Calibration and Use of a Light Stripe Range Sensor Mounted on the Hand of a Robot

Gerald J. Agin

CMU-RI-TR-85-20

The Robotics Institute
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

November 1985

This work was sponsored by the Defense Advanced Research Projects Agency (DoD) ARPA Order No. 3597, Amendment No. 18, monitored by Air Force Wright Aeronautical Laboratories Avionics Laboratory under Contract Number F33615-83-C-1023.

Table of Contents

List of Figures

1. Introduction

This paper describes a system for direct measurement of three-dimensional coordinates by triangulation techniques using a light-stripe projector and solid-state camera mounted on the hand of a robot manipulator.

The ranging apparatus is used in a project whose goal is to produce a representational schema that can be used to compare shape descriptions of solid three-dimensional objects. We are working toward recognizing solid objects on a tabletop and determining their position and orientation. The models and matching procedures are based on "local features" of shape—portions of an object or scene that can be characterized as a portion of a flat plane, corner, edge, cylindrical region, high point, indentation, region of elongation, or the like.

We choose to acquire range data in a random-access fashion. We believe there are several advantages to this approach from a research point of view: Flexibility of viewpoint forces us to think in three dimensions rather than to rely on the two and one-half dimensional analysis that stems from arranging all the range data in the form of a depth map. By using all six degrees of freedom of the robot we are able to move around and view the scene from different points of view. Random-access data gathering focuses attention on the high-level representation issues. The controlling program must be aware of the state of its own knowledge of the scene and possible interpretations, so that it may control the acquisition of further data.

There are two ways of remotely measuring range: by time of flight and by triangulation. Time of flight measurements involve imposing amplitude or frequency modulation on a steerable laser beam, detecting the light reflected from a distant surface, and comparing the phase of the detected modulation with that of the original beam. The technique has so far only been demonstrated in the laboratory [10, 5], but instruments may be expected to be commercially available soon.

Triangulation involves observing the same point from two points of view. Stereo is one way of ranging by triangulation. The major obstacle to the use of stereo in practical vision systems is that it is very difficult and time-consuming to identify points in the two images that correspond to each other. Use of a projector and a camera bypasses this difficulty if the lighting, filters, or thresholds are such that the camera can distinguish the projected pattern from the rest of the scene. There have been many research efforts involving ranging that use one or more projected stripes [1, 2, 4, 7, 11, 13, 14, 15, 16]. Also, there are several commercially available systems for industrial applications.

A useful ranging system must cover the entire area of interest (a *scene*). Alternatives are to use multiple stripes that cover the entire scene simultaneously [4, 16], to steer the projected pattern with rotating mirrors [2, 5, 10, 13], to hold the projector and camera fixed and move the scene past them [7, 11, 15], or to keep the projector and camera in fixed relationship to each other but to move the combination about to scan the scene [1, 14]. We choose this latter approach—mounting the range finder assembly in the hand of a robot—because it gives us the flexibility of viewpoint and random-access operation that we require.

The next section of this paper briefly describes the components of our ranging system. Most of this description is abstracted from an earlier paper [3]. The accuracy and resolution of our system is limited by the precision of the robot and by the precision of our knowledge about the precise relationship among the camera, the projector, and the robot. The following two sections detail our efforts at improving the accuracy of the robot and calibrating the camera and projector. The final section presents some results, with measured accuracies and speeds.

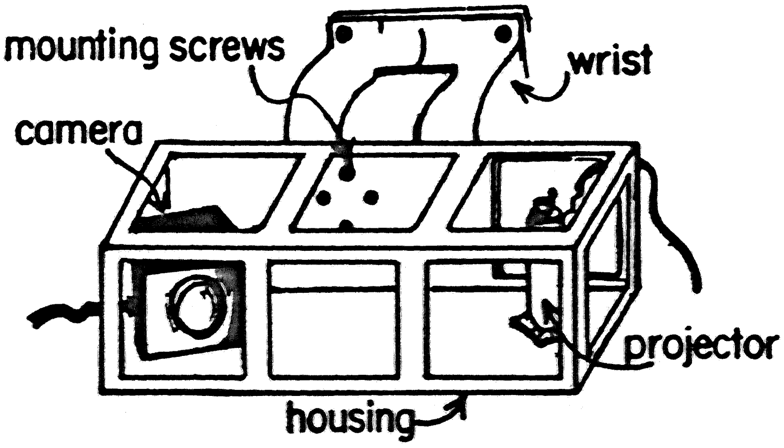

Figure 1-1: Camera and projector in housing

2. System Description

The range finder consists of a light-stripe projector and TV camera mounted in a rigid frame. The manipulator functions as a "programmable tripod," a controllable mount that may be used to place the range finder in an arbitrary pose.

The camera and projector are mounted in an aluminum housing, as shown in figure 1-1. The housing was machined from 4-inch tube stock. The housing gives some protection against accidental collisions, as well as providing a mounting place. The camera and projector are each held. in place from beneath by 1/4-inch camera mount screws. Four holes drilled in the back face provide for connection to the mounting flange of the manipulator. These holes are closer to the camera end of the housing to provide better balance, since the camera is heavier than the projector. The entire assembly, including camera, projector, and housing, weighs approximately 37 ounces (1.05 kg).

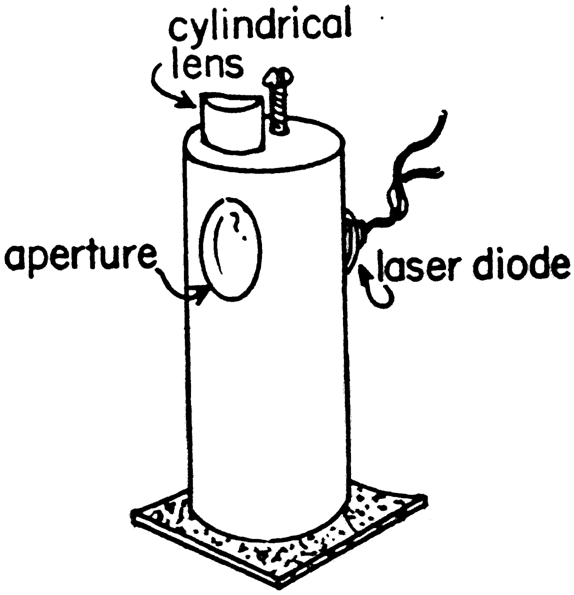

Figure 2-1: Light-stripe projector

The projector consists of a laser diode and a cylindrical lens in an aluminum mount. A closeup view of the projector mount is shown in figure 2-1. The diode, a CW-operated Aluminum Gallium Arsenide Injection Laser, emits 15 milliwatts of infrared radiation at a wavelength of 820 nm. (A visible wavelength would have been preferable, but diodes that had all the other desirable characteristics were only available in infrared wavelengths.) The cylindrical lens focuses the emitted light into a stripe or plane. The length of the stripe is determined by the spread of the original beam, and the width is determined by the size of the emitting area and the magnification of the lens. The size of the emitting area is small: about 40 μm x 15 μm. This configuration casts a stripe of light about 200 mm long and 1 mm wide at a distance of 300 mm from the lens.

The camera is a CID solid-state video camera, with 256 x 256 resolution elements. The lens is 25 mm, f1.4. The silicon photosensitive elements have a peak spectral sensitivity in the near infrared, so the laser stripe shows up well in the camera image. When an infrared spectral filter is placed in front of the lens to block the ambient illumination, only the stripe appears in the image. (Our laboratory is illuminated by fluorescent lamps. Incandescent lamps can not be used, since their infrared emissions would swamp the laser diode output.)

A Machine Intelligence Corporation VS-100 Vision System serves as a preprocessor for output from the TV camera. A programmable threshold converts the TV image to binary. Under good viewing conditions, the light stripe in the thresholded image appears white against a black background. The connectivity algorithm in the vision module has been specially modified for use with stripes that run in a generally top-to-bottom orientation [12]. It finds the tallest stripe in the image, throwing away spurs and other noise. The image coordinates of points at the centerline of the stripe, and the horizontal width of the stripe at each point, are made available to the computer that converts the stripe coordinates to three dimensions.

The projector and camera are mounted on a Unimation PUMA 560 robot with six degrees of freedom. The robot has been recently upgraded to "Mark II" mechanical specifications, and the control system includes VAL-II.

Overall control of the system (and the bulk of the computation) reside in a Vax-780 computer. A dedicated LSI-11 minicomputer handles the real-time requirements of communication with the manipulation and vision subsystems. The LSI-11 communicates over a 9600-baud serial line with the Puma controller, emulating input and output to the VAL-II system terminal. It uses a DRV-11 parallel interface to communicate with the vision module. A library of C language subroutines resides on the Vax for easy access to most of the functions of the manipulator and vision module. Communication between the Vax and the LSI-11 is over a 2400-baud serial line.

Conversion of stripe points in the image to three-dimensional coordinates requires knowledge of the precise positions and orientations of the projector and camera, and of the internal magnification parameters of the camera. We use the notation of homogeneous coordinates [8, 6] to describe positions, orientations, and the relationships among coordinate systems. Let **C** denote the coordinate system of the camera, such that the camera lens points along the z-axis; let **P** denote the projection matrix of the camera, including horizontal and vertical magnification factors; and let **S** denote the coordinate system of the projector, such that the light stripe lies in the y-z plane. Let **Q** be the matrix product $\mathbf{S}^{-1}\mathbf{C}\,\mathbf{P}^{-1}$, and let q_{ij} denote the elements of **Q**. Similarly let **R** be the matrix product $\mathbf{C}\,\mathbf{P}^{-1}$, and let r_{ij} be the elements of **R**. Then it has been shown [3] that an image point on the centerline of the stripe image $\mathbf{X}_i = [x_i y_i 1]^{\mathrm{T}}$ corresponds to the world point $\mathbf{X}_w = [x_w y_w z_w h_w]^{\mathrm{T}}$ according to the formula

$$X_w = K\,X_i,$$

where K is a 4 x 3 collineation matrix whose elements are given by

$$K = \begin{matrix} r_{11} - r_{13}\,q_{11}/q_{13} & r_{12} - r_{13}\,q_{12}/q_{13} & r_{14} - r_{13}\,q_{14}/q_{13} \\ r_{21} - r_{23}\,q_{11}/q_{13} & r_{22} - r_{23}\,q_{12}/q_{13} & r_{24} - r_{23}\,q_{14}/q_{13} \\ r_{31} - r_{33}\,q_{11}/q_{13} & r_{32} - r_{33}\,q_{12}/q_{13} & r_{34} - r_{33}\,q_{14}/q_{13} \\ r_{41} - r_{43}\,q_{11}/q_{13} & r_{42} - r_{43}\,q_{12}/q_{13} & r_{44} - r_{43}\,q_{14}/q_{13} \end{matrix}$$

An accurate determination of C and S is essential to accurate position measurement. C and S depend on two items: the *absolute* position of the robot end effector, which we denote T, and the *relative* positions of the camera and projector with respect to the end effector, which we denote C_T and S_T. The robot control system, residing in the VAL-II control system, is supposed to give an accurate value for the transform T, but our experience has been that that value can be improved upon by careful calibration of the robot. Even though we know several constraints on C_T and S_T from the way the mounting-cage is designed, calibration is still necessary to determine them completely.

The ideal way of carrying out calibration would involve the ability to measure the absolute coordinates of any point in the robot's working volume. Since this is impractical, calibration must rely on techniques that are available and easy to use. These include leveling with a spirit level, and measuring incremental motions with a ruler. Having the camera mounted on the robot is a very effective tool: reference to the camera image tells a lot about the robot's position and orientation.

3. Calibration of the Robot

Robot manipulators made for manufacturing tasks, such as the Puma 560, have specifications on repeatability, but not on absolute accuracy, either in position or orientation. The lack of absolute accuracy in position can be demonstrated by attempting to reach the same point in both "righty" and "lefty" configurations of the robot. If the robot is typical, the difference in positions may be on the order of 10 mm, a far cry from the repeatability specification of 0.1 mm!

Accuracy in orientation is not even specified, but in our case that accuracy is especially critical. Consider that our camera-and-projector ranging device is observing a point approximately 300 mm out from the robot's wrist. An orientation error of one degree in the wrist orientation can shift the point of observation by 5 mm.

Errors in position or orientation could be overlooked if the robot were made to scan only in a constrained way, such as covering the entire scene with a sequence of parallel stripes. Such errors would only shift the scene by a consistent amount. These errors do become a problem, however, when the same portion of the scene is observed from different viewpoints or with the striper in different orientations. Then the errors introduced by the system's inaccuracy will be in different directions at different times, introducing inconsistency into our observations. Our desire to access the scene from an arbitrary viewpoint imposes accuracy requirements on the system that are not necessary in other appl:cations.

One important source of robot inaccuracy is errors in the zero references of the joints: the correspondences between the readings of sensors in each joint and the actual physical position of that joint. The correspondence is established by the following procedure: The robot is placed in a standard position (the "ready" position). A computer program, POTCAL, which runs on the robot controller, drives each joint through its range of motion. The measurements from the transducer, during this motion, are saved on the controller's disk drive. It is the ready position that *defines* the zero references of each joint; any joint that deviates from its nominal placement at the time POTCAL is initiated will have that deviation reproduced in all subsequent moves that depend on that calibration file.

Given a robot with a known error in its zero reference, there are several ways to correct the error. In some cases the mechanical couplings between the motor and the joint itself can be loosened while the joint is brought to the corrected position. Or the robot can be commanded to the corrected ready position and the POTCAL procedure run again. We prefer the latter approach. We obtain a numerical estimate of the error in each joint's zero reference. Then we command the robot to its former ready position and move each joint by the negative of its error estimate to get a corrected ready position. Finally we run POTCAL to generate a new calibration file.

We make no attempt to correct joint 1, the waist rotation. Any inaccuracy in the joint 1 zero reference would only have the effect of rotating our global coordinate system a few degrees, but would not introduce any inconsistency into our measurements.

Joint 2 is the shoulder joint. When the angle of joint 2 is at zero degrees, the upper arm is horizontal, in the "lefty" configuration. At -90° the arm points straight up, and at -180° the arm is horizontal again in the "righty" configuration. If the zero reference for this joint is erroneous (say, in a positive direction) the upper arm will point slightly downward in the lefty configuration and slightly upward when righty.

To estimate the amount of error, place the upper arm in a horizontal position, the lower arm vertically downward, and the wrist in line with the lower arm, as shown in figure 3-1. This is

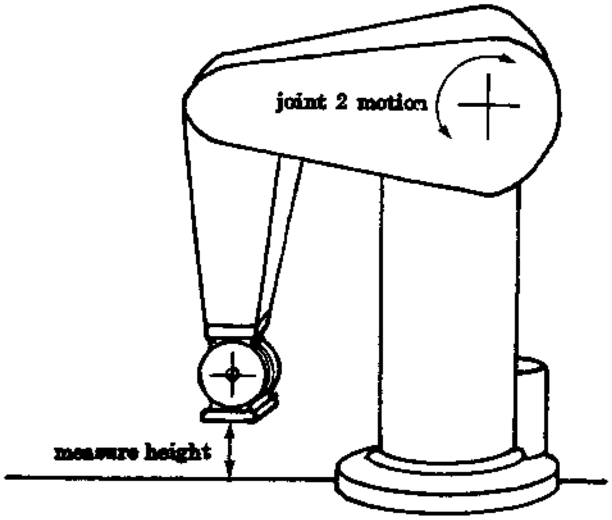

Figure 3-1: Measuring joint 2 error

accomplished by commanding joint angles of (90,0, -180, 90, 0, 45) degrees.[1] Measure the height of the wrist flange above the tabletop. Then switch the arm to its righty configuration, and move the wrist to approximately the same position as before. (It is not possible simultaneously to keep the lower arm vertical and to achieve exactly the same position as before, because of the asymmetrical 0.8 inch offset of the joint 4 axis.) Joint angles of (-51.656, -180, 0. -51,656, 0, 45) degrees will accomplish this. Measure the height again. The error in the joint 2 zero reference is the arctangent of half the height difference divided by 17.0 inches (the length of the upper arm).

A different procedure is needed to calibrate joint 3. When joint 3 is at zero degrees, the lower arm should be exactly perpendicular to the upper arm. At 90° the two segments are parallel, and at 180° they are perpendicular again. We extend the upper and lower arm segments in a nominally horizontal position and use a spirit level to adjust joint 3, as shown in figure 3-2. The nominal joint angles for this position are (90, 0, 90, 90_t 0, 45) degrees, but the correction measured in the previous step must be aided to the joint 2 angle. A spirit level is placed on the flat surface of the wrist assembly, joint 3 is moved under the control of the teach pendant until the surface is level, and the angle of joint 3 is read

[1] The VAL-ll conwiawfs for specifying Joint angles via the console terotna[1] only accept angles In the *range* -1803*<180. *Emn* though joint 3 must mate a positive rotation of 180° from its zero value, the VAL system requires ft to be typed in as negative 180°.

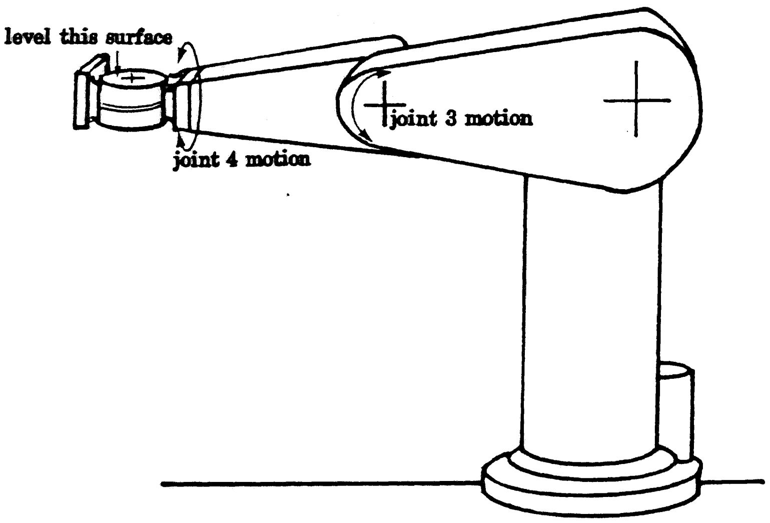

Figure 3-2: Measuring joint 3 and 4 errors

from the robot's system terminal using the HERE # command. The error in the joint 3 zero reference is the difference between the observed angle and 90°.

Note that because joints 2 and 3 have their axes parallel, this technique actually tends to zero the algebraic sum of the joint 2 and joint 3 errors rather than joint 3 alone. This means that even if joint 2 is incorrectly zeroed, a compensating error will be introduced into joint 3 to make the eventual orientation correct.

Joint 4 may be calibrated at the same time as joint 3. In the position shown above, joint 4 is leveled in exactly the same manner, but with the spirit level oriented perpendicular to the former direction.

If the robot is not mounted perfectly level, the above procedures may give incorrect results. This tilt may be cancelled out by measuring the error in both lefty and righty configurations, with the waist joint in all four quadrants, and averaging the eight readings.

When joint 5 is at zero degrees, the axes of joint 4 and joint 6 should coincide exactly. To measure the error, we place the arm so that joints 4 and 6 are nominally vertical, as shown in figure 3-3, with joint positions of (90, 0, −180, −90, 0, 45) degrees. With the spirit level on the upper surface of the light-stripe cage, we measure the motion joint 5 makes to center the bubble. Then we move to the

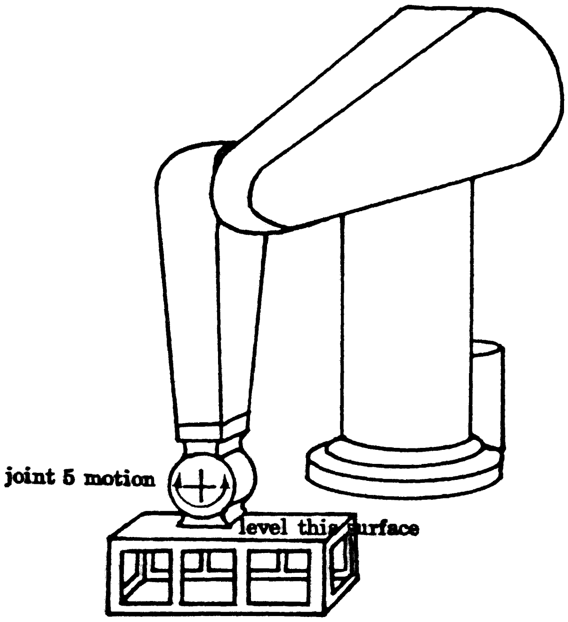

Figure 3-3: Measuring joint 5 error

"flipped" configuration of (90, 0, −180, 90, 0, −135) degrees, and measure again. The average of these two measurements is the error in the joint 5 zero reference. (Any difference between the two measurements reflects either a tilt in the robot mounting, an error in the positions of the first three joints, or an inaccuracy in the alignment of the robot.)

We do not calibrate joint 6 while calibrating joints 2 through 5 because we are interested in the actual orientation of the mounting cage, which may vary whenever the cage is removed and remounted. The adjustment of joint 6 is described in section 4.

Our experience has been that we have been able to measure zero-reference errors with a repeatability on the order of ±0.1°. An error of ±0.1° in joints 2 or 3 corresponds to a position error at the robot wrist on the order of 1 mm. An orientation error of ±0.1° at the wrist corresponds to a position error of about 0.5 mm at the sensed light stripe 300 mm from the wrist.

Correcting the zero reference angles of the robot considerably improves the absolute accuracy of the robot in position and orientation. However, errors still remain in the kinematic model of the robot—errors that are more difficult to measure and to correct. That errors exist is easy to demonstrate: For example, if we take the position shown in figure 3-3 and rotate joint 4 ±90° the configuration will be degenerate, and the top surface of the cage will probably not be exactly level (and there is no way to level it). It may be demonstrated that the surface does not even remain parallel to itself as joint 6 is rotated. It is not known how large the contribution of such errors to the overall accuracy of the range measurement system is, but it seems to be the most significant component of the system inaccuracy.

4. Calibration of the Camera and Projector

As stated above, conversion of stripe points in the image to three-dimensional coordinates requires knowing the position and orientation of the camera, C, and the projector, S. We also need the projection matrix of the camera, P. Since the projector and camera are carried on the robot, C and S are constantly changing. Fortunately, they may be calculated as the matrix product of the absolute pose of the robot wrist, T, and their relative positions with respect to the wrist, C_T and S_T. Since the camera and projector are adjustable in several degrees of freedom, and the camera magnification may be a function of focus, we need a way of rapidly measuring them whenever we suspect something has changed.

The variable part of the calibration information is kept on a computer file on the Vax. It is assumed that an approximately correct file is available at all times. Each step of calibration seeks to refine these measurements. At each stage, the correction required is printed out for the user to see, and the user may write the corrected information back on to the disk file.

Calibration is divided into four stages: wrist, camera aim, camera scale, and projector. These stages may be performed individually or as a sequence. A more detailed description of these steps was presented in an earlier paper [3]; for completeness they are summarized below.

Wrist calibration is equivalent to establishing the zero reference for joint 6 of the robot. Using current calibration information, the calibration program places the mounting cage in a position it believes to be level, with the camera and projector aiming in a horizontal direction. The user is asked to place a spirit level along the top of the mounting cage and adjust joint 6 of the robot using the teach pendant. The amount of correction required is reported to the user and used to update C_T and S_T.

The relative camera pose, C_T, is determined in the camera aim step of calibration. From the mechanical drawings of the cage, we know the coordinates of a point through which the camera's principal ray must pass, but the direction of that ray is unknown. To determine that direction, the light-stripe projector is turned off, the infrared filter is removed from the lens, and the vision module threshold is adjusted to make a contrasting spot on the tabletop visible. Using current calibration information, the camera is aimed straight downward, then the robot is moved to center the spot in the camera's field of view (which means the principal ray now passes through the spot). Next, the robot moves the camera upward a fixed distance. If the old calibration information is correct, the spot will stay centered in the field of view. Any amount by which the spot moves is used to calculate the error in camera aiming direction. The amount is reported to the user and used to update C_T.

The same contrasting spot on the tabletop is used for determining the camera scale. The camera is pointed straight down, from a known height above the tabletop. While maintaining the same height and orientation, the camera is moved horizontally in order to move the spot image up and down and from side to side in the camera's field of view. The ratio between physical motion and image motion is used to calculate the horizontal and vertical magnification of the camera. The ratios of new horizontal and vertical magnification to their previous values are reported to the user, and used to update **P**.

The camera scale calibration step also provides a check on camera "roll," the amount by which the camera is rotated about its principal ray. Although the roll angle is not expected to vary over time, it should be measured at least once. Roll is a component of the camera pose C, but it is not determined by the camera aim calibration procedure. During camera scale calibration, when the camera moves so that the image of the spot is expected to move from side to side, any up-and-down motion of the spot may indicate an incorrect swing angle. Likewise, during an expected up-and-down motion, any perceived side-to-side motion indicates a roll error. If a roll correction is desired, the amount of the correction is reported to the user, and used to update C_T.

Projector calibration is accomplished by shining the laser stripe on the tabletop and observing the result. The infrared filter is replaced on the camera, the laser is turned on, and the threshold is adjusted to observe the stripe, which is actually the intersection of the (unknown) stripe plane and the (known) tabletop. Because the camera position is known from the robot position and the current calibration information, and the equation of the tabletop is known, the actual three-dimensional coordinates of all points on the stripe may be calculated. These coordinates are transformed to be relative to the mounting cage. The robot is moved to a new height above the tabletop and the process repeated. After several views, the accumulated points in the coordinate system of the cage may be used for a least-squares determination of the relative stripe plane. Any discrepancy in orientation between the measured plane and the previous calibration data is reported to the user, and used to update **P**.

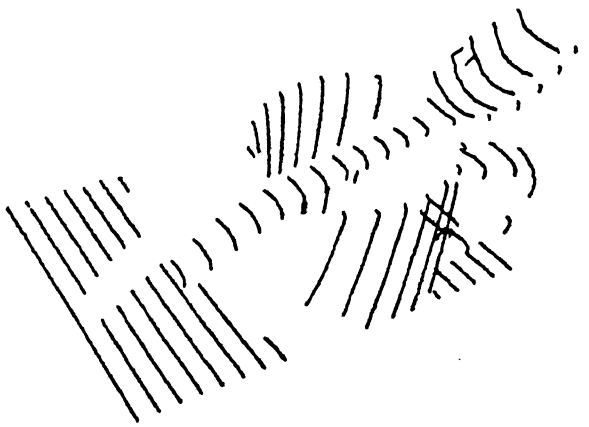

Figure 4-1: Perspective view of a cup and spoon

5. Some Results

Figure 4-1 is a perspective view of a spoon resting on top of a coffee cup. It was generated by stepping the range finder in 10 mm steps over the region of interest. The points from the centerline of the stripe were converted to three dimensions and displayed in perspective.

There are a variety of facilities for viewing three-dimensional data. Figure 4-1 was generated by a program for displaying three-dimensional vectors (i.e., wire-frame models). The user can rotate and translate the image and control the parameters of the virtual camera used to generate the display. Output may be displayed on a Tektronix 4006 terminal or on a laser printer using the POOF program developed by James Gosling and Tom Rodeheffer.

An Evans and Sutherland PS300 system provides another very useful display [9]. This system includes several dedicated processors, storage for vectors and programs, a display screen, and interactive devices for input and output. Points to be displayed are downloaded from our Vax through a serial line to the PS300. The system has a sophisticated programming language for specifying options for display. We have used this language to create an interactive display system where the user may rotate and translate the images in real time by using a set of dials. A special headset for stereo viewing gives a very lifelike image. A software cursor may be moved about the image for measuring coordinates and distances.

We use the PS300 to evaluate the accuracy and repeatability of the ranging system. There are several kinds of accuracies we might wish to evaluate. "Random error," which is mainly digitization noise, can be estimated by scanning a flat surface with several parallel scans of the ranger, and displaying the points on the PS300. We rotate the image on the display so that we are viewing the plane edge-on. This creates a band of vectors on the display, the thickness of which we can measure with the software cursor. The random error in our data appears to be on the order of ±1 mm.

"Directional-dependent error" arises from errors in calibration. If an object is scanned twice with the ranger approaching it from different directions, and we notice a shift in the apparent position of the object, the amount of that shift is position-dependent error. To evaluate direction-dependent error, we use a pyramid for a target because it is possible to locate the peak of the pyramid in the range image by using the cursor. Since there are six degrees of freedom of the manipulator, there are many ways we can vary the direction of approach, direction of scan, and distance from the object. The worst direction-dependent errors we have measured appear to be on the order of ±5 mm.

Scanning the scene of figure 4-1 required about 90 seconds of CPU time, or about 7 minutes of elapsed time. However, this should not be taken as indicative of the ultimate speed of a well-engineered program using this approach. Profiling the program that gathered the data indicates that over 90% of the execution time was spent in routines that communicate between the Vax and the LSI-11. Changing to a more efficient set of communication routines would radically reduce CPU time. Communication between the LSI-11 and the VAL-II controller is also awkward, because the LSI-11 emulates a typist at the VAL system terminal. VAL-II has only a one-character input buffer, so the LSI-11 waits for each character it sends to be echoed before sending the next character. VAL-II has an interface designed for computer-to-computer communication, but we do not yet have the DDCMP protocols needed for the LSI-11 side of the link.

Given a configuration that completely eliminated communication overhead, how fast could we expect the system to perform? There are three times to be considered: arm motion time, image preprocessing time, and coordinate conversion time. For our intended application—random-access scanning of tabletop scenes to recognize industrial parts—the arm motion time will dominate. This will be a function of how far the arm is to move between scans. If we scan in a regular raster pattern, as in figure 4-1 the robot should be able to make the move and settle down in a few hundred milliseconds. Image preprocessing time is also a few hundred milliseconds, and coordinate conversion time will be considerably less than that. When scanning a regular pattern, arm motion can be overlapped with processing the data from the previous image. Therefore it is reasonable to say that a system such as we are discussing should be able to image 3 or 4 stripes of 256 three-dimensional points per second if scanning is done in a regular raster.

References

1. G. J. Agin. "Real-Time Control of a Robot with a Mobile Camera." *9th International Symposium on Industrial Robots*, Society of Manufacturing Engineers, Washington, D. C., March 1979, pp. 233-246.

2. G. J. Agin & T. O. Binford. "Computer Description of Curved Objects." *IEEE Trans. Computers C-25*, 4 (April1976), 439-449.

3. Gerald J. Agin and Peter T. Highnam. "A Movable Light-Stripe Sensor for Obtaining Three-Dimensional Coordinate Measurements". *Proceedings of the SPIE International Technical Symposium*, San Diego, CA, August 21-27, 1982.

4. M.D. Altschuler et al. "The Numerical Stereo Camera." *Proceedings of SPIE (3-D Machine Perception)* 283, (April 1981), pp. 15-24.

5. M. E. Bair. "3D Sensor Development at ERIM." Workshop on Sensors and Algorithms for 3-D Machine Perception, Washington, DC, August 1983.

6. Dana H. Ballard and Christopher M. Brown. *Computer Vision.* Prentice Hall, 1982.

7. R.C. Bolles, J.H. Kremers and R.A. Cain. A Simple Sensor to gather Three-Dimensional Data. Technical Note 249, SRI International, Menlo Park, CA, July 17, 1981.

8. R. O. Duda & P. E. Hart. *Pattern Classification and Scene Analysis.* Wiley Interscience, 1973.

9. *PS 300 User's Manual.* Version P4.V01 edition, Evans & Sutherland Computer Corporation, P. O. Box 8700, Salt Lake City, Utah 84108, 1982.

10. D. Nitzan, A. E. Brain, and R. O. Duda. "Measurement and Use of Registered Reflectance and Range Data in Scene Analysis". *Proc. IEEE 65*, 2 (February 1977), 206-220. Also SRI AIC Tech Note 128, April, 1976..

11. R. J. Popplestone, et. al. "Forming Models of Plane-and-Cylinder Faceted Bodies from Light Stripes." *Fourth International Joint Conference on Artificial Intelligence*, Tbilisi, Georgia, USSR, August 1975, pp. 664-668.

12. C. Rosen et. al. Machine Intelligence Research Applied to Industrial Automation, Eighth Report. Grant APR75-13074, SRI Project 4391, SRI International, Menlo Park, California, August 1978.

13. Y. Shirai & M. Suwa. "Recognition of Polyhedrons with a Rangefinder." *Proc. Second International Joint Conference on Artificial Intelligence*, London, 1971, pp. 80-87.

14. G. J. Vanderbrug, J. S. Albus, and, E. Barkmeyer. "A Vision System for Real Time Control of Robots.". *Ninth International Symposium on Industrial Robots*, Washington, D. C., March 1979, pp. 213-232.

15. M. R. Ward et al. "CONSIGHT: A Practical Vision-Based Robot Guidance System." *Proc. 9th International Symposium on Industrial Robots*, Soc. Manufacturing Engineers, Washington, D. C., March 1979, pp. 195-212.

16. P. M. Will and K. S. Pennington. "Grid Coding: A Novel Technique for Image Processing". *Proceedings of the IEEE 60*, 6 (June 1972), 669-680.