

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave Blank)		2. REPORT DATE 5 Feb 98	3. REPORT TYPE AND DATES COVERED Final Report: 6 Aug 97 - 5 Feb 98	
4. TITLE AND SUBTITLE 3D Scan Systems Integration			5. FUNDING NUMBERS	
6. AUTHORS NURRE, Joseph H. ADDLEMAN, Stephen				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Ohio University Office of Research & Sponsored Prog. 105 Research & Technology Center Athens, OH 45701			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Defense Logistics Agency MMPRT Room 3135 8725 John J. Kingman Road, #2533 Fort Belvoir, VA 22060-6221			10. SPONSORING / MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES			19980623 126	
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; Distribution unlimited			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) The DDFG-T2-P3 Short Term Project conducted at Ohio University in collaboration with Cyberware Inc., developed numerous algorithms for three dimensional data analysis and processing. The algorithms focused on two areas. First, data analysis resulted in software that can extract apparel measurements from scan data. The measurements are taken automatically from scan subjects wearing a minimal set of intentionally placed fiducials. Second, processing algorithms assist in the study of scanner capture resolution. Data artifacts inherent in the technology are simulated. Methods to reduce certain artifacts have been developed. Theoretical descriptions of all algorithms are presented in this report. Source code was released to Cyberware Inc. for inclusion in their CyScan software.				
14. SUBJECT TERMS 3D Scanning, segmentation, apparel sizing			15. NUMBER OF PAGES 60	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT Unlimited	

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-1
298-102

DTIC QUALITY INSPECTED 1

DLA-ARN Final Report
for
US Defense Logistics Agency

on

DDFG-T2/P3: 3D SCAN SYSTEMS INTEGRATION

Contract Number	SPO100-95-D-1014	
Contractor	Ohio University	
Delivery Order #	0001	
Delivery Order Title	3D Scan Systems Integration	
CDRL #	A005	
CDRL Title	Technical Report, Final Report	
Reporting Period	August 6, 1996 - February 5, 1998	
Report Date	February 9, 1998	
Name of PI's	Joseph H. Nurre & Stephen Addleman	
E-mail	nurre@bobcat.ent.ohiou.edu	
Phone	614/593-0103	
Fax	614/593-0007	
Address	School of Electrical Engineering & Computer Science Ohio University Athens, Ohio 45701	Cyberware Inc. 2110 Del Monte Avenue Monterey, California 93940-3797

Submitted to:

Apparel Research Network
Defense Logistic Agency
2800 South 20th Street
Philadelphia, Pennsylvania 19101-8419

Table of Contents

Executive Summary.....	1
1. INTRODUCTION.....	2
1.1 Purpose, scope and objectives.....	2
1.2 Scope of work -- Modifications from original proposal.....	2
1.3 Report organization.....	3
2. DATA ANALYSIS.....	4
2.1 Data Analysis - Software development objectives.....	4
2.2 Data Analysis - Software achievements.....	5
2.2.a Orient data.....	6
A. Identify top of subject.....	7
B. Identify an approximate height level of hands.....	7
C. Identify direction and location subject is standing.....	8
2.2.b Segment data.....	10
A. Identify head data.....	11
B. Identify leg data.....	12
C. Identify arm data.....	18
2.2.c Localize.....	25
2.2.d Find landmark.....	25
2.2.e Measure.....	29
3. SCAN COVERAGE.....	30
3.1 Scan Coverage - Software development objectives.....	30
3.2 Scan Coverage - Software achievements.....	30
3.2.a Scanner configuration review.....	31
3.2.b Simulator system overview.....	36
A. Generic mode menus.....	38
B. Edit scanner mode.....	39
C. Edit track mode.....	39
D. Edit camera mode.....	40
3.2.c Using SimScan.....	40
3.2.d Software design.....	43
3.2.e Operational examples.....	45
A. Two head scanner verses four head scanner.....	47
3.3 Additional software tools.....	48
3.3.a Scan Data Analysis Tool (SDAT)	48
3.3.b Curl ease.....	51
4. DISCUSSION.....	56

DDFG-T2/P3: 3D SCAN SYSTEMS INTEGRATION

Executive Summary

The DDFG-T2-P3 Short Term Project conducted at Ohio University in collaboration with Cyberware Inc., developed numerous algorithms for three dimensional data analysis and processing. The algorithms focused on two areas. First, data analysis resulted in software that can extract apparel measurements from scan data. The measurements are taken automatically from scan subjects wearing a minimal set of intentionally placed fiducials. Second, processing algorithms assist in the study of scanner capture resolution. Data artifacts inherent in the technology are simulated. Methods to reduce certain artifacts have been developed. Theoretical descriptions of all algorithms are presented in this report. Source code was released to Cyberware Inc. for inclusion in their CyScan software.

1. INTRODUCTION

1.1 Purpose, scope and objectives

The DDFG-T2-P3 Short Term Project conducted at Ohio University in collaboration with Cyberware Inc., quantified and analyzed position data obtained from the Cyberware WB4 scanning unit. Quantifying the errors means evaluating measurement resolution and limiting its impact on apparel sizing. Analysis means classifying specific points as being useful for obtaining apparel measurements. A human scan is rich with data but poor with information. The scan points must be identified and labeled with respect to human body features. Extraneous noise points should be eliminated. This data is then available to other measurement algorithms for determining size.

1.2 Scope of work -- Modifications from original proposal

The originally proposed goal of this project was limited to quantifying errors of the full body scanner and evaluating their impact. At the time the proposal was submitted, the Defense Logistic Agency's Apparel Research Network had a Cyberware two headed scanner available for apparel sizing. Before project funds were allocated, however, the scanner was upgraded to four heads. Because of this and other evolution in the technology, it only made sense to change and broaden the scope of the proposal.

Task changes for the funds were documented in Interim Progress Reports. To summarize the changes by task categories:

- Body Fixturing: Research to determine the effects of body sway on scanning was canceled within three months of the start date of the contract. No milestones were reached.
- Catastrophic Measurement Errors: Research to document unexpected catastrophic errors was canceled within three months of the start date of the contract. One milestone was reached and is shown in Appendix A.
- Data Analysis: Research to identify and classify data points for measurement was added to the contract three months from the start date. Results of this work are discussed in this report.

- Scan Coverage: Research to determine hardware resolution was modified. Emphasis was placed on correcting data artifacts, such as curls in the data. Results of this work are discussed in this report.

The scope of the project changed from an emphasis on hardware and subject related studies to a software development project. Ohio University had previously done similar work for the U.S. Air Force's Computerized Anthropometric Research and Design Laboratory, and hence was prepared to meet the challenges of this demanding assignment.

1.3 Report organization

This document is actually a combination of two reports. First, the Data Analysis Task, that is the task related to the analysis and identification of measurement features in the scan data is reported on. Second, the Scan Coverage Task, that is the task concerned with quantifying and eliminating data artifacts is reported on. Finally, conclusions and recommendations are presented in the final section.

2. DATA ANALYSIS

2.1 Data Analysis - Software development objectives

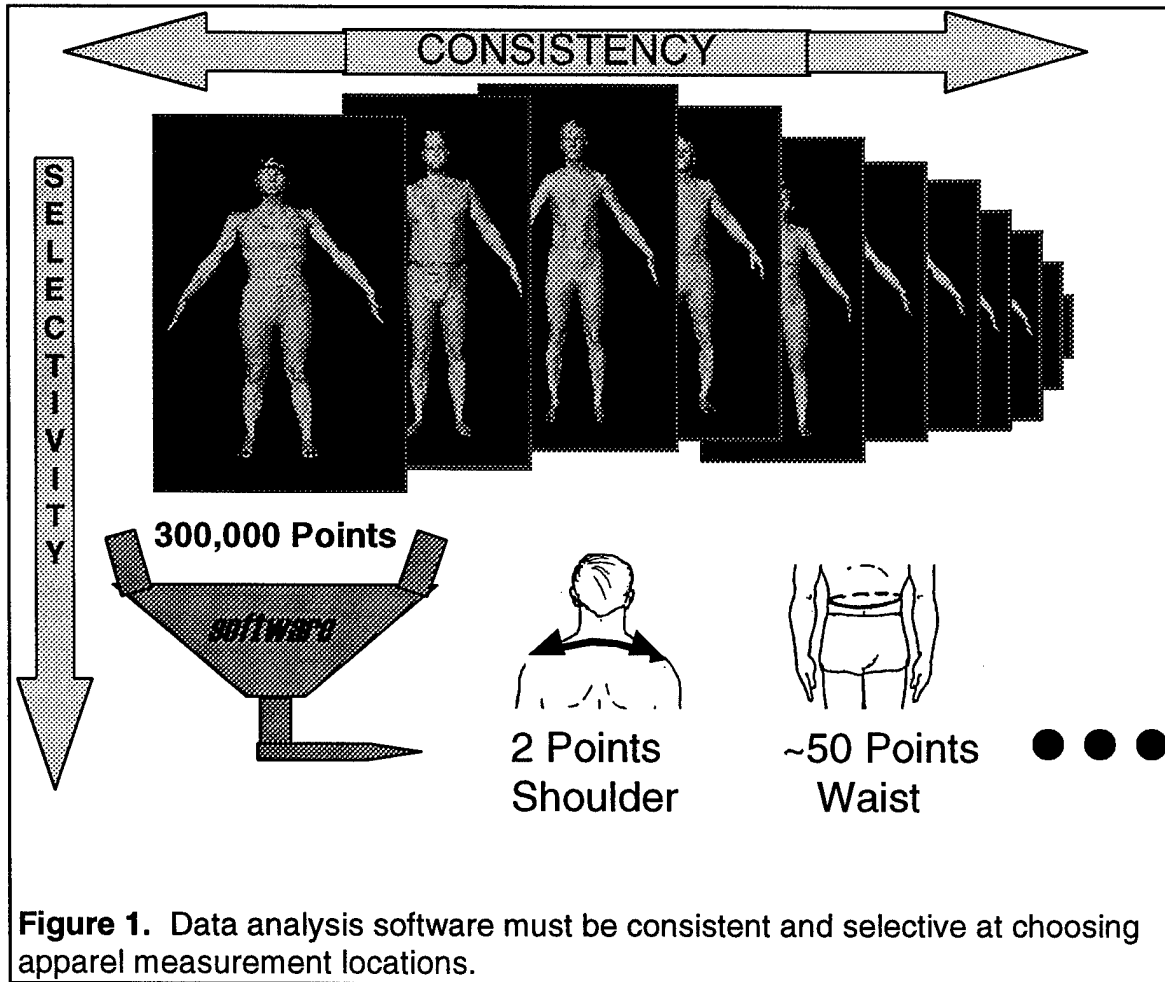


Figure 1. Data analysis software must be consistent and selective at choosing apparel measurement locations.

The Cyberware Whole Body Scanner captures over 300,000 three-dimensional data points from a human subject. The hardware can scan both humans or inanimate objects. The format of the data coming from the scanner is exactly the same for either case. It is the software that must derive meaning from the data.

There are two primary challenges to developing software to work with human scan data. These are shown graphically in Figure 1. First we must select among the 300,000 data points available to find only a specific few points that are important for apparel measurement. For

example, only two points are needed to measure across the subject's shoulders. A waist measurement may require as few as 50 scan data points. If we consider all the information currently required for traditional apparel measurements, only a small percentage of the scan data points will be used for actual sizing calculations. The challenge is to develop automated methods that analyze the hundreds of thousands of data points and use them to zoom in on the required measurement points.

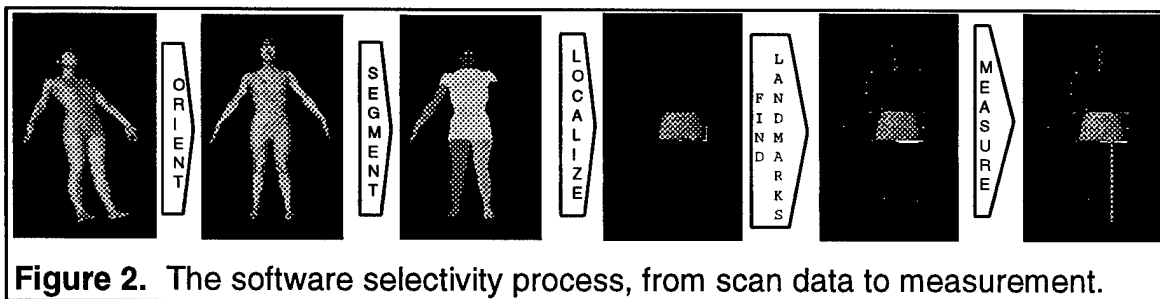
A second challenge of the scan data is the numerous body shapes that may be captured by a scanner. Developing software to identify measurements on a select few subjects is not sufficient. The software must be able to consistently find measuring points, despite the wide variety of shapes that humans come in.

To summarize, all software methods discussed in the first half of this report must meet the two challenges mentioned above. The method must advance the goal of selecting among the data, and the method must be robust with respect to human shape. In fact, the development process itself addressed these two concerns separately. First a software method was tested on a few select subjects. Once, it was shown to work effectively on these subjects, it was then tested on all subjects in the data base. Such a test for robustness usually resulted in the algorithm being revisited, refined and sometimes even scrapped. Although the subject data base continues to grow, methods presented below have proven to be robust.

2.2 Data Analysis - Software achievements

Achieving the goal of selectivity among the scan data is accomplished through several steps. Each step builds on information from the previous step. Figure 2 shows a step by step process used by the software, at the time of this report. First, the front of the subject is determined by fitting an ellipse to the data. Second, the subject is segmented. Segmentation is performed using computational geometry and imaging algorithms. Third, with information on body segments and orientation, data selection can be limited to a specific area. Finally, a landmark (a white sticker intentionally placed on the subject) is located to indicate the point of measurement. Landmarks are a necessary short cut for testing measurements and understanding

the human geometry. In the future, landmarks will be eliminated as research discovers techniques for finding apparel measurements without this short cut.



In the sections below, each of the steps in Figure 2 is described. The software code for the implementations are integrated into a software package named CyScan. Cyberware Inc. is responsible for the infrastructure and the “look and feel” of the CyScan software. Requests for source code should be made to Cyberware. Code contributed to CyScan was inspected for compliance to Cyberware coding standards and quality insurance.

The first two steps, orientation and segmentation, were previously developed under a contract with the U.S. Air Force’s Computerized Anthropometric Research and Design Laboratory. The orientation code has been revised significantly. The segmentation code developed under this contract is a departure from the approach used with the Air Force contract. Specifically, the innerHull algorithms described in Nurre¹ were abandoned and the Cusp locator is newly developed.

2.2.a Orient data

The first step in the information gathering process is to determine the location and direction that the subject was standing at in, at scan time. This was accomplished using the three step process listed below:

- A. *Identify top of subject*
- B. *Identify an approximate height level of hands*
- C. *Identify direction and location subject is standing*

Each one of these steps will now be explained.

A. Identify top of subject

The Cyberware whole body scanner collects data from the subject along horizontal slices, every two millimeters. The scanner starts at the top of the scan space and proceeds down. To find the top of the subject, however, slices are inspected starting at the bottom of the scan space. The software starts with the feet and moves up to the head. Each slice is fitted with an external convex hull. Algorithms that find convex hulls are available in many textbooks on computational geometry. The implementation used in CyScan is the Gift Wrap algorithm presented in Sedgewick.² The function name given to the algorithm is `outerHull`. A convex hull simulates the shape a tape measure makes when it is wrapped around a subject. A tape measure remains convex, only lying on the external most points of a subject. A convex hull is fitted to each slice, and its distance is measured. Hence, an approximate circumference measurement is calculated for every 2mm slice of the subject. When the circumference becomes smaller than a `HUMANLY_POSSIBLE_MINIMUM` variable, the top of the subject has been reached. A minimum variable is used rather than zero because single points, capture from hair strands, have a perimeter of zero.

B. Identify an approximate height level of hands

Subjects being scanned are placed in the standard Anthropometric pose. In this pose, the subject's hands are extended out from the body. A convex hull being fitted to horizontal slices will be largest when it wraps around the hands and torso. While searching for the top of the head, the largest convex hull circumference is determined. The slice at which this occurs is recorded as the height of the hand. The height is only an approximation, and is refined later in the segmentation process.

C. Identify direction and location subject is standing

Finding the location and direction of the subject is accomplished by calculating moments of the data. Moment algorithms are a common tool in computer image processing. The technique is described in standard textbooks such as Horn.³ The moments are calculated by assuming each data point has an equal finite mass. This approximation to the true geometric moment has worked sufficiently well to date. More precise geometric algorithms can be developed if needed.

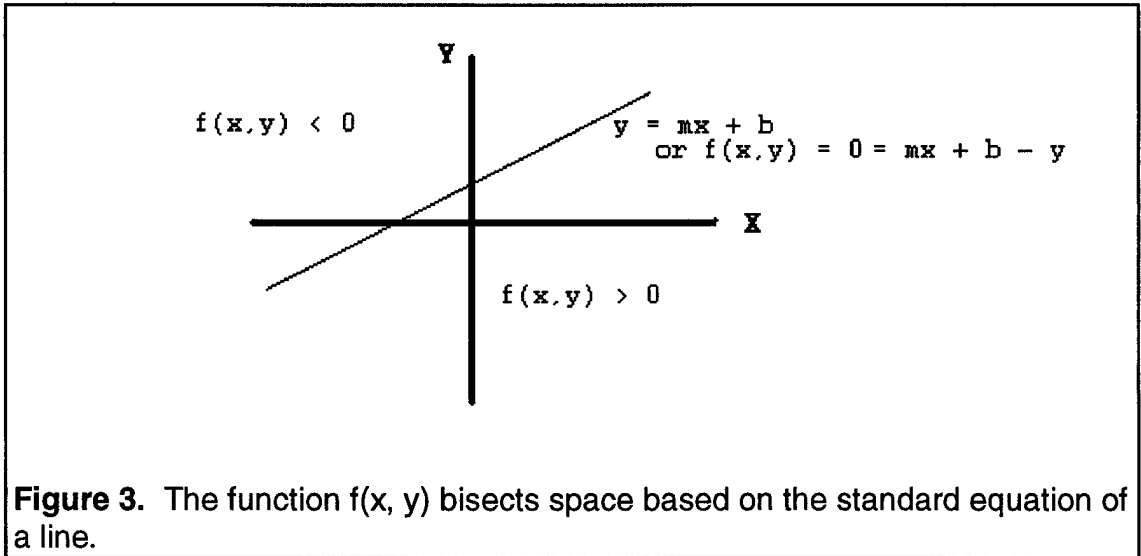
The first moment of the slice data is the center of the points. The second moment of the slice data is equivalent to fitting an ellipse to the data. Slices in the leg area are used to determine position and orientation of the subject. Originally, slices in the upper body were used but it was found that subjects naturally keep legs aligned with their torso. In the upper body, a subject may hold one arm more forward than the other, changing the major axis of the ellipse.

The software method for determining direction starts with the leg slices. Specifically, a data slice one quarter of the way to the approximated hand height is fitted with an ellipse. The software calculates the direction of the major and minor axes. Several additional slices above and below this slice are also fitted to create an average.

The subject's front vector is parallel to the minor axis of the ellipse. The back vector would be parallel too. To determine the front vector, the center of the leg's slice is compared with the center of a feet slice. The center of the feet slice will point to the front. Assume the ellipse's major axis is expressed as the line, $y = mx + b$. The equation of a line can be changed into a function of two variables that bisects the x and y coordinate space. For the major axis line, the function would be:

$$f(x,y) = mx + b - y \quad (1)$$

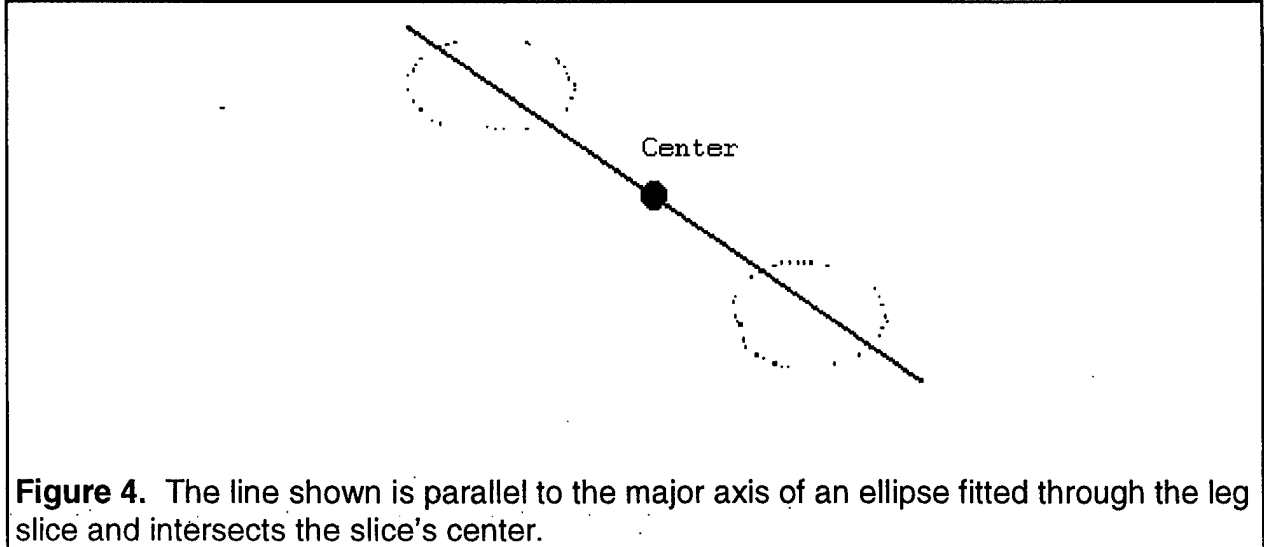
When $f(x,y) = 0$, for equation (1), coordinate points fall on the $y = mx + b$ line. The function then divides the coordinate axis into a positive and negative region, as shown in Figure 3.



The well known variable 'm', slope, presents a problem for computer programs. When the line is vertical, 'm' equals infinity. Let $m = dy/dx$. Hence,

$$f(x,y) = dy*x + dx*b - dx*y \tag{2}$$

where dx and dy represent a slope parallel to the ellipse's major axis. A value of 'b' is chosen such that $f(x,y) = 0$ for the center point of a leg slice. This line is shown in Figure 4. The feet center will fall on the front side of this line, as shown in Figure 5.



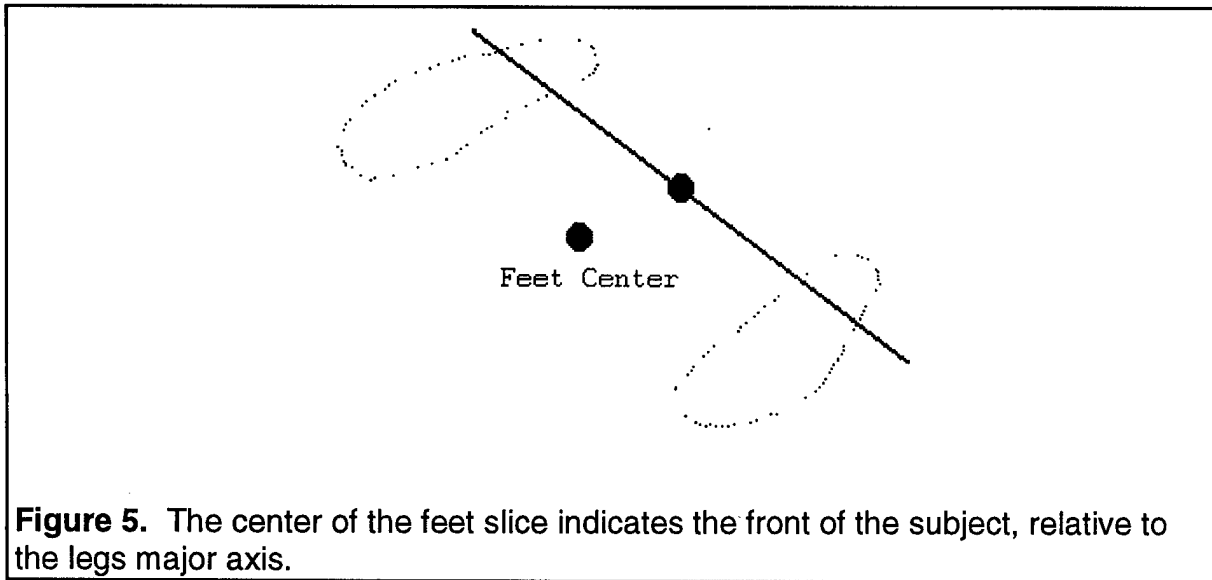


Figure 5. The center of the feet slice indicates the front of the subject, relative to the legs major axis.

Hence, the front vector is assigned in one of the two possible directions, perpendicular to the ellipse's major axis. The feet center point and this assumed front vector should generate the same sign from equation (2). If the signs are different, the assumed direction of the front vector is reversed. The orientation of the subject has now been determined.

2.2.b Segment data

The oriented data is now segmented into extremities. First the head data is identified. The arms are then separated from the trunk. Finally, the legs are separated and the start point of the torso is determined. The order is important. Arms are separated first because, depending on the subject, arms may or may not extend down to the legs. Determining the arm data points first allows them to be ignored when separating the legs, regardless of the subject's arm length. Arms and legs can be further identified as left or right, based on the front vector. The methods will be discussed out of order from actual implementation. Specifically:

- A. *Identify head data*
- B. *Identify leg data*
- C. *Identify arm data*

This presentation order was chosen because the arm segmentation process is a more sophisticated implementation of the leg segmentation process. Hence, understanding leg segmentation is the first step towards understanding the arm separation method.

A. Identify head data

Head data is defined as all points above a determined height. To locate the starting height of the head data, slices are searched for the corner where the neck leaves the shoulder. This is done by analyzing a silhouette of the data. Shown in Figure 6, an external convex hull has been fitted to a silhouette of a scan subject. The greatest perpendicular distance from a point on the silhouette, to the convex hull between the head and a hand, is the corner being searched for.

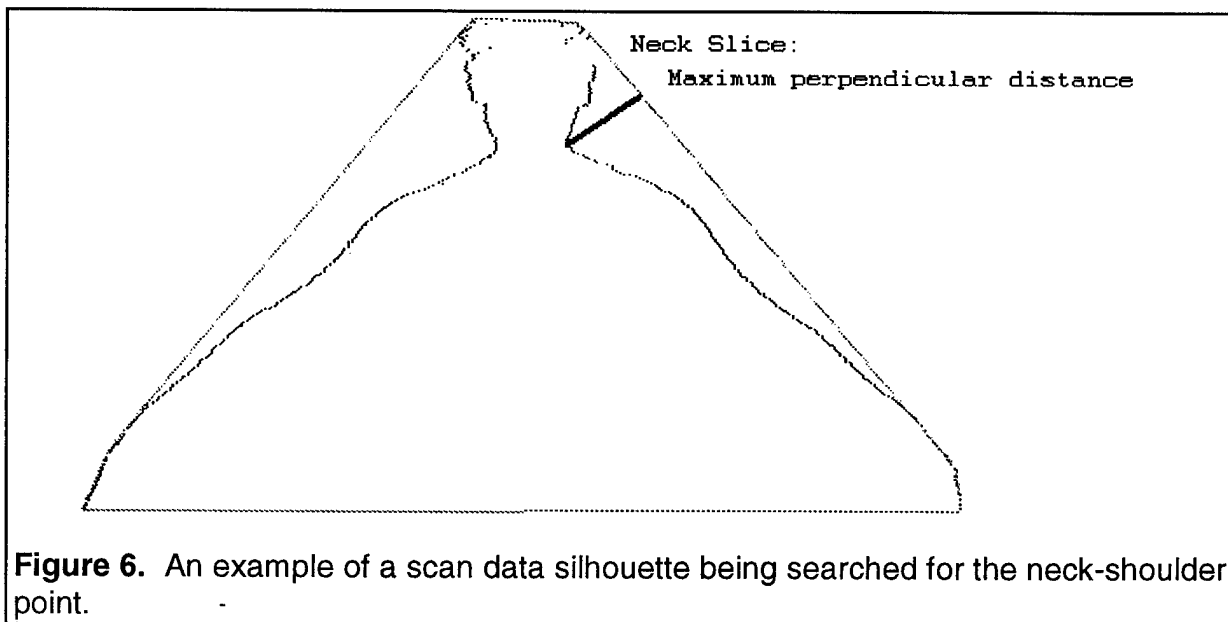


Figure 6. An example of a scan data silhouette being searched for the neck-shoulder point.

Creating a silhouette is accomplished with a profile function. The profile function creates a shadow of the data points in the direction of any specified vector. In this case, the front vector is given to the Profile function which performs a mathematical projection of all points in that direction. Lets assume the data points have been projected onto the x, y coordinates, as shown in Figure 7. The profile function finds the silhouette by determining the minimum and maximum of each projected slice.

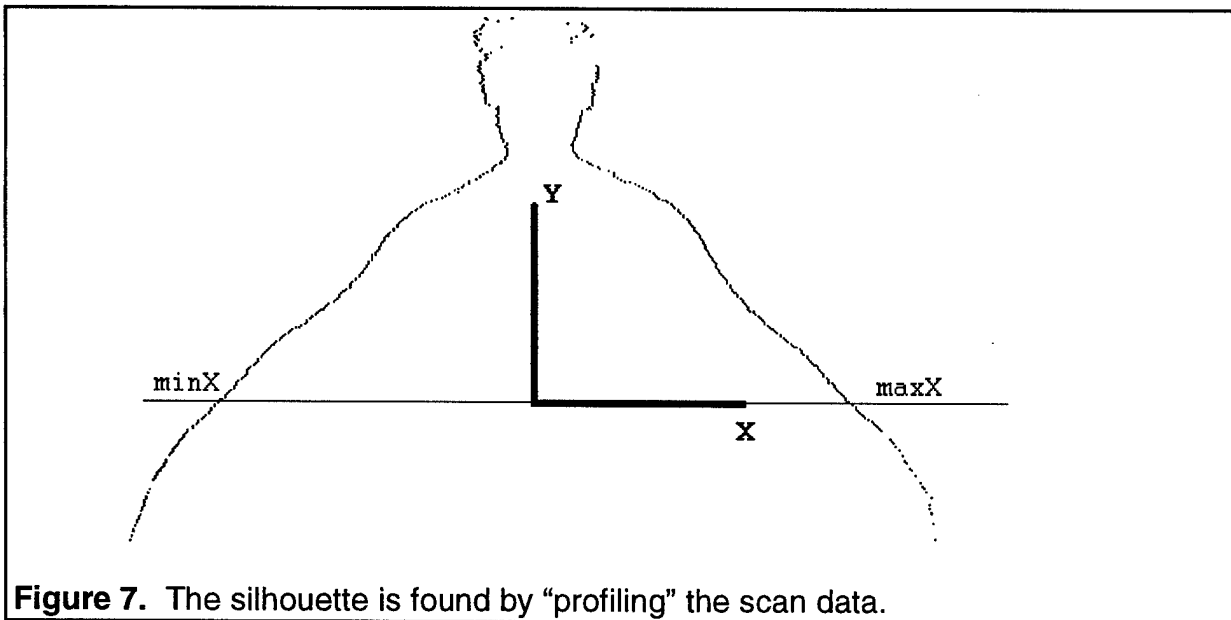


Figure 7. The silhouette is found by “profiling” the scan data.

As described earlier, a convex hull is fitted to the silhouette using the `outerHull` function. One of the characteristics of `outerHull` is to return the data points of the external convex hull in counter clockwise order. The line segment of interest on the hull can be found by searching for the line with an increasing x and decreasing y slope. Only lines in the upper right quadrant of a convex hull will have such a slope.

Once the hull segment has been identified, all points of the silhouette are rotated such that the x coordinate falls on the line segment and the y coordinate is the distance from this hull line. Determining the corner of the neck is simply a matter of searching for the maximum y coordinate along the line segment.

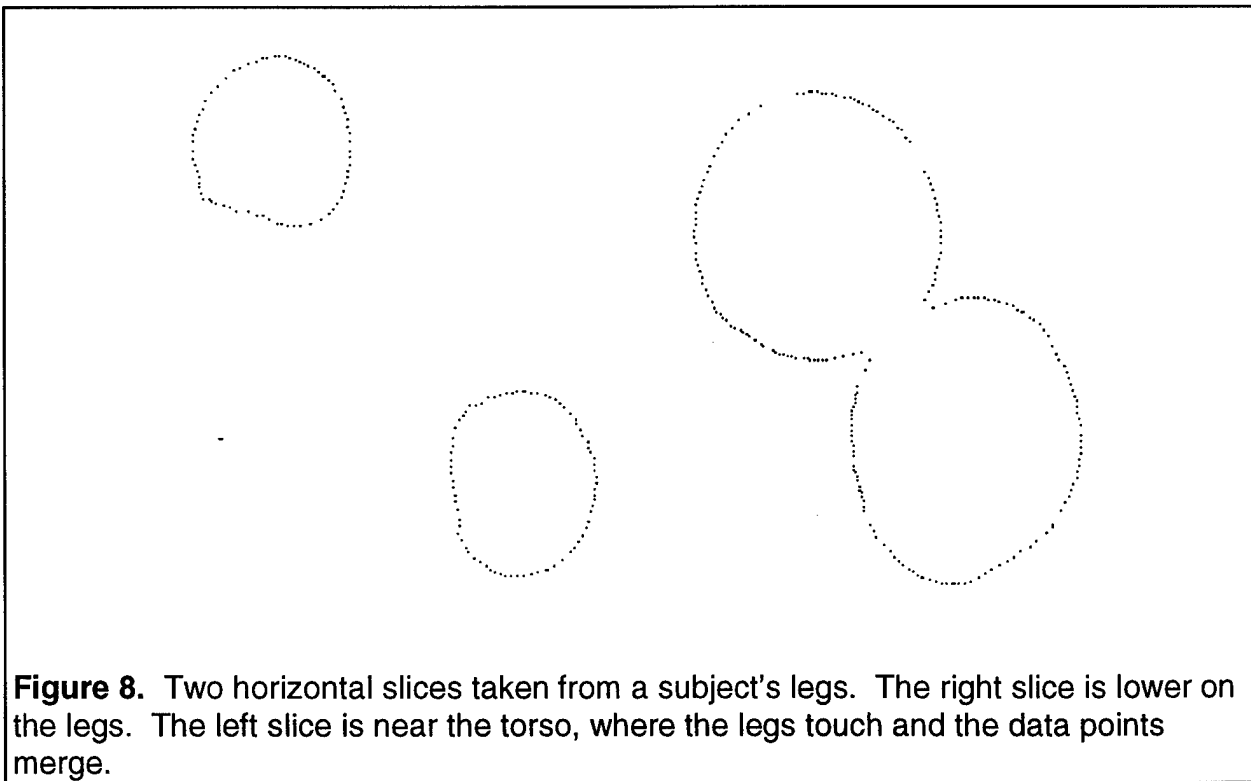
B. Identify leg data

The legs have to be between the approximated height for the hands and the bottom slice of the scan. A slice one quarter of the way up from the bottom towards the hand is chosen as the starting slice for leg segmentation. At this height, there will be good separation between the legs for a subject standing in the standard anthropometric position. This slice data is projected parallel to the front vector. Slice data is essentially two dimensional since all points have the same height.

After projection, the slice data is one dimensional. A large gap in the one dimensional data is caused by the separation of the legs. This gap are easily located algorithmically, and a bisector within the gap, parallel to the front vector, is generated. Hence, the data is classified by the bisector as left or right leg.

The segmentation process must now proceed upward and downward from this slice. The downward direction is simple. The bisector described above is valid for all lower slices. The legs of a subject remain apart down to the feet, in the Anthropometric stance.

Segmentation in the upward direction is a more difficult problem. Leg data is completely separate at the lower slices, but then comes together and actually merges before the legs themselves end at the crotch area. Figure 8 shows two slices, one lower on the legs, and one near the crotch. Although these are both leg slices, they are topologically different. The right figure could be classified as two manifolds, while the left figure is a single manifold. Furthermore, the point at which the leg data merges is different for each subject.



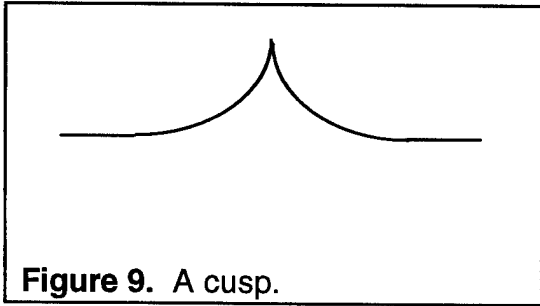


Figure 9. A cusp.

An algorithm is required that separates the data before and after it merges. To accomplish this task, the Cusp algorithm was created. The simplest description of a cusp would be two arcs that meet at a point, as shown in Figure 9. A cusp acts like an arrow pointing the way between the legs. When the legs are

separated, the curvature of the legs still generates a cusp, as shown in Figure 10. Figure 10 also shows the cusp can be used even as the legs come together. The cusp algorithm therefore uses the curvature of the legs to generate a bisector.

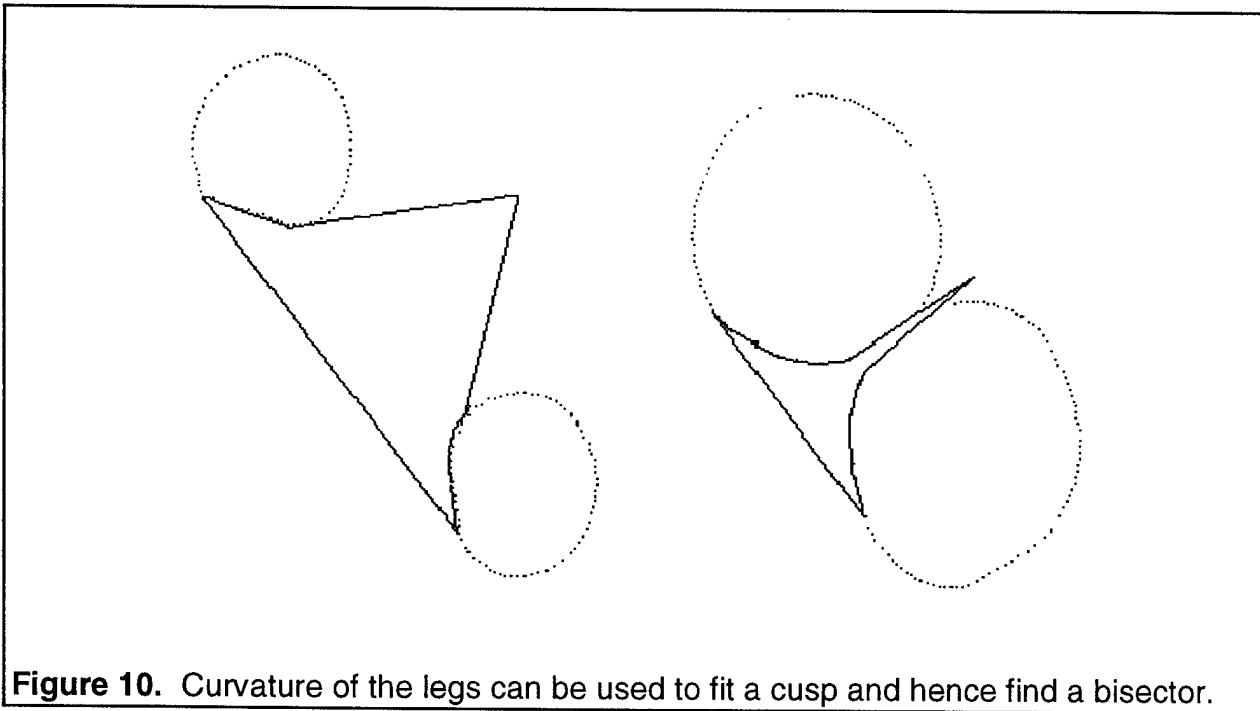


Figure 10. Curvature of the legs can be used to fit a cusp and hence find a bisector.

The cusp algorithm operates on any two dimensional cloud of points. Assume a data cloud exists that can be meaningfully bisected by a cusp fitted to the data. Such a cloud could be a horizontal slice of two human legs (or cylinders) as shown in Figure 11. The domain of the cusp function that bisects the data would be a line segment on a convex hull.

In order to fit the cusp, the distances from each data point to the line segment on the hull are calculated and quick sorted. The points are also quick sorted into sequential order along the

line. The closest point to the line is the first point on the cusp curve. The next point on the cusp curve must fall between the first point on the line and the opposing end point. Hence, the search for the remaining curve points falls on a smaller line segment. From the list of sorted distances, the next closest point to the remaining line segment would be the next point on the cusp curve. Each time a new point on the cusp curve is identified, the line segment shrinks, as shown in Figure 12. Finally in Figure 13, the line segment will shrink to a distance of two data points that are on either side of the cusp. These points are determined to be next to each other by checking their sequential position along the line. A bisector between these last two points will segment the data.

The algorithm, therefore, requires two quicksorts and determines the bisecting cusp in $O(n)$ time, where n may be slightly greater than but no less than the number of data points on the cusp curve. The quicksort data is kept small in the CyScan software by a preprocessing step. The slice is cut in half by a line perpendicular to the front vector that intersects the center of the data points. This removes most of the distant points from the processing, but keeps closer points required to locate the cusp.

The cusp algorithm will attempt to find a bisector for every slice presented to it. The bisector pivots to the left or right when the crotch is reached for both men and women. The bisector for a man pivots because of supporting undergarment and for a women because of bikini lines. The segmentation algorithm must test each slice to determine if the crotch has been reached.

Hence, the height of the crotch is determined by testing a line perpendicular to the bisector. Consider a first line that just touches the front of the legs and a second line that touches the back of the legs. If the angle of the line perpendicular to the bisector is outside the angle of lines drawn on the front and back of the legs, the crotch has been reached.

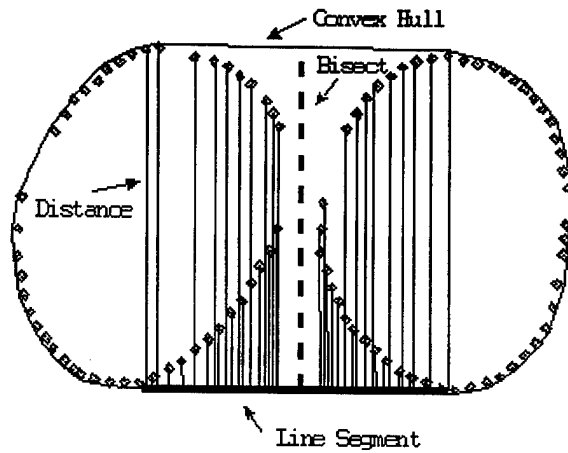


Figure 11. Sample data cloud to be segmented by a bisecting cusp.

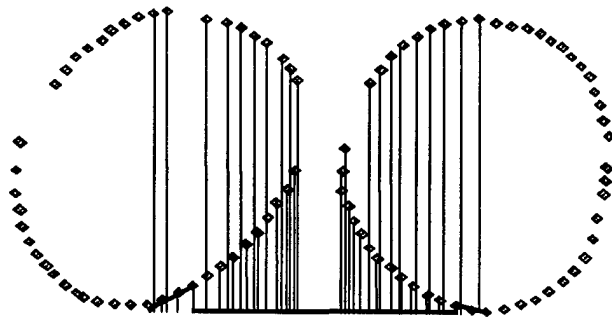


Figure 12. An intermediate step in the cusp fitting process.

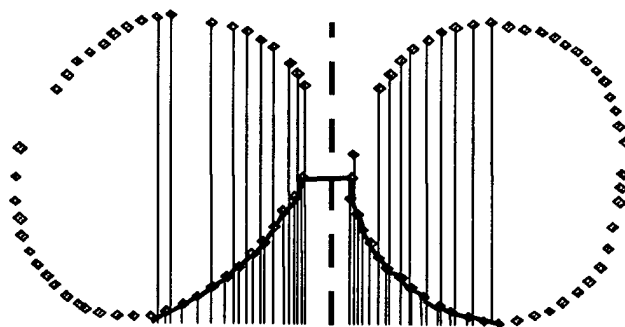


Figure 13. The last two points to be fitted to the cusp can be used to locate the data bisector.

In Figure 14, the front line, the perpendicular line and the back line of a slice are shown. Visualize all three lines intersected at a common point. The perpendicular to the bisector would fall between the two leg lines, hence the crotch has not been reached. In Figure 15, if all three lines intersected at a common point, the perpendicular to the bisector would be outside the two leg lines, hence the crotch has been reached. The crotch heuristic has been successful for all body scans tested to date.

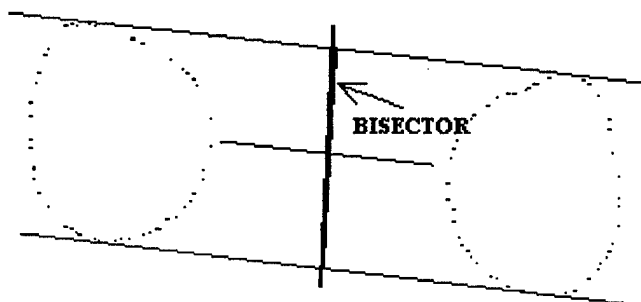


Figure 14. A data slice with good leg separation is fitted with a line at the back of the leg and a line on the front of the leg. The cusp bisector, as well as, the perpendicular to the bisector are shown.

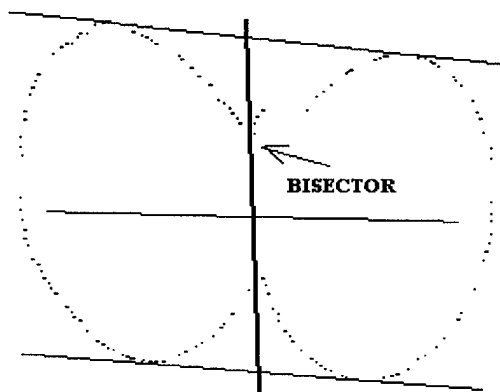


Figure 15. A data slice at the start of the crotch is fitted with a line at the back of the leg and a line on the front of the leg. The perpendicular to the bisector is outside the angle of the front and back lines.

C. Identify arm data

To review, the height of the neck has been calculated, as well as, approximate height for the hands. A slice one quarter of the way up from the hand towards the neck is chosen as the starting slice for arm segmentation. At this height, there should be good separation between arms and torso for a subject standing in the standard anthropometric position. The slice data is projected parallel to the front vector. Slice data is essentially two dimensional since all points have the same height. After projection, the slice data is one dimensional. The two largest gaps in the one dimensional data are caused by the separation between arms and torso. These two gaps are easily located algorithmically, and a bisector within the gaps, parallel to the front vector, is generated. Hence, the data is classified by the two bisectors as: 1. falling left of both bisectors: left arm, 2. falling in-between bisectors: torso, and 3. falling right of both bisectors: right arm.

The segmentation process must now proceed upward and downward from this slice. The downward direction is simple. The bisectors described above are valid for all lower slices. A subject standing in the Anthropometric stance, will have arms that continue to extend away from the torso.

Segmentation in the upward direction is more difficult. The bisectors change, and must be more precise as the arm and chest begin to come together. The arm-torso data presents the same challenges that were seen earlier with the legs. The data begins to merge, as the arm and chest touch, before the end of the arm is reached. Once again, the cusp locator, as described above, is used to find the correct bisector.

The cusp algorithm is applied to both the front and back of each arm-torso slice as shown in Figure 16. The bisector from each cusp should match. If the two cusps do not match up, then a relaxation algorithm is applied to force a match.

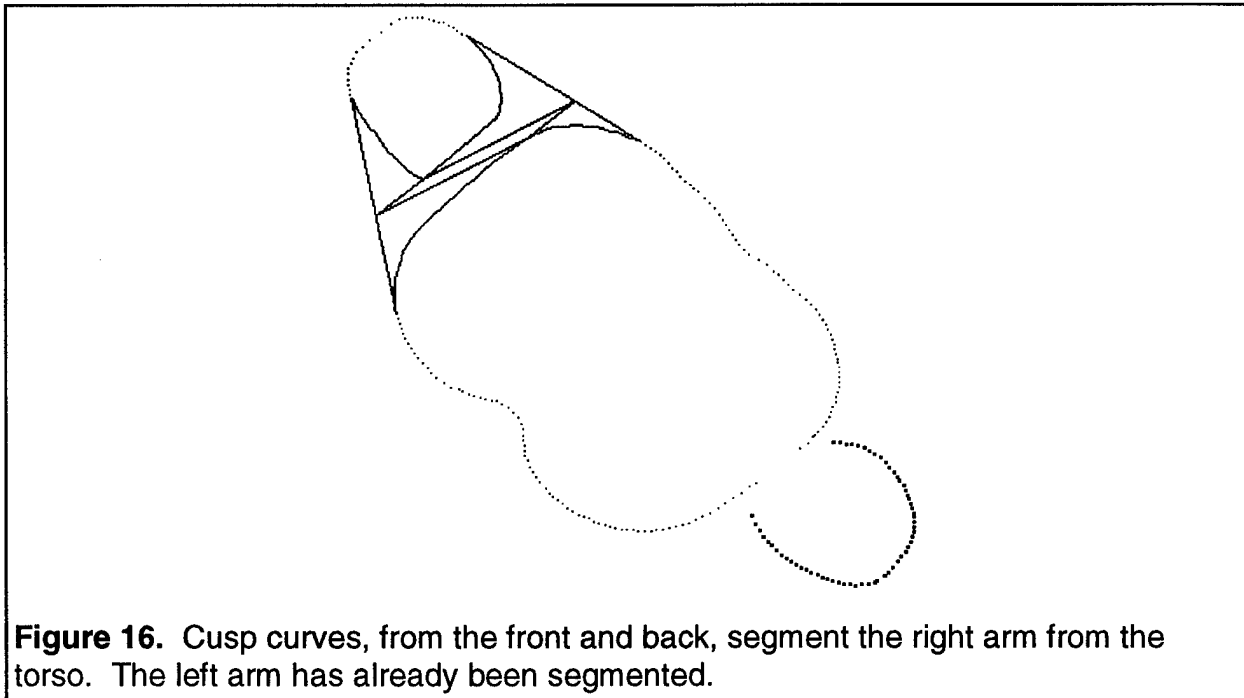


Figure 16. Cusp curves, from the front and back, segment the right arm from the torso. The left arm has already been segmented.

Consider the description of the cusp algorithm given in the previous section. The perpendicular distance value from a point to the line on the hull was used to determine the cusp. If different values were used, such as the distance from the point at an acute angle to the line, a different bisector would be found. Figure 17 shows two cylinders of different diameters with cusps. This is similar to the data configuration of an arm and chest. In Figure 17, the cusp algorithm used perpendicular distances for the front and back bisector detection. A single bisector is difficult to determine. Figure 18 shows the cusp algorithm using vertical distances from each point to the hull lines. The data bisector is obvious. In Figure 18, distances were calculated with the same acute angle from each point to the hull line. The segmentation code must determine the correct acute angle to pass to the cusp algorithm.

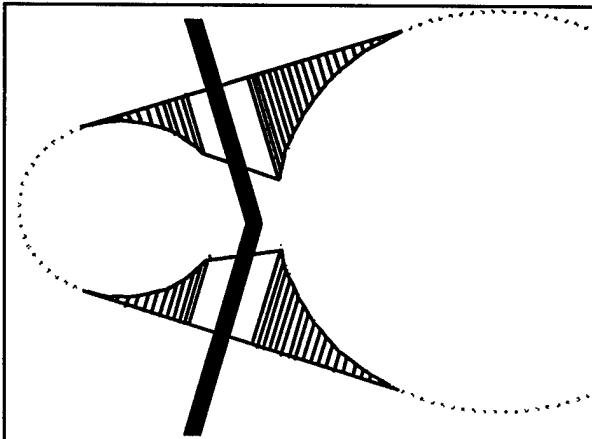


Figure 17. The front and back cusp, using perpendicular distances to the hull lines, generate bisectors that do not match.

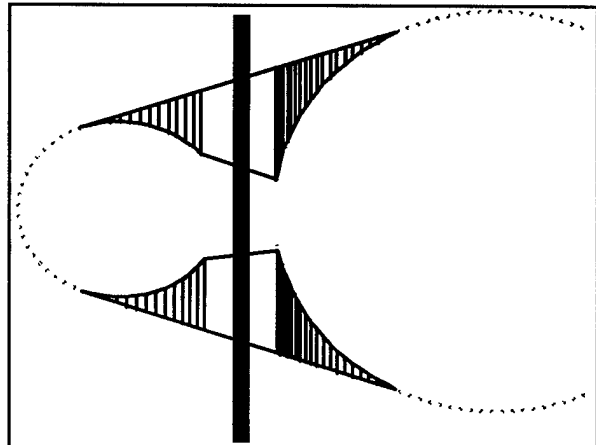


Figure 18. The front and back cusp, using distances at an acute angle to the hull lines, generate the correct bisector.

In the segmentation code, the front hull line and back hull line are compared. Assume a third line between these two lines is created and oriented a half angle from each. A perpendicular of this third line determines the acute angle for distance calculations. Figure 19 shows the third line for the previous two cylinder example.

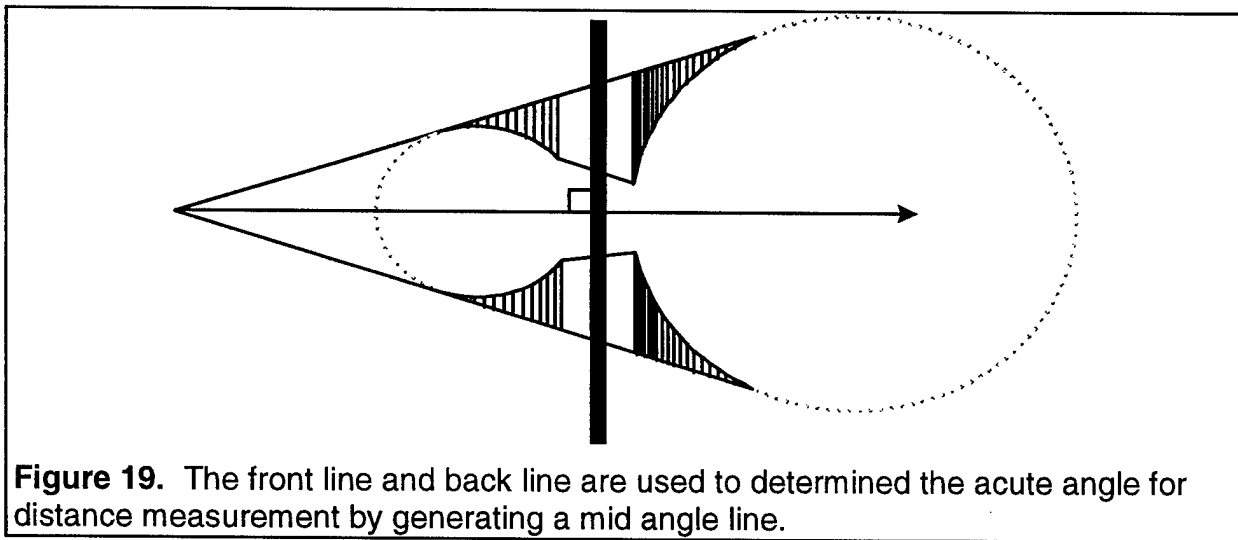


Figure 19. The front line and back line are used to determine the acute angle for distance measurement by generating a mid angle line.

Upon testing the arm segmentation code on numerous subjects, a final refinement was needed. In a significant percentage of the scans, especially near the armpit, the third line did not provide a precise acute angle for the cusp locator. This was due to the non cylindrical nature of muscle and tissue. The front bisector and back bisector, as determined by the cusp routine, did not match. In this situation, the third line angle had to be modified and bisection attempted again.

Figure 20 is used to demonstrate the modification process. In the figure, the original third line generated from the front and back hull lines is horizontal. The front bisector and back bisector are drawn perpendicular (vertical) from the third line to the respective hull lines. If the front bisector was used, part of the back of the arm would be classified as torso. If the back bisector was used, part of the front torso would be classified as arm. The correct bisector is shown with a heavy line. It is a line that connects two points, the point where: 1. the front bisector and front cusp intersect, 2. the back bisector and back cusp intersect. The cusp algorithm would find this bisector if the third line was rotated by angle θ and therefore, be perpendicular to the heavy line.

Hence, the segmentation algorithm modifies the third line and attempts segmentation again. In some instances, the modification process must be repeated several times. In order to prevent oscillation, the modification angle was damped as indicated in the equation below.

$$\theta_{\text{new}} = \theta_{\text{old}} + (\theta - \theta_{\text{old}})(1.0 / N) \quad (3)$$

where θ_{new} = New third line angle

θ_{old} = previous third line angle

θ = As shown in Figure 20

N = number of attempts

When is the armpit reached? The bisecting algorithm divides the data into an arm and a torso. There is a point on the arm that is closest to the bisector for the front of the arm and a point closest to the bisector for the back of the arm. Furthermore, there are two additional points that are identified. Assume the bisector is the central axis of a sliding caliper. As the blades of the caliper close in on the arm, they will be stopped by the furthest protruding point at the front of the

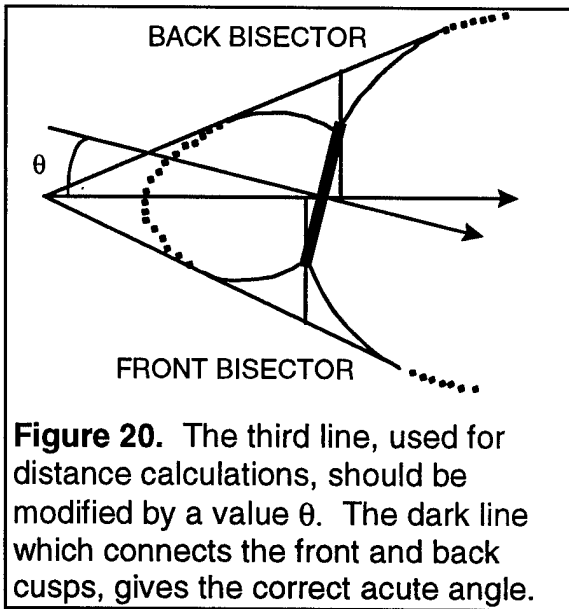


Figure 20. The third line, used for distance calculations, should be modified by a value θ . The dark line which connects the front and back cusps, gives the correct acute angle.

arm and the furthest protruding point at the back of the arm. These four points are used to generate four measurements.

The four points are paired off and classified as the two points in front and the two points in back. A path between paired points is generated by a line parallel to the bisector and a line perpendicular to the bisector, as shown in Figure 21. The length of the parallel line is called the **out** length, while the length of the perpendicular line is

called the **over** length. Hence, there is a front over length, a front out length, a back over length and a back out length.

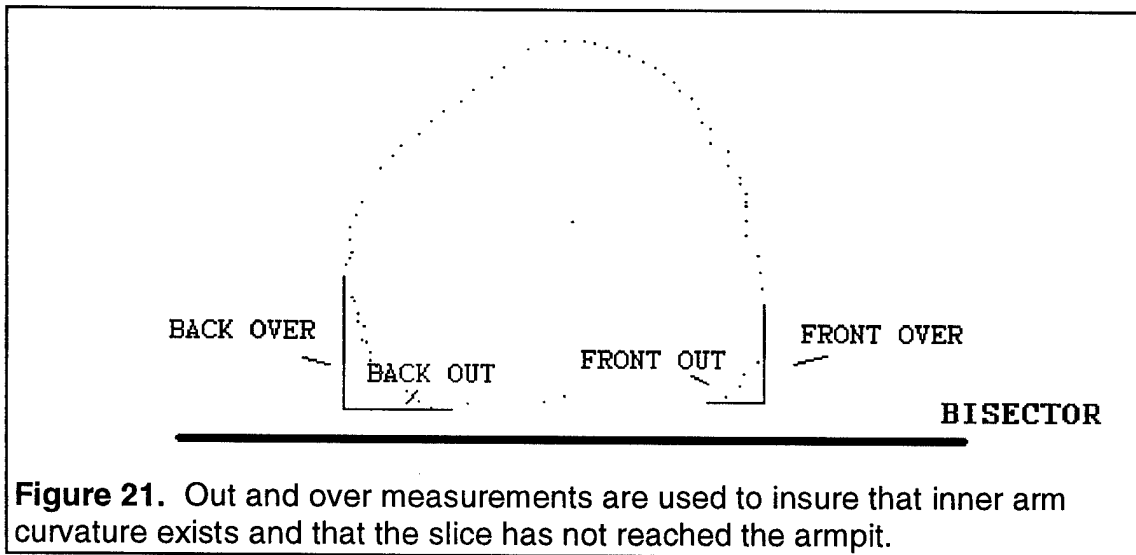


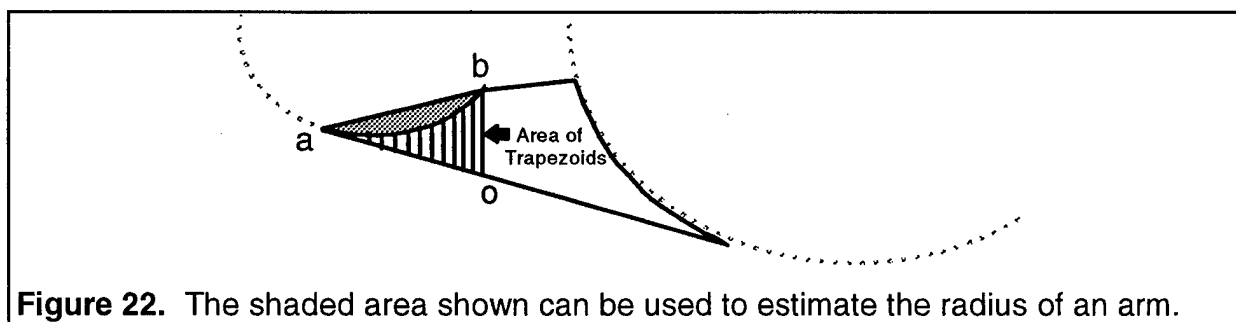
Figure 21. Out and over measurements are used to insure that inner arm curvature exists and that the slice has not reached the armpit.

The over and out measurements are used to detect inner arm curvature. Currently, when all four lengths become less than 4 mm, the armpit is assumed to have been reached.

Each and every slice of the arm is not checked. It was determined that the cusp algorithm could quickly make an estimate of the radius of the arm, while locating the cusp. This is referred as the “on the fly” radius estimator of the cusp routine. Only when the estimated radius of the

arm becomes physically impossible, such as a negative value or it exceeds 100 mm, is the more accurate over-out subroutine called to check the arm.

The radius estimate is made “on the fly” by calculating the area under the cusp to the hull line as shown in Figure 22. The area is accumulated using the trapezoidal rule. The radius estimate is determined by using the shaded area in Figure 22. As described below, this area can be found by subtracting the accumulated trapezoidal areas from triangle *abo*.



Consider the arrangement of geometric primitives shown in Figure 23. \widehat{abc} is a circular arc centered at 'c' with angle ϕ and radius of 'R'. \overline{abc} is an equilateral triangle and \overline{abo} is a general triangle.

Let: L_{ab} = Length of line *ab*

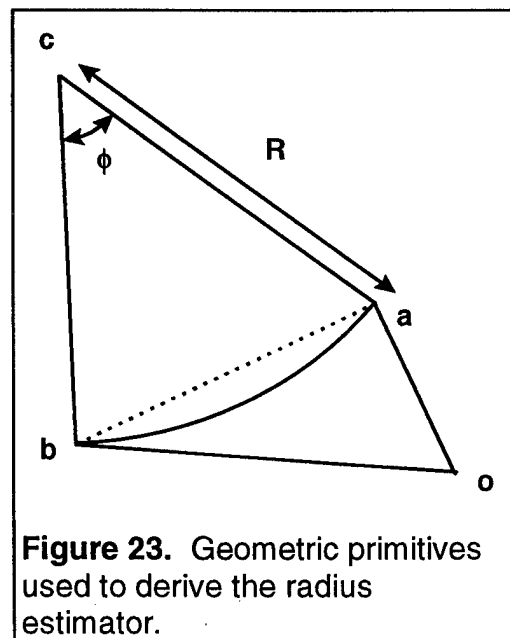
$$A_{\widehat{abc}} = \text{Area of } \widehat{abc}$$

$$A_{\overline{abo}} = \text{Area of } \overline{abo}$$

$$A_{\overline{abc}} = \text{Area of } \overline{abc}$$

The shaded area found by the cusp locator in Figure 22 will be assigned the value A_{cusp} . Assume

$A_{trapezoid}$ is the area found by summing the trapezoids.



From Figure 22 and 23, it can be concluded that:

$$A_{cusp} = A_{\overline{abo}} - A_{trapezoid} = A_{\hat{abc}} - A_{\overline{abc}} \quad (4)$$

where: $A_{\hat{abc}} = \frac{\phi R^2}{2}$

$$A_{\overline{abc}} = \left[\frac{L_{ab}}{2} \left[R^2 - \left(\frac{L_{ab}}{2} \right)^2 \right]^{\frac{1}{2}} \right]$$

Hence,

$$A_{cusp} = \frac{\phi R^2}{2} - \left[\frac{L_{ab}}{2} \left[R^2 - \left(\frac{L_{ab}}{2} \right)^2 \right]^{\frac{1}{2}} \right] \quad (5)$$

The arc angle can be found from the equilateral triangle, \overline{abc} , using:

$$\frac{\phi}{2} = \sin^{-1} \left(\frac{L_{ab}}{2R} \right) \quad (6)$$

Substituting equation (6) into (5) gives:

$$A_{cusp} = R^2 \sin^{-1} \left[\frac{L_{ab}}{2R} \right] + \left[\frac{L_{ab}}{2} \left[R^2 - \left(\frac{L_{ab}}{2} \right)^2 \right]^{\frac{1}{2}} \right] \quad (7)$$

If a normalize R' is defined:

$$R' = \frac{R}{L_{ab}} \quad (8)$$

and substituted into (7) then:

$$A_{cusp} = (L_{ab})^2 \left[(R')^2 \sin^{-1} \left(\frac{1}{2R'} \right) - \frac{1}{2} \left((R')^2 - \frac{1}{4} \right)^{\frac{1}{2}} \right] \quad (9)$$

In equation (9), R' is the normalized radius of the circle, along the cusp. The expression given for R' in the brackets, is highly non-linear. Using a mathematical package called MATLAB, the following approximation was determined:

$$\left[(R')^2 \sin^{-1} \left(\frac{1}{2R'} \right) - \frac{1}{2} \left((R')^2 - \frac{1}{4} \right)^{\frac{1}{2}} \right] \approx \frac{1}{9.13R'} \quad (10)$$

Substituting the approximation back into (9) and returning to the non-normalized radius R , gives:

$$R \approx \frac{L_{ab}^3}{9.13(A_{cusp})} \quad (11)$$

Equation (11) is used to approximate the radius of the arm, in the cusp locator, 'on the fly'.

2.2.c Localize

The localization software was developed and implemented by Cyberware.

2.2.d Find landmark

With over 300,000 data points collected in each scan, extraction of landmark features from geometry alone is a complicated task. Thus, the application of fiducials directly to the scanned subjects simplifies identification and registration of features. As the understanding of scan geometry grows, landmarks will be removed from the subject.

Fiducial markers are topically applied to the subject before scanning is initiated. $\frac{3}{4}$ " diameter matte white self-adhesive stickers are applied directly to the skin or clothing of the subject being scanned at desired locations such as the left/right acromion (shoulder), cervicale (base of neck), stylium (wrist), *etc.* The most important feature of these markers is that they are

highly reflective of laser light and are distinguishable visually from their background in a completed scan.

Due to the nature of the scanner, the pickup of luminance information on the top of the shoulders is limited. Thus, raised three-dimensional markers are required in these positions. These “raised” markers are fiducials given an element of height in addition to their forward circular profile. In scanning, the side of such markers is detected and regular processing may occur.

The Cyberware scanner generates four separate two-dimensional texture maps, one from each scan head, as shown in Figure 24. These texture maps contain gray-scale luminance information and are stored in a typical RGB format. All image processing is performed on these texture maps.

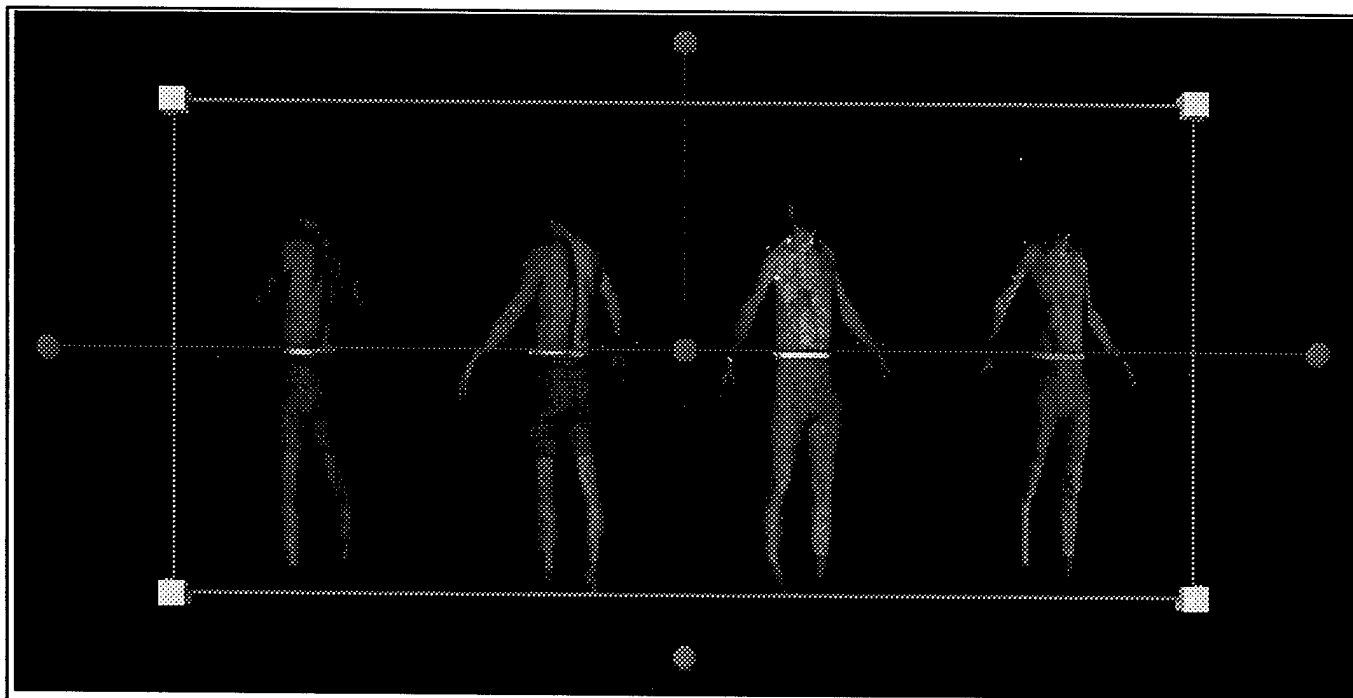


Figure 24. Four two-dimensional texture maps generated in a typical Cyberware WB4 scan. (Participant's identifying features were removed to protect anonymity.)

A pattern recognition method based on conventional two-dimensional image filtering is utilized in finding the highly reflective fiducials in the luminance data. The method is based on

finding the correlation between two images, one being the texture map and the other being an ideal representation of what a marker should look like in a scan. The general idea behind the method is to take a weighted difference between the luminance's of a marker and a region composing its background. If, for instance, a highly luminescent object (with large RGB values) is located on a dark background (with low RGB values), subtracting the average luminance values of the background from that of the object will result in a large positive value (correlation). Conversely, if a dark object were on a light background, using the same technique would result in a negative correlation. Finally, if the object and the background were approximately the same in brightness (very little contrast between them) then the derived result would be a correlation near zero. It is from this line of thought that the following algorithm is adapted.

0	0	0	0	B	0	0	0	0
0	0	B	B	B	B	B	0	0
0	B	B	B	A	B	B	B	0
0	B	B	A	A	A	B	B	0
B	B	A	A	A	A	A	B	B
0	B	B	A	A	A	B	B	0
0	B	B	B	A	B	B	B	0
0	0	B	B	B	B	B	0	0
0	0	0	0	B	0	0	0	0

Figure 25. Matrix used in filtering texture maps. $A = 1/13$, $B = -1/36$. This matrix is convolved with the original texture maps to enhance the appearance of fiducials in a scan.

The key step in devising a filtering technique is to develop a filter kernel (matrix) larger than the size of the fiducials in the image map, which when convolved with the image enhances the brightness of the markers while reducing the background intensity. A typical fiducial, applied to the subject, is

approximately four to six pixels in diameter. In order to have a true center, the matrix dimensions must be odd and thus the innermost region of the filter was selected to have a diameter of five pixels. The coefficients contained in this region of the matrix are equal to one divided by the number of cells in the region, or $1/13$. The entire filter matrix, shown in Figure 25, is a 9 by 9 array, leaving a two-pixel circular border for determination of average background values. Coefficients in the outer circle were set to negative one divided by the number of cells in the region, or $-1/36$. These values are negative as to calculate the weighted difference between the two regions. The remaining corner coefficient values were forced to zero.

In order to convolve the filter with the image, a standard spatial approach is used. Convolution is the process of taking a window and multiplying each weight value by an image pixel value. The center pixel is then replaced with the calculated value. When we move that window across the image, it's known as convolution. Given an input image, a , and an output image, c , of equal dimensions, and the filter, p , with dimensions $(2m-1)$ by $(2n-1)$, the following definition applies:

$$c(x, y) = \sum_{l=-(n-1)}^{n-1} \sum_{k=-(m-1)}^{m-1} a(x+k, y+l) p(k, l) \quad (12)$$

If the result $(x+k, y+l)$ exceeds the dimensions of image a , then the value is set to zero. This formulation calculates only one pixel value of the resultant image, c . In order to compute the entire filtered image, this computation is repeated for every cell of the output image up to the dimensions of the input image, a . For the CyScan implementation, this procedure is repeated four times, once for each texture map which results in four images in which the marker candidates have been enhanced and background luminance is reduced.

Once the filtered image maps are generated, they are then mapped onto the three-dimensional point cloud contained in the scan data by an algorithm already designed by Cyberware. The result is a 3-D point cloud in which all points have an enhanced luminance value, as well as, corresponding position information.

Next, the scan is subjected to a thresholding routine, which eliminates all points with luminance values less than the threshold value. The gray-scale value at which thresholding occurs is between 40-60% of maximum intensity. With the typical number of fiducials used in test scans, only a few hundred points are retained from the original 300,000 data points in the scan, after thresholding is executed.

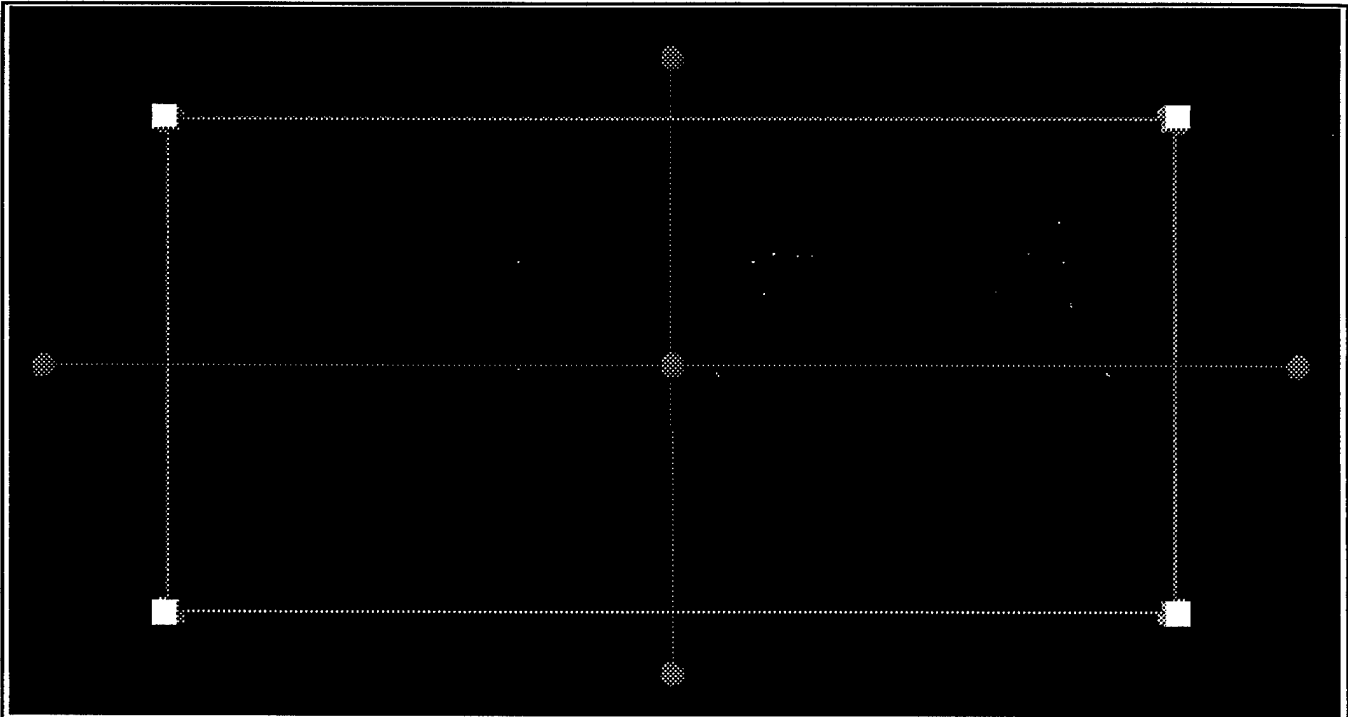


Figure 26. The four two-dimensional texture maps after filtering. Bright points indicate located fiducials candidates.

The points left in the scan, shown in Figure 26, represent all the fiducials candidates that were detected by the filtering process. A cluster of candidates is only a cluster of points. It is not recognized as a marker. The candidates must be grouped together as a marker. The grouping algorithm functions by a recursive, neighbor-locating technique. The distance between two points is used to determine whether or not a point is part of the same marker. Cartesian point distances are calculated and compared to an epsilon value (about 1 cm). If the distances are larger than the set threshold, the points are not included in the same group. After the groups of points have been formed, a center for each group is calculated and the group is named based on the localization software.

2.2.e Measure

The Measurement software was developed and implemented by Cyberware Inc. with significant contribution from Beecher Research Inc.

3. SCAN COVERAGE

3.1 Scan Coverage - Software development objectives

To help answer the question “How well does the Cyberware WB4 scanner meet the minimum scanning requirements for apparel sizing purposes?”, the accuracy of the captured data needs to be analyzed. It is important to know if the scanned data is precise enough to extract the needed sizing information. Part of answering this question involves writing the tools to extract information from the scan. Another part of answering the question is to analyze the data itself for accuracy and consistency. The second half of this report covers the analysis of the scanner and the data it generates.

The scan coverage task was changed from the original proposal, as discussed in Interim Progress Reports. There was an immediate need to address artifacts in the data generated by inherent characteristics of scanner technology. Developing the scanner simulator software helped in the understanding of these artifacts. For example, a Scan Data Analysis Tool (SDAT), which is used to inspect simulator results, proved useful to Cyberware for inspecting actual scanner results. The scanner simulator software is presented in the section below. Section 3.3 titled “Additional Software Tools” describes software useful for processing simulated data, as well as real data.

3.2 Scan Coverage - Software achievements

There will always be a trade off between scan coverage, the quality of the data and the cost of the scanner. Clearly, there is not a simple formula that shows how these issues impact each other. Due to the nonlinear nature of the problem, a simulator was created. This simulator could compare existing configurations, such as two heads versus four heads, and help reveal which design works better. The simulator can be helpful in finding an optimal trade off between cost and performance of future scanners.

3.2.a Simulator configuration review

The laser scanner simulator emulates the behavior of an actual scanner so an analysis of the hardware may be performed. In particular the simulator can show the results of a scan for a given object and given scanner configuration.

Figure 27 is taken from the simulation software and shows how a scanner is depicted with the software tools. Four tracks are displayed as straight lines. On the tracks are rectangular planes that represent the enlarged image plane of a camera. Lines running perpendicular to the track represent the laser plane projectors. A scan head is a set of two cameras and a projector contained on a single track.

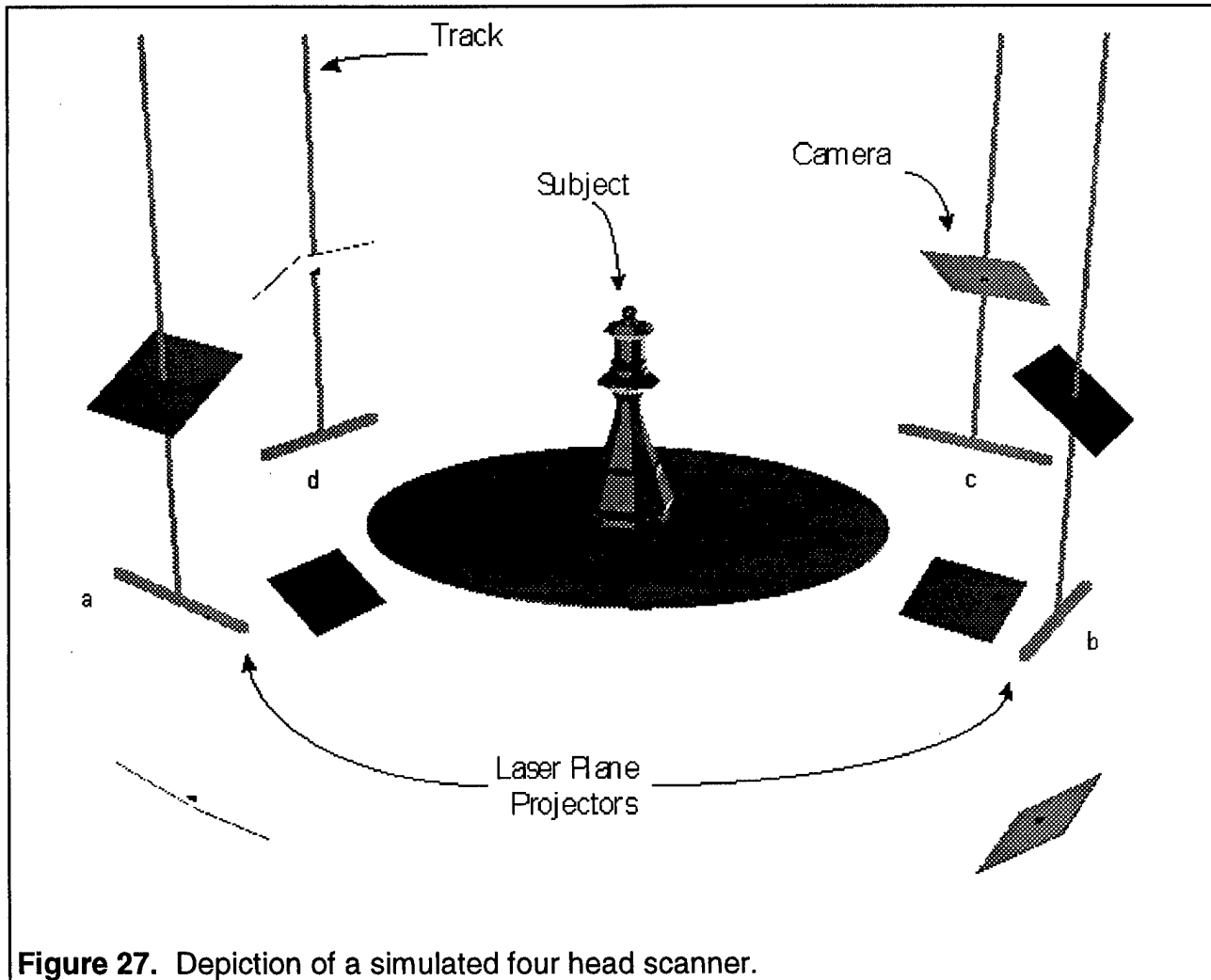


Figure 27. Depiction of a simulated four head scanner.

Items such as tracks, cameras, and laser planes may be combined to form a scanner in any configuration. In addition the parameters describing each component can be changed thereby changing the performance of the over all scanner.

Cyberware places two cameras and a laser plane projector in each scan head. A scan head follows a linear path from top to bottom. Figure 28 compares a sketch an actual Cyberware scan head and its simulator depiction. By placing the two images together, correspondence can be observed.

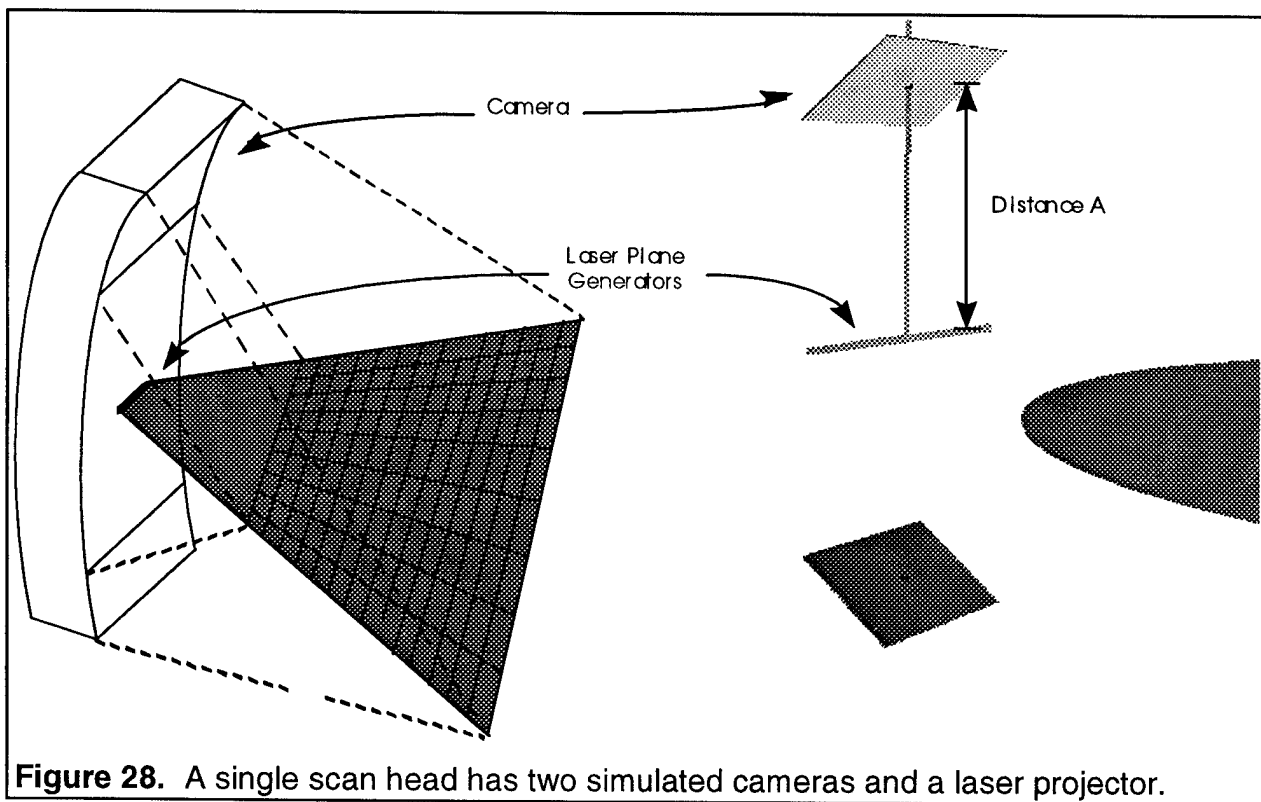


Figure 28. A single scan head has two simulated cameras and a laser projector.

The Cyberware Whole Body scanner is a laser scanning system which uses triangulation to make surface measurements. A plane of laser light is projected onto a surface from a known location in space. The intersection of the laser plane creates a stripe of light on the subject which is detected by a camera. A triangle can be created by the path of the laser light to the subject, the line of sight to the camera and a line drawn from the center of the laser emitter to the camera. There are four geometric cases that must be considered when analyzing this triangulation system. They are:

- Ideal geometry
- Camera image occluded by a surface
- Laser plane occluded by a surface
- Data capture on a significantly sloping surface

Some of the cases may occur simultaneously. Lets consider each of these cases for a single scan head, with a laser plane and one camera. The following discusses each case for the measurement of a single point, using a single camera.

- Ideal Geometry

The ideal geometry occurs when the laser plane intersects the subject and the intersection is clearly recorded by the camera. This is the most common situation and a profile of this geometry is shown in Figure 29.

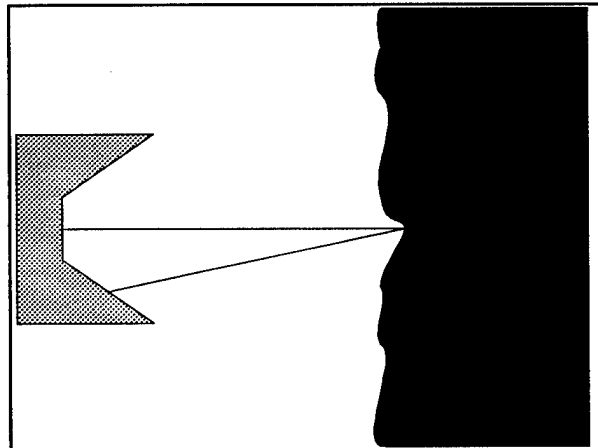


Figure 29. The ideal scan geometry.

- Camera occluded by a surface

When a subject has an extreme surface indentation, the camera can become blocked. This results in a void in the data. An example of this situation is when a subject is wearing wrinkled clothing. This situation is shown in Figure 30.

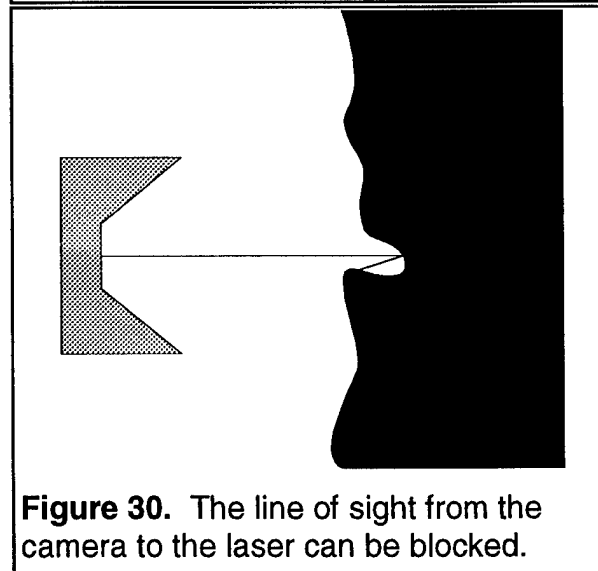


Figure 30. The line of sight from the camera to the laser can be blocked.

- Laser plane occluded by a surface

When one part of the subject's surface blocks another surface from being intersected with the laser plane, data will be missed. An example of this is the under arm area. The arm blocks the torso from receiving laser light. This situation can be seen in Figure 31. Note that one data point is collected, but the data point directly behind it is missed.

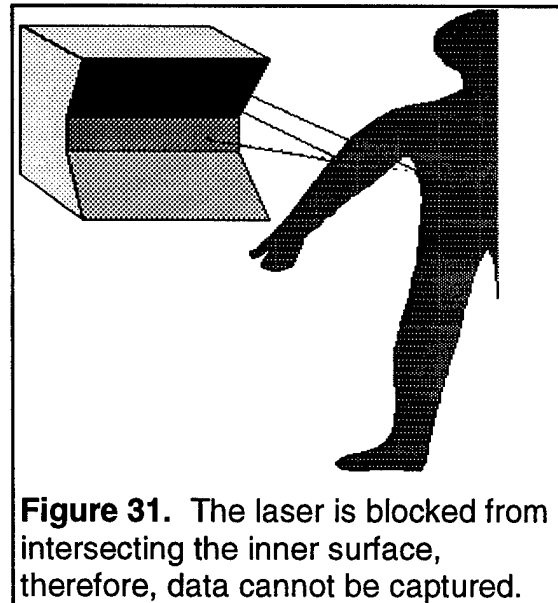


Figure 31. The laser is blocked from intersecting the inner surface, therefore, data cannot be captured.

- Data capture on a significantly sloping surface

When the surface is nearly parallel to the laser plane, a great deal of information can be lost between captured frames. This can happen for two reasons: 1. the surface struck by the laser does not reflect back a sufficient amount of light due to the angle of the surface; 2. there is a significant amount of surface area between the two frames. An example of this geometry occurs in the shoulder area of a human subject. Examples are shown in Figures 32 and 33.

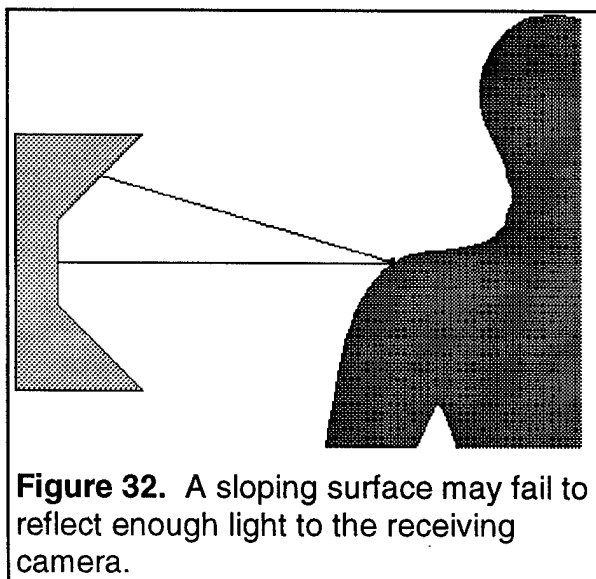


Figure 32. A sloping surface may fail to reflect enough light to the receiving camera.

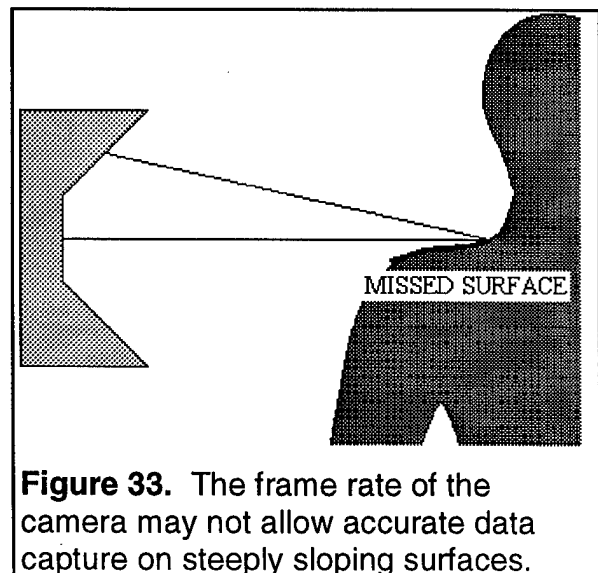


Figure 33. The frame rate of the camera may not allow accurate data capture on steeply sloping surfaces.

All the errors described above are influenced by the configuration of the scanner.

Different aspects of the configuration affect each type of error differently. For error that results from camera occlusion, (Figure 30) the most significant configuration parameter is the distance between the camera and the laser plane. When the distance between the camera and the laser plane is small, the size of the surface indentations on a subject is relatively large and less likely to occlude. However, making this distance small reduces the resolution at which data is captured. It also increases the change in resolution from the front of the scanning area to the back.

For error caused by the occlusion of the laser plane (Figure 31) the most significant scanner parameters are the number and position of the scan heads. More scan heads at a larger variety of angles will reduce the chance that a point on the surface will be missed due to a blocked laser plane. Finally, when error is due to a significantly changing surface slope (Figures 32 and 33) the most significant configuration parameters are the orientation of the scan heads and the path they follow.

Although it is not difficult to identify the configuration parameters that impact scan coverage, it is very difficult to determine what optimal configuration parameters should be. The geometry errors listed above are specific to the subject being scanned. Hence, different configurations will be optimal for different classes of subjects. Furthermore, configuration decisions have a large impact on hardware costs. Due to these facts, the approach was not to attempt to design an optical scanner, but to create a tool that could aid Cyberware in this process.

The computer simulator was created such that the parameters most essential to the scan coverage issues discussed above could be adjusted by the user. These include:

- The position of the camera relative to the laser plane (distance A in figure 28).
- The location of the scan heads (a, b, c, d in figure 27).
- The path on which the scan heads travel (tracks in figure 27).
- The number of scan heads.

Furthermore, internal parameters related to the quality of the data can be altered, such as the resolution of the image plane, the dimensions of the image plane and the focal point. Complete flexibility regarding the subject class was also made available.

3.2.b Simulator system overview

An implementation of the software tool, described in the previous section, was created within the CyScan framework. The functional algorithms required for the tool are referred to as the Laser Scanner Simulator or SimScan for short.

The SimScan Control window has four main modes. The controls for each mode are placed in a window, as shown in Figure 34 item A. The four modes consist of the generic interaction mode and three editing modes. Each of the editing modes alters the Simulation Control window by placing its interaction controls in the window. To summarize, the four main editing modes are:

- A. *Generic mode menus* (Figure 34, item A)
- B. *Edit scanner mode* (Figure 34, item B)
- C. *Edit track mode* (Figure 34, item D)
- D. *Edit camera mode* (Figure 34, item C)

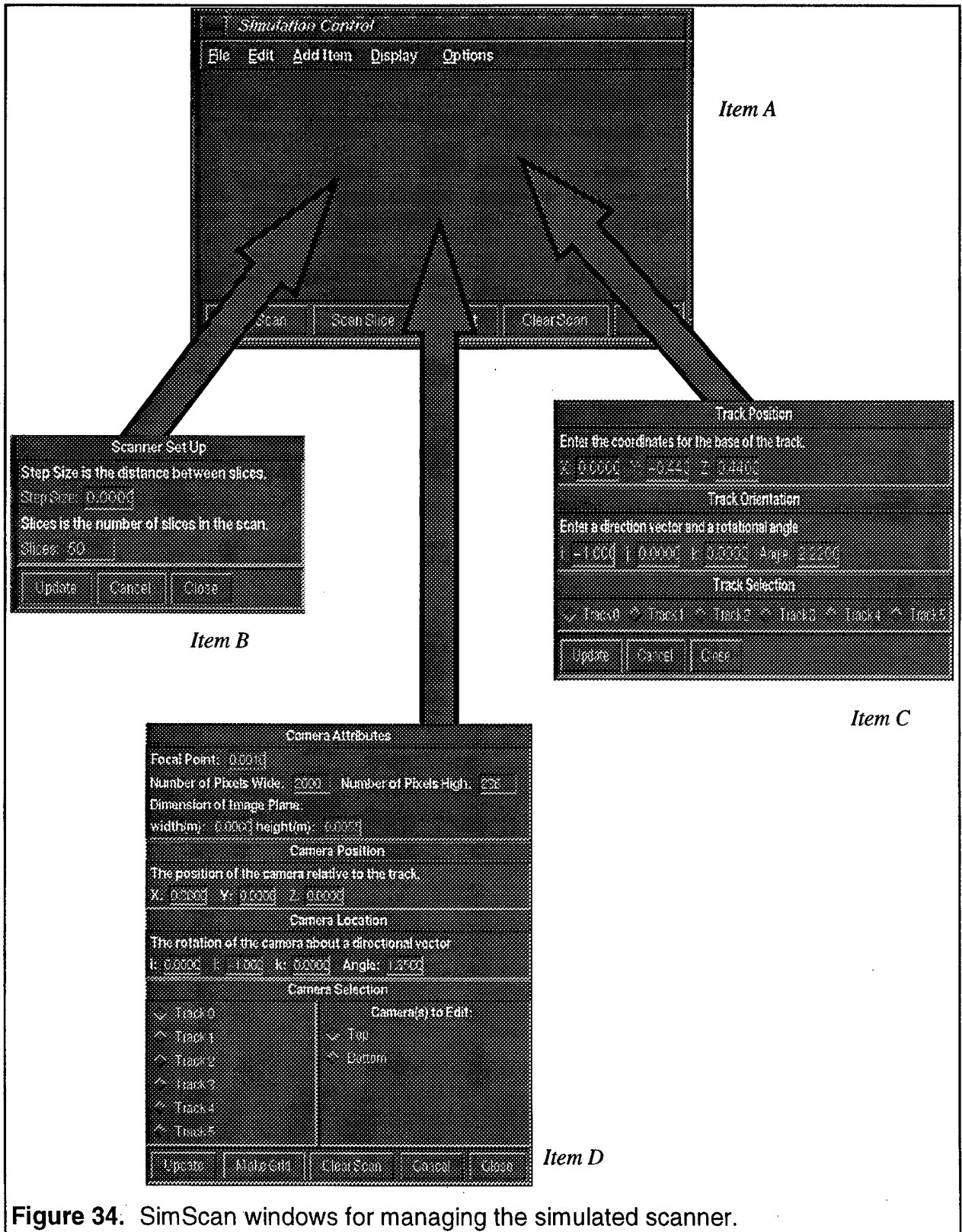
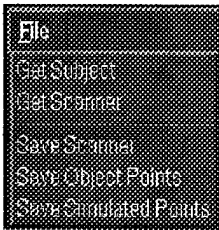


Figure 34. SimScan windows for managing the simulated scanner.

A. Generic mode menus

The menu shown in Figure 34, item A, follows a traditional format. It contains a file, edit, add item, display and options menu. Each menu is described below:

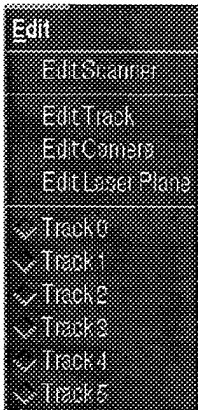
File Menu



- Get Subject - loads an open inventor subject to be scanned.
- Get Scanner - loads a scanner configuration that was created and saved.
- Save Scanner - Saves the current scanner.
- Save Object points - Saves points on the subject's surface that would be created for an ideal scanner at a given resolution.

Save Simulated Points - Saves the points generated by the simulated scanner.

Edit Menu



- Edit Scanner - Puts the simulation control window into Edit Scanner mode. This allows the user to edit parameters pertaining to the entire scanner.
- Edit Track - Puts the simulation control window into Edit Track mode. This allows the user to edit parameters pertaining to the specified track.
- Edit Camera - Puts the simulation control window into Edit camera mode. This allows the user to edit parameters pertaining to the corresponding camera.
- Edit Laser Plane - Reserved for future use.

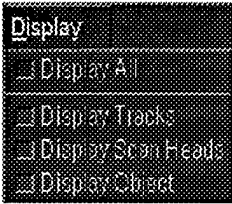
This menu also contains a set of radio buttons that correspond to the tracks in the simulated scanner. The radio buttons allow the user to select which track to edit.

Add Item Menu



- Add Scanner - Adds a track with a scan head to the current scanner.
- Add Track - Adds a track to the current scanner
- Add Scan Head - Adds a scan head to the current track.

Display Menu



This menu has two sets of check buttons. When the first button, Display All, is selected all other buttons become inactive. When the Display All check button is not selected the remaining buttons indicate what is displayed. When the Display All check button is selected, the state of the other buttons are saved.

Options Menu

The options menu was created for future additions to the software.

B. Edit scanner mode

When the Simulation Control window is in the edit scanner mode, the controls shown in Figure 34 item B, are placed in the main window. The simulator collects horizontal data slice in the same fashion as the Cyberware Whole Body scanner. In this mode, the user may control the number of slices to be scanned and the distance between each slice. These controls will impact the time it would take for a real scanner to operate, and the amount of scan coverage. It does not effect the physical geometry of the scanner itself.

C. Edit track mode

When the Simulation Control window is in the edit track mode, the controls shown in Figure 34 item C, are placed in the main window. The position for the base of the track and a orientation may be changed through these controls. Consider Figure 35; the center track has a different tilt angle in the top and bottom figures as set by the Track Orientation input.

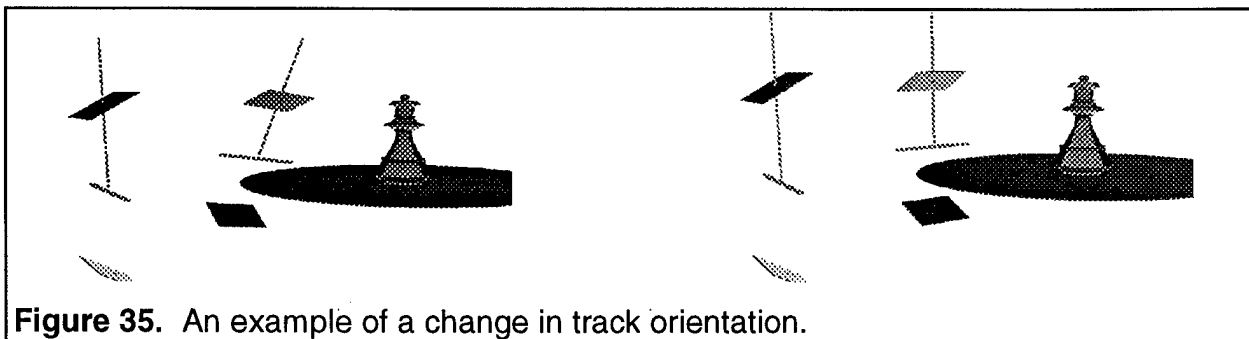


Figure 35. An example of a change in track orientation.

D. Edit camera mode

When the Simulation Control window is in the edit camera mode, the controls shown in Figure 34 item D, are placed in the main window. These controls allow the user to change all the parameters related to the camera. Some parameters such as focal length and pixel size are not readily visible in the display area. However, parameters such as the image plane size and the image plane orientation are displayed to aid the user, as shown in Figure 36.

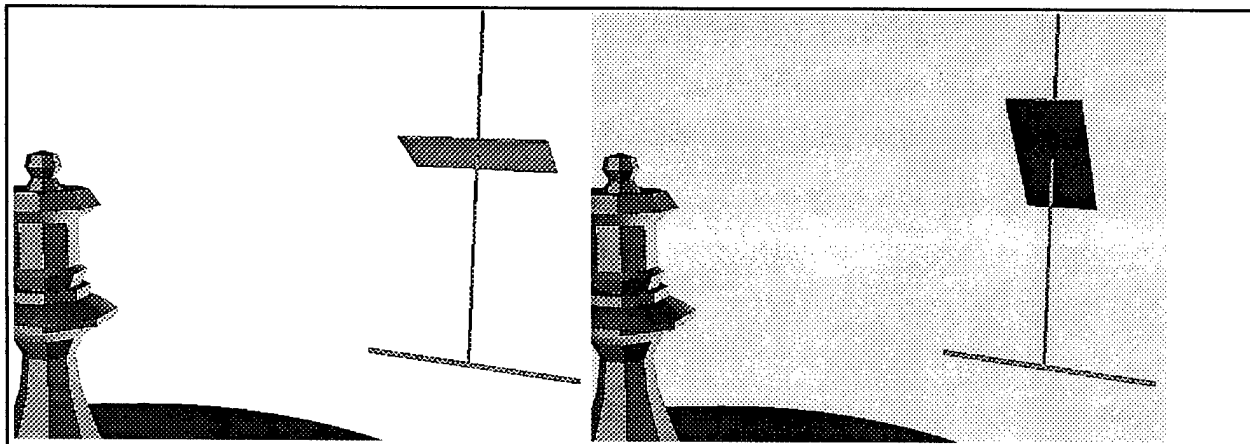


Figure 36. Example of changes in a camera's image size and orientation parameters.

3.2.c Using SimScan

SimScan is started from the CyScan main menu under an item named OU Research Tools. When SimScan is started, the Simulation Control window appears. The user creates a virtual scanner by adding and editing tracks and scan heads. The internal parameters associated with each component can be changed. The following describes the method for creating a simulated Cyberware whole body scanner. Once created, the scanner model can be saved through the file menu for future use.

The first step is to create tracks on which the scan heads will travel. Tracks can be added through the add item menu. The position and orientation of each track is set through the edit

track controls in Figure 34, item B. Four tracks are added and a scan head must be attached to each.

To add a scan head, the desired track is selected and the add item menu is used. Adding a scan head creates two virtual cameras and a virtual laser plane. The camera and laser plane models can be placed on the track similar to the Cyberware configuration.

Next, the internal configuration parameters of the virtual scanner should be checked and refined. The camera model contains most of the internal parameters of interest. These parameters are set through the edit cameras' controls.

For both a real and a virtual camera, each pixel on the camera's image plane corresponds to a region of scan space. If these pixels are projected into the scan space and connected, a grid is formed. The grid is unique for each camera of the scanner. The template grid presented by the CyScan software is the actual camera grid for a Cyberware WB4 scanner under its control. In SimScan, a similar set of points may be generated. Using these grids, the scanning configuration of the SimScan simulator and the Cyberware WB4 hardware may be compared.

Figure 37 demonstrates the comparison of a Cyberware grid and the SimScan grid. The CyScan grid is shown by the crossing lines. The points shown are from the SimScan projection of the image plane. Examining the top image of Figure 37, it is apparent that the points do not match the template grid. Adjustments must be made to the camera's image plane, focal length and orientation using the edit camera controls depicted in Figure 34, item D. The result is displayed in a new projection and is shown in bottom part of Figure 37. This simulator configuration more closely models the Cyberware WB4 scanner.

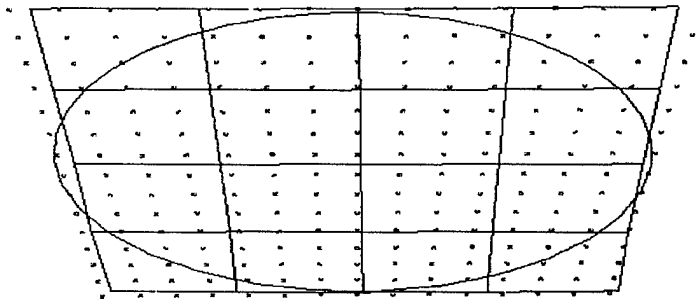
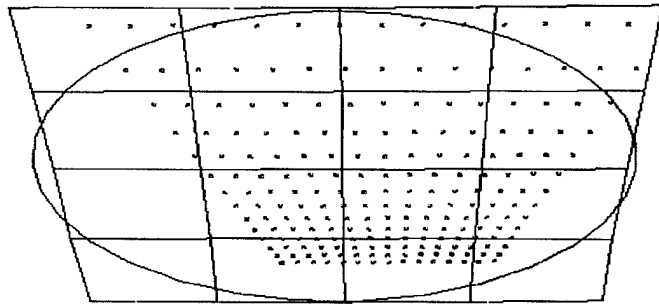


Figure 37. Camera parameters can be changed and compared to the Cyberware template grid. The camera grid is shown as points.

Once a scanner is created, a subject must be placed in the scanning area. By default a sphere is the assumed subject. To use any other subject, an Open Inventor file of the subject must be available. An Open Inventor file contains three dimensional graphics information in a format defined by Silicon Graphics. All CyScan data is stored as Open Inventor files. The Open Inventor subject can be loaded into SimScan, using the file menu item: Get subject.

Once the subject is in place, the user should select the number of slices to be scanned and the spacing between slices. This is done through the edit menu item: Edit Scanner. The simulation is activated by a button labeled Scan Slice. Starting at the base, SimScan scans up the subject, unlike the typical hardware scanner which travels down. The reason the simulated scanner starts at the base is that it executes quicker. Empty slices can be added almost instantaneously once the top height of the subject has been exceeded.

While the simulation takes place, a blue sliding bar indicates how much of the scan has been completed. After the simulation is complete, the points can be viewed and manipulated with CyScan tools just like data collected from hardware.

3.2.d Software design

The scanner was modeled in an object oriented fashion with C++ classes. Over twenty five classes were written. The most significant classes correspond to the major hardware components such as: tracks, scan heads, laser planes and cameras. Just like their hardware counterparts, these software objects can be put together in many different ways. At least one laser plane and one camera is needed to simulate a scan head. At least one scan head and one track is needed to simulate a laser light scanner.

- Tracks - A track defines a path for other objects to move along. The track is represented with a NURBS line defined by four control points. When the track path is a straight line, SimScan takes advantage of this and the simulation process executes faster. The code was optimized for straight line track because Cyberware currently uses them in their scanners. Tracks may contain other objects including other tracks. Two tracks may be combined to create a two dimensional path.
- Laser planes - The laser plane is represented by a set of virtual rays. Virtual rays can reflect from a surface, or pass through it, thereby indicating inner surfaces that are being occluded by a real scanner. Each virtual ray starts at a common origin and passes through a control point. The control points are defined to be collinear, and therefore, the rays are coplanar. A picture of this is shown in Figure 38.

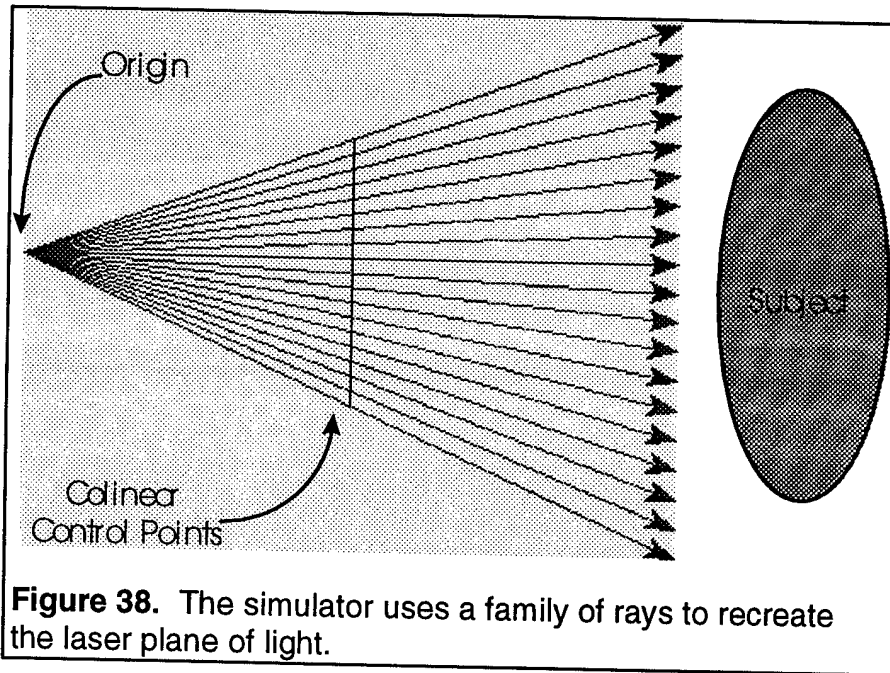


Figure 38. The simulator uses a family of rays to recreate the laser plane of light.

An infinite number of rays would be an exact model of the laser plane of light. However, each additional ray increases the complexity of the simulation and therefore increases the time to perform a simulation. Hence,

the number of control points had to be finite, but set sufficiently high such that at least one ray corresponds to each pixel row for a large range of camera configurations. Therefore, each and every pixel of the camera can potentially image a laser ray for any arbitrary surface. In less extreme camera parameters, there are multiple rays passing through a projected pixel row.

A subject in the scan space will be intersected by laser rays. The intersection points are determined and the information is saved as a set of points.

- **Cameras** - A set of laser intersection points is sent to the camera object. The camera object determines if the points will show up on its image plane. This decision is based on the camera's internal parameters, orientation and location in space. If a point does show up on the image plane, the camera object determines which pixel is illuminated.
- **Scan heads** - Cyberware scan heads each contain two cameras. Once each single camera object has captured all points created by the laser plane, two camera views can be merged. A merge unites cameras by creating a composite image plane that has any pixels turned on in either donating camera.

3.2.e Operational examples

SimScan was tested using different, easily verifiable subject arrangements. For example, an Open Inventor subject was created that contained a bigger box placed on a smaller box. The large box overhung the small box in such a way that the scanning platform and the large box blocked the cameras from detecting the laser rays intersecting the small box, as shown in Figure 39.

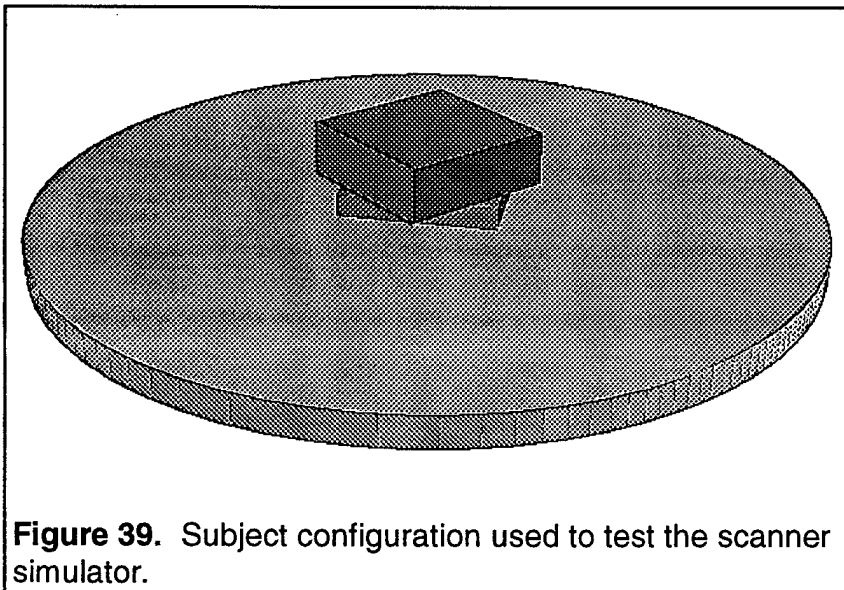
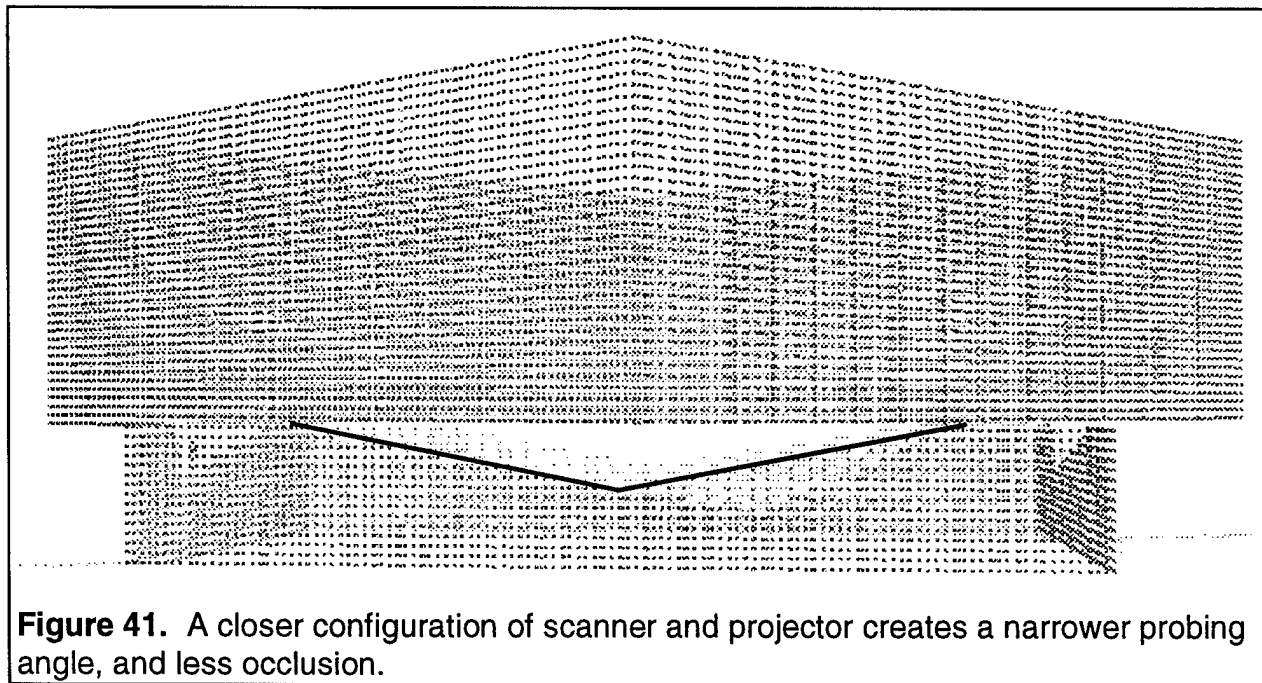
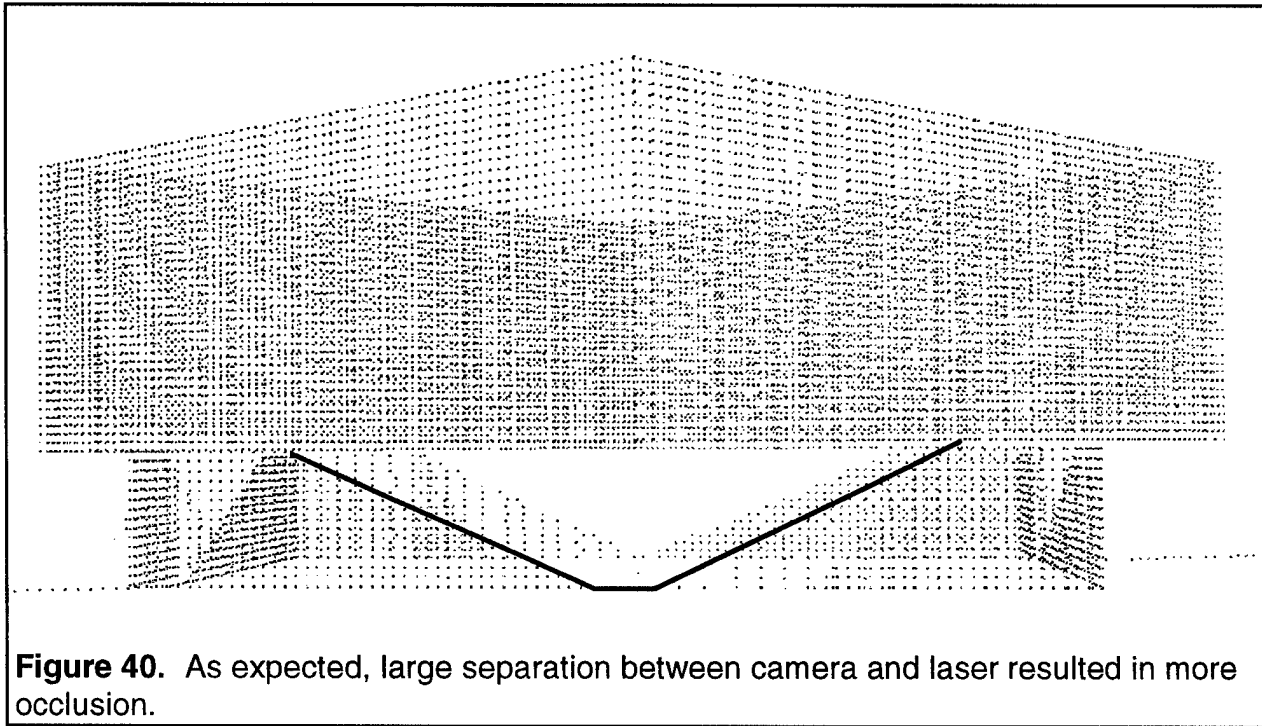


Figure 39. Subject configuration used to test the scanner simulator.

When a simulated scan is conducted, gaps in the data are found as expected due to the cameras being blocked. Changing the camera parameters will cause a change in the occlusion patterns observed in the simulated scan.

In Figure 40, the camera's image planes were set 34 centimeters from the laser plane. As seen in the data, the corner of the large box blocked the camera from collecting data on the surface of the small box. The image planes in the virtual scanner were then moved to be just 14 centimeters from the laser plane. The occlusion pattern below the large box was smaller as shown in Figure 41. By narrowing the distance between the cameras and the laser plane, the triangular line of sight becomes narrower. Points between the platform and large box could be captured. The total number of data points that could be captured on this subject is increased.

Several simple experiments, such as the one above, were conducted. These were used to test SimScan's theoretical performance, as well as, compare the computer models to the actually Cyberware WB4 hardware.



A. Two head scanner verses four head Scanner

SimScan was used to compare subject scans of a two headed scanner, with a four headed scanner. The configuration of the four headed scanner was similar to the Cyberware WB4 hardware. The two headed scanner was identical to the four headed scanner with cameras removed in such a way that two opposing cameras remained.

The subject consisted of two cylinders to simplify the results. One of the cylinders has a large diameter and the second cylinder has a small diameter. The cylinders were spaced about five inches apart to resemble a human's arm and torso. A virtual image of the two cylinders subject is shown in Figure 42.

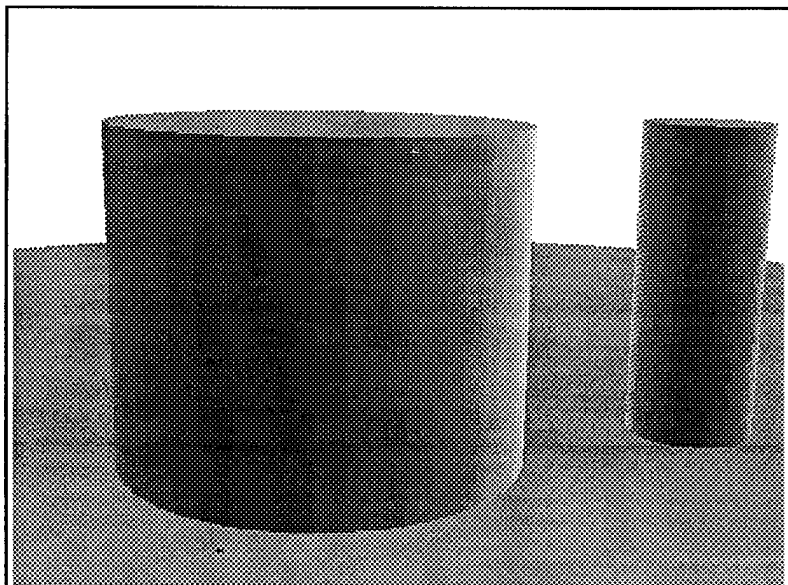


Figure 42. Subject used in comparison of two head and four head scanners.

The results of the simulation, conducted using the two headed scanner, contained a large area of occlusion as shown in Figure 43. Half of the data corresponding to the inner arm area was missing from the smaller cylinder. The missing curvature would have violated the assumptions of the cusp locator in the segmentation software. When the four headed scanner is used

the occlusion is almost eliminated. The simulation conducted using a four headed scanner is shown in Figure 44. The overlapping area between coverage of scan heads helps reduce the amount of occlusion. Furthermore, the configuration of two scan heads that face each other results in no overlapping areas. With four scan heads, the configuration generates overlapping areas which are useful for data verification as explained in the next section on curl removal.

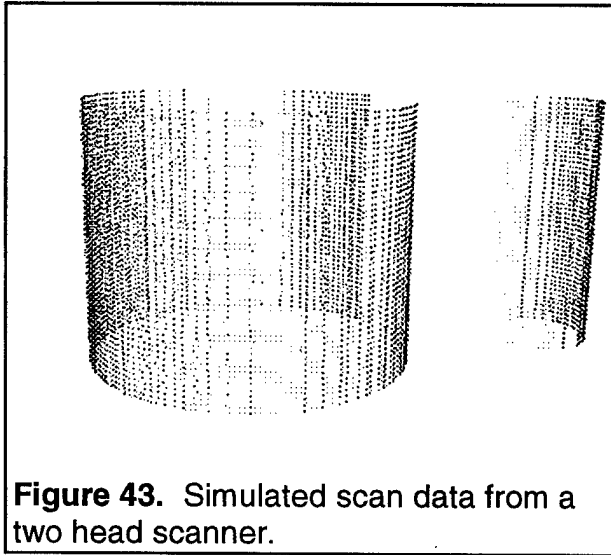


Figure 43. Simulated scan data from a two head scanner.

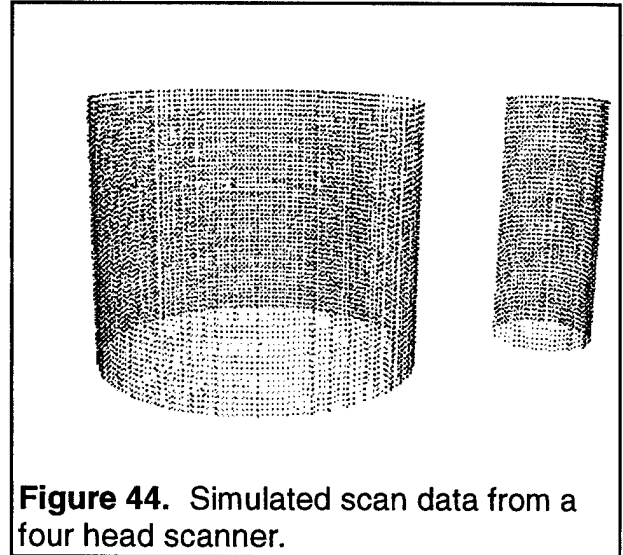


Figure 44. Simulated scan data from a four head scanner.

3.3 Additional software tools

Other software tools were developed in conjunction with the Laser Scanner Simulator or SimScan. These tools proved useful for processing simulated data, as well as real data. Two tools are highlighted in this subsection: Scan Data Analysis Tool and curlEase.

3.3.a Scan Data Analysis Tool (SDAT)

Over 300,000 points are generated when a full sized human being is scanned. When rendered to a screen, the data takes the form of a human, making it difficult to observe geometric relationships between sets of points. To overcome this problem, a tool called the Scan Data Analysis Tool (SDAT) was created.

The tool can be used to observe details about the data generated by SimScan. It can also be used to observe details about data generated by an actual scanner. Cyberware has found it to be a helpful tool for reviewing templates.

The SDAT is invoked through the OU Research Tools menu. A window titled SDAT (see Figure 45) opens up that allows the user to quickly view data in a variety of different ways. One of the most common views is the slice view. This is a slice thickness of 2 mm, positioned

parallel to the floor. This view displays all the points captured in one camera frame for all four cameras. The data maintains the color scheme set up by CyScan, so data from each camera is easily identifiable

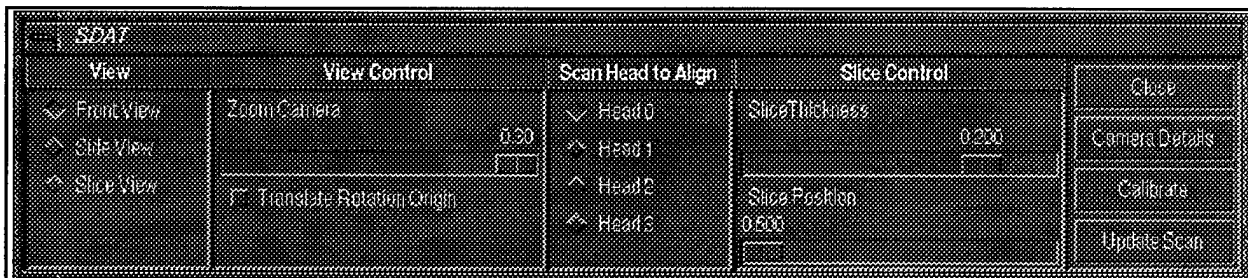


Figure 45. Main window for SDAT.

The *View* section of the SDAT window allows the user to select from one of three previously defined Camera Placement views. These views were originally generated using the Camera Placement window discussed below.

The *View Control* section of the SDAT window allows the user to enlarge the view screen. There is also a check box to translate the origin. When this box is checked, single slider windows can be used to translate the origin.

The *Scan Head To Align* section of the SDAT window allows the user to select which set of data is controlled by the single slider windows. Each time a simulated head is selected, the sliders are zeroed, there by extending the range of the sliders and allowing for fine adjustments.

The *Slice Control* section of the SDAT window is used to control the view volume of the camera. This allows the user to view as much of the scan data as desired. By decreasing the slice thickness to two millimeters, a single slice may be displayed. The slice position slider allows the user to move up and down through the data.

Pushing the *Calibrate* button causes three slider windows to be created. The sliders may be used to adjust the position of SimScan data. When moved, these sliders move the data in a vertical, horizontal or rotational direction within the plane of view.

Pushing the *Camera Details* button in the SDAT window opens the *Camera Placement* window shown in Figure 46. The cameras can be altered any way the user wishes. The camera being altered is the camera that creates the view seen in the image window.

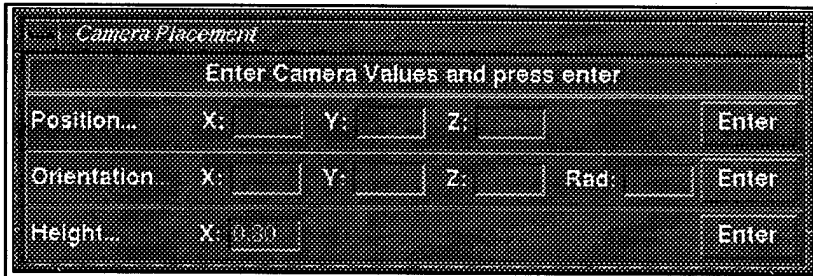


Figure 46. GUI window for camera parameters.

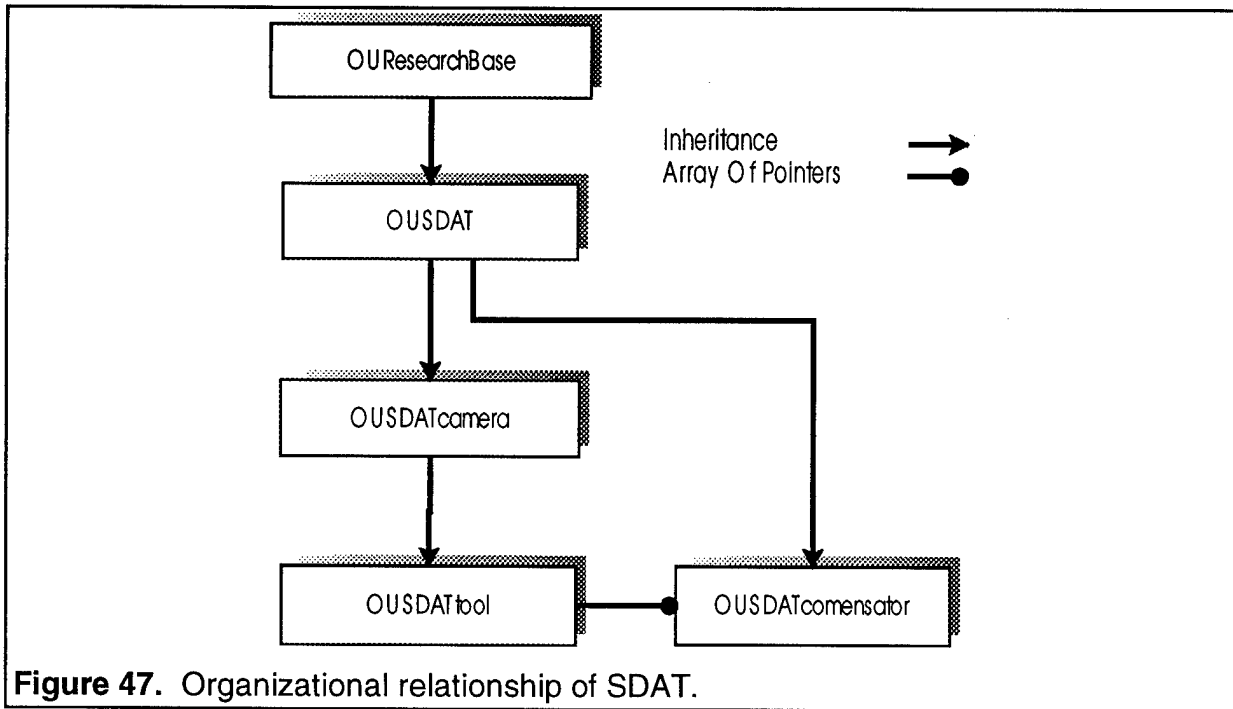
Pushing the *Update* scan button in the SDAT window causes all the data to be shifted by the amount that the scan has been altered with the single slider windows.

With these controls, the user can manipulate the data in many different ways. Once the data has been altered, tools in CyScan can be used to decurl, segment or extract measurements. When the SDAT is closed, the transformations performed on the data are displayed in the form of matrices.

SDAT is a combination of C++ and tcl code. The most important C++ classes are OUSDAT, OUSDATcamera and OUSDATtool. The organizational relationship of these classes is given in Figure 47.

All C++ classes are ultimately inherited from OUREsearchBase. This allows the objects to be stored within a structure created by CyScan that is not destroyed when execution is returned to tcl. When the code is first instantiated, an object of OUSDATtool is created that contains an array of pointers to OUSDATcompensators. An OUSDATcompensator is created for each set of scan data to be analyzed.

A camera is also added to the Open Inventor scene graph that defines the image displayed in the CyScan view window. The original camera is left on the scene graph so that it is not altered by the SDAT code. Other nodes needed to alter the appearance of the data are added to the scene graph. When the analysis is complete, the tool is closed and all nodes are removed from the scene graph.



3.3.b Curl ease

Data recorded from any single scan head tends to curl away from the actual surface, near the fringes, due to optical nonlinearities. The curl noise should be eliminated from the scan before measurement processing proceeds. To eliminate this artifact, an algorithm called “curlEase” was developed. curlEase looks at the data, one scan head at a time. In Figure 48, the solid line was created by connecting data points captured from a single scan head. Data points in front of this line would have been recorded from a different scan head. Points in front of the line must be curl or the laser from the original scan head would have picked them up. Order area calculation are used to categorize curl points by determining which side of the line they fall on. This is equivalent to inspecting the slope-intercept formula of the line.

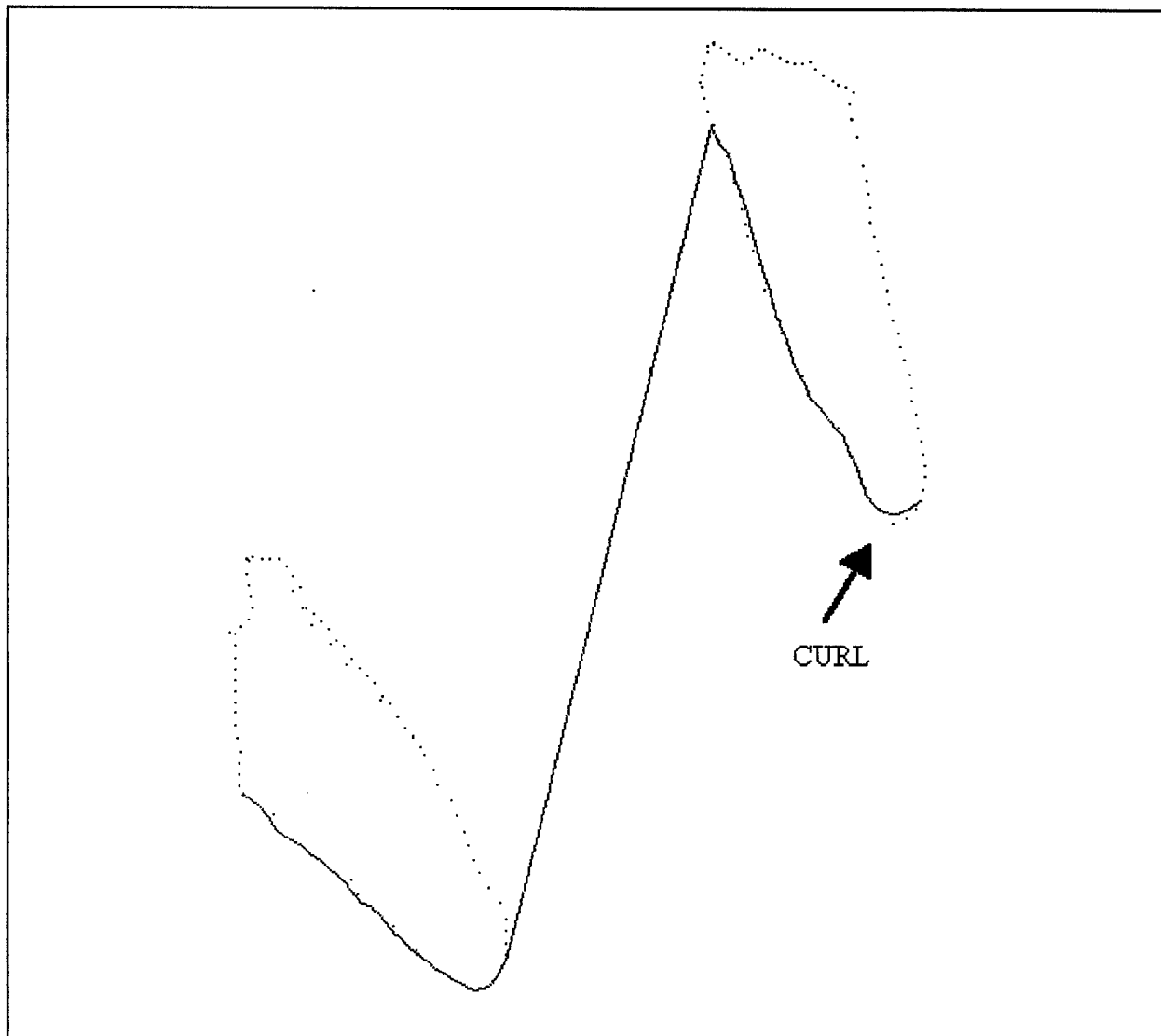


Figure 48. Data capture from one scan head 'curls' away from data (line) captured by a scan head positioned at the bottom of this figure.

The algorithm can also eliminate extraneous data points caused by stray environmental reflections. Assume in Figure 49 that scan head 'X' generates the solid line as described above. Points found by other cameras between the edge of the scan space and the first point acquired by 'X' are suspected of being noise. However, 'X' has limited resolution. Points found by other cameras that are near the first 'X' point should not be eliminated unless they are outside the resolution of 'X'. The uncertainty of camera resolution is represented by the bottom horizontal

lines in the figure. These lines then step to the top as they move away from the data points found by 'X'. Data found by other cameras in the region of the step are not collaborate by 'X', and hence are eliminated as stray noise.

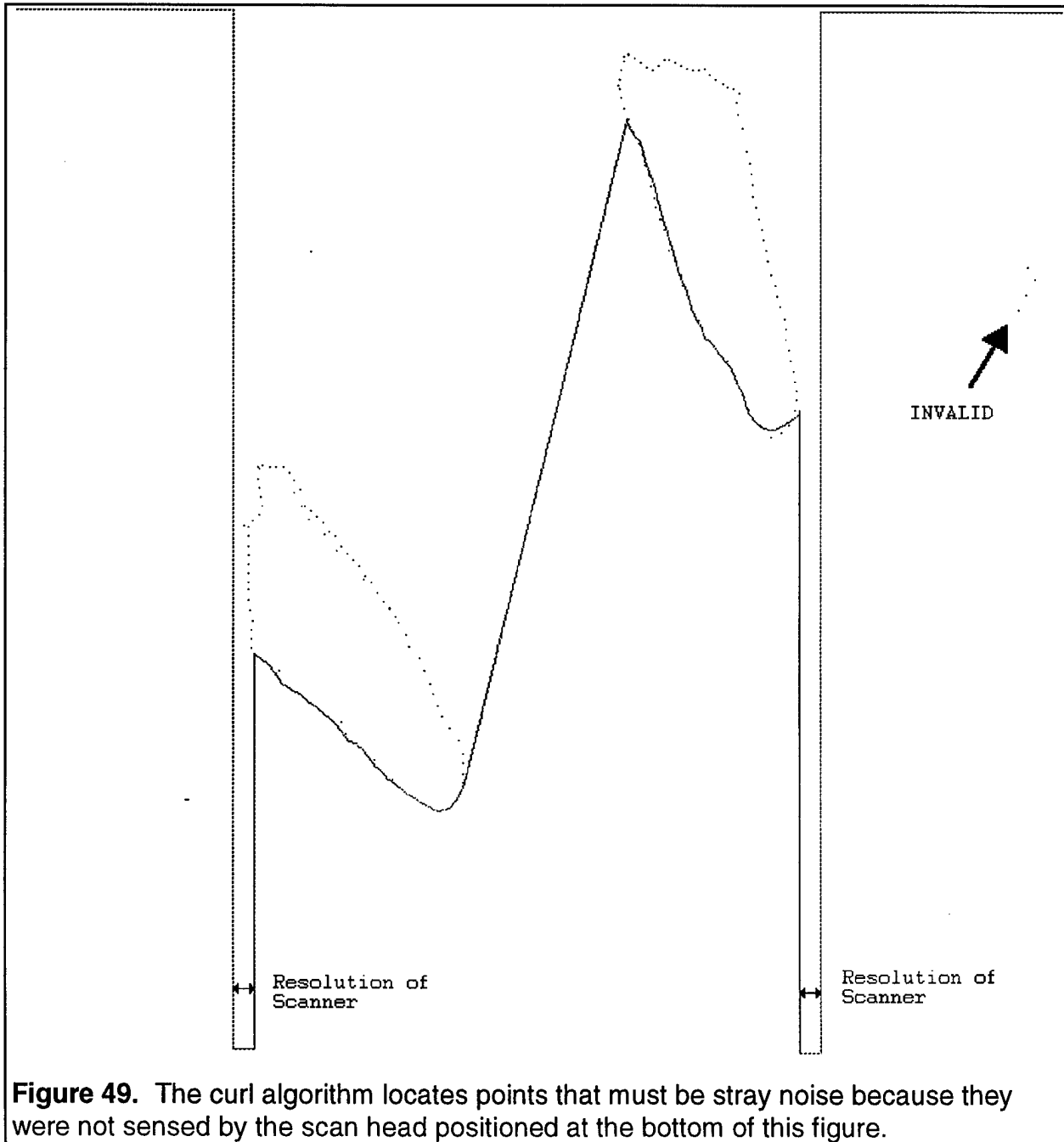


Figure 49. The curl algorithm locates points that must be stray noise because they were not sensed by the scan head positioned at the bottom of this figure.

After testing on numerous scans, a refinement to curlEase was required. Originally, a point was eliminated if it falls in front of the laser plane's intersection with the surface. In Figure 48, the points lower on the left foot fell in front the solid line, and hence were eliminated. It is desirable to keep this information. A tolerance parameter was introduced into curlEase which prevents points within a certain distance from the laser line, from being eliminated.

An underlying assumption of the curlEase algorithm is that a straight line can be drawn between points. This is probably a reasonable assumption when points are close together, but becomes less likely as the points detected from any single head are further apart. The tolerance parameter for distance from the laser line should take this into account. Hence, a tolerance parameter was introduced that controls the height of a triangle generated between two data points, as shown in Figure 50. The further apart the data points are, the larger the triangle is, and the more variance is allowed for other scan head points.

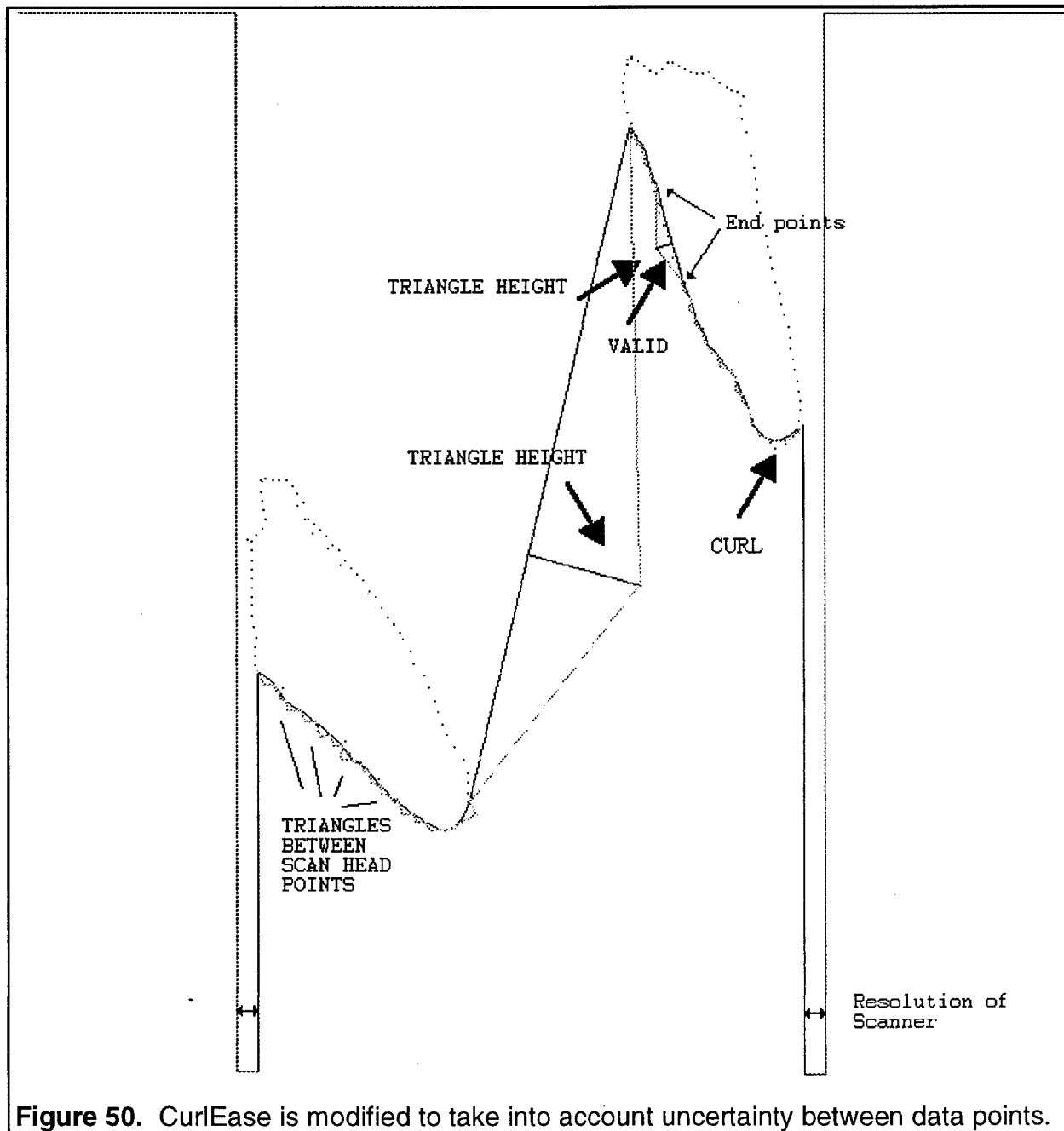


Figure 50. CurlEase is modified to take into account uncertainty between data points.

4.0 DISCUSSION

Numerous software tools were researched and developed under this contract. They include:

- find Moment
- cusp locator
- segment head
- segment arms
- segment legs
- enhance landmarks
- group landmarks
- profile
- SimScan
- SDAT
- curlEase

Many of the algorithms represent break through developments in three dimensional scan analysis.

The techniques above worked on well over a hundred scans. Success presents a challenge. Invariably, a subject will be scanned who confounds an algorithm. At this stage in the research, such subjects will usually have subtle shape differences that are more difficult to analyze. A validation study, currently under way, will quantify the success of the methods for a large scan data base. Maintenance/refinement research on all the techniques above can be expected after this validation study. Measuring the final few percent of the human population may be as difficult as automating the measurements of the majority.

A second direction this research must take is the continued elimination of landmarks. Landmarks are an important crutch to test the science of scanning. Long term success of apparel sizing by scanner depends on minimal subject preparations before scanning.

-
1. J.H. Nurre, "Locating Landmarks on Human Body Scan Data", Int. Conf. On Recent Advances in 3-D Imaging and Modeling, Ottawa, Canada, May, 1997.
 2. R.Sedgewick, *Algorithms in C*, Addison-Wesley Publishing Company, Reading, Massachusetts, 1990.
 3. B.K.P. Horn, *Robot Vision*, The MIT Press, McGraw-Hill Book Company, Cambridge, Massachusetts, 1986.

Appendix A.

Catastrophic Measurement Errors - Only the Scan Form Milestone was achieved. An image of the form is shown below as captured from CyScan.

The screenshot shows a Microsoft Access database window titled "Microsoft Access - [ScanData]". The main form is titled "NATICK BodyScan Data Sheet". The form is divided into several sections:

- PURPOSE:** Includes radio buttons for "Project", "Demo", "Research", and "Other".
- SUBJECT:** Includes fields for "Last Name", "First Name", "Sex", "Phone", "Race", "Ethnicity", "Height", "Weight", "Age", "Pregnancy", and "Other Preparations".
- OPERATOR:** Includes fields for "Last Name", "First Name", "Sex", and "Location".
- SCAN DATA:** Includes a "File Name" field and radio buttons for "Data Appears FINE", "Data SHELF LIFE", and "Data Appears CORRUPTED".

In the bottom left corner, there is a table with the following data:

File Name	File Size
130378_12_08_10_0002	130378_12_08_10_0002

At the bottom of the form, there are fields for "Scan Number", "Date", "Time", "Embedded Key", "CyScan Version", "Scanner Model Number", and "Customs Form". The status bar at the bottom indicates "Record 1 of 29".