Smart Sensor Snow

Thomas C. Henderson, Mohamed Dekhil, Scott Morris, and William B. Thompson

UUSC-98-003

Department of Computer Science
University of Utah
Salt Lake City, UT 84112 USA

February 1998

Abstract

We propose to deploy and exploit a large number of inexpensive sensors to obtain information or trigger actions over a wide geographic area. Sensors may be of diverse physical natures: acoustic, IR, seismic, chemical, magnetic, thermal, etc. We describe here three major issues: (1) sensor distribution patterns, (2) local sensor frames, and (3) autonomous robot sensor snow exploitation techniques.

This work was supported in part by the Advanced Research Projects agency under Army Research Office grants number DAAH04-93-G-0420 and by NSF grant CDA 9024721.

1 Introduction

We have been working on the specification and analysis of robot behaviors for quite some time[1, 2], and more recently we have examined the issue of sensor fusion for large numbers of distributed sensors (or networked sensors) [3].

In this paper we present an approach to exploiting a large number of spatially distributed sensors. This involves the capability of identifying each sensor, determining distances between them, and having them report various sensed information. This can then be exploited to specify application-specific spatial patterns in the sensor set, determine local or global coordinate frames, and to use those to support mobile robot behaviors for robots that communicate with the sensors.

2 Pattern Distributions

When given a rather dense set of distributed sensors, it is sometimes useful to assign a pattern to them automatically and by means of a distributed algorithm. For example, given a 200 by 200 grid of sensors, it might be more useful to group them into larger units (like the sectors shown in Figure 1). What is interesting is that this very sharp box pattern appears globally after running the same reaction-diffusion equation locally in each sensor for a given amount of time. If a smaller set of randomly (uniformly) distributed sensors were used instead, the patterns shown in Figures 2 and 3 are typical of the set of sensors with bits set to 1 and zero, respectively (we have used 200 sensors here out of the 200 by 200 grid). Alternatively, striped patterns (see Figure 4) might be useful in another situation. Less regular figures may be useful in other cases.

We propose to use Turing's reaction-diffusion mechanism[5] to generate such patterns in smart sensor snow. The basis of this mechanism is a set of equations that capture the reaction and diffusion aspects of certain chemical kinetics:

$$\frac{\partial \mathbf{c}}{\partial t} = f(\mathbf{c}) + D\nabla^2\mathbf{c} \tag{1}$$

where $f(\mathbf{c})$ describes the reaction and $D\nabla^2 c$ expresses the diffusion component. The simplest such systems have two *morphogens* or variables; one of these acts as the activator and the other acts as the inhibitor. The two variable system can be modeled by:

$$\frac{\partial u}{\partial t} = \gamma f(u,v) + \nabla^2 u, \frac{\partial v}{\partial t} = \gamma g(u,v) + d\nabla^2 u \tag{2}$$

where u and v are the concentrations of the morphogens, d is the diffusion coefficient and γ is a constant measure of scale. The functions $f(u,v)$ and $g(u,v)$ represent the reaction kinetics. As an

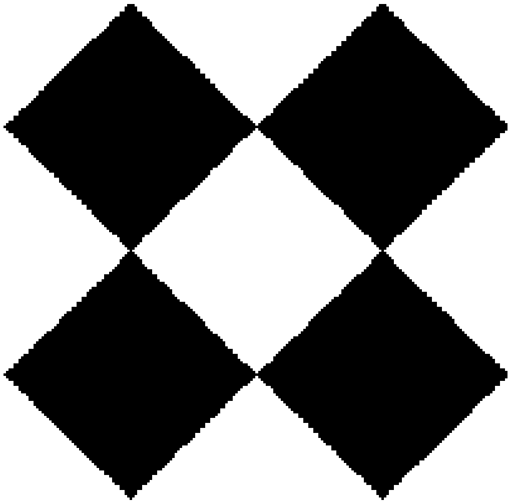

Figure 1: Box Pattern

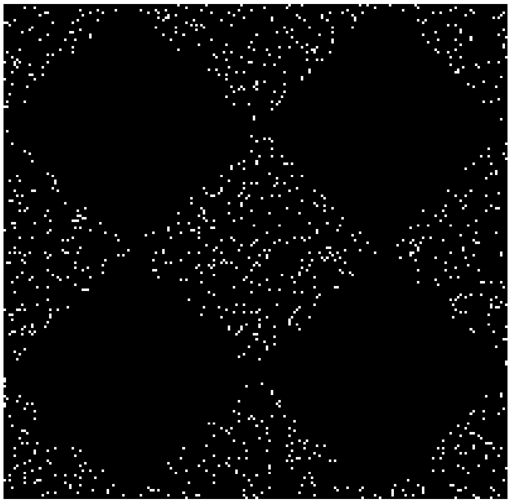

Figure 2: Sensors with Bit greater than zero.

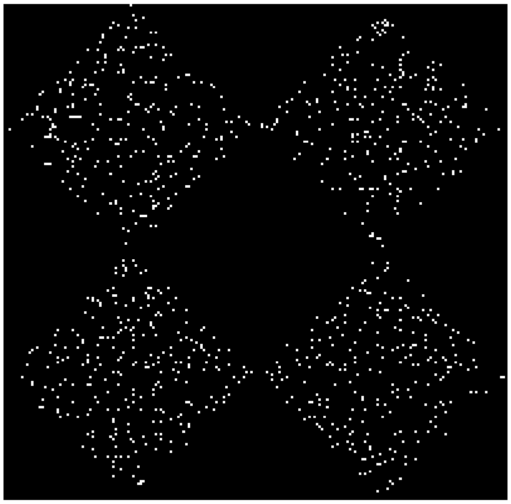

Figure 3: Sensors with Bit less than zero.

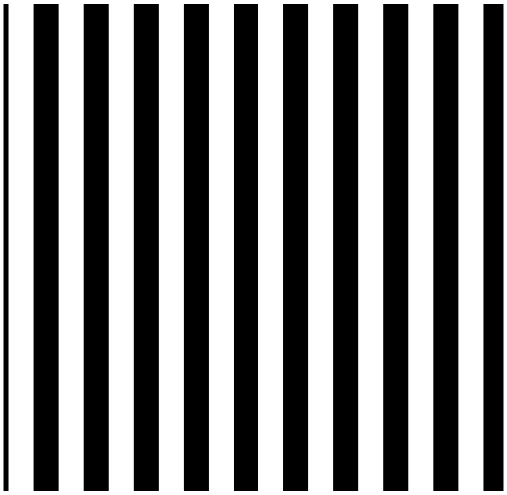

Figure 4: Stripe Pattern

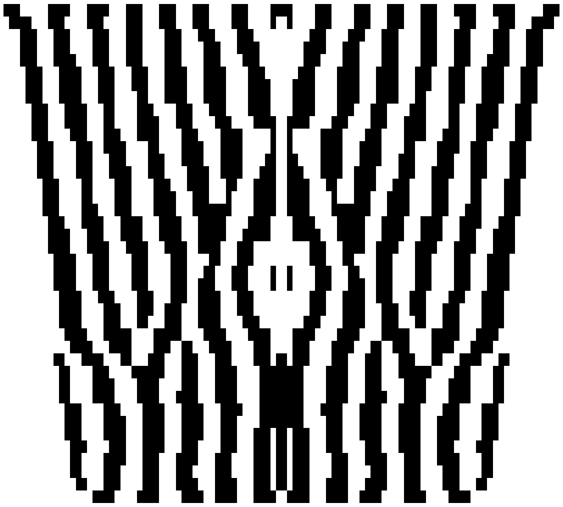

Figure 5: Irregular Pattern

example, we have explored the generation of spatial patterns using the Thomas system of equations (Murray[4] describes this system in detail):

$$f(u,v) = a - u - h(u,v), g(u,v) = \alpha(b - v) - h(u,v)$$
$$h(u,v) = \frac{\rho uv}{1 + u + Ku^2} \tag{3}$$

where a, b, α, ρ, and K are the positive reaction parameters. They define a domain in which Equations 3 become linearly unstable to certain spatial disturbances. This domain is referred to as *Turing space* where the concentrations of the two morphogens will become unstable and result in the patterns shown in Figure 5.

The pattern is the result of each sensor running the equations locally while *diffusing* to its neighbors; a stable solution is thresholded to produce a binary value at each sensor, and the total of these gives the pattern in the sensor snow. (Note: in the example given here, the spatial solution is known analytically and was computed directly.)

For example, the pattern shown in Figure 1 makes a very regular framework within which a robot can move, while Figure 4 provides a set of fixed distances from some radially aligned point. Such a pattern can be used as an incremental encoder for the determination of distance traveled, or as a mechanism for robot rendezvous.

Generated patterns may also be much less regular (see Figure 5) while still having statistically useful properties; e.g., a certain percentage of area covered with *on* bits. The notion is that a mobile robot traveling near a given sensor can determine whether the sensor is on or off (in the final spatial pattern) and use that information to achieve its desired behavior; e.g., follow a stripe in Figure 5. Multiple patterns may exist simultaneously for different goals, and patterns may change over time.

3 Local and Global Frames

To utilize the sensor population for navigation and localization purposes, local and/or global frames of reference need to be established. The assumption here is that the only information each sensor knows about the other sensors is their distance from it. Sensors can have uniquely identifiable tagged chirps, and can broadcast a signal that other sensors hear and relay back after some delay. Thus, a set of distances can be determined and from these an algorithm can compute positions with respect to a local sensor frame. The following is a formulation of this problem and its solution.

Problem: Given a set of n points, where $n \geq 3$, and the distances between each pair of points, $\{d_{ij}\}$, where $i = 1..n, j = 1..n$, and $i \neq j$, it is required to establish a planar frame of reference F where point i can be represented by the coordinates (x_i, y_i), where $i = 1..n$.

First we construct a global frame for the set of points and then show how to construct a local frame in which the frame origin is selected either arbitrarily or relative to a certain event or landmark.

3.1 Global Frame Calculation

In this formulation we show that the global (absolute) coordinates of at least three points, p_1, p_2, and p_3, not laying on a straight line are needed to establish a global frame for a set of n points ($n \geq 3$).

Assume that the global coordinates of the first three points are: (x_1, y_1), (x_2, y_2), and (x_3, y_3), respectively, and the distances between these points and point p_i are d_{i1}, d_{i2}, and d_{i3}, respectively, where $i = 4..n$ (see Figure 6). The distance can be expressed in terms of the unknown coordinates of point i, (x_i, y_i), as follows:

$$(x_i - x_1)^2 + (y_i - y_1)^2 = d_{i1}^2 \tag{4}$$

$$(x_i - x_2)^2 + (y_i - y_2)^2 = d_{i2}^2 \tag{5}$$

$$(x_i - x_3)^2 + (y_i - y_3)^2 = d_{i3}^2 \tag{6}$$

By subtracting Equation 5 from Equation 4 we get:

$$-2x_1x_i + x_1^2 + 2x_2x_i - x_2^2 - 2y_1y_i + y_1^2 + 2y_2y_i - y_2^2 = d_{i1}^2 - d_{i2}^2$$

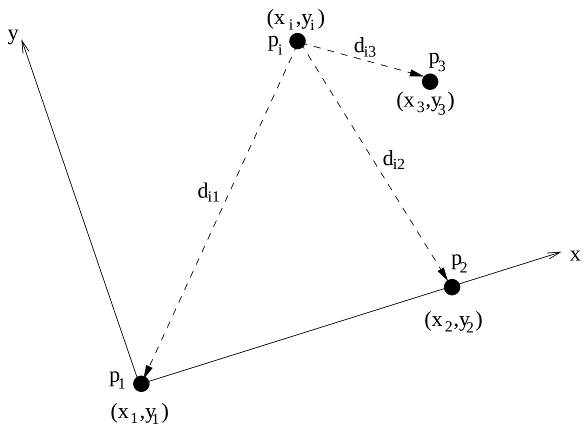

Figure 6: The formulation of the global frame.

and subtracting Equation 6 from Equation 4 yields:

$$-2x_1x_i + x_1^2 + 2x_3x_i - x_3^2 - 2y_1y_i + y_1^2 + 2y_3y_i - y_3^2 = d_{i1}^2 - d_{i3}^2$$

Simplifying the last two equations yields:

$$(x_2 - x_1)x_i + (y_2 - y_1)y_i = C_1 \tag{7}$$

and

$$(x_3 - x_1)x_i + (y_3 - y_1)y_i = C_2 \tag{8}$$

where

$$C_1 = \frac{1}{2}(d_{i1}^2 - d_{i2}^2 - x_1^2 + x_2^2 - y_1^2 + y_2^2)$$

and

$$C_2 = \frac{1}{2}(d_{i1}^2 - d_{i3}^2 - x_1^2 + x_3^2 - y_1^2 + y_3^2)$$

From Equation 7 we get:

$$x_i = \frac{C_1 - (y_2 - y_1)y_i}{(x_2 - x_1)} \tag{9}$$

Substituting 9 into 8 we can calculate y_i as:

$$y_i = \frac{C_2 - \frac{(x_3-x_1)}{(x_2-x_1)}C_1}{(y_3 - y_1) - \frac{(x_3-x_1)}{(x_2-x_1)}(y_2 - y_1)}$$

Simplifying this equation yields:

$$y_i = \frac{(x_1 - x_3)C_1 + (x_2 - x_1)C_2}{y_1(x_3 - x_2) + y_2(x_1 - x_3) + y_3(x_2 - x_1)} \tag{10}$$

Substituting for y_i in Equation 9 we get:

$$x_i = \frac{C_1 - (y_2 - y_1)\frac{(x_1 - x_3)C_1 + (x_2 - x_1)C_2}{y_1(x_3 - x_2) + y_2(x_1 - x_3) + y_3(x_2 - x_1)}}{(x_2 - x_1)}$$

Simplifying this last equation we get the solution for x_i as:

$$x_i = \frac{(y_3 - y_1)C_1 + (y_1 - y_2)C_2}{y_1(x_3 - x_2) + y_2(x_1 - x_3) + y_3(x_2 - x_1)} \tag{11}$$

This formulation shows that the absolute position (x, y) of 3 points not forming a straight line are sufficient to construct a frame and calculate the location of the other points with respect to that frame. The condition that the 3 points should not be on a straight line is necessary, otherwise, the denominator in Equations 10 and 11 will be zero.

3.2 Constructing a Local Frame

The main goal of the sensor snow paradigm is to provide sensory information within a wide spatial area where clusters of small simple sensors are used to identify and locate certain events or actions in the environment. In many cases it is sufficient to use a local (or relative) frame to navigate and locate local events within the cluster range.

A local frame can be viewed as a special case of the global frame, where the position and orientation of the frame can be chosen arbitrarily. We now show how to construct a local frame for a set of n points, where $n \geq 3$.

1. Select any 3 points p_1, p_2, and p_3 not laying on a straight line. Assume that the distances between these points are d_{12}, d_{13}, and d_{23}. The condition for non linearity is that the distance between any two points should be less that the sum of the distances between these two points and the third point:
$$d_{ij} < d_{ik} + d_{jk}$$

2. Set p_1 as the frame origin (i.e., $p_1 = (0, 0)$).

3. Form the x-axis of the local frame as p_1p_2, where p_2 is constrained by a circle centered at p_1 with radius d_{12}. This means that $p_2 = (d_{12}, 0)$.

4. Calculate the location of the third point by selecting one of the two intersection points of two circles centered at p_1 and p_2 with radii d_{13} and d_{23}, respectively (see Figure 7). To calculate the location of p_3 we solve the following two equations:

$$(x_3 - x_1)^2 + (y_3 - y_1)^2 = d_{13}$$
$$(x_3 - x_2)^2 + (y_3 - y_2)^2 = d_{23}$$

Substituting the values of x_1, y_1, x_2, and y_2 from steps 2 and 3 above we get:

$$x_3^2 + y_3^2 = d_{13}^2$$
$$(x_3 - d_{12})^2 + y_3^2 = d_{23}^2$$

By solving these two equations for x_3 and y_3 we get:

$$x_3 = \frac{d_{12}^2 + d_{13}^2 - d_{23}^2}{2d_{12}} \tag{12}$$

$$y_3 = \pm\sqrt{d_{13}^2 - \frac{d_{12}^2 + d_{13}^2 - d_{23}^2}{2d_{12}}} \tag{13}$$

We then select one of the two locations for p_3. Here we select the positive value of y_3.

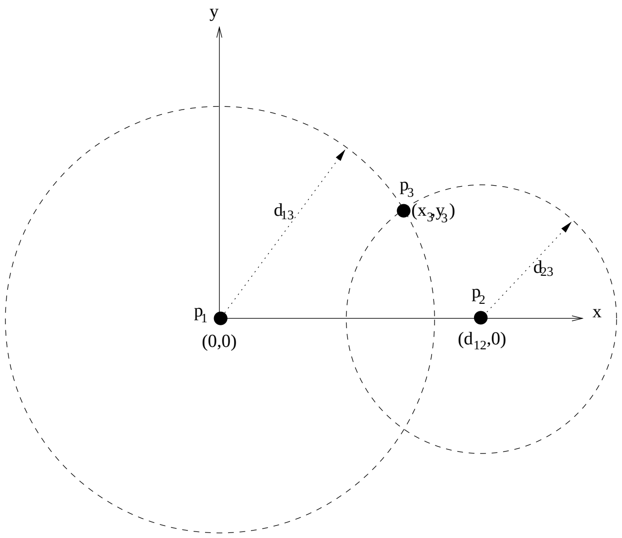

Figure 7: Construction of a local frame.

5. Use the results from the global frame calculation to get the (x, y) locations of the remaining points with respect to this local frame.

3.3 Moving between local frames

Moving from one sensor cluster to another requires a transformation between the two local frames. To determine the transformation functions, there must be at least two points common to both frames. Using a homogeneous transformation, the relationship between the (x, y) locations of a point p_i with respect to both frames can be written as (see Figure 8):

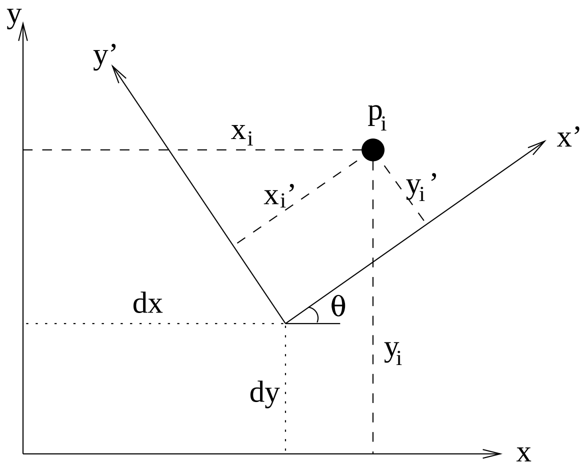

Figure 8: Transformation between frames.

$$\begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & d_x \\ \sin\theta & \cos\theta & d_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i^{'} \\ y_i^{'} \\ 1 \end{bmatrix} \tag{14}$$

Thus, we have two equations in three unknowns; d_x, d_y, and θ.

$$x_i = x_i^{'} \cos\theta - y_i^{'} \sin\theta + d_x \tag{15}$$

$$y_i = x_i^{'} \sin\theta + y_i^{'} \cos\theta + d_y \tag{16}$$

Therefore, we need at least two common points, p_1 and p_2, to solve Equations 15 and 16.

Assume that we have two common points p_1 and p_2 with coordinates (x_1, y_1) and (x_2, y_2) with respect to the first frame, and $(x_1^{'}, y_1^{'})$ and $(x_2^{'}, y_2^{'})$ with respect to the second frame. Substituting these points in Equations 15 and 16 yields:

$$x_1 = x_1^{'} \cos\theta - y_1^{'} \sin\theta + d_x \tag{17}$$

$$y_1 = x_1^{'} \sin\theta + y_1^{'} \cos\theta + d_y \tag{18}$$

$$x_2 = x_2^{'} \cos\theta - y_2^{'} \sin\theta + d_x \tag{19}$$

$$y_2 = x_2^{'} \sin\theta + y_2^{'} \cos\theta + d_y \tag{20}$$

By subtracting Equation 19 from Equation 17 and Equation 20 from Equation 18 we get:

$$(x_1 - x_2) = (x_1^{'} - x_2^{'}) \cos\theta - (y_1^{'} - y_2^{'}) \sin\theta \tag{21}$$

$$(y_1 - y_2) = (x_1^{'} - x_2^{'}) \sin\theta + (y_1^{'} - y_2^{'}) \cos\theta \tag{22}$$

Solving the last two equations results in:

$$\theta = atan2\,(A, B) \tag{23}$$

where

$$A = (x_1^{'} - x_2^{'})(y_1 - y_2) - (y_1^{'} - y_2^{'})(x_1 - x_2)$$

and

$$B = (x_1^{'} - x_2^{'})(x_1 - x_2) + (y_1^{'} - y_2^{'})(y_1 - y_2)$$

Substitute for θ in Equations 17 and 18 to solve for d_x and d_y as follows:

$$d_x = x_1 - x_1^{'} \cos\theta + y_1^{'} \sin\theta \tag{24}$$

$$d_y = y_1 - x_1^{'} \sin\theta - y_1^{'} \cos\theta \tag{25}$$

Finally, we use Equation 14 to transform the rest of the points between the two frames.

4 Exploitation of Sensor Snow

In order for the mobile robots to exploit the sensors effectively, each sensor must have:

- a measure of the density of sensors around it
- a measure of the distance to a sensed event
- the direction to a sensed event
- the direction toward sensor rich areas (i.e., a gradient of sensor density)
- pattern values for various useful patterns in the application (each pattern could be represented as a bit in a concatenated word).

The mobile robots use this information to navigate or to approach or retreat from an event while maintaining some asserted relations (e.g., stay in sensor rich area, stay fixed distance from other robots, etc.). The sensors may also be able to provide a spatial mailbox (or communication medium) between robots by storing some bits of information.

It is also possible to put the sensors only in places such as the mission periphery so as to detect activity in that area. This avoids the necessity of spatially localizing sensor feedback. Another use is to have robots drop sensors themselves and leave a trail; this makes it easier to retrace a path or for other robots to follow (see Figure 9). Navigation behaviors can follow trails such as the one shown in Figure 5 or move from one sensor cluster to another as shown in Figure 10.

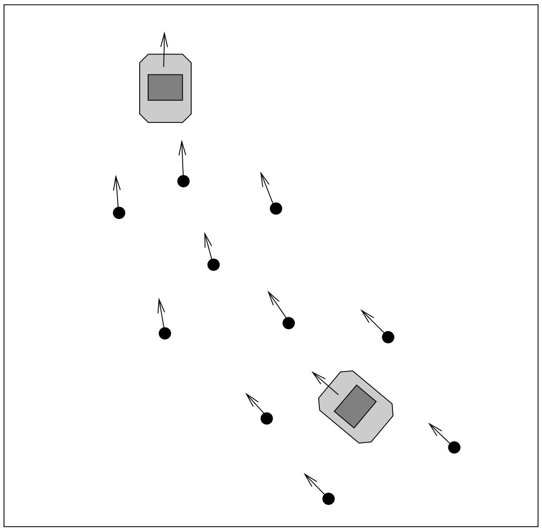

Figure 9: A robot following the trail of another robot.

5 Conclusions and Future Work

We have presented a formal basis for the use of a large number of spatially distributed sensors in order to:

- produce useful patterns in the sensor set,
- produce local frames with the sensors, and
- exploit the sensors to achieve more locally-tuned and robust robot behaviors.

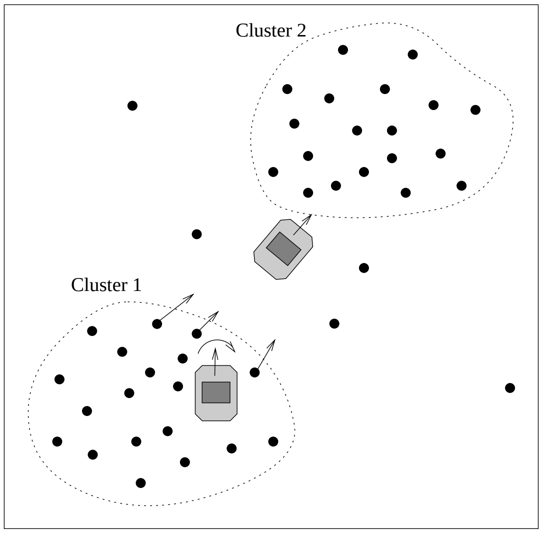

Figure 10: Robots moving between sensor clusters.

We are developing:

- distributed algorithms to calculate these patterns and frames.
- requirements specifications for sensors to be used this way,
- communication techniques between sensors and robots,
- robot behaviors based on these ideas.

Questions that must be addressed include:

- How can mobile robots recognize and identify the individual sensors?
- What happens if 2 sensors are spatially very close (i.e., cannot be resolved spatially)?
- How can local frame coordinates or pattern values be propagated between sensors?

References

[1] DEKHIL, M., AND HENDERSON, T. C. Instrumented sensor systems. *International Journal of Robotics Research* (May 1998), to appear.

[2] HENDERSON, T., AND SHILCRAT, E. Logical sensor systems. *Journal of Robotic Systems 1*, 2 (1984), 169–193.

[3] HENDERSON, T. C., DEKHIL, M., KESSLER, R., AND GRISS, M. Sensor fusion. In *Control Problems in Robotics and Automation* (Berlin, 1998), B. Siciliano and K. Valavanis, Eds., Springer-Verlag, pp. 193–208.

[4] MURRAY, J. *Mathematical Biology*, vol. 19 of *Biomathematics*. Springer-Verlag, New York, 1993.

[5] TURING, A. The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society Society of London B237* (1952), 37–72.