

Building World Models for Mobile Platforms using Heterogeneous Sensors Fusion and Temporal Analysis

Michel Beauvais, Chris Kreucher, Sridhar Lakshmanan

Vehicle Electronics Laboratory, Electrical and Computer Engineering Department

University of Michigan – Dearborn, 4901 Evergreen Dearborn, MI 48128-1491, USA

Phone: (313) 593-5516, Fax: (313) 593-593-9967, Email: michelab, ckreuche, lakshman@umich.edu

Keywords

Autonomous Navigation, Temporal Correlation, Intelligent Vehicles, Deformable Templates.

1 Introduction

The Michigan Offroad Sensor Fusing Experimental Testbed (*MOSFET*) is an autonomous vehicle designed to navigate an unknown terrain which includes obstacles, lanes, and other traditional road-side obstructions. In order to gather data about the world around it, *MOSFET* is equipped with several heterogeneous sensors. These sensors include one forward-looking camera used for obstacle detection and lane tracking; two side-looking cameras used for lane sensing; and a bank of sonar sensors used to detect physical obstacles (see Figure 1).

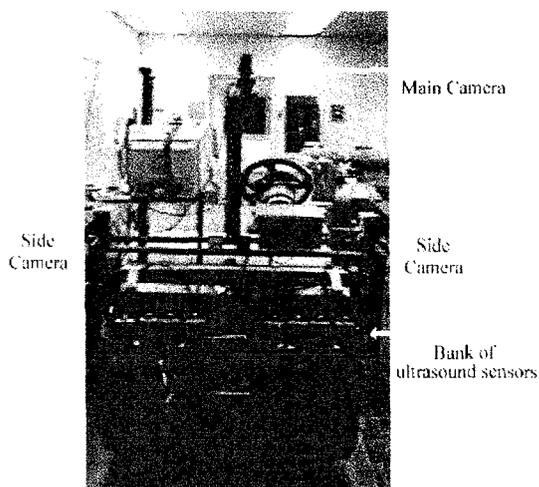


Figure 1: *MOSFET*

MOSFET's intelligence consists of three steps. First, it extracts information from each individual sensor. Next, it accomplishes a systematic fusion of the information extracted from the individual sensors. Finally, it augments this fused information by the use of temporal analysis, which keeps track of previous obstacle information and updates that information in accordance with the movement of the vehicle. *MOSFET* navigates by keeping within the lanes while simultaneously avoiding obstacles and other road-side obstructions.

The crux of this paper is in steps 2 and 3, namely, the process of building a world model for *MOSFET*, based on the information gathered from the individual sensors, and augmenting this model through the use of temporal data. These 2 steps are accomplished in three stages:

Deformable Template Stage - *MOSFET*'s position and orientation, with respect to the left and right lane markers, are obtained by using *STARLITE*, a straight-edge detecting deformable template algorithm that uses a signal-to-noise ratio (SNR) matching criteria (see [1]). The input to *STARLITE* consists of the left and right side-looking camera images. *STARLITE*'s output is subsequently used to detect lane boundaries in the forward-looking camera image. Using position and orientation to fix three parameters of a parabolic deformable template, the lane's curvature is estimated by a Hough transform-like edge accumulation procedure.

Correlation Stage - The ultrasound sensors are used to detect physical obstacles. As a separate step, a color segmentation algorithm called *OLSON* is also used to distinguish obstacles by analyzing the forward-looking camera image. By correlating the two obstacle extraction results, together with the lane information from the deformable template stage, the exact position and extent of obstacles that are relevant to *MOSFET*'s navigability are determined. The lanes and the obstacles detected are all laid out in a grid of *MOSFET*'s immediate world. Each grid element represents a certain distance and angular position from *MOSFET*. Associated with each element is a value proportional to its "relative obstacleness." The relative obstacleness is measured by a weighted combination of the lane, ultrasound, and color algorithm outputs. Grid elements that are outside of the detected lanes are deemed "severe" obstacles.

Augmentation Stage - *MOSFET*'s current obstacle grid of the world is combined with its previous grids. A decayed sum of all the previous "relative obstacleness" values plus the new one constitutes the accumulated obstacleness for each square grid. A cleaning procedure to get rid of stray/isolated single square obstacles is performed.

The ideas contained in this paper are not all new. A fusion of forward-looking visual images and ultrasound data for mobile platforms is given in [2]. A similar fusion algorithm can also be found in [3]. The accumulated grid algorithm for ultrasound and laser range data was presented in [4,5]. Another temporal analysis algorithm for mobile platforms is given in [6]. Suffices to say, a considerable amount of previous work has been reported on the both heterogeneous sensor fusion and temporal information analysis. The contribution of this paper is a novel combination and adaptation of such previously reported ideas along with the use of some new deformable template-based edge detection algorithms and information theory based color segmentation algorithms.

2 The Deformable Template Stage

MOSFET employs a modified version of the *STARLITE* algorithm for side-looking camera lane detection. *STARLITE* utilizes a deformable template model to describe the shape of an ideal lane. As Figure 2 shows, the deformation template's four parameters consist of: the intensity of the background (μ_{pavement}), the intensity of the lane (μ_{lane}), the width of the lane marker (ω), and the position of the lane marker (p).

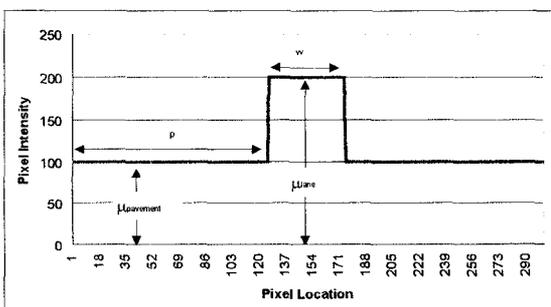


Figure 2: The *STARLITE* deformable template model

STARLITE scans one row of the image and then employs a modified SNR-based matched filter criteria to match the deformable ideal template lane marker to the intensity profile of that scan row:

$$\text{SNR} = \frac{(\mu_{\text{lane}} - \mu_{\text{pavement}})^2}{\sigma_{\text{lane}}^2 + \sigma_{\text{pavement}}^2} \quad (1)$$

As was mentioned earlier, the deformable template stage of this work uses a modified version of *STARLITE*. We process three scan lines instead of just one in order to determine both offset and orientation of the lane with respect to the vehicle. This modified algorithm is applied to the side-looking camera images in order to find *MOSFET*'s position and orientation with respect to the left and right lanes. Figure 3

presents examples of *STARLITE* locating lanes in typical outdoor images.

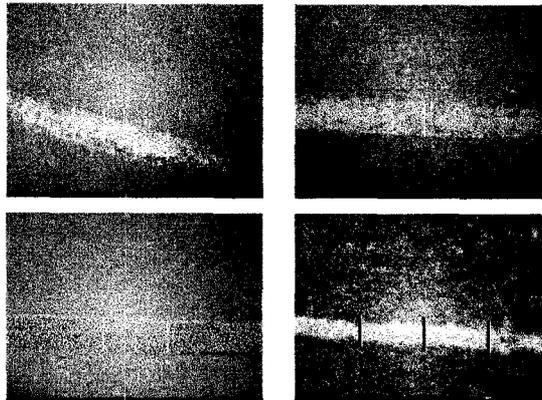


Figure 3: Lane finding using a modified version of *STARLITE*. Lanes are found in three scanlines and are marked with a vertical bar to indicate their position

Once the lane position and orientation have been identified from the side-looking cameras, this information is then fed to another deformable template algorithm that analyzes the forward-looking camera image. Potential lane pixels in the forward-looking camera image are first identified by using a color segmentation algorithm called *OLSON*. *OLSON* assumes that training samples of typical lane markers and backgrounds are available. Using the mean and covariance of the training samples as prior knowledge, *OLSON* transforms a given color image into a gray-tone image. This transformation is effected in such a way that the contrast between the lane pixels and background pixels in the gray-tone image is maximized, in a Fisher information sense. Potential lane pixels are then identified by using a simple (histogram-based) thresholding technique. Figure 4 shows a couple of examples of *OLSON*'s output.

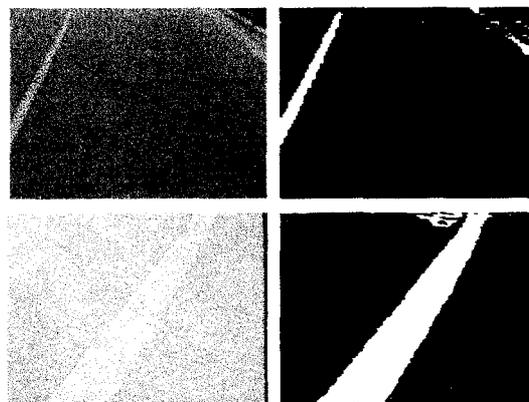


Figure 4: *OLSON* output images for different color lane markers

Notice that *OLSON* works equally well for both white and yellow lane markers.

Lanes in the front-looking camera are modeled as parabolas:

$$r = kc^2 + mc + b \quad (2)$$

where k denotes the curvature, m the orientation and b the offset of the lanes. Since the lanes are assumed to be parallel, they share the same k and m values, and the left and right lanes are distinguished by their differing b_L and b_R values. *STARLITE* determines the m , b_L and b_R values. Therefore, finding the lanes in the front-looking camera is equivalent to fitting two parabolas with the same unknown k to the pixels identified by *OLSON* in the front-looking camera image as being potential lanes. The fitting procedure is accomplished by a 1-D maximization over the k space of a Hough-like edge aggregate measure:

$$k^* = \arg \max_k \sum_{r,c} Lane(r,c) \quad (3)$$

where, $Lane(r,c)$ denotes *OLSON*'s output - a binary value denoting whether or not (r,c) is a potential lane pixel, and the summation is over only those (r,c) 's that satisfy eq. (2).

Figure 5 shows three examples of this combined deformable template lane detection procedure.

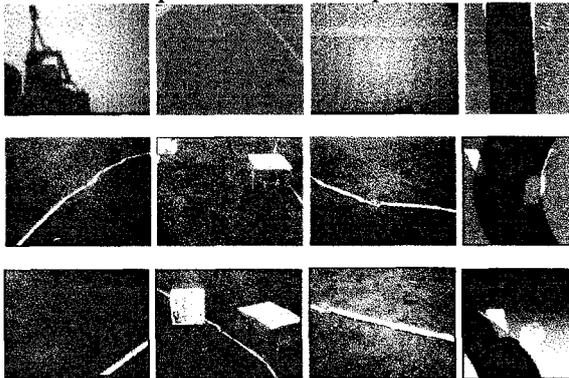


Figure 5: (L to R) Left Camera Image, Main Camera Image, Right Camera Image, Ground Plane Worldmap

Notice that the lane information is accurate even in the presence of obstacles obscuring the lane markers and/or when one of the lanes is invisible in the forward-looking camera image.

3 Correlation Stage

The main camera image is processed again to determine the location of potential obstacles in front of the vehicle. This obstacle location is also

accomplished by *OLSON* (trained on samples of typical obstacles and backgrounds). Figure 6 shows three examples of *OLSON* segmenting a typical main camera image in order to determine the location of potential obstacles:

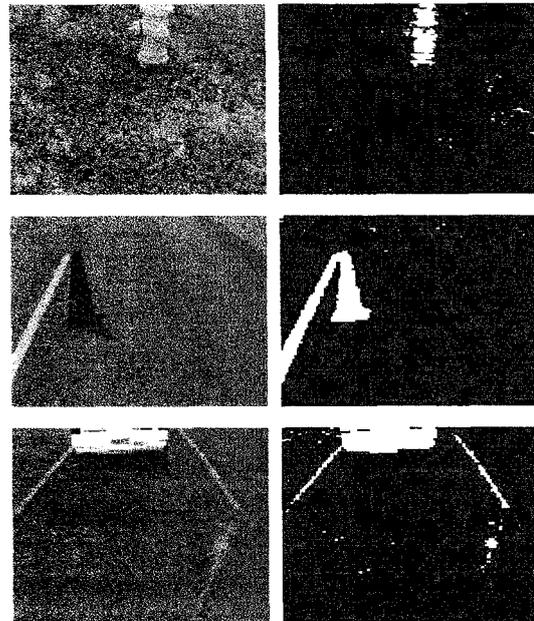


Figure 6: (L to R) Main camera image, *OLSON* binarized image

Notice that *OLSON* locates obstacles of varying colors equally well.

The ultrasound sensors are then used to locate physical obstacles. Since the ultrasonic sensors return only the range of an object as information; the angular position of the obstacle is calculated using the trivial sonar model shown in Figure 7. In this model, the range

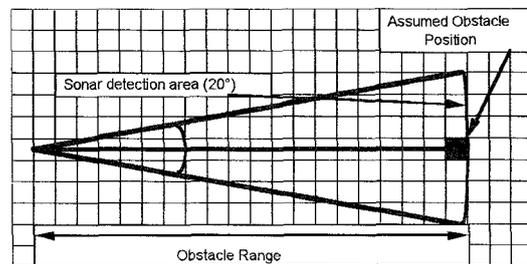


Figure 7: Trivial Sonar Model

information is used along with the known sonar detection cone to assume an obstacle's angular location as the center of the conic region swept out by the sonar.

Both *OLSON* and the ultrasound model result in false alarms and missed obstacles. "True" obstacles are

highlighted by a weighted combination of the two obstacle detection results (in the ground plane), and registered against the detected lanes. Shown in Figure 8 are three examples of such correlated maps of the world in front of *MOSFET*.

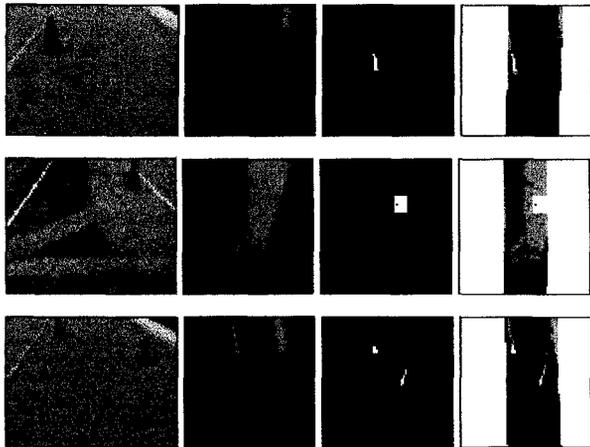


Figure 8: (L to R) Main camera image, *OLSON* obstacle map, sonar obstacle map, combined *OLSON* and sonar static world map.

4 Augmentation Stage

After the information extracted from the various heterogeneous sensors is correlated to form a map of the world in front of *MOSFET*, this map is augmented by use of temporal information. This temporal information essentially consists of similar world maps retained from previous iterations (see Figure 9).

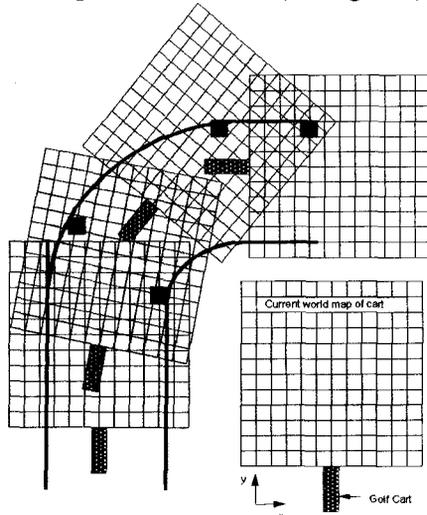


Figure 9: Navigation grid updated for vehicle movement

To understand this augmentation process, one has to understand the accumulated grid model

algorithm. This algorithm divides up the world currently in front of *MOSFET* into a grid where each grid element is at a certain fixed distance and angle relative to *MOSFET*. The correlation stage outputs the relative obstacleness at each grid element of the current world. This same grid element (accounting for changes in its orientation due to *MOSFET*'s motion) may have been a part of *MOSFET*'s previous world maps. If it is, then a time-decayed version of its previous relative obstacleness values are added to the current one. The absolute obstacleness value for each grid point in *MOSFET*'s current path is determined by a decision tree that also takes into account the obstacle's size:

$$Abs_Obs = \begin{cases} 0 & \text{Rel_Obs} = 0 \\ \frac{1}{8} & \sum \text{Rel_Obs} = 0 \\ \frac{1}{4} & \sum \text{Rel_Obs} < Th \\ 1 & \sum \text{Rel_Obs} > Th \end{cases} \quad (4)$$

where the sum is over neighborhood grid elements and *Th* is a predetermined (absolute) obstacleness threshold.

The accumulated grid model algorithm relies on accurate information regarding *MOSFET*'s motion: the distance *s* that the vehicle has moved over the last time increment and the steering angle θ of the vehicle over that time increment, as shown in Figure 10. This information is obtained from *MOSFET*'s motion controller.

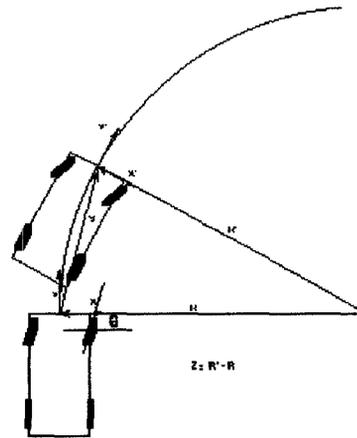


Figure 10: Vehicle Movement Model

Using *s* and θ , and assuming those values stay the same over that small time increment, we calculate three motion parameters. First, the change in the heading angle $\Delta\phi$:

$$\Delta\phi = \frac{s \cdot \sin(\theta)}{w} \quad (5)$$

where w is the wheel base of the vehicle - the distance between the front and rear axles. Once the change in heading angle has been calculated, the translational movement Δx and Δy of the vehicle can be calculated:

$$\left. \begin{aligned} \Delta x &= w * [\cot(\Delta\phi) * (1 - \cos(\Delta\phi)) + \sin(\Delta\phi)] \\ \Delta y &= w * [\cot(\Delta\phi) * \sin(\Delta\phi) + (\cos(\Delta\phi) - 1)] \end{aligned} \right\} \quad (6)$$

These three motion parameters allow the world grid map to be updated. First, the translational movement of the vehicle is used to shift each grid in the world map, so that the position of the vehicle is again the center-front of the grid. Next, the world grid map is rotated to take into account the change in the heading of the vehicle. Each grid is rotated using a lookup table, which contain rotation information in 0.25 degree increments. The lookup table is created by considering the polar coordinates of each grid and subtracting the vehicle heading change from angular position of the grid.

5 Results

Shown in the following sequences are some experiments that demonstrate the efficacy of the dynamic grid method over the corresponding static grid method. *MOSFET* relies on using the forward-looking camera to detect obstacles. One of the main disadvantages of a (tilted) forward-looking camera is its limited field of view. The width of the field of view decreases as the distance toward the camera decreases. Additionally, to achieve the desired maximum viewing distance from the vehicle, an area very close to the vehicle where no visual information is available is created. Figure 11 is a world map sequence as *MOSFET* navigates towards a lane.

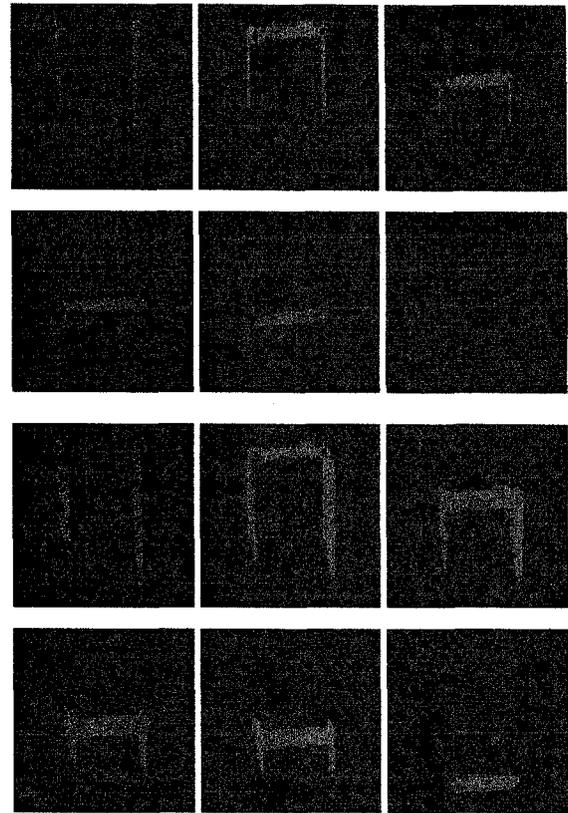
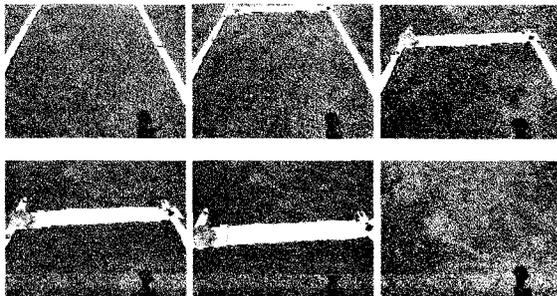
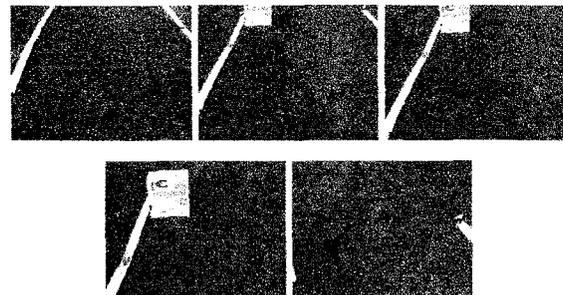


Figure 11: A sequence of front-looking camera images (bottom of left column), the corresponding static world maps (top), and the accumulated grid maps (bottom).

Notice how the accumulated grid model remembers lanes from the past even though those lanes are not currently in *MOSFET*'s current field of view.

Figure 12 is another image sequence as *MOSFET* navigates around an obstacle. Notice how the dynamic model retains obstacle information even after the obstacle leaves the field of view of the main camera.



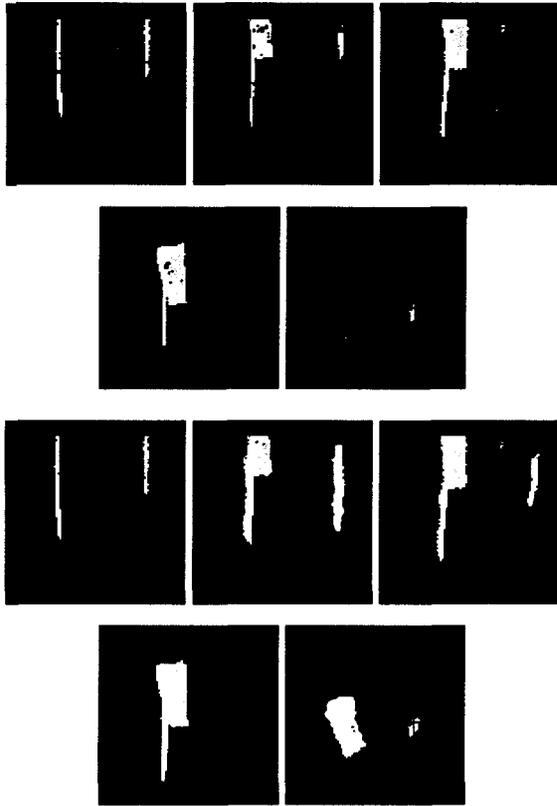


Figure 12: A second sequence of front-looking camera images (previous page bottom right column), the corresponding static world maps (top), and the accumulated grid maps (bottom).

6 Concluding Remarks

This paper presents a systematic procedure for heterogeneous sensor fusion and temporal information analysis. It is applied to create a dynamic map for our autonomous robot *MOSFET*. Lane and obstacle information is correlated to form a static grid of the world in front of *MOSFET*. This grid is then augmented by a lane/obstacle-tracking procedure that uses an accumulated grid model. Experimental results are shown to illustrate the efficacy and robustness of this procedure.

7 References

- [1] R. DeFauw, S. Lakshmanan, N. Narasimhamurthi, and M. Beauvais, "STARLITE: A Steering Autonomous Robot's Lane Investigation and Tracking Element," *Mobile Robots XI and Automated Vehicle Control Systems*, Proc. SPIE 2903, 1996.
- [2] J. Borenstein and Y. Koren, "Real-Time Obstacle Avoidance for Fast Mobile Robots", *IEEE*

Transactions on Systems, Man & Cybernetics, vol. 19, 1989.

[3] D. Langer & C. Thorpe, "Range based outdoor vehicle navigation, collision avoidance and parallel parking", *Autonomous Robots*, vol 2, 1995.

[4] C. Innocenti, et al., "Trajectory planning and real-time control of an autonomous mobile robot equipped with vision and ultrasonic sensors", *Proceedings of the IEEE/RSJ/GI International Conference on Intelligent Robots and Systems*, 1994.

[5] Jörg, Klaus-Werner, "World Modeling for an Autonomous Mobile Robot using Heterogeneous Sensor Information", *Robotics and Autonomous Systems*, May 1995.

[6] Slack, M, "Fixed Computation Real-Time Sonar Fusion for Local Navigation", *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2, 1993.