Sensor Fusion for Mobile Robot Navigation
—A First Subjective Discussion

OLLE WIJK, PATRIC JENSFELT AND BO WAHLBERG*

S3 – Automatic Control
Royal Institute of Technology
S-100 44 Stockholm, Sweden

March 27, 1997

Abstract

The aim of this report is to discuss the use of multi-sensor data fusion for mobile robot navigation. Smart sensory systems are crucial for successful autonomous systems. We will only give a short review of existing techniques, since there exist several recent thorough books and review paper on this subject. Instead we will focus on the main results, limitations and open problems with relevance to the intelligent service robot project at the Centre of Autonomous Systems (CAS) at KTH. We will also describe the on-going experimental work on the NOMAD platform on local navigation using ultra-sonar and infra-red range sensors. We will conclude by discussing some possible future extensions of the project.

1 Introduction

The intelligent service agent project at the Centre for Autonomous Systems at KTH is a mobile robot application, which should operate in unstructured normal indoor environments. The robot must be able to recognize certain objects as obstacles and landmarks. It must be able to navigate to locations within the environment. As few special modifications of the environment as possible should be assumed. In terms of obstacles, it must be able to handle both static and dynamic obstacles. The robot is supposed to determine its own position with respect to a previously defined map. If possible the robot

---

*This work has been funded by Centre of Autonomous Systems at KTH
should also return an estimate of the associated uncertainty. The map is expected to be sparse, i.e., it is not in general a complete CAD model, but rather a mixture of topology and geometry for natural landmark.

The system will be organized according to a four layer architecture, with layers related to hardware interfaces, reactive behavior, deliberative behaviors, and planning and control. The reactive control is driven by the perception, which is mapped to a local map. A potential field approach is used for local navigation.

A more short term objective of the intelligent service agent project is to provide functionality that allows semi-autonomous operation for wheelchairs. For example, the at first glance simple problem of driving through doorways, is not all an easy problem in practice. Robust sensory processing for control is very important in order to obtain a flexible solution, which does not depend on modification of the doorways.

In order to accomplish these tasks sensor systems will play a most critical role. The book by Everett, [1], is a very useful engineering reference book on how to use sensors for mobile robots. Different approaches using sensors for positioning of mobile platforms have recently been reviewed and evaluated in the excellent book by Borenstein, Everett and Feng, [2]. In this book methods for robot positioning are cataloged with respect to relative or absolute position measurements. Relative positions measurements can be further divided into:

- Odometry methods, which uses encoders to measure wheel rotation and/or steering orientation.
- Inertial navigation systems, which uses gyroscopes and accelerometers to measure rotation rates and acceleration.

These methods often provides good short time position estimates, but must be complemented with absolute position estimates to reduce off-set problems. Methods for measuring absolute position includes:

- Methods that uses active beacons to compute absolute position by e.g. triangularization techniques.
- Methods for recognizing artificial landmarks.
- Methods for recognizing natural landmarks.
- Methods based on maps and world models. The idea is to match sensor-based maps and world model maps.
We will more or less concentrate on the two last items above, since they form the basis for the CAS project. It should be noted that artificial landmarks provide a very powerful way to solve specific problems, and that there exist advanced industrial autonomous vehicles navigating using this principle. However, the objective of our research is to go beyond this assumption.

A short summary of the main observations of this report are:

- The field of sensor fusion is very wide and lack a consistent terminology. Many of approaches we have seen can be regarded as quite ad hoc or very theoretical, implying that work on scientific experimental validation of sensor fusion methods remains to be done.

- Sonar is the most studied and used sensor of today. Despite this fact a lot of work remains, e.g. using active arrays of sonars. Having good sensor models is very important to interpret measured data in a correct way.

- Grid based techniques, have proven to be a useful setting for sensor fusion, localization, obstacle avoidance and navigation. We have done a basic experimental evaluation of this techniques with focus on semi-autonomous wheel chairs.

This report is organized as follows. We will start by introducing the basic principles of data fusion in robotics. We will then describe some existing mobile platform from a sensory processing perspective. The concept of using directed sonar sensing for navigation will be discussed in more detail, and we will concentrate on grid based methods. Finally, we will describe our ongoing work. To conclude we will point out the pros and cons of existing methods, together with an open problem discussion.

2 Data Fusion

Data fusion is about deriving information about certain variables from observations of other variables. The application area is huge, see the special issue on data fusion in [3] for a recent overview. An edited collection of survey papers on data fusion in robotics and machine Intelligence is given in [4]. Sensor fusion in general is discussed in [5, 6].

From a statistical perspective, we have the following problem. Given two vector random variables \( X \) and \( Y \), what does the observation \( Y = y \)
tell us about $X$? The complete answer is given by the so-called conditional probability density function,

$$p_{X|Y}(x|y) = \frac{p_{X,Y}(x,y)}{p_Y(y)} \quad (1)$$

Here $p_{X,Y}(x,y)$ is the joint probability density for $X$ and $Y$, and $p_Y(y)$ is the probability density for $Y$. A more intuitive explanation is as follows. It seems reasonable that the joint probability of $X$ and $Y$ should equal the probability of $Y$ times the conditional probability of $X$ given $Y$, i.e.,

$$p_{X,Y}(x,y) = p_{X|Y}(x|y)p_Y(y). \quad (2)$$

By dividing both sides with $p_Y(y)$, we obtain the definition (1). Hence, the conditional probability density function of contains the probabilistic description of the $X$ given $Y = y$.

By using the dual assumption, namely that $X = x$ is given, we obtained the very useful Bayes rule

$$p_{X,Y}(x,y) = p_{X|Y}(x|y)p_Y(y) = p_{Y|X}(y|x)p_X(x) \Rightarrow \quad (3)$$

$$p_{X|Y}(x|y) = \frac{p_{Y|X}(y|x)p_X(x)}{p_Y(y)}, \quad (4)$$

which is the key formula in Bayesian and maximum likelihood estimation theory.

Different estimates of $X$ can now be constructed from its distribution. The (conditional) minimal variance estimate of $X$ equals the conditional mean of $X$ given $Y = y$,

$$\hat{x} = E[X|Y = y] = \int_{-\infty}^{\infty} xp_{X|Y}(x|y)dx. \quad (5)$$

Another useful estimate is the maximum a posteriori estimate, which maximizes the function $p_{X|Y}(x|y)$. This more or less summarizes estimation theory! The rest is design and analysis issues, i.e., formulating the underlying model, specifying probability density functions and calculating quality/variance properties.

The most used probability density function is the Gaussian one (the Normal distribution). The main reason is that the conditional density function also will be Gaussian, and analytic expressions of the minimal variance estimate can thus be obtained. The following example illustrates this fact.
Let $X$ and $Y$ be jointly Gaussian, i.e. $Z = [X' \ Y']$ is Gaussian with mean and covariance

$$m_z = \begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix}, \quad \Sigma_{zz} = \begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{bmatrix}. \quad (6)$$

Then $X$ conditional on $Y = y$ has a Gaussian distribution with mean and covariance

$$m_{x|y} = \bar{x} + \Sigma_{xy} \Sigma_{yy}^{-1} (y - \bar{y}), \quad \Sigma_{x|y} = \Sigma_{xx} - \Sigma_{xy} \Sigma_{yy}^{-1} \Sigma_{yx}. \quad (7)$$

Hence the conditional mean of $X$ given $Y = y$, equals

$$\hat{x} = E[X|Y = y] = \bar{x} + \Sigma_{xy} \Sigma_{yy}^{-1} (y - \bar{y}). \quad (8)$$

Almost all practical estimators are special cases of the above result. The expression is called the fundamental equations of linear estimation in [7]. This reference also provides a very good introduction to estimation theory, in general, and tracking, in particular.

Two of the most important special cases of (8) are given next.

### 2.1 The Kalman Filter

Let $X$ be the state vector of a dynamical system to be estimated and let $Y$ be the observation process. Furthermore, assume Gaussian distributed noise.

The Kalman filter recursion can then be derived by applying the conditional mean estimator (8). First, the so-called measurement update equation is obtained by letting $X$ be the states at the measurement time. The so-called time-update equation is then used to predict the state one step ahead. This predicted state will be used in the next measurement update. This leads to the Kalman filter recursion. We have chosen to not give the corresponding equations here. Instead we refer to Section 3.1 in [8] for details.

Notice that the Kalman filter is based on linearity, and can thus only handle linear models.

The Kalman filter is a key tool for data fusion. The idea is just to include all different sensor outputs in the observation vector $Y$. Corresponding sensor dynamics have to be included in an extended state vector $X$, and the quality of the different sensors are described by the corresponding noise covariance matrices.

---

1The notation $'$ is used for transpose.
2.2 The Fusion Filter

Given two estimators $\hat{\Theta}_1$ and $\hat{\Theta}_2$ of $\Theta$, and the task is to fuse them together to form one single “optimal” estimate $\hat{\theta}$.

Assume that $\hat{\Theta}_1 - \Theta$ and $\hat{\Theta}_2 - \Theta$ are independent Gaussian distributed with zero mean and covariances $P_1$ and $P_2$, respectively. Now let $X = \Theta - \hat{\Theta}_1$ and $Y = \hat{\Theta}_2 - \hat{\Theta}_1$. Then $\Sigma_{xy} = P_1$ and $\Sigma_{yy} = P_1 + P_2$. Hence

$$\begin{align*}
\dot{x} &= P_1(P_1 + P_2)^{-1}(\dot{x}_2 - \dot{x}_1) \quad \Rightarrow \\
\dot{\theta} &= \dot{\theta}_1 + P_1(P_1 + P_2)^{-1}(\dot{\theta}_2 - \dot{\theta}_1) \quad \Rightarrow \\
\dot{\hat{\theta}} &= [P_1^{-1} + P_2^{-1}]^{-1}[P_1^{-1}\dot{\theta}_1 + P_2^{-1}\dot{\theta}_2],
\end{align*}$$

with covariance

$$P_1 - P_1(P_1 + P_2)^{-1}P_1 = [P_1^{-1} + P_2^{-1}]^{-1}. \quad (12)$$

The fusion formula just means that estimates should be weighted together, with weights inversely proportional to their qualities/variances. It is easy to modify the fusion filter to handle correlated estimators.

We have concentrated on quadratic norms, which follows from a Gaussian assumption. However, the sensor noise may have very different characteristics, including existence of so-called outliers.

Data fusion techniques using more robust statistics is described in [4, chapter 5]. Another paradigm is to use a less statistical approach only based bounds of the noise. Bounding approaches for estimation has recently been reviewed in [9]. Reliable sensor models are of utmost importance in model based signal processing. This will be further discussed below.

2.3 Extensions

Realistic dynamic models for robots are often nonlinear. An approximate description of the dynamical systems is obtained by linearizing the nonlinear equations. Using a linearized description leads to the so-called extended Kalman filter. This approximation is only valid for small deviations from the trajectory, which the system is linearized around.

In mobile robotics one often has what is denoted internal states (for the robot) and external states (for the environment, i.e. perception). These can

---

$^2$Notice that estimators are stochastic variables and are denoted by capital letters.
of course be lumped together to one large state space model, but special care must be taken in practise to avoid a complexity explosion. There are still many open research issues on how this should be done.

A similar complexity problem arises when one has several alternative models. A common situation is that one models the environment using a fixed number of geometric objectives (straight lines, corners, etc.) This leads to hypothesis testing algorithms in order to determine which model is valid for a given time period. The optimal solution to this problem is to use banks of Kalman filters (one Kalman filter for each hypothesis) to generate the statistics, which then are used in likelihood ratio tests to decide which model is most probable. The complexity increases exponentially in the number of different models, and approximate methods has to be used in practise. Such techniques are well developed in the field of radar tracking, [7]. The idea of using such techniques in mobile navigation has been studied in great detail, [4, chapter 7][10]. We will discuss some of this work below.

3 System Identification

The idea of this section is to provide a discussion on a more scientific and systematic approach to experimental modeling.

System identification concerns the problem of constructing models of dynamical systems based on observed data. The purpose of the model is often control, but the techniques developed in this field are of much more generality. Sweden has a very strong competence in system identification. The two Swedish books [11] and [12] are the main references in this area. The Matlab software tool “the System Identification Toolbox” (SITB), developed at Linköping University, provides excellent means for rapid application of advanced system identification methods. However, most of methods are for linear so-called black-box models. Research on so-called ”grey-box” identification has been pioneered by Torsten Bohlin at KTH, see [13] The objective is to develop a paradigm for calibrating and validating nonlinear dynamical process models with partially known but uncertain structure, and in the presence of random disturbances from the environment. A software package IdKat has been constructed and applied successfully in a number of industrial applications.

Important tasks in interactive system identification are

- Specification of a model structure,
- Specification of a free parameter set, including scaling factors, search
constraints and initial guesses.

- Methods for determining the likelihood
- Numerical algorithms for optimizing the likelihood
- Statistical tests for falsification (validation)

The procedure for “grey-box identification” is now as follows

```plaintext
While the structure is falsified, repeat
    Refine the model structure
    Fit model parameters

Falsify model: Until falsified, repeat
    Specify alternative model structure
    If any alternative model is significantly better,
    then indicate falsified
```

Important tools are advanced model generators which produce predictors, efficient computation of the loss function, including its gradients, and powerful hypothesis tests. Model validation is the key issue! A Bayesian probabilistic framework is used for likelihood determination and hypothesis testing. From a modelers point of view we have:

- **Size**: The risk of rejecting a correct model (false alarm:) = The Model Builders Risk

- **Power**: The ability of rejecting a false model (discrimination efficiency) = The Model Users Risk

The most powerful test maximizes the power for a given size, and is the Likelihood Ratio Test. The key problem in the validation step is to define the alternative hypothesis, which corresponds to conditional falsification. In practice, unconditional tests, which have no alternative hypothesis, are most often used in tools like the SITB. There are two schools for empirical data modeling:

- **The scientist’s rule**: Proceed until the model explains the data
- **The engineer’s rule**: Proceed until the model satisfies a given purpose.

The purpose of the model can be included in Bayesian estimation, in a quite abstract way. Let us define that a model satisfies the purpose if $model \in \mathcal{G}(system)$. Here $\mathcal{G}(system)$ is the set of models which satisfy the purpose. This set of course depends on the true system.

A control example: A purposive model is one which gives a controller which stabilizes the system.

The probability that the model satisfies the purpose can now formally be calculating by integrating the conditional probability of the system given the data over the set of purposive models. Hence we can determine the conditional probability given the data that an approximate model satisfies the purpose. The difficulty of this approach is of course how to specify the set $\mathcal{G}(system)$.

4 Sonar Sensing for Mobile Robot Navigation

The bats excellent ultrasonic system in terms of interferences and physical conditions has for a long time inspired researchers to develop technical sonar systems. However, there is a huge step left before we are even close to bats, see the recent PhD thesis [14] for ideas how to improve sonar systems.

Ultrasound based techniques (sonar) are probably the most studied sensor approach for “experimental” mobile navigation. One of the main reasons being the low price! The Polaroid ultrasonic ranging system is dominating in most experimental mobile platforms. It should be noted that this system is quite restricted in terms of flexibility to change systems parameters, and that several improvements can be done using a more open sonar systems. In current systems only one sonar at the time can transmit and receive. An interesting approach to investigate would be the use of arrays of sonar sensor, where actually all sensors simultaneously transmit and receive the echoes. There has recently been substantial progress in statistical methods for sensor array processing.

There are basically two different ways to use sonar in mobile robot applications, namely:

1. Sonar-based methods for obstacle avoidance.

The purpose determines which method to use. Grid based techniques, described in Section ?, are typically used for obstacle avoidance, and use very crude models of the environment. There is a dispute on how much world modeling and how accurate maps are really needed for navigation. Leonard, Durrant-Whyte and co-workers argue that detailed and reliable maps are necessary for more precise dynamic localization. The problem is even more complex if one does simultaneous map building and localization. However, from a scientific systems engineering perspective the work of Leonard, Durrant-Whyte and co-workers provides a serious effort and investigation on how to overcome problems with sonars by using advanced estimation and signal processing techniques. Complex models together with Kalman filtering and detection algorithms are extensively used in their work. The concept of directed sensing strategies is used to reduce complexity. The key results of this work are:

- Advanced sensor models, which can predict and explain real sonar data, are extremely important for success. Too simple models and “vision based” evaluation have contributed to a “bad reputation” of sonars. Instead both the sonar and the targets should be carefully modeled, and acoustic phenomena should be taken into account.

- An approach to model based localization has been developed. The work includes techniques for efficient computation of the correspondence between new sensor readings and a geometric world model. Extensive tests with the developed techniques has demonstrated the utility of this approach for in-door navigation.

- A unified approach to navigation using simultaneous map building, obstacle detection and localization, based on a multi-target tracking framework, has been derived.

The book [10] provides an thorough overview on the state-of-art up to 1992. However, there exists many challenging problems to be solved on this approach before one can handle real dynamic environments in real time. More recent work following this line of research is discussed [2, p. 169].

5 Grid Methods

When the mission of a robot is to autonomously explore its environment several natural problems arise, e.g.:

- How should the robot fuse the information, which it experience from its different sensors (i.e. sonars, IR, vision etc)?
• How should the robot handle position uncertainty?

• How should the robot plan its trajectory between its current position and a goal point?

• How can the robot accomplish safe obstacle avoidance?

One approach to handle these problems, is use the Occupancy Grid method.

5.1 Occupancy Grids

Occupancy Grids is certainly a state of the art method in the field of grid based methods. The method was originally developed by Alberto Elfes [4, 15], and is briefly explained here. The idea is to divide the environment into grid cells $C_{ij}$. Typically a 2-dimensional grid is enough to give interesting information about the environment. Each cell can be in two states $s(C_{ij}) = \text{OCCupied}$ or $s(C_{ij}) = \text{EMPty}$, and to each cell there is a probability $P \{ s(C_{ij}) = \text{OCC} \}$ attached, which reflects the belief of the cell $C_{ij}$ being occupied by an object. Since

$$P \{ s(C_{ij}) = \text{EMP} \} = 1 - P \{ s(C_{ij}) = \text{OCC} \}$$

the grid is initialized with $P \{ s(C_{ij}) = \text{OCC} \} = 1/2$. To update the cells when the robot traverses the environment, a stochastic sensor model $p(r \mid z)$ is used. This model is obtained from experiments with the sensor in question and relates a range reading vector, $r$, to the true space range vector, $z$. Given a new range reading, $r$, from a sensor, the idea now is to use the sensor model $p(r \mid z)$ to update the probabilities $P \{ s(C_{ij}) = \text{OCC} \}$ in the Occupancy Grid. This can be done by using Bayes’ theorem, see Equation (4),

$$P \{ s(C_{ij}) = \text{OCC} \mid r \} = \frac{p(r \mid s(C_{ij}) = \text{OCC}) P \{ s(C_{ij}) = \text{OCC} \}}{\sum_{s(C_{ij})} p(r \mid s(C_{ij})) P \{ s(C_{ij}) \}}.$$ (14)

We mention here that the right side of Equation (14) has to be developed further to be computable. Exactly how this is done can be found in [4, 16].

The Occupancy grid method now provides a useful setting for fusing data from different sensors. There are, basically, two main approaches:

• Let the different sensors, each having its own stochastic sensor model $p(r \mid z)$, update the same robot Occupancy Grid. This can be done by using a formula similar to (14).

• Let each sensor type have its own Occupancy Grid map. Each time the maps are updated they are fused together to update a “fused robot map”. Here, again Bayesian theory can be used.
For further comments on fusing data using Occupancy grids we refer to [4].

Considering handling position uncertainty of the robot one can have a global grid map of the environment stored on the robot platform. The fused robot map can then be matched against the global map to reduce uncertainty in position. Work in this field can be found in [17].

Since obstacles are represented by high cell probabilities, the Occupancy Grid is well suited for doing path planning and obstacle avoidance. However, one drawback is that the obstacles tend to be larger than they really are, because of high cell probabilities around the object in question. Of course the physical cell size, which due to computational complexity usually is about 10 cm, sets a limit on the resolution of the objects in the map.

One major drawback with using Occupancy Grids as described above is that when updating the grid, i.e., evaluating Equation (14) for each cell $C_{ij}$, the computational cost is high even for rather small grids. This implies that algorithms that use Occupancy grids for navigation are rather slow. One way of getting around this problem is to use Vector Field Histogram methods, which is described in the next section.

### 5.2 Vector Field Histogram

The Vector Field Histogram is a way of handling fast map building and obstacle avoidance at the same time. The method was originally introduced by Borenstein and Koren [18, 19, 20] and is today used extensively as a method for fast navigation and obstacle avoidance. The base for the method is formed by a robot fixed certainty grid similar to the grid type described in the previous section. However, to obtain a fast map update, the formula (14) is not used since it projects a probability profile onto all those cells affected by a range reading. Instead each cell have an associated certainty value $c_{ij}$ of integer type which reflects the belief of the cell being occupied. The higher (lower) value of $c_{ij}$, the more confidence we have that the cell $C_{ij}$ being occupied (empty). Upon a new range reading, usually only the cells along the line of sight of the sensor are updated. Cells near the range reading are incremented by some appropriate integers, while the cells in between the sensor and the range reading are decremented by some other integers. The question of how much the cells should be incremented or decremented depends on which type of sensors is used. So again it is important to have good sensor models in order to obtain good results. To prevent the integers $c_{ij}$ to grow or decrease too much, they are saturated at some maximum and minimum integer.

While this approach of updating the grid may seem to be an oversimplification compared to using Equation (14), a “probability” distribution is
actually obtained by continuously and rapidly sampling each sensor while the robot is moving. When doing obstacle avoidance a vector field histogram is created from the above described grid. First the grid is divided into, say, \( n \) polar sectors \( S_1, S_2, \ldots, S_n \) (see Figure 1). For each cell \( C_{ij} \) in a given sector, say \( S_k \), one calculate an obstacle vector \( m_{ij} \). The magnitude of \( m_{ij} \) is dependent of the certainty value \( c_{ij} \), and also by the distance between the center of the grid (robot position) and the cell \( C_{ij} \). After this procedure one sum all the obstacle vectors \( m_{ij} \)'s in sector \( S_k \) to form a obstacle density entity \( h_k \)

\[
h_k = \sum_{c_{ij} \in S_k} m_{ij}, \quad k = 1, \ldots, n.
\]  

At this point the entities \( h_1, h_2, \ldots, h_n \) are used to form a histogram, which can be used for obstacle avoidance (Figure 2). Areas in the histogram where the vector magnitudes are big indicates regions with high obstacle density, while areas with low vector magnitudes indicate regions with low obstacle density. By adapting a threshold to the histogram it is possible to localize regions of sectors with low obstacle distance which can be used for obstacle avoidance.

![Figure 1: Robot fixed certainty grid with certainty values \( c_{ij} \). The map is divided into polar sectors.](image1)

![Figure 2: Histogram of the obstacle density. By applying thresholding one can determine sector regions where it is possible to do obstacle avoidance.](image2)
6 Our Activity on the Nomad Platform

In the intelligent service agent project the platforms used to conduct real experiments are a Nomad robot from Nomadic Technologies, Inc and two Denning robots. The Nomad is equipped with sonar, IR, compass and a camera. Further more, rate gyros and GPS have been discussed as well. The Denning robots are equipped with sonars and two cameras. Using real platforms instead of simulations is a major objective in this project; if it does not work in the real (Lab) world, it is useless in practise. Adaptation to changes in the environment is another major objective.

6.1 Sonar and IR

The Nomad platform is, as mentioned above, equipped with both sonar and IR. While sonars cover a wide range of ranges, from a few decimeters up to several meters, IR work only up to approximately a meter. The sensor rings are placed at different heights; the sonar ring close to a meter above the floor and the IR a couple of decimeters above the floor. Both sensor rings, sonar and IR, have the same number of sensor, placed at the same angle with respect to some fixed point.

6.2 Understanding the Sonars

In order to get a good understanding for the sonars, a lot of experiments have been performed. These experiments will make it easier to interpret and use the data that they deliver. The ideal situation would be if one could see what the sonars saw, which of course is impossible, but studying the data that they deliver is a step in the right direction. To quote Leonard and Durrant-Whyte,[10]: One should not try to look for the world as we see it in the sonar data, but instead try to see the world that the sonars see. This indicates that understanding the sonar physics is important.

<table>
<thead>
<tr>
<th>Material</th>
<th>Active beamwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lab wall (rough)</td>
<td>100-130°</td>
</tr>
<tr>
<td>Smooth door</td>
<td>20-25°</td>
</tr>
<tr>
<td>Polaroid spec. sheet</td>
<td>25°</td>
</tr>
</tbody>
</table>

Table 1: The active beamwidth of the sonar when reflected by different materials compared with the specification from Polaroid.

Having performed these experiments some conclusions can immediately be drawn. The sonar response is very much dependent on the material that
interacts with the sonar beam. Table 1 means to illustrate the problem that one faces when trying to interpret the data that the sonars deliver. For a correct interpretation one needs to know what material the object that was detected was made of.

6.3 A Direct Approach

An intuitive approach to use the sonar and IR data is purely reactive control. In our first approach no model is incorporated and the fusion of the data from the sonar and the IR is very simple. We fused the sonar and IR data in such a way that the sensor that were registering the shortest range reading in each direction was used. The idea behind picking the smallest value was to set safety first, i.e., if one would have used some kind of weighted average, a false reading from one of the sensors might have resulted in the robot running into something.

To make the robot avoid running into things we defined applied virtual repulsive forces to the robot that were basically the inverse of the range reading in each direction. This means that the robot will experience a strong repulsive force when it is close to something, which is natural.

In most situations this simple approach actually worked rather well, but the system was not reliable. The major problems that we encountered were:

- The response of the sensors depend heavily on the material in the environment, which was not taken into account at all.

- No way of filtering out false readings.

- The lack of memory means that all objects in the robots way has to be seen at all time to be avoided safely. This is impossible since a reflection is obtained only if the surface of the object fulfills certain criteria concerning its angle towards the sensor. This showed itself mainly when passing through a doorway.

- Occasional oscillative behavior in a narrow corridor, due to the fact that the repulsive forces in the corridor vary much over a short distance.

6.4 Building a local map

Building a robust and reliable avoid behavior has been found to require some kind of memory. Inspired by the work of Borenstein and Koren [21, 20] we have implemented a grid based local map for the robot. So far this map has been updated using only the sonar data. At this early stage we have
been using a ray-trace model for the sonar, which is justified by the motto, try simple first and supported by [21]. The results of these tests show that the avoid behavior is improved. Below (Figure 3) is a sketch of the CVAP-building in the corridor outside the robot lab. To show what the local maps look like, four samples of such maps are shown in Figures 4-7. The size of the cells in these maps were 20x20 mm and the number of cells were 200x200, giving a total size of 4x4m. Note that the coordinate system of the local maps are robot centered. The approximate location of the robot when the maps were saved is given in the sketch (Figure 3) by the letters A-D.

![Figure 3: A sketch of the environment around the robot lab in the CVAP-building.](image)
Figure 4: Sonar based local map of the corridor in the CVAP-building with one closed door and one open. A in the sketch.

Figure 5: Sonar based local map of the door-passage into room 213 in the CVAP-building. B in the sketch.

Figure 6: Sonar based local map of the corridor outside the lab and room 213 in the CVAP-building. C in the sketch.

Figure 7: Sonar based local map of one of the graduate rooms in the CVAP-building. D in the sketch.

If the local map is extended to a size that can hold much more information the figure below (Figure 8) show a possible result. The intention of the local map is not to be this large, but rather to have a size more like the once shown.
above. The global map is updated based on the local maps, i.e., the sensor data is not directly used in the global map.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{figure8}
\caption{Sonar based map of the corridor in the CVAP-building.}
\end{figure}

\section{Motor Powered Wheel-chair}

As an intermediate step towards having a complete service agent robot that can serve dinner, do the dishes and collect the coffee mug you forgot in the other room, we will try to develop a sonar and IR based system for motor powered wheel-chairs. The short terms objectives of the service agent robot fits very well with the needs for a semi autonomous wheel-chair. A motor powered wheel-chair is often driven by a person that has severe motor disabilities.

As of now, the steer input from the driver is passed through a low-pass filter before it is sent to the motors. To aid these people further, it would be of interest to let the wheel-chair have some kind of controller that can do more than just low-pass filter the input signal. The objectives of the intelligent service agent fits very well with the needs for a semi autonomous robot. The driver of the wheel-chair is still the one who decides where to go, but the wheel-chair has some kind of controller that makes sure that the wheel-chair does not run into things or get stuck in doorways etc. In this case the global path-planning is made by the driver of the wheel-chair. This task has to be performed by some high-level controller in the intelligent service agent.
Further into the future the wheel-chair might have the capability to navigate completely on its own, which also fits Making a robot that can navigate using a predefined map and localize itself if it gets lost, which also falls within the objectives of the service agent project. The map should not be, as mentioned in the introduction, a complete CAD model, but rather a sparse map, preferably made by the robot itself during a tour of its surroundings in the beginning of its service at a new place.

The wheel-chair project will act as a motivation for our efforts towards mastering the sensors and the techniques for handling the data, not to mention that it will hopefully help the drivers of the wheel-chairs of the future.

7.1 Current Research on Semi-Autonomous Wheel-Chairs

Some research has already been done in this area by e.g., Borenstein et al [22]. They have been working with a project that they call the NavChair. The objective of this project is the same as ours, i.e., to develop a system that can assist the disabled people in their daily life by helping them in their driving. The person in the wheel-chair is the one who gives the commands using a joystick and the task of the wheel-chair controller is to make sure that the chair does not run into things. The NavChair is a motor powered wheel-chair equipped with Polariod ultrasonic range detectors. The Vector Field Histogram approach is used for the navigation. Some of the problems that are mentioned are that dead-reckoning on a wheel-chair is much more difficult than on a normal test-platform and that using only one controller for all necessary modes (wall-follow, travel in a hallway, door-passage etc.) is hard. Their wheel-chair could handle the hallway very well but got stuck when passing through doors far too often.
Drawbacks with the Vector Field Histogram approach [21]:

- Since the potential field is based on a discrete grid there are bound to be some problems when the robot moves in the gridmap; the forces due to the potential field will change very rapidly from one grid to another. This result in fluctuations in the steer input.

- Traveling in narrow corridors can result in an oscillative motion, as if the chair was bouncing between the walls, due to the large forces that are encountered close to the walls.

- Passing through doorways is difficult since the repulsive forces becomes very large that close to an obstacle (the door frame).

Another team of researchers [23] have been looking at a fully autonomous system where the person in the wheel-chair only has to select a destination from a menu. In contrast to the objectives in the intelligent service agent project they have been using a beacon system for their navigation. Even though it is a passive system with elliptical patterns on the walls, it is without question a case of environment engineering.

8 Fusion of Sonar and..

As mentioned earlier, sensor fusion is the process of combining the information from different sources of sensory information. As with us humans, robots cannot rely on one source of information, since it might not always give the correct description of the world. Vision will no contribute with anything in the dark and sonar will be close to useless in a heavily cluttered environment. Therefore combining different sensors give a much more robust and flexible system and is a prerequisite for a system like the intelligent service agent.

The sources of sensory information do no have to be different types of sensors, it could equally well be two sensors of the same kind. Chang and Song [24] looked at the problem of getting good estimates of the walls of a room using multiple ultrasonic sensors. They use two approaches, one using an artificial neural network (ANN) and another that uses a mathematical model.

Combining different kinds of sensory information has also been looked at, e.g., by Jörg [25]. Jörg looked at the problem of combining the information from 24 sonar sensors and a laser radar for real-time collision avoidance and local path planning. The laser radar provides an excellent angular resolution and accuracy, but will fail to detect certain objects. The sonars has as mentioned before a fairly good accuracy in the radial direction but suffers
heavily from the large beam-width. The idea behind Jörgs method is to try to match the laser radar measurement with the sonar and thereby filter out most of the specular reflections and crosstalk in the sonar data. The base for the fusion is a grid map that is updated with the sensor data.

As an example of fusion at feature level we mention Courtney and Jain [26]. As sensors they used sonars, vision and IR. They manage to do coarse localization, i.e. classify rooms and doorways in an office building, by extracting spatial descriptions from certainty grid maps similar to the type described in 5.2. Each sensor type used their own grid map from which certain moment descriptors (independent of translation, rotation and scale) were calculated. Data from the different rooms and doorways, was collected and stored in terms of the moment descriptors in a database, which could be used when the mobile robot needed to localize itself. In the experiments they used both 3-nearest-neighbor (3-NN) and minimum Mahalanobis distance (MMD) classifiers. The MMD classifier showed slightly better performance than the 3-NN classifier concerning room recognition, while it was the other way around considering doorways. The reported recognition rate was 98% for rooms and 94% for doorways.

9 Conclusion

We have discussed two more or less orthogonal approaches for using sonar in mobile navigation. From a scientific/academic perspective it is important to study very general issues and approaches, were the ultimate aim is full autonomy. However, the engineering perspective is the opposite, i.e. one wants to solve a specific problem, e.g. a sonar sensor based feedback control algorithm for going through narrow doorways. However, the main issue for such research is scalability, i.e. is the solution of more general interest and can it be extended to more complex situations!

Concerning our ongoing research work, the Nomad platform has been used as test bed for investigating semi autonomous control. The objective is to implement simple object avoidance, wall (corridor) following and door passing behaviors, which will act in symbiosis with a human control (joystick). To achieve this goal we have implemented a robot fixed grid map which is very similar to the Vector Field Grid approach [20]. Tests carried out at the CVAP department have been promising and we expect to having a demo working in the mid of April 1997.

We believe that a study of semi autonomy is an appropriate first step towards developing a fully autonomous system. The semi autonomous approach allows sidestepping of some of the most difficult problems with full
autonomy, e.g., global localization and path-planning.

Our next research projects on sensor fusion includes:

- We will continue using the sonar as an experimental platform for system identification approach for validation of sensor models. More advanced sonar concepts, e.g. arrays, will be studied, to improve the spatial resolution.

- We will extend our experimental sensor system with more sensors that sonar.

- We will study the interaction between control and the grid based sensory processing methods discussed above. In particular, an robustness analysis will be done, studying the effects of modeling errors.

- We will study micro-wave radar as a sensor for mobile navigation in connection with our industrial demonstrator project.

- We will study identification of acoustic impedance. The sonar has high accuracy in measuring the range, but the spatial resolution is quite poor. The response from the environment depends very much on the material that reflects the ultrasonic energy.

So far robotics in the industry has been a synonym with the robots that are used in e.g., the car industry doing welding and other very specific tasks. Robots are perfect for these applications since they have the endurance and precision of a machine. But this is only the beginning. If mastered, the number of applications for autonomous robots will be endless. This implies that the industry would be very interested in results that would bring them closer to the point where they can make the kind of robots they are dreaming about. So far we have seen several interesting project with autonomous robots, e.g., autonomous lawn mowers and vacuum cleaners.

Given the current exponential increase in computer power, we see that the development of smart sensor systems as one the main issues for more intelligent “machine” systems.

References


