AIAA-94-1277-CP

LOW COMPUTATION VISION-BASED NAVIGATION FOR A MARTIAN ROVER

Andrew S. Gavin

Massachusetts Institute of Technology Artificial Intelligence Laboratory
Department of Electrical Engineering and Computer Science
545 Technology Square, Cambridge Massachusetts 02139

Rodney A. Brooks

Massachusetts Institute of Technology Artificial Intelligence Laboratory
Department of Electrical Engineering and Computer Science
545 Technology Square, Cambridge Massachusetts 02139

ABSTRACT

In the design and construction of mobile robots vision has always been one of the most potentially useful sensory systems. In practice however, it has also become the most difficult to successfully implement. At the MIT Mobile Robotics (Mobot) Lab we have designed a small, light, cheap, and low power Mobot Vision System that can be used to guide a mobile robot in a constrained environment. The target environment is the surface of Mars, although we believe the system should be applicable to other conditions as well. It is our belief that the constraints of the Martian environment will allow the implementation of a system that provides vision based guidance to a small mobile rover.

The purpose of this vision system is to process realtime visual input and provide as output information about the relative location of safe and unsafe areas for the robot to go. It might additionally provide some tracking of a small number of interesting features, for example the lander or large rocks (for scientific sampling). The system we have built was designed to be self contained. It has its own camera and on board processing unit. It draws a small amount of power and exchanges a very small amount of information with the host robot. The project has two parts, first the construction of a hardware platform, and second the implementation of a successful vision algorithm.

For the first part of the project, which is complete, we have built a small self contained vision system. It employs a cheap but fast general purpose microcontroller (a 68332) connected to a Charge Coupled Device (CCD). The CCD provides the CPU with a continuous series of medium resolution gray-scale images (64 by 48 pixels with 256 gray levels at 10-15 frames a second). In order to accommodate our goals of low power, light weight, and small size we are bypassing the traditional NTSC video and using a purely digital solution. As the frames are captured any desired algorithm can then be implemented on the microcontroller to extract the desired information from the images and communicate it to the host robot. Additionally, conventional optics are typically oversized for this application so we have been experimenting with aspheric lenses, pinholes lenses, and lens sets.

As to the second half of the project, it is our hypothesis that a simple vision algorithm does not require huge amounts of computation and that goals such as constructing a complete three dimensional map of the environment are difficult, wasteful, and possibly unreachable. We believe that the nature of the environment can provide enough constraints to allow us to extract the desired information with a minimum of computation. It is also our belief that biological systems reflect an advanced form of this. They also employ constant factors in the environment to extract what information is relevant to the organism.

We believe that it is possible to construct a useful real world outdoor vision system with a small computational engine. This will be made feasible by an understanding of what information it is desirable to extract from the environment for a given task, and of an analysis of the constraints imposed by the environment. In order to verify this hypothesis and to facilitate vision experiments we have build a small wheeled robot named Gopher, equipped with one of our vision systems.

1. PHILOSOPHY

In the design and construction of mobile robots vision has always been one of the most potentially useful sensory systems. However, it has also become in practice the most difficult to implement successfully. Here at the Mobot Lab we have designed a small, light, cheap, and low power vision system that can be used to guide a mobile robot in a constrained environment. At this point we are using as our target environment the surface of Mars. It is our belief that the constraints of this environment will allow the implementation of a system that provides vision based guidance to a small mobile Martian rover.

For many animals vision is a very important sensory system. In primates (and therefore humans) vision processing is the primary sensory mode and occupies a large portion of the neo-cortex. While it is clear that a variety of senses are essential to a mobile entity, be it robot or animal, vision has a number of substantial advantages. It is able to provide a survey of a fairly broad section of the world (approximately 200 degrees in humans) and at some considerable distance. The regular interplay of light and surfaces in the world allow internal concepts such as color and texture to be extrapolated, making object recognition possible. Additionally, vision is a passive system, meaning that it does not require the emission of signals. No other mode of sensation has all these properties. Chemo-receptors (smell and taste) are inherently vague in directionality. Somatic (touch) input is by definition contacting the organism, and therefore provides no input from distant objects. The auditory system is probably the second most powerful mode, but in order to provide information about inanimate (and therefore silent) objects it must be an active emitter, like bat sonar. However, it is only underground and deep in the ocean that the environment is uninteresting in the visual range of the spectrum.

Despite this, a useful artificial vision system has turned out to be very difficult to implement. Perhaps this is because the complexity and the usefulness of the system are linked. Perhaps it is also because no other system must deal with such a large amount of input data. Vision receives what is essentially a three dimensional matrix as input, two spatial dimensions and one temporal dimension. This input's relationship to the world is that of a distorted projection, and it moves around rapidly and unpredictably in response to a large number of degrees of freedom. Eye position, head position, body position, movement of object in the environment just to name a few. The job of the vision system is to take this huge amount of input and construct some small meaningful extraction from it. Nevertheless, whatever the reason for the difficulty in implementation, it is clear from ourselves and other animals that vision is a useful and viable sense.

For this project we had as our goal the construction of a small vision system that takes as a visual input the view from atop a small Martian rover. Its job would be to then quickly process this input in realtime and provide as output a small bandwidth of information reporting on the relative location of safe and unsafe areas for the robot to go. It might additionally provide some tracking of a small number of features "interesting" to the rover, for example the lander. The vision system was designed to be self contained. It has a pan and tilt camera head and an on board processing unit. It also draws a small amount power and exchanges a very limited degree of information with the host robot. The project has two parts, first the construction of a hardware platform, and second the implementation of a successful vision algorithm.

Professor Rodney A. Brooks, the head of the Mobot lab, has long been the champion of the belief that small cheap systems with a biologically based "behavioral" design can provide excellent results in real mobile robot applications [1]. He has demonstrated this with many small robots that have provided robust powerful performance with very small amounts of processing power. It is fairly widely believed that the Mobot lab's robots are some of the most successful fully autonomous robots built to date. They include notables such as Squirt [2], the tiny fully autonomous robot, and Ghengis and Atilla, a pair of highly robust small legged robots. Professor Brooks also believes that small cheap robots should be used in space exploration [3].

As to the second half of the project, it is our hypothesis that a simple vision algorithm does not require huge amounts of computation. That goals such as constructing a complete three dimensional map of the environment are difficult, wasteful, and possibly unreachable. We believe that the nature of the environment can provide enough constraints to allow us to extract the desired information with a minimum of computation. It is also our belief that biological systems reflect an advanced form of this. They employ constant factors in the environment to extract what information is relevant to the organism.

This theory has already been used in our lab to implement a mobile robot that is "among the simplest, most effective, and best tested systems for vision-based navigation to date" [4; 5; 6]. We believe that these ideas can be combined with what is known about the Martian surface to create a system able to provide useful information to a Martian rover. The few existing Martian surface pictures returned from the Viking landers show a flat landscape of fine dust with protruding rocks.

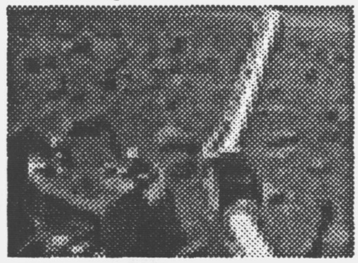
We have implemented several different algorithms on our system. Several of these attempt to extract the same navigational information from the scene. Each of these techniques is sensitive to different features, and so the simultaneous use of multiple approaches can yield added reliability. This algorithms are discussed below in detail.

We believe that it is possible to construct a useful real world outdoor vision system with a small computational engine. This will be made feasible by an understanding of what information it is desirable to extract from the environment for a given task, and of an analysis of the constraints imposed by the environment.

2. THE MOBOT VISION SYSTEM: AN ACTIVELY CONTROLLED CCD CAMERA AND VISUAL PROCESSING BOARD

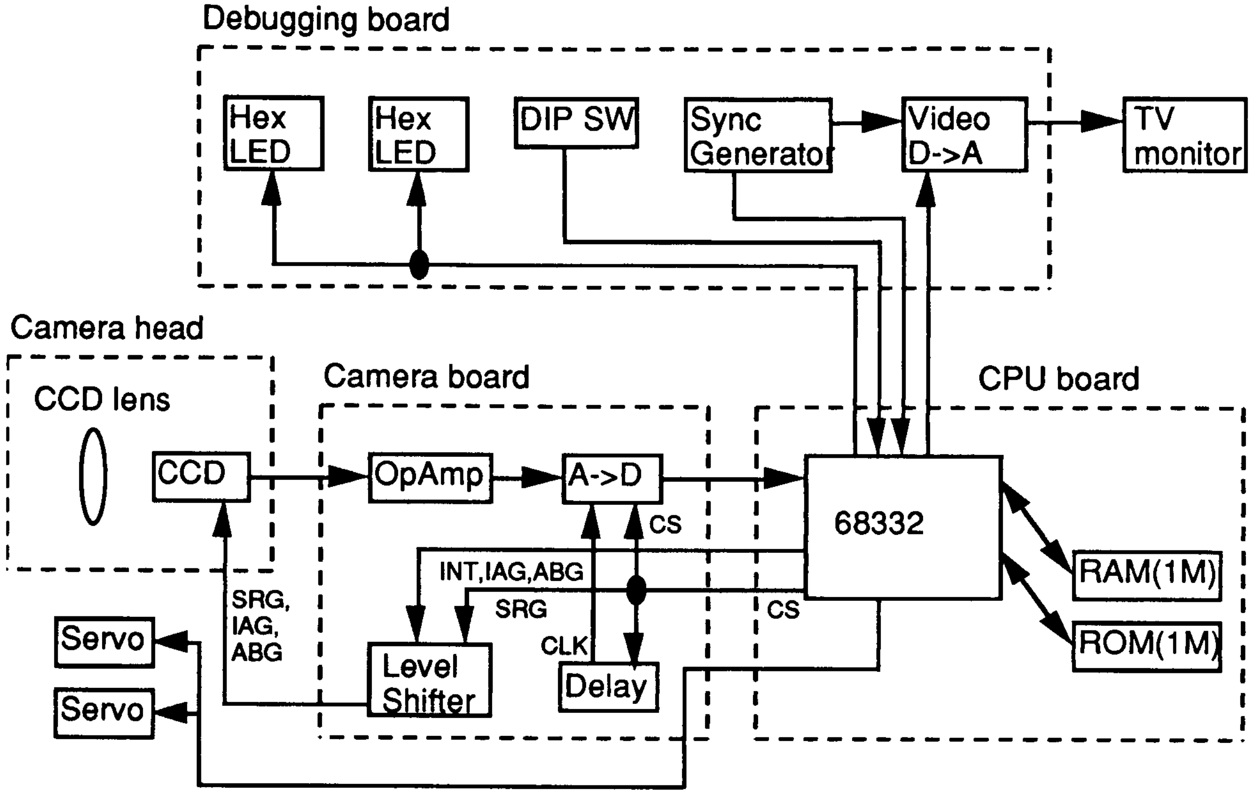
We had as our design goals the creation of a compact vision system which was cheap, simple, and had a low power consumption. We wanted to have everything needed for simple vision built in, including a significant amount of processing power. We chose the Motorola 68332 as the brain. This is an integrated microcontroller with on board RAM, serial hardware, time processing unit (TPU), and a peripheral interface unit. It has a decent amount of horsepower, essentially that of a 16-25 MHz 68020, and has a software controllable clock speed for power regulation. The 68332 was available in a simple one board solution from Vesta Technologies. To this we added 1 Megabyte each of RAM and of EPROM.

In accordance with our philosophy we decided on an image size of 64 by 48 with 256 levels of gray. These images are large enough that most important details of the view are clearly visible and recognizable to humans (see adjacent example image). Any number of gray shades in excess of about 64 are nearly indistinguishable, but 256 is convenient and allows for a neat single byte per pixel representation.



Our chosen resolution requires an image size of 3K bytes. We also chose a frame rate of 10-30 frames per second. We believe that both the chosen frame rate and resolution are more than sufficient to allow the simple visual tasks which we intend this system to accomplish. They also make for a total bandwidth of 30-90K bytes/second, which is well within the amount of data that the 68332 can transfer while leaving lots of free processing time. It takes approximately 7 milliseconds for the processor to clock and read in a single frame. At 10 frames per second this only consumes 7% of a 16MHz 68332's processor time. This allows 93% for processing the images, which amounts to about 186,000 instructions per image, or about 62 instructions per pixel. 62 instructions per pixel, while it might not seem like much, or more than sufficient to do many of the simple vision tasks we have in mind. If more calculation is required, then a faster version of the chip can be used to double the available power.

To the CPU Board we added two piggy back boards: a Camera Board, and a Debugging Board. The Debugging Board has 2 hex LEDs for output, 6 bit DIP switch for input, a reset and debug button, and some video circuitry. These circuits are extremely simple because the 68332's interface module allows nearly any kind of device to be mapped onto the bus. Even the video circuitry is very simple. We use two chips, a video level Digital to Analog Converter (Intech VDAC1800) and a video sync generator. The timing signals are run into the extra two input bits in our 8 bit input port and a trivial program on the CPU watches for transitions and outputs image data to the D to A convertor at the appropriate times. This allows us to display images contained in the CPU at 30 hertz to a video monitor with just two chips. Most of the work is done by the CPU. The system is capable of capturing images from the camera and outputting them to the video display at 30 hertz. Since video output consumes considerable CPU time it is not advisable to run the display at the same time as doing extensive vision calculations. However it provides a nice, easy output of the camera image in realtime which is essential for debugging, and for tuning the camera circuitry. There are also two switches on the Debugging Board to toggle the LED and video hardware for power consumption reasons. Additionally the system is completely capable of running without the Debugging Board attached, so that when the development phase is complete, it can be removed.



Block diagram for the Mobot Vision System

The Camera Board is the heart of the system. It contains a high speed Analog to Digital Converter (Sony CXD1175), some timing circuitry, the CCD level converters, and some output ports for our eye positioning servo actuators. It was one of the design considerations of this project that we should not use analog video at any point in our capture process (the video output on the Debugging Board is the only video circuitry in the entire system). At the resolution and frame rate we desire video adds unnecessary complications. It forces the use of high resolution CCD's, and a 30 hertz frame rate. Besides, since we desire to capture the frames digitally on the CPU Board there is no need to go through this convoluted intermediate stage. Instead we chose a low resolution CCD from Texas Instruments (TC211), clock the chip directly from the processor, amplify the signal, and read it through the A to D converter straight back into the processor. A simple program on the processor generates the needed timings, and since the A to D converter is connected as a memory mapped device, it simply reads in the pixel data. Since the CCD is a 192 by 165 device the input program merely clocks over two out of three pixels and lines in order to subsample the image at 64 by 48. A slightly modified version of the software is capable of grabbing 192 by 144 at the same frame rate, but results in a consequent reduction of the number of instructions available to each pixel (about 7 at 10 hertz).

The generation of the timing turned out to be quite simple [7]. The Integration signal (INT) used to signal the exposure timing was simply connected to a CPU parallel pin, as was the Image Area Gate (IAG) signal used to clock lines downward on the CCD. A third signal, the Anti Blooming Gate (ABG) was generated automatically by the 68332's TPU at no cost in CPU time. The only difficult signal was the Serial Register Gate (SRG) signal which shifts pixels outs of the current row. This signal must be as fast as possible and timed precisely to the A to D converter's sampling clock in order to get the peak of each pixel's signal. Fortunately, since the 68332 automatically puts out a chip select signal to the A to D converter to signal its possession of the bus, we used this as the SRG. By running this chip select signal through an adjustable delay and then into the convertor's sampling clock we were able to match the time it takes the CCD and amplifier to actually output the pixel. The Camera Board has several adjustable potentiometers, an adjustable delay knob, a signal offset knob, and a signal gain knob. All must be adjusted together in order to achieve a good picture.

Also on the Camera Board is the level shifter circuitry used to drive the CCD chip. This was specially designed with both simplicity and low power consumption in mind. The CCD chip requires its clock signals to be at specific analog voltages and so we explored three methods of converting the processor's TTL level signals. The first method was to employ the driver chips sold by the CCD manufacturer for this purpose. We rejected this because of the high power consumption which seems to be unavoidable in high speed clock generation circuitry. The second method was to use an operational amplifier to add analog voltages. Because we wanted a low power circuit, and also wanted to reduce the number of components, we chose the third solution, which was to use analog switches that could toggle the voltage at a reasonably high frequency and which were fast enough for the processor's clock rate. Our circuit resulted in a total power consumption for the Camera Board of less than half a watt (when it is supplied +5V, +12V, and -12V).

From the Camera Board a six wire cable connects to the camera head. Since the robot needed to insure as wide an angle as possible, we explored small short-focal-length lenses. Generally wide angle lenses have several merits, such as a wider depth of focus, which makes a focusing mechanism unnecessary, a smaller F number, which increases the image brightness, and an insensitively to camera head vibration. However, it is sometimes difficult for wide angle lenses to compensate for the image aberrations. After testing several aspheric lenses, pinhole lenses, and CCD lens sets, we decided to use a f=3mm CCD lens from Chinon (0.6" in diameter and 0.65" long, 5g weight).

In front of the lens we placed ND filters in order to prevent over saturation of the CCD. Our CCD is actually quite sensitive and needs at least a 10% pass filter to operate in room level lighting, sometimes it even needs considerably more. In order to expand the dynamic range of the camera the frame grabbing software is designed to calculate the light level of the image and adjust the integration (exposure) time of the frame correspondingly. This adds an extra 10 decibels of dynamic range to the system, allowing it to work adequately in subjective indoor light levels ranging from lights off at night to sunlight streaming in through several large windows.

The camera is mounted on top of two Futuba model airplane servo actuators (FP-S5102). These are very small and light weight, and allow easy and fairly precise control of the camera position by the CPU. The servo actuators are connected via the Camera Board to the 68332's TPU. This allows the generation of their Pulse Width Modulated (PWM) control signals with virtually no CPU overhead. These actuators give the camera both pan and tilt over a fairly wide range (170 degree pan, 90 degree tilt). The CPU has a number of simple routines that allow it to specify both absolute and relative positioning of the actuators, to read where they are, and preforms bounds checking to prevent the actuators from twisting the camera into themselves. The camera head and its servo actuators weigh 68 grams.

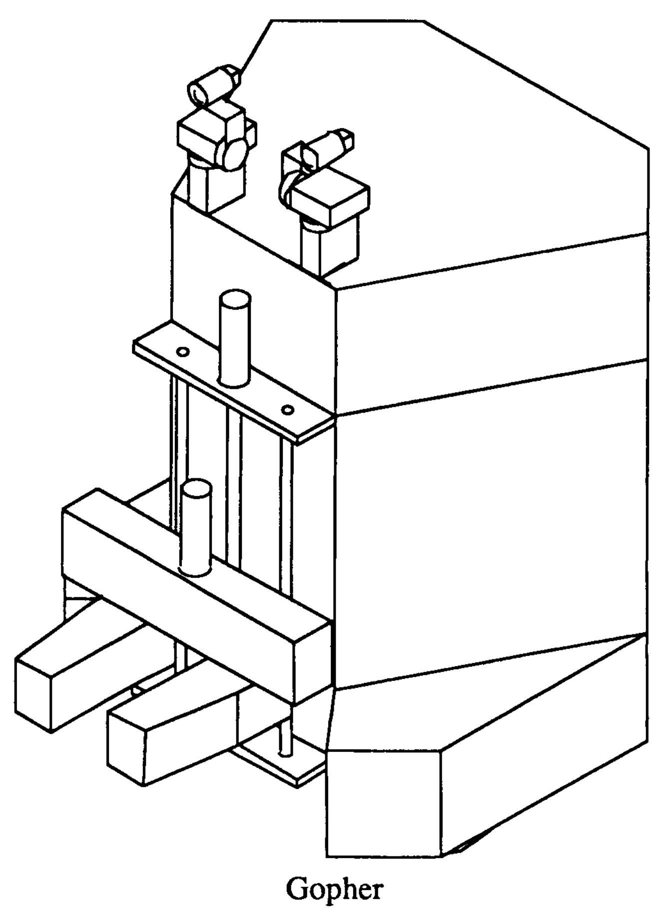
We have gone to a great deal of effort to minimize the power consumption of the Vision System and have been quite successful. The CPU board consumes 0.5 watts of power. The Camera Board also uses 0.5 watts. This means that the entire CPU and vision system consumes less than 1 watt of power. The video out circuitry on the Debugging Board requires an additional 1 watt of power, however there is a switch to disable this circuit, and since its use is for debugging this is not significant. The servo actuators also require some power. When idle they consume a mere 0.05 watts. Unless they are being moved constantly their average power draw is barely more than this.

Cost was also a significant factor in our design. The Vesta CPU Board costs $300. One megabyte of static RAM costs $200 (one could use less to save money, a megabyte of EPROM costs $25 (again one could use less to save money), two servos cost about $130, the CCD costs $35, the driver chips $30, the analog to digital convertor $25, the video chips $50, the power convertor $65, and miscellaneous other chips about $10. This is a total cost of around $700, many significant components of which could be eliminated to save money. One could use less RAM, or forego the servos or Debugging Board, possibly bringing a useful system as low as $350 in parts.

Overall this system is a small, cheap, and low power vision solution. It provides the input of 64 by 48 pixel 256 level gray scale frames at 10-30 hertz from a small camera with CPU controlled pan, tilt, and dynamic range, as well as about 62 680x0 instructions per pixel of processing time at 10 frames per second. All of the electronic circuits fit in a space 15 cubic inches big, consume less than a watt of power, and cost about $700 dollars to build. The available processing time is sufficient to do simple calculations such as blurs, edge detections, subtractions, the optical flow normal, thresholding and simple segmentation. A number of these can be combined to provide useful realtime vision based information to a robot.

3. GOPHER: A PROTOTYPE VISION BASED ROBOT

In order to fully test our system in a real environment we have been building a small vision based robot, Gopher (see diagram). This robot is based on a R2E robot from ISR. The robot is quite small (15 inches



Gopher

tall, 10 inches wide, and 12 inches long). It is wheeled, has a gripper on the front, and a number of sensors, including shaft encoders, touch sensors, and infrared sensors. These motors and sensors are controlled by a number of networked 6811 processors. A master 68332 processor, runs the behavior language which was designed here at the MIT Mobot lab. We have added a flux gate compass accurate to 1 degree and two of our 68332 based vision systems. The boards are all contained within the R2 case, which has been extended vertically for this purpose. We have mounted one of our CCD camera heads on its dual servo base on top of the robot, adding an extra 3 inches to the robot's height.

The Gopher robot provides a small and relatively simple low cost robot with an independently controlled camera and vision subsystem. We have used this system to implement many simple visual based tasks in a coordinated and integrated fashion.

4. Images: How good is 64 by 48

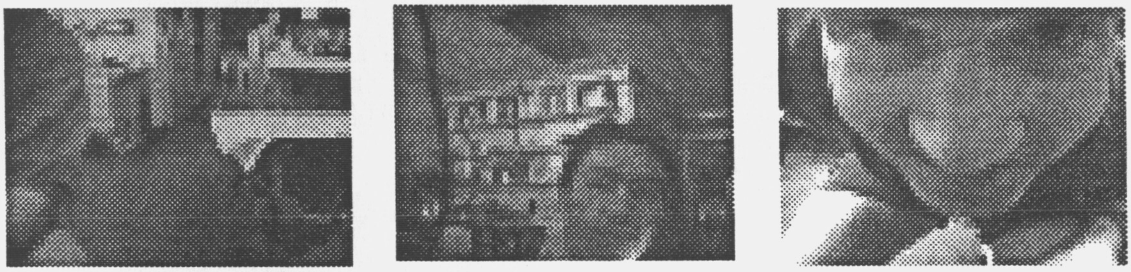
We have equipped the Vision System with a relatively wide angle lens (approximately 60 to 70 degrees). This is most useful for robot applications because the relative characteristics of space and objects around the robot are of more concern than the specific details at any one point.

Human perception of these images is quite good (see example images below). Objects approximately a meter square are easily visible at 20 feet. When angled down toward the ground the camera gives a good view of the space into which a forward driving robot will soon be entering.

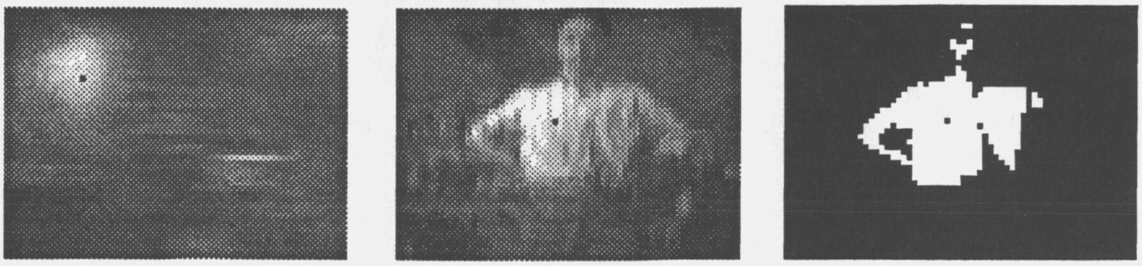
When all of the Vision Board variable knobs (adjustable delay, signal offset knob, and signal gain knob) are tuned properly the system captures an image with a full range of gray scales, which means a smooth clear image to the human eye. These parameters, while interrelated, can be tuned to a specific instance of the system in a few minutes. There is then little need to deal with them again unless a major system component (such as the amplifier, analog switch, or CCD) is exchanged.

The system is however fairly sensitive to light level. The CCD is very sensitive and requires at least a 10% pass filter to operate at normal light levels. In bright sunlight we usually add an additional 33% pass filter. By automatically adjusting the integration time in software we can cope with a moderate range of changing light levels, sufficient to encompass most operating conditions in an indoor environment. At some extremes of this range the image becomes more highly contrasting and less smooth. However this dynamic range is not sufficient to cover multiple environmental extremes, for example outside under sunlight and nighttime. To cope with this additional hardware would be required to increase the dynamic range. We have considered several other options. Filters could be changed as conditions vary, either manually, mechanically, electronically, or possibly using a kind of self adjusting filter (as found in some sunglasses). Additionally we are exploring the possibility of using a CCD with an electronic shutter. This would allow for a significant increase in the dynamic range, but would complicate the production of the CCD control signals.

Example pictures from our system (64 x 48, 8 bit gray)



Blob Tracking



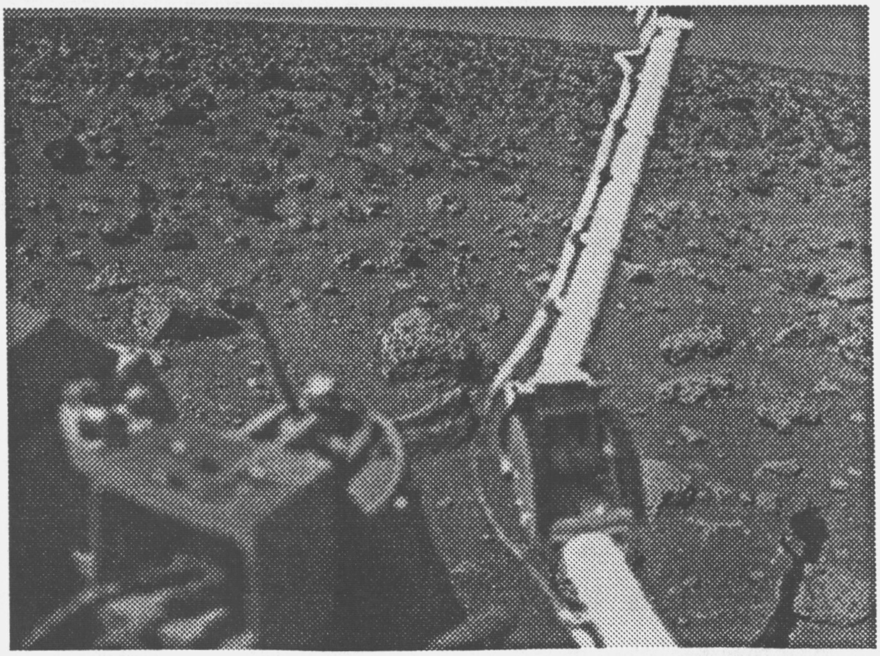
Black dot tracks the blob on this motion blurred image | Black dot marks the center of the segmented shirt | The shirt can be segmented by intensity

5. SOFTWARE

The Mobot Vision System is programmable in either 68332 (680x0) assembly or in C using the Gnu C Compiler (GCC). We have written a variety of basic routines. These setup the system, grab frames, actively adjust the integration time based on image light levels, output video frames, and move the camera via two servo actuators.

It takes approximately 7 milliseconds to grab a frame. This means that 10 frames per second occupies 7% of the CPU, leaving 62 assembly instructions per pixel free at this frame rate. We have coded in assembly a number of basic vision primitives. For a 64 by 48 8 bit image they have the following approximate costs: Blur (11 instructions per pixel), Center/Surround (11 instructions per pixel), Sobel edge detection (6 instruction per pixel), image difference (1 instruction per pixel), Threshold and find centroid (worst case 6 instructions per pixel).

As can be seen from the above figures a number of these basic operators can easily be combined to make a fairly sophisticated realtime (10 fps) visual algorithm. By making assumptions about the environment it is possible to construct algorithms that do useful work based on vision. For example, as a simple test case we have implemented code that thresholds an image and finds the center of any bright "blobs" in the image (see example images). This code requires at worst case 6 instructions per pixel plus some trivial constant overhead. We then use this information to center the camera on the "brightness center" of the image. The result is that the camera will actively track a person in a white shirt, or holding a flashlight. It is able to do so at 15 hertz while outputting video. This might not seem very useful, but by changing the thresholding conditions the camera would be able to track anything that can be segmented with a



Mars from the Viking Lander

small amount of processing. Intensity bands, the optical flow normal, thresholded edges, (and with filters) colored or infrared objects are all easy candidates for this technique.

We have used the Mobot Vision System to implement a host of other visual based behaviors. Here at MIT Ian Horswill has built Polly, a completely visually guided robot that uses an identical 64 by 48 gray scale image and a processor only somewhat more powerful than the 68332. Polly is designed to give "brain damaged tours of the lab." It is capable of following corridors, avoiding obstacles, recognizing places, and detecting the presence of people [5]. We have brought many of these skills to the Gopher system, and to Frankie, another lab robot based on the Mobot Vision System. The algorithms for avoiding obstacles and following corridors are easily within the power of one of these vision vision systems. Ian Horswill has also used one of these vision systems, slightly modified to add an additional camera, to do binocular gaze fixation.

6. Mobots in Space: Applications to Mars

NASA is planning on sending a small autonomous rover to Mars in the next few years as part of the Messur/Pathfinder mission. It is our belief that this rover could benefit from some vision based guidance, and that the approach used in the Mobot Vision System would be very suitable. A remote rover sent to Mars will be very limited by weight, size, and power consumption. The current rover design uses a digital CCD camera system quite similar to ours, as well as a low power processor. For power reasons the rover operates at very slow speeds, and so the algorithms we discussed here could run at a reasonable rate on its small processor.

The surface of mars as captured by the Viking Landers is fairly flat (at least where they landed) and regular (see Viking image above). It consists of a surface of tiny dust particles littered with an fairly Gaussian distribution of rocks. A rover's physical characteristics will determine what size of rocks are hazards and which are not. It would be useful for a vision system to be able to look forward and estimate roughly how much space in each direction there is until rocks that are too large to cross.

The first of our algorithms is texture based. This algorithm depends on a number of assumptions. These assumptions make it possible to extract useful data from an image in a reasonable time, however, it is important to be aware of the limitations they impose. If we assume that the ground is roughly flat, as indeed it is in the Viking images, then rocks that are farther away will be higher up in the image. Additionally the low resolution of the camera is convenient

A simple experiment employing a real Martian Image

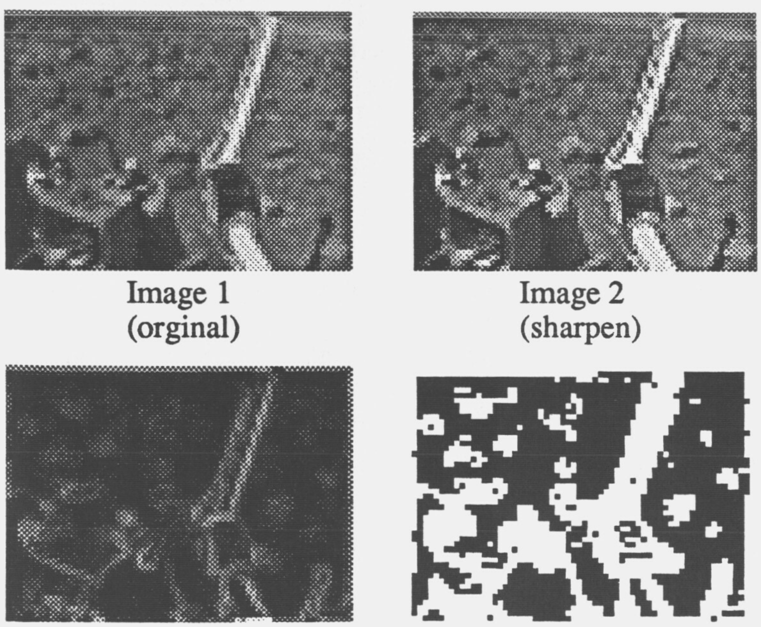
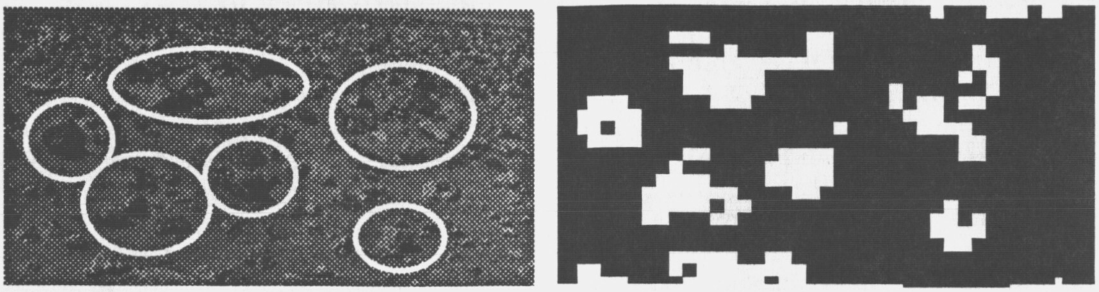


Image 1 (orginal) — Image 2 (sharpen) — Image 3 (find edges) — Image 4 (threshold)

Close up of rock segmentation



High detail original, large rocks circled — Segmentation of large rock clusters

because it will filter the dust and all small rocks into a uniform lack of texture. Texture, and therefore edges, is an indicator of objects or rocks. If the rocks can be segmented, then by starting at the bottom of the image and looking upward for "rocks" we can create a monotonic depth map of the distance to rocks in various directions. This is a simplified approach because it assumes that the ground is pretty flat. However it works surprising well on the Viking test image. Observe the series of images above. Number one is a 64 by 48 image as seen from the Viking Lander. Ignore the lander itself in the image, and assume a rover based camera can be mounted in the front where the rover itself will not be visible. In image 2 we have sharpened the edges of the images, and in image 3 we have used a simple edge detector. Finally image 4 was made by using an intensity threshold. Notice that the larger rocks in the original image are visibly segmented in the final image. By adjusting the threshold it is possible to segment for rocks of various sizes. This method is based on one devised by Horswill [4] and runs in realtime on the Mobot Vision System with processing to spare. We have used this system in several robots to avoid obstacles on flat uniform floors (usually carpets or cement). It works quite robustly.

The second algorithm is based on motion. We have implemented an algorithm that calculates the magnitude of the optical flow in the direction of the intensity gradient in realtime. If we make the assumption that the robot is moving forward roughly in the direction of the camera than the rate of motion of an obstacle is proportional to 1/d where d is the distance of the obstacle from the camera. This means that there will be very little movement in the center of the direction of travel, and more on the edges. Objects will accelerate rapidly as they approach the camera [8]. This large movement can be seen as increased flow, and with thresholding nearby obstacles can be loosely isolated. An even simpler strategy is to turn the robot in such a manner as to balance the flow on either side the robot. If there is more flow on the left go right, and vice versa. A large amount of flow in the

Optical Flow Examples

Light dots indicate flow. Arrow is preferred direction — Image of an office with a person walking by — Flow of center image, arrow points away from flow lower area of the image indicates an rapidly looming object. This technique is very similar to that employed by a number of flying insects. Balancing flow works well in a moving agent to avoid static objects. Some examples of this in action can be seen here. The line with the square on the end is an arrow indicating the direction the robot should go, the cross indicates an estimate of the direction of motion.

Because our algorithms run in real time, it is not necessary to convert to a complex three dimensional map of the world. We can convert straight from an image based map to a simple robot relative map. This is actually much more useful to a robot, and is vastly less computationally intensive. The realtime nature of our calculations applies a temporal smoothing to any errors made by the algorithm. This type of vision calculation tends to be very noisy, but with temporal smoothing, this noise is greatly reduced without have to resort to very difficult and time consuming calculations.

7. Conclusion

The images sampled by visual sensors, be they organic eyes, or inorganic cameras, contain a great deal of useful information about the environment, often available at more distance than other sensors allow. We believe that the limited amount of processing available on our Mobot Vision System, when combined with a low resolution image and reasonable assumptions about the environment will be sufficient to provide a robot with a good deal of useful information. We also believe that a small, light, cheap vision system such as this could be applied to many branches of the robotics field, including space exploration, indoor, and outdoor robots.

8. Acknowledgements

This research has been funded by the Jet Propulsion Laboratory Rover Technology Project, and we are grateful for their support. We would also like to thank Masaki Yamamoto, Ian Horswill, Olaf Bleck, Mark Torrance, Mike Connell, Dave Miller, and Anne Wright for their help.

9. References

[1] Rodney A. Brooks, "Intelligence Without Reason," *Computers and Thought*, IJCAI-91, 1991.

[2] Anita M. Flynn, Rodney A. Brooks, William M. Wells III, David S. Barrett, "Squirt: The Prototypical Mobile Robot for Autonomous Graduate Students," A. I. Memo 1120, MIT AI Lab, 1989.

[3] Rodney A. Brooks and Anita M. Flynn, "Fast, Cheap and Out of Control," AI Memo 1182, MIT AI Lab, 1989.

[4] Ian Horswill, "Analysis of Adaption and Environment," AIJ, 1992.

[5] Ian Horswill, "A simple, cheap, and robust visual navigation system," SAB92, 1992.

[6] Ian Horswill, "Characterizing Adaption by Constraint," *Proceedings 1991 European Conference on Artificial Life*, MIT Press, 1991.

[7] Area Array Image Sensor Products: Data Manual, Texas Instruments, Image Sensor Technology Center, 1992.

[8] Rodney A. Brooks, Anita M. Flynn and Thomas Marill, "Self Calibration of Motion and Stereo Vision for Mobile Robotics," 4th Int. Symposium on Robotics Research, 1987.