REAL-TIME PATTERN RECOGNITION FOR GUIDANCE OF AN AUTONOMOUS UNDERSEA SUBMERSIBLE

Hoa G. Nguyen   Paul J. Heckman, Jr.   A. L. Pai

Undersea Artificial Intelligence and Robotics Branch
Naval Ocean Systems Center
San Diego, CA 92152-5000

Department of Computer Science
Arizona State University
Tempe, AZ 85287-5506

ABSTRACT

This paper reports the initial results of an effort to develop simple and fast vision algorithms on compact and imbeddable hardware for the guidance and control of an autonomous underwater vehicle. The specific application involves tracking underwater cables and chains. Feature points are identified in the underwater video images using a technique which combines segmentation by gray level and run length. Hough transformation is then used to find the straight line in the image. The process is performed at a throughput of approximately 1 image per second using a PC-bus video frame grabber and a PC/AT compatible micro-computer.

1. INTRODUCTION

Traditional methods for guidance of submersibles employ sonars, magnetic sensors, acoustic transponders and optical sensors. Of these, optical imaging sensors (e.g. TV cameras) are the systems of choice for applications that require high image resolution at close range, such as station keeping, control of manipulators, or cable following [1].

Underwater vision has traditionally been a difficult and unique problem because underwater light propagation exhibits such phenomena as backscatter, which reduces the contrast of the image, and forward scatter, which reduces the image resolution. There have been only a few research efforts in the area of underwater pattern recognition [2], and even fewer have been aimed at immediate applications.

Any vision process which controls the behavior of an autonomous submersible must be accomplished in real-time. Unfortunately, the trend in computer vision research in the last few years has been to produce increasingly powerful and complex (and hence non-real-time) algorithms. A majority of these algorithms require large and power-consuming mainframes or special-purpose computers (image-processing workstations connected to minicomputers, artificial intelligence workstations, and supercomputers) to achieve anywhere near real-time performance. These computers cannot be conveniently packaged in small electronic bottles or compartments. Neither can they be supported on the limited energy sources (batteries) which are available on current untethered submersibles.

Therefore a more application-oriented approach was considered in this research effort to address these problems. Image processing techniques incorporating simple, elegant, and optimized vision algorithms were developed for real-time vehicle control using small single-board computers. The Naval Ocean System Center's EAVE-WEST (Experimental Autonomous Vehicle-West) submersible [3] is being used as the testbed (see Figure 1). The vision hardware resides in the artificial intelligence/vision electronics bottle of this submersible and includes a single-board frame grabber and a PC-bus 80286 single-board computer, receiving input from an underwater video camera.

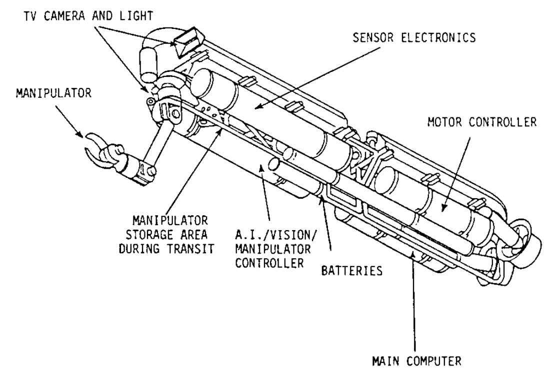

Figure 1. The NOSC Experimental Autonomous Vehicle (EAVE-WEST)

The initial software development reported here has been performed in the laboratory on an 80286 microcomputer system with a PC-bus frame grabber receiving input from a VCR. The frame grabber will be moved to the A.I./vision bottle of EAVE-WEST along with an 80286 single-board CPU for in-water testing. Our current objective is to demonstrate robust and practical image recognition algorithms using simple, off-the-shelf hardware.

2. OPERATIONAL CONSTRAINTS

The Ocean Engineering division at NOSC is heavily involved in developing vehicles for undersea search and recovery. The application selected for this project was thus geared toward such tasks. The targets chosen are the vertical cables and chains which are often connected to inflatable buoys, intrumentation buoys, or acoustic transponders. The operational scenario calls for the

Proc. 1988 IEEE International Conference on Robotics and Automation, Philadelphia, PA; pp. 1767-1770

vehicle to be guided to the moored object by sonar or directional hydrophones. The image recognition process takes over when the object and its cable are visible, and guides the vehicle along the cable to a point where the recovery process can be initiated. The vision computer keeps the vehicle centered on the cable as the cable is traversed by sending periodic steering information to the vehicle controller.

Operational constraints for targets such as cables and chains under the condition described above--as can be seen in Figure 2 (underwater video image of a buoy)--include:

a. Straight and elongated shape. The width of the target in the image is dictated by the type of cable or chain used, the field-of-view of the lens, and the distance from the target to the camera. The maximum width should account for variations in distance and for the spreading due to forward scattering in turbid water; while the minimum width should be greater than 1 to eliminate single-pixel noise.

b. Approximately vertical major axis. Arbitrary limits of +/- 30 degrees from the vertical were used for this initial effort. These can be refined by calculations using specific buoy buoyancy, cable weight and water velocity.

c. Gray-level segmentable target. Figure 2 shows that in natural light the target is darker than the background due to the scattering in the background water. When directly illuminated by an artificial spot light, the target will be lighter than the background. The vehicle controller computer must inform the vision computer whether natural or artificial lighting is being used.

d. Blurred boundaries. The images will tend to be blurry due to the physical properties of forward and back scattering of water. This constraint necessitates the use of recognition algorithms which do not require nicely defined edges and can tolerate gaps.

e. Target recognition speed of approximately 1 image per second. This update rate is necessary to control underwater vehicles.

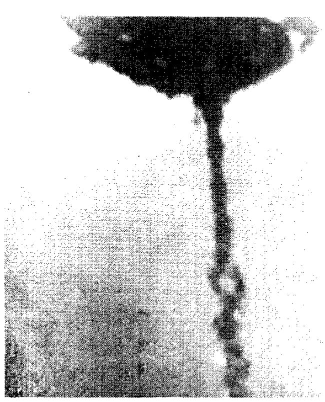

Figure 2. Buoy chain

3. ALGORITHM DEVELOPMENT

Our algorithm can be divided into three parts:

a. Identifying feature points. This operation should be 1-dimensional to allow faster processing.

b. Linking the feature points into a line. This operation should be able to reject extraneous points not belonging to the line, and must be able to tolerate gaps.

c. Determining the location and orientation of the line and reporting to a higher-level vehicle controller.

After these three processes have been accomplished, the algorithm is optimized to achieve the necessary speed.

4. FEATURE POINTS EXTRACTION

The approach found to be most effective for identifying feature points was a combination of segmentation by brightness and run length. A gray-level threshold is picked from the histogram of the image. As the image is scanned horizontally, continuous horizontal groups of pixels are identified which have values below this threshold (constraint 2c, for natural light images) and which have run length between the width limits stated in constraint 2a. The centers of these horizontal segments, our feature points (which may be part of the skeleton of the cable or chain), are marked with white dots in Figure 3.

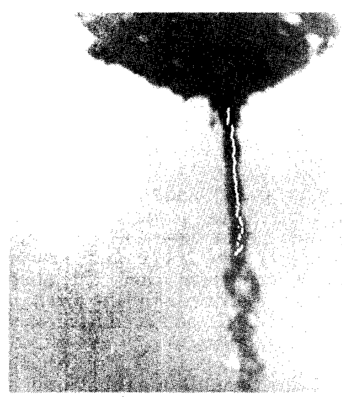

Figure 3. Feature points identified

The target/background gray-level threshold is currently set at the mean of the histogram array. Figure 4 shows the histogram of Figure 2. The target falls on one side of the mean (Mu) of the histogram. Setting the threshold at the mean will allow us to discard half of the image. Further discrimination is achieved using constraint 2a--the target is thinner than other dark areas of the background.

5. STRAIGHT LINE IDENTIFICATION

Several methods for linking points into a straight line were investigated, including chain coding [4,5] and least squares fitting [6]. With these methods, every feature point in the image contributes to the estimation of the line. In the present application, variations in the brightness of the background contribute extraneous clusters of feature points. It is desirable to have only those points which form the longest linear cluster determining the location of the line. The Hough transform was found to be a better method for linking feature points. The Hough transform maps each feature point in the image space into a line in a new parameter space in such a way as to make collinear points map into intersecting lines [7,8].

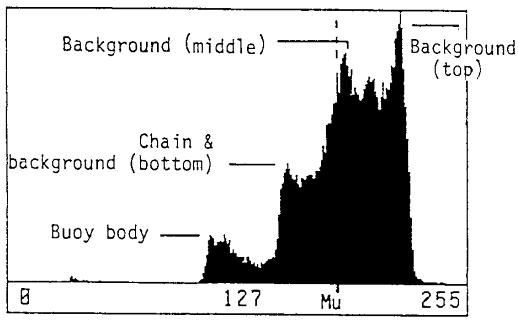

Figure 4. Histogram of buoy chain image

One approach for using Hough transformation to find straight lines involves mapping the feature points from the x-y space into the slope/intercept space [9]. The equation of a line in x-y space is

$$y = mx + c$$

where m = slope of the line, and c = y-intercept.

This equation can be rewritten as

$$c = -xm + y$$

This is also a linear equation in the m-c space, with x = slope and y = c-intercept.

For each feature point identified in the x-y space, the coordinates (x_i, y_i) are used to find the associated line in the m-c space (see Figure 5). These lines are kept in a cumulative 2-dimensional array (m,c). Each line in the m-c space increments the elements in the (m,c) array through which it passes. The element with the highest value--at (M_0, C_0)--is a result of the intersections of the largest number of lines in the m-c space. It also represents the longest linear cluster of feature points in the image, which has slope M_0 and y-intercept C_0. The accuracy and noise tolerance depend on the resolution chosen for m and c. Presently m is the slope of angles at 1-degree intervals, and the resolution for c is 8 pixels.

The Hough transform is simple (linear); can tolerate gaps; and can accommodate noisy, jagged boundaries (by adjusting the resolution of c). Furthermore, the points which are not in the vicinity of the cable or chain do not influence the formation of the line. It is thus appropriate for this application. However, the x- and y-axes have been switched from conventional notation (x is now down, and y is across) to prevent infinite slopes since approximately vertical lines are being sought. The slope m is computed for angles at 1-degree intervals between +/-25 degrees from the vertical in pixel space (approximately +/- 30 degrees on the screen, due to the aspect ratio of the pixels). The resulting line with slope = M_0 and y- intercept = C_0 is shown in Figure 6.

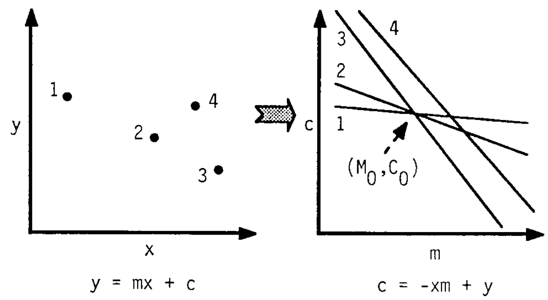

Figure 5. The Hough transform

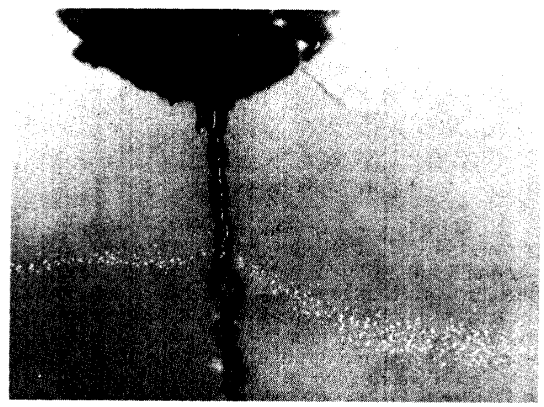

Figure 6. Chain detected

6. LINE LOCATION DETERMINATION

To determine the location of the cable or chain with respect to the direction of travel of the vehicle, the first and last points which contributed to cell (M_0, C_0) of the array can be used. They represent the two visible ends of the line in the image. In this particular application, however, only left/right steering information is required. This can be found by computing the horizontal coordinate of the line at the vertical center of the screen. The steering information is then reported to the vehicle controller computer.

7. OPTIMIZATION

Software algorithms implementing the three steps mentioned above were developed in FORTRAN, with care given to maximizing speed. Look-up tables were used in place of floating-point operations to compute the slopes of the lines. This resulted in a recognition speed on the order of 12 to 30 seconds, depending on the complexity of the image. To meet the required processing throughput of approximately 1 second per image (constraint 2e), two other processes were implemented.

a. Bypassing Noisy Rows: For the more noisy images, a major portion of the processing time is spent on the Hough transformation of false feature points which resulted from the gradual shift of the background brightness across the threshold. Therefore, for these images, bypassing rows in which too many feature points have been identified will improve the processing speed as well as the effective signal-to-noise ratio. Figure 7 shows the result with noisy rows bypassed. The implementation of this procedure resulted in a uniform 10 to 11 seconds per image processing throughput.

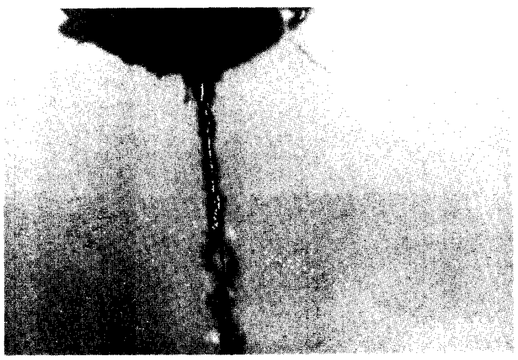

Figure 7. Noisy rows bypassed

b. Skipping Rows: One advantage of the Hough transform is its indifference to gaps in the image. Thus, another method of speeding up this process is to introduce artificial gaps by skipping horizontal rows in the image. Figure 8 shows the result with one in every five rows processed (at a throughput of 2 seconds per image). The required throughput of 1 image per second is achieved at one row in ten. Coincidentally, this is the rate at which the threshold determination process (histogramming and finding the mean) becomes dominant and limits any further improvement.

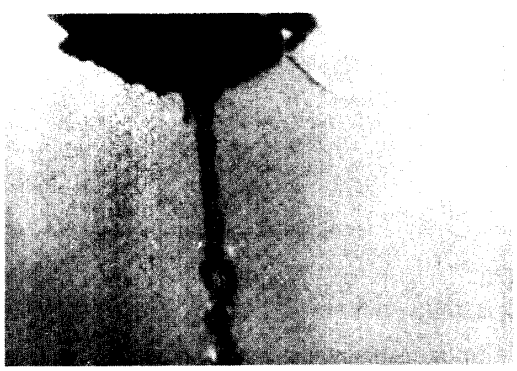

Figure 8. One in every five rows processed

The number of rows that can be skipped depends on the type and quality of the image. However, as more rows are skipped, the number of available feature points decreases. The number of feature points which contributed to the determination of the line and the total number of feature points are thus reported to the vehicle controller computer along with the suggested steering information. If the results are deemed unreliable, the steering suggestion is ignored and another image is processed before any course correction is initiated.

8. CONCLUSIONS

The vision algorithm presented here successfully met the processing throughput requirement for guiding an autonomous underwater vehicle along vertical cables or chains. Other methods for feature points identification and linking are currently being investigated for improved speed and/or reliability before the system is tested in the water. More efficient multiframe processing using short-term invariant features (e.g. brightness and location of past detection) is also being studied.

The approach described here can also be adapted to other underwater vehicle guidance problems, such as automatic docking and following cables on the ocean floor. Preliminary studies indicate that this technique is also applicable to acoustic imaging, which is necessary for extended range in turbid water.

This research effort has demonstrated that real-time vision-based guidance and control of autonomous underwater vehicles is possible with off-the-shelf, low cost, and imbeddable hardware. Combined with new developments in underwater imaging and imbeddable computers, this opens up a dynamic area of applications for further exploration.

ACKNOWLEDGEMENTS

The authors would like to thank Ron Reich for his help in obtaining underwater video images. This project was supported by the Independent Exploratory Development (IED) program at the Naval Ocean Systems Center and the US Navy-ASEE Summer Faculty Research Program.

REFERENCES

[1] *M. P. Shevenell*, "Survey of Autonomous Imaging," Oceans '84 Conference Proceedings, 1984; pp. 224-228

[2] *T. S. Huang* et al., "Toward Automatic Undersea Search Using Pattern Recognition Techniques;" Pattern Recognition, 1983; Vol. 16, No. 3, pp. 307-318

[3] *P. J. Heckman, Jr.*, "Free-Swimming Submersible Testbed," NOSC Technical Report 622, 1980

[4] *H. Freeman*, "On the Encoding of Arbitrary Geometric Configurations," IRE Transactions on Electronic Computers, June 1961; Vol. 10, pp. 260-268

[5] *A. Pugh*, Robotic Technology, Peter Peregrinus, Ltd., London, 1983; pp. 67-81

[6] *W. Cheney* and *D. Kincaid*, Numerical Mathematics and Computing, Brooks/Cole Publishing Co., Monterey, CA, 1980; pp. 267-271

[7] *P. V. C. Hough*, "Method and Means for Recognizing Complex Patterns," U.S. Patent 3069654, 1962

[8] *R. Duda* and *P. Hart*, Pattern Classification and Scene Analysis, John Wiley & Sons, NY, 1973; pp. 335-337

[9] *D. Ballard* and *C. Brown*, Computer Vision, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1982; p. 123