The International Journal of Robotics Research

http://ijr.sagepub.com/

Combining Sonar and Infrared Sensors for Mobile Robot Navigation

A.M. Flynn

The International Journal of Robotics Research 1988 7: 5
DOI: 10.1177/027836498800700602

The online version of this article can be found at:
http://ijr.sagepub.com/content/7/6/5

Published by:

http://www.sagepublications.com

On behalf of:

Multimedia Archives

Additional services and information for *The International Journal of Robotics Research* can be found at:

Email Alerts: http://ijr.sagepub.com/cgi/alerts

Subscriptions: http://ijr.sagepub.com/subscriptions

Reprints: http://www.sagepub.com/journalsReprints.nav

Permissions: http://www.sagepub.com/journalsPermissions.nav

Citations: http://ijr.sagepub.com/content/7/6/5.refs.html

>> Version of Record - Dec 1, 1988

What is This?

Downloaded from ijr.sagepub.com at RMIT UNIVERSITY on May 28, 2014

A. M. Flynn
Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

Combining Sonar and Infrared Sensors for Mobile Robot Navigation

Abstract

Multiple sensors can be used on a mobile robot so that it can perceive its environment with better accuracy than if either sensor were used alone. Sonar and infrared sensors are used here in a complementary fashion, where the advantages of one compensate for the disadvantages of the other. The robot then combines the information from the two sensors to build a more accurate map. Another representation, a modified version of the curvature primal sketch, is extracted from this perceived workspace and is used as the input to two path planning programs: one based on configuration space and one based on a generalized cone formulation of free space.

1. Introduction

In order for a mobile robot to maneuver through its environment and execute any sort of reasonably intelligent task, it should first be able to perceive. That is, it should be able to navigate through its world based on sensory information. In contrast to stationary arm robots that are fixed to a global coordinate frame, a mobile robot's world is essentially unknowable due to cumulative errors, and sensing must be done.

Various types of sensors have been used in the past, such as bumper switches, shaft encoders, sonar transducers (Chattergy 1985), photocells and infrared proximity sensors (Everett 1982a), cameras (Nilsson 1969; Moravec 1981), infrared beacons (Giralt et al. 1983), and laser range finders (Thompson 1979). Each type of sensor, however, has some limitation. Shaft encoders are not accurate when wheels slip, for instance. Sonar sensors have a wide beamwidth and are sensitive to specular surfaces, and cameras require computationally intensive processing. One solution, followed here, is to use redundant sensors and utilize the advantageous characteristics of some in overcoming the disadvantages of others.

This paper describes how two very inexpensive sensors, a sonar range finder and a novel infrared sensor, are coupled to produce data that are better for building a representation of the robot's environment than if the sensors were used individually. The sonar range finder measures the distance to an object, but has poor angular resolution due to its wide beamwidth. In contrast, the infrared sensor, though not able to measure distance accurately, has good angular resolution in detecting the absence or presence of an object. By using both sensors to scan a room, the robot is able to build a better map.

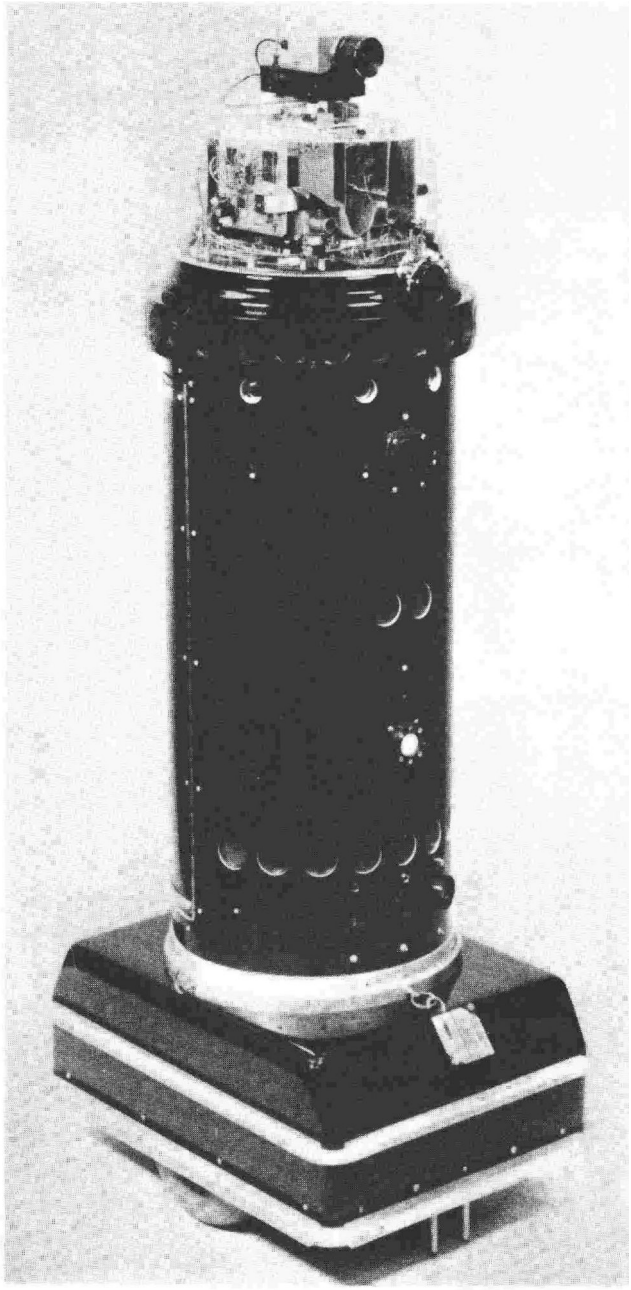
The infrared sensor is able to find edges of doorways and narrow passages that would be otherwise blurred by the sonars. The boundary of data points that is initially created by the sonar readings is redrawn appropriately to mark the doors detected by the infrared. The curvature primal sketch (Brady 1984) is then extracted from this modified boundary, and significant curvature points are used as landmarks for matching between scans as the robot moves. Scans from subsequent moves can be merged, and a room map boundary is created. This is then transformed into a list of polygons in order to provide the necessary input for path planners based on generalized cones (Brooks 1983) or configuration space (Brooks and Lozano-Perez 1985).

The robot used in this work was Robart II (Everett 1985a) built (Fig. 1) by LCDR Bart Everett, Autonomous Systems Project Officer, Naval Ocean Systems

The International Journal of Robotics Research,
Vol. 7, No. 6, December 1988,
© 1988 Massachusetts Institute of Technology.

Downloaded from ijr.sagepub.com at RMIT UNIVERSITY on May 28, 2014

Fig. 1. Robart II—an autonomous sentry robot.



Center. Robart was designed as a sentry robot and was loaned to the Naval Surface Weapons Center as a mobile platform for research and evaluation of sensors and navigation algorithms.

2. Modeling the Sensors

In order to determine the best way to fuse information from multiple sensors, we need a characterization of the strengths and weaknesses of each sensor.

2.1. Characterizing the Sonar Range Finder

The sonar sensor used was a Polaroid ultrasonic transducer (CBD 1984). Its beamwidth, typically measured at the 3-dB point, is roughly 10°, but the transducer is sensitive enough to detect echoes of energy transmitted from the side lobes. In testing, the range finder could detect a 1-in diameter pole up to an angle of 40°. With such a large beamwidth, a robot scanning a room with this sensor perceives all objects to be much wider than they really are.

The range finder is capable of measuring distances to an object with a resolution of 0.12 in through a range of 0.9 to 35.0 ft. This is accomplished by measuring the time of flight between a transmitted pulse and a returned echo and multiplying by the speed of sound. The distance measure, however, is not necessarily the distance in the direction the sensor is pointing, since the width of the beam may cause an echo from one edge to be returned before the echo from the centerline.

Another measurement error is due to specular reflections on smooth surfaces. Due to the large wavelength of sound, about ⅛ in, many surfaces appear smooth. A sonar beam incident on such a surface does not reflect an echo directly back to the sensor. Instead, it bounces off at an angle equal to the angle of incidence, and possibly bounces off other objects before being detected. Hence, the transducer measures a much longer distance than it should. In actual tests against a smooth surface such as Sheetrock, specular reflections occur when the sensor is aimed at an angle less than 25° from the surface. Against a rougher surface such as cinder block, there are no specular reflections at all.

Other errors come about due to atmospheric effects, such as the change in the speed of sound caused by

Downloaded from ijr.sagepub.com at RMIT UNIVERSITY on May 28, 2014

temperature and humidity variations. For instance, distances calculated when assuming 80° F, but where actual temperature is 60°, will be 7 in too long (Everett 1985b).

Finally, there is another type of measurement error due to the form of the sound pulse which is transmitted. The pulse is actually a chirp, 1 ms long, of 56 pulses of four different frequencies. There are 8 pulses at 60 kHz, 8 pulses at 56 kHz, 16 pulses at 52.5 kHz, and 24 pulses at 49.41 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.

Four different frequencies are transmitted to compensate for the fact that different types of surfaces absorb energy of different frequencies. With four frequencies, it is more likely that every surface will return an echo. However, that echo is not necessarily associated with the first pulse, the one from which the time of flight is measured. In the worst case, an echo from the last pulse adds 1 ms onto the actual time of flight. This error corresponds to about 3 in of additional distance measurement.

2.2. Modeling the Infrared Detector

The infrared detector was designed and built by LCDR Everett. It has four levels of power output and four stages of detector sensitivity. The sensor emits a pulse of near-infrared light and then uses a parabolic dish and a near-infrared detector to sense, over a very narrow area, any returned energy. Unlike sonar, the time of flight of the light pulse cannot be easily measured. The sensor merely indicates whether any returned pulse was detected. So, although distance to an object cannot be ascertained, the absence or presence of an object can be determined with very good angular resolution.

The infrared emitter used here is actually slightly more complicated than described above. Four LEDs are used to enable the sensor to incrementally step up the power output so that the range is increased. A good analogy is that of a blind man's telescoping cane. First one LED is fired. If no return is detected, two LEDs are then fired simultaneously, doubling the power output, and extending the range of the sensor. If, again, there is no detection, three LEDs are fired in unison, and, finally, four if necessary. When a returned pulse is detected, the robot notes how many LEDs were fired, and this gives a very rough indication of range. However, it is more beneficial to have the robot scan and note the point at which there is a discontinuity from detection to no detection, or vice versa. This often means a corner has been found, as shown in Fig. 2.

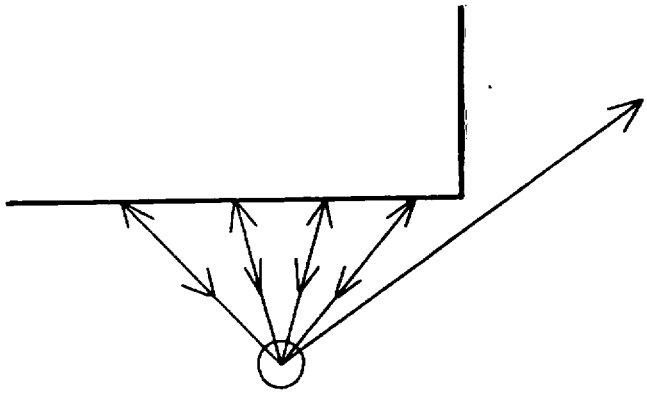


Fig. 2. Infrared sensor finds a corner.

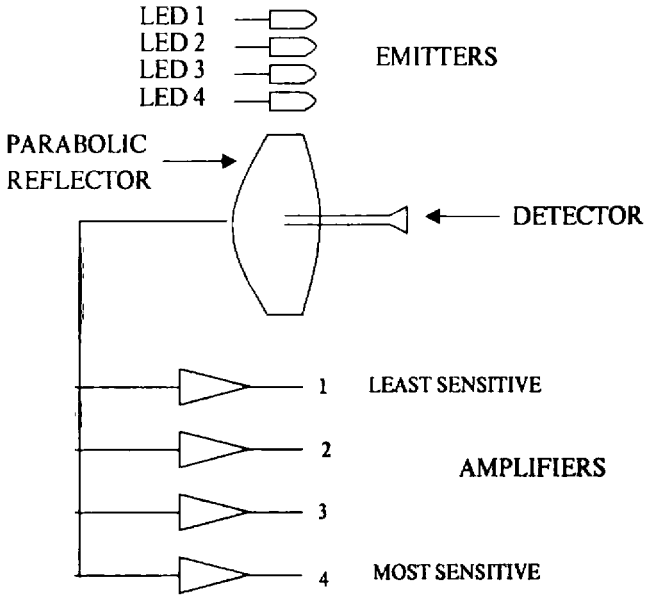
The infrared detector circuitry is also slightly more sophisticated than described above. Instead of a binary Detect/No Detect signal, there are actually four detectors of varying sensitivity. After the first single LED is fired, the robot polls these four detector outputs, starting with the least sensitive one. If any detector has sensed an echo, the robot notes how many LEDs were fired and which level of detector sensitivity was first to pick up the echo. If none detected any return, two LEDs are fired together, all the detectors are polled, and the process is repeated up to the firing of all four LEDs. Figure 3 is a diagram of the infrared sensor.

The resulting data is input to the computer in a two-digit format, such as 31. The first digit signifies how many LEDs were fired, and the second digit tells which detector circuit sensed the echo. In this example, three LEDs were fired, and the first detector sensed the echo.

The maximum range of the sensor, using four LEDs, is approximately 18 ft, although only three LEDs were actually used here. They gave a maximum range of about 10 ft. The range is a function of how much energy is reflected off the surface. Different types of

Downloaded from ijr.sagepub.com at RMIT UNIVERSITY on May 28, 2014

Fig. 3. The infrared sensor.

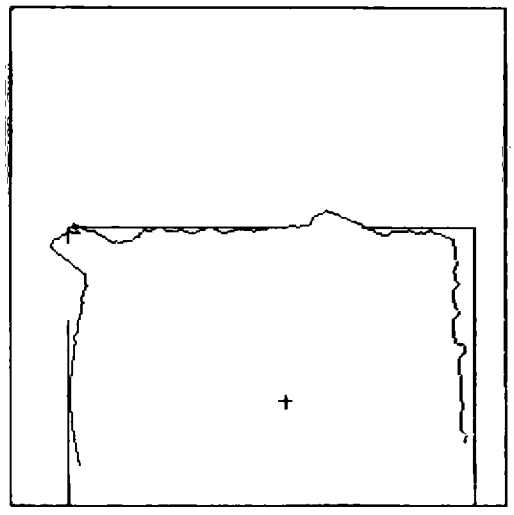


surfaces absorb varying amounts of infrared energy, so this sensor gives only a rough estimate of distance ranges. However, smooth surfaces do not pose the same problems of specular reflection as they do with sonars, because of the much shorter wavelength of infrared light.

3. Combining Data from Sonars and Infrareds

Extensive testing involving different configurations of obstacles was conducted in a basement room approximately 19 ft long by 13 ft wide, with the robot situated in various locations. For each case, the robot remained stationary while scanning its head-mounted ultrasonic and near-infrared sensors across the scene of interest, taking 256 separate readings. The head angle and distance measured for each reading were converted from cylindrical to cartesian coordinates and overlaid onto the actual room map. All the data were displayed on the screen of a LISP machine in a 256 × 256 bit map array.

Fig. 4. Sonar plot of the room.



3.1. Examples of Sonar Errors

Many of the problems associated with ultrasonic ranging show up clearly in plots of the experimental data. Figure 4 shows one sonar plot of the room in which Robart II was positioned 6 ft in front of an open doorway. The walls of the room were relatively rough, with cinder blocks along one side and exposed studs along the other three, resulting in a fairly clean plot. The open doorways, however, are not discernible to the ultrasonic sensor because of beam divergence (the beam is wider than the door openings).

The arc of data points along the left-hand wall in Fig. 4 illustrates the problem of the sonar measuring the shortest distance within the effective beamwidth, as opposed to the distance along the beam centerline. Where the beam is perpendicular to the wall, the distance measurement is fairly accurate, but to either side of that point, the return is shorter than it should be. This is again illustrated in the upper right-hand corner of Fig. 5. When the sonar is pointed toward a corner of the room, either side of the corner is closed to the sensor, so an underestimate in range is always returned. Consequently, the real corners of the room become very hard to locate.

A good example of the specular reflection problem is portrayed in Fig. 6. Two L-shaped obstacles were placed in the room, and the robot is about a foot and

Downloaded from ijr.sagepub.com at RMIT UNIVERSITY on May 28, 2014

Fig. 5. Near-infrared and sonar data in bit-map display for circled region of plot (upper right).

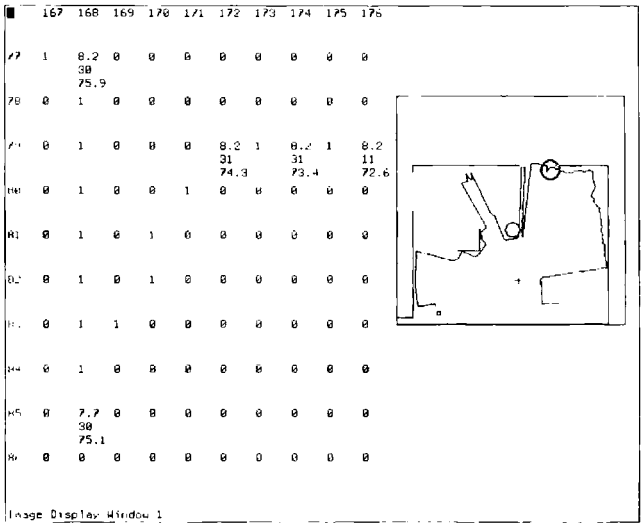
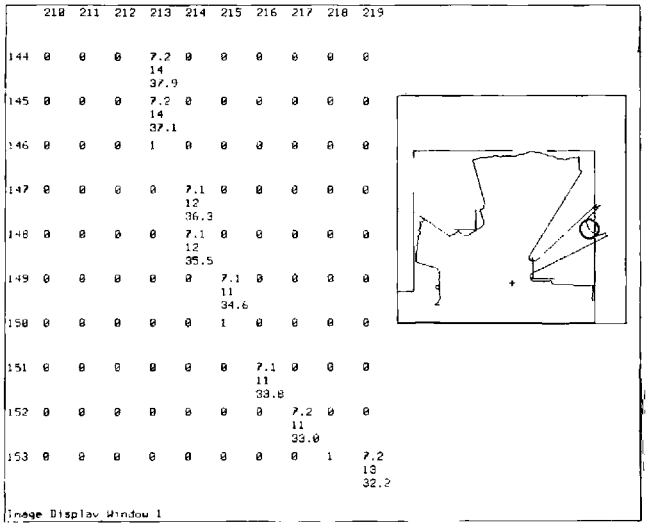


Fig. 6. Specular reflection.



a half from one of them. The obstacle was really a folding closet door that had a series of horizontal slats that were angled downward. The angle of these surfaces caused the beam to bounce off in some other direction before finally being detected by the transducer; thus the distance read was much longer than the actual distance. The result is that the robot is blind to the obstacle right in front of it.

3.2. Infrared Problems

Recorded readings from the near-infrared proximity sensor are carried along in the data structures behind the bit-map displays and can be overlaid onto the sonar boundaries to see where discrepancies between the two sensors occur. The conversion from cylindrical coordinates (in which the data are taken) to cartesian coordinates (in which the data are displayed) creates a sparse boundary. Therefore, filler points are added to the original sonar range data to create a connected boundary. Every data point in the boundary carries with it information depicting its index into the boundary, its *xy* position, the bearing angle (in robot coordinates) for each reading, the radial distance measured by the sonar, and the output of the programmable proximity detector.

Figure 5 shows the actual data values along one small section of the boundary. The circled region on the plot corresponds to the partial array of numbers to the left. The zeros represent white space while nonzero numbers represent either actual data points, filler points, or room obstacles. For real data points, there is a set of three numbers: the radial distance in feet, the infrared sensor reading, and the bearing angle in degrees. Filler points in the boundary are marked as 1's.

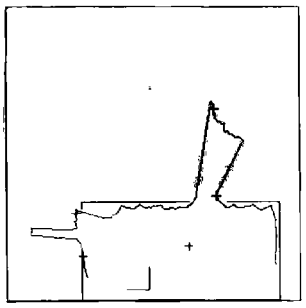
For the particular scan which produced this plot, only three of the four LEDs were used (one was temporarily inoperative), so the reading of 30 at the point (168 85) shows that at the maximum range of the system, nothing was detected when all three LEDs were fired. The reading of 31 at the next real data point, (172 79), means that the most sensitive comparator responded to the presence of reflected energy. As can be seen in the plot, the open doorway is indiscernible to the ultrasonic range finder, yet is easily located by the near-infrared unit.

Under certain conditions, the infrared sensor is capable of locating edges of doorways very well. It does this by noticing changes from where it detects something to where it detects nothing. If there is a long clear distance beyond the door and the edge of the door is within the infrared's range, then that edge can be located with a very good angular resolution.

Figure 7 shows two scenes in which the sonar blurs the doors, but the infrared is able to pick out the edges

Downloaded from ijr.sagepub.com at RMIT UNIVERSITY on May 28, 2014

Fig. 7. Infrared sensor picks out the doors.



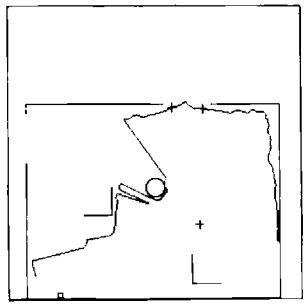
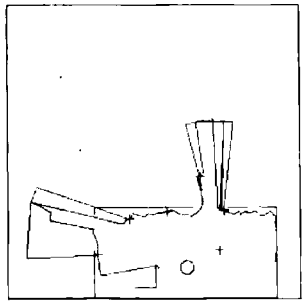
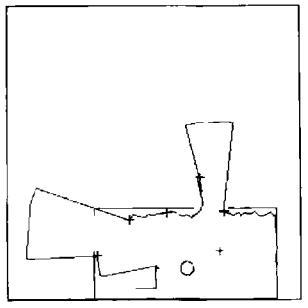


Fig. 8. Redrawn map from combining sensors.





very accurately. The crosses mark where the infrared detects corners or, specifically, where infrared readings of 30 transitioned to anything else. Readings of 30 indicate that all three LEDs were fired and no echo was detected. In the plot on the left, the sonar is able to see a passageway through the door at the top of the picture and through the door on the left. However, the sonar smears the edges of the left-hand doorway, but the crosses marked by the infrared make it obvious that the infrared was able to find the doorway edges where they really were. Similarly for the plot on the right, the crosses at the top of the room show that the infrared correctly finds the edges of the doorjam again, while the sonar is completely blind to them.

It was hoped that the stepped power output of the emitter and the four levels of sensitivity of the detector would enable the sensor to detect less drastic changes in depth and therefore enable the sensor to discern if one object was a few feet in front of another. It turned out, however, that the sensor was not capable of such performance because the varying reflectivities of surface materials in the room precluded any attempt at correlating range with either emitter output or detector sensitivity. This means that the infrared sensor cannot be relied on to override erroneous sonar readings due to specular reflections.

The crux of the problem can be seen by comparing Figs. 5 and 6. In Fig. 6, the point (215 149) shows an incorrect sonar measurement (due to a specular reflection) of 7.1 ft, and the infrared reads 11. However, in Fig. 5, at the point (176 79), the sonar measures a correct distance of 8.2 ft, and the infrared reading is also 11. Since the 11 reading can be achieved to a distance of 8.2 ft in some circumstances, we cannot rely on the infrared sensor to detect specular reflection errors made by the sonar. The distance scaling ability of this infrared sensor is too coarse for ranging. Nevertheless, it remains well suited for detecting, with very fine angular resolution, edges, and large depth discontinuities such as doorways, where there is a large open space in the room beyond.

3.3. Rules for Sensor Fusion

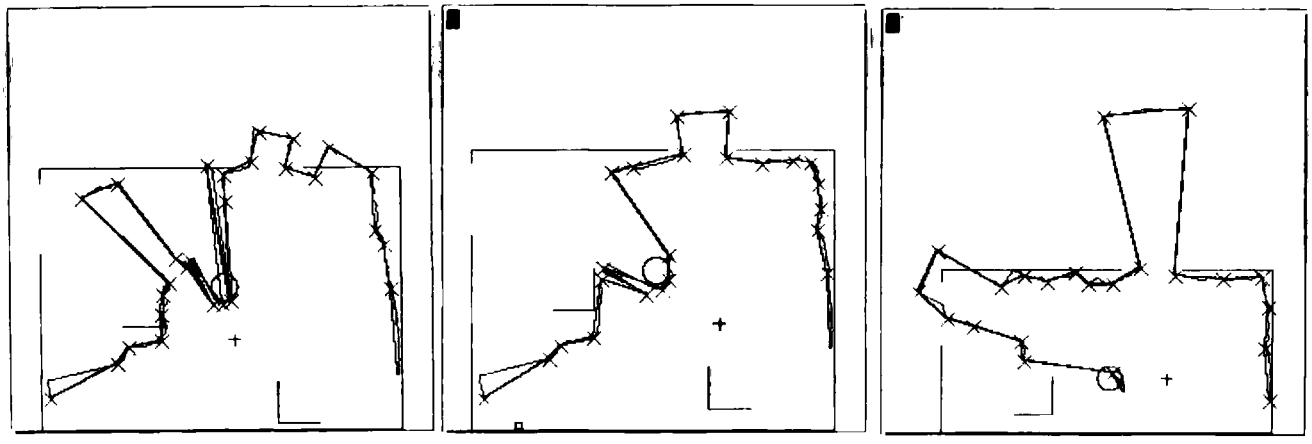
After experimentally characterizing both the Polaroid ultrasonic ranging system and the programmable proximity detector, it is possible to formulate a set of rules for how the information from each sensor should be combined to build a world model. These rules will tell when data from either sensor is valid and which sensor to rely upon when conflicting information is obtained:

1. If the sonar reading is greater than the maximum range for the near-infrared sensor, then ignore the near-infrared output.
2. If the sonar reading is at its maximum value, then the real distance is greater.
3. Whenever the infrared sensor detects a "30"-to-other discontinuity (a change from no detection to detection), and the associated sonar reading is less than 10 ft, then a valid depth discontinuity has been detected.

Using these few simple rules, we can redraw the original sonar boundary to take into account any features found by the near-infrared sensor. This is done by finding a pair of edges and redrawing the boundary

Downloaded from ijr.sagepub.com at RMIT UNIVERSITY on May 28, 2014

Fig. 9. Curvature primal sketch of three views.



between them to be an arc at the maximum of either the horizon of the infrared or the furthest sonar reading returned between the two edges of the opening. Figure 8 shows an original boundary and the redrawn map based on information from both sensors. The original boundary is shown in the left-hand picture with the new boundary overlaid on it at the locations where the programmable proximity sensor marked the edges. The right-hand figure shows the modified boundary after combining the near-infrared data with that from the sonar, with significantly more pronounced resolution of actual position. This modified boundary is then passed to the next stage of processing, which builds a representation of the room better suited for a path planner.

4. Building a Representation

The refined map created by combining data from these two completely different sensors needs to be converted to a representation suitable for planning intelligent tasks, such as navigating through a workspace. This is achieved by first transforming the refined data to an intermediate representation, a modified version of the curvature primal sketch, which is convenient for merging separate views between robot moves. The curvature primal sketch representation can then also be easily converted into a polygonal representation of the world suitable for path planners (Brooks 1983; Brooks and Lozano-Perez 1985).

Basically, the curvature primal sketch describes a boundary by a spline whose knot points are significant changes in curvature (Brady 1984). These significant curvature changes are located by first choosing a set of primitive curvature types: corners, ends, smooth joins, etc. These primitives are parameterized as $f_i(s)$, where s is the arc length along the sonar boundary. This parameterized boundary is convolved with a Gaussian of standard deviation σ:

$$G_\sigma(t) = (1/\sqrt{2\pi\sigma}) \exp(-t^2/2\sigma^2).$$

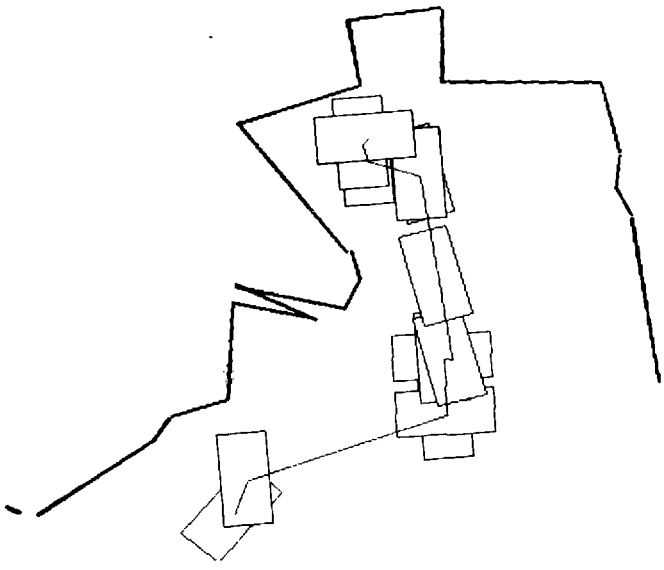
The first and second derivatives of the convolution of this Gaussian with the boundary, namely, $(G_\sigma * f_i(s))'$ and $(G_\sigma * f_i(s))''$ are analyzed for maxima, minima, and inflection points to locate the significant curvature changes.

Figure 9 shows an example of the curvature primal sketch of the robot's view of the room from three positions. In the middle picture, the robot has moved 2 ft to the right with respect to the left-hand picture. In the right-hand picture, the robot has moved 2 ft forward. Instead of fitting a spline to these knot points, straight-line approximations are used, appropriate for indoor applications in which walls, desks, tables, and other obstacles are most often composed of straight edges.

There are some nice features of the curvature primal sketch that make it a desirable representation for building a room map out of several views from differ-

Downloaded from ijr.sagepub.com at RMIT UNIVERSITY on May 28, 2014

Fig. 10. A found path.



ent positions as the robot goes exploring. First, the knot points are convenient for matching subsequent scans because they have local support. That is, the curvature primal sketch is not based on any global properties of the data plot of the room, such as length, width, or aspect ratio. Rather, the knot points are determined only by their relationship to their neighbors. This is important when one is trying to merge several views of the room, because from different viewpoints the global shape of the room can change drastically. (Some objects become occluded while others are uncovered.) The knot points in the curvature primal sketch, however, will stay close to the same location for incremental changes in robot position. As the robot moves, previously occluded areas can be seen to become uncovered. However, edges that were visible from both perspectives have similar knot points.

Another feature of the curvature primal sketch is that it provides a more concise representation than the raw data and acts as an important intermediate representation before converting to one more suitable for planners. Path planning programs typically expect a list of polygons as input for their simulated worlds. The straight-line segments between knot points can easily be converted to very thin rectangles, translated, and rotated appropriately.

This conversion is shown in Fig. 10, where the left-hand plot of Fig. 9 is represented as a set of linked rectangles of width 1 and input to a planner based on generalized cones (Brooks 1983). This planner finds freeways and channels in the room by carving up the free space into generalized cones. Generalized cones have a spine and are parameterized by some sweeping rule that says how the width to either side of the cone varies. All pairs of edges of the polygons in the workspace are compared, and heuristics are used to prune the number of cones generated. Paths from start to goal are determined by searching the spines of the cones for the optimal route. The robot translates along the length of the spines and rotates at the intersection of two spines.

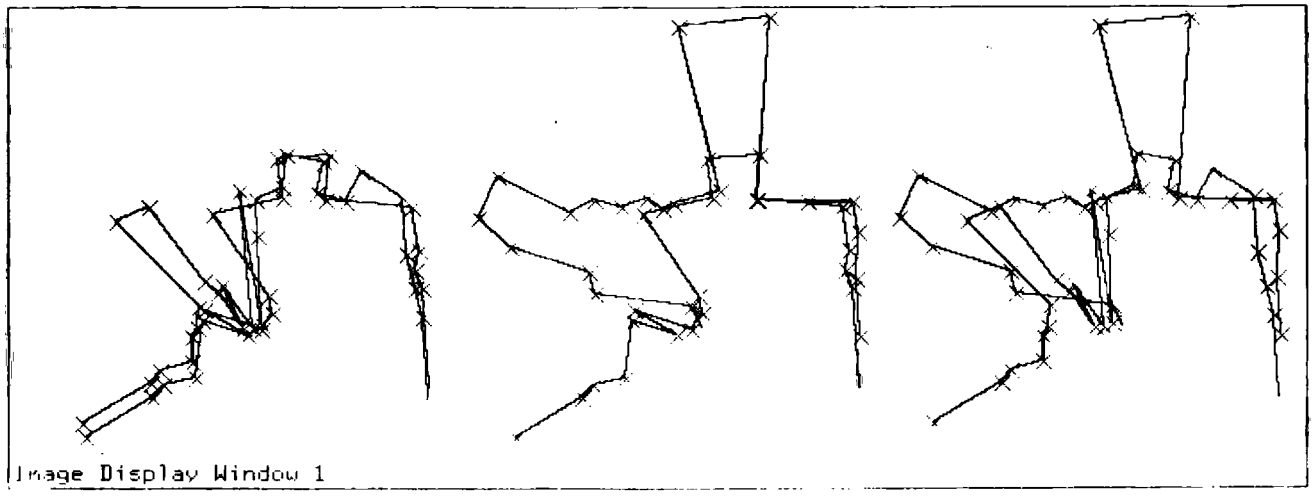
4.1. Future Directions

All of this has been an example of how raw data taken from one scan of a room has been refined and converted into the appropriate representation for a planner. However, what is really desired is a global map built by merging several scans of the room taken from different locations. Problems in doing this arise from having uncertainties both in the sensed data and in the distance and direction moved by the robot. In trying to merge several views from different perspectives, the robot has to be intelligent about what new pieces of data to include in its global model and what old pieces to throw away. Basically, it must match what it can and make decisions about what to do with the rest.

This problem is illustrated in Fig. 11. The earlier examples of three different views of the room are translated and scaled appropriately, then overlaid two at a time so as not to become too cluttered. It is clear that some edges, such as the wall to the right, tend to match up between views but that other edges, such as those behind the circular obstacle, tend to change. From the first position, the robot sees an area behind and to the left of the circular obstacle. From the second position, that area becomes occluded. After the robot moves forward, however, it again sees the same area behind the obstacle, only now it sees it from the right side. The robot should infer that these areas are connected and that a small object lies in the middle of the room. The desired output of a program that was

Downloaded from ijr.sagepub.com at RMIT UNIVERSITY on May 28, 2014

Fig. 11. Three views overlaid two at a time.



intelligent about how to do such a merge might look something like Fig. 12, on which a configuration space planner has been run. This planner is guaranteed to find a solution if one exists, as opposed to the heuristic strategy employed in the generalized cone planner.

The curvature primal sketch representation can be useful for deciding which pieces of what boundaries should be thrown away and which pieces should be maintained in the final map. Certain edges of the boundary came about because of the robot's particular angle or line of sight. These edges are represented by the filler points mentioned earlier, when the connected boundary was created from the sparse sonar data. The connected boundary was created from the raw data in order to make the curvature primal sketch code run, but explicitly keeping this information about filler points can be useful for marking fake edges between knot points. Knowledge about which edges between knot points are fake and which are real is useful for hypothesizing where obstacles can become occluded or uncovered.

After deciding what are close matches and what can be thrown away or added, algorithms appropriately unioning and intersecting the appropriate polygons can be taken from work in computational geometry (Weiler and Atherton 1977). Basically, a union of the plots shown in Fig. 11 would produce the desired output of Fig. 12. The difficulty lies in making decisions about what constitutes a match and about what polygons to intersect or union when there is no match. After a global room map is created, it can then be used as a model of the robot's entire workspace upon which it can plan tasks.

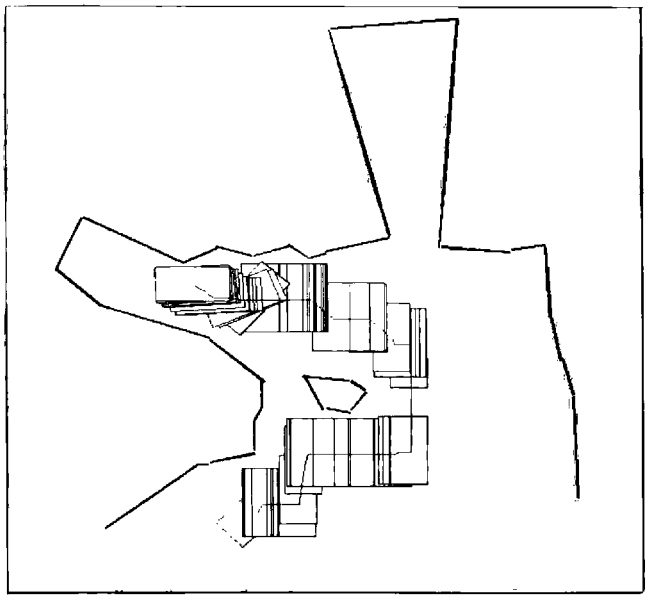
5. Conclusion

Two inexpensive sensors, a Polaroid sonar transducer and a novel infrared sensor have been combined to produce a refined map of the robot's workplace, which is suitable for use by intelligent path planning programs. The sonar had the effect of blurring both obstacles and passageways, although it was quite accurate in providing distance information to the nearest object. The infrared sensor, although unable to provide any accurate distance measurement, was able to detect the absence or presence of an obstacle with very good angular resolution. It could very reliably pick out edges of doorways that were invisible to the sonar.

A refined map was created by combining sonar and infrared data, and this map was then converted into an intermediate representation, the curvature primal sketch, which represented the boundary with knot points that marked significant changes in curvature of the smoothed boundary. The original contour was then approximated by connecting these known points

Downloaded from ijr.sagepub.com at RMIT UNIVERSITY on May 28, 2014

Fig. 12. Merged map with configuration space planner.



with straight-line segments. From this representation, it was a simple step to convert to a representation expected by path planning programs, namely, a list of polygons. Each straight-line segment in the boundary was converted to a list of very thin linked polygons that were translated and rotated appropriately.

An even more important reason for choosing the curvature primal sketch as an intermediate representation was that it had certain features that made it amenable to creating a global room map by merging views of the room taken from different perspectives. Because the calculation of the knot points was based on local information, the curvature of neighboring points, they represented landmarks that were invariant from move to move, provided the moves were small. Therefore, these knot points could be used to match and track similarities between data scans taken on subsequent moves. A method for taking several plots and merging them into a global map was discussed. Consequently, two examples of intelligent programs, path planners, were shown, which were run not on simulated data but on real data produced from very cheap sensors.

References

Brady, J. M. 1984. *The Curvature Primal Sketch.* MIT AI Lab Memo 758.

Brooks, R. A. 1983. Solving the find-path problem by good representation of free space. *IEEE Systems, Man and Cybernetics* 13:190–197.

Brooks, R. A., and Lozano-Perez, T. 1985. A subdivision algorithm in configuration space for Findpath with rotation. *IEEE Systems, Man and Cybernetics,* pp. 224–233.

Chattergy, R. 1985. Some heuristics for the navigation of a robot. *Int. J. Robotics Res.*

Commercial Battery Division. 1984. *Ultrasonic Ranging System.* Polaroid Corporation, Cambridge, MA.

Everett, H. R. 1982a. A microprocessor controlled autonomous sentry robot. Master's thesis, Naval Postgraduate School.

Everett, H. R. 1982b. A computer controlled sentry robot: a homebuilt project report. *Robotics Age,* March/April.

Everett, H. R. 1985a. A second generation autonomous sentry robot. *Robotics Age,* July. pp. 29–32.

Everett, H. R. 1985b. A multielement ultrasonic ranging array. *Robotics Age,* July.

Everett, H. R., and Flynn, A. M. 1986. A programmable near-infrared proximity detector for mobile robot navigation. *Proc. SPIE Conf. on Advances in Intelligent Robotics Systems* 727:221–230.

Giralt, G., Chatila, R., and Vaisset, M. 1983. An integrated navigation and motion control system for autonomous multisensory mobile robots. *Proc. First Int. Symp. Robotics Research.*

Moravec, H. P. 1986. *Robot Rover Visual Navigation.* UMI Research Press, Ann Arbor.

Nilsson, N. J. 1969. A mobile automation: an application of artificial intelligence techniques. *Proc. IJCAI-1,* pp. 509–516.

Thompson, A. M. 1979 (Tokyo). The navigation system of the JPL robot. *Proc. IJCAI-5,* pp. 335–337.

Weiler, K., and Atherton, P. 1977. Hidden surface removal using polygon area sorting. *Computer Graphics* 11(2): 214–222.

Downloaded from ijr.sagepub.com at RMIT UNIVERSITY on May 28, 2014