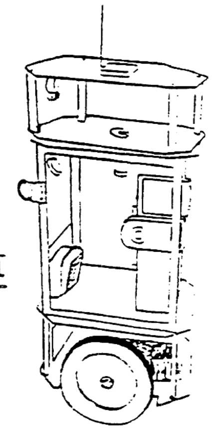
ROBOT BUILDER

The official publication of the Robotics Society of Southern California
P.O. Box 3227, Seal Beach CA 90740, Meetings the 1st Tuesday @ 7:00 PM at MTI College



UPCOMING EVENTS CALENDAR — APRIL 1991

April 2 — RSSC Meeting at MTI College: 7-9 PM
Topic: Mobile Robot Navigation
April 6 — RSSC Robot Project Workshop, at Jerry Burton's
April 30 — RSSC Board Meeting, At Jerry Burton's Lab.
May 7 — RSSC Meeting at MTI College: 7-9 PM
Topic: Robotic Intelligence Software
May 11 — RSSC Robot Project Workshop, at Jerry Burton's
May 19 — ACP Swap Meet
May 28 — RSSC Board meeting, at Jerry Burton's Lb.
June 4 — RSSC Meeting at MTI College:7-9 PM
Topic: RF Datalink

PRESIDENTS MESSAGE

I think last month's Lab meeting was one of the most productive we've ever had. We were able to put a scope on the HPC sonar circuits and further our understanding of just how Synpet implemented their sonars. Unfortunately for the sonar amplitude SIG, the HPC implementation will not yield an amplitude.

The main reason the amplitude signal is not availiable to measure is a diode network designed to limit the returning voltage. The diodes actually clips the voltage information we wanted to convert to a digital value so that it wouldn't damage the 4066 bi-lateral switch chip. This chip acts as a switching set to direct the energy from the selected sonar to the proper amplifiers.

The Heath Hero design implemented an independant circuit for their sonar system and used the value of the return amplitude to activate a trigger when the peak value exceeded a preset threshhold. It appears this is the type of sonar design we need to implement. I'm going to use the control board from my bench based hero jr to supply this circuit which will be driven from one of the parallel ports.

Jess and some of the others figured out how to use PC-Anywhere by hooking 2 lap-tops together. We are now only waiting for someone to build the cables, so we can have about a 40 foot 'leash' for RSSCy. This will allow us to monitor what the software is doing while he is operationally roaming around.

April's meetings will be concerned with navigation (its kind of the closure of the HPC discussion in Feb, and sonars in March). I'll be demonstrating the initial version of the navigation software at the Tuesday MTI meeting and the Saturday Lab will focus on calibration of the opto encoders and examination of the motor stall phenomena that the ROBOT has been experiencing.

Joe has obtained a commitment from Tom Hurst of Orange Coast College to supply us with a room and some facilities (storage and equipment ?) That's the good news, and now the bad news is that we can't meet on Tuesday night. Thursday night is the best for them. We will be discussing this at the April meeting and hope to get member approval/disapproval of moving the meeting place to OCC on a Thursday night........JB

RSSC Board Meeting 26 Feb. 91

The Society monthly board meeting was held 26 Feb. 91 at Jerry Burton's. The meeting is held regularly on the fourth Tuesday of the month at 7 P.M. at Jerry's lab. All board members except Joe McCord were in attendance at the meeting. Joe was traveling with his robots that day.The topic of the monthly meeting was discussed. Jerry will continue presenting RSSCY's Sonar and Navigation operation for the month of Feb. and March. Don Golding will present some soft-ware topics for the April meeting. Roger Ruszkowski will present the wireless communication link for RSSCY at the May meeting.Old and new projects were reviewed by the committee group heads. Jerry asked each of us what progress we were making on our respective projects. A tip of the hat to Jesse for the fine job he is doing on the news letter. Roger offered to lead the ACP swap meet appearance, Sunday 24 March 1991 ....RR

RSSC General Meeting 5 Mar. 91

The Societies general monthly meeting was held 5 Mar. at MTI College. The meeting is held regularly on the first Tuesday of the month at 7 P.M. at the college in Orange. This months meeting was again a packed event. Attendance exceeded 30 people. Jerry started the meeting with introductions and welcomes to the new faces. The main topic for the evening was the operation of the Sonar used by RSSCY. After a short break old and new business was entertained in an orderly manner. Then things wandered a bit as formal procedures gave way to random access. Old business started with the engineering note book. I need to thank several members who made contributions this month. We are over flowing our current meeting room and Joe Mc Cord is seeking a larger place to meet. Joe has located two alternate places and is seeking additional locations. Joe, like the rest of us, works his society duties on a volunteer basis and with limited time. Thank you Joe for doing the leg work on this task. Joe will report as progress is made. We will make a slow orderly relocation of the meeting location so we do not loose contact with any one. I have purchased the parts to assemble a wireless communication link for RSSCY that will not require FCC licenses or limit who may operate the link. Once I get the link hardware and soft-ware developed on my test computers, I will ask the Hardware and Software groups to help me install the hardware on RSSCY and tailor the software at one of the Saturday work shops. New business was limited and the Robot Laboratory was set for 9 Mar. The meeting adjourned at 9 P.M. ....RR

RSSC Robot Laboratory 9 Mar. 91

The Societies Robot Laboratory monthly meeting was held 9 Mar. at Jerry Butron's laboratory. The meeting is held regularly on the first Saturday of the month at 10 A.M. More than 10 members showed up to contribute to activities. The two main areas of study were the Polaroid sensors on the ROBOT and the com link between computers. Effort was expended examining just what echo each of the sensors was receiving as a return. The other major effort was to boot a com link between two computers by Jess using the serial ports. This was to simulate a terminal monitoring the ROBOT computer. The computers were a 286 Compact Portable III communicating with a Tandy Laptop Portable(floppy only, no hard drive). The software demonstrated was pcAnywhere and Aterm. Each computer was able to boot the other machine and the final demonstration was for the small floppy driven Laptop to control the big machine, booting and running large and complicated programs (Word Star 6.0 etc). The communication rates for these tests was 19.2 Kbaud with out either machine missing a beat........JJ

ACP Swap meet

The ACP swap meet event on 24 March got off to a shaky start. The battery in the voice transmitter was down and we didn't have a sign up sheet or any membership flyers. However we made do with what we had. A fresh battery was obtained locally and an old clip board was pressed into service from Jess's mobil office for the sign up sheet. The front page of a back issue of a ROBOT BUILDER was glued down on the table as advertizement. Tim Lewis had worked up some demonstration software for RSSCY for use at this swap meet. It seemed to generate a lot of interest. Maybe we will see some of this at the general meeting. We should give Tim some time at the one of the meetings to show this off to everyone. At least sixteen people showed great interest in what the Society is doing and they gave us their names and addresses so we could contact them later for possible club membership......JJ

ACTION ITEMS

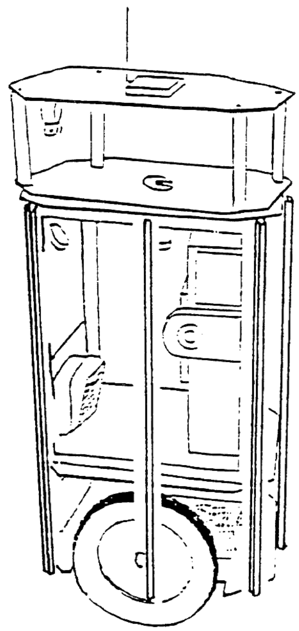
These are some of the current items that are needed for the completion or expansion of the club ROBOT development. There was discussion at the RSSC meetings and some have been assigned a task leader.

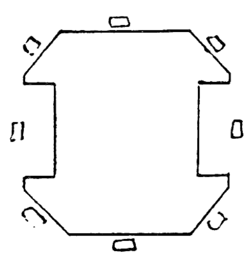
1. The "DOCKING STATION" has been assigned to Tim Lewis. This station will be used for charging and must have an automatic interconnect. It also requires some type of location identifier for the ROBOT. Tim says he still needs help with ideas!

2. An "RF LAN" or computer to computer interconnection. A need is to assist in debugging of ROBOT software. Rodger Ruszkoski and Jess Jackson are both contributing to an answer on this one. Rodger is developing an RF modem via a wireless phone concept. Jess is generating an interim solution with a long tether wire connecting the serial ports of both computers. The software

misses. The need is immediate and therefore the design must be simple.

Talking the problem over with some of the club members a target design was determined. This design was passed by the Mechanical Guru(Mark Frank) of the club and he had only a few choice words of "nice but how about this".





Bump Detector Bar Locations

A final design of the mechanical portion was determined. The first design technique was to hinge eight(8) upright pressure bars from the upper deck (just below the neck). This would be adequate but the leverage of contacting a high obstacle like a table would be rather low. Mark suggested that the pressure bars be hinged in the middle so that the table would press the bar more easily. It seems that the sonar covers the mid range well but has trouble with the high and low obstacle areas.

All eight(8) pressure bars switches would be electrically interconnected to two inputs. The software needs a wakeup alert that the ROBOT has crashed into something. This will be accomplished by feeding all eight lines into an eight input nangate chip so one output can be fed to an interrupt line on the HPC board.

The second input is to a parallel input port. This helps implement an "OOPs" file that the software can read.

The sequence in as follows. The ROBOT gets too close to an obstacle and depresses one of the pressure bar switches. This generates an interrupt to the software to terminate all drive and the ROBOT then stops trying to run through something. The software then reads the "OOPs" file input to determine exactly from which direction the collision occured. Software then makes a decision and moves to free the ROBOT from the collision condition.

By augmenting the sonar with bump detectors a more reliable means of control should be achieved, since in many cases the sonar doesn't "see" what it just ran into. The HPC currently only detects collisions by noting that one or more of the drive motors has stalled. There are a number of strategies that the club must investigate and solve to midigate this most difficult problem. ....JJ

ARTICLE REVIEW

For the last few months I have been bringing you a review of various design articles written about robots. Once again for April I bring you the highlights of another rather different type of robot. In the March issue of the ROBOT BUILDER I reviewed the "Basic Educational Robot Trainer" called "BERT" for short. This months robot is called "KEN". The author/builder says that the name KEN doesn't stand for anything, he just thought that it was a nice sounding name. So if you want to dream up something to represent as an acronym for KEN, be my guest. I think it should be "Kenimatic Energy Neutator". Whats your guess?

This is a very interesting and simple machine and much can be learned from reviewing the design. It is a simple two chip design that takes the development through to the "autonomous wander mode" capability. Some of you may even want to build one for yourself as it is claimed that the completed basic machine can be completed for under a $100. The microprocessor selected also has much growth potential (additional I/O ports and unused functions) and much capability expansion can be added.

The the KEN project was developed by Mr. Freddy Eady. This name sounded like a pseudonym to me when I first read the authors name but upon further investigation, it seems to be valid.

His goal with this project was to reduce the overall project complexity of constructing a ROBOT for the first time builders by the use of stepper motor propulsion. This allows much simplified electronics as we will cover later.

He covers a lot about the theory of both unipolar and bipolar stepper motors. He didn't point out that there is also a tripolar motor or 3 phase stepper motor. Mr. Eady does a good job with both the stepper motor theory and the theory of operations of the entire system.

The complete schematics and instructions for KEN are available in the article and these should be enough to assemble the mechanical hardware and the circuit board.

It was very interesting that this simple design generates most of the major functions that our ROBOT has on the High Performance Card(HPC). The design of KEN's brain board consists of a Motorola 8748H microprocessor and a ULN2803 octal driver chip. This ULN2803 is some macho chip as it can drive eight 1/2 ampere loads at 50 volts DC per load. The microprocessor is configured as two eight bit ports, one input to receive the wheel motion pulses, and the output port to generate the pulses passed to the driver chip to activate the stepper motors.

The KEN system is extremely simple in that it only requires two stepper motors, two wheels, a GEL CELL, a CPU, a driver chip, a crystal oscillator and a 5 volt regulator as the major parts. As you can see, it is simple from the electronics standpoint. The EPROM has built in routines that generate pulses to operate the stepper motors. Additional routines detect a stalled condition of either one or both of the motors as it runs into an obstacle. Other subroutine branch conditions then reverse the motors and KEN backs away from the obstacle and turns and continues his travels.

I am concerned, however, that the software listing was not included in the article. I would have expected at least a flow chart of the software. He does indicate the availability of the listing on a BBS in the 516 area code or it can be obtained from the author. He states at the end of the article that if you don't like the way it works then change it. He doesn't indicate how the software is loaded into the CPU or the interfaces needed to load new programming. To expand the program within KEN, some type of support system would be needed. I haven't had time to obtain the literature on the 8748H yet to determine the exact nature of the additional capability available or what it takes to modify or replace the existing program in the EPROM. Some of you programmer type club members out there HELP, HELP, HELP.

This is a very interesting design and I think it has a lot of value to the first time robot builder. There are some basic building blocks presented here that the club members could easily expand on for bigger and better machines.

The electronics Parts kits are available from Mr. Fred Eady, 1217 McDonald Street, Fayetteville, TN 37334 for $41.

This article was found in RADIO ELECTRONICS magazine, April issue of 1991. I'll place a copy in the engineering note book. If you would like a copy of your own, see Roger or your editor and we'll see you get a copy..........JJ

THE ROBOTEER

At the April meeting we will be discussing Navigation. In past articles I outlined the overall design of my version of a Navigation system. In those articles I talked about the theory behind the system. I finally have an initial working version of a Navigation module and have now run head-on into reality.

The primary problem that I encountered with Dumbot is that it is an open loop system with respect to position determination. By this I mean when the program commands the robot to turn 38 degrees and move 2 feet, it has no way of determining where it actually is at the end of the move. It just works out the geometry mathematically and dead reckons to an estimated position.

The HPC sends pulses to the motors at a specific rate to attain a given speed. The only way it has of determining motor movement is by counting the pulses being returned from the opto isolators located on the motor shafts. The pulses are input via interrupt lines. What happens if the HPC misses a pulse interrupt? The software will then continue to drive the motor to makeup for one additional pulse, since the HPC counts the number of pulses from each wheel. In this way the software can determine when to stop the motors. Conversely, if the wheel slips on the surface then you get a motor pulse indicating movement traveled when in fact the distance was not actually covered.

When the HPC is commanded to move at a particiular speed, it ramps the speed up to the indicated speed and

continued on page 41

then after an appropriate time to complete the move, reduces the voltage to the motors to brake and stop at EXACTLY the right spot. Since, the HPC has no way of determining which direction the motors are turning, it must rely on detecting a period where the sensor wheel is just stopped and reversing which provides an interval with no pulse or the so called zero-point.

What happens if the HPC misses the zero point? It continues to get pulses, becomes confused and finally aborts with a catch-all abort of stalled motors (60008). If the robot were to use bi-phase encoders (two pulses per motor, with the encoders offset) you could determine the motor direction, and stop the robot before the wheels started backwards.

I have built program logic to attempt to overcome the problem and it works pretty well most of the time. When I get a motor stall abort I read the distance it moved and determine if an adjustment needs to be made. For example, suppose a command to move forward 10 units (one foot) was given and the distance returned is 8 units. This indicates that the robot didn't move the full foot as directed so another move of 2 is required to make up the difference. What happens if a distance of 12 is returned, when only 10 was commanded. What does this mean ? It turns out that this is a case of missing the zero-point. The robot actually went forward 10 units and then in reverse for 2 units for a total of 12. I Need to command a move of forward 2 to make up for the back-up problem. The same thing can happen on a turn. The amount turned can be less than or greater than the amount requested.

Another problem that needs to be addressed with regards to turns is that the HPC seems to turn too far, i.e. a 180 degree turn, looks to be really about 183 degrees. A possible way to overcome this problem is to derive a multiplier to adjust for the inherent inaccuracy of the hardware.

Another problem is drag due to different surfaces, which is a whole other can of worms. Thus far I have just been working in the Lab with a hard tile surface. When I get Dumbot working Ok on that surface, then I'll move to the carpet.

If Dumbot had some other means of determing where it was in the environment to compare with the open-loop dead reckoning of the motors then adjustments for any errors that accumulate could be made.

For instance if there were an infra-red beacon at a known location that the robot could trangulate on then the system would be closed. An internal compass would also help to compensate for angular errors. A caster wheel with X,Y counters (like a track- ball or mouse) would give an independant method of determining the X,Y motion of the robot independant of the dead reckoning of the motor control system.

A manually controlled robot does not have this problem since, YOU, the operator, close the loop and compensate by using the visual feedback to alter your control. Try manual control blindfolded and see how well you do !

I mention these problems, not as a criticism of the HPC as much as to bring to your attention the larger issue of open versus closed control systems and some of the issues that must be addressed in the real world. In order to eventually reach our overall goal of "Get me Beer", we find that it gets more complicated the closer we get to implementing it. It is easy to come up with a schema for solving this problem conceptually, but acomplishing it in the real world is a lot more difficult. But, Hey that's what makes robotics so challenging ! ......JB connecting the computers will be "Pcanywhere" for the ROBOT and "Aterm" for the monitoring computer.

3. "SONAR MONITOR". The proposal is for a hand held receiver that is tuned to the 40 khz sonar output. It should be capable of responding to the transmitted pulses, determine the amplitude and present it on some type of digital or analog display. No takers to work on this one yet.

4. SONAR BEAM FOCUSING. The present Polaroid transducers have a rather wide(30 degrees) beam spread. To better locate an obstacle, the beam width needs to be focused and reduced to as narrow a beam as possible. Your editor is working this item and is reported in the "SENSORS" column of this issue.

5. SONAR RETURN SIGNAL AMPLITUDE. Jerry B proposed this action item at last months general meeting. He needs amplitude information from the sonar return to allow more exact determination of the pointing angle to the reflector or obstacle. No assignee as of yet.

6. HPC ALTERNATE. The source for additional HPC boards seems to be exhausted. The other members and new members that would like to start a ROBOT project need the HPC functionally. The design of this functionally is the subject of this action item. Item unassigned.

7. NEW MOTOR CONTROL BOARD. (Same as #6)

8. TI VOICE BOARD REPLACEMENT There are a number of members now evaluating the COVOX concept and becoming adept at using the unit. I need your feedback reports of the integration of the COVOX into the ROBOT's functionally for the ROBOT REPORTER.

9. SINGLE BOARD COMPUTER. The electrical and the software groups must handle selection of this item for the club and if they don't get busy your editor is going to select one to start the development of my own machine.

10. HEADING SENSOR. Heading sensor is required as part of the design of our autonomous ROBOT system. This is a long term requirement and action remains open at the present time.

11. VOICE RECOGNITION ENHANCEMENT Tom Carroll submitted this item in the Feb ROBOT BUILDER. Special Interest Group(SIG) to support this item is yet to be formed. Item presently open. Sonar beam focusing study will help with the directional mike part of this project.

These are the action items for April. As you have ideas and\or potential solutions to any of these problems, call Jerry B or myself, set up a SIG or bring the idea to the general meeting........JJ

ACTION ITEM 4. SONAR BEAM FOCUSING

This section of the ROBOT BUILDER is to cover reports from the various SIGS working on the action items. I had started research on a column covering IR sensors for the SENSORS section of this issue, but I got so intrigued playing with the Polaroid transducer that I'll share some of the results obtained so far.

The sonar beam as produced by the Polaroid transducer has a wide beam (30 degrees) width. This does not give the ROBOT the definition and resolution required with out some post processing in the software for adequate obstacle identification.

There are ways to improve the resolution. One way is to focus the sonar energy to achieve a narrow beam. This has been reported in a number of papers on the subject. Another way is to combine information with an IR sensor which can be focused more accurately. This combination with the acoustical sensors have also been reported in previous research papers indicating some degree of success. A low dollar laser range finder would also be excellent. See last months SENSOR column for what it may take to develop a laser-based ranging device for our robot.

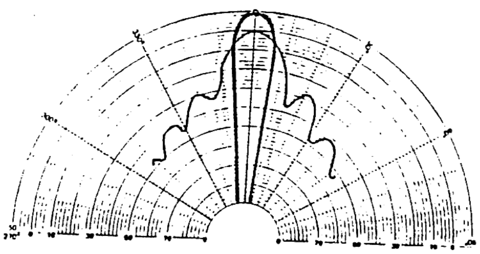
At present the only means the society robot has of determining where things are is through its sonar sensors. The sonar used by the ROBOT are "Instrument Grade" Polaroid ultrasonic transducers. The HPC controller controls and reads four(4) sonar sensors in a sequential manner.

Each transducer head acts as both a transmitter and receiver of the sound energy. When transmitting it sends a short ultrasonic pulse and when receiving it acts as a microphone to convert the reflected sound wave to an electrical pulse. The distance is calculated by multiplying the elapsed time from the transmission pulse to reception of the return signal times the speed of sound all divided by two.

The following data concerning the

accuracy of a focused beam is a synthesis of my own preliminary experimentation with the Polaroid sonar transducer.

There are several sources of inaccuracy in Polaroid sonar. The beam from the naked transducer is quite wide as shown in the radiation pattern. The estimated total beam width of the first horn tested is roughly 10 to 15



Estimated Beam Pattern

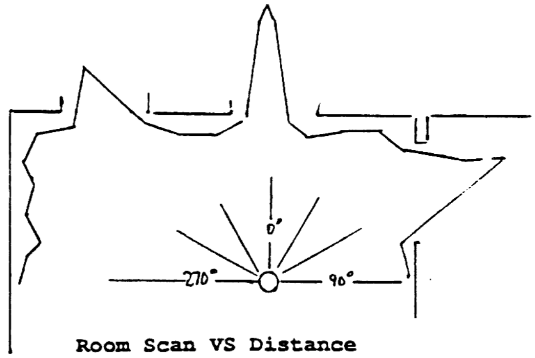
degrees. Actual practice seems to indicate that the horn acts to increase the transducer sensitivity so that it also detects echoes. In testing, the range-finder could detect a 1/4 inch diameter wire hung 3 or 4 feet in front of the horn. This seems that the horn and receiver combination is "TOO HOT" to me and I'll reduce the gain in further tests.

There had been reports of toilet paper tubes and other types of tubes being tried with "NO SUCCESS". It seems that the tubes or horns have to transform and match the acoustic impedances of the transducer to the air column to work properly.

The high directivity of all horns stem from the phasing and pressure effects, making the volume of the receptor greater from the front than from the sides or rear. The mouth, length, shape, and frequency range to be received, all determine the directivity.

The preliminary horn/transducer combinations that I have tried have all shown promise. Refer to the room diagram and it can be seen that the narrowed beam was able to look through the doorway when centered on the opening. These preliminary findings seem to indicate a lobe width of under 12 degrees or so with the first preliminary horns. I don't know yet whether I have just cut off the side lobes or have focused all of the acoustical energy down the throat and out the horn.

The sonar is capable of measuring distances to an object with a resolution of 0.12 inch through a range of 0.9 to 35.0 feet. The distance measured by a naked transducer is not necessarily the distance in the



Room Scan VS Distance

direction the sonar is pointing, since the width of the beam may cause an echo from a side lobe to be returned. The polar plot seems to indicate that the narrowed beam can look deeper into corners and more accurately determine the shape of a room.

As you look at the results from my pulser equipment, you will see a constant "under measurement" of all distances. I haven't yet had time to determine the source of the error. as I'm also trying to "get out a news letter". The errors could be temperature, humidity, or just the timing accuracy of the frequency standard on the board I am using.

I'll keep you informed as the investigation proceeds. Please feel free to contribute your ideas to this study. ....JJ

SENSORS

This section of the ROBOT BUILDER is devoted to the coverage of the various sensors that an autonomous machine may require to properly navigate through the maze of the real world.

This month I am going to walk through a real design problem of a set of sensors that are needed on the club robot.

As the club robot is operated in various modes such as a general wander mode at meetings and work shops, we began to observe some unusual movement or actions. He would normally be rather well behaved for a while but would crash into some things with out warning.

The question then becomes "why didn't the sonar sensors pick up a return from the obstacle"? Jerry B. has conducted extensive study into the sensitivity of the acoustical returns from various material types and densities. From this data we have to realize that the sonar sensors are not 100% reliable as a detection source.

The club(Jerry) came to the decision that there has to be some supplementary detection device added to detect obstacles that the sonar system

An **Experimental Robotic Vehicle** (a developmental model of the proposed Mars rover) avoids obstacles as it plans and follows the shortest route to a goal position.

Robotics Society of Southern California
P.O. Box 3227
Seal Beach, CA 90740

PM
28 MAR
1991

Roger Ruszkowski
18409 Renault
La Puente, CA 91744

Optoelectronic Ranging Sensor for Robotic Vehicle

Distance would be inferred from texture and changes thereof in the scene.

NASA's Jet Propulsion Laboratory, Pasadena, California

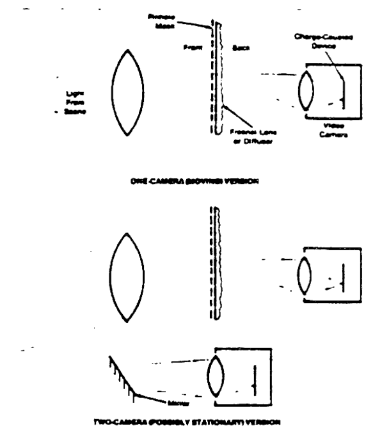
A proposed optoelectronic ranging system for a robotic vehicle would provide information on the distances to points in a natural scene. The system would have no moving parts, require little computation, and consume only a few watts of power. The system would be passive in the sense that it would not include any artificial sources of light, relying instead on sunlight reflected from the scene.

According to the basic principle of operation, pinholes arrayed in a gridlike pattern on an otherwise opaque mask would be used to subsample the very fine detail in the scene (e.g., natural textures at a scale of a millimeter or less). In a typical scene, this detail contains the majority of the information about whether or not a given point in the scene is in, near, or out of focus and is, therefore, a potential source of information as to whether or not the point in question lies at the focal distance. Essential to this concept is the fact that natural sunlight is so bright that even with sparse subsampling, there remains sufficient signal-to-noise ratio to justify the analysis of the patterns of high spatial and/or temporal frequency in the sampled light.

The pinhole mask would be placed at the prime focus of a lens of large aperture, short focal length, and high quality. The size of the pinholes would be at or near the size of the diffraction-limiting spot of the lens — typically, about 2 μm. A diffuser or Fresnel lens would be placed behind the pinhole mask to direct the light coming through the pinholes onto a charge-coupled-device video camera. The distance from the mask to the camera and the focus of the camera lens would be adjusted so that each pinhole would be imaged on one (and only one) picture element of the charge-coupled-device array.

As the vehicle moved, differences among the outputs of each of the picture elements of the camera would be computed for successive frames, and the magnitudes of the differences would be averaged spatially and temporally in such a way as to obtain a significant signal only from those picture elements at which the terrain was in focus. The concept that underlies this approach is that the intensity of light passing through a pinhole could vary rapidly and by a large amount only if the point of the terrain corresponding to the pinhole were in focus. Thus, if the camera were moved between frames by an amount approximately equal to or greater than the projected size of the pinhole on the terrain, then the frame-to-frame differencing would produce the full brightness texture at the pinholes where the scene was in focus, but would produce little or no brightness texture where the image was blurred by lack of focus.

In an alternative version, one camera would be focused on the back side of the pinholes as before, while another would be focused on small reflective rings around the pinholes on the front side. The area of a ring would equal the area of a pinhole. Spatial and/or temporal differencing would be performed, but according to a different plan. In this version, motion would not be essential, and focus or the lack thereof could be deduced on the basis of the strong nonlinear component in the spatial variation of brightness over the rings and pinholes in focus versus the nearly linear variation out of focus. In yet another variation applicable to both versions, the pinhole array could be corrugated in such a way that alternate rows of pinholes were on different focal planes, representing different distances.



The **Passive Optical Ranging Sensor** would provide a measurement of distance by the use of a pinhole mask to sample texture in the scene and thereby determine whether the portion of the scene corresponding to each pinhole is in focus (that is, whether is lies at the focal distance).

Planning the Route of a Robotic Land Vehicle

The distance traveled is minimized to the extent possible consistent with avoidance of obstacles.

NASA's Jet Propulsion Laboratory, Pasadena, California

An algorithm enables an experimental robotic land vehicle to follow automatically a route that is computed on the basis of terrain-height-map data (see figure). A computer executing the algorithm merges coarser global topographical data with finer local topographical data that it obtains through a stereoscopic video system as the vehicle moves along, then smoothes the merged elevation map, interpolates the map to evenly-horizontally-spaced grid points, and differentiates the map to produce data on slopes and roughness for use in calculations of traversability and the optimum route(s).

In processing the merged terrain-height data, the algorithm takes a probabilistic approach in which the statistical weight of each datum increases with the accuracy with which it is known and its nearness to the interpolated output point. Then the interpolation, smoothing, and calculation of slopes involve a weighted-least-squares fit of planes to small areas around the output points, using the residuals of the fit to estimate roughnesses.

The next task is to compute a cost function that takes account of both the distance traveled and the probability that each region to be crossed is traversable. In general, this function could take into account such factors as the energy expended in going uphill, the need to move more slowly over rough ground, and the effects of going in different directions. It could even include a negative cost to account for the desirability of gaining a closer look at a terrain feature of interest. However, in the experimental version, the cost function is computed from only two components: the distance traveled and the probability that the slope or roughness may be too large to allow the vehicle to pass safely.

If the video system, inclinometers, or mechanical feelers detect an excessive slope or other previously unidentified obstacle after the vehicle travels part of a tentatively planned route, the vehicle can backtrack. Therefore, the cost function accounts for the probability and cost of backtracking.

The main route-planning computation is iterative. It involves the use of one array of data that represent a forward growth from the starting position and another array of data that represent backward growth from the goal position, to integrate the cost of going from the starting point to each point in the grid and the cost of going from each point in the grid to the goal point. At each iteration, each point in the forward or backward array is replaced by the minimum of its previous value and the eight values obtained by adding, to the previous values of its eight nearest neighbors, the cost of moving from each such neighbor to the point (in the forward array) or from the point to each such neighbor (in the backward array).

At the end of the iterations, each point in the forward array contains the total cost of moving from the starting point to this point by the cheapest route, and each point in the backward array contains the cost of moving from this point to the goal by the cheapest route. The sum of the two arrays is an array in which each point contains the total cost of moving from the start to the goal through this point. The optimum path is the one along which the values at the points in the same array have the minimum value. This value equals the value in the forward array at the goal point and the value in the backward array at the starting point.