A Practical Stereo Vision System

Bill Ross

The Robotics Institute, Carnegie Mellon University

Abstract

We have built a high-speed, physically robust stereo ranging system. We describe our experiences with this system on several autonomous robot vehicles. We use a custom built, trinocular stereo jig and three specially modified CCD cameras. Stereo matching is performed using the sum-of-sum-of-squared differences technique.

1. Introduction

Range-finding systems, such as ladar (laser range-finding) and stereo vision, have proven particularly useful in the development of autonomous robotic vehicles. The product of these systems is typically a range image in which it is possible to detect obstacles, roads, landmarks, and other terrain features.

Stereo vision techniques offer a number of advantages to the designer of a robotic vehicle. Stereo relies on low-cost video technology which uses little power, is mechanically reliable, and emits no signature (unlike ladar). A stereo system also allows more flexibility; most of the work of producing a stereo range image is performed by software which can easily be adapted to a variety of situations.

To date, ladar, sonar and single-camera vision have proven to be more popular than stereo vision for use on robotic vehicles. Two machines which have used stereo successfully are JPL's Robby and Nissan's PVS vehicle. The PVS system, however, does not need to produce a complete depth map [1], while Robby's stereo does not need to operate at very high speeds. The primary obstacle to stereo vision on fast vehicles is the time needed to compute a disparity image of sufficient resolution and reliability. With faster computers becoming available every year, performance is already much less of an issue.

We have succeeded in building a fast and robust stereo system for use on our vehicles: two robotic trucks, NAVLAB and NAVLAB II [2], and an 8-legged walker called Dante. To achieve a useful level of performance, we have been willing to trade resolution, image size and accuracy to gain speed and reliability. As higher-performance computing becomes available for these vehicles, we will be able to take immediate advantage of it by increasing the dimensions and depth resolution of our images.

2. System Design

Outdoor mobile robots typically share several requirements in a ranging system: reliable performance in an unstructured environment, high speed, and physical robustness. These requirements are crucial for two applications we envisioned when building our system: low-speed cross-country navigation and high-speed obstacle detection.

Obstacle detection is an important part of the system which drives our autonomous trucks at speeds of up to 55 MPH. The requirements for this task are that all major obstacles to movement be detected and that the system run quickly enough to allow time to stop the vehicle or avoid the obstacle. In many cases, the vision algorithms used to navigate the truck have no knowledge of the three-dimensional structure of the environment and cannot perform obstacle detection. Obstacle detection, when performed, used to be accomplished by a second process using sonar or laser range-finding. Since these sensors are short range, they are impractical for use at high speeds where long stopping distances are needed. Our stereo vision system can be tuned, through choice of lenses and camera separation, to detect obstacles at a variety of ranges, even out to 100 meters or more (which would be needed for obstacle detection at 55 MPH).

Another application for stereo is to provide terrain information for a cross-country navigation system. In this instance, each range image generated by stereo is converted into a two-dimensional elevation map showing ground contours and obstacles. This map is then merged with previous maps and used by planning software to gen-

1063-6919/93 $03.00 © 1993 IEEE

erate a path for the vehicle to follow through the terrain. Since the planner requires detailed information, the stereo range image must be accurate and of fairly high resolution. Fortunately, cross-country vehicles in rough terrain do not need to move at highway speeds and, since they work from a map, can plan moves ahead of time. This means that range images do not need to be generated as quickly for this application as for obstacle detection.

These two applications suggest a single system which can be tuned to produce images of increasing quality in an increasing amount of time. Estimated system requirements are detailed in a table below.

In both cases, the computation requirements are considerable and demand a simple, fast stereo technique. Our system, already working well at slower speeds on several vehicles, will meet these requirements and more. Our system is able to do this because of a number of important developments: trinocular stereo, the sum of sum of squared differences matching technique, and careful attention to detail in the design and implementation of the various system components. Each of these aspects of the system will be discussed below.

	OBS. AVOID	X-COUNTRY
Minimum range	3m	3m
Maximum range	50-100m	25m
Depth resolution	40cm @ 15m	20cm @ 15m
CPU time	0.1 sec	5 sec
Image size	256*120	512*240

Typical requirements for two stereo applications

3. Trinocular Stereo

We chose to build a three-camera (trinocular) stereo system over the more usual two-camera model. The initial motivation for this choice was the hope that larger amounts of data would make the matching process easier. Moreover, the presence of the third camera was expected to help in the resolution of ambiguities when performing a match (consider the matching of two images of a repetitive pattern, such as a brick wall). Studies have shown that the benefits of a third camera outweigh the associated computational cost. Dhond and Aggarwal [3] found that the increase in computation due to a third camera was only 25% while the decrease in false matches was greater than 50%.

In our experience, the addition of the third camera produces a dramatic improvement in the quality of the range images obtained. Since our system requires the use of a long (1 meter) baseline, it may be that the third camera is important to bridging the dramatic disparities between the outer images when viewing objects close to the robot.

In a practical robot system, the trinocular approach has other advantages. When human help is far away, such as for our robot Dante, which will explore Antarctica, the third camera allows the robot to fall back on a two-camera system in the event that any of the three cameras fails. Finally, the trinocular approach makes good use of the typical 3-band video digitizer which can support one monochrome camera on each of the red, green and blue channels.

The Dante Robot

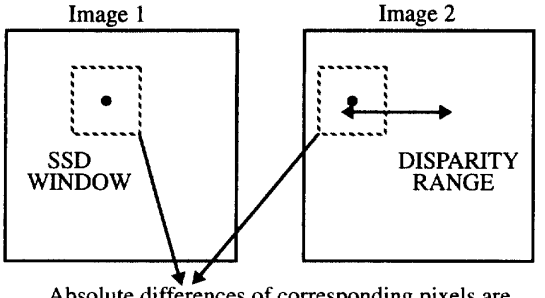
4. The SSSD Method

Once a set of three images has been taken, it is necessary to convert them into a range image. The fundamental principal behind this type of stereo is that, when the same scene is imaged by more than one camera, objects in the scene are shifted between camera images by an amount which is inversely proportional to their distance from the cameras. To find out the distance to every point in a scene, it is therefore necessary to match each point in one image with corresponding points in the other images. There have been many methods used to perform this matching (many successful), including feature-based matching, multi-resolution matching and even analog hardware-based matching. Our approach to perform this match is to use an SSSD (sum of sum of squared differences) window. This technique, developed by Okutomi and Kanade [4] has proven to have many advantages.

The SSSD method is simple and produces good results. The technique also places no limitation on the scope of the stereo match. This allows production of small,

low resolution images to be performed as easily as production of larger, high resolution images. Even more importantly, the technique easily allows the incorporation of our third camera. Because of its regularity, the SSSD method is easily adaptable to both MIMD and SIMD parallel machines. Lastly, as shown below, the SSSD makes it easy to compute a confidence measure for each pixel in the range image.

The SSSD method is used to determine which pixels match each other between our input images. When looking for matching pixels, we have several clues to help us. The first is that, due to the geometry of our cameras, which are arranged in a line, we know that matching pixels will occur on the same scanline in each image. Due to the baseline of the cameras, we also know that the disparity (horizontal displacement of a pixel) must fall within a certain range. For each pixel in the first (right-hand, in our case) image, we need, then, to look at a small range of pixels on a single scanline in each of the other images. The pixel in this range that produces the best match is considered to be same point in the real scene. Once we have this match, we can then immediately calculate the range to that point in the scene.



Absolute differences of corresponding pixels are summed to find the SSD for a given disparity value.

Computation of the SSD (2-camera case)

The trick, of course, is to figure out which in the range of possible pixels is the right match. For two images, the SSD method works by comparing a small window around the pixel in the original image to a window around each of the candidate pixels in the other image. The windows are compared by summing the absolute (or squared) differences between the corresponding pixels in each window. This yields a score for each pixel in the range. The pixel with the lowest score has a window around it which differs the least from the window around the original pixel in the right-hand image.

The sum of sum of squared differences (SSSD) is simply the extension of the SSD technique to 3 or more images. In our case, we have three camera images; for each pixel we perform an SSD match between the right-hand image and the center image as well as between the right-hand and left-hand images. For each disparity "D", we must examine the window shifted by D pixels in the left-hand image and by only D/2 pixels in the center image. When the SSD of both pairs of windows has been computed, the two SSD values are summed to produce a single score (the SSSD) for that disparity value.

The size of the SSSD window is an important parameter in the stereo computation. A larger window has the effect of smoothing over small errors in the stereo matching while also smoothing away many of the details and smaller features in the image. We typically use as small a window as we can that will still produce a fairly error-free range image (typically, 10 rows by 20 columns). The SSSD window does not have to be square, and we find for our applications that it is better to widen the window, sacrificing horizontal resolution, than to increase the height at the expense of vertical resolution.

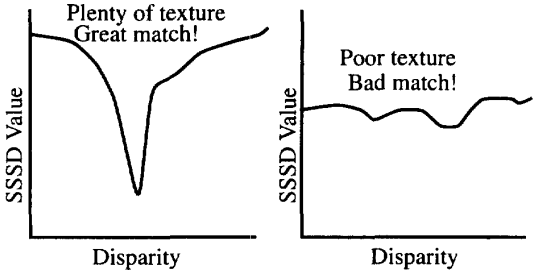
In Okutomi and Kanade's original SSSD system, variable window sizes for each pixel in the image were used to achieve the best results for each portion of the image. Also, disparities were sampled at the sub-pixel level (with interpolation of image pixels) to increase depth resolution. These enhancements, while giving superior results, are too slow for our application so they are not used.

We used a number of techniques to speed up our computation. Due to our wide camera baseline, we typically have a disparity range of 120 pixels to search. Instead of checking for sub-pixel disparity, however, we do the opposite. The wide baseline of the jig gives us acceptable resolution at longer ranges, but it gives us much more resolution than we need at short ranges (2cm resolution at 3m range). To speed things up, it is therefore possible to skip many disparities at the shorter ranges while checking the full range of disparities at longer ranges. This has the effect of equalizing our resolution over the range of the system while reducing the number of disparities calculated to about 50.

When performing the SSSD, we have improved performance by reversing the order of the computation. Instead of finding the SSD between two sets of windows and then summing these values, we first compute the differences between the whole images and sum them to produce a single image representing the match at that disparity. The window around each pixel is then summed to produce the SSSD for that pixel. The summation of windows can be done very quickly because we maintain rolling sums of columns to speed the computation.

Another technique we use to speed up computation is to reduce the sizes of the input images. For typical cross-country work, the full vertical resolution is not necessary so we use images of 512 columns by 240 rows. For obstacle avoidance, a smaller image of 256 by 120 pixels will suffice because small details in the scene are not important for this application.

For all the compromises made in the interests of speed, the range images produced by this system are surprisingly clean. Sometimes, however, the SSSD technique will break down when there is not enough texture in the image to perform a good match.



A good SSSD curve **A bad SSSD curve**

For example, an image of a smooth, white wall will produce the same SSSD score for every disparity; a graph of the SSSD values will look like a flat line. When there is plenty of texture, there is almost always a clear minimum SSSD value on the curve.

To make use of this phenomenon, we produce a "confidence" value for each pixel in the range image. This is a measure of the flatness of the SSSD curve. If a pixel in the range image has a confidence level below a pre-defined threshold, it can be ignored as unreliable. The confidence value for each pixel is computed by taking the average of the percent of change between successive SSSD values. For a given pixel, the confidence value C can be expressed as a function of the SSSD values for that pixel, S(d) for the range of computed disparities dmin through dmax:

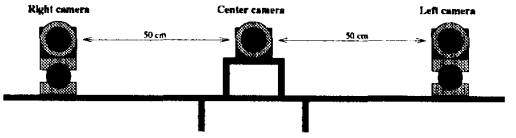
$$C = \left(\sum_{d=d_{min}+1}^{d_{max}} \frac{MAX(S(d), S(d-1))}{MIN(S(d), S(d-1))} - 1 \right) / d_{max} - d_{min}$$

5. Hardware Details

The development of several pieces of special hardware turned out to be critical to the success of our stereo system. The most complex item was the jig used to hold our three cameras. The SSSD algorithm we use requires that the stereo cameras be displaced by equal amounts along a straight baseline. Each camera is pointed in the direction precisely perpendicular to the baseline, and the roll and elevation of the cameras are adjusted to cause the scanlines of each camera to be vertically coincident. Our experiments have showed that this camera alignment must be quite precise if the system is to work well. While misalignment could perhaps be corrected in software, in the interests of speed it was decided to build a mechanical fixture which would guarantee alignment.

Unfortunately, we have found that typical CCD cameras and lenses exhibit considerable differences in image alignment with respect to the camera body. It was not possible to simply bolt the cameras into a precisely machined stand. Instead, an adjustable mount was needed for two of the cameras which allows them to be carefully aligned with the third camera.

The camera fixture, or "jig", consists of a rigid bar, 1 meter long, with mounting points for 3 cameras. The center camera is mounted to a fixed platform while the left and right cameras are attached to adjustable platforms.



Front view of stereo camera jig

The adjustable platforms have three rotational degrees of freedom and are designed to allow minute adjustments to be made in the orientation of the cameras. The platforms may also be rigidly fixed in place with locking screws to keep them well aligned during rough handling.

The choice of baseline (distance between cameras) is critical. With a very short baseline, there is little change between the three camera images. This translates to poor depth resolution. On the other hand, longer baselines will have the effect of decreasing image overlap (and thus, the effective field of view) and complicating the process of finding stereo matches. Our choice of a 1 meter baseline was a trade-off between these two concerns and was intended to give us good depth resolution without ruining our field of view. Due to this choice, depth resolution at 15 meters is not as good as hoped; however, at closer ranges resolution is still very good.

The cameras used are small Sony XC-75 monochrome CCD cameras. These cameras were found to be more mechanically sturdy than average. Our previous cameras had a slightly flexible mounting system for the CCD element which would slip out of alignment on the first bump. The Sony cameras were modified by adding a stiffening/mounting plate to the bottom. This plate serves to stiffen the camera body as well as to provide a better mount point than the standard single-bolt tripod mount. Another advantage of the XC-75 is the electronic shutter system which serves to prevent motion blur in the images.

Auto-iris lenses are a must for outdoor work. We chose autoiris lenses with a focal length of 8mm which give a moderately good field of view. 6mm lenses would have produced a greater field of view, but we found that these short focal length lenses introduced enough distortion to significantly degrade the quality of our results. Since we did not want to use CPU time to unwarp each image, we chose to substitute 8mm lenses.

As in the case of the cameras, some modifications to the lenses were necessary. We were unable to find any lenses which were mechanically sturdy enough to resist the vibration and bumps of our vehicles. The average lens is comprised of three important assemblies: the lens elements, a focus mechanism and a camera mount. The several sets of threads which comprise the focus mechanism are typically very sloppy, and, since they remain mechanically unloaded, they allow the lens elements to move relative to the camera mount. The movement of the lens elements causes a shift in the image falling on the CCD. In some lenses, a light tap on the lens body was enough to put the image out of alignment by as much as 10 pixels. Our solution to this problem was to discard all but the lens elements. We fashioned a new, single piece aluminum adapter between the lens elements and the Sony camera which allows no movement between the two. Of course, this also had the advantage of permanently fixing the focus which is a benefit on a moving vehicle.

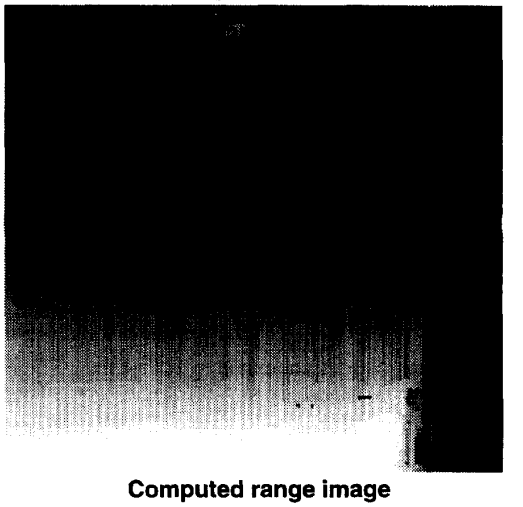
The digitizer used to capture the video images for processing was a conventional 24-bit color framegrabber. The 3 monochrome cameras were synced to each other and connected to the red, green and blue channels of the digitizer. The video gain feature on the digitizer was found to be useful for the balancing of the gains between the three cameras. We found that if the images were not close enough in gain-levels, our results were badly affected.

6. Results

Original right-hand image

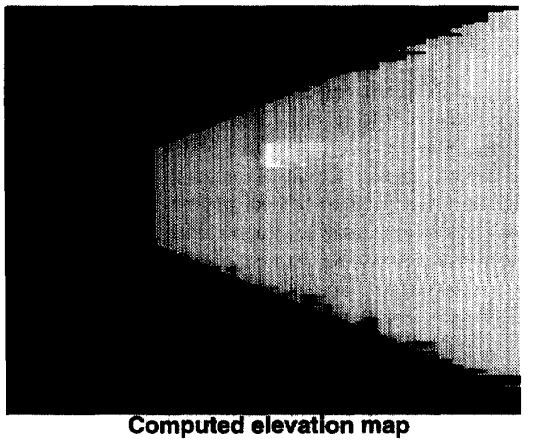
Results obtained to date with this system have been very encouraging. We have successfully driven our HMMWV truck through a field of obstacles at a nearby slag heap under autonomous control. The system has also guided our 8-legged robot (Dante) during outdoor runs. In December 1992, this system will be used to guide Dante during the exploration of a live volcano in Antarctica.

An example input image is shown above. This image is one of a set of three images taken with the stereo jig. The image shows a mostly flat piece of terrain containing a traffic cone. The image below is the computed range (disparity) image. The lighter-colored portions of this image represent areas closer to the camera while darker pixels are more distant. The traffic cone appears as an area of medium-gray in the center left of the image. The image contains a number of errors including the blobs floating at the top of the image and the fall-off at the right side. These errors are easily removed during post-processing when we generate an elevation map.



Computed range image

The elevation map, which shows the terrain from above as in a topographical map, is the map most commonly used to plan routes for our robots. The elevation map generated from this range image is shown below.



Computed elevation map

The elevation map is a view of the terrain seen from

above. In this map, the robot is situated at the center of the left-hand side of the image and is facing towards the right. Lighter areas in the image represent higher elevations while darker shades of grey represent depressions. The black border represents the area outside the field-of-view of the sensor. The traffic cone can be seen as the vertical white line near the center of the image.

Our stereo system has been implemented on several conventional workstations as well as on a number of parallel machines including an iWarp, a 5-cell i860 and a 4096 processor Maspar machine. Our algorithms have proven to map well to parallel machines, and, as can be seen in the preceding table, this has led to dramatic improvements in performance. The times for the iWarp are given because this machine is used on our NAVLAB vehicles.

MACHINE	IMAGES	DISPARITIES	TIME
Sun Sparc II	256*240	16	2.46 sec
16 Cell iWarp	256*240	16	0.35 sec
64 Cell iWarp	256*240	16	0.15 sec
Sun Sparc II	512*240	27	8.03 sec
Sun Sparc II	512*480	88	56.34 sec
64 Cell iWarp	512*480	88	2.18 sec

Times for a variety of stereo computations

This stereo system has been tested in a wide range of environments including a slag heap, a variety of roads, indoors, and in wooded areas. The results have been very good in almost every situation. The only failing of the system is that it does not perform well in areas which exhibit few features and little surface texture. These areas have included sky, newly paved roads and fresh snow cover (our Antarctic mission is in a mostly snow-free area). Even a small addition of texture to a scene, such as tracks in the snow or leaves on a new road, has proven to be enough to overcome this problem. The confidence measure, which is essentially a measure of surface texture, does a good job of detecting areas which are too smooth to yield good matches. This means that, while you may not get any useful data, the robot will not be misled by low-texture errors in the range image.

Other errors in the range image are inevitable due to the occasional presence of areas in the input image which fool the SSSD matching process. Typically, we have found that these errors are both small enough to ignore, and also produce features which are easily filtered out. For example, if a small object seems to be floating in the air in front of the robot, it is ignored.

7. Conclusion

We have developed a very successful stereo vision system which has proven itself through application to several real-world robots. The keys to the success of this system were a simple, straightforward approach to the software and attention to hardware details. This system has made it clear that stereo is a practical, competitive alternative to other perception systems in use on mobile robots.

Future work with this system will concentrate on two areas: increasing the speed of the system, and improving the quality of the images. Speed improvement is expected to be possible through further parallelization of the algorithm as well as the use of faster hardware. Improvements in the algorithm may include the use of variable window sizes and sub-pixel disparity checking.

8. Acknowledgments

The author is grateful for technical help from Mark DeLouis, Martial Hebert, Takeo Kanade, Hans Thomas, Chuck Thorpe and Jon Webb. Jim Moody, Donna Fulkerson and Carol Novak were a great help as editors. Chris Fedor deserves credit for the Dante photo. Many others were also a great help in this research.

This research was partly sponsored by DARPA under contracts "Perception for Outdoor Navigation" (contract number DACA76-89-C-0014, monitored by the US Army Topographic Engineering Center) and "Unmanned Ground Vehicle System" (contract number DAAE07-90-C-R059, monitored by TACOM). Partial support was also provided by NSF under a grant titled "Massively Parallel Real-Time Computer Vision".

9. References

1. Ozaki, Tohru and Ohzora, Mayumi and Kurahashi, Keizou (1989) An Image Processing System for Autonomous Vehicle. In SPIE Vol. 1195 Mobile Robots IV 1989.

2. Thorpe, Charles E. (editor). "Vision and Autonomous Navigation, The Carnegie Mellon Navlab". Kluwer Academic Publishers, 1990.

3. Dhond, Umesh R. and Aggarwal, J.K. (1991) A Cost-Benefit Analysis of a Third Camera for Stereo Correspondence. In International Journal of Computer Vision, 6:1, pp. 39-58.

4. Okutomi, Masatoshi and Kanade, Takeo (1991) A Multiple-Baseline Stereo. In CVPR proceedings, 1991.