Proceedings of Workshop on Foundations of Information/Decision Fusion: Applications to Engineering Problems
August 7-9, 1996, Washington D.C.

Multisensor Fusion for Sensory Intelligence in Robotics

Avi Kak and Akio Kosaka
School of Electrical and Computer Engineering
Purdue University
W. Lafayette, IN 47907-1285
kak@ecn.purdue.edu, kosaka@ecn.purdue.edu

Abstract

We will present case studies in the fusion of different sensory types on arm and mobile robots. We will first present a system in which an infrared heat sensor and ultrasonic sensors mounted on a mobile robot are used together to detect human intruders. In this case, multisensor fusion is necessitated by the fact that each sensor by itself is inadequate for the mission. The signals from the different sensors are fed into a neural network that makes the intruder/no-intruder decision. Our second case study concerns the use of multiple sensors in a robotic workcell for the purpose of recognizing and localizing complex objects using the smallest number of look angles. Our final case study will discuss a 3D vision system for fast object recognition in which range and color are fused in a manner that results in highly computationally efficient algorithms for object recognition.

1 Introduction

It is now clear that there exist many situations in sensor-based robotics in which the use of multiple sensors is unavoidable. Reasons for this are not hard to understand. Any given sensor yields information about only one of the many attributes of the environment and oftentimes information must be gleaned simultaneously about more than one attribute so that a robot can make sense of what it is looking at.

Once a decision is made to employ multiple sensors, the issue then becomes one of how to integrate the information. This issue can be more difficult than it sounds, especially when the sensors are disparate, as is often the case. In some cases, this issue of sensor integration is further complicated by the fact that objects in a robot's environment may need to be examined from different viewpoints. It would obviously be foolish to use all the available disparate sensors from all the possible viewpoints. The question then becomes one of how to decide what sensor to invoke from which viewpoint in order to most quickly decide the identity and the pose of an object in the robot's environment. And even after this problem is solved, there remains the more mundane problem of how best to account for the differences in the coordinate frames of the sensors when each examines the scene from a different viewpoint.

Even when a single viewpoint suffices for recognizing and localizing objects, one must come to grips with the algorithmic details of how to combine information from disparate sensors. When a sensor is used to measure a particular attribute of a scene, decision thresholds must be applied to the raw sensor output in order to determine the strength of the attribute. When multiple sensors are used, it is not clear whether or not there should be a relationship between the thresholds for the different sensors.

Another problem concerns how to combine the output of sensors that produce what could be referred to as the qualitative measures of some attribute of a scene with the output of sensors that produce quantitative information. For example, consider a sensor that tells us whether an object surface is planar, cylindrical, or spherical. Now consider another sensor that yields information on the albedo of each surface. The question then becomes one of how to best combine these two different types of sensor outputs into a single algorithm. One could perhaps treat the shape categories as discrete points along some attribute axis. But it is not clear if that is the best way to go.

Last but not the least, there is the problem of deciding whether the sensors should compete or cooperate with one another or work in some other negotiation mode in order to resolve expeditiously the hypotheses about the content of a scene.

These then are the main problems associated with sensor fusion in the context of sensor-based robotics. In what follows, we will discuss three case studies. We will first discuss briefly a system for detecting human

intruders. In this system, we had to combine the outputs of ultrasonic and infrared sensors for unequivocal recognition of humans and for the system to not get confused by air-conditioning ducts, walls, etc. In the next case study, we will discuss a system that takes into account viewpoints for deciding what sensor to invoke from which direction for recognizing and localizing an object in the least amount of time. Finally, we will talk about our latest system in which we have combined range and color information for recognizing and localizing 3D objects for robotic bin-picking.

2 Fusion of Ultrasonic and Thermal Signals

One of the applications for mobile robots is for security patrol, especially under after-hours conditions, often at night. Such robots must be able to detect human intruders reliably. Using vision for such an application is evidently not feasible. Alternative sensors that could be deployed include ultrasound, passive infrared heat sensors, microwave Doppler, etc. Each of these sensors has its own advantages and disadvantages if used only by itself. For example, ultrasound would pick up any object capable of returning an echo, and that would include both intruders and other stationary or not so stationary obstacles. So all by itself, ultrasound would not suffice. Infrared sensors would pick up a human intruder, but would be subject to false alarms caused by such ambient heat sources as heating vents, radiators, steam pipes, and other heat-emitting objects. Therefore, thermal sensors by themselves cannot do the job. Microwave Doppler sensors could be used but are extremely sensitive to the motion of the robot, since any such motions distort the Doppler signals returned by objects.

Given the shortcomings associated with each of the sensors if used individually, we initially decided to use all three of them in a cooperative neural-network based framework. Unfortunately, the sensitivity of the microwave Doppler sensor to the motion of the robot turned to be so great as to render this sensor useless. (We could have used this sensor at those instances when the robot was stationary, but we did not pursue that avenue of research as it ran counter to our goals.) So we ended up integrating just the two remaining sensors, but, fortunately, the end results were resounding successful, although limited by the maximum distance of 8 feet between the robot and the intruder.

While the details are provided in [1], we will recapitulate here some of the salient features of the integration of the infrared sensor and the sonar sensor for intruder detection. Shown in Fig. 1 (a) is a photo of the mobile robot, while the schematic in Fig. 1 (b) shows the various systems mounted on the robot. The infrared pyroelectric sensor and the ultrasonic sensor used for intruder detection experiments are mounted on the side as shown. Note that the ultrasonic sensor paired up with the infrared sensor is separate from such sensors in the semi-ring of ultrasonic sensors that is used by the robot for obstacle avoidance during routine navigation. The infrared sensor is a dual element pyroelectric detector, on the front of which is mounted is Fresnel lens that results in a narrow horizontal and tall vertical beam. This beam intersects all objects in the vicinity of the robot on its left as the robot passes the objects.

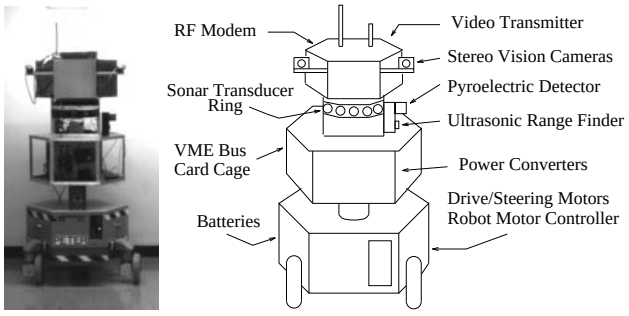


Figure 1: *(a) Photograph of the mobile robot. (b) Schematic drawing of the mobile robot.*

The energy output by the two sensors is integrated and fed into the two input nodes of a 3-layer feed-forward neural network trained by the backpropagation learning algorithm. The network structure, determined empirically, consists of two input units, two hidden layers each consisting of 6 hidden units, and one output unit in the output layer. In addition, three bias units [1, 4] are included in the network to represent the internal thresholds of the units in the hidden and output layers as shown in Fig. 2. The output unit going high corresponds to the detection of a human intruder.

The backpropagation learning algorithm is a gradient descent error-correction algorithm that minimizes the errors between the desired outputs and the actual computed outputs by modifying the connection strengths, or weights, between the units in the network [7, 8, 9]. The actual training process of the neural classifier involved 60 training patterns, on which 30 represented patterns corresponding the presence of

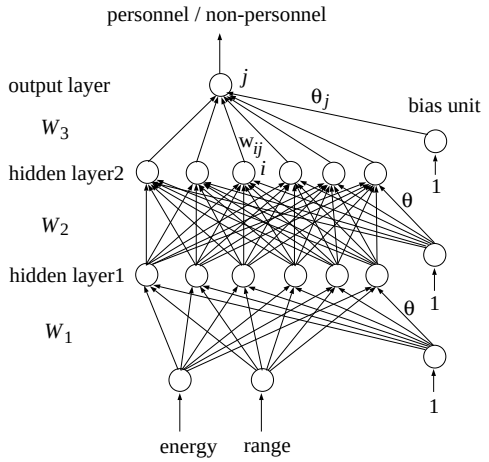


Figure 2: *The network diagram of the neural classifier.*

an intruder and 30 when no intruder was present.

As long as the maximum distance between the robot and the intruder was no greater than 8 feet, the system worked with 100% accuracy in detecting human intruders and 0% false-alarm rate.

3 Multi-sensor Fusion in a Robotic Assembly Cell

We will now discuss how sensory integration can be achieved in a robotic workcell if the goal is to recognize and localize 3-D objects in such a way that all available sensors and all available viewpoints are used optimally. The details are provided in [3].

The basic problem addressed in this section is this: Let's say we are provided a library of model objects that can be arbitrary but stable poses on a platform in a robotic workcell. Let's say it is not possible to discriminate between many them from any single viewpoint using any of the available sensors. In other words, no matter what sensor is used, the view yielded from any viewpoint is such that, for many of the objects in the library, we could form multiple hypotheses about object identity and pose. So given the data observed from any viewpoint, how do we optimally choose the next viewpoint and the next sensor so that all the currently held hypotheses are maximally disambiguated. So, in order to be more precise, this is more an exercise in sensory cooperation than in sensor fusion in the usual sense. Although the former is subsumed by the latter, the latter is more evocative of a simultaneous utilization of the information being produced by all the sensors, as was the case for the intruder detection system discussed previously. In the robotic workcell case under discussion, the sensors are invoked one at a time; at each given time and for each given viewpoint, the sensor that maximally disambiguates the hypotheses is selected.

Since the system must reason about viewpoints, that evidently raises the question of how to represent all the available viewpoints efficiently and compactly. What comes to rescue here the aspect graph representation of solid objects, as promulgated by Koenderink and Van Doorn [6]. Each node of an aspect graph is a topologically distinct "stable view" of the object and each arc represents a transition between such views. For illustration, shown in Fig. 3 (a) is the aspect graph of the 2-D object shown in Fig. 3 (b).

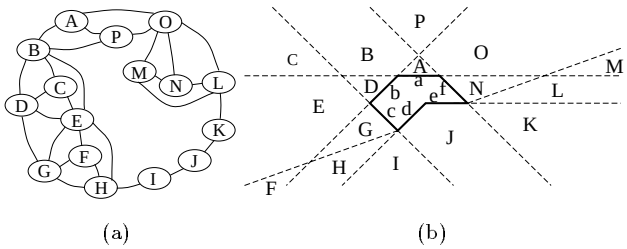


Figure 3: *(a) Aspect graph of the rightmost object of Fig. 5. (b) Regions which view the different aspects of the rightmost object of Fig. 5.*

The second major issue that this kind of a system must address is the representation of uncertainty associated with an object and pose hypothesis. The sources of uncertainty are many. Given the usual noise in sensory information and the artifacts associated with the extraction of features from data, mid-level groupings extracted from the sensory data might match with more than one feature on more than one object. To illustrate with a simple example, suppose from viewpoint $V1$ shown in Fig. 4, we are able to extract two edges $S1$ and $S2$ shown there and suppose the model library contains just two 2D objects shown in Fig. 5 whose boundary edges are labeled 1 through 4 for one of the objects and a through f for the other. Now even without any noise in the sensory information, the two sensed lines $S1$ and $S2$ will match two object edges in four different ways, as shown in Fig. 6, resulting in four different object and pose hypotheses. As should be obvious, in the presence of noise and artifacts, this number of hypotheses could be even larger. For example, considering that the lengths of the equally-long object edges 2, 4, b, c, d, and f are not that different from the lengths of the equally-long object edges 1, a, and e, it would not be incorrect

to assume that the sensed feature $S1$, although corresponding strictly only to one of the set $\{1,\ a,\ e\}$ would with high probability also match any of the set $\{2,\ 4,\ b,\ c,\ d,\ f\}$. How to represent these probabilities and then how to update the resulting object and pose hypotheses as further sensed information becomes available are major issues unto themselves. In [3], we have shown how Dempster-Shafer theory can be used for this purpose. The theory allows a computationally efficient approach to the calculation of a belief value for each object hypothesis.

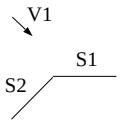


Figure 4: *Two edges, as observed from viewpoint* $V1$.

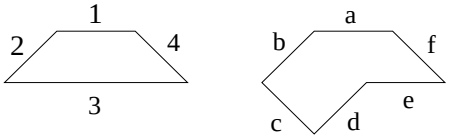


Figure 5: *Two 2-D object models.*

After object hypotheses and their associated belief values have been calculated from the sensed data from some given viewpoint, one must determine what viewpoint and what sensor to use next. To illustrate what goes into the reasoning required for this, we will invoke the same example we used above. Say that after the viewpoint $V1$, we next use the viewpoint $V2$ shown in Fig. 7 for the purpose of disambiguating between the four hypotheses shown in Fig. 6. For three of the hypotheses, as shown in Fig. 7, this new viewpoint should reveal a new object feature, namely $S3$. It is clear from the figure that the appearance of this new feature does not help us disambiguate between the third and the fourth hypotheses. On the other hand, if we were to use the viewpoint $V2$ shown in Fig. 8, we can uniquely distinguish between the four hypotheses. In [3], we have explained how a Dempster-Shafer based formalism, together with a search space consisting of the viewpoint nodes of an aspect graph, can be used to carry out this kind of an optimal selection of the next viewpoint and the next sensor to use.

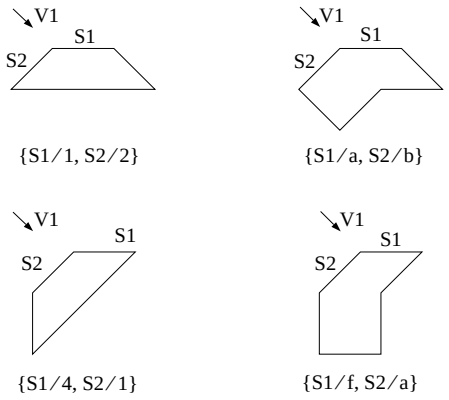


Figure 6: *Edges visible from viewpoint* $V1$*, and corresponding hypotheses.*

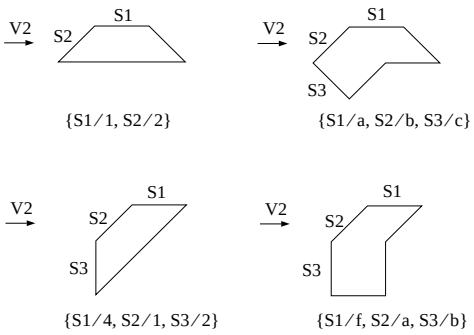


Figure 7: *Edges visible from proposed viewpoint* $V2$*, and possible resulting hypotheses.*

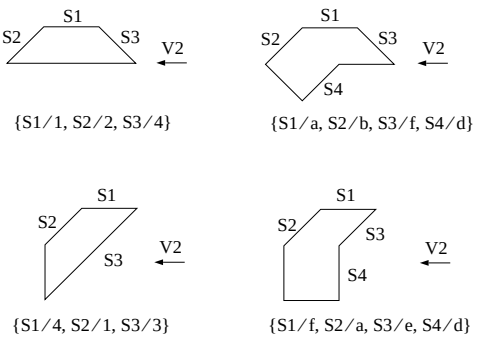


Figure 8: *Edges visible from proposed viewpoint* $V2$*, and possible resulting hypotheses.*

4 Fusion of Color and Range for Fast Object Recognition

Our last case study is about our latest 3D vision system for robotic bin picking. This system, called MULTI-HASH for reasons that will be clear presently, is able to quickly recognize and localize individual objects in bins of the sort shown in Fig. 9. The details on MULTI-HASH are provided in [2].

Figure 9: *A pile of objects for robotic bin picking. Some of these objects are colored. These colors may or may not be seen depending on whether the Proceedings will allow for color material at all.*

Any model-based computer vision system that uses color presents an interesting challenge: How to describe object colors to the system? It is well known that humans are not very objective at discriminating between the different shades of the same hue. Often, humans do not even possess correct labels for some of the shades of a given hue. In MULTI-HASH this problem was resolved by incorporating into the system a learning module that allows the system to figure out for itself the color of an object surface, thus relieving the human of a task that is almost impossible. During the learning phase, objects are shown to the system in different poses (by placing them in a sandbox). The system analyzes the data for the visible surfaces of the object and the segmented regions are displayed on a workstation. At the same time, the corresponding model object is also displayed in another window on the workstation. Through mouse-clicks and key-strokes, the rendered image of the model object is rotated until it corresponds roughly to the pose of the sensed object. Then, again through mouse-clicks, the human establishes the correspondences between the model object surfaces and the sensed object surfaces.

Fig. 10 shows in the lower left part a sensed image of a object placed in the sandbox during the learning phase. Shown on the right at the same level is a segmented image. In the row above are the model objects. The human clicks on the model object that corresponds to the sensed object and then through key-strokes rotates it in the manner described above. Finally, as just described, the human establishes correspondences the segmented surfaces and the model-object surfaces.

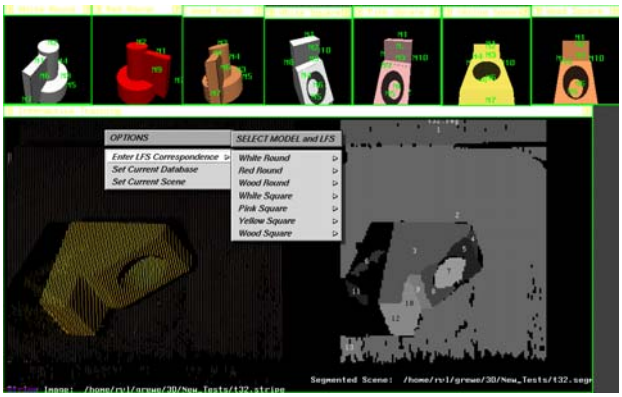
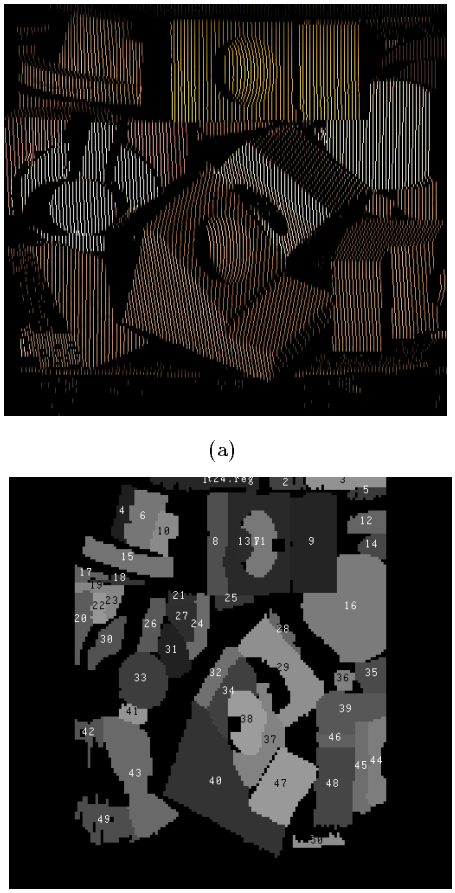


Figure 10: *A screen dump from an SGI machine displaying how interactive training takes place in MULTI-HASH. On the left in the main window is a color structured light image of the training scene and on the right is the segmented image. As with the previous figure, the colors in the structured light image may or may not be seen by the reader.*

A special color structured-light sensor was built by Lynne Grewe in our laboratory for simultaneously collecting both the range and the color data. As in a conventional structured-light scanner, the scene is scanned with a laser stripe that yields the range data. Each projection of the laser stripe is followed by illuminating the scene with a broad white-light stripe. The part of the scene that is illuminated by the white-light stripe is sampled by a color camera at only those pixels that were originally illuminated by the laser stripe. In this manner, registered color and range images are collected in MULTI-HASH. Shown in Fig. 11 (a) is a typical color structured-light scan of a scene and shown in (b) is a segmentation obtained. We believe the reader would be impressed with the quality of the segmentation.

The recognition strategy that is built up by MULTI-HASH during the learning phase first constructs a decision tree from all the data collected and then translates the decision tree into a hash table for fast object recognition and pose calculation. To ex-



(a)

(b)

Figure 11: *(a) Composite color structured light image of a typical test scene. (b) The segmentation map.*

plain why a decision tree is used, we must first explain the motivating reasons for using a hash table for object recognition. This we will do next.

Note that an important goal here is to divide up the attribute space in which objects and their different unique poses are represented into disjoint regions, each region corresponding to a different pose for a different object. (It is important to bear in mind that while a 3-D object possesses an infinite number of poses, it is sufficient to consider, as in the case of aspect graphs, a set of topologically distinct poses, each pose characterized by a set of, say, object surfaces and a vertex.) Consider, for illustration, the simple case of two object/pose classes shown in a 2-D attribute space in Fig. 12. The circles and crosses shown in the two regions are the points corresponding to the objects/poses shown during the learning phase. Note that all the circles correspond to the case of the same

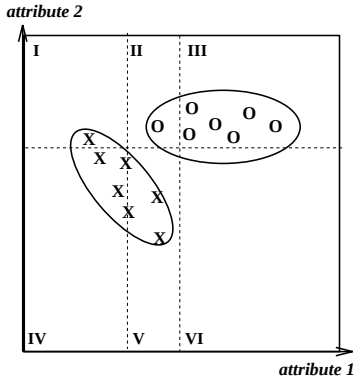


Figure 12: *The crosses represent the samples collected during the learning phase when a given type of object is shown to the sensor a number of times such that the same surfaces are visible to the sensor. The circles shown are either for a different object or for the same object but with a different set of surfaces made visible to the sensor. When the two regions shown are non-overlapping, as illustrated here, it is not too difficult to come up with the bins of a hash table so that each bin will point to a single pose class. This is the case for the six bins shown.*

object surfaces being made visible to the sensor; each circle corresponds to the object being in a different pose even though the sensor is seeing the same surfaces, or a different object although of the same type. So the spread represented by the presence of many circles is owing to the fluctuations introduced by the dependence of many of the attributes on the slant angle between the sensor and the surface. This point is described in detail in [2]. The same is true for the crosses. Both the regions shown in Fig. 12 can be represented by density functions estimated from the points shown. Now, when the two regions are overlapping, the ideal decision boundary between the two regions shown will be some curve (a hypersurface, more generally speaking) whose precise form will depend on the criterion chosen for minimizing the probability of misclassification. However, for fast object recognition, it is best to approximate this decision boundary by lines (planar surfaces for the general case) orthogonal to the attribute axes, as in Fig. 13 (a). When we extend these lines (surfaces) to span the entire attribute space, we end up with a hash table, as shown in Fig. 13 (b). In each box of such a hash table is deposited a pointer to the corresponding object pose and identity. So when the relevant attributes are measured for a scene object, the location in the attribute space of the point derived from the scene data immediately tells us as what object identity and pose class

the scene object belongs. The precise pose of the object can subsequently be calculated by using, say, the approach outlined the appendix of [5].

Now that the reader understands our reasons for why we wish to divide up the attribute space formed by color and range attributes into bins whose walls are orthogonal to the attribute axes, the question then becomes one of how to go about doing so. As discussed in [2], ideally one would want an optimal hash table, one in which the bins are as pure as possible and that contains the fewest number of bins. (The fewer the number of object classes straddled by a bin, the purer it is.) Unfortunately, constructing optimal hash tables is exponentially complex and, therefore, not practically feasible for model object libraries of even moderate size. The alternative is to construct what turn out to be good hash tables through the tool of decision trees. For details, the reader is referred to [2] where we have discussed the performance of the system.

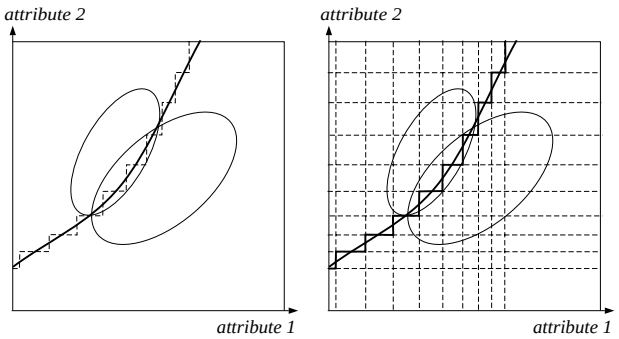


Figure 13: *(a) The bold curve shows the decision boundary for the case where the two class regions are overlapping. The dashed line indicates the approximation of the decision boundary by lines orthogonal to the attribute axes. (b) A hash table is generated by extending the dashed lines shown in (a). In each box of such a hash table is deposited a pointer to the corresponding object pose and identity.*

5 Concluding Remarks

Looking at the history of science and technology, since the arrow of automation has never pointed downwards, it is obvious that robots will only get smarter in the years to come. While it is anyone's guess how fast robotic intelligence will evolve and when it will reach a level so that autonomous robots will play a truly useful role in our societies, there is no disputing the fact that the knowledge we already possess can be deployed to make the current breed of robots much more intelligent and autonomous than what one sees in the factories of today. In this paper, we have reviewed three contributions from our laboratory that demonstrate how sensor fusion can be used to enhance the level of autonomy and intelligence of robots.

6 Acknowledgements

The discussion in this paper is based on the doctoral dissertation work in the Robot Vision Lab at Purdue by Seth Hutchinson, Lynne Grewe, and Min Meng, and the technical support provided by Matt Carroll. This work was supported by various organizations including National Science Foundation, U.S. Army Picatinny Arsenal, and Office of Naval Research.

References

[1] M. S. Carroll, M. Meng, and W. K. Cadwallender, "Fusion of ultrasonic and infrared signatures for personnel detection by a mobile robot," *Sensor Fusion IV: Control Paradigms and Data Structure, Proc. SPIE*, Vol. 1611, pp. 619-629, 1991.

[2] L. Grewe and A. C. Kak, "Interactive learning of a multiple-attribute hash table classifier for fast object recognition," *Computer Vision and Image Understanding*, Vol. 61, No. 3, pp. 387-416, 1995.

[3] S. A. Hutchinson and A. C. Kak, "Planning sensing strategies in robot work cell with multi-sensor capabilities," *IEEE Trans. on Robotics and Automation*, Vol. 5, No. 6, pp. 765-783, 1989.

[4] T. Khanna, *Foundations of Neural Networks*, Addison-Wesley Publishing Company, Reading, Massachusetts, p. 95, 1990.

[5] W. Kim and A. C. Kak,"3-D object recognition using bibartite matching embedded in discrete relaxation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 3, pp. 224-251, 1991.

[6] J. J. Koenderink and A. J. Van Doorn, "The internal representation of solid shape with respect to vision," *Biol. Cybern.*, Vol. 32, pp. 211-216, 1979.

[7] R. Lippman, "An introduction to computing with neural nets," *IEEE ASSP Magazine*, Vol. 4, pp. 4-22, 1987.

[8] D. E. Rumelhart, J. L. McClelland, and PDP Research Group, *Parallel Distributed Processing*, Vol. I, MIT Press, Cambridge, Massachusetts, 1987.

[9] P. K. Simpson, *Artificial Neural Systems*, Pergamon Press, New York, 1990.