

## Development of a Single-Axis Edge Detection System

Federal Manufacturing & Technologies

R. A. Hanshaw

KCP-613-6290

Published February 2000

Final Report

Approved for public release; distribution is unlimited.



Prepared Under Contract Number DE-ACO4-76-DP00613 for the  
United States Department of Energy

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade names, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

All data prepared, analyzed and presented has been developed in a specific context of work and was prepared for internal evaluation and use pursuant to that work authorized under the referenced contract. Reference herein to any specific commercial product, process or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof or Honeywell International Inc.

Printed in the United States of America.

This report has been reproduced from the best available copy.

Available to DOE and DOE contractors from the Office of Scientific and Technical Information, P. O. Box 62, Oak Ridge, Tennessee 37831; prices available from (865) 576-8401, FTS 626-8401.

Available to the public from the National Technical Information Service, U. S. Department of Commerce, 5285 Port Royal Rd., Springfield, Virginia 22161, (703) 487-4650.

Honeywell

Federal Manufacturing  
& Technologies

P. O. Box 419159

Kansas City, Missouri

64141-6159

A prime contractor with the United States

Department of Energy under Contract Number

DE-ACO4-76-DP00613.

## DEVELOPMENT OF A SINGLE-AXIS EDGE DETECTION SYSTEM

R. A. Hanshaw

Published February 2000

Final Report

R. A. Hanshaw, Project Leader

### Contents

Section

Abstract

Summary

Discussion

Scope and Purpose

Activity

Background

Vision System

Motor Control

Software

Measurement and Design Considerations

Accomplishments

Future Work

References

Appendices

A. Software Routines

B. Data

## Illustrations

Figure

1 SIP Trioptic CMM With Laser-Based Edge Detector

2 Camera Mounting Illustration

3 Trioptic With New Vision System

4 Graphical User Interface (GUI)

5 Computer Screen During Software Operation

## Tables

Number

1 Command Listing and Descriptions

2 Trioptic Measurement Results vs. NIST Measurements Results

## Abstract

A SIP (Societe Genevoise d'Instruments de Physique) Trioptic coordinate measuring machine was modified for calibration of high quality single-axis glass standards to an uncertainty of  $\pm 0.000020$  inch. The modification was accomplished through the addition of a frame grabber board, vision software, a high-resolution camera, stepper motors, a two-axis motor controller, and an HP-IB interface card. An existing temperature system (hygrometer, barometer, laser interferometer system, and optics) was retained as part of the system. An existing Hewlett Packard computer was replaced with a personal computer to accommodate the frame grabber board. Each component was integrated into the existing system using Visual Basic. The system was automated for unattended measurements by creating a machine programming language, which is recognized within the main program.

## Summary

This project was a result of concerns over the capability to calibrate high quality single-axis glass standards on a SIP (Societe Genevoise d'Instruments de Physique) Trioptic coordinate measuring machine (CMM). The laser edge detection system on the CMM was becoming increasingly erratic. The time required to set up the device had increased up to two days, and there was concern that the device would stop operating completely in the near future. The manufacturer was no longer in business; therefore, the device could not be repaired. The system was also manually operated with temperature effects being a concern.

The original computer, a Hewlett Packard 300 Series, was replaced by a personal computer running under the Windows NT operating system. The replacement was necessary in order to use a half-slot PCI bus frame grabber board. The frame grabber board is connected to a 2/3" black and white CCD progressive scan camera with an array of 1024 X 1024 square pixels. Integration of the frame grabber board and camera into a motion/vision system was accomplished with general machine vision software running under a custom operator interface created using Microsoft Visual Basic.

The SIP Trioptic CMM had originally been fitted with DC motors on the X and Y axes. In order to automate the inspection of glass scales, the DC motors were replaced with stepper motors. The selection of stepper motors was due to the availability of a two-axis controller, which was set up for stepper motors. The DC motors were removed and a fixture designed in order to place the stepper motors into the same positions on the CMM. The controller was interfaced to the PC through the serial port.

Most of the effort on this project was expended on the software needed to collect data from each device connected to the computer. Visual Basic was selected in order to minimize the time needed to learn the programming and its ability to quickly create operator interfaces.

Incorporation of the hardware and software into a measuring system capable of automated inspection of glass scales to an uncertainty of  $\pm 0.000020$  inch for lengths of two inches and less was accomplished in less than a year. Unattended inspection of glass scales has resulted in time savings of greater than 50 percent.

## Discussion

### Scope and Purpose

This project was initiated as an Advanced Design and Production Technologies (ADAPT) project. The specific objectives of the ADAPT program are (1) to maintain competence in key production processes; (2) to develop and demonstrate production capability for components, subassemblies, and assemblies acceptable for weapons use and which have application to enhance the safety, security, or reliability of the enduring stockpile; (3) to maintain viable design, development, and production interrelationships within the Nuclear Weapons Complex (NWC); (4) improvement of current production processes; and (5) development of processes for future production and ensuring they are production capable, including supporting feasibility demonstration studies. The purpose of this project was to maintain and improve the capability to calibrate small single-axis glass standards.

The scope of the project was limited to a SIP (Societe Genevoise d'Instruments de Physique) Trioptic coordinate measuring machine (CMM) where small single-axis glass standards (less than 14 inches in length) at Federal Manufacturing & Technologies (FM&T) are calibrated. This project is also limited to single-axis measurements since the machine's Y and Z axes have excessive hysteresis and backlash.

### Activity

### Background

The SIP Trioptic is a coordinate measuring machine constructed of cast iron. The machine was purchased with steel scales, which were later replaced with a three-axis laser interferometer system. The first device used with the Trioptic for calibration of glass scales was a photoelectric microscope. Using the photoelectric microscope, calibration of a stage micrometer (one-inch glass scale) required approximately 10 hours of manual measurements. This system was replaced in 1990 with a laser-based edge detector (see Figure 1). The laser edge detector sent a laser beam through the optics onto the glass scale. The beam was then reflected back from highly reflective lines onto a detector cluster. The detectors sensed a change in the reflectivity of the object and sent a trigger pulse. The trigger pulse was used to latch a reading on a single-axis laser, which was mounted on a tripod beside the machine. The detection of the edges was accomplished very quickly with a continuous movement of the table. The data was collected and stored on a Hewlett Packard 300 Series computer. The system could repeatedly detect an edge within five microinches.<sup>1</sup>

The laser edge detector had a number of weaknesses. The laser/detector system was a large bulky piece of equipment that had to be mounted approximately five inches in front of the normal measurement device location. This limited the amount of Y-axis travel. Since a single-axis laser was used as the measurement scale, it could be placed in line with the scale; therefore,



Figure 1. SIP Trioptic CMM With Laser-Based Edge Detector

the abbe' error was still minimized. The single-axis laser was, however, a drawback since it took additional space next to the machine and could easily be bumped out of alignment. Another problem involved a turning mirror, which was used to direct the beam onto the detector cluster. This mirror required very precise adjustment before the system could adequately detect edges. When this system was first installed, an adjustment mechanism had been designed that made this process much easier. However, over time the system became so difficult to adjust that it took two to three days to adjust the system to the point that edges could be detected correctly. It was felt that a repair was needed to bring the system back to its original condition but, since the manufacturer was no longer in business and it was such a specialized device, a repair to the system could not be made. There was also excessive heat transfer from the operator to the glass standard since the operator had to be present to operate the motors on the CMM. As the measurements progressed, the temperature could be seen to rise by as much as  $0.3^{\circ}$  C. Although this created only a small amount of error for scales less than an inch in length, there were a number of scales 12 inches in length that needed to be calibrated on the system. Lastly, at high magnifications the laser spot size was very small. With the objective used for calibration of stage micrometers, the spot size was  $7.5\mu$  m. This resulted in a very small sampling area of the line.

### Vision System

The laser-based edge detector was replaced with an 'iMAGING Technology' frame grabber board, 'Sherlock<sup>32</sup>' machine vision software and a Hitachi 2/3-inch black and white CCD progressive scan

camera. In order to use a frame grabber board, the Hewlett Packard 300 Series computer was replaced with a PC running under the Windows NT operating system.

The frame grabber board is a half-slot PCI bus digital image capture card with 4 MB of high speed, dual-ported, linear mapped video memory.<sup>2</sup> This particular board was selected because of its speed (images moved to PC in less than four msec), its on-board frame buffer memory, camera compatibility, and cost.

The machine vision software was considered the most important aspect of the vision system for a number of reasons. The system needed to be running within a short amount of time. This meant that the machine vision software had to be very easy to learn and implement. Example routines as well as software support were considered important. After reviewing various packages and using some demonstration samples, it was felt that the 'Sherlock<sup>32</sup>' would be the easiest to program, especially for the application for which it would be used. Another important consideration was the sub-pixel accuracy. The capability of the software to resolve beyond the individual pixel was necessary in order to increase the resolution beyond what could be accomplished with magnification and camera resolution.

Camera selection was based on resolving power, pixel type, speed, and cost. There were a number of cameras with similar features. The Hitachi KP-F100 was selected due to its compatibility with the frame grabber board and the cost. The camera is a monochrome high-resolution progressive scan camera with a digital output and an array of 1.3 million square pixels. The camera that was attached to the Laser Edge Detector was a 400,000-pixel array. The increase in resolution was approximately 3 times.

The optics system, which had been used with the Laser Edge Detector, was retained. The Nikon microscope contains an objective lens turret with objective lenses ranging from 10X to 200X. The most commonly used objective lens of 40X resulted in a magnification of approximately 1300X on the 12-inch monitor. The optics had been offset approximately five inches from the intended location of a measurement probe in order to accommodate the Laser Edge Detector. The new system was being set up to use the three-axis laser system; therefore, the optics needed to be moved back into the intended position in the Z-axis carriage. With the optics moved to the correct position, glass scale measurements could take place in line with the laser beam. This would remove the effects of abbe' error and result in a much better uncertainty, especially for the longer glass scales. The original probing system of the Trioptic was designed to mount into a 2.5-inch diameter hole in the Z-axis carriage. A knob on the left side of the Z-axis carriage activated the probe system clamping mechanism. A mounting fixture for the optics was designed to be mounted into the hole in the carriage. The original clamping mechanism was utilized as well as an additional clamp on the top of the Z carriage. An L bracket was also installed on the Z-axis carriage to lend greater support to the side of the optics (see Figure 2). With the optics installed in the Z-axis mounting hole, the entire Y-axis range was again usable.



Figure 2. Camera Mounting Illustration

The Hewlett Packard 300 Series computer was replaced with a PC with a system speed of 166 MHz running under Windows NT. The selection of NT was necessary for compatibility with the machine vision software. The new system is shown in Figure 3.



Figure 3. Trioptic With New Vision System

### Motor Control

In order to utilize the potential of the vision system, the system had to be automated. The two-axis controller that was available could not control the DC motors on the machine. In order to make use of an existing Unidex 11 controller, two stepper motors were needed. The motors had to fit within the same cavities where the DC motors were located. A size 34 stepper motor fit that requirement. The torque for the motor was 370 oz-in. at 400 rpm. The torque drops down to 120 oz-in. at 2400 rpm. This was expected to be adequate for the Trioptic. A simple fixture was designed and fabricated to attach the motor flange to the existing mounting holes. The motor did fit into the existing cavity with a small modification to the motor housing for the cable to pass through. The controller was connected to the computer's serial port. Before further tests could be run, the software had to be developed to display the X and Y coordinates.

### Software

The majority of the work on this project centered on the development of software. None of the original software could be used on the new system since it had been developed around a single-axis laser and the Laser Edge Detector. The new software would have to integrate all of the individual components into a single system that would operate without the presence of the operator. The system components consisted of a three-axis laser interferometer system, a barometer, hygrometer, temperature system, frame grabber

board, and a motor controller. The frame grabber board had its own software, which would have to be given commands as well as the motor controller.

Three types of software were being used in the department at the time: Visual Basic, Microsoft Fortran, and HT Basic. The Visual Basic was selected for its ease of use, short learning time, and capability to create user interfaces quickly. Although software routines had already been written in Fortran for data collection from a three-axis laser interferometer system, it was decided that it would be faster in the long run to convert the routines to Visual Basic.

Initially the requirements for the user interface were as follows:

- Data collection for temperature, humidity, and barometric pressure with one mouse click on a control. The data shall be displayed within an area set up on the control for environmental factors.
- Within the environmental factor area, a text box shall be displayed for setting the material thermal coefficient of expansion.
- A control shall be included to switch between inch and metric results.
- Coordinate data shall not be displayed until environment data is collected and laser compensation applied.
- A control shall be set up to allow the operator full access to the machine vision software in order to set thresholds, the window for data collection, and various other parameters present in the machine vision software.
- A control shall be present on the user interface, which will allow the operator to select from various files in order to load the appropriate settings for threshold and windowing.
- Controls shall be placed in the interface to control the positioning and speed of the motors.
- A text dialog area shall be present to display errors or help information.
- A control shall be placed on the user interface to move the X and Y axes a user-specified distance from the present position. Positioning must be within  $\pm 0.000050$  inch of the desired position.
- A control shall be placed on the interface to move the X and Y axes a user-specified distance from the zero point. Positioning must be within  $\pm 0.000050$  inch of the desired position.
- Measurement in the X-axis direction shall occur with a right mouse click.
- The user interface shall be positioned outside of the video window which is displaying the object.

As the software matured, further controls and conveniences were added in order to automate the system and to make it easier to operate.

The first step in creating the software centered on the development of routines to collect the temperature, humidity, and barometric pressure. This data, along with the thermal coefficient of expansion, was used to determine the laser compensation. Data collection from the laser interferometer system was then interpreted into Visual Basic from Fortran routines that had been developed on another CMM. Two text areas were set up on the user interface for display of the X and Y coordinates with a control button next to each coordinate text location. The control buttons were set up to zero the selected axis at the current position. The interface to the general machine vision software was taken from example software, which was included with the software. The machine vision software could then be accessed by easily understood commands, which were demonstrated in the example routines.

The next step was to create some simple controls for rough positioning of the CMM. At this point, the motor controller was not capable of simultaneous software and joystick control. Although the joystick was attached and present, it could not be activated within the software without an optional ROM chip. The control that was developed consisted of a left X-axis movement button, a right X-axis movement, Y-axis movement buttons in both directions, a stop button, and a speed adjustment slider. The controls

were found to work as well as could be expected, but fine positioning was very difficult. The optional ROM chip was later purchased to add the joystick functionality within the software. Addition of the joystick made manual fine positioning much easier. It was discovered at this point that the motors did not have quite enough torque at higher speeds. The speed of the motors was limited in order to eliminate the problem.

Up to this point the software development had proceeded rapidly with very few problems; however, the next step was to move the machine to within  $\pm 0.000050$  inch of the desired position. This presented much greater difficulty than had been expected. Much of this was due to the age and construction of the Trioptic as well as some difficulty with the motors. Movement of each machine stage is accomplished by sliding on a film of oil. There are no bearings on which to slide. The stages move easily but there is some friction involved. When the stage is stopped, a small amount of force remains unless the motor is reversed slightly. This small amount of force reveals itself through a slow drift of 50-75 microinches.

The Trioptic also has excessive hysteresis. When reversing the direction of travel, the motors rotate approximately  $\frac{3}{4}$  of a turn before any motion occurs. The controller resolution was known to be 2000 steps/revolution, and the specific distance per step was found by checking the distance traveled after movement of a specific number of steps. However, it was also determined that the steps per inch were not as constant as expected either. This could be caused by some slippage during acceleration. This was further complicated after viewing the effect of reversing the Y-axis direction. This revealed that there was an excessive amount of backlash present in the Y axis. A movement algorithm had to be defined to take into account the hysteresis, drift, and backlash. The algorithm was developed upon consideration of the method used in manual operation of the machine. During manual operation, the operator always approaches the measurement from the same direction. Upon nearing the measurement location, the operator slows the movement down and very gradually approaches until within approximately 20 to 40 microinches. At this point, the pressure is held constant and the machine stage is allowed to drift into the correct position at which point only a very small rotation in the opposite direction removes the excess force. The movement algorithm was then set up to always move approximately 0.0015 inch in the negative direction from the desired position (slightly more than the hysteresis present in the system), then to move close to the desired position at which point very small increments are made until the position is reached. At this point, a slight reversal in direction removes the drift from the system. Using this procedure, the Y axis repeated accurately enough to be considered for future use in measurements. Before this, the idea of using the Y axis had been discarded due to the sloppiness of the movement.

Now that the motion was acceptable, a routine had to be established to calibrate the vision system. This involved finding the length per pixel. A routine was created that measured the distance of a vertical line from an initial point to approximately 20 different locations within the field of view. The average distance per pixel was output to a file named 'pixelcal.dat'. The same process was repeated for the Y axis with a horizontal line and the average distance per pixel output to the same file. The distance was approximately six microinches per pixel at 1300X magnification (on a 12" monitor). A repeatability test was conducted in order to determine the ability of the system to repeat zero anywhere within the field of view. A vertical line was moved to a number of different locations within the field of view, and X-axis measurements were taken at each location (see Appendix B for results). The standard deviation of the X-axis readings was 0.000003 inch. In use, measurements are confined to a much narrower window of approximately 1.5 to 2 times the width of the largest line. The repeatability is much better within the window sizes commonly used.

With the addition of an X-axis measurement routine, the software was ready to perform some basic tests to establish additional needs. A stage micrometer was placed on the machine and aligned. The machine was positioned with the software and measurements were taken at each line on the scale. As

measurements were taken on lines spaced 0.02 millimeter apart, it became obvious that the initial line would need to be centered in the measurement window and that the measurement window would need to be as narrow as possible. Instead of creating a separate routine to center a line within the measurement window, the routine that is used to establish the X-axis zero point was modified to center the line within the window. When the X-axis zero occurs twice, this signals the software to center the line within the window. With the centering, the system appeared to be adequate to measure any type of high quality glass stage. The next step was to automate the system.

The system needed to be capable of measuring a variety of glass scales without the presence of the operator. In order to accomplish the necessary automation of the measurements, some type of command language was required along with an interpreter routine. The commands needed to be simple, familiar, and easy to understand in order to allow off-line programming. The operator should also be able to create the machine program without knowing the commands as in a 'learn' mode. It was decided that each control button on the graphical user interface (GUI) would contain the command. A control button was added that would place the system in a 'learn' mode. A left mouse button click on the 'learn' button would cause an operator prompt for the program name. At this point, the 'learn' mode is active and activation of a control button would cause a line of text with the correct command to be output to the machine program file. An incorrect line of code is erased by a control button labeled 'DeLn' (see Figure 4). Once the 'learn' mode is active, a press of the same control button stops program creation. Table 1

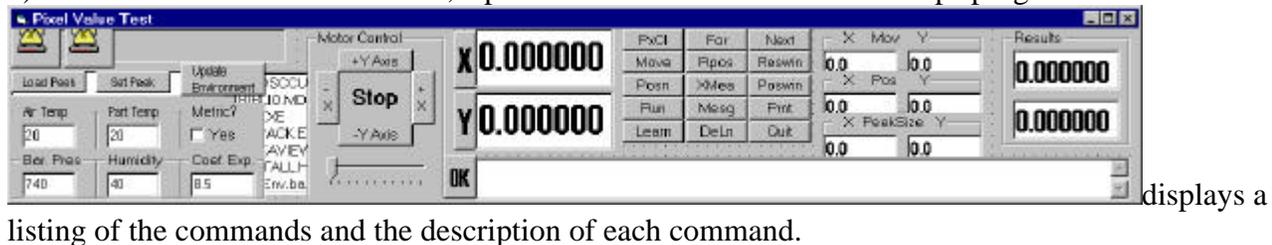


Figure 4. Graphical User Interface (GUI)

The GUI was designed to display only relevant controls and textboxes. For example the 'X Mov Y', 'X Pos Y', and 'X Peek Size Y' text areas are not shown unless the 'Move', 'Posn', or 'Reswin' buttons are activated. The results text area (upper right hand corner) displays the results of measurements, while the text box on the bottom is used for any type of prompts or messages. As the mouse pointer moves over a control button, a description of the button appears in a gray text box in the upper left-hand corner. The two controls with telephones pictured on them are invisible to the operator at all times and are for serial port communications. The coordinate data from the laser system is displayed in two text areas in the upper center of the form. The X and Y zero buttons are to the immediate left of these text areas. Motor controls are located to the left of the X and Y zero buttons, and the environmental data are displayed in the text boxes to the left of these. Activation of either the 'Load Peek' or 'Run' control buttons causes the form to change size and display only the drive, directory, and file text areas for selection of the proper file. The 'Run' button, when activated, allows the operator to select a machine program and begins program execution. A small sample machine program is shown in Appendix A.4.

Table 1. Command Listing and Descriptions

Command	Description
Move(2.02,1.01)	Causes the machine to move an absolute distance from the current position. In this example, 2.02 represents a 2.02 unit move in the X axis direction and 1.01 represents a 1.01 unit move in the Y axis direction.
Posn(3.1,2.2)	Causes the machine to move a distance relative to the zero point. This first number is the distance in the X direction, the second number is the distance to move from zero in the Y axis direction.
For(1,11,2)	Represents the beginning of a loop. Where 1 is the starting number, 11 is the Ending number, and 2 represents the increment. Numbers only are allowed in these fields.
Rpos	Reads only the position data from the laser system.
Xmea	Combines the vision system data with the interferometer data to result in an X axis measurement. (Y-axis measurements are not being used at this time).
Mesg(This is a prompt)	Outputs a prompt in the bottom text area (next to the OK button) for the operator to respond. In this case, the text displayed would be 'This is a prompt'.
Next	Represents the end of a loop began with the 'For' command.
Reswin(25,690)	Causes the window to be resized. In this example, the window would be 25 units wide and 690 units in height.
Poswin(300,50)	Moves the position of the window. In this example, the upper left-hand corner of the window would be moved 300 units from the left side of the main window and 50 units from the top of the main window.
Prnt(Text)	Prints a message input by the operator to the data file. In this case, the word 'Text' would be output to the data file. This is especially useful for printing the program name and control number in the data file.
GetEnv	Collects data from the temperature system, barometer, and hygrometer. Recalculates the laser compensation and applies the new compensation. This command will also cause a re-zero wherever it occurs.
LoadPeek(filename)	Loads a file, which contains a specific set of vision system parameters. Parameters such as threshold can be specifically set for each scale if needed.
Xzer(1)	Causes the X-axis zero point to be set at the current position. The 1

indicates centering of the line will occur. A 0 would not cause recentering of the line within the window.

Yzer                      Cause the Y-axis zero point to be set at the current position. Exact positioning of the Y-axis has not been necessary.

An additional visual basic program was written to monitor the laboratory temperature. The program reads the temperature and humidity in the lab and records the time, temperature, and humidity in a file, which includes the date in the name of the file. In this way, the engineers can monitor the temperature of the lab from their desk. It also makes it easier to analyze the lab's performance. The 'AT' command is used to activate the laboratory temperature-monitoring program. The program runs in the background and is invisible to the operator.

### Measurement and Design Considerations

The number of error sources needed to be reduced in order to obtain the lowest possible measurement uncertainty. Since the measurements occur along the X axis, the uncertainty contributions from movement in the Y or Z axis could be eliminated by keeping the Y and Z axes stationary during measurements. This was accomplished through manual alignment of the part to the X axis both horizontally and vertically and through careful setup of the software. A leveling table is used for the vertical alignment with the X axis, and manual movement of the glass scale is used for the horizontal alignment. The leveling table serves another purpose by raising the glass vertically in line with the laser beam. This eliminates the abbe' error due to the X-axis pitch while positioning of the glass scale horizontally in line with the beam eliminates the abbe' error due to X-axis yaw. The software routines for movement of the X and Y axes were created to ensure that movement of the Y axis occurred only when the operator inputs a change in the position in the Y-axis text box. Through these considerations, the uncertainty was reduced as low as possible. (Figure 5 shows the software operation.) Table 2 displays the results of measurements taken on a stage micrometer in comparison to measurements taken on the same standard at the National Institute of Standards and Technology (NIST). The uncertainty of the NIST measurements is +/-0.000002 inch.



Figure 5. Computer Screen During Software Operation

Table 2. Trioptic Measurement Results vs. NIST Measurements Results (in inches)

Trioptic Run #1	Trioptic Run #2	Trioptic Run #3	Trioptic Average	NIST Certified Values	Difference
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.099991	0.099989	0.099989	0.099990	0.099988	0.000002
0.199987	0.199987	0.199988	0.199987	0.199986	0.000001
0.399957	0.399955	0.399954	0.399955	0.399954	0.000001
0.499958	0.499960	0.499958	0.499959	0.499957	0.000002
0.599970	0.599972	0.599970	0.599971	0.599971	0.000000
0.699958	0.699959	0.699960	0.699959	0.699958	0.000001
0.799946	0.799948	0.799946	0.799947	0.799948	-0.000001
0.899928	0.899931	0.899934	0.899931	0.899930	0.000001
0.999924	0.999928	0.999928	0.999927	0.999928	-0.000001
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

0.010034	0.010027	0.010033	0.010031	0.010031	0.000000
0.020040	0.020032	0.020039	0.020037	0.020038	-0.000001
0.030040	0.030041	0.030040	0.030040	0.030038	0.000002
0.040026	0.040025	0.040028	0.040026	0.040026	0.000000
0.050008	0.050010	0.050008	0.050009	0.050008	0.000001
0.060012	0.060007	0.060011	0.060010	0.060010	0.000000
0.070037	0.070038	0.070037	0.070037	0.070037	0.000000
0.080025	0.080022	0.080025	0.080024	0.080025	-0.000001
0.090012	0.090015	0.090015	0.090014	0.090016	-0.000002
0.099997	0.099999	0.099997	0.099998	0.099997	0.000001
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000980	0.000979	0.000978	0.000979	0.000976	0.000003
0.001979	0.001982	0.001983	0.001981	0.001980	0.000001
0.002987	0.002988	0.002991	0.002988	0.002985	0.000003
0.003980	0.003982	0.003980	0.003981	0.003979	0.000002
0.004976	0.004983	0.004976	0.004978	0.004975	0.000003
0.005977	0.005976	0.005975	0.005976	0.005975	0.000001
0.006973	0.006972	0.006972	0.006972	0.006971	0.000001
0.007956	0.007964	0.007959	0.007960	0.007961	-0.000001
0.008981	0.008982	0.008979	0.008980	0.008980	0.000000
0.009991	0.009991	0.009987	0.009989	0.009991	-0.000002
0.010969	0.010968	0.010970	0.010969	0.010967	0.000002
0.011967	0.011967	0.011967	0.011967	0.011965	0.000002
0.012976	0.012976	0.012974	0.012975	0.012974	0.000001
0.013972	0.013967	0.013962	0.013967	0.013963	0.000004
0.014966	0.014962	0.014958	0.014962	0.014960	0.000002
0.015967	0.015958	0.015955	0.015960	0.015959	0.000001
0.016965	0.016963	0.016962	0.016963	0.016962	0.000001
0.017963	0.017960	0.017962	0.017962	0.017962	0.000000
0.018961	0.018963	0.018961	0.018962	0.018962	0.000000
0.019978	0.019972	0.019969	0.019973	0.019973	0.000000

## Accomplishments

The original system was modified through replacement and addition of a number of components. The laser-based edge detector was replaced with a vision system which consists of a high-resolution camera and a high-speed image capture PCI bus card. The location of the optics was changed in order to allow measurement of glass scales in line with the laser beam in order to eliminate abbe' error. The original motors were replaced with stepper motors and a two-axis controller. The computer was replaced with a PC running under the Windows NT operating system. All of the components of the system were integrated and automated through creation of software, which is easily used and understood. Machine programs can be created offline or on the same computer while an object is being measured.

The capability to measure high-quality glass scales to an uncertainty of  $\pm 0.000020$  inch has been maintained. The time required to measure the scales has been reduced by approximately half through automation and a reduction in setup time.

## Future Work

The Y-axis measurement capability has not been developed. Although the uncertainty would not be as low for Y-axis measurements, the repeatability of measurements along the Y axis indicate that the development effort would be worthwhile.

The two-dimensional measurement capability of the system needs to be developed and tested to determine the capability to measure small grid plates and two-dimensional glass scales.

## References

1R. N. Burton, *Development of a Length Scale Calibration System Using Laser Edge Sensing*, (Final Report). Honeywell Federal Manufacturing & Technologies: KCP-613-4630, January 1992 (Available from NTIS).

2The iMAGES Report, 'Best Practices in Automated Imaging and Machine Vision', Volume 5, Number 2, Spring 1998, © 1998, Imaging Technology Incorporated.

## Appendix A

### Software Routines

#### A.1 Vision System Calibration Routine

```
Private Sub CalibratePixels_GotFocus()
```

```
Dim CalRange As Single
```

'This routine is executed if the CalibratePixels button (PxCl) is pressed.

'The form will become invisible in order to keep any other routines from executing.

```
Form1.Visible = False
```

```
If CalFlg = 1 Then
```

```
Mesg = "Place edge on the right end of Line A and click OK."
```

```
ElseIf CalFlg = 2 Then
```

```
Mesg = "Place edge on the left end of Line A and click OK."
```

```
ElseIf CalFlg = 3 Then
```

```
Mesg = "Place edge on the top of line A and click OK."
```

```
ElseIf CalFlg = 4 Then
```

```
Mesg = "Place edge on the bottom of line A and click OK."
```

```
End If
```

'This subroutine is made up of four sections, one sets up the X axis calibration,

'another sets up the Y axis calibration, one section sets up the filename.

```
If Indx < 11 Then
```

```
If Indx <> 2 Then Indx = 1
```

```
If Lsp = 0 Then
```

```
If 0 = SPIinit(0) Then
```

```
Unload MainWizardForm
```

```
End If
```

```
x = SPWndOnTop(hwnd, 1)
```

```
End If
```

```
Lsp = 1
```

```
Load.Visible = False
Setpeek.Visible = False
Dir1.Path = "C:\sher32\Programs\"
Dir1.Visible = False
Drive1.Visible = False
File1.Visible = False
Form1.Top = 9264
Form1.Width = 12924
Motor.Visible = True
Atxt.Visible = True
Ptxt.Visible = True
Bptxt.Visible = True
Htxt.Visible = True
Atemp.Visible = True
Ptemp.Visible = True
Barom.Visible = True
Relhum.Visible = True
Environment.Visible = True
Pcof.Visible = True
MetFr.Visible = True
'show the picture
If CalFlg = 1 Then
Response = SPLoadFile("xpixelcal.inv")
Response = SPSOLive("soA")
'continuously take pictures
```

```

x = SPInspectContinuous()

End If

CmdTxt.Visible = True

CmdTxt.Text = Mesg

CmdButton.Visible = True

'set CmdButton as the default so that the CmdBut Subroutine is executed when the
'Enter key is pressed.

CmdButton.Default = True

ElseIf Indx = 11 Then

CalRange = CalCoord(2) - CalCoord(1)

'Check the variable B of the peek. This variable determines whether the subject
'has been placed within the field of view. It must be greater than 30.

x = SPQueryVar("varB")

' If x < 30 Then

' Mesg = "Readjust focus or move edge further onto line and click OK."

' Indx = 1

' CmdTxt.Visible = True

' CmdTxt.Text = Mesg

' CmdButton.Visible = True

' CmdButton.Default = True

' Exit Sub

' End If

Xmtxt.Text = CStr((-CalRange / 20) * Un)

Ymtxt.Text = "0"

Call MoveAxis(Xmtxt.Text, Ymtxt.Text)

```

```

Pixelxinit = 0

Laserxinit = 0

For I = 1 To 10

'Variable A is the midpoint of two lines.

x = SPQueryVar("varA")

x = Mid$(x, InStr(1, x, "(") + 1, InStr(1, x, ",") - 2)

Pixelxinit = x + Pixelxinit

IER = 0

Call ColData(IER, Errmsg$)

Response = Errmsg$

If IER > 0 Then MsgBox ("Error occurred in data col. routine")

x = -Mid$(Xch, 1, 9)

Laserxinit = x + Laserxinit

Next I

'Average the ten readings to set the initial readings.

Pixelxinit = Pixelxinit / 10

Laserxinit = (Laserxinit / 10) / Un

Inchesperpixel = 0

For Indx = 1 To 8

Xmtxt.Text = CStr((-CalRange / 10) * Un)

Ymtxt.Text = "0"

'Move the subject throughout the viewport.

Call MoveAxis(Xmtxt.Text, Ymtxt.Text)

'Take data on both the laser and the camera pixels and return with

'Inchesperpixel set.

```

```

GoSub GetDatax

Inchesperxpixel = x + Inchesperxpixel

Next Indx

'Average the inches per x axis pixels.

Inchesperxpixel = Inchesperxpixel / 8

'Save this piece of data in the pixelcal data file.

Open "C:\sher32\pixelcal.dat" For Output As #1

Print #1, "Inchesperxpixel= ", Inchesperxpixel

Close #1

Pixelxinit = 0

Indx = 2

'Now load in the viewport for the calibration of the y axis pixel calibration

'and repeat what was done for the X axis.

Response = SPLoadFile("ypixelcal.inv")

Response = SPSOLive("soA")

x = SPInspectContinuous()

CmdTxt.Visible = True

CmdTxt.Text = "Place an edge on the Top of lineA and click OK."

CmdButton.Visible = True

CmdButton.Default = True

ElseIf Indx = 12 Then

CalRange = CalCoord(4) - CalCoord(3)

x = SPQueryVar("varB")

' If x < 30 Then

' Form1.Visible = True

```

```

' Mesg = "Readjust focus or move edge further onto line and click OK."

' Indx = 2

' CmdTxt.Visible = True

' CmdTxt.Text = Mesg

' CmdButton.Visible = True

' CmdButton.Default = True

' Exit Sub

' End If

Xmtxt.Text = "0"

Ymtxt.Text = CStr((-CalRange / 20) * Un)

Call MoveAxis(Xmtxt.Text, Ymtxt.Text)

Pixelyinit = 0

Laseryinit = 0

For I = 1 To 10

y = SPQueryVar("varA")

y = Mid$(y, InStr(1, y, "(") + 1, InStr(1, y, ",") - 2)

Pixelyinit = y + Pixelyinit

IER = 0

Call ColData(IER, Errmsg$)

Response = Errmsg$

If IER > 0 Then MsgBox ("Error occurred in data col. routine")

y = -Mid$(Ych, 1, 9)

Laseryinit = y + Laseryinit

Next I

Pixelyinit = Pixelyinit / 10

```

```

Laseryinit = (Laseryinit / 10) / Un
Inchesperypixel = 0
For Indx = 1 To 8
Ymtxt.Text = CStr((-CalRange / 10) * Un)
Xmtxt.Text = "0"
Call MoveAxis(Xmtxt.Text, Ymtxt.Text)
GoSub GetDatax
Inchesperypixel = y + Inchesperypixel
Next Indx
Inchesperypixel = Inchesperypixel / 8
Open "C:\sher32\pixelcal.dat" For Append As #1
Print #1, "Inchesperypixel= ", Inchesperypixel
Close #1
Indx = 0
Pixelyinit = 0
CalFlg = 1
End If
'Make the form visible again and exit.
Form1.Visible = True
Exit Sub
GetDatax:
PixelxCoord = 0
LaserxCoord = 0
For I = 1 To 10
x = SPQueryVar("varA")

```

x = Mid\$(x, InStr(1, x, "(") + 1, InStr(1, x, ",") - 2)

PixelxCoord = x + PixelxCoord

IER = 0

Call ColData(IER, Errmsg\$)

Response = Errmsg\$

If IER > 0 Then MsgBox ("Error occurred in data col. routine")

x = -Mid\$(Xch, 1, 9)

LaserxCoord = x + LaserxCoord

Next I

LaserxCoord = (LaserxCoord / 10) - Laserxinit

PixelxCoord = (PixelxCoord / 10) - Pixelxinit

If in Metric, LaserxCoord will be in metric. This must be converted to english.

x = (LaserxCoord / Un) / PixelxCoord

Return

GetDatay:

PixelyCoord = 0

LaseryCoord = 0

For I = 1 To 10

y = SPQueryVar("varA")

y = Mid\$(y, InStr(1, y, ",") + 1, InStr(1, y, ")") - InStr(1, y, ",") - 1)

PixelyCoord = y + PixelyCoord

IER = 0

Call ColData(IER, Errmsg\$)

Response = Errmsg\$

If IER > 0 Then MsgBox ("Error occurred in data col. routine")

y = -Mid\$(Ych, 1, 9)

LaseryCoord = y + LaseryCoord

Next I

LaseryCoord = (LaseryCoord / 10) - Laseryinit

PixelyCoord = (PixelyCoord / 10) - Pixelyinit

y = (LaseryCoord / Un) / PixelyCoord

Return

End Sub

## A.2 Routine for Measurements in the X Axis

Public Sub MeasureXaxis()

'Edge detection along X axis.

Dim Adjustx As String

Dim CurDate As String

'The following line will need to be changed later to allow different types of

'edgefinds. Use this for now\*\*\*\*\*

'if a peek for X axis measurements is not loaded, load this one.

If XmeaFl = 0 Then

Response = SPLoadFile("xpixelcal.inv")

Response = SPSOLive("soA")

x = SPIinspectContinuous()

End If

'If a data file has not been opened perform the following.

If DFileFlg = 0 Then

'If Typ <> "Append" Then

```
CmdTxt.Text = "Enter the data file name and click OK."  
  
CmdTxt.Visible = True  
  
CmdTxt.SetFocus  
  
CmdTxt.SelStart = 0  
  
CmdTxt.SelLength = Len(CmdTxt.Text)  
  
Typ = "Output"  
  
DFileFlg = 1  
  
'End If  
  
Indx = 4  
  
CmdButton.Visible = True  
  
CmdButton.Default = True  
  
Call ZeroXaxis(0)  
  
Do Until Indx = 0  
  
DoEvents  
  
Response = "File Name?"  
  
Loop  
  
'If the data file has been opened then perform this section.  
  
ElseIf DFileFlg = 1 Then  
  
CurDate = Date  
  
Form1.Visible = False  
  
CmdTxt.Text = ""  
  
CmdButton.Default = False  
  
If Typ = "Append" Then Open "C:\sher32\data\" + DFile For Append As #1  
  
If Typ = "Output" Then  
  
Open "C:\sher32\data\" + DFile For Output As #1
```

```

Print #1, "Created " + CurDate + " Trioptic, CS-09P-217, Iss A, 02/04/1999"

Print #1, "Data File= " + DFile

Print #1, " "

Print #1, "Air Temp= " + CStr(Atmp) + " Part Temp= " + CStr(Ptmp) + " Humidity= " + CStr(Rhum)

Print #1, "Barom Press= " + CStr(Bpres) + " Exp Coef= " + CStr(Pcoef) + " Cor Factor= " + CStr(Cmp)

Print #1, " "

End If

PixelxCoord = 0

LaseroxCoord = 0

LaseryCoord = 0

For I = 1 To 5

x = SPQueryVar("varA")

x = Mid$(x, InStr(1, x, "(") + 1, InStr(1, x, ",") - 2)

PixelxCoord = x + PixelxCoord

Call ColData(IER, Errmsg$)

Response = Errmsg$

If IER > 0 Then MsgBox ("Error reading Laser")

x = -Mid$(Xch, 1, 9)

y = -Mid$(Ych, 1, 9)

LaseroxCoord = LaseroxCoord + x

LaseryCoord = LaseryCoord + y

Next I

LaseroxCoord = LaseroxCoord / 5

LaseryCoord = LaseryCoord / 5

PixelxCoord = (Pixelxinit - PixelxCoord / 5) * (Inchesperpixel * Un)

```

```
LaserxCoord = LaserxCoord + PixelxCoord
LaserxCoord = Format(LaserxCoord, "###0.000000")
LaseryCoord = Format(LaseryCoord, "###0.000000")
Print #1, LaserxCoord, LaseryCoord
XTxtRes.Text = LaserxCoord
YTxtRes.Text = LaseryCoord
If Un = 25.4 Then
XTxtRes.Text = Format(XTxtRes.Text, "###0.000000")
YTxtRes.Text = Format(YTxtRes.Text, "###0.000000")
End If
Beep
Close #1
If Learn = "ON" Then
Open ProgNam For Append As #2
Print #2, "Xmea"
Close #2
CmdRecNum = CmdRecNum + 1
End If
Typ = "Append"
Indx = 0
MvIndx = 0
Form1.Visible = True
End If
End Sub
```

A.3 Positioning Routine

Public Sub PositionAxis(Xposition As String, Yposition As String)

Dim Xsetback As Single

Dim Ysetback As Single

MSComm1.Output = "D" + vbCrLf

'Positions each axis (relative to 0 point)

MSComm1.Output = "J" + vbCrLf

IER = 0

IER = 0

'Get current position from laser electronics.

Call ColData(IER, Errmsg\$)

Response = Errmsg\$

If IER > 0 Then MsgBox ("Error reading Laser")

Xlaser = CSng(-Xch): Ylaser = CSng(-Ych)

'Mx and My will be set to 0 to keep the axis from moving unless

'absolutely necessary. For single axis measurements only one axis

'should move during the measurements.

'If the position is within 0.000030 inches then leave it alone.

If Abs(CSng(Xposition) - Xlaser) < (0.00003 \* Un) Then Mx = 0

'If the position is the same as the last position command, then leave it alone.

If Xposition = Xlast Then Mx = 0

If Yposition = Ylast Then My = 0

If Abs(CSng(Yposition) - Ylaser) < (0.00003 \* Un) Then My = 0

'If both a set to 0 don't move anything.

If Mx = 0 And My = 0 Then GoTo EndPositionAxis

'The first move is always a negative .013 inches from the destination

'However, if the movement is in the negative direction, an extra .010 inches  
'is necessary due to the hysteresis of approximately .011 inches.

$$Xsetback = 0.015 * Un$$

$$Ysetback = 0.015 * Un$$

If (CSng(Xposition) - Xlaser) < 0 And Mx <> 0 Then

$$Xsetback = (0.025 * Un) + (Abs(CSng(Xposition) - Xlaser) * (0.0027 * Un))$$

End If

If (CSng(Yposition) - Ylaser) < 0 And My <> 0 Then

$$Ysetback = (0.025 * Un) + (Abs(CSng(Yposition) - Ylaser) * (0.0065 * Un))$$

End If

'Move to the negative side of the target by 0.013 inches.

'Again note the Abs(CSng(Mx))>0 which will keep the axis from moving if Mx=0

$$Xnum = (CSng(Xposition) - Xlaser - Xsetback) * (Abs(CSng(Mx)) > 0)$$

'Calculate an approximation for the steps to move

'The 0.0001 is to prevent division by 0

$$Xnum = Xnum * 4560 / Un$$

$$Ynum = -(CSng(Yposition) - Ylaser - Ysetback) * (Abs(CSng(My)) > 0)$$

$$Ynum = Ynum * 4450 / Un$$

'if both are 0 don't move anything

If Xnum = 0 And Ynum = 0 Then GoTo EndPositionAxis

$$Xmv = Mid$(CStr(Xnum), 1, InStr(CStr(Xnum), ".") + 1)$$

$$Ymv = Mid$(CStr(Ynum), 1, InStr(CStr(Ynum), ".") + 1)$$

MSComm1.Output = "I X F6000 D" + Xmv + "Y F1800 D" + Ymv + "\*" + vbCrLf

Do

Buffer\$ = Buffer\$ + MSComm1.Input

```
Loop Until InStr(Buffer$, "%") > 0
```

```
Buffer$ = ""
```

```
MSComm1.Output = "Q" + vbCrLf
```

```
Buffer$ = MSComm1.Input
```

```
Buffer$ = ""
```

'Following movement to the -X or -Y side of the target, a movement needs to be

'made to eliminate most but not all of the hysteresis.

```
Xnum = (0.01 * Un) * (Abs(CSng(Mx)) > 0)
```

```
Xnum = Xnum * 4560 / Un
```

```
Ynum = -(0.01 * Un) * (Abs(CSng(My)) > 0)
```

```
Ynum = Ynum * 4560 / Un
```

```
Xmv = Mid$(CStr(Xnum), 1, InStr(CStr(Xnum), ".") + 1)
```

```
Ymv = Mid$(CStr(Ynum), 1, InStr(CStr(Ynum), ".") + 1)
```

```
MSComm1.Output = "I X F3000 D" + Xmv + "Y F1000 D" + Ymv + "*" + vbCrLf
```

```
Do
```

```
Buffer$ = Buffer$ + MSComm1.Input
```

```
Loop Until InStr(Buffer$, "%") > 0
```

```
Buffer$ = ""
```

```
MSComm1.Output = "Q" + vbCrLf
```

```
Buffer$ = MSComm1.Input
```

```
Buffer$ = ""
```

```
For Incr = 1 To 35
```

```
IER = 0
```

```
Call ColData(IER, Errmsg$)
```

```
Response = Errmsg$
```

```

If IER > 0 Then MsgBox ("Error reading Laser")

Xlaser2 = CSng(-Xch): Ylaser2 = CSng(-Ych)

XCoorTxt.Text = -Mid$(Xch, 1, 9)
YCoorTxt.Text = -Mid$(Ych, 1, 9)

If Un = 25.4 Then

XCoorTxt.Text = Format(XCoorTxt.Text, "###0.00000")
YCoorTxt.Text = Format(YCoorTxt.Text, "###0.00000")

End If

Xptxt.Text = Xposition: Yptxt.Text = Yposition

'Fine positioning. Note the Abs(CSng(Mx))>0 which will set Xnum=0 if Mx=0
Xnum = (CSng(Xposition) - Xlaser2) * (Abs(CSng(Mx)) > 0)
Xnum = Format(Xnum, "##0.000000")
If Abs(Xnum) < (0.00003 * Un) Or Xnum > 0 Then Xnum = 0#
Ynum = -(CSng(Yposition) - Ylaser2) * (Abs(CSng(My)) > 0)
If Abs(Ynum) < (0.00006 * Un) Or Ynum < 0 Then Ynum = 0
If Xnum = 0 And Ynum = 0 Then GoTo RelaxTorque2
Xnum = Xnum * 2500 / Un
Ynum = Ynum * 1800 / Un
Xnum = Format(Xnum, "##0.000")
Ynum = Format(Ynum, "##0.000")
Xmv = Mid$(CStr(Xnum), 1, InStr(CStr(Xnum), ".") + 1)
If Abs(Xnum) < 0.1 And Abs(Xnum) > 0# Then Xmv = CStr(0.1 * (Xnum / Abs(Xnum)))
Ymv = Mid$(CStr(Ynum), 1, InStr(CStr(Ynum), ".") + 1)
If Abs(Ynum) < 0.1 And Abs(Ynum) > 0# Then Ymv = CStr(0.1 * (Ynum / Abs(Ynum)))
Speed$ = "1500"

```

```
MSComm1.Output = "I X F" + Speed$ + " D" + Xmv + "Y F" + CStr(CInt(Speed$) / 2) + " D" + Ymv + "*" + vbCrLf
```

```
Do
```

```
Buffer$ = Buffer$ + MSComm1.Input
```

```
Loop Until InStr(Buffer$, "%") > 0
```

```
Buffer$ = ""
```

```
MSComm1.Output = "Q" + vbCrLf
```

```
Buffer$ = MSComm1.Input
```

```
Buffer$ = ""
```

```
Next Incr
```

```
RelaxTorque2:
```

```
'Backoff 0.001 inch in order to eliminate the drift.
```

```
If Mx = 1 Then Xnum = (((0.001 * Un) / 0.0002196)) / Un + 0.00001
```

```
If My = 1 Then Ynum = (((-0.001 * Un) / 0.00021708)) / Un + 0.00001
```

```
Xmv = Mid$(CStr(Xnum), 1, InStr(CStr(Xnum), ".") + 1)
```

```
Ymv = Mid$(CStr(Ynum), 1, InStr(CStr(Ynum), ".") + 1)
```

```
MSComm1.Output = "I X F500 D" + Xmv + "Y F500 D" + Ymv + "*" + vbCrLf
```

```
Do
```

```
Buffer$ = Buffer$ + MSComm1.Input
```

```
Loop Until InStr(Buffer$, "%") > 0
```

```
Buffer$ = ""
```

```
MSComm1.Output = "Q" + vbCrLf
```

```
Buffer$ = MSComm1.Input
```

```
Buffer$ = ""
```

```
MSComm1.Output = "K" + vbCrLf
```

```
MSComm1.Output = "C" + vbCrLf
```

Pause (0.3)

MSComm1.Output = "##" + vbCrLf

MSComm1.Output = "S" + vbCrLf

EndPositionAxis:

'Set Movement flags and save last position input to compare to new position

'input. Don't move an axis unless necessary.

Mx = 0

My = 0

Xlast = Xposition

Ylast = Yposition

Ps.Default = False

MvIndx = 0

Call ColData(IER, Errmsg\$)

Response = Errmsg\$

If IER > 0 Then MsgBox ("Error reading Laser")

Xlaser2 = CSng(-Xch): Ylaser2 = CSng(-Ych)

Xptxt.Text = -Mid\$(Xch, 1, 9)

Yptxt.Text = -Mid\$(Ych, 1, 9)

If Un = 25.4 Then Xptxt.Text = Format(XCoorTxt.Text, "###0.00000")

If Un = 25.4 Then Yptxt.Text = Format(YCoorTxt.Text, "###0.00000")

End Sub

#### A.4 Machine Program Example

CE20822.prg Created on 3/18/1999

CS-09P-221, Iss. A, 03/18/1999

LoadPeek(cn20822.inv)

Metr

GetEnv

Prnt(CE20822.PRG, CS-09P-221, Iss. A, 03/18/1999)

Resize(40,400)

Mesg(Align scale and position the first mark within the window.)

For(1,3,1)

Xzer(0)

Xzer(1)

Xmea

Posn(.01,0.0)

Xmea

Posn(.02,0.0)

Xmea

Posn(.03,0.0)

Xmea

Posn(.04,0.0)

Xmea

Posn(.05,0.0)

Xmea

Posn(.06,0.0)

Xmea

Posn(.07,0.0)

Xmea

Posn(.08,0.0)

Xmea

Posn(.09,0.0)

Xmea

Posn(.1,0.0)

Xmea

Posn(.2,0.0)

Xmea

Posn(.3,0.0)

Xmea

Posn(.4,0.0)

Xmea

Posn(.5,0.0)

Xmea

Posn(.6,0.0)

Xmea

Posn(.7,0.0)

Xmea

Posn(.8,0.0)

Xmea

Posn(.9,0.0)

Xmea

Posn(1,0.0)

Xmea

Posn(0,0.0)

Xmea

Next

Appendix B  
Data

B.1 Repeatability Data File

Created 8/5/99 Trioptic, CS-09P-217, Iss A, 02/04/1999

Data File= Repeat2.dat

Air Temp= 20.426 Part Temp= 20.4285 Humidity= 35.7

Barom Press= 740.763 Exp Coef= 0.0000085 Cor Factor= 0.999732271938057

X axis Y axis

-0.000001 -0.000001

-0.000007 -0.000001

-0.000005 0.000000

-0.000004 0.000000

-0.000001 -0.000001

-0.000007 0.000000

0.000001 0.000000

-0.000005 -0.000001

-0.000005 -0.000002

-0.000004 0.000000

-0.000002 0.000000

0.000006 -0.000001  
-0.000001 -0.000001  
-0.000004 0.000000  
-0.000007 -0.000001  
-0.000002 0.000000  
-0.000001 0.000001  
-0.000004 0.000000  
-0.000001 0.000000  
-0.000004 0.000000  
0.000003 -0.000001  
-0.000004 -0.000001  
-0.000004 0.000000  
0.000000 0.000000  
-0.000002 0.000001  
-0.000001 -0.000001  
0.000001 0.000000  
0.000003 -0.000001  
-0.000001 0.000000  
-0.000001 0.000000  
-0.000001 0.000000  
-0.000004 0.000000  
-0.000005 -0.000001