An Integrated Vision Touch-Probe System for Dimensional Inspection Tasks

Marilyn Nashman, William Rippey, Tsai Hong Hong, Martin Herman
National Institute of Standards and Technology
Building 220/B124
Gaithersburg, MD 20899

Abstract

This paper discusses the integration of vision and touch sensors in a coordinate measuring machine (CMM) controller used for dimensional inspection tasks. A real-time hierarchical control system is presented in which a vision system extracts positions of features on a part to be inspected, and then guides a touch probe to efficiently measure these features. The probe is tracked by the vision system as it scans surfaces so that it's motion can be visually servoed. Minimalist sensor-derived representations, involving only task-specific information, are used in this process. Although the camera itself remains uncalibrated, a real-time calibration of very limited scope is performed each processing cycle to transform the task-specific image information into 3-D information used as feedback to guide the probe.

1. Introduction

A coordinate measuring machine (CMM) is a highly accurate 3 degree- of-freedom cartesian robot, often used for dimensional inspection of mechanical manufactured parts. Dimensional inspection involves measuring the relative geometry of surface features and determining whether they are within tolerance. Examples of geometry evaluated include shapes of smooth surfaces, distances between edges, positions of holes, and diameters and shapes of holes. For many applications, such as in the automobile or aircraft industries, measurement accuracies on the order of 25 micrometers (.001 inches) are required. Virtually all CMMs in use today use touch-trigger probes. When such a probe pushes against a surface with enough force to exceed a certain probe deflection, a signal is sent to the CMM controller to read the machine’s scales (i.e., the positions of all the machine axes). Collection of data using such probes is very slow, usually about one point per second. Such probes are not suited for gathering information along object edges since the precise location of edges is not well defined by models. Edge features however contain very important information for the measurement system. Also, these probes are not suited for gathering dense surface data, which is important for measurement of complex-geometry parts.

This paper describes advanced data acquisition methods being developed at NIST in the Next Generation Inspection System (NGIS) project. The main goals of this project (involving a consortium of companies organized by the National Center for Manufacturing Sciences) are to increase the speed and flexibility of data acquisition using CMMs while still maintaining today's accuracy. Our goal is to increase data acquisition by two orders of magnitude, increase the density of data acquisition, and make it easy to measure different parts with highly variable geometry. We are integrating advanced sensors (including a video camera, an analog touch probe, a point laser triangulation probe, and an analog capacitance probe) with an advanced control system.

In this paper, we focus on the video camera, the analog touch probe, and the control system for controlling probe motion, data acquisition, and interactions between the sensors. The analog touch probe differs from the touch-trigger probe in that it permits a continuous signal to be extracted, which represents the deflection of the probe tip as the probe is scanned over a surface. As the probe is scanned by the CMM arm, an accurate synchronization procedure synchronizes the probe data with the machine scales.

The camera is stationary relative to the part being measured and is positioned so that both the part and the probe are within its field of view. In order to make the system easy to use, camera calibrations should be minimal and/or capable of being performed very quickly by inexperienced personnel. This would allow shop floor personnel to easily place the camera wherever the viewpoint seems most appropriate.

Because of the low resolution of the camera, visual data are not accurate enough to use for actual part measurements. However vision can provide estimates of positions of features of interest on the part. Using real-time vision, the probe is guided to these features of interest, and probe measurements are then used for the actual inspection data. Probe guidance is accomplished by tracking it in the imagery as it moves. This allows parts to be measured even if an accurate a priori model is not available, and, in some cases, allows us to bypass the time-consuming step of part set-up, which uses fixturing or other methods to register the part with the model.

For example, when the probe is scanning over a surface, the motion of the probe is controlled by information from three sources, the camera, the machine scales, and the probe itself. Vision provides information about positions of part features (e.g., edges, holes, grooves, protruberances). The machine scales, when used in conjunction with vision, provide the distance of the probe from these features. The probe data provides the displacement of the probe from the part surface. We intend to use this to demonstrate the following capabilities:

(a) Smooth control of the scanning probe over the surface. In order to achieve maximal speed during surface scanning, we want to scan quickly over smooth portions of the surface but slowly over portions with edges. These edges can be detected visually to indicate when to speed up or slow down.

(b) Control of the probe while it is tracking an edge. Visual feedback will provide the edge contour so that it can be tracked by the probe.

In a non-scanning mode, vision will be used to locate features of interest on the part so that the probe can be moved directly to these features to obtain measurement data.

This paper describes in detail a necessary primitive capability for our system: controlling the speed of the scanning probe using visual feedback. As mentioned above, while the probe is scanning a surface, smooth portions of the surface may be scanned very quickly, but the probe will fly over the edge, without measuring all of it, unless it is scanned slowly. Our approach achieves real-time intelligent behavior by using minimalist sensor-derived representations. In such representations, a minimal amount of information required to achieve the given task is extracted from the sensors [4][5]. The representations contain only task-relevant information. In the example of reducing the probe's speed when it approaches an edge, the task-relevant information needed from the sensors is the current distance from the probe to the edge. This is a value in 4- D space-time (since it's value changes over time) and we demonstrate that it can be extracted using an uncalibrated camera.

In Section 2, we discuss the strengths and weaknesses of camera imaging sensors and touch probes. Section 3 discusses various approaches to sensor integration. Section 4 describes the NGIS hierarchical control system architecture. Section 5 describes the integrated probe-vision system algorithms. Section 6 describes the integrated vision-probe experiment, and Section 7 discusses future work.

2. Vision and Touch Sensors

In order to use the combination of a camera and a touch probe to its best advantage in an inspection task, we compare the strengths and weaknesses of each sensor. The most obvious characteristic of a camera is the fact that it is a non-contact sensor. The advantages of using visual information are speed and the global nature of the data. Although camera data are generally noisy, an entire scene can be read in 16 milliseconds. Depending on the image processing hardware and the complexity of the algorithms performed, simple features can be extracted very easily. The bandwidth for visual information is very high: a typical image can contain between 65,000 and

262,000 pixels depending on the camera resolution. Although a full resolution non-magnified camera image produces less accurate results than a touch probe, it can quickly locate and measure object edges, corners, and centroids. Among the very few manufacturers of dedicated coordinate measuring machine systems who have added vision capabilities to their CMM systems[18], the strength of obtaining global information is sacrificed for increased accuracy by using very high-magnification camera lenses on a limited field of view. In such systems, camera information is used only to reinforce probe data.

The problems associated with using camera data can be divided into two classes: geometric constraint problems and radiometric constraint problems [17]. Geometric constraints include visibility, field of view, depth of field, and pixel resolution. The radiometric constraints include illumination, specularity, dynamic range of the sensor, and contrast. Section 5 discusses our use of polarizing filters and polarized lighting to reduce specularity. We are not currently using an active vision system which could potentially overcome the geometric constraint problems.

A touch probe is a contact sensor. The information it extracts is of a local nature; the data apply only to the specific point touched. Since information is read one point at a time, data acquisition is very slow. The bandwidth of touch probe data is very low which makes it unsuitable for rapid high-density data acquisition.Touch probes are also quite crash-prone. Nevertheless, they are highly accurate measuring sensors, and there is very little noise associated with their data [3]. Touch probes are best suited for measuring simple geometric features. Within the constraints of their range of motion, they are not affected by viewpoint or lighting considerations.

3. Sensor Integration

When a single sensor is used to sense the environment, it output is often simple to interpret, but the user of such a system must rely completely on the accuracy and integrity of that data. Single sensor systems are limited in their ability to sense and identify meaningful features under varying conditions. A single source of information can only provide partial information about an environment, and that information is often insufficient to constrain possible interpretations and to resolve ambiguities [8]. The use of multiple sensors to perform a task overcomes the problems caused by relying on a single sensory input, but creates other problems concerning the interpretation and possible merging (fusion) of multiple sensory outputs. A great deal of research has been directed at ways of combining the information from a multiple sensory system. Most methods use measures of statistical uncertainty to model sensor readings. Measures of confidence in the individual sensor readings are updated based on the uncertainty measures [2][8][9][12].

Multiple sensory systems offer many advantages over single sensory systems. Their primary benefit stems from the use of diverse sensors which produce logically distinct outputs. The output from each individual sensor contains some level of uncertainty caused by noise in the system, difficulties in obtaining measurements, calibration errors or sensor degradation [6]. In a multi-sensory system, the diversity of information is used to overcome the limitations of the individual components [8][9].

The outputs from multiple sensors can be classified into three categories based on their interactions. These interactions are labelled competitive interactions, complementary interactions, and cooperative interactions. The features to be perceived are considered dimensions in a space of features [11] and can be either dependent or independent of each other. Competitive information interaction is defined as the interaction between sensors observing the same object in a scene which measure dependent features in the feature space. The information provided by these sensors is called redundant information since the sensed information is not adding additional knowledge to the system. The integration of competitive sensors results in either an increased confidence in the validity of the extracted information when the readings support each other, or, conversely, a lowered confidence indicating a possible sensor error when the readings disagree. The use of a touch probe and highly magnified camera information in a CMM provides an example of competitive information. Both sensors are designed to supply information about the three-dimensional position of an object feature.

Complementary information interaction occurs when two or more sensors supply information about the same object in a scene but each sensor observes features that are independent of the features observed by the other sensors. In such cases, each sensor provides partial information about the feature in the environment. A simple example of complementary sensor interaction is the integration of information returned by a thermometer measuring the temperature of an object and a range finder measuring the distance from the sensor to the object. The returned information from one sensor can neither strengthen nor weaken the information from another sensor in this configuration, but the combination of returned information provides the user a greater understanding of the sensed object.

Cooperative information interaction occurs when one sensor's observations guide the actions of another sensor. The information obtained from one modality directs the other sensor to obtain new information relative to a feature of interest. Allen extensively discusses the cooperative interaction of vision and touch in [2]. Figure 1 describes the operation of a cooperative information interaction. The first sensor in this system, S_1, processes its output, d_1 in process T_1. The output from this process is used to guide the actions of sensor S_2 which operates on its data, d_2, in process T_2 and produces output C_i. The dotted line from process T_1 to S_1 represents a closed feedback loop in which the processed output from the sensor S_1, is used to guide its own placement. The feedback loop does not have to be present in a cooperative system, but its presence adds to the sensors'

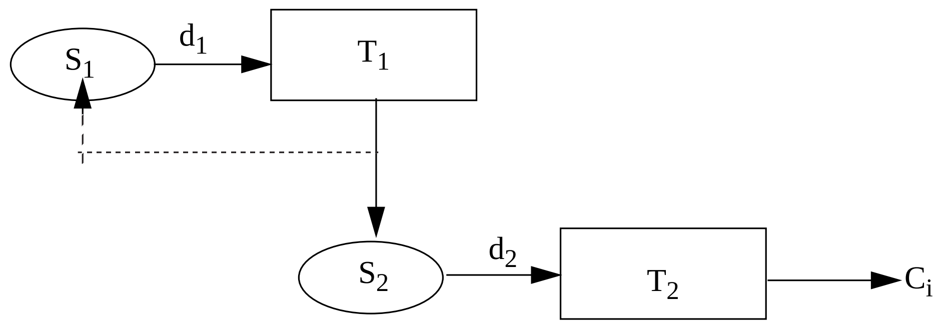

Figure 1 Cooperative Information Interaction

capabilities. The integrated system described in Section 5 takes advantages of cooperative interaction.The goal of this system is to combine the strength of vision systems, the ability to gather high bandwidth global information, with the strength of a touch probe, the ability to obtain highly accurate 3-D information.

4. Integrated System Architecture

The NGIS testbed is designed according to the architecture guidelines of the Real-time Control System (RCS) described in [1]. The architecture defines a hierarchy of controller nodes, each with an assigned set of responsibilities that include sensory processing (SP), world modelling (WM) and behavior generation (BG) (Figure 2). The Servo Level is the lowest level of control; it translates commanded positions into commands to motors and actuators. The next higher level, Prim, computes inertial dynamics and generates smooth trajectories. Its output to the Servo Level consists of evenly spaced trajectory points. The third level, E-move, transforms symbolic commands for "elemental" movements into strings of intermediate poses which define motion pathways that are free of collisions and kinematic singularities. The fourth level, Task, is the highest level implemented in our system. It transforms goals defined in terms of desired actions to be performed on objects into a series of E-moves designed to achieve these actions.

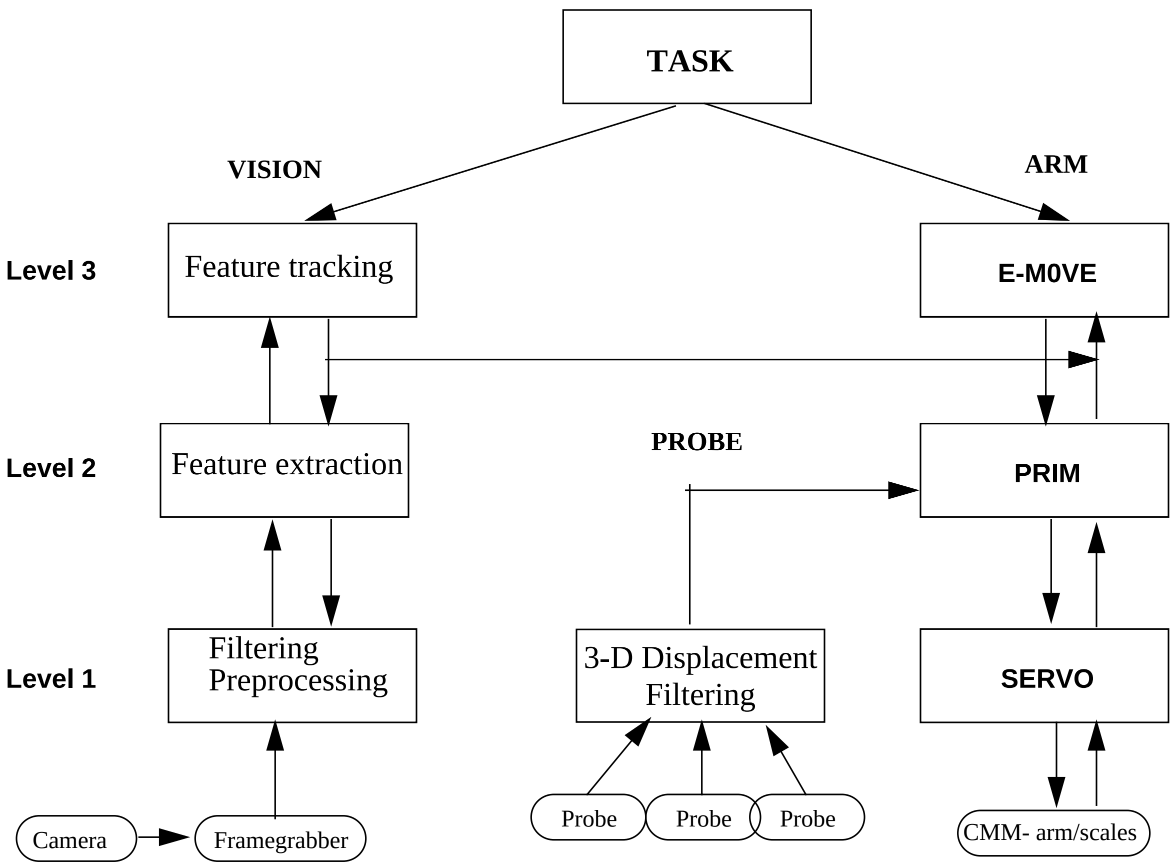

Figure 2 NGIS Control System Architecture

The role of the sensory processing system is to monitor and analyze information from multiple sources in order to recognize objects, detect events, and filter and integrate information. Sensory processing is also hierarchically decomposed into levels which define the scope of the operations at each level (Figure 2). Processing at the lowest level is limited to gathering raw information (readings) from each sensor, filtering the information, and when applicable, enhancing it. In a vision system, one or more cameras acts as the sensing agent. Level 1 Vision processing reads a digitized image frame from the camera and performs filtering and/or enhancement in order to improve the image quality [1]. The analog touch probe reads accurate three dimensional displacement of the probe tip in the probe coordinate system. The information extracted from a touch probe at Level 1 processing consists of continuous values of probe deflection. These readings are filtered before they are made available to the arm hierarchy. As shown in Figure 2, input to the Probe hierarchy can be read from multiple probes. In our application, a single analog touch probe supplies input.

At the next higher level, Level 2, the image output from Level 1 is analyzed in order to detect two dimensional image features such as edges, corners, and attributes of regions such as area, perimeters, centroids, and cavities (holes). If sufficient information exists, the two dimensional features are transformed into three dimensional coordinate space. In this way, features extracted from image processing may be expressed in a coordinate system common to that of other positional sensors.

The third level of the sensory processing system for the vision system is responsible for processing related to tracking features on moving objects and collision avoidance. In the arm hierarchy, the filtered displacement values computed by the probe system are integrated with the filtered arm scale readings using a priori calibration constants obtained for the machine and probe. This results in representation of the probe tip in the machine 3-D coordinate system. Figure 2 describes the functional architecture of the NGIS integrated system. Each rectangle represents a controller node in either the sensory processing hierarchy or the controller hierarchy at a specific level. Although it is not explicitly drawn, each node in Figure 2 consists of SP, WM, and BG modules as shown in Figure 3.

Figure 3 RCS Controller Node

5. Integrated Vision Touch-Probe System

The NGIS testbed contains a three degree of freedom CMM equipped with a touch probe and a

Figure 4 NGIS testbed

stationary camera mounted on the CMM surface. The left image in Figure 4 displays the probe attached to the CMM arm measuring the surface of a rectangular step object. The right image in Figure 4 shows the relationship of the camera mounted on the CMM table to the probe. The camera output is captured by a framegrabber which digitizes it to a 512 x 512 resolution. The CMM table moves in the Z direction (Figure 5), and the arm moves the probe in positive or negative XY directions. The controller positions the CMM table and the probe in 3-D coordinates relative to a

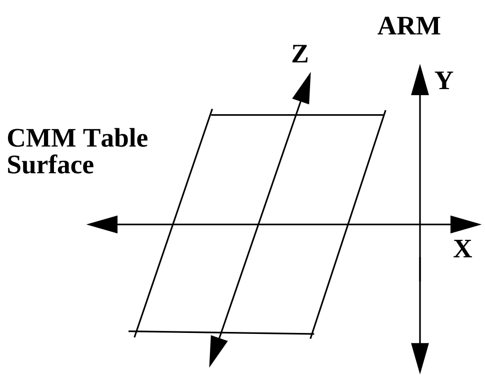

Figure 5 CMM Coordinate System

pre-defined origin in order to position the probe for measuring the object.

The camera currently used in our experiments is mounted on the CMM table, but it is stationary relative to the object on the table. Lighting conditions and specularity create problems for this application since many of the parts being measured are machine finished to a high gloss. The glare and reflections from overhead lighting introduce shadows and artifacts that interfere with the image processing algorithms. Any efforts to reduce specularity in this environment must be practical as well as effective. A CMM is used in a manufacturing environment; it would not be practical to install multiple lighting sources or special purpose lighting to improve the appearance of a computer image. We have introduced two relatively simple and inexpensive measures to reduce specularity; both make use of polarizers. Reflections are composed of diffuse and specular components; if the specular component could be removed, the remaining information would be easier to interpret [20]. We attempt to remove specular components using polarized light filters. By using two linear polarizing filters in front of the camera lens, oriented 90 degrees out of phase with each other, the imaging system can suppress scattered light rays which contribute to reflections. In addition, we reduce some of the light glare by placing sheets of polarizing filters directly over the florescent light fixtures in the laboratory.

The vision-probe system is used in a cooperative interaction integration mode in our experiments. Global information generated by the vision system is used to guide the movement of the touch probe across the surface of the step-block. Vision provides information about positions of part features of interest, e.g. edges, and, in conjunction with the machine scales, the distance of the probe from these features. The probe data provides the displacement from the part surface. Using this information, the controller commands the probe to scan quickly over smooth portions of surfaces and slowly over portions with edges.

Visually derived feedback in the control system consists of the distance (in millimeters) between the current position of the probe and the edge of interest. There are five steps used to generate the image information needed to guide the probe and alert the controller when the probe is approaching an object edge.

• Extract edge pixel locations and orientations.

• Fit edge pixels to line segments corresponding to the edges of the object.

• Extract probe position and direction of motion before probe motion is initiated.

• Filter computed probe position and predict the position of the probe in the next image.

• Track probe as it moves along the object's surface.

In order to compute the edges in the image, a Sobel edge extraction is performed on the full image [14]. The spatial derivatives, $\frac{\partial I}{\partial x}$ and $\frac{\partial I}{\partial y}$, are computed using 3 x 3 spatial gradient operators. The magnitude is thresholded at a pre-determined grey scale in order to preserve only edgels having strong grey scale support. A list of edge positions in image space and their corresponding orientations is generated.

A Hough transform [10] is used to fit the extracted edges into straight line segments. Figure 6, showing representative results for the step-block, displays a graph of the extracted edge points overlaid with the lines (shown as dotted lines) selected by the Hough transform.

Figure 6 Extracted Edge Pixels and Fitted Lines

Tracking of the probe in the video imagery occurs as follows. First the controller places the probe in a known location on the object surface. Then, in an off-line operation, the system operator determines the approximate location of the probe in image coordinates by simply examining the image. The operator's crude estimate in image coordinates is used to generate a small window in the image from which an accurate 2-D position of the probe can be computed. It is assumed that the window contains an image of the probe; if not, an error is generated. In order to segment the probe in the image, we take advantage of its distinguishing characteristic, its vertical orientation. The horizontal gradient is computed and dynamically thresholded. The use of dynamic thresholds eliminates the problems caused by inconsistent lighting, shadows, poor contrast, and reflection. We choose the bottom-most pixel above threshold as the 2-D location of the probe in Figure 7. Figure 7a shows an enlarged view of the image window containing the touch probe based on the operator's estimate of the initial location. The original image quality is poor because of low contrast in the scene and the shadow cast by the probe on the CMM table (lower left corner). Figure

(a) (b) (c)

Figure 7 (a) raw image of probe
(b) gradient image of probe (c) thresholded gradient image

9b shows the grey-scale results of applying a 3x3 gradient mask to extract the horizontal gradient. Figure 7c shows the results of applying a dynamic threshold to the gradient image based on a cumulative histogram. The probe location is computed to be the coordinates of the lowest above threshold pixel in the window. A 10 x 10 grey scale template centered at the probe position is generated for the correlation tracking algorithm described below.

Assuming that the probe's motion while scanning along the surface is a straight line path, we next compute a vector in the image plane in the direction of motion using the initial probe position and a second image position computed either as the probe moves or entered by the operator. The intersection in the image plane of the motion vector and the nearest object edge represents the probe's 2-D goal point. The distance between the initial position and the goal position is measured in pixels.

We track the probe as the arm moves towards its goal position using the predicted probe image velocity and sum of absolute differences (SAD) correlation algorithm. A predictive filter is used to filter and predict the probe position and velocity at the next time interval [7][15][19]. At each processing iteration, the search direction and window used for the SAD correlation is updated based on the predicted probe image position and velocity. In Figure 8, P is the computed 2-D probe

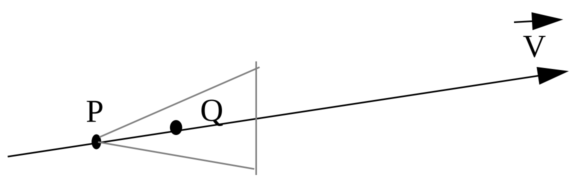

Figure 8 SAD Search Window

position, Q is the predicted position, and V is the vector representing the probe motion and direction. The search window for the next SAD iteration is shown by the grey triangle. The probe position is computed to be the position which yields a minimum value for the sum of absolute differences over the search space. The correlation template is updated each cycle to reflect the current grey scale information representing the probe position.

Feedback to the arm system consists of cyclical updates of the 3-D distance between the probe and the nearest edge object. In order to provide this information, we use the following relationships. Let P_{2D} be the probe position in the 2-D image plane at time t_0 and let V_{2D} be the velocity in the image plane measured between time t_0 and time t_{-1}. Both of these parameters can be measured, since the probe's positions in the image have been extracted. At each processing cycle, we recompute the image goal position by intersecting the motion vector generated by P_{2D} and V_{2D} with the object edge. (This works best for straight-line motion of the probe.) Using the updated 2D goal point and the current probe position, we compute D_{2D} to be the distance between the two points. Assuming constant velocity, the time required to reach the goal point, in both 2-D and 3-D coordinate systems, is $T = D_{2D} / V_{2D}$. In 3-D space, the probe position and velocity, P_{3D} and V_{3D}, are computed by the Arm e-move level and made available to the vision system through a common memory communication process. The 3-D distance to the edge can therefore be computed each cycle as $D_{3D} = V_{3D} * T$. This distance is computed in millimeters every processing cycle. In many situations, the probe motion will be along a curved, rather than straight, path, and certainly the probe velocity is not constant. However, since the image goal point and the time to reach the goal are updated every processing cycle, we expect that as the probe approaches the goal, the estimated distance to the goal will become more and more accurate. Notice that, in effect, we are re-calibrating this distance in real-time every processing cycle. This distance is a task-specific value, as it is used to determine the speed to scan the probe. As this distance becomes smaller, the probe must be slowed down. Since no other 3-D information about the environment need be known, the camera itself can remain uncalibrated.

6. Experiments and Results

The equipment used in the NGIS testbed laboratory includes a three degree of freedom coordinate measuring machine as described in Section 5, multiple interchangeable probes, a 16 mm black and white CCD camera, a framegrabber, Sun SPARC4 workstations, and a VME-based multiprocessor system running a VxWorks operating system. Code is written in C++ on the workstations, and downloaded to the VxWorks system.[1] The purpose of the experiment described here is to demonstrate our real-time control system in which computer vision is used to determine the distance from the probe to a goal edge as the probe scans over a planar surface. This distance is used to control the speed of the probe as it nears the edge. Although the testbed is equipped with many probes, this experiment uses only a single touch probe.

The experiment is initiated by the controller task level (Task) shown in Figure 2. Commands are sent to the e-move level (E-move) to move the arm and to the vision Level 3 process via a common-memory interface. Table 1 describes the commands generated by Task which are

Table 1: Demonstration Scenario

Acting Level	Commanded Action
Arm E-move	Move arm near object
Vision Level 3	Extract and define edges of object in field of view
Arm E-move	Place probe at pre-defined starting position on object surface
Vision Level 3	Extract position of probe on object surface
Vision Level 3	Determine 2-D goal point of expected probe trajectory
Arm E-move	Scan surface until proximity to goal detected
Vision Level 3	Loop and track probe; report proximity of edge
Vision Level 3	Terminate tracking

sent to either Arm E-move or Vision Level 3 processes. Note that the command to extract and define edges of the object in the field of view occurs just once for a particular field of view. This means that the edge extraction and Hough transform processes, which are time consuming, are run only once. The probe tracking process, on the other hand, is updated every processing cycle. Both

1. Certain commercial equipment, instruments, or materials are identified in this paper in order to adequately specify the experimental procedure. Such identification does not imply recommendation or endorsement by NIST, nor does it imply that the materials or equipment identified are necessarily best for the purpose.

the arm control and vision processes decompose their commands for the levels hierarchically below them. Status information is always sent to the requestor process indicating either completion of the command or an error condition. Visual tracking is terminated when the closest edge is detected as being less than a pre-defined δ millimeters away from the current probe position.

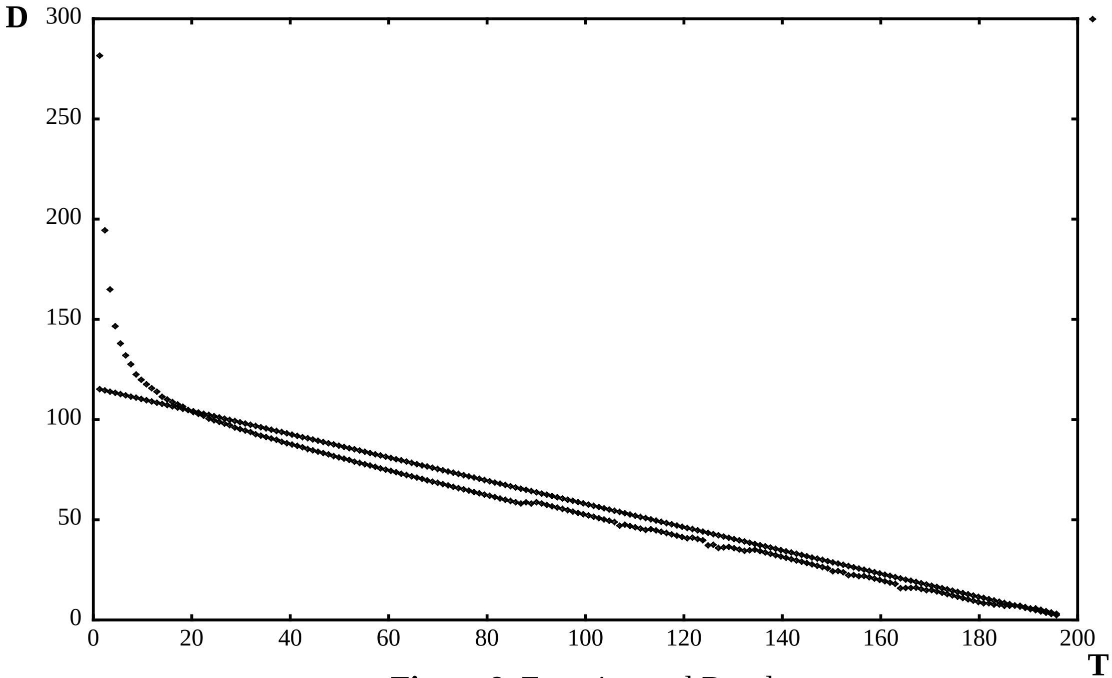

Figure 9 Experimental Results

Figure 9 describes the results of an early experiment. The y axis represents distance in mm of the probe position from the object edge; the x-axis represents time units. The straight line represents the 3-D distance between the probe and the goal point as measured by the arm. This distance is plotted for each processing cycle. The curve represents the visually derived real-time estimate of the 3-D distance remaining to the object edge. This information is computed during run time. The endpoints of the curves near t = 200 differ by only 0.64 mm; the vision system reports a “distance-to-go” of 0.64 mm when the arm controller has reached the goal point. Early measurements in the run are not robust as can be seen by the spurious data points below t = 20. Here distances used to estimate velocities are very small, and inexact registration between 2-D and 3-D information is exaggerated. 3-D estimates taken between t= 20 and t = 200 represent meaningful information. As distances increase, the registration error effects are reduced. Visual feedback in the controller can be used to vary arm velocity based on distances to features of interest.

Visual processing is performed on the multiprocessor system; other than the framegrabber, no special image processing hardware is used. This results in longer than expected processing time for the more computationally intensive operations such as convolutions, Hough transform, and SAD algorithm. The time critical component for real-time visual processing is the SAD tracking algorithm; edge extraction and line fitting are performed only once for each run. A tracking iteration is performed every 210 milliseconds, or at a rate of a little less than 5 Hz. This time is decomposed as shown in Table 2. We expect that the tracking related computations will be reduced when existing faster microprocessor boards are installed.

Table 2: Image Processing Timing

Type of Processing	Time (in 10 millisecond units)
Digitize and store image frame	5
SAD correlation	15
Filter and predict next probe location	1

7. Future Work

Our initial experiments were performed on a simple geometric object (Figure 4). We will perform similar experiments on complex parts. We are also implementing additional image processing capabilities in order to detect features such as curved edges, holes, and grooves on complex parts. We are investigating fast, easy o perform, calibration methods in order to register image space with CMM space; this will give us the ability to visually servo the arm to features of interest and to inspect and follow either linear or curved contours of interest. Our ultimate goal is to develop sensor-servoed scanning control algorithms that can be transferred to manufacturing plants.

References

[1] J. Albus, “Outline for a Theory of Intelligence”, IEEE Transactions on Systems, Man and Cybernetics, 21(3):473-509, 1991.

[2] P. K. Allen, Robotic Object Recognition Using Vision and Touch, Kluwer Academic Publishing, Boston, MA., 1987.

[3] P. K. Allen, R. Bajcsy, “Object Recognition Using Vision and Touch”, Proceedings of the 9th International Joint Conference on Artificial Intelligence, Los Angeles, CA, August, 1984, pp 1131-1137.

[4] J. Aloimonos, I. Weiss, a. Bandyopadhyay, “Active Vision”, International Journal of Computer Vision 1:333-356, 1988.

[5] D. Ballard, C.Brown, “Principles of Animate Vision”, CVGIP: Image Understanding, 56(1):3-21, 1992.

[6] S. S. Blackman, “Theoretical Approaches to Data Association and Fusion”, SPIE Volume 931 Sensor Fusion, 1988, pp 50-55.

[7] Coombs, D., Herman, M., Hong, T., Nashman, M., “Real-Time Obstacle Avoidance Using Central Flow Divergence and Peripheral Flow”, ICCV, June 1995.

[8] H. F. Durrant-Whyte, “Sensor Models and Multisensor Integration”, International Journal of Robotics Research, Vol 7 #6, 1988, pp 97-113.

[9] G. D. Hager, Task-Directed Sensor Fusion and Planning, Kluwer Academic Publishers, Boston, MA, 1990.

[10] R. Haralick, L. G. Shapiro, Computer and Robot Vision, Addison-Wesley Publishing Co., 1992.

[11] R. C. Luo, M. G. Kay, “Multisensor Integration and Fusion in Intelligent Systems,” IEEE Transactions on Systems, Man, and Cybernetics, Vol 19, No 5, September/October 1989, pp 901-931.

[12] R. C. Luo, M. Lin, R. S. Scherp, “The Issues and Approaches of a Robot Multi-Sensor Integration”, Proceedings IEEE International Conference on Robotics and Automation, Raleigh, NC, Volume 3, April 1987, pp 1941-1946.

[13] Nashman, M., “The Use of Vision and Touch Sensors for Dimensional Inspection Tasks”, Manufacturing Review, Vol. 6, Number 2, June 1993, pp 155-162.

[14] A. Rosenfeld and A. Kak, Digital Picture Processing, Vol. 1, Second Edition, Academic Press, 1982.

[15] H. Schneiderman, M. Nashman, “A Discriminating Feature Tracker for Vision Based Autonomous Driving”, IEEE Transactions on Robotics and Automation, Vol. 10 #6, 1994.

[16] M. J. Swain, M. Stricker, Eds., “Promising Directions in Active Vision”, NSF Active Vision Workshop, Chicago, Ill., 1991.

[17] K. Tarabanis, R. Y. Tsai, “Sensor Planning for Robotic Vision: A Review”, Robotics Review II, Ed. O. Khatib, J. Craig, Lozano-Perez, MIT Press, 1992, pp 113-136.

[18] VIEW Precis 3000 Manual, VIEW Engineering Inc., 1989.

[19] A. J. Wavering, R. Lumia, “Predictive Visual Tracking”, Proceedings of SPIE Intelligent Robots and Computer Vision 7, Boston, MA,1993.

[20] L. B. Wolff, “Using Polarization To Separate Reflection Components”, CVPR, 1989, pp 363-369.