Trends in Optical Engineering, 1(1993)

Multi-sensor fusion : A perspective*

Jay K. Hackett and Mubarak Shah
Computer Science Department, University of Central Florida, Orlando, FL 32816, USA

Abstract

Multi-sensor fusion deals with the combination of complementary and sometimes competing sensor data into a reliable estimate of the environment to achieve a sum which is better than the parts. Some of the commonly used sensors are video camera, tactile sensor, laser range finder, infra-red sensor, and sonar. Multi-sensor systems have applications in automatic target recognition, autonomous robot navigation, and automatic manufacturing. This paper presents a current survey of the state of the art in multi-sensor fusion. We have surveyed papers related to fusion and classified them into six categories: scene segmentation, representation, 3-D shape, sensor modeling, autonomous robots, and object recognition. A number of fusion strategies have been employed to combine sensor outputs. These strategies range from simple set intersection, logical *and* operations, and heuristic production rules to more complex methods involving non-linear least squares fit and maximum likelihood estimates. Sensor uncertainty has been modeled using Bayesian probabilities, and support and plausibility involving the Dempster-Shafer formalism.

1 Introduction

Sensor fusion combines the output from two or more devices that retrieve a particular property of the environment. Commonly used sensors are a video camera, range finder, sonar, and tactile sensor. The sensor outputs can be combined at the lowest level. For instance, depth maps obtained from the laser range finder and sonar can be combined to produce a better depth estimate. In an autonomous robot environment, vision can be used to locate objects, sonar can be used to determine the distance of an object from the camera to compute three dimensional locations, a tactile sensor can be used to detect the contact between the robot end effector and the object. Multi-sensor systems have applications in automatic target recognition systems, automatic manufacturing, and autonomous robots.

The advantages of using multiple sensors are many. Since each sensor output contains noise and measurement errors, multiple sensors can be used to determine the same property, but with the consensus of all sensors. In this way, sensor uncertainty can be reduced. The output of a single sensor in some cases may be ambiguous and misleading, in which case another sensor can be used to resolve the ambiguity. For instance, vision does a poor job in scenes with shadows, so a range sensor can be used because it does not have such problems. In some cases, multiple sensor data can be integrated in a way that can provide information otherwise unavailable, or difficult to obtain from any single type of sensor. In an object recognition system, where a single sensor might not be able to constrain the system to produce a unique interpretation, multiple sensors can be employed to introduce multiple position constraints which will reduce the search space. Multiple sensors can be used to obtain multiple views of the same scene, in this way a large part of the environment can be sensed, and the problem of occlusion can be dealt with. Every sensor is sensitive to a different property of the environment; in order to sense multiple properties, it is necessary to use multiple sensors. The system can be made fault tolerant by designing redundancy into the system. This means that a system using multiple sensors that sense a single property can be used. In case of failure of any single sensor, the system will still be able to function.

The addition of more sensors to the system gives rise to more data which need to be processed and intelligently combined. It is true that if one sensor provides a certain level of performance, then obviously two sensors would be better, and three would be even better; and with many more sensors we should be able to build a system that is much better than a system with just one sensor! The question is, Is this really true? [Nahin 1977]. Massaging a lot of inaccurate data does not produce good data, it just requires a lot of extra equipment and may even reduce the quality of the output by introducing extra time delays and unwarranted confidence [Fowler 1979]. When raw sensor measurements are imprecise and noisy, they need to be modeled and characterized. Methods need to be developed for determining consistency of data, and fusion of consistent data. The precise operations involved in fusion depend on the level of data used. The most simple case of fusion for a multi-sensor configuration that records the same property of the environment

*The research reported in this paper was supported by the Center for Research in Electro Optics and Lasers (CREOL) University of Central Florida under grant number 20-52-043 and by the National Science Foundation under grant number IRI 87-13120

is to combine the data using averaging. Here it is assumed that nothing is known a priori about the sensors. Thus each sensor's measurement is equally likely. In the cases when it is known that a particular sensor reading is more reliable than the others, a weighted average can be be used instead. Appropriate weights can be assigned proportional to the reliability of the sensor. This simple heuristic method will have problems when a large number of sensors are used and sensor interaction is more complex. There are formal approaches involving probability distributions to model these situations. Sensor fusion deals with the selection of a proper model for each sensor, and identification of an appropriate fusion method.

This paper deals with a current survey of research in the multi-sensor fusion area. We have confined ourselves to the papers related to vision, AI, and robotics. We are not aware of any previous survey done on the multi-sensor fusion topic, except a summary paper by [Mitiche and Aggarwal 1986], which summarizes their own group's research in sensor integration. A workshop on Multisensor Integration in Manufacturing was held in 1987 and a technical report by [Henserson et al. 1987] summarizes the conclusions of this workshop. Another workshop on Spatial Reasoning and Multisensor Fusion in the same year was organized by AAAI [KaK and Chen 1987]. A special issue of International Journal of Robotics and Automation, edited by Brady in December 1988 [Brady 1988] contained many papers on multisensor fusion. Other literature sources of the literature on multi-sensor fusion are: IEEE Transactions on Automation and Robotics and regularly scheduled conferences: Conference on Robotics and Automation, Computer Vision and Pattern Recognition, International Conference on Pattern Recognition, International Joint Conference on AI.

The organization of the rest of the paper is as follows. In the next section, we discuss some issues related to a multisensor system. In particular, we outline various possible configurations of a multi-sensor system, and describe a general framework for fusion and integration systems using block diagrams. Section three is devoted to the description of commonly used sensors and a few specialized sensors. There are a few fusion strategies which have been used in various forms in most of the papers surveyed. Therefore, in order to make this paper self contained we have included some introductory material on fusion strategies in section four. Finally, section five deals with the survey of existing methods. We have classified the papers into six categories: segmentation, representation, shape, sensor modeling, autonomous robots, and recognition.

2 Multisensor Systems

The span of possible multi-sensor systems can be described by the product of three variables: sensor, property, and data, with two possible values, single or multiple, for each variable. This yields a total of eight different configurations. For instance, the single sensor, single property, single data configuration is an example of a system having only one sensor (e.g., one visual image obtained by a video camera). A single sensor, single property, multiple data is the configuration in which a single sensor records a property as a function of time, (e.g., a sequence of images describing a dynamic scene). An instance of multiple sensor, single property, single data configuration is a system with many range finders employed for redundancy purposes. A multiple sensor, multiple property, multiple data configuration is most general and complex. An example of this configuration is an autonomous robot with several sensors. All possible configurations considering three variables are enumerated in the table shown in Figure 1. The first four rows in the table involve only a single sensor, and are given here for the sake of completeness, but those cases will not be discussed in this paper.

There are several different methods for combining multiple data sources. Some of them are *deciding*, *guiding*, *averaging*, *Bayesian statistics*, and *integration*. Deciding is the use of one of the data sources during a certain time of the fusion process. Usually the decision as to which source to use is based upon some confidence measures or the use of the most dominant or more certain data. Averaging is the combination of several data sources, possibly in a weighted manner. The weights can be assigned based upon confidence values. This type of fusion ensures that all sensors play a role in the fusion process, but not all to the same degree. Guiding is the use of one or more sensors to focus the attention of another sensor on some part of the scene. An example of guiding is the use of intensity data to locate objects in a scene, and then the use of a tactile sensor to explore some of the objects in more detail. Integration is the delegation of various sensors to particular tasks. For instance, the intensity image may be used to find objects, the range image can then be used to find close objects, and then a tactile sensor can be used to help locate and pick up the close objects for further inspection. In this case, the data is not fused but is used in succession to complete a task. Therefore, there in no redundancy in sensor measurements.

Approaches to sensor fusion can be put into one general framework as shown in Figure 2. In this Figure, the sensors are shown by circles, and their outputs are denoted by $X_1, X_2, \ldots, X_n$. Corresponding to each sensor i, there is an input transformation denoted by f_i, which is shown by the oval shape. The input transformation could be the identity transformation, which does nothing to the input; that is the input and output are the same. On the other hand, it could be a simple operation like edge detection, or a more complex task like object recognition, which will output a list of possible interpretations of objects present in the scene. The fusion is performed in the large rectangular block. We have listed a number of possible fusion strategies which can be used. The most simple fusion strategy will be the one in which raw sensor measurements of the same property obtained by multiple sensors are combined. For instance, focus and stereo range data can be combined using Bayes' rule. In another case, the sonar and infra-red depth measurements can be combined using simple if-then rules, or the range and intensity edge maps can be fused by using the logical *and* operation. On the other hand, a more complex fusion strategy might use weighted least-squares fit to determine an object's location and orientation using multiple sensor measurements.

Sensor	Property	Data	Example
single	single	single	single visual image
single	single	multiple	sequence of visual images
single	multiple	single	multispectral image (color, etc.)
single	multiple	multiple	color image sequence
multiple	single	single	focus and stereo ranging ([Krotkov and Kories 1986])
multiple	single	multiple	sonar and stereo data sequence ([Matthies and Elfes 1988])
multiple	multiple	single	thermal and vision ([Nandhakumar and Aggarwal 1987])
multiple	multiple	multiple	range, vision, sonar, tactile sequences ([Giralt et al. 1985]) ([Ruokangas et al. 1986])

Figure 1: Multi-sensor configurations

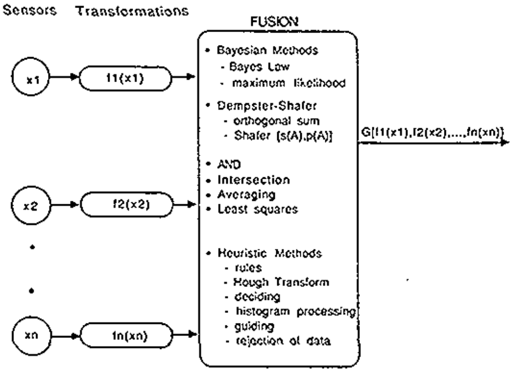

Figure 2: A Block diagram of Multisensor Fusion

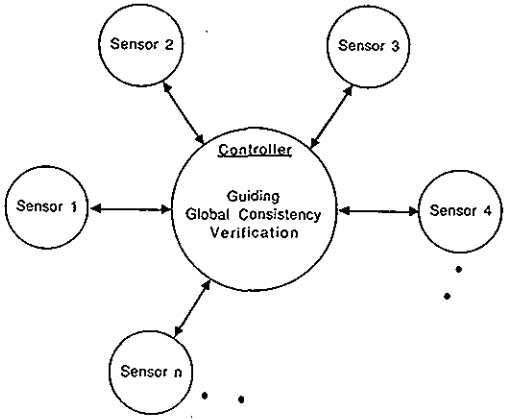

Figure 3: A Block diagram of Multisensor Integration

A general block diagram for multi-sensor integration is shown in Figure 3. Multi-sensor integration is the *use* of several sensors in a sequential manner to achieve a particular task. With integration, a particular sensor performs a subtask or provides a particular piece of information. The next sensor may proceed by using any previous information to perform its own tasks. In this way, each sensor is used as an expert in its modality, but an overall consensus of all of the sensors is not achieved. Therefore, there is no redundancy in sensor measurements. For integration to work properly, there must be some type of control which organizes the flow of data from one sensor to the other. Integration is a much simpler process than fusion since the controller uses the data from only one sensor to guide the actions of the other sensors. With fusion however, the data from different sensors must usually be put into equivalent forms to allow the fusion to occur. The benefit of fusion occurs because the output is achieved by the consent of all of the sensors.

In order for data from multiple sensors to be fused, there must be some method to relate data points from one sensor with corresponding data points from the other sensors in the system. The *registered* data points will easily allow for gathering of sensor information about one particular point in the scene. The registration can be done rather easily between some sensors, for instance intensity and range data can be registered by using known fields of view, tilt, and pan angles of the sensors. In case of different fields of view, image data for one sensor may not have corresponding data for the other sensor. The main problem we face in multi-sensor systems, and the one we want to solve through registration, is that sensors might provide data from different physical parts of an object. For instance, consider using a tactile sensor and an intensity camera. A typical tactile sensor has about 10 sensor elements per square inch. A video camera has much higher resolution (though a lens may still reduce the spatial resolution). The tactile sensor is also usually small in nature (perhaps 3 in. by 3 in.). The camera can give information about a much larger area. Thus the registration of the two sensors is very limited because only a few points have information from both sensors. Thus an obvious thing to do is to use the intensity camera as an overall data source and use the tactile sensor for detailed data.

3 Sensors

A sensor is a device that is used to retrieve information about a particular property of the environment. There are nearly as many kinds of sensors as there are properties. Existing sensors include a video camera, range finder, tactile sensor, sonar, infra-red, etc.

The video camera gathers intensity values which correspond to the light (of wavelength 400-700 nm) reflected from a position in the scene being sensed. The intensity value is a combination of many properties. The illumination, surface orientation, and reflectance properties all play a role in the value being sensed. Thus, shadows, surface texture, or uneven lighting can all affect very significantly the accuracy of the image. The resolution of a video camera typically ranges from 128 × 128 to 512 × 512 pixels. Since a video camera is measuring the intensity of scene points, we use the following terms interchangeably: video camera, vision, and intensity.

The laser range finder emits laser light of a certain frequency and records the returned light. The light that is returned is processed by a phase shifting calculation module. The difference in phase shift between the retrieved and the emitted light is proportional to the distance that the light has traveled. However when the light travels longer than the time for one wavelength, the results will be ambiguous since the phase shift calculator has no way to determine whether one or more wavelengths has passed. The range ambiguity is dependent upon the type of laser being used. The value is normally between 30 and 100 feet. Thus for objects farther than 30 feet away, the distance found is incorrect. Many laser range finders can also retrieve reflectance information from the scene. This is simply found by measuring the amplitude of the light that is retrieved from the scene. The reflectance information is similar to a video camera image except that illumination and shadows do not play a role in the image formation process. The laser range finder has the advantage that it can be used in the dark. An example of an existing laser range sensor is the Odetics 3-D mapper which sends out light at a frequency of 820 nm. It can record a resolution of 128 × 128 pixels with a minimum distance measurement of 1.5 ft and a maximum of 30.74 ft. The distance is encoded in 8 bits to produce 256 different range values. The ambiguity in distance occurs after 30.74 ft.

The tactile sensor is composed of an array of touch sensors which measure the force applied to the individual locations when the sensor comes in contact with the object. Many tactile sensors work by detecting changes in distance between two plates of a capacitor as described in [Siegel et al. 1986, Cameron et al. 1988]. The change in capacitance is proportional to the force applied to one plate of the capacitor. If an array of force sensors is formed then a force contact profile or image can be generated. The force profile is useful in determining surface orientation and surface characteristics such as holes. However, this sensor is active, meaning that it must come in contact with the objects in the scene. Therefore, it is usually attached to a robot arm which is programmed to make contact with an object.

The sonar sensor provides a single range value which indicates the distance that an object lies from the sensor. Typically, a sonar sensor can provide a measuring span of from 1.0 ft. to 30.0 ft. with 1 inch resolution [Matthies and Elfes 1988]. In the Polaroid sonar sensor [Flynn 1988], a sound chirp of 1 ms duration of 56 pulses at four different frequencies is transmitted. There are 8 pulses at 60 kHz, 8 pulses at 56 kHz, 16 pulses at 52.5 kHz, and 24 pulses at 49.1 kHz. The time of flight measurement begins with the rising edge of the first pulse transmitted and ends with the detection of the first echo. The sonar sensor has many disadvantages however. Typically, the sonar sensor has a 40 degree beamwidth angle which makes precise positioning very difficult. The sensor also has the tendency to smooth out the surfaces of objects. It provides the distance of the strongest reflective object in the field of view.

The thermal sensor is similar to a video camera but instead of measuring reflected visible light it measures infrared light. This sensor also has the advantage that it *can* be used in the dark, but it normally depends on some type of heat source such as the sun since the sensor measures heat lost from the surface of objects. Thus this sensor is not very effective for indoor scenes. One type of infra-red sensor is discussed in [Nandhakumar and Aggarwal 1987]. Thermal sensors are sensitive to the thermal spectrum ($1\mu m$ to $14\mu m$). However, most infra-red sensors concentrate on only a small portion of that spectrum, (i.e., $3\mu m$ to $5\mu m$, for instance).

The infra-red distance sensor sends out an infra-red pulse which when reflected back can be used to approximate the distance. This is possible since the distance will be proportional to the amplitude of the energy returned. However, it can only detect distances from 10 to 15 ft, and some surfaces will absorb the signal which will distort the distance readings.

[Orrock et al. 1983] present a new sensor which incorporates both vision and range sensors in one module. The vision sensor is a standard camera with a fixed focus lens that is mounted inside the box. The range sensor, called 'through the camera lens' (TCL) sensor, is a multiple point sensor that is similar to a focusing mechanism of an SLR camera. The scene light enters the box and is diverted to each of the sensors by way of mirrors. The TCL consists of 24 pairs of sensors which in a SLR camera will provide displaced measurements when the camera is out of focus. However, since the camera is provided with a fixed focus lens, the displaced measurements will provide range information, similar to disparity in stereo algorithms. A shift calculation algorithm is used to convert displacement to actual range distance. Useful range measurements can be obtained from 10 to 100 cm with at worst a 3% error. This type of sensor is very useful since both data sources are in registration with one another. This, however, sensor has one major drawback. The range data suffers from the same problems as does the intensity data, (i.e., it also suffers from illumination problems such as shadows).

[Shekhar et al. 1986] present a *centroid sensor* which is a 2-D contact sensor. It computes the center of pressure that occurs on the sensor's surface. The sensor consists of a rubber pressure conductive layer that is sandwiched between two layers of conductive film. The layer of rubber provides a resistance that varies based upon the pressure of force exerted on the sensor. Therefore, current is induced in those parts of the sensor where contact is made. The current can be measured so that first order moments can be computed. From the first order moments, center of pressure equations are derived.

[Krotkov and Kories 1986] present two additional ways that range data can be retrieved relatively cheaply. These are by *focus ranging* and *stereo ranging*. Focus ranging is accomplished as follows: set the focal length of a vision camera to its maximum, select a window in the image, focus the lens on the window, determine the sharpness of focus by computing the gradient magnitude of the image window, then compute the distance by using the Gaussian lens law, which relates depth to focal length and depth of field of the lens, and the focus movement. This type of ranging is accurate to between 1 and 3 meters. Stereo ranging is performed as follows: set the focal lengths of two cameras to their minimum, capture both images, find interesting lines or points in both images, compute disparity for the image pair, and compute the distance by using triangulation. The disparity is computed by using a predict and test algorithm. Thus the original prediction can be revised. But it does makes correspondence errors occasionally. This method is also accurate to within 1 to 3 meters. Both of these ranging methods are passive, while the laser range finder is active and costs much more. The focus ranging method, however, will not return an instantaneous range value since each camera must be focused while retrieving the value. Also both range data sources provide very sparse data.

So far we have been talking about physical sensors. However, in our discussion not every sensor is a physical device. Our concept of sensors is more general, and is similar to the logical sensor used by [Henderson and Shilcrat 1984]. A video camera is a physical device which can be used to capture a sequence of images in a given time. Using this sequence, optical flow can be computed. Now, we can talk about an optical flow sensor whose output is a displacement vector at each pixel, where the displacement vectors are computed using a sequence of frames.

4 Fusion Strategies

Each fusion approach is unique to some extent, however, certain key fusion methods and their variations have been employed by many authors. In this section, we will summarize some commonly used fusion strategies. In section six, we discuss current research that uses these strategies. Fusion methods can be classified broadly into two categories: direct and indirect fusion. In methods related to direct fusion, the raw sensor measurements are combined, while in indirect methods a transformation of the sensor measurements is fused. Before the sensor measurements can be fused, whether directly or indirectly, their consistency has to be checked, (discussed in section 5). Bayesian theory has traditionally been used to model uncertainty in many disciplines for some time, thus there exists a well developed body of literature in this area. Therefore, a great number of approaches surveyed use Bayesian statistics as a fusion strategy. This will be discussed in subsection 4.1. Shafer-Dempster theory is another formalism that is used to model uncertainty. It has certain advantages over Bayesian approaches. A few authors have also used the Shafer-Dempster approach for fusion. We will summarize this method in subsection 4.2.

4.1 Bayesian Approaches

Bayesian statistics is very useful in combining multiple sensor values since sensor uncertainty can easily be incorporated. The state of the environment is decided based upon sensor measurements, knowledge about the types of states expected, as well as sensor uncertainty. New measurements can change the probability of a state occurring. A number of approaches surveyed in this paper make use of maximum likelihood, a well known Bayesian approach, as a fusion strategy. In this section we will review some of the basic concepts related to Bayesian approaches.

4.1.1 Direct Methods

The simpler forms of fusion employ raw sensor measurements directly. In this section, we will describe the maximum likelihood and Bayes' law for direct fusion. Assume that the sensor output is denoted by the vector $X = (x_1, x_2, \ldots x_n)$, and the object property (e.g., position, orientation, etc.) being estimated is denoted by Θ. We will be using two conditional probabilities: $p(X|\Theta)$ and $p(\Theta|X)$. $p(X|\Theta)$ is the probability of sensor output being X given that the object property is Θ, and $p(\Theta|X)$ is the probability of object property being Θ given that the sensor output is X. In our case, $p(X|\Theta)$ can be computed from the sensor model, while $p(\Theta|X)$ is the *a posteriori* probability which we want to determine. These two probabilities are related by *Bayes' Law*, which states:

$$p(\Theta|X) = \frac{p(X|\Theta)p(\Theta)}{p(X)} \tag{1}$$

where $p(X)$ and $p(\Theta)$ are the unconditional probabilities of the sensor output and object property being X and Θ respectively.

Assume that in our system there are k sensors, which give the following readings: $\mathcal{X} = (X^1, X^2, \ldots, X^k)$. We would like to develop the best estimate of the object property Θ using these k sensors readings. This can achieved by using the *likelihood estimate*. In the likelihood estimate we compute Θ such that the following is maximized:

$$p(\mathcal{X}|\Theta) = \prod_{l=1}^{k} p(X^l|\Theta) \tag{2}$$

It is usually easier to deal with the logarithm of the likelihood than the likelihood itself, because the product can be changed to the sum, and the terms involving exponents can be simplified. Let $L(\Theta)$ be the log-likelihood function:

$$L(\Theta) \;=\; \log p(\mathcal{X}|\Theta) = \sum_{l=1}^{k} \log p(X^l|\Theta) \qquad (3)$$

Assume that the readings from the sensors follow Gaussian density functions. Then $p(X^l/\Theta)$ is given by:

$$\begin{aligned} p(X^l|\Theta) \;&=\; \frac{1}{(2\pi)^{n/2}|C_l|^{1/2}} \qquad (4) \\ &\quad * \exp(-\frac{1}{2}(X^l-\Theta)^t C_l{}^{-1}(X_l-\Theta)) \end{aligned}$$

where C_l is the variance-covariance matrix, t denotes the transpose, and $|\,|$ denotes the determinant. Now, the expression for likelihood (equation 3) becomes:

$$\begin{aligned} L(\Theta) \;&=\; \sum_{l=1}^{k} \log p(X^l|\Theta) \qquad (5) \\ &=\; \sum_{l=1}^{k}(-\frac{1}{2}\log[(2\Pi)^n|C_l|] \qquad (6) \\ &\quad -\frac{1}{2}(X^l-\Theta)^t C_l{}^{-1}(X^l-\Theta)) \end{aligned}$$

The best estimate $\tilde{\Theta}$ of Θ can be found by differentiating L with respect to Θ, equating the result to zero, and computing the value of Θ, as follows:

$$\tilde{\Theta} = \frac{\sum_{l=1}^{k} C_l^{-1} X^l}{\sum_{l=1}^{k} C_l^{-1}} \qquad (7)$$

There are variations of equation 5 which have been used in the literature, for instance [Luo et al. 1988] maximize:

$$\sum_{l=1}^{k} p(\Theta|X^l)\, p(X^l) \qquad (8)$$

Since $p(\Theta|X^l)p(X^l) = p(X^l|\Theta)\, p(\Theta)$, the difference between this and equation 5 is that the logarithm is not used, and $p(\Theta)$ has been deleted. This results in more complicated expression involving a transcendental equation which is solved by an iterative scheme.

In the case when there are only two sensors in the system, and each sensor measurement, X_i, is a scalar, then the best estimate from equation 7 will be:

$$\tilde{\Theta} = \frac{({\sigma_2}^2)x^1 + ({\sigma_1}^2)x^2}{({\sigma_1}^2) + ({\sigma_2}^2)}$$

This equation shows the weighted average of two sensor readings; the weight is inversely proportional to the standard deviation of each sensor. In some cases, Bayes' law can be used directly to fuse the data coming from one sensor at multiple time instances, or the data from multiple sensors at one time instance. [Matthies and Elfes 1988] use *Occupancy grids* to represent the space around a robot that is occupied by objects so that obstacle avoidance may occur. The occupancy grid is a 2D array of cells that contains probability values which denote the chance of the cell containing an object or part of an object. The probability of a cell being occupied $p(OCC|R)$, given sensor reading R, using Bayes law is given by:

$$p(OCC|R) = \qquad (9)$$
$$\frac{p(R|OCC)\,p(OCC)}{p(R|OCC)\,p(OCC) + p(R|EMP)\,p(EMP)}$$

where $p(OCC)$ and $p(EMP)$ are *a priori* probabilities of a cell being occupied, and empty, respectively. This is modified for sequential updating based on multiple readings as:

$$p(OCC|R_{x+1}) = \qquad (10)$$
$$\frac{p(R_{x+1}|OCC)\,p^k(OCC)}{p(R_{k+1}|OCC)\,p^k(OCC) + p(R_{k+1}|EMP)\,p^k(EMP)}$$

where $p(OCC|R_{k+1})$ is the cell being occupied given $k+1$ readings, $p^k(OCC)$, $p^k(EMP)$ are unconditional probabilities of a cell being occupied, and empty respectively, based on k readings.

4.1.2 Indirect Methods

The previous section dealt with the direct fusion of raw sensor measurements for the multiple sensor, single property configuration. In some cases, one sensor measurement can be related to other sensor measurements by some known transformation. This can happen in the multiple sensor, multiple properties configuration. In these cases sensor measurements can be fused indirectly. The work of [Heeger and Hager 1988] is an example of indirect fusion. They fuse optical flow data and camera motion parameters in order to obtain consistent object motion and depth information, and use it for segmenting the scene into moving and stationary objects. They develop a linear equation that relates image velocity (optical flow) to camera motion. They consider the camera motion to be a vector $\vec{D} = (T_x, T_y, T_z, \Omega_x, \Omega_y, \Omega_z)$, and the optical flow to be $\vec{\Theta} = (u, v)$ where T_x, T_y, T_z are translations, and Ω_x, Ω_y, Ω_z are rotations. They can be related as follows:

$$\begin{bmatrix} u \\ v \end{bmatrix} = A(z)B(z)\vec{D} \qquad (11)$$

where $A(z) = \frac{1}{z}\begin{bmatrix} f & 0 & -x \\ 0 & f & -y \end{bmatrix}$,

$$B(z) = \begin{bmatrix} -1 & 0 & 0 & 0 & -Z & \frac{yZ}{f} \\ 0 & -1 & 0 & Z & 0 & \frac{-xZ}{f} \\ 0 & 0 & -1 & \frac{-yZ}{f} & \frac{xZ}{f} & 0 \end{bmatrix},$$

and (x, y, z) are image coordinates, Z is the depth. Let $p = 1/z$, $C(p) = A(1/p)B(1/p)$. Now, optical flow can be expressed as:

$$\vec{\Theta} = A(1/p)B(1/p)\vec{D} \qquad (12)$$

The joint likelihood for optical flow Θ, and depth $\vec{D}$ assuming Gaussian distributions becomes:

$$\log[f(\vec{D}, p)] \;=\; \frac{1}{2}(\tilde{D}-\vec{D})^t C_D^{-1}(\tilde{D}-\vec{D}) \qquad (13)$$

$$\begin{aligned}&+\frac{1}{2}(\tilde{\Theta}-\vec{\Theta})^t C_{\Theta}^{-1}(\tilde{\Theta}-\vec{\Theta})\\ =\ &\frac{1}{2}(\tilde{D}-\vec{D})^t C_D^{-1}(\tilde{D}-\vec{D}) \qquad (14)\\ &+\frac{1}{2}(\tilde{\Theta}-C(p)\vec{D})^t C_{\Theta}^{-1}(\tilde{\Theta}-C(p)\vec{D})\end{aligned}$$

where C_{Θ}^{-1}, and C_D^{-1} are variance-covariance matrices. They solve the above expression for $\tilde{D}$ with some fixed p, then they compute the *Mahalanobis* distance, which is used for consistency checking. The actual value of p is found by numerically minimizing the Mahalanobis distance.

In the model based system where the object model (in terms of 3-D vectors) and the sensor locations are known, unknown translation and rotation parameters of the object with respect to the sensors can be computed by fusing the sensor readings. [Shekhar et al. 1988][Shekhar et al. 1986] use a weighted least squares fit to fuse various sensor readings. Assume that the kth object point is denoted by vector p_k in the object coordinate system. If the object can rotate and translate, then its coordinates P_k in the sensor coordinate system are given by:

$$P_k = Rp_k + h$$

where R is the rotation matrix, and h is the translation vector. Our aim is to find R and h which are consistent with several sensor readings. Shekhar *et al.* treat translation and rotation separately. For instance, for computing h they consider the following. For any distance measurement D_i along a direction n_i, there is a distance d_i in the model computed along n_i. These two measurements are related as follows:

$$D_i = d_i + n_i^t h \qquad (15)$$

$$n_i^t h = D_i - d_i \qquad (16)$$

For multiple measurements the above equation becomes;

$$Ch = d$$

where $C = [n_1, n_2, \ldots, n_n]^t$, $d = [d_1, d_2, \ldots, d_n]^t$. Now, introducing the weights we get:

$$w_p Ch = w_p d$$

where the weights w_p are the δd_i^2 (expected errors in distance). This equation is solved using the standard least squares fit with pseudo inverse method.

[Durrant-Whyte 1986][Durrant-Whyte 1988] considers the problem of consistent updating and propagation in a multiple sensor environment. Each object in a system is represented by a six dimensional description,vector, consisting of information about the location and orientation of the object. Using description vectors, the equivalent homogeneous transformation relating the description vector of one object to that of another object is computed. For instance, in the case of three objects, let T_1 relate the coordinate frame of object-1 with object-2, T_2 relates the coordinate frame of object-2 with object-3, and T_3 relates the coordinate frame of object-3 with object-1. If all object measurements are consistent then $T_1T_2T_3 = I$, where I is the identity transformation. Let us assume that in the beginning the measurements are consistent, that is $T_1T_2T_3 = I$. Assume that a new measurement is made about object-1, which affects a change in T_1, so that $T_1' = T_1 + \delta T$. Now, $T_1{}'T_2T_3 \neq I$. Therefore, we need to find new T_2', T_3' to achieve $T_1'T_2'T_3' = I$. In general, this problem involves a set of non-linear matrix equations. Durrant-Whyte uses'differential transforms to approximate the consistency conditions by a set of linear matrix equations. He finds E, the vector of diagonal change-matrices, resulting from changes due to an update, of transform T_i by minimizing the quadratic objective function $L = \frac{1}{2}E^T\Omega^{-1}E$, where Ω is a diagonal matrix of the variance-covariance matrices representing the uncertainty in each relation in the world model.

4.2 Dempster-Shafer Approaches

In Dempster theory the probability is assigned to propositions, (i.e., to subsets of a frame of discernment Θ). This is a major departure from the Bayesian formalism in which probability masses can be assigned to only singleton subsets. When a source of evidence assigns probability masses to the propositions represented by subsets of Θ, the resulting function is called a basic probability assignment (bpa). Formally, a bpa is a function $m : 2^{\Theta} \Longrightarrow [0,1]$ where $0.0 \leq m \leq 1.0$, $m(\Phi) = 0$, $\sum_{x \subset \Theta} m(X) = 1$. Dempster's rule of combining states that two bpa's, m_1 and m_2, corresponding to two independent sources of evidence, may be combined to yield a new bpa m:

$$m(X) = K \sum_{X_1 \cap X_2 = X} m_1(X_1)m_2(X_2) \qquad (17)$$

where

$$K^{-1} = 1 - \sum_{X_1 \cap X_2 = \Phi} m_1(X_1)m_2(X_2). \qquad (18)$$

This combination is termed the *orthogonal sum*. For each sensor, a mass distribution is formed which divides the input data into portions that provide belief for different propositions. Thus the sum of all masses which contribute to the belief for a proposition X will be denoted by m(X). If each mass function from a sensor is thought of as a 1-D line, then two mass functions will become a 2-D box. Thus we can compute the orthogonal sum by an intersection of the components of the mass functions to find the total belief attributed to a proposition by both sensors. With three sensors, another dimension is added to the box to form a cube.

The Shafer theory is based upon an interval of uncertainty, [s(A),p(A)]. Here s(A) denotes the support for a proposition A being true and p(A) is the plausibility of proposition A. The interval between p(A) and s(A) denotes the uncertainty about proposition A. If the uncertainty is zero then we simply have a Bayesian approach since the support for the proposition A is equal to the maximum likelihood. Support may be interpreted as the total positive effect a body of evidence has on a proposition, while plausibility represents the total extent to which a body of evidence fails to refute the proposition. Support for a proposition A is the total mass ascribed to A and to its subsets, the plausibility of A is one minus the sum of the mass assigned to $\neg A$, the uncertainty of A is equal to the mass remaining. More formally, $S(A) = \sum_{A_i \subset A} m(A_i)$, $P(A) = 1 - S(\neg A)$.

The Dempster-Shafer method is different from Bayesian approaches since an interval of uncertainty is used, while a Bayesian approach uses only one value which represents the *probability of a proposition being true. The Bayesian ap*proach also has difficulty in maintaining consistency when propositions are related. This is the case since many Bayesian approaches require independent measurements about the environment. Bayesian methods also require more complete information since a single point value must be computed. While the Dempster-Shafer method allows only an interval based upon the uncertainty to be computed.

However, the Dempster-Shafer approach cannot be used directly if measurements from different sensors are not independent. Thus a new framework can be developed which uses the intersection of the two measurements, X_1 and X_2.

$$m(X) = K \sum_{X_1 \cap X_2 = X} INT(X_1 \cap X_2) m_1(X_1) m_2(X_2)$$

and

$$K^{-1} = 1 - \sum_{X_1, X_2} [1 - (X_1 \cap X_2)] m_1(X_1) m_2(X_2)$$

where $INT(X_1 \cap X_2) = \frac{MAX_{1,2}[MIN(m_1(X_1), m_2(X_2))]}{MIN_{1,2}[MAX(m_1(X_1), MAX(m_2(X_2))]}$

4.3 Consistency Check

Before sensor measurements can be combined, we have to make sure that the measurements represent the same physical entity. Therefore, we need to check consistency of sensor measurements. The *Mahalanobis* distance, T, is very useful for determining which data should be fused. It is defined as:

$$T = \frac{1}{2}(X_1 - X_2)^t C^{-1} (X_1 - X_2) \tag{19}$$

where X_1 and X_2 are two sensor measurements, and C is the sum of variance-covariance matrices related to the two sensors. The minimal 'distance' will indicate a consistency between the two measurements. The Mahalanobis distance will be larger when two measurements are very inconsistent and it will decrease when uncertainty becomes less.

[Krotkov and Kories 1986] use Mahalanobis distance as a consistency test for a system with two sensors; the reading from each sensor is a scalar denoted by x^1, x^2. The above equation (equation 17) simplifies to:

$$T = \frac{(x^1 - x^2)^2}{\sqrt{\sigma_1{}^2 + \sigma_2{}^2}} \tag{20}$$

where σ_1 and σ_2 are the standard deviations of sensor measurements x^1 and x^2. If $T \leq T_\alpha$, where T_α is some threshold, then the sensor measurements are consistent.

[Luo et al. 1988] use probability distances d_{ij}, and d_{ji} as the consistency check between sensors i and j.

$$d_{ij} = | \int_{x_i}^{x_j} P_i(x|x_i) P_i(x_i) dx | \tag{21}$$

$$d_{ji} = | \int_{x_j}^{x_i} P_j(x|x_j) P_j(x_j) dx | \tag{22}$$

where P_i, and P_j are *a priori* probabilities related to sensors i and j, and $P_i(x|x_i)$, and $P_j(x|x_j)$, are the conditional probabilities.

5 Survey of Existing Methods

This section surveys papers dealing with multi-sensor fusion and other related topics. We have attempted to classify the papers into six broad categories: segmentation, representation, 3-D shape, sensor modeling, autonomous robots, and recognition. This classification, however, is not strict; there might be some papers which belong to more than one category. For each group of papers, we have included a summary table listing the authors, sensors used, and fusion type employed in each paper in that particular category. The largest group of papers deals with segmentation, and the smallest group of papers discusses object recognition. Due to the fact that segmentation is the earliest perception task, and involves lower level processing, fusion at that level is simpler, in general. Object recognition is the most sophisticated task, and involves fusion at the feature level.

5.1 Segmentation

Segmentation is one of the most basic low-level processes in computer vision. There are two types of segmentation: region-based and edge-based. In the region based segmentation, an attempt is made to group pixels in an image based on their similarity to one another. The *similarity* is usually based on the raw pixel values. In edge-based segmentation, the boundaries of objects are identified by locating pixels where the change in pixel values is high. If segmentation is done accurately then each region should correspond to one object or one area of interest in the image. The *segmented* image can then be used for object recognition or other *vision* processes. Several multi-sensor fusion approaches which result in segmentation will be discussed. The majority of the papers deal with segmentation using range and intensity images. This is due to the fact that range and intensity images are readily available in a registered form from a laser *range finder or a structured light sensor. Fusion is performed* mostly at a lower level using pixel values and their distributions. Heuristics like deciding, guiding, and filling in are used. The use of other sensors like thermal with vision has been limited, since a thermal sensor would mostly be useful for scenes containing objects with large temperature variations. Also, a thermal sensor does not provide direct 3-D information like a range sensor does. Moreover, modeling of thermal images is more complex than it is with range images. The combination of contact sensors like tactile with vision at the lower level for segmentation purposes is not practical owing to large resolution differences. Segmentation or labeling has also been done at the feature level using intensity and light striping sensors, and surface orientation estimates ob-

Authors	Sensor Data	Fusion Type
[Duda et al. 1979]	range - vision	selection
[Hackett and Shah 1988, Hackett and Shah 1989]	range - vision	deciding
[Wong and Hayrapetian 1982]	range - vision	registration
[Zuk et al. 1985]	range - vision	rejection of data
[Gil et al. 1983]	range - vision	*and*
[Nandhakumar and Aggarwal 1987]	thermal - vision	heat flux - surface orientation
[Mitiche 1984]	stereo range - motion	rigid body theorem
[Hu and Stockman 1987]	light stripes - vision	rules
[Moerdler and Kender 1987]	shape from texture	Hough transform
[Duncan et al. 1987]	-	deciding

Figure 4: Approaches to segmentation

tained by multiple shape from texture algorithms. The first of such papers [Moerdler and Kender 1987] uses production rules for combining two sensor outputs, and the second uses a Hough-like transformation to consolidate various orientation constraints.

[Duda et al. 1979] perform segmentation by locating planar surfaces with the use of registered *reflectance* and *range* data. The major horizontal and vertical surfaces are found by using the range data. A histogram scheme is used to find the horizontal surfaces and a Hough transform is used to find the vertical surfaces. Any remaining surfaces are found by using histograms of intensity that contain data only from the, as yet, unassigned pixels. This method of segmentation has problems when curved surfaces are added to the scene, since the Hough transform method used is sensitive only to vertical planar surfaces and the histogram scheme used is sensitive only to horizontal surfaces. This method uses the range data extensively, but uses the reflectance data only to find non-vertical and non-horizontal planar surfaces.

[Hackett and Shah 1988][Hackett and Shah 1989] use registered range and intensity data in order to segment a scene. A histogram for each data source is found and the most significant peak from all of the histograms is extracted. The valley points of the peak are used to segment the scene. The process will repeat after new histograms are calculated until no significant peaks are found. In this way a global segmentation is achieved. The segmented image is then passed to a local region merging process where regions are merged if the boundary between two regions is weak. In the intensity images, weakness is defined by the sum of the gradient of pixels along the boundary. In the range image, weakness is the quantitative measure of jump boundary between two regions. The merging continues until all boundaries are *strong*. This method can be used with other sensors as long as methods exist that allow the extraction of boundary strength.

[Wong and Hayrapetian 1982] use range data to guide the extraction of objects in the intensity image. A method of transforming sensor location to location in 3-D space is also presented. The method may be applicable to registering two images from different sensors. A 3-D array $f(x,y,z) = i$ is formed where (x,y) are intensity image coordinates, z is the distance found in the registered range image, and i is the intensity at (x,y). A histogram of the range data is used to select n thresholds. For each threshold, the array $f(x,y,z)$ is partitioned into n arrays. Thus, in essence, the range data is used to partition the array based upon the distance of objects (features) in the original data. It is then unclear how segmentation proceeds. It is possible that each of the n partitions can be segmented by normal intensity segmentation methods. However the purpose of such a segmentation is unclear since no results are shown.

[Zuk et al. 1985] use a range and reflectance sensor to segment the road from non-road regions, so that navigation can occur. The fact that roads tend to have a smoother texture than the surrounding environment does is used. Range pixels are rejected where the corresponding reflectance value is very small. Small reflectance values usually indicate that the range data cannot be trusted. There is no real sensor fusion occurring, but the reflectance information is used only to neglect inaccurate range pixels. The texture is then measured on each scan line of the revised range image. Thresholds are chosen which separate weak textures, medium textures, and strong textures from one another. Another algorithm is used to produce a new image that has connected edges between smooth and rough areas in the image. These edge points define areas where accessible paths exist in the scene.

[Gil et al. 1983] use vision and range sensors to segment a scene. Each image is converted into an edge map representation. For finding edges in the intensity image, the Kirsch edge operator with non-maximal suppression is used. Edges in the range image are found by computing the *angle of curvature* and marking areas of large curvature as range edges. The edge maps for the two images may not be registered even though the original images were registered. This is true since each image needs different edge finding algorithms which cause the edge pixel to occur on either side of the actual edge. The two edge maps are fused as follows: each pixel in the intensity image is *AND'ed* with a k by k neighborhood of the corresponding pixel in the range image. This method is considered a local *AND* operation. The problem with this type of segmentation is that a small gap in the edge boundary will cause two regions to *flow* or *melt* together.

[Nandhakumar and Aggarwal 1987] use thermal (infrared) and intensity data to estimate surface heat fluxes which can be used in segmentation of outdoor scenes. The infra-red value obtained at the sensor is dependent upon an object's *surface temperature* as well as the *surface emissivity* which is shown to be fairly constant for outdoor scene objects. Other small contributions to the infra-red value are from *solar radiation* and from infra-red reflections from other objects. Objects that are present in normal outdoor scenes are examined for conductive *heat fluxes* that emanate from their surfaces. These heat fluxes are estimates of average heat flux for a type of object. The intensity image is used to compute the *solar absorptivity* and the difference in angle between the projection of the surface normal to the heat or light source. A calibration is performed on the camera to yield the surface normal angle for each region (pixel) in the intensity image. It is assumed that the surface orientation for each pixel is known. This is a major restriction unless some way to compute or determine orientation is known. The *solar absorptivity* and *surface normal* angles are used to compute the amount of heat absorbed by the surface of an object. The ratio of heat conducted from the surface into the object to the heat absorbed by the surface is computed for each pixel. It is shown that this value varies widely for objects in outdoor scenes. This ratio is used to directly *segment* the scene. However, this value is incalculable for areas where shadows occur in the scene. In other words, without a significant loss of heat from a surface the segmentation will not work. For areas of the scene where there are errors, an alternate segmentation method must be used.

[Mitiche 1984] combines stereo range data and optical flow to determine three dimensional motion, and segments the objects based on their motion. The fusion is accomplished by constraining the problem to rigid body objects. The 3-D motion of scene points is determined from 2-D optical flow, correspondence (registration) parameters, and depth information. This takes the form of relating velocity of two points in separate binocular images. If the correspondence between these points and all camera parameters are known then the depth information can be directly obtained. The rigid body theorem, which directly relates optical flow and depth information, is used as a test for rigidity. The locations where the theorem is not satisfied signify the boundary between objects with different motions. Hence, this simple technique is used to segment the scene. By example, it is shown that this method is better at obtaining depth information and segmentation than by using binocular vision only.

[Hu and Stockman 1987] fuse intensity data with sparse 3-D information obtained from a light striping sensor. The intensity image is found by turning off the light striping projector, thus the two data sources are in registration with each other. Since the 3-D data is sparse, the intensity information is used to *fill in* where 3-D information is unavailable. They used rules based upon the stripe pattern and contour type found. These rules return the type of surface present at the contour and label for the contour. The surfaces can be convex, concave, saddle, etc. and the contour types are defined as *extremum*, *blade*, *fold*, *shadow*, and *mark*. The image contour is found by performing edge detection on the intensity data. The triggering of rules is simply a segmentation or labeling process. For instance one of the rules used is: If two regions meet at a contour and the stripes continuously cross the contour (stripes are end-to-end connected at the contour) but the normals are discontinuous, then the contour is a fold. It is possible that a contour label cannot be determined uniquely, (i.e., an ambiguity exists). For example, an ambiguity exists in determining whether a contour is a jump (blade) boundary or a roof (fold) boundary. To resolve ambiguity, a relaxation procedure can be used. Both 3-D and intensity data have been used here to reduce the ambiguity that exists so that less complex relaxation labeling is needed.

[Moerdler and Kender 1987] describe a method for integrating surface orientations derived from various shape-from-texture algorithms and use it for surface separation. The fusion process is not one of fusing data from different sources but fusion of output from different algorithms working on the same image. Many of the shape from texture algorithms provide inconsistent or incorrect orientation information because of noisy input images or bad algorithms. Each shape from texture output is a texel patch with a list of orientation constraints and the expected accuracy of the constraints on the patches. The expected accuracy is dependent on the algorithm that returned the texel patch. The orientation constraints are mapped to points on a Gaussian sphere. Each constraint is assigned a circle on the surface of the sphere. Each of two constraints constructs two circles that overlap to produce two intersection points. One of which is the orientation of the visible surface patch, and the other intersection is the invisible patch and is not used. All orientation constraints for each texel are consolidated in the single most likely orientation by a Hough like transformation. This method has problems with shadows, mostly because the shape from texture algorithms are unable to correctly discern the shadows from image objects.

[Duncan et al. 1987] use the sensor that provides the best data for segmenting a particular part of the image. A *learning automata* is presented that allows for rewarding and penalizing of the use of certain sensors. If the segmentation is proceeding well with a particular sensor then then segmentation is allowed to continue. Otherwise, the sensor is penalized and will most likely relinquish control to another sensor. Since it is assumed that the images are registered, there is no need to reorganize the segmentation when sensors are switched.

5.2 Representation

A number of papers deal with the building of representations of objects and space by using multiple sensors. Representation techniques include octree, occupancy grid, spherical octree, 3-D position and orientation vectors, tactile and other visual features. An octree is a hierarchical representation of space, in which a cube of space is decomposed into eight equal volumes. Each volume (octant) may be split if it is not homogeneous, giving rise to a tree representing the workspace. Homogeneous nodes in the tree, called leaves, may represent empty space, where it is known that no object exists. The spatial representation is useful in computing free paths for trajectory analysis, and for answering questions about the identities of objects or the features in given locations. A spherical octree structure is a generalization of the rectangular octree structure. This data structure di-

Authors	Sensor Data	Fusion Type
[Kent et al. 1986]	multiple visual views	least squares
[Shekhar et al. 1986]	tactile - centroid	weighted least squares
[Matthies and Elfes 1988]	sonar - stereo range	Bayes law
[Stansfield 1988]	stereo - tactile	guiding
[Crowley 1986]	multiple range sensors	abstract
[Chen 1987]	multiple visual views	spherical octree

Figure 5: Representation methods

vides a solid sector or the whole spherical shell in the world space into eight octants. There is the initial radial distance from the viewing position which limits the universe to a solid sector or the whole spherical shell. This representation is rotation invariant. The occupancy grid is a 2-D array of cells that contains probability values which denote the chance of the cell containing an object or part of an object. If a value in the grid is high then an object is probably occupying that space.

A simple method for fusion of representations obtained from multiple sensors is to use Bayes law directly, as is done by [Matthies and Elfes 1988]. Two papers use the least squares [Kent et al. 1986], and weighted least square techniques [Shekhar et al. 1988] for fusion. Transformations are defined by relating the sensor outputs with object models, and the least squares fit is used to estimate the unknown parameters.

[Kent et al. 1986] build a representation from fusion of multiple views. They consider a robot operating in a constrained environment (e.g., a metal working factory cell). An outside knowledge base supplies CAD descriptions of objects to be expected in the world. The system uses this information to build an internal representation which consists of an *octree* to hold information about the environment's volume and a *feature based* representation to hold recognition information about the volumes. Each octree volume contains a link to feature based information about that particular volume. In the operational mode, the system attempts to register the internal representation with the world. This is done by predicting features from the internal representation which are expected to be present in the scene. The predicted features are matched with the actual features present in the image. This information is used in a least squares technique to register models with features found in the scene, resulting in updated position and orientation information about each object. This information is then used to update the octree representation. The final octree can then be used to allow for robot *path planning* and for computing exact object location in 3-D space. The system is also able to deal with the unexpected objects. The 3-D information related to these objects is obtained from multiple views of the same object by sensor motion.

[Chen 1987] presents a spherical octree model that allows sensor data to be added to the octree over time. The octree contains information about landmarks and obstacles in the environment. As a new viewing position occurs, a new position in the octree is modified, thus fusion is accomplished. As more data is fused a better path can be found. The spherical octree is very useful for looking at a 360 degree field of view around the moving robot. A good path of navigation can be found by intersecting a cone of base diameter equal to the diameter of the robot with a portion of the spherical octree. If no objects are present in the intersection then a obstacle-free path exists.

[Shekhar et al. 1986, Shekhar et al. 1988] present a method for matching the orientation and position of objects in the scene with objects in a database using multiple sensors. There are six degrees of freedom (rotations and translations in three dimensions) between the object model and the scene object that need to be determined. Each sensor provides a measurement and an error tolerance that determines the possible error present. An estimate of orientation is obtained by taking several sensor measurements of vertices and by computing an error vector. The magnitudes of the error vector are used as the weights of a weighted least squares method. The position error is also solved by a weighted least squares method by considering distance measurements taken from the sensors. Fusion occurs in the weighted least squares method. Since all sensor measurements are used to constrain the orientation and position parameters, sensor measurements that are more reliable will be assigned a higher weight. The authors chose to use a tactile array and two centroid sensors in their experiments. All sensors are mounted to a robot gripper, where the position of the sensors is known. However, the authors state that *exact* location of objects that come in contact with the sensors is not possible since mechanical errors are present and errors occur when commanding the gripper. The tactile sensor provides endpoints of tactile contact edges and two centroid sensors provide two vertex points each.

[Matthies and Elfes 1988] use *occupancy grids* to represent the space around a robot that is occupied by objects so that obstacle avoidance may occur. The occupancy grid is a 2-D array of cells that contains probability values which denote the chance of the cell containing an object or part of an object. If a value in the grid is high then an object is probably occupying that space. In the current implementation sonar and stereo sensors are used. For each sensor, a separate occupancy grid is used; the two grids are combined to form an overall world grid. When new information is gathered, each cell in the grid is updated by combining the new value with previous values using Bayes law. Note that temporal (time) updates are made to the grid. A sonar map is built by using the occupancy grid of cell size from 0.5 ft. to 1.0 ft. depending on the expected world size. Two stereo cameras are also used to provide sparse range data.

The cameras are nearly in registration with the sonar sensors. Stereo correspondence is achieved by using the Ohta-Kanade dynamic programming stereo algorithm. The two occupancy grids are are fused by using the same Bayesian approach.

[Stansfield 1988] uses a vision and a tactile sensor in order to generate object models. The vision sensor first provides orientation and position information, and the tactile sensor provides feedback as to the manipulation aspects of the objects. The visual system is composed of two binocular cameras. The edges are found in both images and a simple scan-line pixel matching algorithm is used for correspondence in order to determine sparse 3-D data. At the same time a 2-D region-based segmentation is performed which uses similar gray scales to group pixels. The information obtained from both the 2-D segmentation and 3-D edge information is used to eliminate edges which are not part of the region edges. This reduces the error and effects of noise. An extensive set of tactile features are then extracted by using specially coded robot algorithms. These features are roughness, elasticity, contours, edges, and corners and are determined by algorithm experts. For example, compliance is the amount of movement of a surface when a force is applied to it. The algorithm expert works by exerting a force F on the surface and by recording the initial position of the sensor. The robot then exerts a force $2F$ and again records the position. The linear distance between the two measurements is termed the compliance. The author uses the visual information to gather rough orientation and position parameters about the objects so that the tactile sensor can be *guided* to that location. The visual data can provide information as to how the tactile sensor should explore the surfaces so as to gather the most knowledge. For instance, a series of algorithm experts determine the rough shape properties and the tactile experts can determine surface properties and more precise shape descriptions. The final output of this system is a hierarchical shape representation of the object which can also include other features such as elasticity, roughness, etc. These extra features may aid in object recognition. The tactile sensor can only determine features for the points on the surface which are visible to the camera since vision must guide the tactile sensor. Perhaps this can be remedied by using a second camera at another viewpoint, but this has not been explored by the author.

[Crowley 1986] uses several range sensors in order to construct a 3-D surface model of the objects encountered. The models are defined by 3-D generalized surface patches and generalized contours. The generalized surface patch is defined by position, surface normal, size, velocity of patch, and a list of bounding contours. The uncertainty in surface normal and confidence in the patch is also included. When two patches intersect, a generalized contour (boundary line) is found. A confidence is defined as the ratio of the number of supporting votes to the total number of votes about a patch. This confidence can be updated by considering the possible errors in the boundary line between patches. The boundary line includes a cylinder of uncertainty around it which is based upon the error in intersection of the patches that form the contour. The patches and contours and their associated uncertainties are updated by using new sensor data. First, the patches and contours are computed, then the uncertainties are refined by using the new data that is shown to be consistent. Data which is contradictory is used to increase the uncertainty in the existing patches. Any patches or contours whose uncertainty is very high will be removed.

5.3 3-D Shape

The papers in this category deal with methods for computing the depth information by using multiple sensors. [Krotkov and Kories 1986] fuse focus and stereo ranging at the lowest level using a Bayesian approach. Their method is composed of explicit steps of consistency checking and verification of one sensor output with the other one. In the remaining papers the data is fused at the intermediate level. For instance, [Heeger and Hager 1988] combine optical flow and camera motion parameters to compute depth using maximum likelihood. While [Shaw et al. 1988] fuse microwave radar data and surface orientation obtained from a visual image, and [Wang and Aggarwal 1987] combine surface information obtained from occluding contour and light striping data.

[Heeger and Hager 1988] fuse optical flow data and camera motion parameters in order to obtain consistent object motion and depth information. They develop a linear equation that relates image velocity (optical flow) to camera motion. Thus the optical flow for a pixel is given by a single point and the camera motion provides a line in the optical flow space. The slope of the line will provide distance information for that measurement. If the point is on the line then consistency is achieved. However, this is rarely the case because of moving objects in the scene. Both sensors are modeled by contaminated Gaussian noise distributions, which assumes that noise corrupts the measurements. The *maximum likelihood* method is then used to fuse both sensor measurements. From the camera parameters, the optical flow estimate can be derived and thus the motion distance is computed. The Mahalanobis distance between measurements is often used to decide if the fused value is consistent with other measurements. In this case, the Mahalanobis distance is minimized in order to determine the most likely estimate.

[Henserson et al. 1988] apply a logical sensor system (see Sensor Modeling subsection) along with specialized algorithms to locate 3-D structure in the scene. The experiments are performed by using several views of a vision camera. The authors assume that the combination of several views from the same sensor is a form of multi-sensor fusion. They discuss several ways to determine 3-D structure. For example, if two lines are known to be perpendicular, as is the case with many corners, then only one camera view with three perpendicular lines and one other line is necessary to directly solve for 3-D structure. However, for more complicated cases such as non-perpendicular lines, up to three views of lines are necessary to provide enough information to solve for 3-D structure. It is assumed that angular invariance holds, (i.e., angles between 3-D lines (edges) of rigid objects do not change). The interpretation of 3-D data is encoded, as algorithms, into specialized structures (logical sensors) which can be fired depending upon the required actions. These

Authors	Sensor Data	Fusion Type
[Heeger and Hager 1988]	camera motion - optical flow	maximum likelihood
[Henderson and Shilcrat 1984] [Henderson and Hansen 1986]	multiple visual views	angular invariance
[Shaw et al. 1988]	microwave radar - vision	guiding, minimization
[Krotkov and Kories 1986]	focus ranging - stereo ranging	verification
[Wang and Aggarwal 1987]	light striping - occluding contours	rules

Figure 6: Methods for determining 3D shape information

structures use linear equations to solve several simultaneous equations; each of which provides one unknown. The method of propagation is also used. This allows the use of known line and feature configurations for use in determining orientation for other lines in the images. For instance, if there are two lines in the scene and the orientation of one of them is known, then three views are necessary to determine the orientation of the other line.

[Shaw et al. 1988] use a microwave radar system which finds range and Radar Cross Sections (RCS) and a vision sensor for robots in space. The visual data helps to convert the RCS information into spatial data. Thus fusion is used to guide the conversion of RCS data into a complete surface model. For the visual images, the occluding contours are extracted from a thresholded image and sparse surface normals are extracted by using photometric stereo or shape from shading. For the microwave image, a range estimate and polarized radar scattering cross sections are found. A minimization procedure is then used to minimize the error between the predicted and sensed RCS's. The output expected is a more accurate shape description.

[Krotkov and Kories 1986] use focus and stereo ranging to find the distance of objects in the workspace. An uncertainty measure is derived for each sensor which is a function of the actual range value. Both of these ranging methods have greater errors for large distances. A maximum likelihood method is used to determine the proper range value by using two independently computed range values and their variances. Generally, since focus ranging has a higher accuracy, a higher weight is applied for all focus ranging data. After the stereo data is retrieved, the focus ranger attempts to verify the range value returned by the stereo. A similar procedure is applied to verify the focus ranging values. If the two measurements prove to be inconsistent then they are not used in the fusion stage. Consistency is defined as: $|\frac{Z_1-Z_2}{\sqrt{(\sigma_1^2+\sigma_2^2)}}| < T$ where Z_1 and Z_2 are the two range measurements, σ_1 and σ_2 are the standard deviations of the measurements and T is some consistency threshold. This method will probably not work in "real-time" since each camera must be focused while the range value is retrieved. Also both range data sources provide very sparse data which may not be acceptable for very detailed analysis. However, for verification of sensor data, registration may not be necessary since both sensors are only able to extract very broad feature range values.

[Wang and Aggarwal 1987] determine surface descriptions by using occluding contours and a structured light approach. A transparent regular grid is placed in front of the light source so that the grid is cast upon the scene. This grid can be used to determine sparse 3-D information about the scene objects. Multiple views are taken in order to get complete 3-D data about the objects. Back projection - calculation of camera parameters and viewpoint from known scene points and information - is used to construct bounding volumes from intersecting views. The 3-D structures obtained from occluding contours and striped coded images are fused using the following qualitative rules: (1) If the stripe coded image is unavailable in a particular direction then the structure predicted by the occluding contour is used, (2) If only one partial surface structure (striped light) and one or more contour generating lines are intersected in the same direction then structure predicted by the the stripe coded image is chosen, (3) If more than one partial surface structures (striped light) is intersected by the radial sampling line, then the average of partial surface structure is used.

5.4 Sensor Modeling

Modeling of sensor characteristics is very important for a multi-sensor fusion system. Sensor measurements in general are imprecise and contain errors and uncertainties. The measurement error can be approximated by a probability distribution. The Gaussian distribution is commonly used. The estimate of various distribution parameters like mean and variance-covariance matrices is needed in a sensor fusion system. [Durrant-Whyte 1988] employs the summation of two Gaussians to model uncertainty in the sensor measurements. [Porrill 1988] discusses a method for updating the variance-covariance matrix. A simulation module which allows a multi-sensor system to be modeled and tested before construction is very useful. [Henderson and Shilcrat 1984, Henserson et al. 1987, Henderson and Hansen 1986], in a series of reports, have advocated the use of a logical sensor system to simplify simulation. Another important step in a multi-sensor-multi-data configuration is the problem of correspondence or registration; [Fernandez 1985] discusses several algorithms to solve this problem.

[Henderson and Shilcrat 1984] and [Henderson and Hansen 1986] propose logical sensor systems to *standardize* the use of sensors. A logical sensor is defined by four components: the name of the sensor, the number and type(s) of output (eg. real numbers, integers), a selector which is used to allow the sensor to be used, and a list of other methods and sensors which may be used to create the same output as this sensor, in case of failure. When a logical sensor system is found to be faulty, an alternate

Authors	Sensor Data	Fusion Type
[Henserson et al. 1988]	abstract	logical sensors
[Luo et al. 1988]	abstract	maximum likelihood
[Porrill 1988]	stereo	update covariance matrix
[Durrant-Whyte 1988]	geometric sensors	maximum likelihood
[Fernandez 1985]	abstract	maximum likelihood
[Harmon et al. 1986]	abstract	guiding, deciding, averaging
[Huntsburger and Jayaramamurthy 1987]	4 visual views	Shafer-Dempster

Figure 7: Approaches to sensor modeling

strategy is invoked which can retrieve the same information with comparable accuracy. An acceptance test is used to determine if the new information is acceptable. If not, then another strategy is used. If none is found then an error is returned. A simulation module is designed that will allow a multi-sensor system to be modeled and tested before construction. A sample simulation is presented that tries to recognize and inspect parts. A CAD based representation is used to model the objects. The Multi-sensor Knowledge System is trained by the CAGD system which provides various views of each object. From the multiple views, features are extracted. Once the features are found, a part detection algorithm for that object is made by the system. The algorithm determines which sensors will be used and how they will be used.

[Luo et al. 1988] use a Gaussian *probability density function* to model errors in measurement of sensors. The sensor distributions (as computed by sensor noise models) are compared by computing the "distance" between the probability distribution functions. The distance is considered to be a consistency check between two sensor measurements. If the difference is very large then data from one of the sensors is considered to be inconsistent with the other sensors and thus is not used for fusion. Next, the distance matrix is formed by taking all possible sensors and computing the difference in probability distribution functions for all possible fusions. The matrix is simply an ordered set of distances between two probability distribution functions. A *relation matrix* is formed by taking a binary threshold of the difference matrix. If the resulting matrix contains a 1 in the i, j position then there is supporting evidence that sensor i contains accurate data. The supporting evidence was found by sensor j. The matrix is considered to be a graph where 1's are nodes and edges are placed where 1's are adjacent. The largest connected subgraph (clique) will correspond to a group of sensors that provide the most accurate representation of the environment and the data that those sensors provide will be used in fusion. This method considers two levels of fusion. The first is a consistency that is applied between all of the sensors. If a sensor supports and is supported by other sensors then it will be used for fusion. The second is the actual fusion of data which is a performed by using a *maximum likelihood* method and some modified form of maximum likelihood. The authors do not state which method provides the best fused value, but it is assumed that either one will work well.

[Porrill 1988] deals with determining the exact value for a sensor measurement by considering errors involved with sensor calibration, actual feature location, and the actual sensor measurement of that feature. Typically the combination of these three parameters constructs a closed (constrained) system. Thus if one parameter contributes some error to the system then the other parameters would need to be adjusted in order to maintain consistency. If one parameter has errors then there is no way to determine the correct constrained parameters unless some other information is available, namely the covariance matrix and expected values of the measurements. In an iterative method, a better estimate for the parameters is obtained. At the same time the covariance matrix is updated to reflect the new parameters. The iteration continues until the mean squared error is minimized. Thus if the initial covariance matrix can be computed for the particular geometric sensor data then multi-sensor fusion can be obtained. The author works with two stereo images for which the covariance matrix is computed by considering camera projection errors and disparity errors.

[Durrant-Whyte 1986][Durrant-Whyte 1988] uses a summation of two Gaussian distributions to model uncertainty in sensors that provide geometric data such as lines, surface orientation, centroid, etc. For each observation made of the environment, a description vector, which contains information about the *positions* and *orientations* of objects, is added to the system. One of the Gaussians is found from the sensor noise and accuracy characteristics and the other is not known exactly. This is called a *contaminated Gaussian distribution*. A mean of the measurements is calculated and any measurement that deviates very far away is discarded and not used for fusion. The new measurements now belong to a pure Gaussian distribution. The fusion of the new measurement set is performed by a *maximum likelihood* method. Consistency checking is performed by using a differential matrix. The entire world (scene) is thought to be a closed system in the sense that the combination of measurements must produce a fixed interpretation. Thus if we have two measurements and we determine that one has changed, then we must change the other by an appropriate amount. However, this becomes a little more difficult when a large number of measurements have been taken. The problem is then reduced to solving a linear system which results in a minimization of a constrained objective function which provides a solution to the consistency problem.

[Fernandez 1985] considers a multiple sensor, single property, multiple data configuration and addresses the problem of data association or registration. It is assumed that noise in data from the sensors follows a Gaussian distribution. Several algorithms are presented that will match data points from one sensor with data points of other sensors. One algorithm is a *maximum likelihood* estimate that measures the similarity among data vectors of different sensors. The *Mahalanobis* distance, as discussed earlier, is used to determine the amount of difference that occurs between two measurements. This algorithm is not good because it assumes a one-to-one match between data vectors. If two sensors provide different information or missing information, then one-to-none matches are not possible. Another method is called *Sensor to Universe* mapping. This involves creating a matrix which contains a union of data observation vectors. This is accomplished by forming a matrix for each sensor where the column vectors are the data observations and by placing the matrices side by side to form a super-matrix. The super-matrix is then multiplied by orthogonal transformations to produce independent and dependent columns. It is known that the column vectors of each sensor should be close to being linearly dependent if they are data observations of the same scene. The theoretical research has been discussed but no real world results of these methods are presented.

[Harmon et al. 1986] use a distributed blackboard to fuse data where the environment is modeled as a set of objects in which each object has a set of properties. Each property contains a single data value which has an associated confidence, error, and time measurement. The confidence measure indicates how accurate or correct the measurement is thought to be by the sensor. If the confidence is very high for a measurement then the system can consider using it in making decisions about the environment. The time measurement denotes the time at which a sensor reading was taken. If each sensor contributes independent information then the data may easily be added to the system blackboard with no conflicts. However, when sensor observations are of the same property value (i.e., not independent), then merging of data must occur in the blackboard. They propose to use *averaging*, *guiding*, and *deciding* for fusion. The blackboard is designed into a shared system memory. It holds all of the objects, properties, and confidence values. The fusion takes place when new data is added from the sensor systems.

[Huntsburger and Jayaramamurthy 1987] integrate sensor data to provide shape information. Edge maps and segmentation are performed on a sequence of four frames of each registered sensor image. Preliminary shape and motion characteristics of objects in the images are computed. The preliminary information is then improved by using the multiple sensor information. Dempster-Shafer theory is used to allow each sensor to provide an expert opinion about the objects that are present. This allows data to be intersected.

5.5 Autonomous Robots and Navigation

The paper by [Ruokangas et al. 1986] is a good example of sensor integration for an autonomous robot. Integration is used in the Automation Sciences Testbed (ASTB). They use vision, acoustic ranging, and force torque sensors in a controlled robot workcell. They consider a task of acquiring bolts from known positions and inserting them into holes in an arbitrarily shaped object placed at arbitrary locations in the field of view of the camera. The authors demonstrate one and two sensor configurations for this task and outline the possible problems. For instance, if vision alone is used to locate objects without any height information, the acquired images might be out of focus, and the location information will be only 2-D. Therefore, adding the second sensor, an acoustic ranger, in the configuration can provide the distance from the camera to the objects and can be used to correctly position the camera. While the vision system provides scene gauging and object location, the acoustic ranger provides the data to determine the camera's correct focal distance, and hence, the vision system's gauge scale. The system is further augmented by incorporating a third sensor, a force-torque sensor, to provide real-time modification of a globally determined hole position. The force-torque sensor which is mounted on the robot arm sends three orthogonal forces and torques to the real-time trajectory modification software.

[Barnes et al. 1983] describe a system which integrates multiple sensors for an autonomous robot. They use vision, tactile, and proximity sensors, and consider simple tasks like *pick and place*, in which an *Objecta* is moved from *Pointa* to *Pointb*. The steps in this task include getting the location of object A by using far vision (overhead camera), moving above object A using a capacitative proximity sensor, fine-tuning the location of object A by using near vision (an on gripper camera), and then the opening jaws of the gripper. The theme of this paper is a design of a knowledge based system for monitoring robot tasks by checking various expectations against what is actually being perceived. The system is composed of three parts: the task knowledge which is represented in Minsky's frames, the agenda and the history stack. Each frame has many slots which are filled as new sensor data arrives. Within the task knowledge are action frames, instruction frames, perception frames, sensor frames, and object frames. Action frames hold all actions that need to be performed. Instruction frames hold all programming instructions that the robot controller should execute. The perception frames deal with perceiving errors that occur dynamically. The sensor frames deal with sensor limits and accuracies. Object frames hold information about objects that are manipulated during the task. The *agenda* handles placing the frame that will perform the next subtask onto a list. Once the correct frames are selected, a sensor frame is initiated that will use the correct sensor to gather more data. As frames are acted upon and results are validated, the frames are written to a *history stack*. The system can be sensor driven if desired. New sensors can be added easily with only a change in programming.

[Flynn 1988] uses both a sonar and an infra-red distance sensor in order to build a map of the surrounding environment. Each sensor is modeled based upon its operating characteristics and data is collected by scanning the scene and converting the cylindrical data coordinates into cartesian coordinates. Several rules are supplied which delete or modiy the data when certain distance values or depth discontinuities are present. These rules are: (1) If the infra-red sensor detects a very large distance discontinuity and the sonar reading is less than 10 feet, then accept the fact that a discontinuity is indeed present. (2) If the sonar reading

Authors	Sensor Data	Fusion Type
[Ruokangas et al. 1986]	vision - range - force	guiding
[Barnes et al. 1983]	vision - tactile - proximity	guiding
[Flynn 1988]	sonar - IR distance	rules
[Luo et al. 1988]	sonar - vision	guiding
[Shafer et al. 1986]	sonar - stereo - range	guiding
[Giralt et al. 1985]	sonar - vision - range	guiding
[Tutk et al. 1987]	color - range	guiding

Figure 8: Approaches to Autonomous robots and navigation

is larger than the infra-red sensor's maximum range, then use only the sonar reading. (3) If the sonar reading is at its maximum, then expect the actual distance to be larger than the sonar indicates. The rules allow both sources of data to provide a better estimate of the actual scene edges. The combined information is transformed to an intermediate representation called the curvature primal sketch, and then is converted into a polygonal representation of the world suitable for path planning.

[Luo et al. 1988] use a visual and an ultrasonic sensor for grasping objects with a robot arm. The robot workspace consists of a variable speed conveyor belt which serves as the manufacturing cell. Thus, the system must be able to track objects as they move down the conveyor. The vision sensor is used to describe the conveyor belt objects, while an ultrasonic sensor is used to find the distance from the camera to the objects. This distance is used to convert image coordinates into true distance quantities in inches. Integration is being performed because the ultrasonic sensor is used only for depth information in order for the vision system to determine size characteristics. The objects on the conveyor are assumed to vary only by a translation from one object to the next. Thus the velocity of the objects and the conveyor can be determined solely from a pair of intensity images by a simple optical flow computation. Once motion is detected on the conveyor, the end-effector may begin to move toward the object and calculate a trajectory in order to intercept the object.

[Shafer et al. 1986] use a stereo camera system, a laser range sensor, and sonar sensors to control the navigation of a truck. An estimate of the position of the truck is maintained over time. A local map database controller is used to coordinate the various subroutines that are used by the sensor modules. In this way, each subroutine can be executed in parallel. Two-dimensional information is used to locate edges of the roads; then 3-D sensors are used to scan the area between the edges of the roads to find any obstacles. This is integration and is occurring at a low level. Integration is also used when 2-D sensors are used to capture data about far away objects when the range sensor is not usable or is inaccurate.

[Giralt et al. 1985] use fourteen ultrasonic sensors, a video camera, and a range sensor to detect obstacles for robot navigation. The space in which the robot works is represented in two levels: the topological level and the geometrical level. The topological level is represented by assigning places (objects) to nodes and arcs represent connections of places. The geometric level adds lengths to the arcs which represent distances between places. In order to move from one place to another, the robot follows a path on the connectivity graph. A graph search may be used to find the shortest distance or smallest cost path. The laser range finder is used first to get an initial outline of the perceived space. Then the robot moves to areas where no information is available to gain more knowledge to add to the graph. All objects in the space are considered to be obstacles, thus the video camera looks for obstacles, then the range finder gathers surface orientations and positions in order to construct the connectivity graph. The ultrasonic sensors are used to detect very close obstacles which are less than two meters away so that close information can be added to the graph. If the robot determines that an object is movable, then it does not have to be inserted in the graph since the ultrasonic sensors can detect it upon encountering the object. As can be seen this is classified as an integration method.

[Tutk et al. 1987] use both color camera and laser range sensors to build a description of the environment. A *reasoning* system computes an obstacle-free path to follow on the road. The reasoning system converts sensor data into world coordinates by using time information associated with the data measurement and simple edges for the road, and it predicts the location of the road in future images. A *navigation* system plans the trajectory and the *pilot* system executes the trajectory. The images are segmented into road and non-road regions by using a line of a given slope to separate road from non-road in a red and blue feature color space. The slope can be pre-determined by considering seasonal or environmental changes. Road regions are separated from the background by computing a threshold by sampling certain pixels where the road was predicted to be in the images (normally the lower portion). This type of segmentation amounts to a red minus blue segmentation. The edges of the road are then found by using boundary tracing and then once both sides of the road are found a scene model is constructed by using small portions of the edges. Three dimensional information, from the laser range sensors, about the road is used to form a trajectory for path following.

5.6 Recognition

One of the important goals of a multi-sensor system is to be able to recognize objects from its sensory inputs. Object recognition is a well developed area of research in computer vision, and there are a number of approaches for recognizing

Authors	Sensor Data	Fusion Type
[Rodger and Browse 1987]	vision - tactile	consistency
[Bajcsy and Allen 1985]	vision - tactile	guiding/filling in
[Luo and Tsai 1986]	vision - tactile	hierarchical
[Magee et al. 1985]	vision - range	guiding
[Garvey et al. 1983]	frequency - pulse width	Shafer-Dempster

Figure 9: Methods for recognition

objects using vision [Lowe 1985], range [Reeves and Taylor 1989], and tactile [Hillis 1982] sensors. The aim of using multiple sensors for object recognition is to decrease feature ambiguity and to reduce the search space during matching. We have been able to find only four papers dealing with object recognition using multiple sensors. Three of these papers demonstrate methods by using real scenes. [Rodger and Browse 1987] use visual and tactile features to recognize objects in the synthetic scenes, however, they do not distinguish among features from different sensors. They use positional and placement constraints to reduce the number of possible interpretations.

[Luo and Tsai 1986] use a straightforward two stage method for recognizing objects. First visual features are used to discriminate objects, then tactile features are employed, if necessary. [Allen 1987][Bajcsy and Allen 1985] use stereo vision to guide a tactile sensor, and then the tactile sensor is used to fill in 3-D data where the stereo has obtained information. [Magee et al. 1985] use range and intensity data. All of the above approaches consider sensor output which is a two dimensional array. The paper by [Garvey et al. 1983] uses Shafer-Dempster theory to recognize the emitter types by using one dimensional signals of pulse width and RF frequency.

[Rodger and Browse 1987] fuse visual and tactile data to recognize and locate the positions of objects. Objects are modeled as gravitationally stable polyhedra that are assumed to rest on a plane. This allows the objects' positions to be referenced by a single rotation (around the z axis) and two translations (on x and y axes). For vision, the features are simply *straight line* segments. For tactile sensing, the features are corner, edge, or flush contacts between the object and the sensor. Interpretations about the object are made for each sensor feature and other features are examined to allow the set of possible interpretations to become smaller. A consistency procedure is applied after each feature is examined to insure that all previous features still lend themselves to the interpretation found. The use of more than a few models will drastically slow the system since many matches will be required. Thus, the search space must be reduced if quicker searches are to be made. Another deficiency is that if one sensor failed or gave incorrect data then the system may be unable to form a correct interpretation of an object since consistency would be reduced.

In [Allen 1987][Bajcsy and Allen 1985], Bajcsy and Allen perform integration of vision and tactile data. Two cameras are used in stereo to produce sparse depth information. Three dimensional models of all objects are created and stored as surface patches in the database. Each intensity image is segmented and a stereo matcher is used to compute sparse depth information for the scene. A tactile sensor is used to provide information in places where the stereo depth information is lacking. To integrate both vision and touch information, very simple surface patches (first-order equations) are computed that approximate the surfaces in the scene. The vision guides the tactile sensor to gather more data points, usually on the interior of regions. The tactile data is then used to compute higher order surface approximations with the help of the extra data. The surface patches that are created are matched with the model database in an attempt to recognize the objects in the scene.

[Luo and Tsai 1986] use tactile and visual information to recognize objects. The intensity image is an overhead view of the object to be recognized. If all objects can be discriminated from one another by using the intensity data then no further processing is needed. Otherwise, as many tactile images as needed to discriminate between all of the objects are used. The visual features are *perimeter*, *eccentricity*, and *moment of inertia*. The tactile features are *perimeter*, *centroids*, and *direction of principal axes*. A robot gripper holds two tactile sensors; one on each side so that both faces of the grasped object are in contact. A decision tree is used to match 2-D object database features with actual 2-D image features. The first stage in the tree will always use visual features and all other levels will use tactile features. Thus the intensity data is used only in the first step since it is a high resolution sensor. Then tactile images taken from different orientations are used to gather other features. This type of processing is *sequential* and can be called an integration method. The use of *moments* in feature based recognition is not applicable to a robust recognition system since occlusion will produce inaccurate calculations of moments.

[Magee et al. 1985] use orientation invariant features such as circles and intersection of lines for matching with the object database. Range data is used only to capture 3-D data about selected points on the circular features that are found in the intensity data. In a sense, the intensity data is guiding the use of the range sensor. A graph is constructed where the nodes in the graph represent features and the arcs represent distances and orientations between the features. A matching ratio is computed for each of the scene graphs versus the model database graph. The ratio considers the number of arcs that match in length and the total number of arcs in the graph. The largest ratio is chosen as the best match in a greedy approach. The matching ratios of the best matches for each object are added together and divided by the number of unknown matches. The maximum of this quotient is chosen as the correct match and that model object is chosen as the object in the scene. The authors have considered only single pose object matching, (i.e., translation and rotation of objects is not allowed). This is a *major* restriction. Also, the use of range data is very limited since it is used only to find 3-D information about selected points in the scene.

[Garvey et al. 1983] use Dempster-Shafer theory for integrating knowledge from disparate sources. They consider a situation in which five emitters transmit one dimensional signals which are characterized by frequency and pulse width. These sensory measurements contain uncertainties. Given such a signal with frequency and pulse width characteristics, the aim is to determine which emitter produced such a set of signals. The first step is to convert sensory measurements into probability mass (belief) distributions, and then combine the two by using Dempster's orthogonal sum. Next, for each sensor the support and plausibility are computed from their combined mass distributions.

6 Summary

We have examined papers which describe various approaches to multi-sensor fusion. The sensors which have been employed in a multi-sensor environment include a video camera, tactile sensor, range finder, sonar, infra-red sensor, and torque sensor. The researchers have investigated the use of multiple sensors for scene segmentation, object recognition, autonomous robot navigation, building 3-D representation, and simulation and modeling of multi-sensor systems. Strategies for combining sensor measurements include Bayes law, maximum likelihood, Dempster-Shafer, logical *and*, set intersection, weighted least squares, Hough transform, guiding, integration, deciding, verification, and global consistency.

7 Future Work

It has become obvious from this survey that the current state of the art in multi-sensor fusion is in its infancy. There are, therefore, promising areas of future work in almost all categories discussed in this paper, and other related topics to multi-sensor fusion. One of the most important areas which will have a significant impact on the research in multi-sensor fusion is in sensor design. The majority of currently available sensors are slow, less robust, and expensive. Due to high cost of sensors (e.g. range finder), very few laboratories are equipped with more than two sensors. This scenario is reminiscent of vision research ten years ago, when the cameras and digitizing equipment were beyond the reach of every institution. Now, inexpensive cameras and digitizer boards for PC's with high resolution monitors are available at an affordable cost, which has made vision research a wide spread activity. Therefore, it is expected that the situation related to the availability of other sensors (range, infra-red, etc.) will improve in the future, and more research groups will be involved in multi-sensor fusion research. Another related issue is the availability of registered sensor data, for example registered intensity, thermal and range data, which will be useful for fusion at the lower level to achieve scene segmentation. In the future, with the availability of registered data we will experience an increase in the research activity in the segmentation area.

Uncertainty management has been active in the past, and will remain popular due to its mathematical elegance. The validity of sensor uncertainty models need to be justified, and robust methods for approximating the model parameters (e.g. variance-covariance matrix) need to be explored. In the past the effort in uncertainty management has mostly been limited to fusion at the lower level (depth values, or position and orientation vectors). In the future we will see more work completed on fusion at the feature level. The features in one sensor's data need to be interrelated to the features in another sensor's data. This will also necessitate a design of generalized representation techniques which can be employed for multiple sensors.

Another important area where multi-sensor systems can really make a difference is in object recognition. Surprisingly, very little work has been done on this topic. Multiple sensors not only can provide multiple views of objects, but they can also impose more constraints to reduce the search space during matching, since each sensor is sensitive to different a modality. For certain features there will be a direct correlation between sensors. A line segment in the visual image, for instance, should match to an edge in the tactile image, assuming both sensors have similar view and range. A feature supported by two sensors should have precedence over a feature supported by only a single sensor.

Finally, the implementation of multi-sensor fusion systems in real time needs special architectures employing parallel processing. There are several promising areas for the future work including the work on: mapping the current sensor fusion algorithms to available parallel architectures, and implementation of the fusion methods in specialized hardware so that chips can be designed, and manufactured.

References

[Allen 1987] Allen, P.K. Kluwer Academic Publishers, Boston, 1987.

[Bajcsy and Allen 1985] Bajcsy, R. and Allen, P. The MIT Press, Cambridge, MA, pages 81-86, 1985.

[Barnes et al. 1983] Barnes, D.P., Lee, M.H. and Hardy, N.W. In *Proc. 3rd Int. Conf. on Robot Vision and Sensory Controls*, pages 457–465, November 1983.

[Brady 1988] M. Brady, editor. *Int. Journal of Robotics and Automation*. MIT Press, Dec., 1988.

[Cameron et al. 1988] Cameron, A., Daniel, R., and Durrant-Whyte, H. In *Proc. 1988 IEEE Int. Conf. on Robotics and Automation*, pages 1062–1067, April 1988.

[Chen 1987] Chen, S. In *Proc. of the 1987 Workshop on Spatial Reasoning and Multi-Sensor Fusion*, pages 201–210, October 1987.

[Crowley 1986] Crowley, J.L. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 1455–1462, April 1986.

[Duda et al. 1979] Duda, R.O., Nitzan, D. and Barrett, P. *IEEE PAMI*, 1:259–271, 1979.

[Duncan et al. 1987] Duncan, J.S., Gindi, G.R. and Narenda, K.S. In *Proc. of the 1987 Workshop on Spatial Reasoning and Multi-Sensor Fusion*, pages 323–333, October 1987.

[Durrant-Whyte 1988] Durrant-Whyte, H.F. Kluwer Academic Publishers, Norwell, Mass., 1988.

[Durrant-Whyte 1986] Durrant-Whyte, H.F. In *Proc. 1986 IEEE Int. Conf. on Robotics and Automation*, pages 1464–1469, April 1986.

[Fernandez 1985] Fernandez, M.F. In *IEEE*, pages 277–281, 1985.

[Flynn 1988] Flynn, A.M. *The International Journal of Robotics Research*, pages 5–14, 1988.

[Fowler 1979] Fowler, C.A. In*IEEE Trans. Aerospace Electronic Systems*, pages 2–10, Jan, 1979.

[Garvey et al. 1983] Garvey, T.D., Lowrance, J.D. and Fischler, M.A. *IJCAI*, pages 319–325, 1983.

[Gil et al. 1983] Gil, B., Mitiche, A. and Aggarwal, J.K. *CGIP*, 21:395–411, 1983.

[Giralt et al. 1985] Giralt, G., Chatila, R. and Vaisset, M. The MIT Press, Cambridge MA, pages 191–214, 1985.

[Hackett and Shah 1989] Hackett, J.K. and Shah, M. *Optical Engineering*, pages 667–674, June 1989.

[Hackett and Shah 1988] Hackett, J.K. and Shah, M. In *PROCIM '88 (The First Florida Conference on Productivity and Competitiveness in Manufacturing through Computer Integrated Manufacturing)*, pages 33–34, Nov, 1988.

[Harmon et al. 1986] Harmon, S.Y., Bianchini, G.L., and Pinz, B.E. In *Proc. IEEE 1986 Int. Conf. on Robotics and Automation*, pages 1449–1454, April, 1986.

[Heeger and Hager 1988] Heeger, D.J., and Hager, G. In *IEEE ICCV*, pages 435–440, 1988.

[Henderson and Hansen 1986] Henderson, T. and Hansen, C. Technical report, The University of Utah, Technical Report UUCS-86-114, September, 1986.

[Henderson and Shilcrat 1984] Henderson, T. and Shilcrat, E. Technical report, The University of Utah, Technical Report UUCS-84-002, March, 1984.

[Henserson et al. 1988] Henderson, T., Weitz, E., Hansen, C. and Mitiche, A. *The Int. Journal of Robotics Research*, 7:114–137, Dec. 1988.

[Henserson et al. 1987] Henderson, T., et al. Technical report, The University of Utah, Technical Report UUCS-84-002, February, 1987.

[Hillis 1982] Hillis, W. *The International Journal of Robotics Research*, pages 33–44, 1982.

[Hu and Stockman 1987] Hu, G. and Stockman, G. image. In *Proc. of the 1987 Workshop on Spatial Reasoning and Multi-Sensor Fusion*, pages 138–147, October 1987.

[Huntsburger and Jayaramamurthy 1987] Huntsburger, T.L. and Jayaramamurthy, S.N. In *Proc. of the 1987 Workshop on Spatial Reasoning and Multi-Sensor Fusion*, pages 345–350, October 1987.

[KaK and Chen 1987] A. Kak and S. Chen, editors. *Proc. of the 1987 Workshop on Spatial Reasoning and Multi-Sensor Fusion.* AAAI, Oct., 1987.

[Kent et al. 1986] Kent, E.W., Sheiner, M.O. and Hong, T. In *Proc. 1986 IEEE Int. Conf. on Robotics and Automation*, pages 1634–1639, 1986.

[Krotkov and Kories 1986] Krotkov, E. and Kories, R. In *Proc. 1986 IEEE Int. Conf. on Robotics and Automation*, pages 548–553, April 1986.

[Lowe 1985] Lowe, D.G. *Perceptual Organization and Visual Recognition.* Kluwer Academic Publishers, Boston, 1985.

[Luo et al. 1988] Luo, R.C., Lin, M. and Scherp, R.S. In *IEEE Journal of Robotics and Automation*, volume 4, pages 386–396, August 1988.

[Luo et al. 1988] Luo, R.C., Mullen Jr., R.E. and Wessell, D.E. In *Proc. 1988 IEEE Int. Conf. on Robotics and Automation*, pages 568–573, April 1988.

[Luo and Tsai 1986] Luo, R.C. and Tsai, W. In *Proc. 1986 IEEE Int. Conf. on Robotics and Automation*, pages 1248–1253, April 1986.

[Magee et al. 1985] Magee, M., Boyter, B., Chieu, C. and Aggarwal, J. *IEEE PAMI*, 7:629–637, November 1985.

[Matthies and Elfes 1988] Matthies, L. and Elfes, A. In *Proc. 1988 IEEE Int. Conf. on Robotics and Automation*, pages 727–733, April 1988.

[Mitiche 1984] Mitiche, A. In *Proc. IEEE First Conf. on Artificial Intelligence Applications*, pages 156–160, Dec. 1984.

[Mitiche and Aggarwal 1986] Mitiche, A. and Aggarwal, J.K. *Optical Engineering*, 25:380–386, March 1986.

[Moerdler and Kender 1987] Moerdler, M. and Kender, J. In *Proc. of the 1987 Workshop on Spatial Reasoning and Multi-Sensor Fusion*, pages 272–281, October 1987.

[Nahin 1977] Nahin, P.J. *Air Univ. Rev.*, pages 2–16, Sept. - Oct., 1977.

[Nandhakumar and Aggarwal 1987] Nandhakumar, N. and Aggarwal, J.K. In *IEEE ICCV*, pages 83–92, 1987.

[Orrock et al. 1983] Orrock, J.E., Garfunkel, J.H. and Owen, B.H. In *Proc. 3rd Int. Conf. on Robot Vision and Sensory Controls*, pages 419–425, November 1983.

[Porrill 1988] Porrill, J. *The International Journal of Robotics Research*, pages 66–77, 1988.

[Reeves and Taylor 1989] Reeves, A. and Taylor, R. *IEEE PAMI*, 11:403–410, April, 1989.

[Rodger and Browse 1987] Rodger, J.C. and Browse, R.A. In *Proc. of the 1987 Workshop on Spatial Reasoning and Multi-Sensor Fusion*, pages 13–20, October 1987.

[Ruokangas et al. 1986] Ruokangas, C., Black, M., Martin, J. and Schoenwald, J. In *Proc. 1986 IEEE Int. Conf. on Robotics and Automation*, pages 1947–1953. 1986.

[Shafer et al. 1986] Shafer, S.A., Stentz, A. and Thorpe, C. Technical report, Carnegie Mellon University, Technical Report CMU-RI-TR-86-9, April, 1986.

[Shaw et al. 1988] Shaw, S.W., deFigueiredo, R.J.P. and Krishen, K. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 1842–1846, April 1988.

[Shekhar et al. 1986] Shekhar, S., Khatib, O. and Shimojo, M. In *Proc. 1986 IEEE Int. Conf. on Robotics and Automation*, pages 1623–1628, April 1986.

[Shekhar et al. 1988] Shekhar, S., Khatib, O. and Shimojo, M. *The Int. Journal of Robotics Research*, 7:34–44, Dec. 1988.

[Siegel et al. 1986] Siegel, D., Garabieta, I., and Hollerbach, J.M. In *Proc. IEEE 1986 Int. Conf. on Robotics and Automation*, pages 1286–1291, April, 1986.

[Stansfield 1988] Stansfield, S.A. touch. *The International Journal of Robotics Research*, pages 138–161. 1988.

[Tutk et al. 1987] Turk, M.A., Morgenthaler, D.G., Gremban, K.D. and Marra, M. In *Proc. 1987 IEEE Int. Conf. on Robotics and Automation*, pages 273–280, March 1987.

[Wang and Aggarwal 1987] Wang, Y.F. and Aggarwal, J.K. In *Proc. 1987 IEEE Int. Conf. on Robotics and Automation*, pages 1098–1103, March 1987.

[Wong and Hayrapetian 1982] Wong, R.Y. and Hayrapetian, K. In *Proc. 1982 IEEE Computer Society Conference on Pattern Recognition and Image Processing*, pages 518–520, June 1982.

[Zuk et al. 1985] Zuk, D., Pont, F., Franklin, R. and Dell'Eva, M. Technical report, Environmental Research Institute of Michigan, Ann Arbor, Michigan, November 5, 1985.