VISION SYSTEM PARAMETER SELECTION FOR FLEXIBLE MATERIALS HANDLING

by

Craig A. Jungwirth

SUBMITTED TO THE DEPARTMENT OF
MECHANICAL ENGINEERING IN PARTIAL
FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF

BACHELOR OF SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 1988

Signature Redacted

Signature of Author ______________________
Department of Mechanical Engineering
May 27, 1988

Signature Redacted

Certified by ______________________
Professor Harry West
Thesis Supervisor

Signature Redacted

Accepted by ______________________
Professor Peter Griffith
Chairman, Department Committee

VISION SYSTEM PARAMETER SELECTION FOR FLEXIBLE MATERIALS HANDLING

by

Craig A. Jungwirth

Submitted to the Department of Mechanical Engineering on May 27, 1988 in partial fulfillment of the requirements for the degree of Bachelor of Science.

Abstract

The Charles Stark Draper Laboratory has developed a vision-guided, fully-automated folding-sewing machine under the auspices of the Textile/Clothing Technology Corporation for the fabrication of suit sleeves. The vision system which guides this machine was designed to provide two-dimensional data in order to guide robot motions. Vision system algorithms consult a tree-structured database for geometry-dependent variables. Previously, an experienced user could come up with rough parameters for a new cloth geometry in a few minutes. The work presented here integrates the vision system operation with a cloth-folding expert system which was added to the machine. Issues in manual and automatic selection of vision system parameters for varying cloth geometries are addressed.

Thesis Supervisor: Professor Harry West
Title: Assistant Professor of Mechanical Engineering

Acknowledgements

I would like to thank my family for their support. Thanks also goes to my thesis advisor, Professor Harry West, for his patience and guidance. Finally, I would like to thank Mike Foley, Ed Bernardon, and the TCI staff at Draper for allowing me to join in their project.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 The Apparel Industry

Apparel manufacture is a labor-intensive industry. Reduction in variable labor costs therefore presents itself as an area in which the application of automation would significantly reduce production costs across the entire apparel industry. Given that the apparel industry is also the third largest consumer industry in the United States, with shipments of over $56.1 billion in 1982 [Bernardon 86], cost reduction could lead to greater profitability as well as competitiveness with foreign competitors.

Foreign competitors captured 13.7% of the U.S. apparel market in 1975 [Bernardon 86]. These firms are able to capture such a large share of the domestic apparel market because of the low wages garnered by overseas garment workers and other economic factors. Retention of market share in the face of foreign competition, therefore, is another motivation for reducing labor costs.

1.2 The Draper/[TC]^2 Folder-Sewer Machine

1.2.1 Evolution of the Draper/[TC]^2 Folder-Sewer Machine

A group of apparel industry firms formed the Textile/Clothing Technology Corporation ([TC]^2) in order to explore the feasibility of introducing automation into their industry. The level of sophistication which present-day robots and computers have attained allows the consideration of automation as a viable alternative to laborers. [TC]^2 therefore enlisted the expertise of Charles Stark Draper Laboratory personnel in 1981 to develop a vision-guided, fully-automated folding-sewing machine to fabricate suit sleeves. The project, which has evolved through several iterations and produced prototype machines, is manifested in the Draper/[TC]^2 folder-sewer machine shown in Figure 1-1.

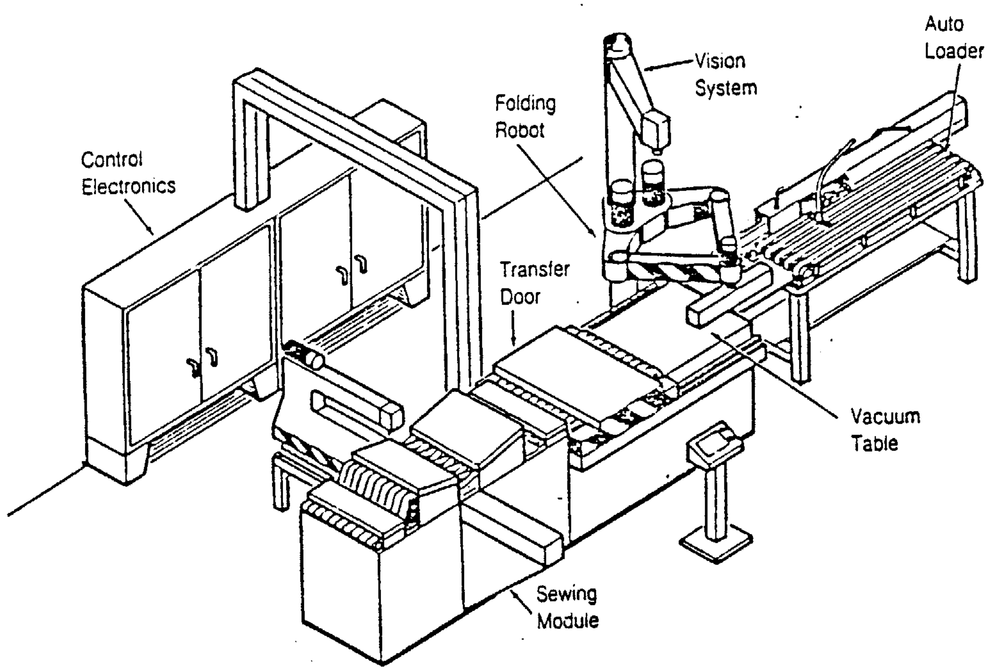


Figure 1-1: The Charles Stark Draper Laboratory/[TC]^2 Folder-Sewer Machine

1.2.2 Operation of the Draper/[TC]^2 Folder-Sewer Machine

The Draper/[TC]^2 folder-sewer machine consists of a folding station and a sewing station. Cloth is fed to the folding station from a staging station onto the perforated vacuum table. This table is coated with a retro-reflective material. Two vacuum blowers equipped with gates constrain the cloth to the two-dimensional plane of the table. A CCD camera is mounted above the folding table. Four fiber-optic cable bundles surrounding the camera light the table. An IBM 7547 industrial robot with an end-effector designed and built at Draper manipulates cloth placed on the vacuum table's surface.

The machine's vision system provides cloth coordinate data to guide robot motions. Draper personnel along with consultants from SPARTA, Inc. developed this vision system to guide the manipulation of the non-rigid cloth [Wyschogrod 87]. Coordinates provided by the vision system also serve to control the machine's sewing station.

A bank of belts transports the cloth between the folding and sewing stations. This bank of belts and a second set manipulate the cloth underneath a Singer sewing machine. Once sewing operations are completed, the first set of belts transports the cloth back to the folding table. This series of operations is repeated until all folding and sewing operations are completed. The completed garment is then ejected by the second set of belts.

1.3 Enhancements to the Draper/[TC]^2 Folder-Sewer Machine

Draper personnel proposed a series of enhancements to the operation of the folding station of the machine. A user interface, folding expert systems, and program generator were developed.

1.3.1 The User Interface

The user interface allows the user to monitor and control actions at the folding station. Implemented in C using the SunView window utility on a Sun Workstation, the user interface allows direct control of robot functions, invocation of the folding expert system, monitoring of workcell actions, and previewing of automatic program generator and folding expert system output.

The purpose of the the user interface for the folding domain is to set the folder states and build tasks which are sent to a process controller for execution or text generation. The states can be set through a graphical interface by the user or automatically by using the folding expert system. Because states can represent all possible folder component configurations, a sequence of states can be used to create robot programs. The user interface sets the states graphically, eliminating the need for the user to write robot control programs.

The user interface has a window for each of the folder components and is shown in Figure 1-2. Control buttons and data entry fields within each window are used to control the components. A communication window allows for text communication with the user. And pull-down menus are used to set states and tasks and to control robot program creation.

For the folding domain, the user interface is primarily responsible for taking all inputs from the user and from the vision system [Foley 87]. It must then create databases regarding cloth geometry and qualitative characteristics. The user interface then processes the user requirements for the type and accuracy of fold desired and communicates the initial and goal states for the fold to the expert system. It does this by assembling a file of robot actions to guide expert system processing.

1.3.2 The Folding Expert System

The folding expert system was designed in order to assist personnel in programming the folding station of the Draper/[TC]^2 Folder-Sewer machine. It accepts as input cloth characteristics and coordinate data and generates the required robot and end effector operations to manipulate the cloth successfully into a desired, folded position.

The folding expert system is the successful result of attempts to define and implement a generalized, uniform architecture for describing high-level robotic planning systems. The output of this system is then transformed into commands which direct actions in the workcell to reach a goal given a set of initial conditions and a file of actions.

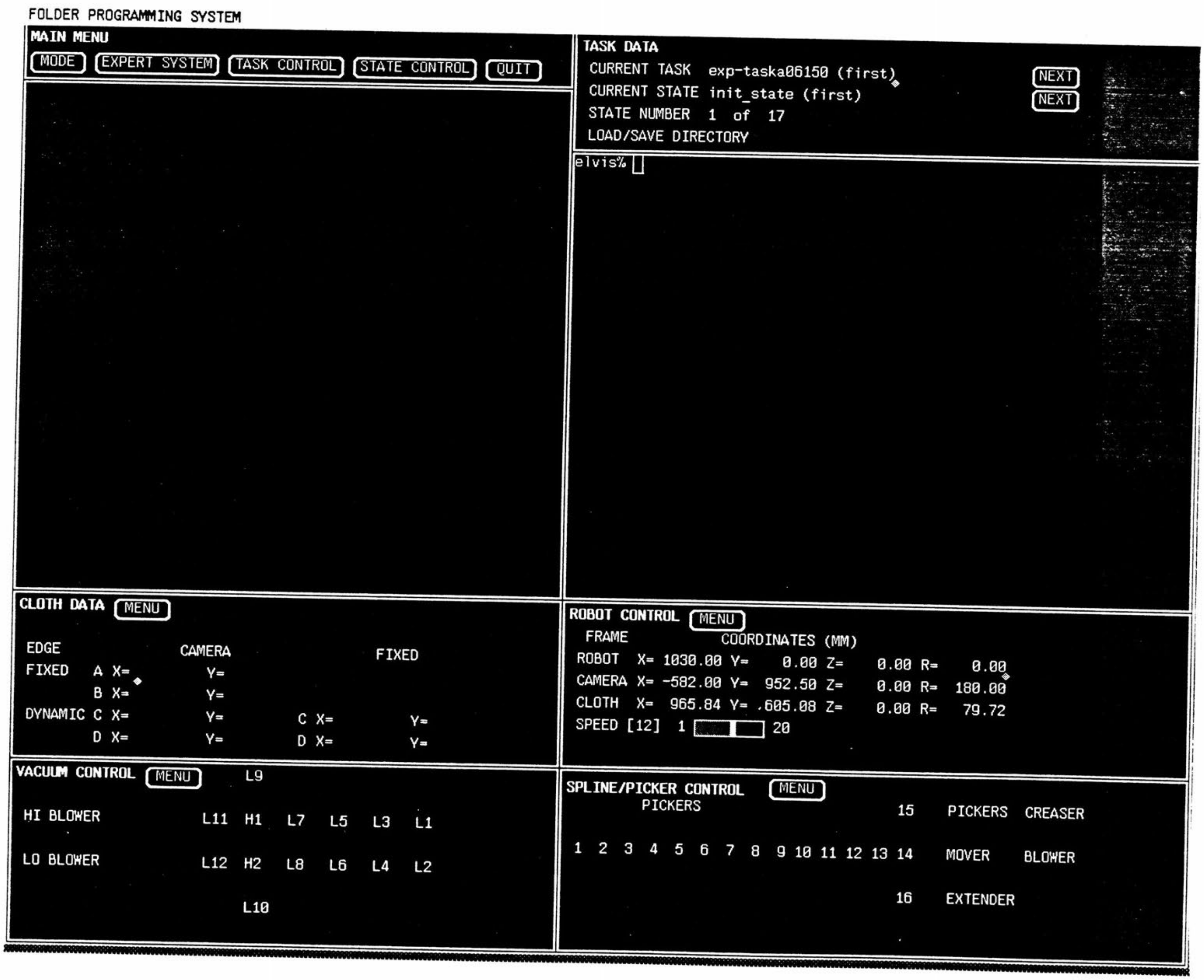


Figure 1-2: The Draper/[TC]^2 folder-sewer machine user interface

The folding expert system allows operational control over the folding station of the Draper/[TC]^2 folder-sewer machine and is based on a uniform architecture capable of describing many robotics applications. Any system which has been defined within the constraints of this architecture may similarly control its robotic applications. This descriptive architecture is based on a unique set of attributes which completely describe the system at any time. Secondly, there is one, and only one, finite set of actions which may may change the state of the system by changing the attributes of the system.

Given a system's initial state and the effect of actions on system attributes, a file of robot actions may be constructed. A task planner takes the initial states, actions, and attributes and constructs the commands required to meet a goal state. Given an initial and final state, a task level generator may search a system's file of robot actions to effect the state transition. The folding expert system uses such methods to translate user input, expressed as states, into folding commands.

1.3.3 The Program Generator

Programmable, automated machinery systems are being rapidly introduced into manufacturing environments today. These complex systems, which are usually composed of several intricate components utilizing widely-varying advanced technologies require programmer/operators to possess technical skills and expertise commensurate with the level of machinery sophistication [Bernardon 87]. But present-day, automated machinery system programming is still an arduous and time-consuming task, even for the most skilled programmer/operators.

Three obstacles impeded successful implementation of programmable, automated machinery systems: lack of training centers and courses for programmer/operators; the phenomenal expense of those few training programs which exist; and the time operators require to become efficient system experts by repeatedly programming and using the equipment. Few skilled and experienced programmer/operators are entering the work force. Compounding the shortage of new labor, there are very few workers who are currently able to operate non-programmable, automated machinery systems who are training to operate programmable, automated machinery systems and gaining the experience mentioned above. The skills of the work force are not keeping pace with the caliber and rate of infusion of the new technology. Clearly, the need for a computer-based and intelligent automated machinery programming system exists.

An automatic program generator was proposed and implemented for Draper's automated machinery system: the Draper/TC^2 folder-sewer machine. The program generator transforms a set of engineer-provided hardware specifications and operational goals into machine-specific code for the system. It is interposed between the software and hardware of the system's expert system/user interface modules and its major workcell components and devices.

Specifically, the program generator translates and communicates instructions from the expert system/user interface modules to the appropriate devices. In addition, the program generator polls the devices it controls, gathers and organizes the data, and transmits it to the expert system/user interface for subsequent interpretation and actions.

Chapter 2

The Draper/[TC]^2 Machine's Vision System

2.1 Vision System Design Considerations

The objective of the Draper/[TC]^2 machine's vision system was to allow the development of "a vision guided, fully-automated folding-sewing device for the construction of suit sleeves." [Wyschogrod 87] The folding station of the machine and the non-rigid nature of the cloth presented several design requirements.

The cloth must be sewed to a certain accuracy, even though the cloth is not cut accurately and repeatably. Accuracy is defined as the degree of alignment of cloth edges. The materials supplied to the Draper/[TC]^2 machine were cut manually. As a result, slight variations due to operator error and pattern wear occurred. The vision system therefore had to supply cloth coordinate data over a range of variation in cloth profiles.

Cloth characteristics entered heavily into vision system design considerations. The edges of some cloth types would fray when cut. Depending upon the degree of fraying, these frays could be mistaken as corners or contours of the cloth by conventional vision systems. Thus, the sensitivity of the vision algorithms developed was required to be adjustable. A set of parameters was therefore developed to address varying degrees of folding accuracy and cloth fraying.

The Draper/[TC]^2 folder-sewer machine was designed to accommodate a variety of cloth sizes and styles, including sleeves, for folding and sewing. The vision system had to be flexible enough to recognize these varying sleeve profiles. This objective was accomplished by coating the folding station table with a retro-reflective material. This material provided sufficient contrast between any shade of cloth and the table surface for vision system detection. Also, the system attained a level of programmability through its parameters mentioned above. These parameters, further explained below, allowed the system to be adjusted for a variety of cloth patterns and sizes.

Economic constraints dictated the use of a single, fixed camera to view the entire folding area. The use of the single, fixed camera reduced additional vision hardware requirements and their associated costs.

Finally, cycle times required the vision algorithms to consider "areas of interest," rather than the camera's entire field of view. These windowing techniques allowed the vision system to analyze only those areas where the cloth piece was expected to be placed. The accuracy of the machine's loading device allowed this data reduction method to be used.

2.2 Vision System Algorithms

The Draper/[TC]^2 sewer-folder machine's vision software accepts a digitized image and a picture request as input. One of two types of output is produced, depending upon the type of picture request specified. For breakpoint requests, locations of breakpoints needed by other parts of the system are returned. For coefficient requests, two breakpoints on the contour are returned, as well as a set of five coefficients that represent a quartic that best fits the contour between those two breakpoints.

The major software modules which implement this input-output relationship are shown in Figure 2-1. The image produced by the CCD camera is digitized. The digitized image is then analyzed and raw edge crossing data is converted to run length table data. A run length table is an efficient, array-like data structure which stores information about white-to-black and black-to-white crossings. This run length table data is then manipulated as described below to determine the required corners and edges. Corner and edge identification are described below on two levels: first, on the level of defining a window within an image; and second, on the level of defining breakpoints and contours within a window.

2.2.1 Windows

A window is defined as the area of an image in which the user is interested. Vision information is drawn from a window in a three-step process: clipping out of an image only that data which is associated with the window in question; finding all of the discontinuities along the contour which lies within the window (thereby creating the run length table mentioned above); and selecting the points of interest from the run length table.

In order to improve efficiency, the run length table of the image of interest is extracted. The contour tracker then collects the points of the image which define its edge and orders them in a clockwise fashion from some reference point. The image is eroded and/or dilated to ''clean it up.'' Dilation and erosion refer to smoothing operations which eliminate "hairs" or "pits" along a contour within a window.

2.2.2 Breakpoints/Corners

Corner detection is accomplished by the contour discontinuity finder (see Figure 2-2). Essentially, the finder considers each point in the run length table and the neighbors which vision system parameters arc_length and min_angle dictate. This software module then computes a ''secant'' to determine whether or not the point at hand be considered a breakpoint (*i.e.*, a corner).

Not all breakpoints need be passed to the system which is attempting to use information from the image, however. Required breakpoints are, therefore, selected by the next software module. Several vision system parameters exist which, when matched against the breakpoints in the run length table, may or may not have to be satisfied in order

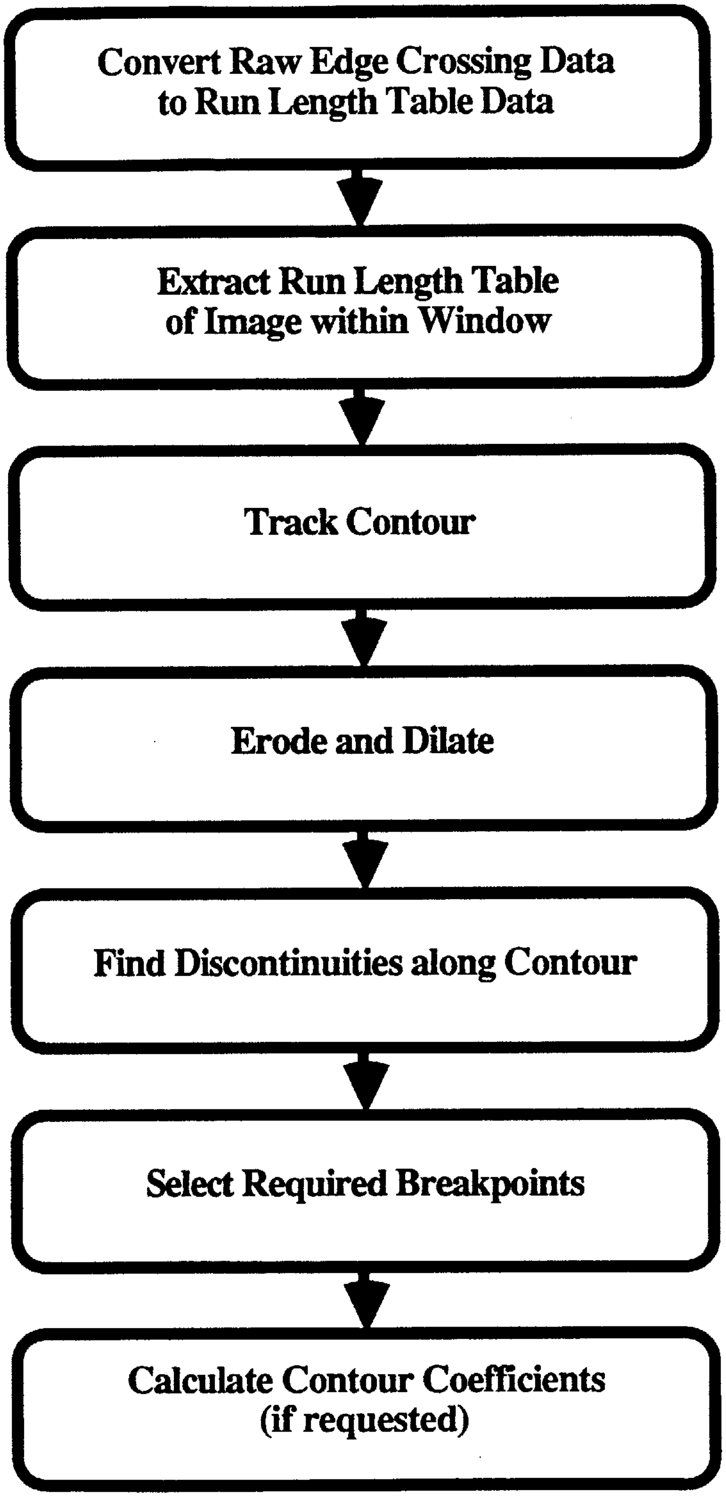
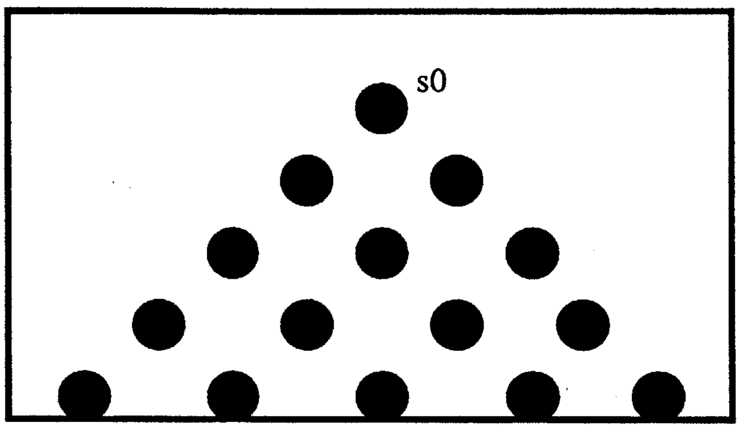


Figure 2-1: Software Modules of the Draper/[TC]^2 Vision System



Orientation of points within a specified distance to the left and to the right of point s0 with respect to the outlined window is determined. Subtraction of the average left orientation from the average right orientation determines a local maxima. If the difference in orientations is greater than the minimum angle specified by the user, the point is declared a candidate breakpoint.

Figure 2-2: Operation of the discontinuity-finding software module for the breakpoint to be passed to the ''real world.'' These criteria are contained in the BP_NODE structure and are explained in subsection 2.3.1 on page 20. Those breakpoints which meet the criteria are passed through the contour coefficients module, if a contour request was received, and then to the user interface.

2.3 Vision System Parameters

Geometry-specific parameters for the Draper/[TC]^2 folder-sewer machine vision system are stored in a statically allocated and initialized parameter tree. By storing the parameters in this fashion, a user may apply vision system algorithms described above to a number of window images by simply setting these parameters to their correct values. The structure of this parameter tree is shown in Figure 2-3 and parameter definitions are given below.

2.3.1 Definitions

The wd_parms_ptr in WD_TABLE points to an object of type WD_PARMS, which is defined below. Arc_length is a measure of resolution along the contour of the geometric breakpoint finding algorithm. Min_angle is used to determine the orientations of neighboring points relative to the current point.

Max_cnt_lgth and min_cnt_lgth determine when dilation and erosion will occur. The system keeps a running count of a contour's length as it is tracked. If the number of points on a contour exceeds a preset maximum, erosion and dilation occur. Select_bp_ptr points to an object of type BP_WIN, which is also defined below.

WD_PARMS is a union pointed to by the wd_parms_ptr field of WD_TABLE. It contains pointers to FIXED_WD and REL_WD structures. Both the FIXED_WD and REL_WD structures contain an absflag field, which is a true boolean if WD_PARMS values are to be accessed via their respective structures.

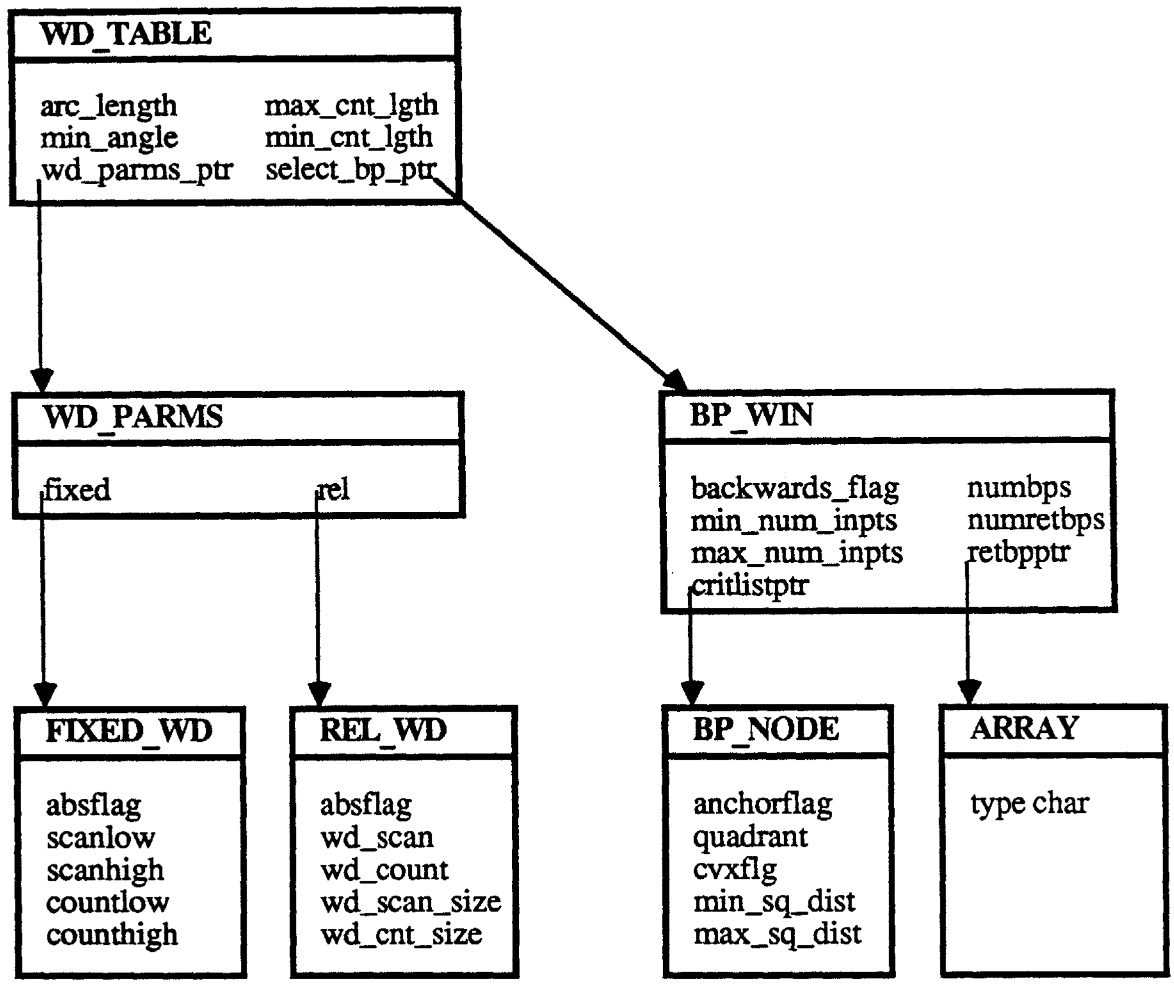


Figure 2-3: Vision system parameter tree

The FIXED_WD structure also contains scanlow and scanhigh fields, which indicate the minimum and maximum number of scan line counts for a window, respectively. Countlow and counthigh similarly give the lower and upper bounds, respectively, of the number of camera line counts for a window.

The REL_WD structure contains four fields in addition to the absflag field. Specifically, wd_scan and wd_count give the location of the window in the sleeve centroid coordinate system. Wd_scan_size and wd_cnt_size fields give the dimensions of the window centered at the point specified by wd_scan and wd_count.

The select_bp_ptr pointer of the WD_TABLE points to the BP_WIN structure. BP_WIN contains the following fields: a backwards flag to determine whether the data stored in a run length table is stored in "forward" or "reverse" order; minimum and maximum number of inputs fields which give the lower and upper bounds, respectively, of geometric breakpoints that the software module will accept; and number of breakpoints and number of breakpoints to return fields which specify the number of points that must be matched and returned, respectively. BP_WIN also contains two pointers: the first to an array of more parameters which contains the criteria for matching of points along a contour and the second to an array of indices for each breakpoint which was found and that is to be returned to other software modules.

The first of the pointers mentioned above references the BP_NODE structure. The BP_NODE structure contains an anchorflag field, which, when its boolean is set true, requires the point which satisfies all of the criteria mentioned below to immediately follow the previous point. This ensures that no "spurious" geometric points can exist between this point and the previous point. This structure also contains a quadrant field, which specifies in which direction from the current breakpoint the next breakpoint candidate must be and a convex/concave field, which specifies whether the next breakpoint must be closer to or farther away from the interior of the object than its two nearest neighbors.

2.4 Vision Algorithms and Parameters

The combination of the vision software modules outlined in Figure 2-1 and the parameters described in the previous section comprise the Draper/[TC]^2 folder-sewer machine's vision system. A link between the vision software and the user interface was proposed which would allow a cloth workpiece to be loaded onto the folder table and folded automatically.

Initially, identification of cloth breakpoints to the folding expert system was pursued. Completing this component would allow preprogrammed suit sleeve profiles to be loaded on the folder table and an image of the sleeve to be captured and analyzed. Identified breakpoints could then be plotted and passed along to other relevant software modules.

Secondly, manual adjustment of two vision parameters was proposed in order to control the number of breakpoints identified and displayed by the software module mentioned above. Two slider bars were proposed to allow for adjustment of these parameters. Reprocessing of the image would then occur with the altered parameters. Subsequent user interaction would also be allowed to achieve the desired number of breakpoints.

Finally, methods of automatic vision parameter selection were proposed. These three levels of implementation are detailed in the chapters which follow.

Chapter 3

Vision System Processing

The series of folding and sewing operations required to produce a suit sleeve were assembled by coordinating the vision system, the IBM 7547 robot, the vacuum table, the banks of transport belts, and the Singer sewing machine. Programming of each of these modules was carried out manually. And the vision system parameters for each of the images required to fabricate the suit sleeve were chosen by an expert user.

This expert user considered sleeve placement on the folding table to specify the area of interest, or window, which the vision algorithms would analyze. Analyzing a smaller section of the entire image allowed the system to process the data at a sufficiently fast rate. In addition to specifying the window boundaries, the expert user also specified an arc length and a minimum angle.

Arc length is a rough measure of the resolution along the contour of the vision system's geometric breakpoint finding algorithm [Wyschogrod1 87]. It is used to find discontinuities in the contour. The minimum angle field is the angle, expressed in degrees, at which two line segments meet at a discontinuity.

The discontinuity-finding algorithm determines candidate breakpoints through a three-step process. First, a point along the contour at hand is chosen and the average orientation of all of the points to the left of the chosen point within the distance specified by arc length is calculated. The average orientation of all of the points to the right of the chosen point within the distance specified by arc lengths is then calculated. If the difference between these two averages is greater than the value specified by minimum angle, then the chosen point is declared a candidate breakpoint.

Additional vision system parameters provide criteria for selection of desired breakpoints. These selection criteria are used in processing the set of candidate breakpoints produced by the discontinuity-finding algorithm. Those candidate breakpoints which matched the criteria associated with the image are allowed to filter through this software module. They are then returned to the system as the breakpoints of the analyzed image.

3.1 Default Vision System Parameters

Arc lengths were generally chosen such that they were never longer than the most closely-space detail which the user desired to detect. In addition, smaller averaging distances increased processing time and made the discontinuity-finding algorithm overly sensitive to frayed cloth edges and other ''noise'' which appeared in the image. Typical arc lengths ranged from 0.35" to 0.58" for the sample sleeve used in this study.

Minimum angle, the other relevant parameter altered for effective vision system processing, was set at 90 degrees for the sample sleeve. When a breakpoint is formed by two curves, rather than two line segments, which meet, the minimum angle is not precisely the angle at which the contours meet. However, an approximation guided by this principle proved successful in analyzing sleeve images.

3.2 Vision System Operation with Default Parameters

The C programming language in the SunView environment of a Sun Microsystems workstation was used to implement these vision algorithms. Using the default vision system parameters which the expert user had pre-programmed, a window was chosen which specified the area of interest for the sample suit sleeve analyzed in this study. This operation is shown schematically in Figure 3-1.

Pixel information from the specified area was then excised and sent to the vision algorithms described above. This information was processed using default arc length and

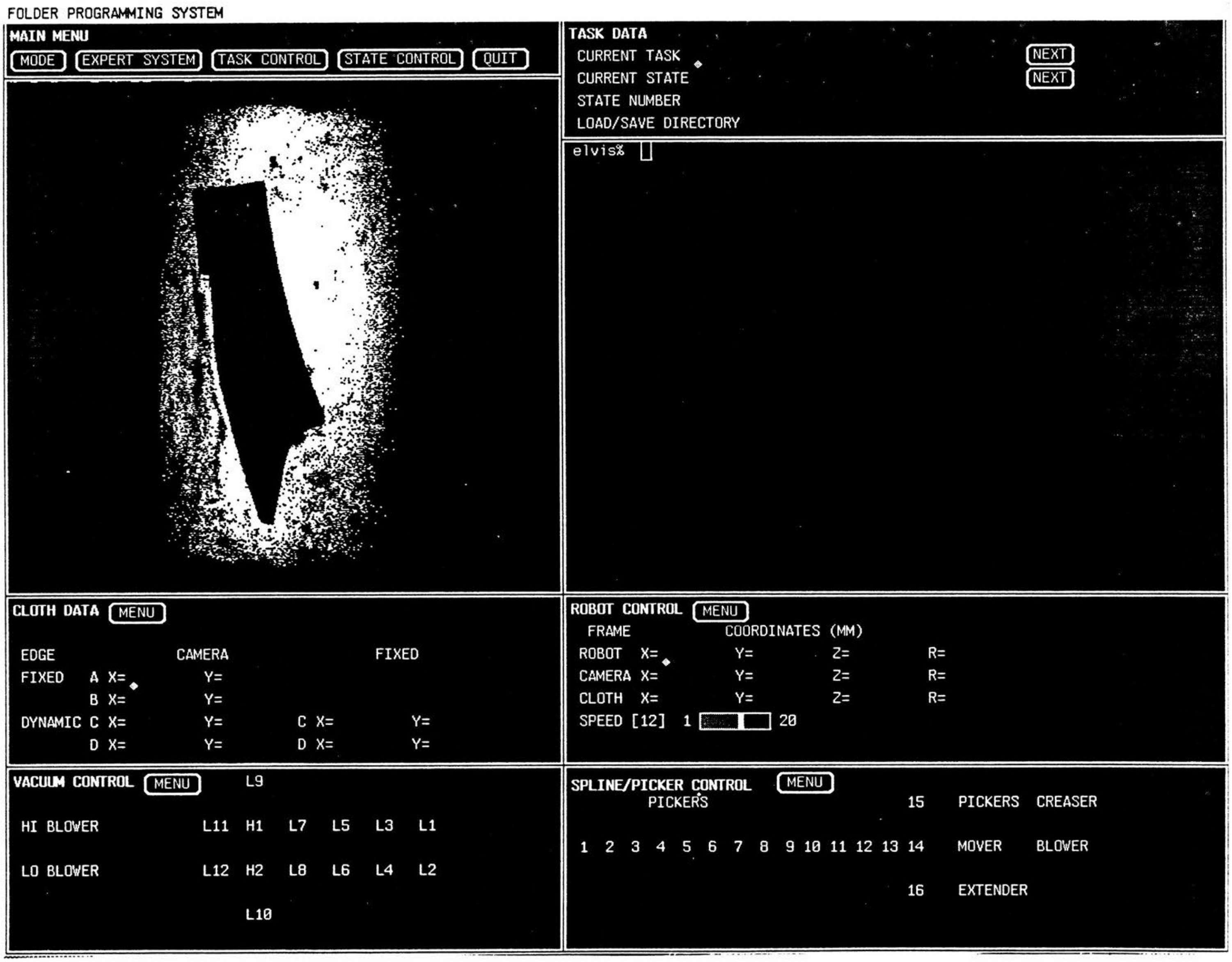


Figure 3-1: Sample sleeve as imaged by vision system

minimum angle parameters for the cloth pattern. The vision software produced an array of integers of desired breakpoints and passed control back to the main routine.

Chapter 4

Vision System Integration

4.1 Addition of the Trapezoidal Cloth Test Piece

The vision system algorithms identify breakpoints necessary to folding operations. Previously, users could identify these corners manually via mouse input to the user interface. This section describes the steps taken to automatically provide the folding expert system with the coordinates of a trapezoidal cloth test piece to be folded.

4.1.1 Vision System Parameter Specification

Addition of the trapezoidal cloth test piece to the folding station's ''repertoire'' required selection of the vision system parameters outlined in section 2.3.1. These parameters were chosen in accordance with the methods outlined in Wyschogrod's *User's Manual for TCI Vision System* [Wyschogrod1 87].

The vision system consults several files to identify the number of pictures and windows within those pictures it recognizes. One picture containing two windows was specified. The first window identified an area encompassing the base of the trapezoid as an area of interest. The second window identified an area bounding the top of the trapezoid as the second area of interest. The trapezoidal test piece, as imaged by the system and displayed on the user interface, is shown in Figure 4-1.

Criteria for properly identifying discontinuities were also specified. Each window contained a single contour along which discontinuities were detected. Both contours are linear and identical cutting techniques were used to form each one. Therefore, identical discontinuity parameters were specified for both windows.

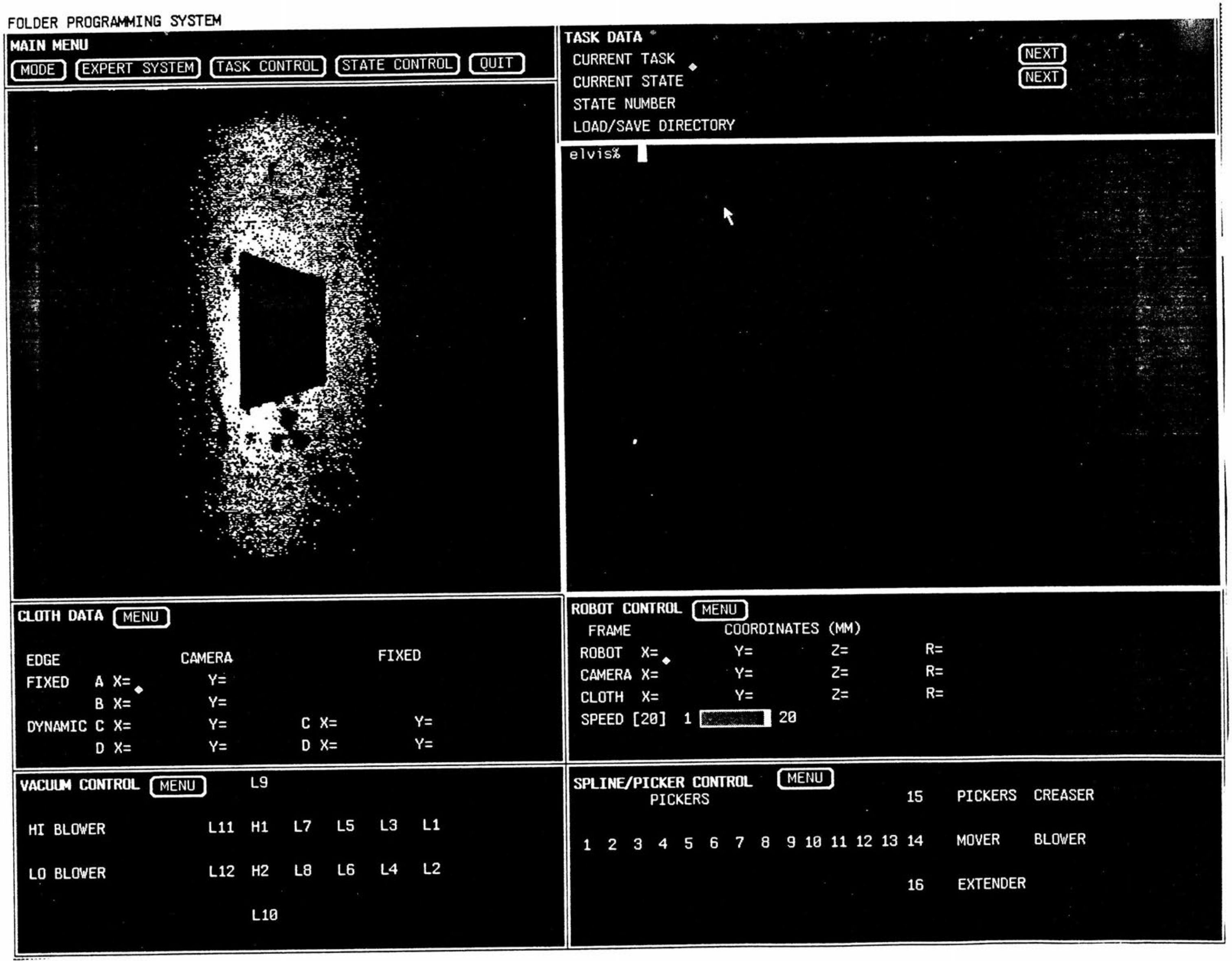


Figure 4-1: Trapezoidal, cloth test piece as imaged by vision system and displayed on user interface

Each window required a set of criteria to discriminate between candidate breakpoints and those breakpoints which were required for subsequent, successful folding operations. Values for these criteria were determined by two means. First, some physical quantities, such as distance from the edge of the test piece to the edge of the folding table, were measured and substituted into equations outlined in Wyschogrod's *User's Manual for TCI Vision System* [Wyschogrod1 87]. Second, rules listed in the manual also determined some of the values of these criteria. Finally, an array indicating the number of desired breakpoints was specified.

4.2 Modifications to Vision System Algorithms

The abridge_list function of the "Select Required Breakpoints" software module (see Figure 2-1) was modified. This modification communicated the required breakpoints the vision system identified to the folding expert system via a vision data file. This file contained a list of double precision, floating point integers which would then allow the folding expert system to generate the required commands to fold the trapezoidal test cloth piece.

Chapter 5

Manual Adjustment of Vision Parameters by User

Vision processing using default vision system parameters proved an acceptable solution for those cloth profiles which conformed to the specifications to which an expert user had programmed the system. However, when defects in workmanship, such as poorly-cut cloth samples, and cloth characteristics, such as excessive fraying, occurred, processing of an image with default vision system parameters either identified too many or too few breakpoints.

In order to identify additional breakpoints or eliminate identification of spurious breakpoints, a method of adjusting the sensitivity of the discontinuity-finding algorithm was proposed. Experimentation proved that the arc length parameter was the most effective candidate parameter to alter in order to adjust this sensitivity. In addition, the minimum angle parameter could be adjusted so as to increase or decrease the strictness of breakpoint identification.

5.1 Arc Length Parameter Adjustment

Arc length adjustment directly affected the sensitivity of the discontinuity-finding algorithm. Given a processed image with too few breakpoints identified, decreasing the arc length would produce an image with additional breakpoints plotted. Decreasing arc length increased the system's sensitivity to detailed features of the sleeve, but also increased its sensitivity to noise, such as frays, along the contour. In general, however, an acceptable value for the arc length parameter could be selected in order to identify the correct number of desired breakpoints for subsequent processing by the folder-sewer machine components.

Increasing arc length decreased the sensitivity of the discontinuity-finding algorithm.

Details and noise which were smaller than the arc length were effectively ignored, thereby masking out most unwanted details. However, those desired breakpoints which were identifiable only through details smaller than the arc length specified by the user were masked out of the breakpoint identification process.

5.2 Minimum Angle Parameter Adjustment

A partial solution to the problem of undesired breakpoint masking was adjustment of the minimum angle parameter. The minimum angle, defined as the angle at which two line segments of arc length met, was able to affect identification of breakpoints which the arc length parameter sometimes masked out. Those breakpoints which occurred at the edges of windows were sometimes identified as candidate breakpoints and sometimes masked by the arc length parameter. Depending upon the angle at which the segment or contour intersected the window's edge, the minimum angle parameter served to identify some of these edge points as candidate breakpoints.

Chapter 6

Automatic Selection of Vision Parameters

Automatic selection of vision parameters is the objective of this study. Given the highly-automated nature of the folding operation and operation of the Draper/[TC]^2 folder-sewer machine, parameter selection for the vision system should be automated, as well. There are many methods for automating vision system parameter selection. The solutions proposed below fall into four categories: a statistical history of cases; individual case histories; general rules for parameter selection; and a combination of two or more of these methods.

6.1 Proposed Methods of Automated Vision System Parameter Selection

Various sleeve shapes and cloth types have been folded and sewed on the Draper/[TC]^2 folder-sewer machine. A comprehensive manual or automatic cataloging of the parameter data for all of the sleeve shapes and cloth types would establish a set of case histories. These case histories could then be statistically analyzed, again, manually or automatically, in order to determine any commonalities among the cases. Information gained from this data collection and analysis could then be consulted by novice users and used as an aid by intermediate and expert users as they program vision system parameters.

A second solution to automated vision system parameter selection may be the cataloging of individual, rather than simply statistical summaries of, case histories. An efficient organizational scheme for the memory to store these individual case histories must first be devised. But with this significant technical challenge resolved, search methods may be used to match appropriate vision system parameters to situations presented to the vision system for analysis.

Mathematical formalization of Wyschogrod's vision system parameter selection rules (see [Wyschogrod1 87]) is another method of automating parameter selection. This ''rule-based'' approach would eliminate the need to store the parameter tree mentioned in section 2.3. However, this technique might increase the time required to process the vision algorithms as they are currently implemented with hard-coded values stored in a parameter tree. Also, rule-based vision algorithms might fail when presented with a situation outside of their domain. Limiting the applicable domain, however, could solve this problem.

Another solution to automated vision system parameter selection is the development of a database or expert system. The first step to implementing this solution is capturing the knowledge of an expert vision system user. This solution essentially formalizes Wyschogrod's rules, mentioned above, syntactically. A knowledge base may then be established and accessed. Whether an array-like structure or a full-fledged database is sufficient for the storage and retrieval of this information is sufficient is questionable. The adequacy of these methods would dictate whether an expert system would be needed in order to enable or enhance system performance.

Finally, combining case-based and rule-based approaches for vision system parameter selection would be an ideal, but by far the most complex, solution to the problem of automated vision system parameter selection. The case-based elements of the system would include parameter data collected and analyzed manually or automatically and/or parameter data organized in an array, database, or expert system. The rule-based elements of the system would include Wyschogrod's vision system parameter selection rules.

The combination of these case-based and rule-based elements would allow the system maximum flexibility in selecting vision system parameters. If the system were presented with a situation which it had been pre-programmed to accommodate or ''seen'' previously, a scan of the statistical or individual case histories would provide a convenient method of supplying the correct vision system parameters. If a situation which fell within the system's predefined domain was encountered for which no statistical or individual case history existed, the rule base could be invoked to devise a solution for the required vision system parameters. The parameters generated for this situation and the situation's success or failure would then serve as another case history. This case history could then be added to the case-based memory on a statistical and individual basis. Thus, a system combining the experience of an expert user with the fundamental rules this expert works from would prove most flexible in automatically selecting vision system parameters.

Chapter 7

Conclusions

Two vision system parameters, arc length and minimum angle, were defined and their effects on vision system performance were investigated. Increasing the value of the arc length parameter made the vision system discontinuity-finding algorithm less sensitive to "noise" in the picture, such as frayed edges along the edge of a sleeve. Decreasing the value of the arc length parameter increased the sensitivity and processing time of the discontinuity-finding algorithm. Adjusting the minimum angle parameter was seen to have the most effect on identifying breakpoints near the edge of the area of interest.

Vision parameters for a trapezoidal cloth test piece were calculated. These parameters were then programmed and used by vision system algorithms to identify required breakpoints (or corners) for the test piece. The coordinates of these breakpoints were then provided to the system's folding expert system for subsequent folding of the test piece.

A method for manual adjustment of these parameters for greater operational flexibility of the vision system was proposed. Finally, a variety of case-based and rule-based methods for automatic vision parameter selection were also proposed.

References

[Bernardon 86] Bernardon, Edward.
Robots in the Apparel Industry.
Technical Report, The Charles Stark Draper Laboratory, October, 1986.

[Bernardon 87] Bernardon, Edward.
Proposal for an Intelligent Automated Machinery Programming System.
Technical Report, The Charles Stark Draper Laboratory, October, 1987.

[Foley 87] Foley, Michael.
The Expert User Interface.
Technical Report, The Charles Stark Draper Laboratory, November, 1987.

[Wyschogrod 87] Wyschogrod, Daniel and Edward Bernardon.
Vision Guided Manipulation of Non-rigid Parts: A Case Study.
Technical Report, SPARTA, Inc. and The Charles Stark Draper Laboratory, November, 1987.

[Wyschogrod1 87]
Wyschogrod, Daniel.
User's Manual for TCI Vision System
SPARTA, Inc., 1987.

[Wyschogrod2 87]
Wyschogrod, Daniel.
Basic Maintenance Manual for TCI Vision System
SPARTA, Inc., 1987.

[Wyschogrod3 87]
Wyschogrod, Daniel.
Algorithmic Outline of TCI Vision System.
Technical Report, SPARTA, Inc., March, 1987.