systems such as the TTA [13] this problem is already solved by the architecture.



## 8 Related Work

Most implementations of certainty grids in the literature assume *a priori* known statistical parameters about the sensor behavior, examples are [18, 19, 20]. A notable exception is presented in [21] where an adaptive algorithm is described based on Adaptive Fuzzy Logic (AFL). As it is the case for the described scenario with intermittent sensor failures, the experiments yielded a hysteresis in the performance measure for a recovering sensor. However, the AFL approach has the disadvantage of a relatively high complexity of  $\Omega(2^n_{sensors})$ , which is crucial for its implementation on embedded platforms.

Thrun presents a forward sensor model approach [22] that overcomes some sensor discrepancy problems by using a global interpretation of a sensor measurement instead of regarding single grid cells. This approach is related to our algorithm in the way that it uses information from multiple cells in order to decide for the interpretation of a sensor's measurement, but instead of measurement errors, it takes effect on discrepancies of the sensor models.

A related problem to the problem described in this paper is given by multi-robot mapping [23]. As well as in our approach information from redundant sensors has to be agreed for the purpose of a dependable sensor grid. Yet, multi-robot mapping depends heavily on solving the MSE problem, at least to find the relative positions of robots to each other, while in our case, the relative positions of the sensors on the robot are known *a priori* with sufficient accuracy.

## 9 Conclusion

The main contribution of this paper is the introduction and evaluation of two methods for handling faulty measurements when building a world model in form of a certainty grid map. The innovative aspect lies in implementing a mutual cross-checking of sensor measurements into a certainty grid method in order to infer about sensor degradation and to reduce the influence of sensor errors on the generated grid.

An analysis of distance sensors as they are typically used for robotic applications showed that the presented approach is relevant with respect to the effects of typical sensor errors. The approach, nevertheless, is a generic one that is applicable to various sensor types.

In the presence of faulty measurements, all methods show a better behavior than the Bayesian fusion. From the examined performance and the resource requirements, the MSE algorithm and the fault-tolerant median selection are most promising. While the fault-tolerant median selection shows the best performance, the MSE algorithm needs a significantly less amount of memory to store the certainty grid. However, if the certainty grid is only generated for a limited space while a second data structure is used as a global map, the resource requirements for the fault-tolerant median se-

lection are also suitable for the implementation in embedded systems.

The described algorithms support on the one hand the usage of cheap low-cost sensors for map building and on the other hand a condition-based maintenance scheme for robots where sensor deprivation is expected during system lifetime.

## Acknowledgments

This work was supported by the Austrian FWF project TTCAR under contract No. P18060-N04. I would like to thank Christian El-Salloum, Roman Obermaisser, Claudia Pribil, and Ingomar Wenzel for comments on an earlier version of this paper.

## References

- [1] U. Raschke and J. Borenstein. A comparison of grid-type map-building techniques by index of performance. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 3, pages 1828–1832, May 1990.
- [2] A. Elfes. Robot navigation: Integrating perception, environmental constraints and task execution within a probabilistic framework. In *International Workshop on Reasoning with Uncertainty*, pages 93–129, Amsterdam, Netherlands, 1995.
- [3] C. S. Andersen, C. B. Madsen, J. J. Sørensen, N. O. S. Kirkeby, J. P. Jones, and H. I. Christensen. Navigation using range images on a mobile robot. *Robotics and Autonomous Systems*, 10:147–160, 1992.
- [4] B. Y. S. Rao, H. F. Durrant-Whyte, and J. A. Sheen. A fully decentralized multi-sensor system for tracking and surveillance. *International Journal of Robotics Research*, 12(1):20–44, 1993.
- [5] A. Hoover and B. D. Olsen. Sensor network perception for mobile robotics. In *Proceedings of the IEEE International Conference on Robotics and Automation, San Francisco, CA*, pages 342–347, April 2000.
- [6] D. Kortenkamp, R. P. Bonasso, and R. R. Murphy. *Artificial Intelligence and Mobile Robots: Case Studies of Successful Robot Systems*. AAAI Press, 1998.
- [7] W. Elmenreich, W. Haidinger, H. Kopetz, T. Losert, R. Obermaisser, M. Paulitsch, and C. Tröhdhandl. Initial demonstration of smart sensor case study. *DSoS Project (IST-1999-11585) Deliverable PCE3*, April 2002.
- [8] R. R. Murphy. Biological and cognitive foundations of intelligent sensor fusion. *IEEE Transactions on Systems, Man and Cybernetics*, 26(1):42–51, January 1996.
- [9] K. E. Foote and D. J. Huebner. Error, accuracy, and precision. Technical report, The Geographer's Craft Project, Department of Geography, University of Texas at Austin, 1995.
- [10] B. Moshiri, M. R. Asharif, and R. Hosein Nezhad. Pseudo information measure: A new concept for extension of Bayesian fusion in robotic map building. *Information Fusion*, 3(1):51–68, 2002.

- [11] A. Schörgendorfer. Extended confidence-weighted averaging in sensor fusion. Master's thesis, Technische Universität Wien, Institut für Technische Informatik, Vienna, Austria, 2006.
- [12] M. C. Martin and H. P. Moravec. Robot evidence grids. Technical Report CMU-RI-TR-96-06, The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA, 1996.
- [13] H. Kopetz and G. Bauer. The Time-Triggered Architecture. *Proceedings of the IEEE*, 91(1):112–126, January 2003.
- [14] L. Matthies and A. Elfes. Integration of sonar and stereo range data using a grid-based representation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 727–733, Philadelphia, PA, USA, 1988.
- [15] J. Lundelius and N. Lynch. A new fault-tolerant algorithm for clock synchronization. In *Proceedings of the 3rd Annual ACM Symposium on Principles of Distributed Computing*, pages 75–88, Vancouver, Canada, August 1984.
- [16] W. Elmenreich. *Sensor Fusion in Time-Triggered Systems*. PhD thesis, Technische Universität Wien, Institut für Technische Informatik, Vienna, Austria, 2002.
- [17] E. W. Dijkstra. Self-stabilizing systems in spite of distributed control. *Communications of the ACM*, 17(11):643–644, 1974.
- [18] R. C. Luo and M. Kay. Multisensor integration and fusion in intelligent systems. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(5):901–930, September–October 1989.
- [19] M. A. Abidi and R. C. Gonzalez. The use of multisensor data for robotic applications. *IEEE Transactions on Robotics and Automation*, 6(2):159–177, April 1990.
- [20] R. R. Brooks and S. S. Iyengar. *Multi-Sensor Fusion: Fundamentals and Applications*. Prentice Hall, New Jersey, 1998.
- [21] O. Cohen and Y. Edan. Adaptive fuzzy logic algorithm for grid-map based sensor fusion. In *Proceedings of the IEEE Intelligent Vehicle Symposium*, pages 625–630, Parma, Italia, June 2004.
- [22] S. Thrun. Learning occupancy grids with forward sensor models. *Autonomous Robots*, 15:111–127, 2003.
- [23] A. Birk and S. Carpin. Merging occupancy grid maps from multiple robots. *Proceedings of the IEEE*, 94(7):1384–1397, 2006.