AL/CF-TR-1996-0095

ARMSTRONG LABORATORY

INTEGRATE 1.25: A PROTOTYPE FOR EVALUATING THREE-DIMENSIONAL VISUALIZATION, ANALYSIS AND MANIPULATION FUNCTIONALITY (U)

Dennis B. Burnsides
Patrick M. Files

SYTRONICS, INC
4443 DAYTON-XENIA ROAD, BLDG. 1
DAYTON, OHIO 45432-1949

DTIC QUALITY INSPECTED 3

Jennifer J. Whitestone

CREW SYSTEMS DIRECTORATE
HUMAN ENGINEERING DIVISION
WRIGHT-PATTERSON AFB, OHIO 45433-7022

May 1996

INTERIM REPORT FOR THE PERIOD OCTOBER 1994 TO MAY 1996

19971028 023

Approved for public release; distribution is unlimited

AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6573

NOTICES

When US Government drawings, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Please do not request copies of this report from the Armstrong Laboratory. Additional copies may be purchased from:

National Technical Information Service
5285 Port Royal Road
Springfield, Virginia 22161

Federal Government agencies and their contractors registered with the Defense Technical Information Center should direct requests for copies of this report to:

Defense Technical Information Center
8725 John J. Kingman Road, Suite 0944
Ft. Belvoir, Virginia 22060-6218

TECHNICAL REVIEW AND APPROVAL

AL/CF-TR-1996-0095

This report has been reviewed by the Office of Public Affairs (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

KENNETH R. BOFF, Chief
Human Engineering Division
Armstrong Laboratory

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
May 1996

3. REPORT TYPE AND DATES COVERED
INTERIM, October 1994 to May 1996

4. TITLE AND SUBTITLE
Integrate 1.25: A Prototype for Evaluating Three- Dimensional Visualization, Analysis, and Manipulation Functionality (U)

5. FUNDING NUMBERS
F41624-93-C-6001
PR: 7184
TA: 08
WU: 50

6. AUTHOR(S)
*Dennis B. Burnsides
*Patrick M. Files
Jennifer J. Whitestone

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
*Sytronics, Inc.
4433 Dayton-Xenia Road, Bldg 1
Dayton OH 45432-1949

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Armstrong Laboratory, Crew Systems Directorate
Human Engineering Division
Human Systems Center
Air Force Materiel Command
Wright-Patterson AFB OH 45433-7022

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AL/CF-TR-1996-0095

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This report documents the functionality available through INTEGRATE, a Silicon Graphics-based software package, to visualize, analyze, and manipulate three-dimensional topographic data. The analysis capability represented by this software is robust, flexible, and instrumental in applying 3-D anthropometry toward the improved fit of protective equipment, clothing, commercial head gear, and medical devices. Tutorials are available to guide the user through representative applications.

14. SUBJECT TERMS
Surface Scanning
Anthropometry
Human Body Topography
Visualization Analysis Manipulation

15. NUMBER OF PAGES
146

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. Z39-18
298-102

THIS PAGE INTENTIONALLY LEFT BLANK

PREFACE

This research was conducted by the Computerized Anthropometric Research and Design (CARD) Laboratory of the Human Engineering Division, Crew Systems Directorate, Armstrong Laboratory, Wright-Patterson Air Force Base, Ohio. The work was performed under the Scientific Visualization of Anthropometry for Research and Design (SVARD) Contract Number F41624-93-C-6001.

TABLE OF CONTENTS

LIST OF FIGURES v

1.0 INTRODUCTION 1

1.1 Background 1

1.2 Functionality 2

1.3 Object Pool 2

2.0 GENERAL OPERATING INSTRUCTIONS 3

2.1 Starting INTEGRATE 3

2.2 INTEGRATE STATUS WINDOWS 3

2.3 Operating Features 6

3.0 TUTORIALS 9

3.1 Tutorial One: Basic Moves 10

3.2 Tutorial Two: Registration Techniques 12

3.3 Tutorial Three: Point Picking 16

3.4 Tutorial Four: Feature Envelopes 19

3.5 Tutorial Five: Radial Difference Maps 25

3.6 Tutorial Six: Manipulating Whole Body Data 28

3.7 Tutorial Seven: Manipulating Stereophotogrammetry Data 31

4.0 INTEGRATE COMMANDS 35

5.0 INTEGRATE'S AUDIT TRAIL FUNCTION 103

6.0 REFERENCES 105

APPENDIX A TUTORIALS: IMAGE DATA AND SCRIPT FILES 106

APPENDIX B ANATOMICAL LANDMARKS: DESCRIPTIONS AND ILLUSTRATIONS 118

APPENDIX C LANDMARK FILES: ANATOMICAL AND AUXILIARY LANDMARKS 124

APPENDIX D COMMANDS: FUNCTIONALITY FOR HEAD AND WHOLE BODY DATA 127

APPENDIX E FILE FORMATS: DESCRIPTION OF IMAGE DATA FILES SUPPORTED BY INTEGRATE VERSION 1.25 131

APPENDIX F STEREOPHOTOGRAMMETRY: USERS' MANUAL FOR STEREO SLICE 135

APPENDIX G CYBERWARE WB4 DATA: CONVERTING WHOLE BODY SCANNER DATA INTO AN INTEGRATE-COMPATIBLE FORMAT 138

LIST OF FIGURES

Figure 1. Active Object Status Window 3
Figure 2. Object Summary Window 4
Figure 3. Global Status Window 6
Figure 4. Registration of subject with helmet scan for visualizing subject/equipment interface 12
Figure 5. Subject scan with color file (monochrome for this publication) and marked landmarks 16
Figure 6. Two subjects wearing the same size and model helmet 19
Figure 7. Pupil envelopes for five subjects in the same size and model helmet 20
Figure 8. Radial Difference Map (RDM) of the total contact burn mask with respect to the subject's face. Contrasting colors (monochrome for this publication) represent different degrees of fit 25
Figure 9. Shown in this figure is the full body scan data with the body segments separated from the torso 28
Figure 10. Example of a male subject from the stereophotogrammetry survey in the seated position 31
Figure 11. Conclose joining the first and last selected points 51
Figure 12. Erosion of surface data 63
Figure 13. Seam correction with fix_seam 66
Figure 14. Thinning an object 94
Figure 15. Trimming noise from the top of an object: before trimming (the object on the left) and after trimming (the object on the right) 97
Figure 16: With walls set to 698 702, only a cross section of the object appears 99

THIS PAGE INTENTIONALLY LEFT BLANK

INTEGRATE 1.25: A PROTOTYPE FOR EVALUATING THREE-DIMENSIONAL VISUALIZATION, ANALYSIS, AND MANIPULATION FUNCTIONALITY

1.0 Introduction

1.1 Background

The Computerized Anthropometric Research and Design (CARD) Laboratory, Human Engineering Division, Armstrong Laboratory at Wright-Patterson Air Force Base, Ohio has been using surface scanning technology to improve equipment design applications since 1987. CARD Lab researchers evaluated many commercial software packages, such as Computer-Aided Design (CAD), to determine their utility for manipulating surface data for USAF equipment designs. These software programs, however, proved incapable of providing the unique functions required to analyze topographic data on people and their equipment. For this reason, the CARD Lab developed INTEGRATE as a prototype system to test the functionality required to visualize, analyze, and manipulate surface data. The current version of INTEGRATE, version 1.25, offers new functionality to meet the needs of current USAF engineering anthropometry challenges. Researchers now use INTEGRATE 1.25 to test and evaluate new helmet systems, develop augmentative files for surface scan databases, record human-equipment interface geometries, examine newly developed surface scan formats such as whole body data, and prepare surface data for rapid prototyping systems (Robinette et al., 1994; Robinette et al., 1992; Whitestone et al., 1995; Whitestone et al., 1993; Whitestone et al., 1992).

Because INTEGRATE was designed as a prototype, user friendliness was not a high priority for the developers. However, a few hours' experience with the program and this document should familiarize the user with commonly used commands and the general architecture of the software. This document contains five sections:

1. Introduction,
2. General Operating Instructions,
3. Tutorials,
4. INTEGRATE Commands,
5. INTEGRATE's Audit Trail Function, and
6. Appendices.

The user should read the General Operating Instructions before beginning the Tutorials. It is highly recommended that the user "walk through" at least the first two tutorials before beginning new sessions. The INTEGRATE commands are organized in alphabetical order with examples of the use of each command. Finally, the Appendices are included to provide the user with additional information such as anatomical landmark definitions and illustrations, listings of files needed for the tutorials, and batch files for routine sessions.

1.2 Functionality

The goal in the design of INTEGRATE is to provide for future functions so that no changes in the basic program, functions, and data structures will be needed to add any new function. New functions can and will be added quickly when the need arises.

1.3 Object Pool

The Object Pool keeps track of all the information for each object. INTEGRATE can work with an arbitrary number of objects at one time (the present limit is "a lot"). A rule of thumb is that approximately 1.31 million points (10 head scans) can be in the Object Pool at one time. These objects can be displayed or hidden by the user. The amount of memory in use is displayed as a percentage (1.31 million points = 100%) in the **Global Status Window**.

2.0 General Operating Instructions

2.1 Starting INTEGRATE

INTEGRATE was originally developed on the Silicon Graphics 4D models. While the CARD Laboratory at the time of this publication is using 5.3 version of the operating system, INTEGRATE should run on older versions of the operating system. To start the INTEGRATE program, login to the Silicon Graphics system, then type **integrate**. The screen will show the X and Y axes, and will indicate **No Active Object** in the Active Object Status Window in the lower left corner of the screen.

2.2 INTEGRATE Status Windows

INTEGRATE has 3 status windows across the bottom of the screen:

the **Active Object Status Window**,

the **Object Summary Window**, and

the **Global Status Window**.

2.2.1 The Active Object Status Window, located in the lower left corner of the screen, contains information about the current Active Object. Many INTEGRATE commands operate on the Active Object, so it is important to view the current status of an object before modifying it. Figure 1 provides an example of the Active Object Status Window.

```
SubjFile: 072_53p
LandFile: 072_53p.lnd
Active:1   Lon Thin:2   Lat Thin:2
Left:0  Right:512  Lower:0  Upper:256
Angles: X:   0.0  Y:   0.0  Z:   0.0
Center: X:   0.0  Y: 200.1  Z:   0.0
Offset: X:   0.0  Y:   0.0  Z:   0.0
```

Figure 1: Active Object Status Window

The following information appears in the **Active Object Status Window**:

- SubjFile: the name of the file containing the original data points.
- LandFile: the name of the file containing the landmark points.
- Active: the number of the Active Object (this object).
- Thin Factors: the number of longitudes and latitudes INTEGRATE skips when displaying the object.
- Corners: the Left and Right longitudes and the Lower and Upper latitudes of the subsection of the active object.
- Angles: the X, Y, and Z rotation angles from the original object position to the displayed object position.
- Center: the X, Y, and Z offsets to center the object in the axis system.
- Offset: the X, Y, and Z offsets to move the object from its original (centered) position to its displayed position.

2.2.2 The **Object Summary Window**, located in the lower right corner of the screen, lists every defined object, its file name, and its display status. This window is color-coded to help determine which image in the display area is associated with which object. Figure 2 provides an example of the Object Summary Window.

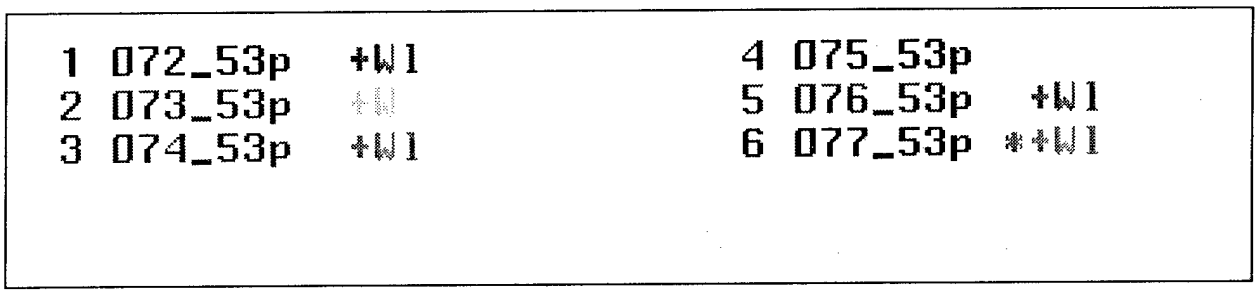


Figure 2: Object Summary Window.

Each object's summary appears in this order:

(object number) (subject file name) (display status)

The object number is the number to use to select that object for use in a command. The subject file name helps determine which object is to be selected, and the display status indicates the status of an object.

Table 1 below defines the symbols used in the display status line:

Symbol	Meaning
*	This is the active object.
+	This object is on the screen now, not hidden.
W	Wireframe display is on for this object.
P	Point display is on for this object.
S	Surface display is on for this object.
T	The surface display for this object is transparent.
l	Landmark display is on for this object.
c	Contour and circumference display is on for this object.

Table 1: Display status line definitions.

2.2.3 The **Global Status Window** is located between the active object status window and the object summary window. The global status window contains information about the INTEGRATE environment, such as eye position, pick mode, and clipping wall locations. Figure 3 provides an example of the global status window:

```
Walls:  100:1400:1300
Eye: X:0 Y:0 Z:700 Dist:700
Data Path: /hgu53p/
Memory Use:  2% (256K)
PICK OFF
```

Figure 3: Global status window.

The global status window contains the following information:

- Walls: Clipping Wall positions and the distance between them.
- Store: If Store is visible on line 1, measurement storage is enabled.
- R: If R is displayed on line 1, RGB (full color) mode is enabled.
- G: If G is displayed on line 1, GOURAUD shading is enabled in RGB mode.
- Eye: Eye position and distance with respect to the center of the coordinate system.
- Data Path: The prefix INTEGRATE adds to a load command file name to locate the file.
- Memory Use: A rough estimate of the percentage of the available memory being used to store object information (10 head scans=100%).
- Pick Mode: Pick mode is on or off.

2.3 Operating Features

INTEGRATE has a number of operating features that help the user manipulate displayed data. These include the **echo buffer**, use of the **up and down arrow keys**, and **point picking**.

2.3.1 The **Echo Buffer** is a section in the lower left corner of the screen which displays the commands as they are typed. The area immediately above the echo buffer displays the status of operations in progress,

reports operator errors, and displays command usage information for complex commands. The echo buffer also supports:

- the Home key (go to start of command),
- the End key (go to end of command),
- the Delete key (delete char at cursor),
- the Backspace (<-) key (delete char left of cursor),
- the Insert key (insert a blank at the cursor), and
- the left and right arrow keys (move cursor without changing text).

2.3.2 The **arrow keys** recall the previous command in the command history list to the echo buffer. This feature is circular; when the oldest available command is displayed, the **up arrow** cycles to a blank line, then repeats starting with the newest command. The **down arrow** key recalls the next command in the command history list to the echo buffer.

2.3.3 **Point Picking** consists of 3 steps:

1) enabling point picking (PICK ON),
2) selecting a pick mode (PICKMODE), and
3) picking points with the mouse by placing the cursor and clicking the left mouse button.

The **INTEGRATE** cursor is the same size as the pick region, so the points within the cursor boundaries will be picked and processed according to the pick mode when the left mouse button is pressed. In some pick modes, the center mouse button clears/resets the processing for that mode. For example, in Pick Mode CON3P, if the center mouse button is pressed after the second point is picked, the Pick Mode will be reset to restart CON3P picking, with the next point being used as point 1. The right mouse key brings up a "popup" menu which can be used in place of the keyboard for many of the INTEGRATE commands.

The available function keys are listed across the top of the screen. These keys are user-configurable through a file called fkey.tbl, which resides in the INTEGRATE directory. The INTEGRATE directory is

accessed through an environment variable called INTEGRATE. An example command creating this environment variable is: "setenv INTEGRATE /home/code/INTEGRATE." This command can be placed in the user's .login or .cshrc file so that it will be activated when the user logs in. If the INTEGRATE environment variable is not set, the MAN command will not work, since it uses the INTEGRATE variable to find the users_guide file, which contains a text version of this document.

3.0 Tutorials

INTEGRATE supports a great deal of functionality, but it is only with experience that the user will be able to take full advantage of the tools available. The following tutorials lead the user through some common INTEGRATE activities to demonstrate the process used to generate the end result.

The image data and batch files needed for these tutorials should be available with INTEGRATE version 1.25. A listing of the necessary files for each tutorial is found in Appendix A.

To see the end result of each tutorial, run the tutorial batch files. To run the batch file for the first tutorial, Basic Moves, type this command in INTEGRATE:

@tutorial_1

To run the batch file for the second tutorial, Registration Techniques, type:

@tutorial_2

Each tutorial has a corresponding batch file. To gain experience with the INTEGRATE commands, however, new users should execute each tutorial step by step, without using the batch files.

The tutorials are as follow:

Tutorial_1	Basic Moves
Tutorial_2	Registration Techniques
Tutorial_3	Point Picking
Tutorial_4	Feature Envelopes
Tutorial_5	Radial Difference Maps
Tutorial_6	Manipulating Whole Body Data (Cyberware WB4)
Tutorial_7	Manipulating Stereophotogrammetry Data

The tutorials are presented in a table format. Tutorial steps appear in the left column, and the commands used to carry out each step appear in the right column. Refer to section 4.0 Commands for additional information on how the commands work.

3.1 Tutorial One: Basic Moves

This tutorial introduces the user to the basic commands needed to manipulate the object on the screen. The user will learn to initially position the object, move the eyepoint, turn off and on landmarks, change the representation of the object from wireframe to surface, and other essential functions for visualizing the image.

The files needed for this tutorial are:

010_53p
010_53p.rgb
010_53p.lnd

Steps	Commands
1. Enter INTEGRATE and begin session.	integrate
2. Load the unhelmeted scan file of the subject.	cload 010_53p
3. Load the anatomical landmark file associated with this image.	lload 010_53p.lnd
4. Rotate the subject around the Y axis so he is directly facing you. Notice that the positive Y axis value causes counterclockwise rotation. This rotation initially places the object so that the "front" of his face is facing you. This coincides with the default eyepoint of INTEGRATE which is "front". All subsequent eyepoint commands (e.g. "back") will correspond with the object (e.g. back of the head). The axis and amount of rotation will depend on the orientation of the object when scanned.	rotate 0 75
5. Change the eyepoint to view the image from the right. Note that this does not MOVE the object, it only changes your viewpoint.	right
6. Change the eyepoint to view the image from the back.	back
7. Change the eyepoint to view the image from the left.	left
8. Change the eyepoint to view the image from the top.	top
9. Change the eyepoint to view the image from the bottom.	bottom
10. Change the eyepoint to view the image from the front.	front
11. Move the object 50 mm along the x axis. Notice that the object is moving relative to the screen.	move 50
12. Move the object 50 mm along the y axis. Notice that the object is moving relative to the screen.	move 0 50
13. Change the eyepoint to view the image from the right.	right

Steps	Commands
14. Move the object 50 mm along the z axis. Notice that the object is moving relative to the screen.	move 0 0 50
15. Move the object back to the original origin.	move -50 -50 -50
16. Change the eyepoint back to the front.	front
17. Trim away extraneous image noise at the top and bottom of the subject.	trim 0 0 0 -55 trim 0 0 55
18. Make a copy of the object and save it as object #2.	copy 1 2
19. Create voids in the image data.	ruin 1 2
20. Hide the second object.	hide 2
21. Fill in the voids on the image and the gap at the top of the subject's head. Notice that "1" was first selected to perform the operations on the first object.	1 do fill toupee 200 205
22. Turn off the axes and the status windows.	axes boxes
23. Turn the status windows back on and change the representation of the landmark locations from crosshairs to "L#" with the number denoting the landmark number found in the landmark file. See Appendix C.	alt_land
24. Turn on the landmark list to view the active landmarks and their coordinates in the object coordinate system.	landlist
25. Turn off the landmark list and turn on the help list.	landlist help
26. Turn off the help list and turn on the function keys listing. Turn off the landmarks.	help fkeys
27. Turn on the landmarks.	land
28. Turn off the wireframe mode and apply the surface routine to the object to show texture and color.	wireframe surface fullcolor 010_53p rgb
29. Calculate the volume and surface area of the object.	volume surface_area

3.2 Tutorial Two: Registration Techniques

This tutorial demonstrates registration techniques used to visualize a subject within a helmet system, as shown in Figure 4. This registration technique can be used for examination of a subject within any protective equipment item. As shown in Figure 4, an "x-ray" view is provided, allowing the designer to look inside the human-equipment interface.

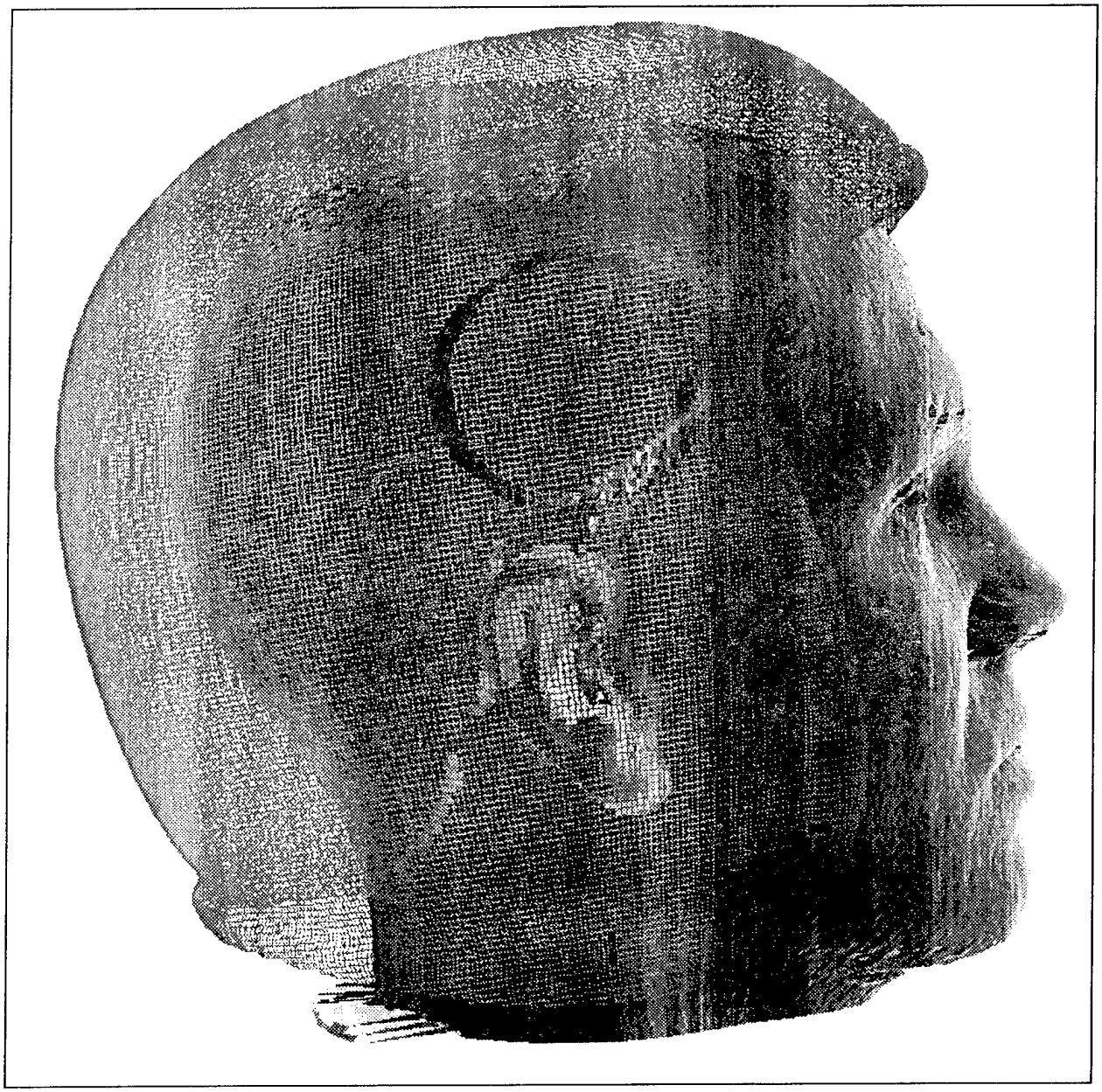


Figure 4: Registration of subject with helmet scan for visualizing subject/equipment interface.

Three image files are needed for the registration procedure:

1) one scan file of the subject with at least three visible anatomical landmarks,
2) one scan file of the same subject expertly fitted with a helmet system and showing at least the same three anatomical landmarks plus three reference landmarks on the helmet system, and
3) one scan file of just the helmet system with the same three helmet reference landmarks.

In this example, the landmarks have been identified and saved to a landmark (*.lnd) file for each image file. The *.rgb files are color files associated with each scan. The files used in this tutorial are:

010_53p, 010_53p.rgb, 010_53p.lnd
010_53ph, 010_53ph.rgb, 010_53ph.lnd
53psize5, 53psize5.rgb, 53psize5.lnd

Steps	Commands
1. Enter INTEGRATE and begin session.	integrate
2. Load the unhelmeted scan file of the subject.	cload 010_53p
3. Rotate the subject around the Y axis so he is directly facing you. Notice that a positive Y axis value causes counterclockwise rotation.	rotate 0 75
4. Trim away extraneous image noise at the top and bottom of the subject.	trim 0 0 0 -55 trim 0 0 55
5. Fill in voids on the subject and the gap at the top of the subject's head.	do fill toupee 200 205
6. Load the anatomical landmark file associated with this image.	lload 010_53p.lnd
7. Load the helmeted scan file of the same subject with his helmet donned.	cload 010_53ph
8. Trim the noise from this image.	trim 0 0 0 -50 trim 0 0 45 0
9. Load the landmark file associated with this image.	lload 010_53ph.lnd

Steps	Commands
10. Register the helmeted image with the unhelmeted image. Notice that the second image is rotated and translated into the coordinate system of the first image, and that lregister is used to align the scans as the common landmarks are anatomical landmarks.	lregister 2 1
11. Change the viewpoint to view the images from the right.	right
12. Use the walls command to "slice" through the data to examine the alignment of profiles.	walls 695 699 walls +5 walls +5 walls full
13. Change the eyepoint back to the front.	front
14. Load the helmet scan (and landmark file) for registration with the helmeted image file.	cload 53psize5 lload 53psize5.lnd
15. Register the helmet scan with the helmeted image file and view the alignment. Notice that zregister is used to align the scans as the common landmarks are auxiliary landmarks.	zregister 3 2 right walls 698 699 walls +5 walls +5
16. Hide the helmeted scan and show only the subject and the scan of the helmet alone. This final configuration illustrates the position of the subject within the helmet.	hide 2 walls full
17. Change the subject file to a surface and the helmet scan to a wireframe of lower resolution.	1 wireframe surface 3 thin 2 2
18. Change the helmet scan to a transparent surface.	wireframe surface transparent
19. Change the subject file to represent the color information.	fullcolor 010_53p rgb

Steps	Commands
20. View this configuration from different viewpoints.	front left back front

3.3 Tutorial Three: Point Picking

This tutorial demonstrates how to access and implement the point picking capability to generate a landmark file for the scan data. In this case, a head scan is loaded into INTEGRATE and the anatomical landmarking sequence initiated. This is a canned landmarking sequence that includes 42 head and face anatomical landmarks. The landmark selection order, shown in the global status window, has been established to allow the user to begin landmark selection on the right side of the head and progress around the head, working from top to bottom. The actual landmark file, however, lists the landmarks not in the order of selection, but in the order found in Appendix C. If a new landmark *picking order* is required, the command **new_order** can be used to establish an order for picking the points. Figure 5 shows a head scan file with anatomical landmarks.

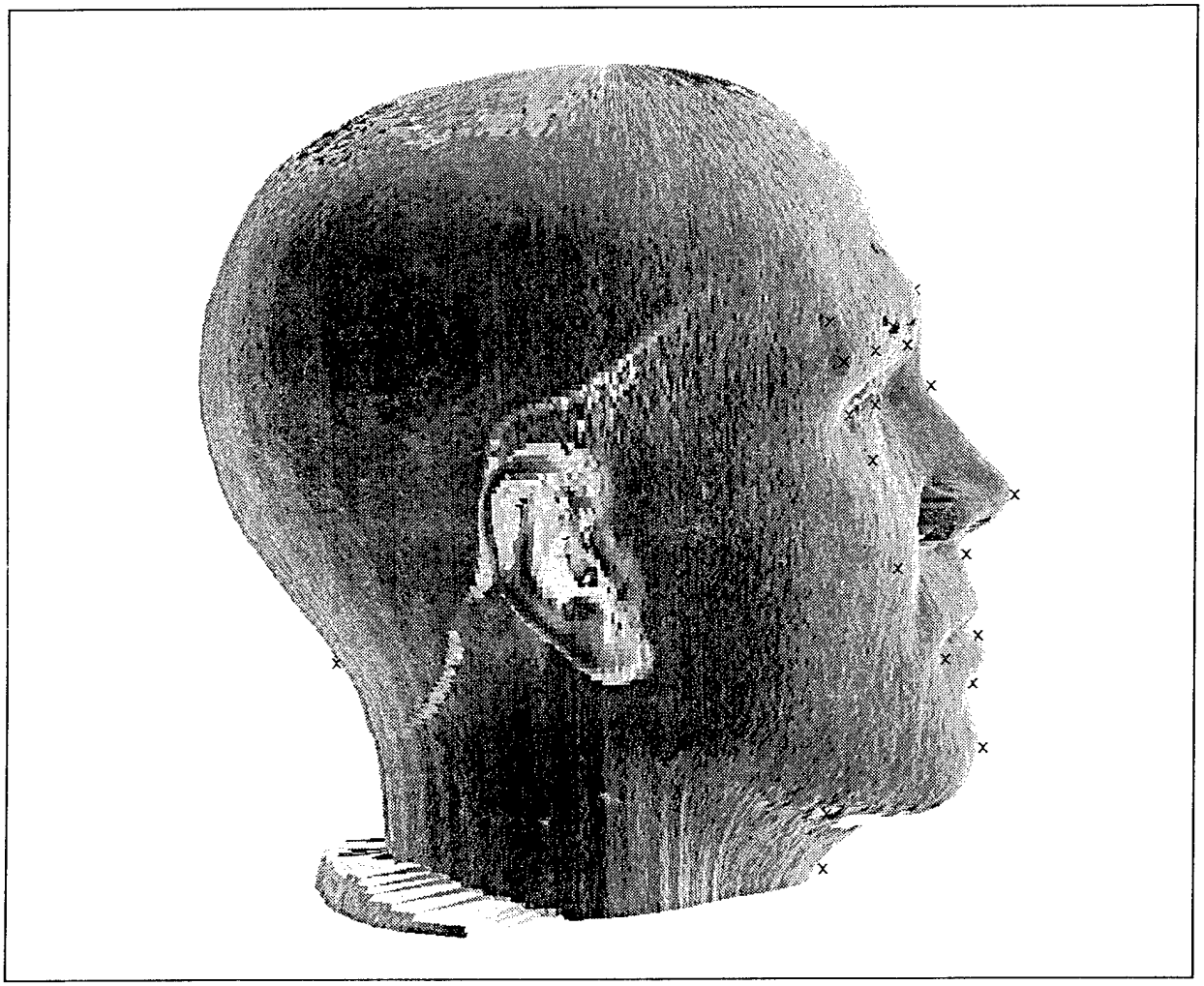


Figure 5: Subject scan with color file (monochrome for this publication) and marked landmark locations.

User-defined or arbitrary reference landmarks can be selected and stored in the landmark file using the command **pickmode aux_land.** A landmark file format is found in Appendix C.

Appendix B provides an illustration of the head and face anatomical landmarks. Refer to this figure during the landmarking process. For further clarification, definitions of the landmarks are also included in Appendix B. The landmark to be picked appears in the Global Status Window.

The files needed for this tutorial are:

010_53p
010_53p.rgb
010_53p.lnd

Steps	Commands
1. Enter INTEGRATE and begin session.	integrate
2. Load the head scan file.	cload 010_53p
3. Trim away extraneous image noise at the top and bottom of the subject.	trim 0 0 0 -50 trim 0 0 58
4. Fill in voids on the subject and the gap at the top of the subject's head.	do fill toupee 200 205
5. Change the surface from wireframe to color representation. Color mode clearly displays the color landmark dots.	wireframe surface fullcolor 010_53p rgb
6. Rotate the subject to the right to prepare for landmark selection.	rotate 0 75
7. Begin landmarking session.	pick on pickmode land

Steps	Commands
8. Use the mouse to put the cursor on the first landmark to be picked, the right tragion (near the inner ear; refer to the diagram in Appendix B). Click the left mouse button to select the landmark. Look in the global status window for the prompt that lists the next landmark to be picked. Pick several more landmarks.	
9. Skip backward in the landmark list to re-pick the last landmark. When the prompt in the global status window changes to the previous landmark name, re-pick that landmark.	skip -1
10. Save the landmark locations to a landmark file.	lwrite land_010.lnd
11. Display the landmark list to view the landmark coordinates.	landlist
12. Turn off the landmark list and turn the function key display back on.	landlist fkeys
13. Compare the selected landmarks with the standard landmark file for the subject. Load the subject file with its landmark file and rotate it into the same orientation as the original subject.	cload 010_53p 010_53p.lnd right rotate 0 75
14. Change the original subject back to wireframe to compare the landmarks.	1 surface wireframe
15. Change the viewpoint to examine the scans from different views.	front left back front

3.4 Tutorial Four: Feature Envelopes

This tutorial demonstrates how INTEGRATE can be used to generate feature envelopes for equipment items such as a helmet system. Feature envelopes describe the spatial location and orientation of areas of interest (i.e., features) with respect to a well defined, easily duplicated coordinate system. For a given helmet system, this definition could include the range of pupil location along all three coordinate axes or the volume which contains the aggregate of all ears for a given population.

These anthropometric design envelopes defined for an existing helmet are based on one critical factor: the relationship of the head to the helmet. Helmet systems do not fit the human head in exactly the same way across a sample of people. Figure 6 illustrates two subjects wearing the same helmet.

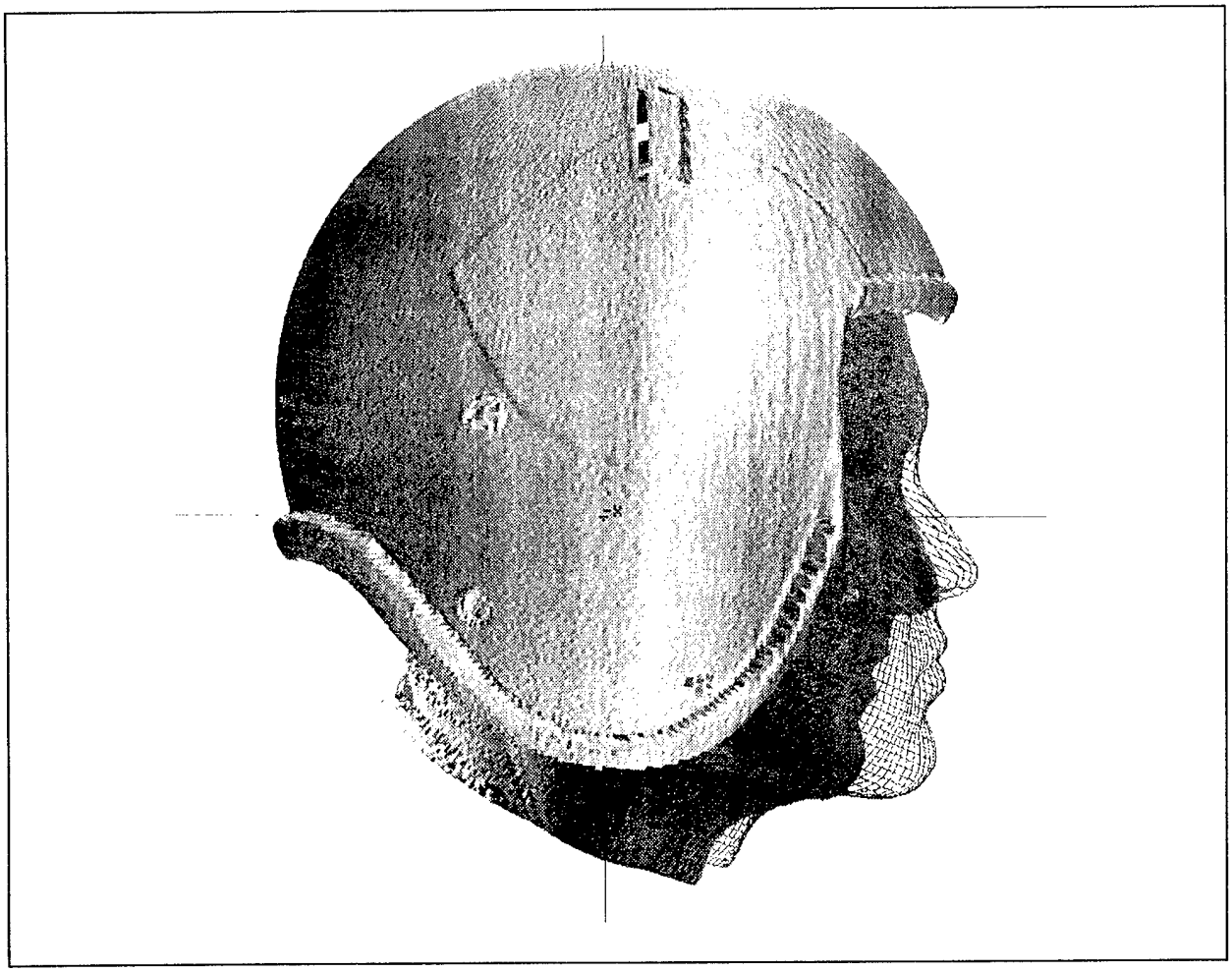


Figure 6: Two subjects wearing the same size and model helmet.

The orientation of the head with respect to the helmet system is entirely dependent on the shape of the helmet, the liner system, and the added peripherals, such as optics or earcups. All of these components must be fit optimally to the individual and, as a result, the helmet system "sits" on the head in a slightly different manner for everyone. In order to study these anthropometric design issues, researchers need surface scanning combined with the tools available in INTEGRATE. An example of the pupil envelopes of five subjects for a USAF helmet system is shown in Figure 7.

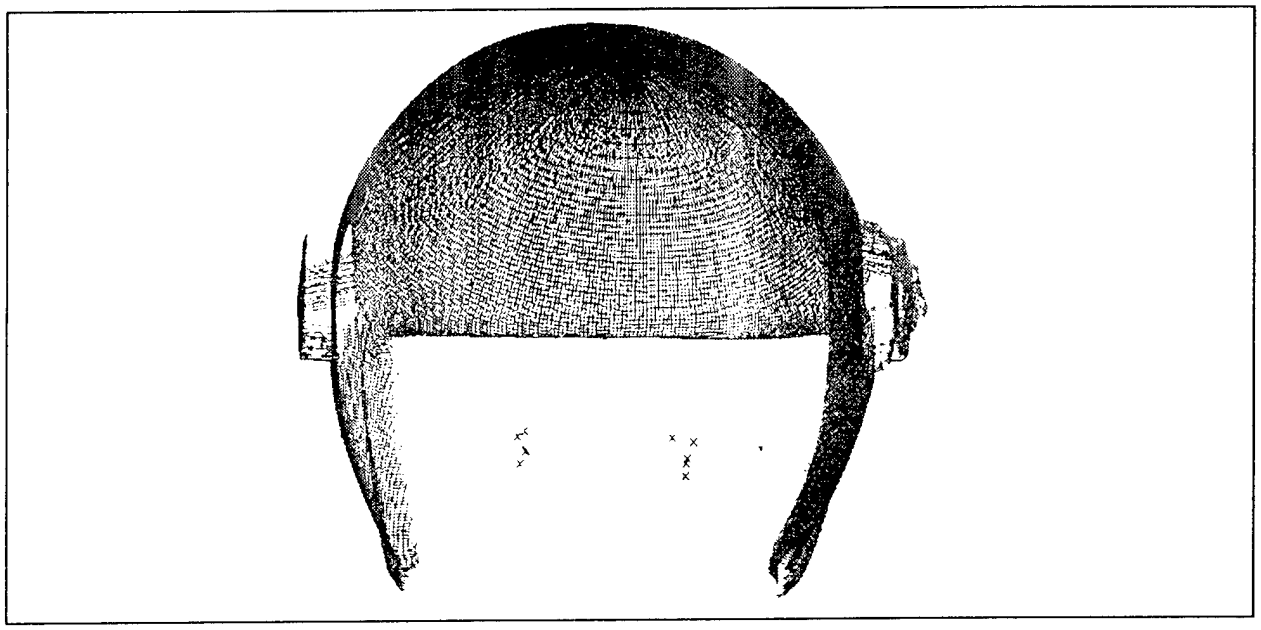


Figure 7: Pupil envelopes for five subjects in the same model and size helmet.

As in Tutorial 1, this tutorial consists of aligning an encumbered (helmeted) scan with that of a scan of the helmet alone and aligning the unencumbered (bare head) scan with that of the helmeted scan. This is performed by registration of the helmet landmarks found on the helmet scan with common landmarks found on the encumbered scan and registration of anatomical landmarks. The location of the subject can then be viewed with respect to the helmet coordinate system. Specifically, the locations of the pupils for each subject can be determined with respect to the helmet system. This is performed, in this tutorial, for a total of five subjects.

In this example, the landmarks have been identified and saved to a landmark (*.lnd) file for each image file. The *.rgb files are color files associated with each scan.

The following files are needed for this tutorial:

53psize5, 53psize5.rgb, 53psize5.lnd

100_53p, 100_53p.rgb, 100_53p.lnd

101_53p, 101_53p.rgb, 101_53p.lnd

102_53p, 102_53p.rgb, 102_53p.lnd

104_53p, 104_53p.rgb, 104_53p.lnd

105_53p, 105_53p.rgb, 105_53p.lnd

100_53ph, 100_53ph.rgb, 100_53ph.lnd

101_53ph, 101_53ph.rgb, 101_53ph.lnd

102_53ph, 102_53ph.rgb, 102_53ph.lnd

104_53ph, 104_53ph.rgb, 104_53ph.lnd

105_53ph, 105_53ph.rgb, 105_53ph.lnd

Steps	**Commands**
1. Enter INTEGRATE and begin session.	integrate
2. Load the helmet scan with the helmet reference landmarks.	cload 53psize5 53psize5.lnd
3. Rotate the helmet into a helmet-based coordinate system. This coordinate system is based on easily located, symmetric, consistent reference marks on the helmet.	align xz z1 z5 z3 z3
4. Load the first subject's encumbered (helmeted) scan with landmark files containing both helmet reference landmarks and anatomical landmarks.	cload 100_53ph 100_53ph.lnd
5. Register (align) this scan with the helmet scan using the common helmet landmarks.	zregister 2 1
6. Load the first subject's unencumbered (bare head) scan with anatomical landmarks.	cload 100_53p 100_53p.lnd
7. Register (align) this scan with the encumbered scan. In effect, this aligns the subject with the helmet scan.	lregister 3 2

Steps	Commands
8. Copy the pupil landmark locations to the helmet scan. This associates the pupil locations with respect to the global helmet system.	1 copyland l1 3 l34 copyland l2 3 l38
9. Load the second subject's encumbered (helmeted) scan with landmark files containing both helmet reference landmarks and anatomical landmarks.	cload 101_53ph 101_53ph.lnd
10. Register (align) this scan with the helmet scan using the common helmet landmarks.	zregister 4 1
11. Load the second subject's unencumbered (bare head) scan with anatomical landmarks.	cload 101_53p 101_53p.lnd
12. Register (align) this scan with the encumbered scan. In effect, this aligns the subject with the helmet scan.	lregister 5 4
13. Copy the pupil landmark locations to the helmet scan. This associates the pupil locations with respect to the global helmet system.	1 copyland l3 5 l34 copyland l4 5 l38
14. Load the third subject's encumbered (helmeted) scan with landmark files containing both helmet reference landmarks and anatomical landmarks.	cload 101_53ph 101_53ph.lnd
15. Register (align) this scan with the helmet scan using the common helmet landmarks.	zregister 6 1
16. Load the third subject's unencumbered (bare head) scan with anatomical landmarks.	cload 102_53p 102_53p.lnd
17. Register (align) this scan with the encumbered scan. In effect, this aligns the subject with the helmet scan.	lregister 7 6
18. Copy the pupil landmark locations to the helmet scan. This associates the pupil locations with respect to the global helmet system.	1 copyland l5 7 l34 copyland l6 7 l38
19. Load the fourth subject's encumbered (helmeted) scan with landmark files containing both helmet reference landmarks and anatomical landmarks.	cload 104_53ph 104_53ph.lnd

Steps	Commands
20. Register (align) this scan with the helmet scan using the common helmet landmarks.	zregister 8 1
21. Load the fourth subject's unencumbered (bare head) scan with anatomical landmarks.	cload 104_53p 104_53p.lnd
22. Register (align) this scan with the encumbered scan. In effect, this aligns the subject with the helmet scan.	lregister 9 8
23. Copy the pupil landmark locations to the helmet scan. This associates the pupil locations with respect to the global helmet system.	1 copyland l7 9 l34 copyland l8 9 l38
24. Load the fifth subject's encumbered (helmeted) scan with landmark files containing both helmet reference landmarks and anatomical landmarks.	cload 105_53ph 105_53ph.lnd
25. Register (align) this scan with the helmet scan using the common helmet landmarks.	zregister 10 1
26. Load the fifth subject's unencumbered (bare head) scan with anatomical landmarks.	cload 105_53p 105_53p.lnd
27. Register (align) this scan with the encumbered scan. In effect, this aligns the subject with the helmet scan.	lregister 11 10
28. Copy the pupil landmark locations to the helmet scan. This associates the pupil locations with respect to the global helmet system.	1 copyland l9 11 l34 copyland l10 11 l38

Steps	Commands
29. Hide all objects except the helmet scan with the pupil feature envelopes.	hide 2 hide 3 hide 4 hide 5 hide 6 hide 7 hide 8 hide 9 hide 10 hide 11
30. View the pupil envelopes for the helmet from different viewpoints.	right left back front

3.5 Tutorial Five: Radial Difference Maps

This tutorial demonstrates how a combination of INTEGRATE commands can be used to quantitatively evaluate the radial differences between cylindrical surface scans. Given two scans, the differences can be calculated along each radial value from a reference scan to a second scan. This is referred to as a Radial Difference Map (RDM). For this example, a total contact burn mask, or a full mask which covers the entire face, is compared to the original scan data of the subject for whom the mask was fabricated. A radial difference map indicates the degree of fit of this mask for this subject. Figure 8 is an RDM of a subject's head scan and a scan of his mask.

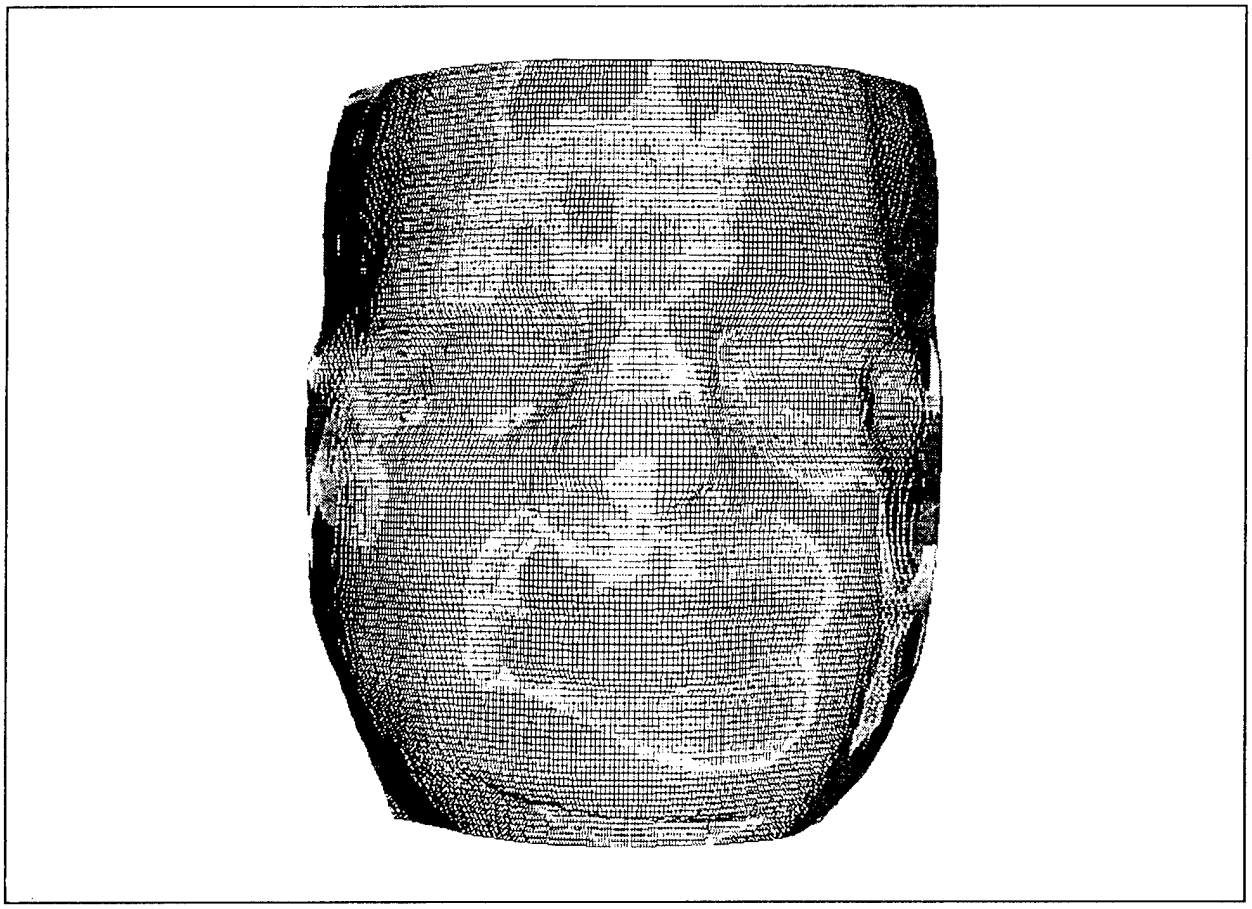


Figure 8: Radial Difference Map (RDM) of the total contact burn mask, with respect to the subject's face. Contrasting colors (monochrome for this publication) represent different degrees of fit.

NOTE: For this example, the two scans have been registered to align the surfaces, resampled to transform both into the new coordinate system, and trimmed to the same values. All of these steps are required before performing a radial difference map.

In this example, the landmarks have been identified and saved to a landmark (*.lnd) file for each image file. The *.rgb files are color files associated with each scan.

The following files are needed for this tutorial:

face, face.rgb

mask, mask.rgb

Steps	Commands
1. Enter INTEGRATE and begin session.	integrate
2. Load the head scan file.	cload face
3. Move the scan to the center of the screen.	move 0 200
4. Load the scan of the subject's mask and move it to the center of the screen.	cload mask move 0 200
5. Perform a positive subtraction of the mask with respect to the face. The second object, scan 2, will be replaced by the subtraction results.	possub 1 2
6. For the radial values of the face found to be greater than those of the mask, eliminate all difference values greater than 1 mm.	threshold 2 ge 1
7. Identify these radial values for the face scan. The results will be saved as the second object.	and 1 2
8. Perform these steps again using another scan file of the mask.	cload mask move 0 200
9. Perform a negative subtraction of the mask with respect to the face. The third object, scan 3, will be replaced by the subtraction results.	negsub 1 3
10. For the radial values of the face found to be less than those of the mask, eliminate all difference values greater than 1 mm.	threshold 2 ge 1

Steps	Commands
11. Identify these radial values for the face scan. The results will be saved as the third object.	and 1 3
12. Perform these steps again using another scan file of the mask.	cload mask move 0 200
13. Perform a positive subtraction of the mask with respect to the face. The fourth object, scan 4, will be replaced by the subtraction results.	possub 1 4
14. For the radial values of the face found to be greater than those of the mask, eliminate all difference values less than 1 mm.	threshold 4 lt 1
15. Identify these radial values for the face scan. The results will be saved as the fourth object.	and 1 4
16. Perform these steps again using another scan file of the mask.	cload mask move 0 200
17. Perform a positive subtraction of the mask with respect to the face. The fifth object, scan 5, will be replaced by the subtraction results.	negsub 1 5
18. For the radial values of the face found to be greater than those of the mask, eliminate all difference values less than 1 mm.	threshold 5 lt 1
19. Identify these radial values for the face scan. The results will be saved as the fifth object.	and 1 5

3.6 Tutorial Six: Manipulating Whole Body Data

This tutorial demonstrates visualization and manipulation of whole body scan data. The routine used to convert the Cyberware WB4 whole body data to an **INTEGRATE** compatible format is found in Appendix G. The format of the whole body image is considerably different from head scan data and sometimes requires different commands for manipulating this object. An example of this is "eyepoint." To view the entire image, the eyepoint is changed from the default value of 700 mm to about 3000 mm. This allows the user to visualize the whole body data within the bounds of the screen axis system. Commands to segment the whole body data are also demonstrated within this tutorial. See Figure 9.

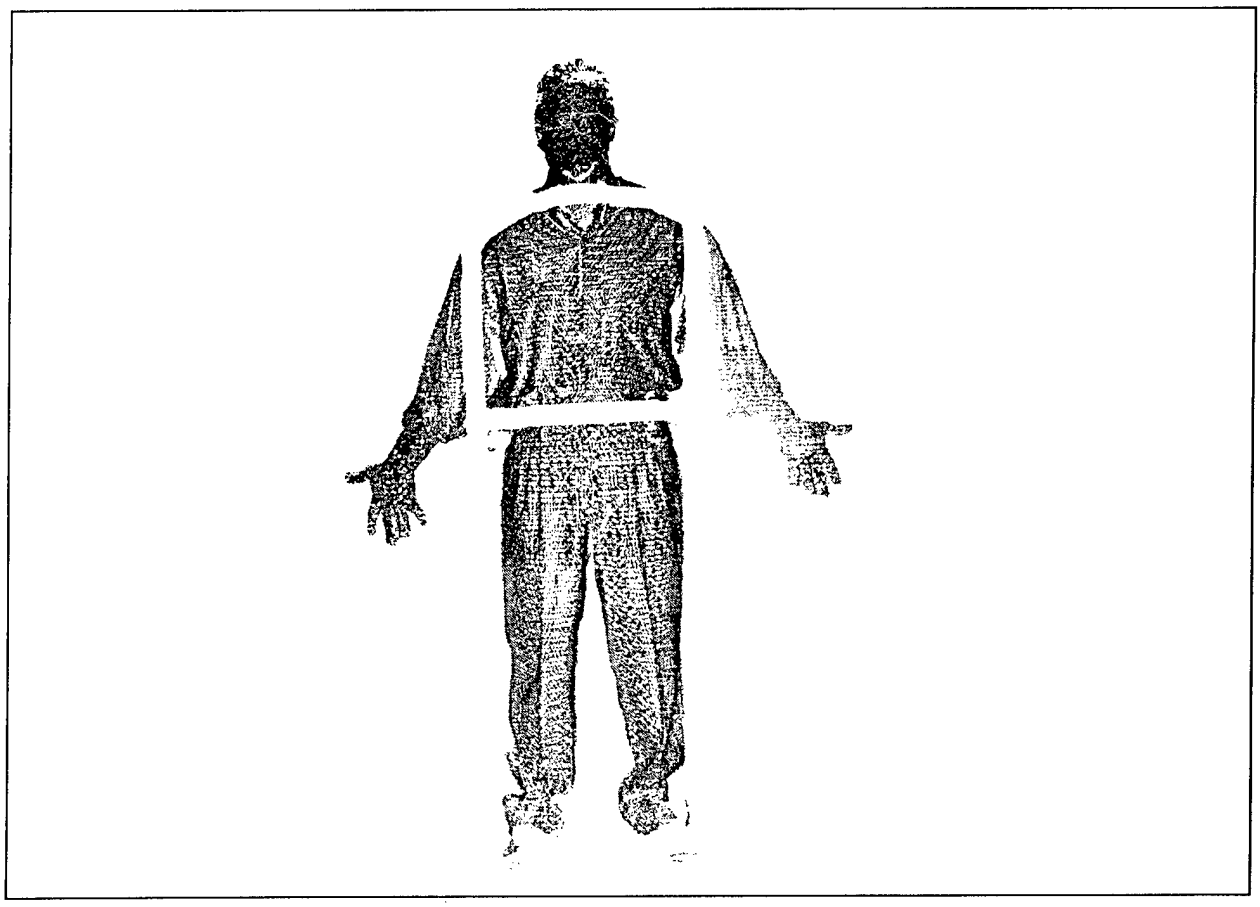


Figure 9: Shown in this figure is the full body scan data with the body segments separated from the torso.

The files needed for this tutorial are:

dr_boff.g

std.mtx

Steps	Commands
1. Clear the screen of axes, boxes, and function key commands.	axes off boxes off fkeys off
2. Expand the walls and change the eyepoint to allow full visualization of the whole body image.	walls 10 7000 eyedist 3000
3. Move the eyepoint to the front.	front
4. Turn the wireframe off and the points on.	option wireframe off option points on
5. Load the image file and a transformation matrix to position it in the middle of the screen. From this viewpoint, the subject is viewed from the side.	gload dr_boff.g mload std.mtx
6. To view the object from the front, the "eye" command is used.	eye 3700
7. Turn on the pick mode and move the subject forward slightly so that the entire object can be seen.	pick on move 100 70
8. Turn the points off and the surface on.	points off surface on
9. Turn the surface off and the points on. Notice that "off" and "on" are optional.	surface points
10. Segment the left arm. Note: object #1 is selected before performing the next operation.	movie_seg uz-234 1
11. Segment the right arm.	movie_seg lz260 1
12. Segment the head.	movie_seg ly610 1
13. Segment the torso.	movie_seg ly120 uy609 lz-233 uz259 1

Steps	Commands
14. Segment the legs.	movie_seg uy119 lz-233 uz259
15. Hide object #1.	hide 1
16. Move the body segments apart from the torso.	2 move 0 0 -50 3 move 0 0 50 4 move 0 50 6 move 0 -50

3.7 Tutorial Seven: Manipulating Stereophotogrammetry Data

This tutorial demonstrates the commands needed to manipulate a whole body scan derived from stereophotogrammetry as shown in Figure 10. Stereophotogrammetry was used in the 1970's to acquire whole body surface data (McConville, et al., 1980). Appendix F discusses the technique used to isolate segment data from the stereophotogrammetry image files. This tutorial allows the user to articulate the body segments as if they were rotating about the joint axes. Also demonstrated in this tutorial is the power of using "superobjects." By linking all of the body segments to the torso, the segments can either be manipulated independently or as a whole body.

Figure 10. Example of a male subject from the stereophotogrammetry survey in the seated position.

The files used in this tutorial are:

cs, link1, sit1

m2.1c, m2.2c, m2.3c, m2.4c, m2.5c, m2.6c, m2.7c, m2.8c, m2.9c, m2.10c, m2.11c, m2.12c, m2.13c, m2.14c, m2.15c, m2.16c, m2.17c, m2.18c, m2.19c, m2.1cs, m2.2cs, m2.3cs, m2.4cs, m2.5cs, m2.6cs, m2.7cs, m2.8cs, m2.9cs, m2.10cs, m2.11cs, m2.12cs, m2.13cs, m2.14cs, m2.15cs, m2.16cs, m2.17cs, m2.18cs, m2.19cs,

Steps	**Commands**
1. Enter INTEGRATE and begin session.	integrate
2. Turn the wireframe option off and the surface on.	option wireframe off option surface on
3. Load in the stereo data segment files. Notice that this can be quickly read in using the script file "cs" by typing "@cs m2" at the prompt.	cloads -a m2.1c cloads -a m2.2c cloads -a m2.3c cloads -a m2.4c cloads -a m2.5c cloads -a m2.6c cloads -a m2.7c cloads -a m2.8c cloads -a m2.9c cloads -a m2.10c cloads -a m2.11c cloads -a m2.12c cloads -a m2.13c cloads -a m2.14c cloads -a m2.15c cloads -a m2.16c cloads -a m2.17c cloads -a m2.18c cloads -a m2.19c

Steps	Commands
4. Link the segments of the stereo data to the torso using the super object command. First link the head and neck to the torso.	super link 2 1 super link 3 2
5. Link the right arm to the torso.	super link 7 8 super link 6 7 super link 3 6
6. Link the left arm to the torso.	super link 10 11 super link 9 10 super link 3 9
7. Link the right leg to the torso.	super link 14 15 super link 13 14 super link 12 13 super link 5 12
8. Link the left leg to the torso.	super link 18 19 super link 17 18 super link 16 17 super link 5 16
9. Link the lower torso to the upper torso.	super link 4 5 super link 3 4
10. Move the segments of the body into a seated position. Notice that segment 3 is the torso to which all other parts are anchored.	3 move 0 -80
11. Slightly bend torso at waist.	5 rotate -10
12. Rotate legs.	12 rotate -80 14 rotate 90 16 rotate -80 18 rotate 90

Steps	Commands
13. Rotate arms.	6 rotate 0 30 7 rotate -90 8 rotate 0 -100 9 rotate 0 -30 10 rotate -90 11 rotate 0 70
14. Turn the subject to the right and rotate the entire subject 360 degrees.	right rotate 0 30 (repeat this 11 times)

4.0 INTEGRATE Commands

This command list briefly describes the INTEGRATE commands in alphabetical order. Each description explains the use of the command and the required parts of the command. The Usage, Example, and Result section of each description demonstrates how to set up a command, what an actual command might contain, and what would happen in INTEGRATE if the example command was executed. In the Usage line, parameters that appear in parentheses () are required; parameters that appear in brackets [] are optional.

Executing commands

Many simple INTEGRATE commands can be executed with the function keys. The function key commands appear at the top of the INTEGRATE screen. If the function key list disappears, press function key F7 to display it again.

Simple commands can also be executed by pressing the right mouse key and selecting the command from the menu windows.

Commands that require additional parameters (for example, the distance to move an object on the screen) must be executed from INTEGRATE's command line. The command line is at the bottom left of the screen, just above the first blue information box, and is marked by a flashing cursor.

Toggle commands

Toggle commands, such as wireframe, surface, and land, turn INTEGRATE features on or off, like a light switch. For example, enter **wireframe** to display an object's wireframe and enter **wireframe** again to turn off the wireframe display.

Nobody's Perfect

INTEGRATE is a powerful software tool, but it has some limitations of which the novice user should be aware:

1. There is no "undo" command. Once an INTEGRATE command is entered, it has to run its course. The INTEGRATE operator should double-check each command before executing it.
2. INTEGRATE is not case-sensitive. Since INTEGRATE does not differentiate between upper and lower-case letters, the operator should type commands and file names with lower-case letters only.
3. INTEGRATE does not discriminate between file types. When the operator mistakenly attempts to load a landmark file with the command for loading an image file, INTEGRATE tries to execute the command. This can produce unexpected results, but it provides great flexibility for naming files.
4. Perspective commands (front, back, and side) sometimes need adjustment. When INTEGRATE loads an image file, the image may be in an awkward orientation. In order to make the front, back, and side commands work correctly, the operator should enter the front command and then use rotate commands to reorient the image. Once the image is oriented correctly for one perspective command, the other perspective commands should work also.

LIST OF COMMANDS

!
$
@
abssub
add
add_to_land
addobj
align
alt_land
and
avgland
axes
back
balltest
black
bottom
boxes
cd
center
centroid
circumference
cload
colors
comment
conclose
contour
contours
copy
copyland
copyseg
cursor
cwrite
cybermovie
delete
delland
delpnt
delseg
delta
derive
diff
dilate
displace
distance
do fill
do smooth
drawline
erode
exit
eye
eyedist
fcwrite
fill
filter
filtseg
fix_seam
fkeys
front
fullcolor
gcv
gload
gouraud
gwrite
help
hide
histogram
interpolate
jump
land
landlist
left
lload
lregister
lwrite
merge
man
median
mload
modland
move
move_vertex
movie_segment
mwrite
nameland
negsub
new_order
newland
option
ortho
pause
pick
pickmode
planes
ploads
points
pop
possub
print
push
pwrite
recolor
refresh
remark
resample
rgb
right
rotate
ruin
select
set
shade
show
side
skipx
sleep
smooth
split
store
subject
super
surface
surface_area
thin
threshold
top
tops
toupee
transparent
trim
volume
walls
white
wireframe
wload
wwrite
xload
xwrite
zload
zregister
zwrite

!<shell command>

The ! command prefix activates a UNIX shell command from within INTEGRATE, either from the command line or from a batch file. It automatically pushes the INTEGRATE window so the user can see other windows to observe actions resulting from the command. After the command is complete, INTEGRATE pops to the surface when the user presses any key.

$<shell command>

The $ command prefix activates a UNIX shell command from within INTEGRATE, either from the command line or from a batch file. It does not change any window configuration, and it does not wait for the user to press any key after the command is completed.

@<filename>

The @ command prefix activates a batch file specified by <filename>. It allows commands to be grouped into standard sequences to reduce mental gymnastics and repetitive typing.

A command file (without the @) can also be specified as part of the command line that starts INTEGRATE. For example: "integrate load52 spinfast" starts INTEGRATE, loads, trims, and rotates subject 52 (load52), then spins the viewer's eye around the object (spinfast).

Command files, also called batch files, can be parameterized (e.g. "@spinvar 2" which provides a parameter of 2 to the spinvar command file), and can provide a limited ability to support non-sequential operations such as looping or if-then-else constructs (see the JUMP command).

abssub

This command performs an absolute subtraction on two objects along each radial value. Specify a reference object and a replace object of the same size (that is, with the same trim values), and INTEGRATE subtracts one object from the other. The second object is subtracted from the first object, and the absolute value of the difference is retained. INTEGRATE stores the subtraction result in the replace object. It is recommended that the objects first be registered and resampled before this operation.

Usage: abssub (reference object) (replace object)
Example: abssub 1 2
Result: INTEGRATE subtracts object 2 from object 1 and stores the result in object 2.

add

This command performs an addition on two objects. The user specifies a reference object and a replace object of the same size (that is, with the same trim values), and INTEGRATE adds the objects together. INTEGRATE stores the addition result in the replace object.

Usage: add (reference object) (replace object)
Example: add 1 2
Result: INTEGRATE adds objects 1 and 2 and stores the result in object 2.

add_to_land

This command adds an XYZ offset in the screen coordinate system to the specified standard or auxiliary landmark of the active object. The user specifies a landmark number (for example, L2 for a standard landmark or Z2 for an auxiliary

landmark) and an offset for the X-axis. The user can also specify offsets for the y and z axes, but those parameters are optional. An example of this function would be to find the mid-point of the tragions (using the split command) while the head is aligned in the Frankfurt Plane axis system, then to add Beier's constant (8.3, 0, 31.2 mm) to convert the landmark to the approximate position of the Center of Gravity of the head.

Usage: add_to_land (Z#/L#) X Y Z

Example: add_to_land z2 8.3 0 31.2

Result: The auxiliary landmark z2 now represents the center of mass location.

addobj

This command replaces a section of the grid of one object with the same grid section of a second object. Addobj needs two parameters: the object to be modified and the object to be added to the specified object.

Usage: addobj (to object) (from object)

Example: addobj 1 2

Result: Object 1 is now a combination of objects 1 and 2.

align

This command aligns an object to the screen axis system according to three specified landmarks on the object. When the alignment is complete, the first specified landmark will be at the origin, the second landmark will be on the specified axis, and the third landmark will be on the specified plane. If a fourth landmark is specified, the object will be moved to the projection of the fourth landmark on the specified axis.

Align needs from four to nine parameters:

The first parameter is two or three lower case characters, which may be x, y, or z. The first character of the first parameter is the name of the axis to be defined by the first two landmarks.

The second character of the first parameter is the name of the axis perpendicular to the first axis, and on the plane defined by the first axis and the third landmark.

The third character of the first parameter is the name of the axis along which the object will be shifted if a fourth landmark is specified. If a fourth landmark is specified but a third character is not, INTEGRATE shifts the object along the axis defined by the first two points.

The second through ninth parameters specify landmarks or longitude/latitude coordinates. Points can be specified by landmark number, either standard (L) or auxiliary (Z), or by the longitude and latitude of the landmark. Three or four points must be specified. These points define the axis named above. If a fourth point is specified, INTEGRATE moves the center of the axis system along the axis named by the third character of the first parameter.

> Usage: align axis1axis2[axis3] land1 land2 land3 [land4]
> Example: align xy l1 l2 l3
> Result: INTEGRATE rotates the object into a coordinate system defined as follows: the X-axis passes through standard landmarks 1 and 2, the Y-axis is

perpendicular to the X-axis passing through standard landmark 3, and the Z-axis is defined as the cross product of the X and Y axes. The origin is defined by standard landmark L1.

alt_land [on/off]

This command toggles the landmark point display from L# or Z# form (landmarks individually labelled) to X or + form (landmarks marked but not labelled). The X/+ form reduces screen clutter when landmark labels are not needed. L# or X designate "standard" landmarks (e.g. Tragions or Infraorbitale), while Z# or + designate "auxiliary" landmarks, which are defined only for a specific study.

Usage: toggle command
Example: alt_land
Result: Landmarks change from x or + to L# or Z#.

and

This command performs a logical AND operation on two objects. Points with a value of zero in the objects' data are considered binary zeros, while non-zero values are considered binary ones. The user specifies a reference object and a replace object of the same size. The values of the reference object are stored in the replace object wherever the two objects AND to a binary one.

Usage: and (reference object) (replace object)
Example: and 1 2
Result: Object 2 is replaced by the radial values of object 1 at the non-zero radial locations of object 2.

avgland

This command averages the standard landmark sets from a

selected group of objects to produce a new landmark set which represents the centroids of corresponding landmarks. INTEGRATE attaches the new landmark set to the Active Object. The newly defined landmarks can be left as-is, meaning that they stay exactly where the are computed to be, or they can be projected onto the surface of the Active Object.

Avgland requires at least two parameters: (surf/asis) and a list of objects to be included in the average. Note: a single landmark set may be copied by using avgland with only one object.

Usage: avgland (surf/asis) obj1 obj2... object#
Example: avgland surf 1 2 3 4
Result: INTEGRATE averages the standard landmarks of objects 1, 2, 3, and 4 and projects the averages onto the surface of the active object.

axes [on/off]

This command turns the X, Y, and Z axes on or off.

Usage: toggle command
Example: axes
Result: The axes appear or disappear.

back

This command moves the user's "eye" to the back of the object.
Note: The object's coordinates do not change. When back is executed, it is as if the viewer moved behind the object to see the back of it. To change an object's coordinates, use move or rotate.

Back has one optional parameter: a distance. If the distance is positive, the viewer's eye will be positioned that much further than the default distance (normally 700; see eyedist) away from the object. If the distance is negative, the viewer's eye will be positioned that much closer to the object.

Usage: back [+/-number of mm]
Example: back 300
Result: The viewer sees the back of the object, 300 mm further away from it than before.

balltest

This command evaluates the accuracy of the Cyberware head scanner by comparing radii computed for the calibration ball with the true values. Balltest needs one parameter, the latitude to use for the radius compare.

Use **pickmode point** to select a latitude on the scan of the calibration ball. This is the latitude to specify in the balltest command.

Usage: balltest (latitude)
Example: balltest 125
Result: INTEGRATE computes the dimensions of the calibration ball and displays the dimensions in the lower left corner of the screen. The dimensions should match the actual dimensions of the ball.

black

This command sets the screen background color to black. Landmark and object points will change colors so that they will show up against the black background.

Usage: black

Example: black

Result: The screen background color turns black.

bottom

This command moves the viewer's eye to the bottom of the object.

Note: The object's coordinates do not change. When bottom is executed, it is as if the viewer moved under the object to see the bottom of it. To change an object's coordinates, use move or rotate.

Bottom has one optional parameter: a distance. If the distance is positive, the viewer's "eye" will be positioned that much further than the default distance (normally 700; see **eyedist**) from the object. If the distance is negative, the viewer's eye will be positioned that much closer to the object.

Usage: bottom [+/- number of mm]

Example: bottom

Result: The viewer sees the bottom of the object.

boxes [on/off]

This command turns the status boxes on or off. This can be useful for making snapshot/screen dumps or for increasing the available viewing area of the screen.

usage: toggle command

Example: boxes

Result: The status boxes at the bottom of the screen appear or disappear.

cd

This command changes the directory to be used for loading data files.

The format of the cd command is "cd <path>" where path is a standard UNIX path descriptor, such as "/spare/anthro/data/minisurvey".

Usage: cd (path)
Example: cd /headfiles/survey
Result: The directory for reading data files changes to /headfiles/survey.

center

This command moves the active object to put the specified point at the center of the axis system. There are two forms of the command: "center x y z," which names the coordinates of the point to be centered, and "center L#/Z#," which names the landmark (L for a standard landmark or Z for an auxiliary landmark) at the point to be centered.

Usage: center L#/Z#
Example: center z1
Result: The active object moves so that auxiliary landmark Z1 is at the center of the axis system.

centroid

This command computes the centroid of either the standard landmarks or the auxiliary landmarks and stores the result in the specified landmark location. Centroid requires two parameters: the landmark to hold the result and whether to compute the centroid of the standard landmarks (L or STD) or the auxiliary landmarks (Z or AUX).

Usage: centroid (Z#/L#)
(Z/L/aux/std)
Example: centroid z3 aux
Result: INTEGRATE computes the centroid of the auxiliary landmarks and stores the resulting centroid in landmark Z3.

circumference

This command computes a circumference line completely around the object where a plane specified by either three points or by two specified points and the center of the object intersects the surface of the object.

Usage: circum (L#/Z#/lg lt) (L#/Z#/lg lt) [L#/Z#/lg lt]
Example: circum z2 z3
Result: Integrate creates a line around the circumference of the object on the plane defined by the center of the object and auxiliary landmarks 2 and 3.

Circumference accepts its parameters in a variety of ways: any of the two or three specified points can be a longitude/latitude coordinate pair <lon> <lat>, a standard landmark number (L#), or an auxiliary landmark number (Z#).

If a single parameter is specified, INTEGRATE assumes it is either a latitude or a landmark to use as a latitude reference. The resulting circumference line is drawn along the specified latitude from the left trim limit to the right trim limit. The first point (leftmost longitude) is repeated at the end of the circumference to form a complete circle.

A circumference can also be generated by picking two or three points (pickmode cir2p or cir3p) using the mouse in point picking mode.

cload, cloads*

This command reads in a scan file. Cload has four optional parameters, one necessary file name, and one optional file name. The parameters are:

a, for an ASCII header;

b, for a binary header;

c, if there is a color file associated with the scan file and the color file is to be read in; and

n, if there is no color file or if you don't want to load the color file.

If any of these parameters are used, group them together and precede the first parameter with a "-" (dash). The current default is binary and no color (-bn).

After the parameters, if any, type the name of the scan file to be read in. After the scan file name, the user can choose to add the name of the landmark file associated with the scan file.

Usage: cload [-abcn] point_file [land_file]
Example: cload headscan headscan.lnd
Result: INTEGRATE reads in a scan file called headscan and its associated landmark file, headscan.lnd.

***cloads is used for loading in stereophotogrammetry segment data.**

colors

This command manipulates the object color table. The colors command can update the background colors or update the object colors.

Updating the background colors requires the following format:

COLORS 0 background text box

where:

background= the background color,

text= the text color,

box= the background color of the information boxes.

Updating the object colors requires the following format:

COLORS n bfeat bpnts wfeat wpnts

where:

n= the object number to change,

bfeat= the color for features, such as landmarks and contour lines, when the background is black,

bpnts= the color for points and wireframe when the background is black,

wfeat= the color for features, such as landmarks and contour lines, when the background is white,

wpnts= the color for points and wireframe when the background is white.

The available colors are:

black = 0	medium gray = 14
red = 1	bluish red = 15
green = 2	greenish red = 16
yellow = 3	bluish green = 17

blue = 4	reddish green = 18
magenta = 5	greenish blue = 19
cyan = 6	reddish blue = 20
white = 7	light red = 21
dim red = 8	light green = 22
dim green = 9	light yellow = 23
dim yellow = 10	light blue = 24
dim blue = 11	light magenta = 25
dim magenta = 12	light cyan = 26
dim cyan = 13	

Usage: colors object# bfeat bpnts wfeat wpnts

or

colors 0 background textbox

Example: colors 1 22 24 5

Result: Object 1 changes color. When the background is black, points and wireframe are light blue, and landmarks and contour lines are light green. When the background is white, landmarks and contour lines are magenta, and points and wireframe are dim blue.

comment

This command annotates the session audit trail with a text string. The comment appears in the INTEGRATE session record, stored in the directory from which INTEGRATE was launched.

Usage: comment {string}

Example: comment starting new session

Result: "starting new session" appears in the INTEGRATE session record.

conclose

This command closes a series of point-picked contours using pickmode MUL2P or pickmode MUL2A. After the contour points are picked, conclose creates the contour line between the first and last points picked.

Usage: conclose

Example: conclose

Result: INTEGRATE completes the connected contours by drawing a contour line between the first and last points picked.

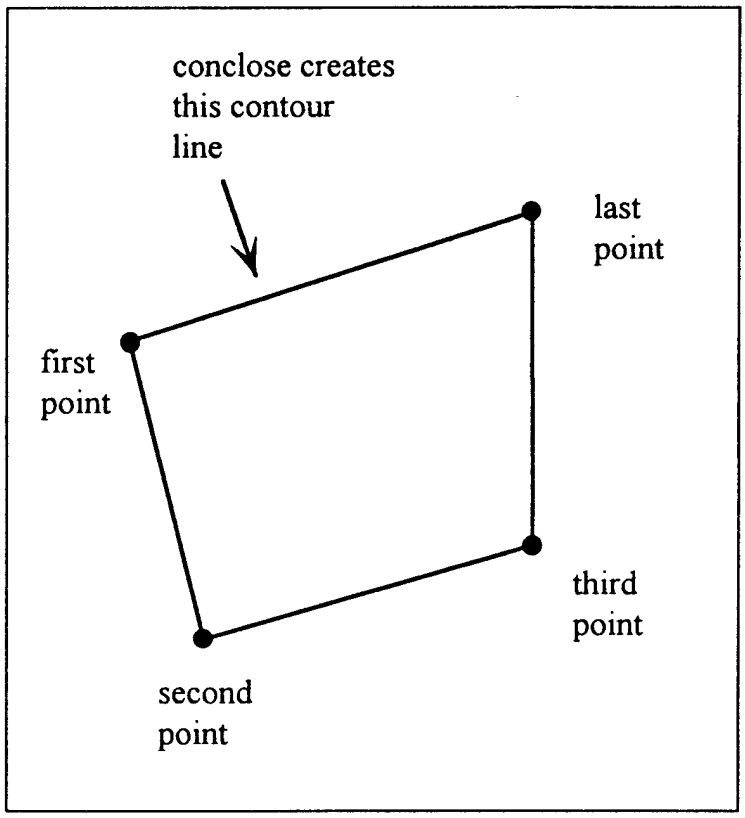


Figure 11: Conclose joining the first and last selected points.

contour

This command computes a contour line from one point to another using an optional third point or the object center to

establish the plane of the contour.

Usage: contour [L#/Z#/lg lt] [L#/Z#/lg lt] [L#/Z#/lg lt]

Example: contour z1 z2

Result: INTEGRATE draws a contour line on the surface of the active object that connects auxiliary landmarks Z1 and Z2 and passes through the object's center.

Contour accepts its parameters in a variety of ways: any of the two or three specified points can be a longitude/latitude coordinate pair <lon> <lat>, a standard landmark number (L<#>), or an auxiliary landmark number (Z<#>).

If a single parameter is specified, INTEGRATE assumes either a longitude or a landmark to use as a longitude reference. INTEGRATE draws the resulting contour line along the specified longitude from the lower trim limit to the upper trim limit.

A contour can also be generated by picking two or three points (**pickmode** con2p or con3p) using the mouse in point picking mode. Draw a connected series of two-point contours consecutively by using pickmode mul2p, then picking points. With this pickmode, the endpoint of one contour automatically becomes the startpoint of the next contour. Pickmode mul2a also creates a series of two-point contours, but instead of using the object center as in mul2p, it uses the Z-axis at the average latitude of the two points for its third point.

contours [on/off]

This command turns the display of contour lines on or off.

Usage: toggle command
Example: contours
Result: Contour lines appear or disappear.

copy

This command copies one object to another, eliminating the points trimmed from the first object and keeping only the points designated by the first object's thin factor.

Copy accepts one optional parameter: the number of the object to be copied. If no object number is specified, the active object is copied. Copy creates a new object in the first available object pool slot.

Usage: copy object
Example: copy 3
Result: INTEGRATE creates a copy of object 3.

copyland

This command copies one or more landmarks from one object to another. Copyland can consolidate landmarks from two or more sets into a single set. An example might be the consolidation of the Tragions from a subject data set with the landmarks from a helmet scan to allow analysis of the relationship of Tragions to helmet position.

Usage: copyland (Z#/L#) (from object) (Z#/L#) [count]
Example: copyland l1 2 l2 3
Result: INTEGRATE copies landmarks 2, 3, and 4

from object 2 and stores them in landmarks 1, 2 and 3 on the active object.

Copyland operates on the active object and requires three parameters: the destination landmark number (Z# or L#), the source object number, and the source landmark number (Z# or L#). An optional 4th parameter specifies the number of consecutive landmarks to copy. If the 4th parameter is not specified, INTEGRATE only copies the source landmark.

copyseg

This command copies the area of an object bounded by specified contours to a new object. Copyseg needs a list of contours which describe the segment to be copied. Each contour displayed on an object is numbered at approximately the midpoint of the contour. Use this number in specifying contours for copyseg. Two special contours, t and b, are used to specify that the top of the object or the bottom of the object is to be used as a contour boundary. "Copyseg t b" would copy the entire active object (as thinned and trimmed). Copyseg creates a new object in the first available object pool slot.

Usage: copyseg contour1... contour#
Example: copyseg 1 2 3
Result: INTEGRATE makes a copy of the segment bounded by contours 1, 2, and 3.

cursor [on/off]

This command enables or disables a surface tracking cursor for the active object. The surface tracking cursor is a crosshair that conforms to the contours of the surface of the object.

Note: When cursor is executed, INTEGRATE redraws the screen each time the cursor moves. To avoid redraw delays, use the cursor command with wireframe display, and trim the object before executing cursor.

Usage: toggle command
Example: cursor
Result: The surface tracking cursor appears or disappears.

cwrite

This command writes out a (new) Cyberware-format file. Cwrite requires a new filename as a parameter. It always writes the data with the new/modified ASCII header.

Usage: cwrite filename
Example: cwrite face_scan
Result: INTEGRATE writes the data in the active object to a new file called face_scan. INTEGRATE stores the file in the directory from which INTEGRATE was launched.

cybermovie

This command copies a cyberware object into a new object, converting it to a MOVIE.BYU representation. Cybermovie works on the active object only, and requires one parameter: whether the object is completely closed (WRAP) or is a partial surface (NOWRAP). This parameter determines whether INTEGRATE creates polygons to connect the object's last longitude to the first longitude.

Usage: cybermovie (wrap/nowrap)

Example: cybermovie wrap

Result: INTEGRATE creates a copy of the active object in MOVIE.BYU format. INTEGRATE stores the file in the directory from which INTEGRATE was launched.

delete

This command removes one or more objects from the object pool or one or more sub-objects from an object. Delete needs one parameter, which is either the number of the object to be removed or a range of objects to removed (e.g. DELETE 1-10). After objects have been removed, new objects can be read in to replace them.

If two or more parameters are specified, the first is the object (or objects) to be modified, and the subsequent parameters are the sub-objects to be deleted from the specified object. The object itself is not removed when two or more parameters are specified.

Usage: delete [object]

Example: delete 3

Result: INTEGRATE removes object 3 from the object pool.

delland

This command deletes a landmark value from the standard landmark list. The slot in the list remains, but the coordinates of the landmark are zeroed. Delland requires one parameter, the standard or auxiliary landmark number of the landmark to delete (L# or Z#).

Usage: delland (Z#/L#)

Example: delland z13

Result: INTEGRATE deletes the landmark value stored in auxiliary landmark Z13.

delpnt

This command deletes (voids) one point from the data set. Delpnt accepts its parameter(s) in a variety of ways: the point can be specified as a longitude/latitude coordinate pair <lon> <lat>, a standard landmark number (L#), or an auxiliary landmark number (Z#). Points can also be deleted by picking points with the mouse when pickmode is set to delpnt.

Usage: delpnt (Z#/L#) or delpnt <lon> <lat>

Example: delpnt z24 or delpnt 189 56

Result: INTEGRATE deletes the point at auxiliary landmark Z24 or the point at longitude 189, latitude 56.

delseg

This command deletes (voids) all of the points within the boundaries of a specified set of contours. Delseg needs a list of contours which describe the segment to be deleted. Each contour displayed on an object is numbered at approximately the midpoint of the contour. Use this number in specifying contours for delseg. Two special contours, t and b, are used to specify that the top of the object or the bottom of the object is to be used as a contour boundary. "Delseg t b" would delete all of the points in the entire active object (as thinned and trimmed).

Usage: delseg contour1... contour#

Example: delseg 2-6

Result: INTEGRATE deletes the region defined by contours 2 through 6 from the active object.

delta

This command colors the surface of an object according to its distance from a reference object. Points on the object with larger radii are shown in yellow, fading to red with increasing distance. Points on the object with smaller radii are shown in cyan, fading to blue with increasing distance. Delta needs one parameter: the number of the reference object. It always operates on the active object. If no reference object is specified, the distance for color-coding is computed from the mean radius of the active object.

Usage: delta reference_obj
Example: delta 3
Result: The color of the active object changes to reflect its distance from object 3.

derive

This command derives a missing landmark from a second landmark and the resultant from combining the two landmarks. Derive needs three parameters and has two optional parameters. The required parameters are a destination landmark number (Z# or L#), the resultant landmark number (Z# or L#), and the other landmark used to produce the resultant (Z# or L#). You can specify an optional total weight and second landmark weight if the resultant was produced with unequal weights.

Suppose you want to create a new landmark, Z3, between landmarks L1 and L6.

• L1 • Z3 • L6

You would use **split** to create Z3, the resultant landmark. If data is lost and L6 disappears

• L1 • Z3

you can recreate it with **derive**. To recreate L6, type

derive L6 Z3 L1

If you used weighting in the split operation, you could use weighting in the derive command, also.

Usage: derive (Z/L destination landmark#) (Z/L landmark 1) (Z/L landmark 2) [weight 1] [weight 2]

Example: derive z5 L28 L14 5 .5

Result: INTEGRATE recreates landmark Z5, using 5 as a total weight and 0.5 as a second landmark weight.

diff

This command displays and prints the difference between the rotation angles of two scans. The displayed difference is the difference to rotate one scan into the axis system of the other scan. This command ignores any offset differences due to different centers of rotation. It requires two parameters: the "final" object and the "beginning" object. For example, DIFF 1 2 provides the angles necessary to rotate object 2 into the same orientation as object 1. The display and print show the angular rotation around the X axis, the Y axis, and the Z axis. If the rotation around the Z axis is significant, the other two angles may be slightly in error.

Usage: diff obj1 obj2

Example: diff 1 2

Result: INTEGRATE displays and prints the angles needed to rotate object 2 into the same orientation as axis 1.

dilate

This command performs a morphological dilation on an object. After an object has been **eroded**, dilate completes the smoothing process. Dilate expands the data so that a structuring element with an origin placed at the original data fits. The structuring element used in this instance is a cylinder with a spherical top. The user specifies the radius of the sphere or both the cylinder and the sphere. (If only the sphere is specified, the cylinder is set to the same radius.) The default dilation operation is positive. A negative dilation can be performed by adding the modifier "minus."

Usage: dilate (sphere size) [cylinder size] [MINUS]
Example: dilate 5 2
Result: If erode has already been executed, INTEGRATE smoothes the active object.

displace

This command applies the present displacement matrix for a MOVIE.BYU object to each point, then resets the displacement matrix to the identity matrix (no rotations or translations). This allows a permanent change of axis system when the object is written out (see GWRITE).

Usage: displace object
Example: displace 3
Result: The next time object 3 is loaded, it will appear in the same position it was in when displace was executed. INTEGRATE assigns that position

to the object.

distance

This command computes the total surface distance along either contours or circumferences of the active object. Distance requires a list of contours or circumferences to be measured. Each contour displayed on an object is numbered at approximately the midpoint of the contour. Use this number in specifying contours. Distance computes the contour distance for each contour and displays the sum of all the distances.

Usage: distance contour1... contour#
Example: distance 3 5
Result: INTEGRATE displays the surface distance of contours 3 and 5.

do fill

This command replaces void points in the active object with an approximation based on surrounding points.

Usage: do fill
Example: do fill
Result: INTEGRATE fills in missing points on the active object.

do smooth

This command replaces each point in the active object's data set with the average value of the point and its neighbors, resulting in smoother data surfaces.

Usage: do smooth
Example: do smooth
Result: INTEGRATE smoothes the surface of the

active object.

drawline

This command draws a straight line from one landmark through another landmark, with an optional length specified. Drawline requires two parameters: the landmark at the origin of the line (Z# or L#), and a landmark that the line is to pass through. An optional length parameter specifies the length of the line. If length is not specified, the line ends at the 2nd landmark.

Usage: drawline (L#/Z#) (L#/Z#) [length]
Example: drawline L1 L32
Result: INTEGRATE draws a line connecting landmarks 1 and 32.

erode

This command performs a morphological erosion on an object. Erosion shrinks the data so that the origin of a structuring element fitted within the data becomes the new location for a data point. The structuring element used in this instance is a cylinder with a spherical top (see figure 10 below). The user specifies the radius of the sphere or both the cylinder and the sphere. If only the sphere is specified, the cylinder is set to the same radius. The larger the sphere and cylinder, the greater the erosion that occurs. The default erosion operation is positive. A negative erosion can be performed by adding the modifier "minus". (An opening is an erode followed by a dilate.)

Usage: erode (sphere size) [cylinder size] [minus]
Example: erode 4 5
Result: INTEGRATE takes the "sharp edges" off the

data. To complete the smoothing process, execute **dilate.**

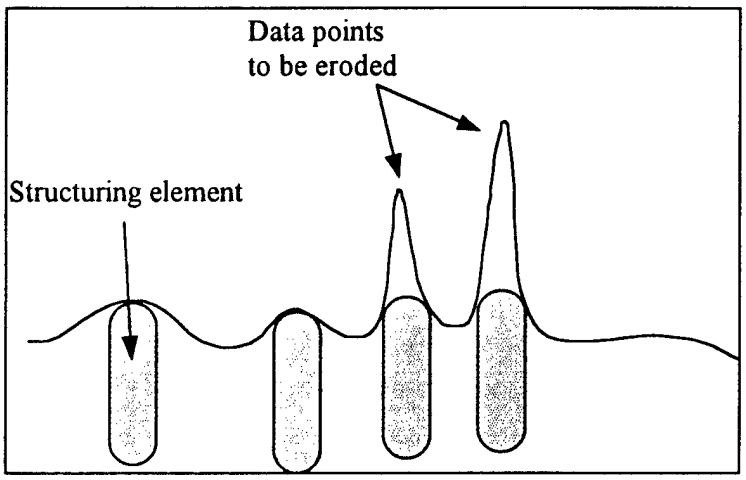


Figure 12: Erosion of surface data.

exit

This command ends INTEGRATE. Pressing F12 or selecting the exit command from the right mouse key menu also ends INTEGRATE.

Usage: exit

Example: exit

Result: The INTEGRATE session ends when the user presses Shift Y.

eye

This command changes the perspective from which the user views the active object. Eye does not change the object's position in the axis system; rather, the viewer's "eye" moves to the front, side, top, or bottom of the object.

Eye needs three parameters, the X, Y, and Z location of the eye. Eye 0 700 0 puts the viewer's eye on top of the object. Eye 0 0 700 puts the viewer's eye in front of the object. Eye 700 0 0 puts the viewer's eye beside the object.

Usage: eye (distance along X in mm) (distance along Y in mm) (distance along Z in mm)

Example: eye 0 0 300

Result: Your "eye" moves 300 mm along the Z axis, much closer to the active object.

eyedist

This command resets the default eye distance to the specified distance. The default distance is set at 700. When it is changed, the **front, back, left, side, right, bottom**, and **top** commands use the new distance as the default distance for computing eye position.

Usage: eyedist (distance)

Example: eyedist 300

Result: The eye distance is set to 300. When a perspective command (front, right, top, etc.) is executed, the viewer's "eye" is 300 mm from the object.

fcwrite

This command writes out an ASCII fullcolor (24-bit) file. It requires a single argument which is the base name of the file to be written to. The suffix ".color" is appended to the base name.

Usage: fcwrite filename

Example: fcwrite head

Result: A file called head.color is created which

contains the green, and blue color components for each vertex.

fill [on/off]

This command enables or disables automatic void fill for an object after a command, such as cload or resample, which might create new voids in the data.

Usage: toggle command
Example: fill
Result: INTEGRATE's fill function is enabled or disabled.

filter

This command filters the data with one of the INTEGRATE smoothing filters. Select a type of smoothing filter and a scale factor to determine the strength of the filter. (The larger the scale, the larger the number of adjacent points involved in the filter function.) Options are: GAUSSIAN, DISCRETE, or GREEN filters. The filter may be applied latitudinally, longitudinally or in both directions. Note: When using **filter** on a trimmed area, points outside the area are used in calculations. This may result in shrinkage from the rest of the data. See **filtseg**.

Usage: filter (GAUSS/DISCRETE/GREEN) scale (LAT/LON/BOTH)
Example: filter gauss 3 lat
Result: INTEGRATE smoothes the active object.

filtseg

This command is identical to the filter command except that the edge of a trimmed area is replicated and used in place of data outside the area. This helps to prevent shrinkage.

Usage: filtseg {GAUSS/DISCRETE/GREEN} scale {LAT/LON/BOTH}

Example: filtseg gauss 3 lat

Result: INTEGRATE smoothes the active object.

fix_seam

This command corrects any mismatch between the sides of the seam where the end of the data set meets the beginning. The mismatch is caused by subject movement during the scan. Fix_seam operates on the active object only. Note: For best results, make sure the object's trim boundaries correspond with the physical seam.

Usage: fix_seam (active object)

Example: fix_seam

Result: INTEGRATE corrects seam mismatch in the active object.

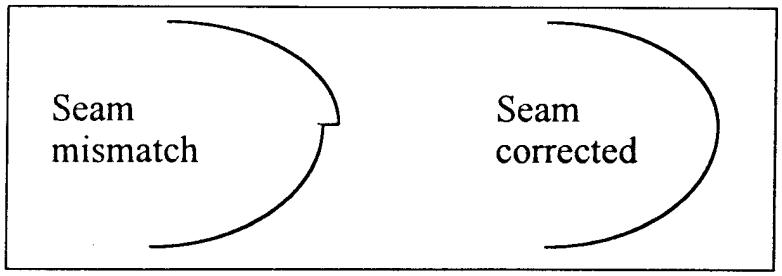


Figure 13: Seam correction with fix_seam.

fkeys

This command turns the function key display on or off. It works the same as the other toggle commands.

Usage: toggle command

Example: fkeys

Result: The function key display at the top of the screen appears or disappears.

front

This command moves the viewer's "eye" to the front of the object. Front has one optional parameter: a distance. If the distance is positive, the viewer's eye will be positioned that much further than the default distance (700 mm) from the object. If the distance is negative, the viewer's eye will be positioned that much closer to the object.

Usage: front [distance]
Example: front 300
Result: The viewer is now looking at the front of the object from a distance of 300 mm.

fullcolor

This command allows the use of all available color information (up to eight bits each of red, green, and blue) instead of the abbreviated color (four bits of red and green, three bits of blue) normally available. Fullcolor requires one parameter: the base file name of the color file. INTEGRATE automatically adds a .color or .rgb extension to the filename you specify. The data path is also added to the filename. Fullcolor has two optional parameters: min and max. If min and max are specified, the color information is modified as follows: for each color axis (r, g, b) all values below min are set to 0, all values above max are set to 255, and all values between min and max are rescaled to the range 0 to 255. For example, FULLCOLOR 001_53P 0 128 tells INTEGRATE to read all color information from file 001_53P.RGB (or 001_53P.COLOR) and rescale all values from 0 to 128 to

the range 0 to 255 (values of 128 become 255, values of 64 become 128, etc.).

Usage: fullcolor base_color_file_name [min max]
Example: fullcolor 101_53p
Result: INTEGRATE reads all color information from file 101_53p.

gcv

This command makes a logarithmic plot of the Generalized Cross Validation equation for head scan data. Generalized Cross Validation is a method for determining the best discrete Gaussian filter scale for the given data. The minimum value of the plot is a conservative estimate for the best scale. INTEGRATE also calculates other statistics (mean, standard deviation) to help determine the appropriate scale.

Usage: gvc [defaults] or [start steps_decade total_steps]
Example: gvc
Result: INTEGRATE makes a logarithmic plot of the GCV equation for the active object.

gload

This command reads in a MOVIE.BYU file primarily for display. Gload takes one argument: the name of the MOVIE.BYU file to be loaded.

Usage: gload movie_file [land_file]
Example: gload body_scan.g
Result: INTEGRATE loads body_scan.g.

gouraud

This command toggles gouraud shading on or off while in

RGB mode (see the rgb command). Gouraud shading presents a smoother-looking image.

Usage: toggle command
Example: gouraud
Result: The active object appears with gouraud shading in RGB mode.

gwrite

This command writes out a MOVIE.BYU file. Gwrite takes one argument, the name of the MOVIE.BYU file to be written out.

Usage: gwrite movie_file
Example: gwrite body_scan.g
Result: INTEGRATE writes body_scan.g to the directory from which INTEGRATE was launched.

help [on/off]

This command turns the list of available commands on or off.

Usage: toggle command
Example: help
Result: The command list appears or disappears.

hide

This command temporarily removes an object from the screen without removing it from the object pool. Hide has one optional parameter: the number of the object to hide. If an object number is not specified, INTEGRATE hides the active object.

Usage: hide [object number]

Example: hide 5

Result: Object 5 disappears, but remains in the object pool.

histogram

This command creates a histogram of an object. The histogram is limited to ten equally spaced, user specified intervals. INTEGRATE stores the histogram in a file called histogram.dat. INTEGRATE then activates jot, SGI's window-based, full-screen editor. Jot displays the histogram so it can be edited.

Usage: histogram obj interval

Example: histogram 1 30

Result: A jot window appears containing the histogram of object 1 in intervals of 30 mm.

jump

This command acts as a "go to" command in a batch file. Jump can be dependent on a condition. Conditions currently supported are: always (always jump), count (jump a specified number of times), and smooth (jump based on iterative smoothing criteria).

Jump needs two parameters: the condition for the jump, and the comment line in the batch file to go to.

Usage: jump condition [comment identifier]

Example: jump count 5 * start here

Result: When INTEGRATE reaches the jump command in the batch file, INTEGRATE goes to the line containing "* start here" and begins executing commands at that point. INTEGRATE executes the

jump five times.

land [on/off]

This command displays or hides the landmark points for the active object (if landmarks have been read in for this object).

Usage: toggle command
Example: land
Result: The active object's landmarks appear or disappear.

landlist [on/off]

This command turns the list of standard landmarks on or off. If an Active Object is selected and has assigned landmarks, the "world" coordinates of the standard and auxiliary landmarks for the Active Object will also be displayed when landlist is executed.

Usage: toggle command
Example: landlist
Result: The landmark list appears or disappears.

left

This command moves the viewer's "eye" to the left side of the active object.

Usage: left [distance]
Example: left
Result: The viewer is now looking at the left side of the object.

lload

This command loads a landmark file for the active object. Lload needs one parameter: the name of the landmark file to be loaded.

Usage: lload (landmark file name)

Example: lload 010_53p.lnd

Result: INTEGRATE loads the 010_53p.lnd landmark file.

lregister

This command registers (aligns) an object to another object by least-squares fitting of corresponding standard landmarks.

Lregister needs two parameters: the number of the object to be registered and the reference object number.

Usage: lregister obj ref_obj

Example: lregister 3 2

Result: INTEGRATE registers object 3 to object 2, effectively translating and rotating object 3 into the position of object 2.

lwrite

This command writes a new landmark file. Lwrite has one required parameter and one optional parameter. The name of the landmark file to be written is required. Rotate (or just r), the optional parameter, rotates and translates the XYZ coordinates of the object.

Usage: lwrite file_name [rotate]

Example: lwrite new_landmarks.lnd r

Result: INTEGRATE writes the landmark file new_landmarks.lnd to the directory from which INTEGRATE was launched, rotating and translating the object's XYZ coordinates.

man

This command displays this manual. To turn the manual off, type :q and PRESS ENTER.

Usage: man

Example: man

Result: This manual appears in a window on your screen.

median

This command replaces a point with the median value of the points in its neighborhood. This helps eliminate data spikes. Median requires one parameter: the size of the neighborhood window for median computation.

Usage: median window_size

Example: median 20

Result: For every point in the active object, INTEGRATE computes a median value and assigns that value to the point. INTEGRATE uses the 20 surrounding points for the calculation.

merge

This command will merge the points from two objects to become a third object. Merge requires 2 parameters: the numbers of the objects to be merged. An optional third parameter specifies whether to use the maximum radius, (max), the minimum radius (min), or the average radius (avg) in areas where the objects overlap.

Usage: merge (object1) (object2) [min,max,avg]

Example: merge 1 2 avg

Result: The radial values from object 1 and object

2 will be averaged and saved in the next available object number.

mload

This command loads a saved displacement matrix. Mload requires one parameter: the name of the saved file.

Usage: mload (matrix file)
Example: mload head_scan.mtx
Result: INTEGRATE loads the head_scan.mtx displacement matrix and transforms the active object into a new coordinate system.

modland

This command positions the landmark pointer on a specified element in the landmark list. When picking points, modland allows the user to return to any landmark in the list and reassign coordinates for that landmark. Modland's parameter indicates which landmark is to be picked or re-picked next.

Usage: modland (L#/Z#)
Example: modland l32
Result: The landmark pointer moves directly to landmark 32 in the landmark list.

move

This command moves the active object along the X, Y, and/or Z axes. Move needs three parameters: the distance in millimeters to move the object along each axis. These distances will be added to the object's current position. The current position appears in the blue box in the lower left corner of the screen.

Usage: move (distance along X) (distance along Y)

(distance along Z)

Example: move 0 100 0

Result: The active object moves 100 mm up the Y axis.

move_vertex

This command allows the movement of individual MOVIE.BYU object vertices. It requires four parameters: the vertex number to move (see pickmode point), and the X, Y, and Z distances to move the vertex. This is useful for hand-editing specific objects when the object is inaccurate and no automated method is available for correcting it.

Usage: move_vertex (vertex number) (X distance) (Y distance) (Z distance)

Example: move_vertex 11234 1 2 3

Result: Vertex 11234 moves 1 mm along the X axis, 2 mm along the Y axis, and 3 mm along the Z axis.

movie_segment

This command extracts a segment of a mesh object by specifying bounding planes in the directions of the X, Y, and Z axes. A complete bounding box can be specified by l(x/y/z)#, which provides the lower boundary on the specified axis and u(x/y/z)#, which provides the upper boundary on the specified axis. All six possible boundaries can be specified, with positive or negative values, as appropriate. Note: The actual screen position of the object (not its native coordinate system) determines which points will be copied.

Usage: movie_segment l(x/y/z)# u(x/y/z)#

Example: movie_segment lx-100 ux100 ly-100

uy100 lz-100 uz100

Result: INTEGRATE copies all the points within a 200 mm cube around the origin of the active object and stores the copied points in a new object.

mwrite

This command writes out the most recent displacement matrix of the active object to a specified file.

Usage: mwrite file

Example: mwrite head_scan.mtx

Result: INTEGRATE writes the active object's displacement matrix to a file called head_scan.mtx. INTEGRATE stores the file in the directory from which INTEGRATE was launched.

nameland

This command assigns a name to an auxiliary (Z) landmark. The form of the command is: nameland Z# new_landmark_name. New_landmark_name must not contain blanks.

Usage: nameland zlandmark# new_name

Example: nameland z2 helmet_landmark1

Result: Auxiliary landmark Z2 is renamed helmet_landmark1.

negsub

This command performs a subtraction on two objects along each radial value. Specify a reference object and a replace object of the same size (that is, with the same trim values), and INTEGRATE subtracts one object from the other. The second object is subtracted from the first object, and the results less than zero (negative radial values) are retained.

INTEGRATE stores the subtraction results in the replace (second) object.

Note: For best results, **register** and **resample** the objects before executing negsub.

Usage: negsub reference_obj replace_obj

Example: negsub 2 1

Result: INTEGRATE subtracts object 1 from object 2 and stores all negative values in object 1.

new_order

This command reads in a file with a new order for landmark picking for the 42 standard landmarks or a subset of those landmarks. One parameter is required: the file name of the file with the new order typed in it. The file should have landmark numbers (separated by a space) in the order that the landmarks are to be picked. This command requires an active object to run.

Usage: neworder filename

Example: neworder special_landmarks

Result: INTEGRATE reads in the landmark order file called special_landmarks.

newland

This command allows manual entry of a landmark. It operates on the active object and requires 4 parameters: the destination landmark number (L# or Z#) and the X (right-left), Y (up-down), and Z (near-far) coordinates in the screen coordinate system. The user can include a name for the landmark, also.

Usage: newland (Z#/L#) X Y Z [new landmark

name]

Example: newland z10 20 20 20 helmet2

Result: INTEGRATE creates a new auxiliary landmark, Z10, at the specified coordinates and names it helmet2.

option

This command controls preset options which determine the initial state of an object after it is loaded, or in some cases after it has been transformed. Option may be followed by any on/off command. For example, OPTION WIREFRAME OFF will cause all future objects to be loaded without immediately displaying the wireframe form. Similar options could be OPTION SURFACE ON to turn on the surface form of an object as soon as it is loaded, or OPTION FILL ON to cause voids to be automatically filled as part of the object load process, and after any significant manipulation of an object.

Usage: option (command) on/off

Example: option land off

Result: When you load an object and its landmark file, the object appears with the landmarks hidden.

ortho

This command toggles between the normal perspective view and an orthographic view of the object space. This is useful for removing parallax from the view to better interpret relationships between points.

Usage: toggle command

Example: ortho

Result: Objects toggle from normal view to

orthographic view or from orthographic view to normal view.

pause

This command supports batch processing by stopping the execution of a batch file until the operator presses a key to continue. Note: a batch file can also be paused while it is running by pressing any key. Do not use Escape (Esc) to pause a batch file. Escape terminates batch processing.

Usage: pause
Example: pause
Result: When INTEGRATE reaches the pause command in the batch file, command execution stops until the operator presses a key. Note: Do not press Escape (Esc) to continue. Escape terminates batch processing.

pick [on/off]

This command enables or disables point picking mode.

Usage: toggle command
Example: pick
Result: Point picking mode is enabled or disabled.

pickmode

This command sets the point picking mode to perform specific operations. It has two parameters: a mode for selecting a specific point near the cursor, and a mode for using the point to automatically perform an operation.

Selection options are: centroid, closest, or median.

Centroid chooses the average longitude and latitude of all

the points which were detected in the pick region (the points indicated by the cursor).

Closest chooses the longitude and latitude of the point in the pick region closest to your eye position.

Median chooses the longitude of the longitudinal median point and latitude of the latitudinal median point in the pick region.

> Usage: pickmode (centroid/closest/median)
> Example: pickmode closest
> Result: INTEGRATE chooses the longitude and latitude of the point in the pick region closest to the viewer's "eye."

Operation options are: con2p, con3p, cir2p, cir3p, mul2p, mul2a, land, auxland, distance, delpnt, and point.

Con2p causes every odd point to be the start point of a contour, and every even point to be the end point of a contour, with the center of the object defining the plane of the contour.

Con3p causes every group of three points to define a contour plane, with the contour running from the first point to the second point.

Cir2p and *cir3p* work the same as con2p and con3p, except that they create a complete circumference. (Note that contours and circumferences do not work properly if they

encounter a boundary of the object.)

Mul2p creates multiple, consecutive two-point contours (the third point is the object center), with the second point of a contour becoming the first point of the next contour.

Mul2a creates multiple, consecutive two-point contours (the third point is the Y-axis at mid-latitude), with the second point of a contour becoming the first point of the next contour.

Land creates a new standard landmark from every selected point.

Auxland creates a new auxiliary landmark from every selected point.

Distance computes the straight-line distance from the first selected point to each point selected thereafter.

Delpnt deletes the selected point.

Point reports the longitude, latitude, radius, and XYZ value for each selected point.

Usage: pickmode
(con2p/con3p/cir2p/cir3p/mul2p/mul2a/land/auxland
/distance/delpnt/point)
Example: pickmode auxland
Result: INTEGRATE creates an auxiliary landmark at the point the user picks by clicking the left mouse

button.

planes [on/off]

This command turns the XY, YZ, and XZ reference planes on or off.

Usage: toggle command
Example: planes
Result: The reference planes appear or disappear.

pload

This command loads a polygon mesh file in stanford .ply format. After loading, the object is identical to a movie.byu-format mesh. All commands that work with movie.byu files can be applied to the loaded object. Pload requires one parameter: the name of the .ply file to be loaded.

Usage: pload (file name)
Example: pload 052.ply
Result: INTEGRATE loads the polygon mesh file called 052.ply.

points [on/off]

This command enables or disables a display of the scan data for the active object as individual points.

Usage: toggle command
Example: points
Result: INTEGRATE turns the active object's point display on or off.

pop

This command redraws the **INTEGRATE** window over any other windows. It is equivalent to the window menu POP option.

Usage: pop

Example: pop

Result: Other open windows on the screen disappear behind the INTEGRATE window.

possub

This command performs a subtraction on two objects along each radial value. Specify a reference object and a replace object of the same size (that is, with the same trim values), and INTEGRATE subtracts one object from the other. The second object is subtracted from the first object, and the results greater than zero (positive radial values) are retained. INTEGRATE stores the subtraction results in the replace (second) object.

Note: For best results, **register** and **resample** the objects before executing possub.

Usage: possub reference_obj replace_obj

Example: possub 2 1

Result: INTEGRATE subtracts object 1 from object 2 and stores all positive values in object 1.

print

This command calls up the snapshot tool which saves a section of the screen for printing. It requires no parameters.

Usage: print

Example: print

Result: The snapshot tool appears. Use the snapshot tool to capture a section of the screen for printing.

push

This command causes all other windows on the screen to appear on top of the **INTEGRATE** window. It is equivalent to the window menu PUSH option.

Usage: push
Example: push
Result: Open windows on the screen appear on top of the INTEGRATE window.

pwrite

This command writes out a polygon mesh object in stanford .ply format. Pwrite requires one parameter: a name for the file to be written.

Usage: pwrite (file name)
Example: pwrite 101.ply
Result: INTEGRATE writes the polygon mesh object to a file called 101.ply and stores the file in the directory from which INTEGRATE was launched.

recolor

This command rescales the color file values for the active object to maximize the available information. Use recolor to make an object lighter or darker when it's displayed in full color. Recolor requires two parameters: the minimum color to distinguish from black and the maximum color to distinguish from white. All colors between the min and the max will be rescaled to evenly fill the color space between black and white. Note that a negative minimum is equivalent to adding a positive offset to all color values.

The minimum color is usually set to zero. If the user specifies a maximum value less than 256, the object appears

lighter; if the user specifies a maximum color greater than 256, the object appears darker.

lighter ⇐ 256 ⇒ darker

Usage: recolor min max

Example: recolor 0 198

Result: The active object becomes lighter when displayed in full color.

refresh

This command supports batch processing by redrawing the screen in the middle of a sequence of batch operations. Normally the screen is not redrawn during a batch sequence.

Usage: refresh

Example: refresh

Result: When INTEGRATE reaches the refresh command in a batch file, INTEGRATE redraws the objects on the screen.

remark

This command inserts a text string in the session audit trail. The text string appears in the INTEGRATE session record, stored in the directory from which INTEGRATE was launched.

Usage: remark (string)

Example: remark starting new session

Result: INTEGRATE inserts "starting new session" in the session record.

resample

This command copies an object to a new object while re-establishing an orientation to the standard cylindrical grid system with respect to the center of the screen axis system.

Resample accepts two optional parameters: the number of the object to resample, and the number of interpolated points to include in the sample. If an object number is not specified, the active object is resampled.

If the number of extra samples is not specified, it is set to 4, which normally gives good results. The available range is from 0 to 16. The number of samples must always be the second parameter. If the active object is being resampled, use a dash, a 0, or the number of the active object for the first parameter.

Resample creates a new object in the first available slot in the object pool.

Usage: resample (obj) (# of points)
Example: resample - 16
Result: INTEGRATE creates a copy of the original object, but transforms it to a new coordinate system defined by the object's orientation to the center of the screen.

rgb

This command toggles between color map mode (limited to 2048 colors) and RGB mode (full 24-bit color). When the fullcolor command is applied to an object, the full color is available for viewing the surface when in RGB mode.

Usage: toggle command
Example: rgb
Result: The active object appears in full color.

right

This command moves the viewer's "eye" to the right side of the object. Right has one optional parameter: a distance. If the distance is positive, the viewer's eye will be that much further away from the object than the default distance. If the distance is negative, the viewer's eye will be that much closer to the object.

Usage: right [distance]
Example: right
Result: The viewer sees the right side of the object.

rotate

This command rotates the active object around the X, Y, and/or Z axes. Rotate needs three parameters: the angle to rotate the active object around each of the three axes. These angles will be added to the current position. The current position is shown in the blue box in the lower left corner of the screen.

Usage: rotate (degrees around X) (degrees around Y) (degrees around Z)
Example: rotate 0 30 20
Result: INTEGRATE rotates the active object 30 degrees around the Y axis (counterclockwise) and 20 degrees around the Z axis (counterclockwise).

ruin

This command randomly creates void patches in an object. The command requires both the object and a copy of the object to operate. After execution, the copy of the object will contain only the data of the newly created voids. It is used for testing various object editing tools.

Usage: ruin (object to ruin) (copy object)

Example: ruin 3 4

Result: INTEGRATE creates voids in object 3 and stores the voided data in object 4.

select

This command selects which object is the active object. Select needs one parameter: the number of the object to be selected. Objects can also be selected by typing in just the object number.

Usage: select (object number)

Example: select 3

Result: Object 3 is now the active object.

set

This command sets a parameter in the ASPEC for an object. Set requires two parameters: a parameter name and a new parameter value. Useful parameter names are RSHIFT, NAME, STUDY, SCAN_TYPE, VERSION, LTOFF, LGOFF, FILLED, AND SMOOTHED. Other names should be used with **EXTREME CAUTION**. The parameter value for NAME, STUDY, or SCAN_TYPE should be a string with no embedded blanks. The value for all other parameter names should be an integer, generally less than 512.

Usage: set (parameter1) (parameter2)

Example: set study "traditional"

Result: The information contained in the header under STUDY_TYPE will be changed to read traditional.

shade

This command restores or updates a pseudo-lighting shaded surface to an object.

Usage: shade [object #]
Example: shade
Result: INTEGRATE updates the shading on the active object.

show

This command displays an object that has been hidden. Show has one optional parameter: the number of the object to show. If an object number is not specified, INTEGRATE shows the active object.

Usage: show [object #]
Example: show 3
Result: INTEGRATE displays object 3.

side

This command moves the viewer's "eye" to the left side of the object. Side has one optional parameter: a distance. If the distance is positive, the viewer's "eye" will be that much further away from the object than the default distance. If the distance is negative, the viewer's "eye" will be that much closer to the object.

Usage: side [distance]
Example: side
Result: The viewer sees the left side of the object.

skip

This command skips over a landmark slot when picking landmarks. Skip has one optional parameter: the number of landmark slots to skip. If a skip number is not specified,

INTEGRATE skips one slot. If a negative number is specified, INTEGRATE skips backward in the landmark list.

Usage: skip [value]
example: skip -1
Result: INTEGRATE skips backward one slot in the landmark list.

sleep

This command supports batch processing by forcing the batch process to stop for a given number of seconds, in order to give the operator time to observe the state of an image before processing continues. Sleep accepts one parameter: the number of seconds to wait before continuing. If the number of seconds is not specified, the batch file pauses for one second.

Usage: sleep [value]
Example: sleep 10
Result: INTEGRATE pauses for 10 seconds when it reaches the sleep command in the batch file.

smooth [on/off]

This command enables/disables automatic smoothing for an object after a command, such as cload or resample, which disturbs the smoothness of the data. To execute smoothing for the active object, use **do smooth**.

Usage: toggle command
Example: smooth
Result: Automatic smoothing is turned on or off.

split

This command computes a new landmark at the mid-point between two other landmarks. It operates on the active object and requires three parameters: the destination landmark number (L# or Z#) and the two defining landmarks (L# or Z#). There are two optional parameters for this command, weight1 and weight2, which cause the new landmark to be positioned proportionally between the two defining landmarks. The weight function might be used for determining a combined Center of Gravity from the CG's of two objects of different weights, such as a human head and a helmet system.

Usage: split ([Z/L]destination landmark #) ([Z/L] first landmark #) ([Z/L] second landmark #) [wt1] [wt2]

Example: split z20 z1 z2 .5 2

Result: INTEGRATE creates a new auxiliary landmark, z20, between landmarks z1 and z2. The weights, .5 for z1 and 2 for z2, tell INTEGRATE to position z20 80% of the way toward z2.

• z1 • z20 • z2

INTEGRATE uses the following equation to determine the new landmark's location:

$$\frac{(\text{weight}_1 \bullet \text{coordinates}_1)+(\text{weight}_2 \bullet \text{coordinates}_2)}{\text{weight}_1 + \text{weight}_2}$$

store [on/off]

This command enables or disables storage of data from various measurement commands to a disk file. When store is enabled, results of VOLUME, SURFACE_AREA,

DISTANCE, and PICKMODE DISTANCE point picks are stored to "measures.txt" with appropriate labels.

Usage: toggle command
Example: store
Result: Data from measurement commands are stored in a file called measures.txt in the directory from which INTEGRATE was launched.

subject [on/off]

This command turns all display modes for the active object on or off. Subject on is the same as **show** and subject off is the same as **hide**. Subject by itself works like any other on/off command; it toggles between on and off.

Usage: toggle command
Example: subject
Result: INTEGRATE hides or shows the active object.

super

This command allows several objects to be grouped into a "super object" so all the objects can be moved or changed together. In keeping with the concept of a "super-object," this is a "super-command" with six command modifiers: MAKE, ADD, RELEASE, DELETE, LINK, and UNLINK. MAKE creates a new super-object, ADD adds objects to a super-object, RELEASE removes one or more objects from a super-object, DELETE deletes the super-object, LINK attaches a sub-object and applies an offset that causes the sub-object to rotate around the same rotation point as the super-object, and UNLINK removes the offset and detaches the sub-object. Note that the DELETE command (by itself) applied to a super-object is identical to the SUPER DELETE

command.

Usage:

SUPER MAKE sub-obj1 ... sub-obj#

SUPER ADD super-obj sub-obj1 ... sub-obj#

SUPER RELEASE super-obj sub-obj1 ... sub-obj#

SUPER DELETE super-obj

SUPER LINK super-obj sub-obj

SUPER UNLINK super-obj sub-obj

Example: supermake 2 3 4

Result: INTEGRATE groups objects 2, 3, and 4 into a super object.

surface [on/off]

This command enables or disables a display of the scan data for the active object as shaded surface polygons.

Usage: toggle command

Example: surface

Result: INTEGRATE turns the surface display on or off for the active object.

surface_area

This command computes the displayed surface area of the active object.

Usage: surface_area

Example: surface_area

Result: INTEGRATE computes the surface area of the active object and displays the result in the lower left corner of the screen.

thin

This command sets the frequency of longitude and latitude lines to be shown on the active object. Thinning an object

speeds up some INTEGRATE functions.

Thin needs two parameters: the longitude thin factor and the latitude thin factor. For example, for a thin factor of 2, INTEGRATE displays every second data point; for a thin factor of 3, INTEGRATE displays every third data point, and so on.

Usage: thin value value

Example: thin 2 2

Result: INTEGRATE displays only every second data point along each longitude and latitude.

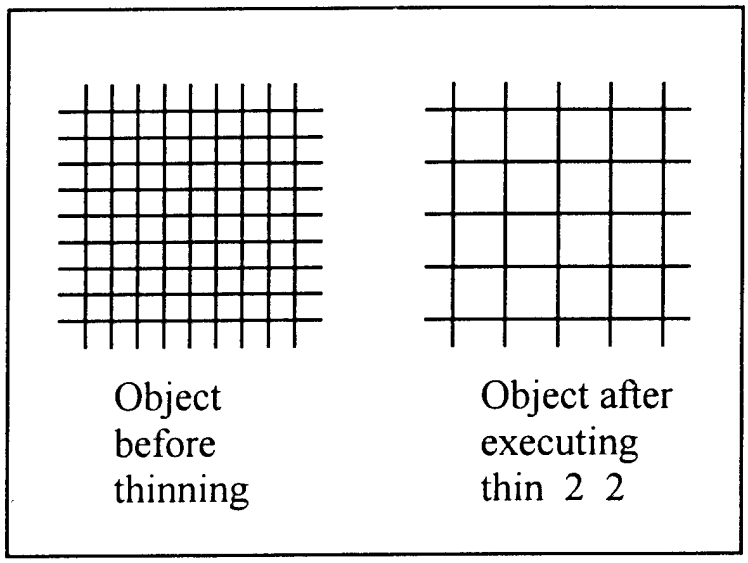


Figure 14: Thinning an object.

threshold

This command performs a threshold operation on an object. The object and threshold values are specified by the user. The qualifiers eq (equal), ne (not equal), lt (less than), le (less than or equal), gt (greater than), ge (greater than or equal) refer to the values to be zeroed. For example, "threshold 2 lt

55" means all points in object 2 below the 55 threshold should be set to zero.

Usage: threshold (object #) (eq/ne/lt/le/gt/ge) (value)
Example: threshold 4 lt 2
Result: INTEGRATE eliminates all radial values less than 2 mm on object 4.

top

This command moves the viewer's "eye" to the top of the object.

Top has one optional parameter: a distance. If the distance is positive, the viewer's eye will be positioned that much further from the object. If the distance is negative, the viewer's eye will be positioned that much closer to the object.

Usage: top [distance]
Example: top 200
Result: The viewer now sees the top of the active object, 200 mm further away from the object than before.

tops

This command converts snap.rgb files from snapshot format to postscript format. It requires one parameter: the name of the file where the postscript commands will be stored.

Usage: tops postscript_file
Example: tops figure.ps
Result: The snapshot file is converted to postscript format and stored in figure.ps in the directory from which INTEGRATE was launched. The figure.ps file

can be printed on any postscript printer.

toupee

This command fills in the top of the head of the active object. Note that this command works best when the object is positioned so that the (estimated!) highest point on the head is centered on the Y axis. This command needs two parameters: the lowest latitude for the toupee, and the highest latitude for the toupee. Check the object coordinates in the blue box on the lower left for the coordinates to use for the toupee. Note that the low latitude must be within the current trim area.

Usage: toupee (bottom of toupee latitude) (top of toupee latitude)
Example: toupee 196 203
Result: INTEGRATE places a cap or "toupee" on the void on top of the active object.

transparent [on/off]

This command makes the surface display for an object partially transparent, allowing visualization of the detail of the grid or of inner objects. Because all objects use the same transparency mask, a transparent object will not be visible inside another transparent object.

Usage: toggle command
Example: transparent
Result: The active object becomes transparent.

trim

This command modifies the starting and ending longitude and latitude so that only the necessary part of the active object will be displayed. Trim needs four parameters which

will change the starting longitude, ending longitude, starting latitude, and ending latitude. These parameters will be added to the current values. To reduce the ending longitude and latitude, use a negative number. The current starting and ending longitude and latitude are displayed in the blue box in the lower left corner of the screen.

Usage: trim left_long right_long lower_lat upper_lat

Example: trim 30 -100 50 -50

Result: INTEGRATE trims the active object.

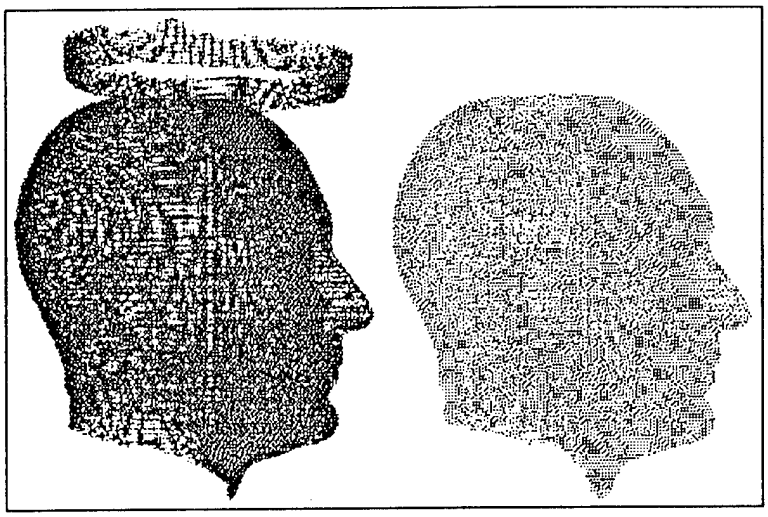


Figure 15: Trimming noise from the top of an object: before trimming (the object on the left) and after trimming (the object on the right).

volume

This command computes the volume between the surface of the active object and the center of the object. This command also generates the coordinates of the center of volume.

Usage: volume

Example: volume

Result: INTEGRATE computes the volume and center

of volume of the active object and displays the result in the lower left corner of the screen.

walls

This command sets the clipping planes. The clipping planes control the size of the viewing area.

Walls needs two parameters: near or far.

near - points closer to the eye than the near value will not be displayed (initial value 100).

far - points farther from the eye than far will not be displayed (initial value 1400).

Walls also accepts two other parameters: full and half. "Walls full" automatically sets the near wall to 100 and the far wall to twice the distance of the viewpoint from the origin of the grid. "Walls half" automatically sets the near wall to 100 and the far wall to the distance of the viewpoint from the origin of the grid, eliminating from view the back half of an object centered on the origin. If the viewpoint is moved using eye, top, front, or side, the walls may need to be adjusted to prevent clipping of the object.

Usage: walls near/far

Example: walls 698 702

Result: INTEGRATE limits the viewing area to the space between 698 and 702.

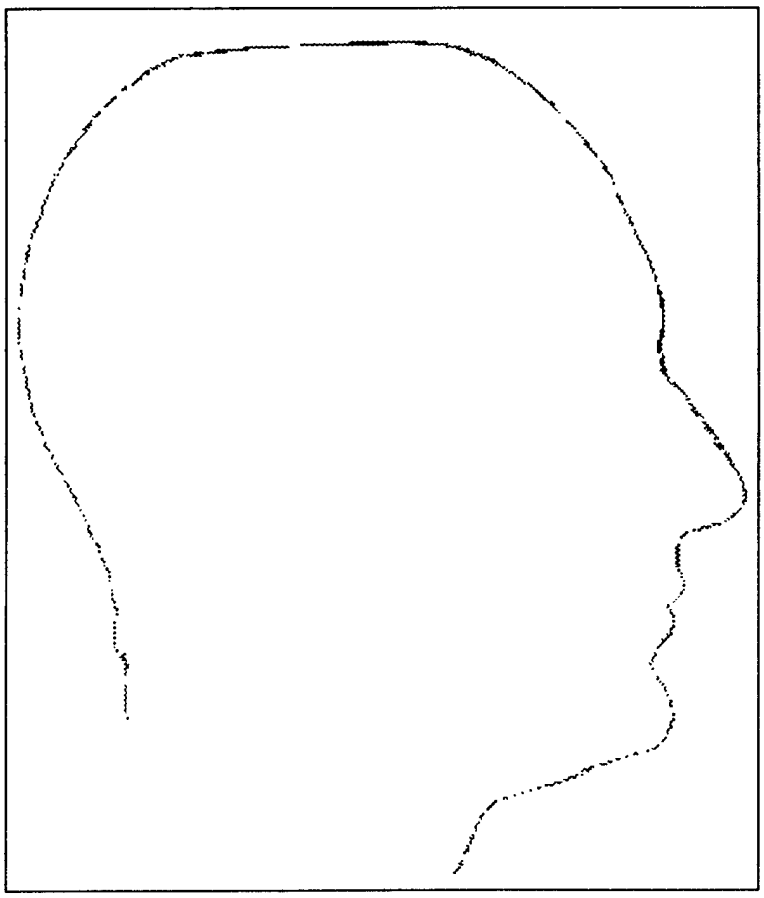


Figure 16: With walls set to 698 702, only a cross section of the object appears.

white

This command sets the screen background color to white. Landmark and object points will change colors so that they will show up against the white background. Typically white is used to prepare a screen for printing to reduce the total number of pixels which must be transferred to the printer.

Usage: white

Example: white

Result: The screen background turns white.

wireframe [on/off]

This command enables or disables wireframe display of the

active object.

Usage: toggle command
Example: wireframe
Result: Wireframe display of the active object turns on or off.

wload

This command reads in a polygon mesh file in wavefront .obj format. After loading, the object is identical to a movie.byu-format mesh. All commands that work with movie.byu files can be applied to the loaded object. Wload requires one parameter: the name of the .obj file to be loaded.

Usage: toggle command
Example: wireframe
Result: Wireframe display of the active object turns on or off.

wwrite

This command writes out a polygon mesh file in wavefront .obj format. Wwrite requires one parameter: the name of the file to be written.

Usage: wwrite (file name)
Example: wwrite 025.obj
Result: INTEGRATE writes the mesh object to a file called 025.obj and stores the file in the directory from which INTEGRATE was launched.

xload

This command reads in data stored in XYZ coordinate format (see xwrite).

Usage: xload file

Example: xload head_scan.xyz

Result: INTEGRATE reads in the file head_scan.xyz.

xwrite

This command writes scan data XYZ coordinates to an ASCII (text) file. If the -g (grid) option is specified, longitude, latitude, and radius are also written to the file. If the -a (all) option is specified, void points are written to the file; otherwise, only non-void (radius > 0.0) points are written. If the -w (waterline) option is specified, the data points are written in latitude-major order (all points at the same latitude grouped together) instead of longitude-major order. If no options are specified, the filename to be written is the first parameter. If options are specified, the filename to be written is the second parameter.

Usage: xwrite [-agw] file

Example: xwrite -a ascii52.2.xyz1

Result: INTEGRATE writes data, including void points, to a file called ascii52.2.xyz1, stored in the directory from which INTEGRATE was launched.

zload

This command reads in previously stored contour data (see zwrite). Zload requires one parameter: the name of the contour file to be read in. Note: if a contour file exceeds the maximum allowed size of a single contour line, the contour will be broken into separate contour lines as required.

Usage: zload file

Example: zload contours

Result: INTEGRATE reads in the contour data file called contours.

zregister

This command registers an object to another object by least-squares fitting of corresponding auxiliary landmarks. Zregister needs two parameters: the number of the object to be registered and the reference object number.

Usage: zregister obj refobj

Example: zregister 2 1

Result: INTEGRATE uses the auxiliary landmarks to align object 2 on object 1.

zwrite

This command writes out the points which make up a contour or circumference line. The first parameter to zwrite is the name of the file to be written. All following parameters are contour numbers to be written.

Usage: zwrite file contour1... contour# [rotate]

Example: zwrite contour_file 3 4 5 r

Result: INTEGRATE writes contours 3, 4, and 5 to a file called contour_file in the directory from which INTEGRATE was launched.

5.0 INTEGRATE'S Audit Trail Function

INTEGRATE's Audit Trail maintains a record of all user commands entered during an INTEGRATE session. The Audit Trail file allows the user to:

- analyze an INTEGRATE session to discover the cause of unsatisfactory results,
- record the history of a modified dataset so future users can evaluate the validity of the final data,
- create a batch file that will automatically reproduce the results of the session.

INTEGRATE stores the commands in a file called AUDITFILE.xxxx, where xxxx is the first four characters in the name of the INTEGRATE system host. For example, a system whose host name is falcon will produce audit trail files called AUDITFILE.falc. The audit trail files are stored in the directory from which INTEGRATE was launched.

INTEGRATE stores all commands executed during the INTEGRATE session except actions initiated during point picking. For example, if the user deleted points from a dataset with **pickmode delall**, the deletion of every point in the dataset would not appear in the audit file. If all the pickmode actions were included, it would be difficult to find more useful information in the file.

IAUDIT is a program that allows the user to view and manipulate INTEGRATE's audit trail files. Instructions for using IAUDIT appear below.

5.1 Using IAUDIT

Follow these steps to view and manipulate audit trail files:

1. Change directory (cd) to the directory from which INTEGRATE was launched for the relevant session.
2. Type **iaudit** and press Enter.
3. When the list of audit trail files appears, determine which audit file contains the relevant session record. The sessions are numbered and are listed by date, time, and user name. A few lines of the session list might look like this:

```
42: *** Integrate Session 3/15/1996  8:55:32 chris
43: *** Integrate Session 3/18/1996 10:15:12 mark
44: *** Integrate Session 3/19/1996 14:39:10 josephine
```

4. Type an IAUDIT command that contains the option for carrying out the required action. The table below defines the IAUDIT options:

iaudit (with no options)	Lists the INTEGRATE sessions by date and time.
-a	Lists every command that was executed in the session or sessions.
-b	Creates a batch file that includes all "first level" commands. Commands from subsidiary batch files are not included, though the @(batch file name) commands that launch batch files are included.
-d	Deletes sessions from the session list.
-t	Creates a batch file that does not include any batch file commands, not even the @(batch file name) commands that launch batch files.

5. To **list all the commands** in sessions 5 through 10, type a command that looks like this:

```
iaudit -a 5 10
```

6. To **create a batch file** from the commands in sessions 24, type a command that looks like this:

```
iaudit -b 24>newbatch
```

where "newbatch" is the name of the batch file to be created.

7. To **delete sessions** 10 through 20 from the session list, type a command that looks like this:

```
iaudit -d 10 20
```

8. To **create a batch file** from the commands in session 12, **excluding any secondary commands**, type a command that looks like this:

```
iaudit -t 12>newbatch
```

where "newbatch is the name of the batch file to be created.

9. Use a text editor to edit the batch files. Some commands may need to be deleted, and some commands may need to be combined into one command.

6.0 References

1. McConville, J.T., Clauser, C.E., Churchill, T. D. , Cuzzi, J. and Kaleps, I. (1980). Anthropometric Relationships of Body and Body Segment Moments of Inertia, Air Force Aerospace Medical Research Laboratory, Air Force Systems Command, Wright-Patterson Air Force Base, OH.

2. Nurre, J.H., Whitestone, J.J., Hoffmeister, J.W., and Burnsides, D.B. (1995), Removing Impulse Noise From Human Head Scan Data, AL/CF-TR-1995-0054, Armstrong Laboratory, Wright-Patterson Air Force Base, Oh.

3. Robinette, K.M., and Whitestone, J.J. (1994) The Need for Improved Anthropometric Methods for the Development of Helmet Systems. Aviation, Space, and Environmental Medicine, pp. A95-A99, May 1994, Aerospace Medical Association, Alexandria, VA.

4. Robinette, K.M. and Whitestone, J.J. (1992). Methods for Characterizing the Human Head for the Design of Helmets, AL-TR-1992-0061, Crew Systems Directorate, Human Engineering Division, Armstrong Laboratory, Wright-Patterson Air Force Base OH.

5. Whitestone J.J., Slemker T.C., Ause-Ellias K.L., Richard R.L., Miller, S. (1995), Fabrication of Total Contact Burn Masks -- Employing Human Body Topography and Computer-Aided Design and Manufacturing (CAD/CAM), Journal of Burn Care and Rehabilitation, Burn Science Publishers, Inc.

6. Whitestone, J.J. (1993), Design and Evaluation of Helmet Systems Using 3D Data. Proceedings of the Human Factors and Ergonomics Society 37th Annual Meeting - 1993, Vol. 1, pg. 63, The Human Factors and Ergonomics Society, Santa Monica CA.

7. Whitestone, J.J. and Robinette, K.M. (1992). High Resolution Human Body Surface Data for the Design of Protection Equipment. Proceedings of the 2nd Pan Pacific Conference on Occupational Ergonomics, Safety and Environmental Protection Research Institute, MMI, Wuhan, China.

APPENDIX A

TUTORIALS: IMAGE DATA AND SCRIPT FILES

FILES NEEDED FOR TUTORIAL_1

SCRIPT FILE	IMAGE FILE(S)	COLOR FILE(S)	LANDMARK FILE(S)
tutorial_1	010_53p	010_53p.rgb	010_53p.lnd

SCRIPT FILE FOR TUTORIAL_1

```
cload 010_53p
lload  010_53p.lnd
rotate 0 75
right
back
left
top
bottom
front
move 50
move 0 50
right
move 0 0 50
move -50 -50 -50
front
trim 0 0 0 -55
trim 0 0 55
copy 1 2
ruin 1 2
hide 2
1
do fill
toupee 200 205
axes
boxes
boxes
alt_land
landlist
landlist
help
help
fkeys
land
land
wireframe
surface
fullcolor 010_53p
rgb
volume
```

surface_area
white
print

FILES NEEDED FOR TUTORIAL_2

SCRIPT FILE	IMAGE FILE(S)	LANDMARK FILE(S)
tutorial_2	010_53p	010_53p.lnd
	010_53ph	010_53ph.lnd
	53psize5	53psize5.lnd

SCRIPT FILE FOR TUTORIAL_2

```
cload 010_53p
rotate 0 75
right
trim 0 0 0 -55
trim 0 0 55
toupee 200 205
do fill
lload 010_53p.lnd
cload 010_53ph
trim 0 0 0 -50
trim 0 0  45
lload 010_53ph.lnd
lregister 2 1
right
walls 695 699
walls full
front
cload 53psize5
lload 53psize5.lnd
zregister 3 2
right
walls 695 699
hide 2
walls full
```

FILES NEEDED FOR TUTORIAL_3

SCRIPT FILE	IMAGE FILE(S)	COLOR FILE(S)	LANDMARK FILE(S)
tutorial_3	010_53p	010_53p.rgb	010_53p.lnd

SCRIPT FILE FOR TUTORIAL_3

```
cload 010_53p
trim 0 0 0 -50
trim 0 0 58
do fill
toupee 205 207
wireframe
surface
fullcolor 010_53p
rgb
pick on
pickmode land
right
rotate 0 75
***begin point picking
```

FILES NEEDED FOR TUTORIAL_4

SCRIPT FILE	IMAGE FILE(S)	LANDMARK FILE(S)
tutorial_4	53psize5	53psize5.lnd
	100_53ph	100_53ph.lnd
	100_53p	100_53p.lnd
	101_53ph	101_53ph.lnd
	101_53p	101_53p.lnd
	102_53ph	102_53ph.lnd
	102_53p	102_53p.lnd
	104_53ph	104_53ph.lnd
	104_53p	104_53p.lnd
	105_53ph	105_53ph.lnd
	105_53p	105_53p.lnd

SCRIPT FILE FOR TUTORIAL_4

```
cload 53psize5
lload 53psize5.lnd
align xz z1 z5 z3 z3
cload 100_53ph
lload 100_53ph.lnd
zregister 2 1
cload 100_53p
lload 100_53p.lnd
lregister 3 2
1
copyland l1 3 l34
copyland l2 3 l38
cload 101_53ph
lload 101_53ph.lnd
zregister 4 1
cload 101_53p
lload 101_53p.lnd
lregister 5 4
1
copyland l3 5 l34
copyland l4 5 l38
cload 102_53ph
lload 102_53ph.lnd
zregister 6 1
cload 102_53p
lload 102_53p.lnd
zregister 7 6
1
copyland l5 7 l34
copyland l6 7 l38
```

```
cload 104_53ph
lload 104_53ph.lnd
zregister 8 1
cload 104_53p
lload 104_53p.lnd
lregister 9 8
1
copyland l7 9 l34
copyland l8 9 l38
cload 105_53ph
lload 105_53ph.lnd
zregister 10 1
cload 105_53p
lload 105_53p.lnd
lregister 11 10
1
copyland l9 11 l34
copyland l10 11 l38
hide 2
hide 3
hide 4
hide 5
hide 6
hide 7
hide 8
hide 9
hide 10
hide 11
right
back
left
front
```

FILES NEEDED FOR TUTORIAL_5

SCRIPT FILE	IMAGE FILE(S)	COLOR FILE(S)	LANDMARK FILE(S)
tutorial_5	face	face.rgb	
	mask	mask.rgb	

SCRIPT FILE FOR TUTORIAL_5

```
cload face
move 0 200
cload mask
move 0 200
possub 1 2
threshold 2 ge 1
and 1 2
cload mask
move 0 200
negsub 1 3
threshold 2 ge 1
and 1 3
cload mask
move 0 200
possub 1 4
threshold 4 lt 1
and 1 4
cload mask
move 0 200
negsub 1 5
threshold 5 lt 1
and 1 5
```

FILES NEEDED FOR TUTORIAL_6

SCRIPT FILE	IMAGE FILE(S)	COLOR FILE(S)	MATRIX FILE(S)
tutorial_6	dr_boff.g		std.mtx
fbsetup			

SCRIPT FILE FOR TUTORIAL_6

```
axes off
boxes off
fkeys off
@fbsetup
gload dr_boff.g
mload std.mtx
eye 3700
pick on
move 100 70
points off
surface on
surface off
points on
movie_seg uz-234
1
movie_seg lz260
1
movie_seg ly610
1
movie_seg ly120 uy609 lz-233 uz259
1
movie_seg uy119 lz-233 uz259
hide 1
2
move 0 0 -50
3
move 0 0 50
4
move 0 50
6
move 0 -50
```

FILES NEEDED FOR TUTORIAL_7

SCRIPT FILE (S)	IMAGE FILE(S)	COLOR FILE(S)	MATRIX FILE(S)
tutorial_7	m2.1c, m2.2c, m2.3c,		
cs	m2.4c, m2.5c, m2.6c,		
link1	m2.7c, m2.8c, m2.9c,		
sit1	m2.10c, m2.11c, m2.12c,		
	m2.13c, m2.14c, m2.15c,		
	m2.16c, m2.17c, m2.18c,		
	m2.19c,		
	m2.1cs, m2.2cs, m2.3cs,		
	m2.4cs, m2.5cs, m2.6cs,		
	m2.7cs, m2.8cs, m2.9cs,		
	m2.10cs, m2.11cs, m2.12cs,		
	m2.13cs, m2.14cs, m2.15cs,		
	m2.16cs, m2.17cs, m2.18cs,		
	m2.19cs		

SCRIPT FILE FOR TUTORIAL_7

```
option wireframe off
option surface on
cloads -a m2.1c
cloads -a m2.2c
cloads -a m2.3c
cloads -a m2.4c
cloads -a m2.5c
cloads -a m2.6c
cloads -a m2.7c
cloads -a m2.8c
cloads -a m2.9c
cloads -a m2.10c
cloads -a m2.11c
cloads -a m2.12c
cloads -a m2.13c
cloads -a m2.14c
cloads -a m2.15c
cloads -a m2.16c
cloads -a m2.17c
cloads -a m2.18c
cloads -a m2.19c
* link the parts of a stereo subject together
* parts must be linked (at least right now) from the outside in
*
* head and neck
super link 2 1
super link 3 2
```

```
* right arm
super link 7 8
super link 6 7
super link 3 6
* left arm
super link 10 11
super link 9 10
super link 3 9
* right leg
super link 14 15
super link 13 14
super link 12 13
super link 5 12
* left leg
super link 18 19
super link 17 18
super link 16 17
super link 5 16
* lower torso
super link 4 5
super link 3 4
* move the segments of the body into a seated position
* segment 3 is torso to which all other parts are anchored
3
move 0 -80
* slightly bend torso at waist
5
rotate -10
* rotate legs
12
rotate -80
14
rotate 90
16
rotate -80
18
rotate 90
* rotate arms
6
rotate 0 30
7
rotate -90
8
rotate 0 -110
9
rotate 0 -30
10
rotate -90
```

11
rotate 0 70
right
3
rotate 0 30
rotate 0 30
rotate 0 30
rotate 0 30
rotate 0 30
rotate 0 30
rotate 0 30
rotate 0 30
rotate 0 30
rotate 0 30
rotate 0 30
rotate 0 30

APPENDIX B

ANATOMICAL LANDMARKS: DESCRIPTIONS AND ILLUSTRATIONS

ANATOMICAL LANDMARK DEFINITIONS

CHEILION: the corners of the mouth formed by the juncture of the lips.

ECTOCANTHUS: the outer corners of the eyes; the lateral canthus

ENDOCANTHUS: the inner corners of the eyes; the medial canthus

FRONTOTEMPORALE: The point of deepest indentation of the temporal crest from the frontal bone above the browridges.

GLABELLA: Landmark title for the most forward point in the midline of the forehead between the brow ridges.

GONION: A corner of the jaw; the lateral point of the corner of the mandible (jaw bone).

INFRAMALAR: The most inferior point of the zygomatic process of the maxilla.

INFRAORBITALE: The lowest point on the inferior margin of the orbit or eye socket.

INFRAZYGION: The inferior border of the zygomatic arch directly below zygion.

MENTON (LANDMARK): Title for the inferior point of the mandible (tip of the chin) in the midsagittal plane.

NUCHALE: The lowest bony point on the base of the back of the skull in the mid-sagittal plane.

PROMENTON: The most anterior projection of the soft tissue of the chin in the midsagittal plane.

PRONASALE (LANDMARK): Title for the tip of the nose.

PUPIL: The center of the contractile (usually round) aperture in the iris of the eye; the center of the pupil.

SELLION: The point of greatest indentation of the nasal root depression. (the point of greatest indentation where the bridge of the nose meets the forehead.)

STOMION: The point of contact between the upper and lower lips in the midsagittal plane.

SUBMANDIBULAR: Under the mandible or lower jaw.

SUBNASALE: The point inferior to the nose where the base of the nasal septum meets the philtrum; the point of the intersection of the groove of the upper lip (philtrum) with the inferior surface of the nose in the midsagittal plane.

SUPRAECTOCANTHUS: The most protruding point of the browridge located on the same vertical axis as ectocanthus.

SUPRAENDOCANTHUS: The most protruding point of the browridge located on the same vertical axis as ectocanthus.

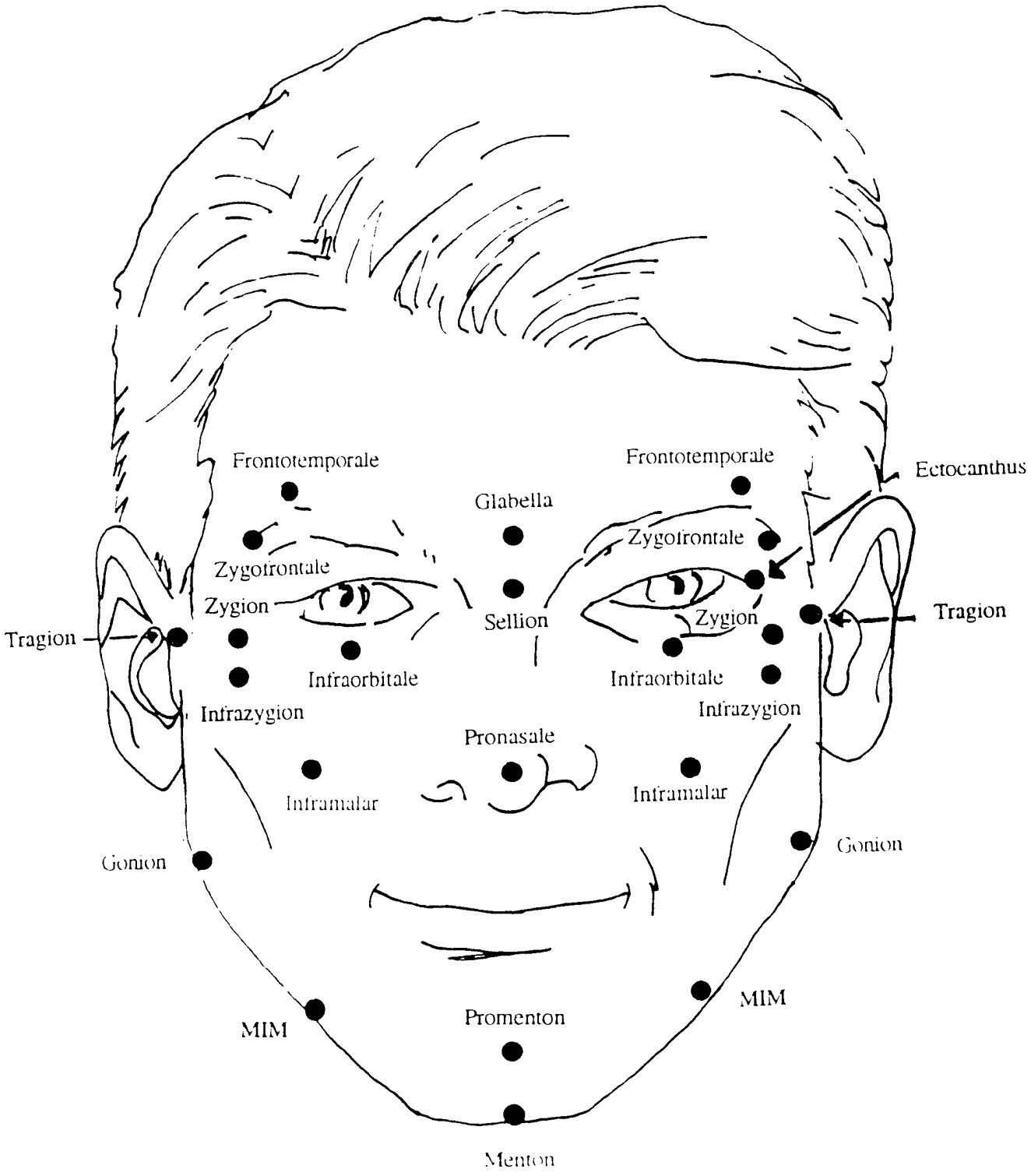
SUPRAMENTON: The point of greatest indentation of the mandibular symphysis.

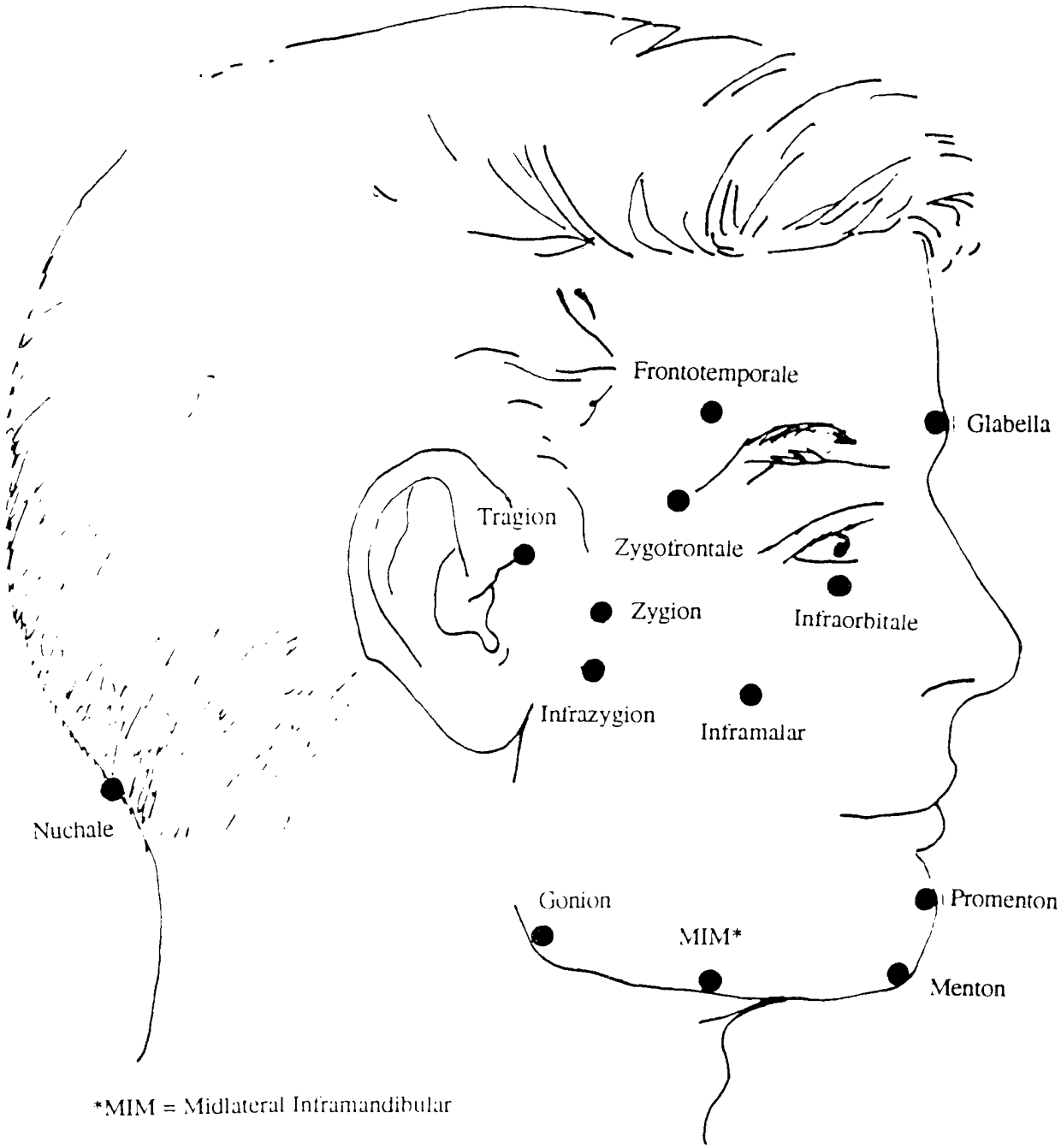
SUPRAPUPIL: The most protruding point of the browridge located on the same vertical axis as the corresponding right or left pupil.

TRAGION: point located at the notch just above the tragus of each ear. this point corresponds approximately to the upper edge of the ear hole (external auditory meatus) of the skull.

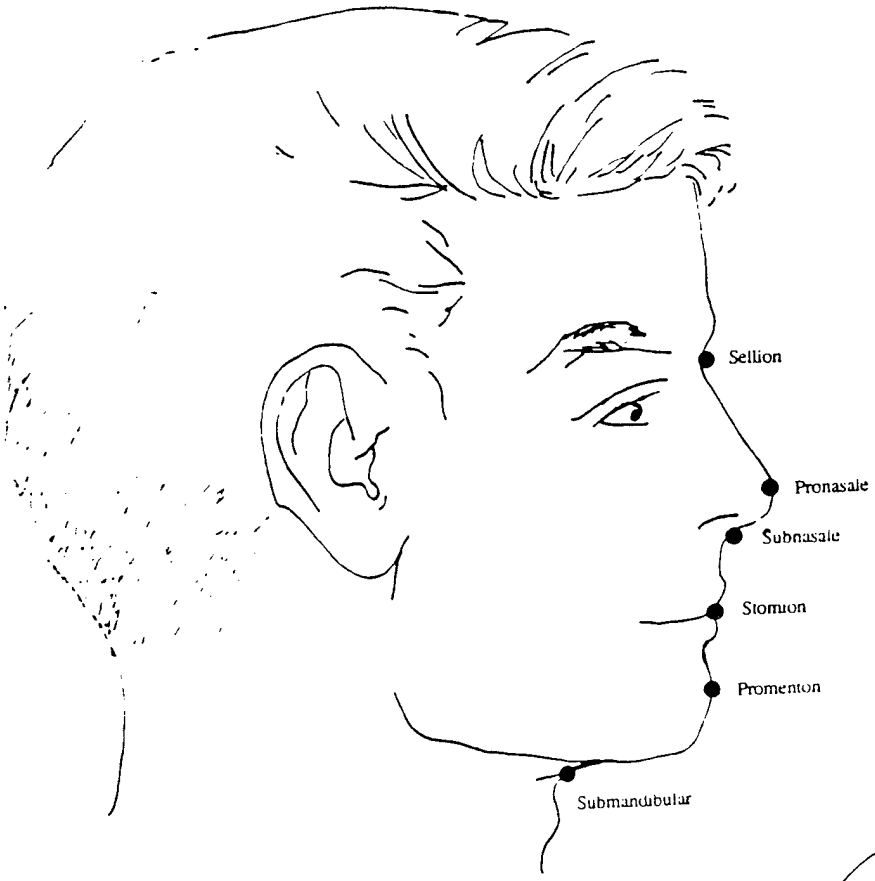
ZYGION: the lateral point of the zygomatic arch.

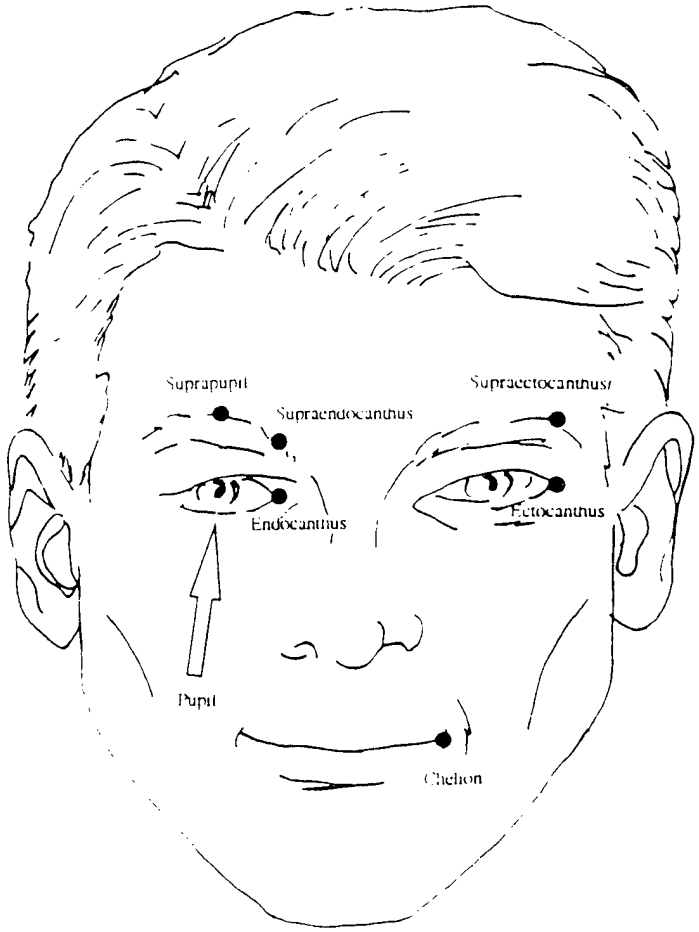
ZYGOFRONTALE: the most lateral point of the frontal bone where it forms the upper margin of the bony eye socket.


Frontotemporale
Frontotemporale
Ectocanthus
Glabella
Zygofrontale
Zygofrontale
Zygion
Sellion
Zygion
Tragion
Tragion
Infraorbitale
Infraorbitale
Infrazygion
Infrazygion
Pronasale
Inframalar
Inframalar
Gonion
Gonion
MIM
MIM
Promenton
Menton



*MIM = Midlateral Inframandibular


Sellion
Pronasale
Subnasale
Stomion
Promenton
Submandibular


Suprapupil
Supraendocanthus
Supraectocanthus
Endocanthus
Ectocanthus
Pupil
Cheilion

APPENDIX C

LANDMARK FILES: ANATOMICAL AND AUXILIARY LANDMARKS

SUBJECT_ID = 010_53p
SCAN_TYPE = NO TYPE
STUDY_NAME = * * * NO STUDY * * *
LAND_STUDY = New Study
STD_LAND = 42
AUX_LAND = 0
STANDARD =

1	371	116	72.66	-11.54	181.31	-71.74	Right Tragion
2	343	124	74.03	-35.69	193.81	-64.86	Right Zygion
3	344	118	72.86	-34.34	184.43	-64.26	Right Infra Zygion
4	346	78	55.40	-24.91	121.91	-49.49	Right Gonion
5	283	67	68.70	-64.97	104.72	-22.35	Right Mid Infra Mandibular
6	299	153	83.38	-72.03	239.14	-41.99	Right Frontotemporale
7	297	146	86.20	-75.52	228.20	-41.57	Right Zygofrontale
8	280	110	90.75	-86.84	171.93	-26.35	Right Infra Malar
9	288	137	80.96	-74.80	214.13	-30.99	Right Ectocanthus
10	259	138	75.82	-75.77	215.69	-2.79	Right Endocanthus
11	281	129	84.46	-80.52	201.63	-25.51	Right Infraorbitale
12	237	161	89.27	-86.86	251.64	20.62	Glabella
13	237	143	93.29	-90.76	223.51	21.55	Sellion
14	233	122	117.98	-113.31	190.69	32.85	Pronasale
15	234	111	102.96	-99.23	173.49	27.46	Subnasale
16	234	75	106.66	-102.80	117.22	28.44	Promenton
17	231	65	92.67	-88.35	101.60	27.98	Menton
18	257	93	96.21	-96.20	145.36	-1.18	Right Chelion
19	234	96	106.02	-102.18	150.05	28.27	Stomion
20	216	94	99.29	-87.57	146.92	46.80	Left Chelion
21	234	53	58.81	-56.68	82.84	15.68	Submandibular
22	196	130	84.98	-62.97	203.19	57.07	Left Infraorbitale
23	215	138	75.36	-66.02	215.69	36.34	Left Endocanthus
24	186	137	80.62	-52.66	214.13	61.04	Left Ectocanthus
25	182	70	68.10	-41.90	109.41	53.69	Left Mid Infra Mandibular
26	176	154	84.04	-46.69	240.70	69.88	Left Frontotemporale
27	175	148	85.42	-46.58	231.32	71.60	Left Zygofrontale
28	189	110	91.67	-62.39	171.93	67.16	Left Infra Malar
29	132	123	77.00	-3.78	192.25	76.91	Left Zygion
30	133	117	76.22	-4.68	182.87	76.08	Left Infra Zygion
31	128	81	61.23	0.00	126.60	61.23	Left Gonion
32	108	118	74.80	18.17	184.43	72.56	Left Tragion
33	495	93	85.38	83.53	145.36	-17.68	Nuchale
34	271	139	80.98	-79.62	217.26	-14.83	Right Pupil
35	286	148	88.59	-82.65	231.32	-31.89	Right SupraEctocanthus
36	269	153	86.10	-85.01	239.14	-13.68	Right SupraPupil
37	256	150	87.00	-87.00	234.45	0.00	Right SupraEndocanthus
38	203	140	84.67	-67.39	218.82	51.27	Left Pupil
39	192	150	88.79	-62.79	234.45	62.78	Left SupraEctocanthus
40	205	153	89.22	-72.31	239.14	52.26	Left SupraPupil
41	219	150	87.33	-78.48	234.45	38.30	Left SupraEndocanthus
42	234	87	104.15	-100.38	135.98	27.78	SupraMenton

AUX =
END =

```
SUBJECT_ID = 010_53ph
SCAN_TYPE  = NO TYPE
STUDY_NAME = * * *  NO STUDY  * * *
LAND_STUDY = New Study
STD_LAND = 42
AUX_LAND = 6
STANDARD =
 1 -999 -999    0.00    0.00    0.00    0.00
 2 -999 -999    0.00    0.00    0.00    0.00
 3 -999 -999    0.00    0.00    0.00    0.00
 4 -999 -999    0.00    0.00    0.00    0.00
 5 -999 -999    0.00    0.00    0.00    0.00
 6 279 130  109.69 -105.35  203.19  -30.55    Right Frontotemporale
 7 280 123  109.70 -104.98  192.25  -31.85    Right Zygofrontale
 8 271  86  109.52 -107.67  134.42  -20.05    Right Infra Malar
 9 -999 -999    0.00    0.00    0.00    0.00
 0 -999 -999    0.00    0.00    0.00    0.00
11 268 104  107.21 -106.05  162.55  -15.74    Right Infraorbitale
12 232 130  123.92 -118.59  203.19   35.97    Glabella
13 232 114  121.49 -116.26  178.18   35.26    Sellion
14 231  90  136.23 -129.87  140.67   41.14    Pronasale
15 -999 -999    0.00    0.00    0.00    0.00
16 231  49  111.87 -106.65   76.59   33.78    Promenton
17 -999 -999    0.00    0.00    0.00    0.00
18 -999 -999    0.00    0.00    0.00    0.00
19 -999 -999    0.00    0.00    0.00    0.00
20 -999 -999    0.00    0.00    0.00    0.00
21 -999 -999    0.00    0.00    0.00    0.00
22 201 105  106.89  -83.45  164.12   66.79    Left Infraorbitale
23 -999 -999    0.00    0.00    0.00    0.00
24 -999 -999    0.00    0.00    0.00    0.00
25 -999 -999    0.00    0.00    0.00    0.00
26 185 128  111.11  -71.54  200.06   85.02    Left Frontotemporale
27 185 121  111.76  -71.96  189.12   85.51    Left Zygofrontale
28 192  88  109.96  -77.76  137.54   77.75    Left Infra Malar
29 -999 -999    0.00    0.00    0.00    0.00
30 -999 -999    0.00    0.00    0.00    0.00
31 -999 -999    0.00    0.00    0.00    0.00
32 -999 -999    0.00    0.00    0.00    0.00
33 -999 -999    0.00    0.00    0.00    0.00
34 -999 -999    0.00    0.00    0.00    0.00
35 -999 -999    0.00    0.00    0.00    0.00
36 -999 -999    0.00    0.00    0.00    0.00
37 -999 -999    0.00    0.00    0.00    0.00
38 -999 -999    0.00    0.00    0.00    0.00
39 -999 -999    0.00    0.00    0.00    0.00
40 -999 -999    0.00    0.00    0.00    0.00
41 -999 -999    0.00    0.00    0.00    0.00
42 -999 -999    0.00    0.00    0.00    0.00
AUX =
 1 387 122  121.91    4.49  190.69 -121.83 Helmet LAND 1
 2 354  94  115.02  -41.39  146.92 -107.32 Helmet LAND 2
 3 233 171  123.48 -118.60  267.27   34.39 Helmet LAND 3
 4 115  92  114.98   18.26  143.80  113.52 Helmet LAND 4
 5  83 118  122.87   64.46  184.43  104.61 Helmet LAND 5
 6 493  80  107.78  104.87  125.04  -24.90 Helmet LAND 6
END =
```

APPENDIX D

COMMANDS: FUNCTIONALITY FOR HEAD AND WHOLE BODY IMAGE DATA

COMMAND	Works With Cyberware?	Works With .g/.ply/.obj?
!	Yes	Yes
$	Yes	Yes
@	Yes	Yes
abssub	Yes	No
add	Yes	No
add_to_land	Yes	Yes
addobj	Yes	No
align	Yes	No
alt_land	Yes	Yes
and	Yes	No
avgland	Yes	No
axes	Yes	Yes
back	Yes	Yes
balltest	Yes	No
black	Yes	Yes
bottom	Yes	Yes
boxes	Yes	Yes
cd	Yes	Yes
center	Yes	Yes
centroid	Yes	Yes
circumference	Yes	No
cload	Yes	No
colors	Yes	Yes
comment	Yes	Yes
conclose	Yes	No
contour	Yes	No
contours	Yes	Yes
copy	Yes	No
copyland	Yes	Yes
copyseg	Yes	No
cursor	Yes	No
cwrite	Yes	No
cybermovie	Yes	No
delete	Yes	Yes
delland	Yes	Yes
delpnt	Yes	No
delseg	Yes	No
delta	Yes	No
derive	Yes	Yes
diff	Yes	No
dilate	Yes	No
displace	No	Yes
distance	Yes	No
do	Yes	No
drawline	Yes	No
erode	Yes	No
exit	Yes	Yes
eye	Yes	Yes
eyedist	Yes	Yes
fcwrite	Yes	Yes
fill	Yes	No
filter	Yes	No
filtseg	Yes	No
fix_seam	Yes	No
fkeys	Yes	Yes
front	Yes	Yes
fullcolor	Yes	Yes

gcv	Yes	No
gload	No	Yes
gouraud	Yes	Yes
gwrite	No	Yes
help	Yes	Yes
hide	Yes	Yes
histogram	Yes	No
interpolate	Yes	No
jump	Yes	Yes
land	Yes	Yes
landlist	Yes	Yes
left	Yes	Yes
lload	Yes	Yes
lregister	Yes	Yes
lwrite	Yes	Yes
man	Yes	Yes
median	Yes	No
merge	Yes	No
mload	Yes	Yes
modland	Yes	Yes
move	Yes	Yes
move_vertex	No	Yes
movie_segment	No	Yes
mwrite	Yes	Yes
nameland	Yes	Yes
negsub	Yes	No
new_order	Yes	Yes
newland	Yes	Yes
option	Yes	Yes
ortho	Yes	Yes
pause	Yes	Yes
pick	Yes	Yes
pickmode	Yes	Yes
planes	Yes	Yes
pload	No	Yes
points	Yes	Yes
pop	Yes	Yes
possub	Yes	No
print	Yes	Yes
push	Yes	Yes
pwrite	No	Yes
recolor	Yes	No
refresh	Yes	Yes
remark	Yes	Yes
resample	Yes	Yes
rgb	Yes	Yes
right	Yes	Yes
rotate	Yes	Yes
ruin	Yes	No
select	Yes	Yes
set	Yes	No
shade	Yes	No
show	Yes	Yes
side	Yes	Yes
skip	Yes	Yes
sleep	Yes	Yes
smooth	Yes	No
split	Yes	No
store	Yes	Yes

subject	Yes	Yes
super	Yes	Yes
surface	Yes	Yes
surface_area	Yes	Yes
thin	Yes	No
thres	Yes	No
top	Yes	Yes
tops	Yes	Yes
toupee	Yes	No
transparent	Yes	Yes
trim	Yes	No
volume	Yes	No
walls	Yes	Yes
white	Yes	Yes
wireframe	Yes	Yes
wload	No	Yes
wwrite	No	Yes
xload	Yes	No
xwrite	Yes	No
zload	Yes	No
zregister	Yes	Yes
zwrite	Yes	No

APPENDIX E

FILE FORMATS: DESCRIPTION OF IMAGE DATA FILES SUPPORTED BY INTEGRATE VERSION 1.25

```
LAND_STUDY = <landmark study identifier>
STD_LAND = 42
AUX_LAND = <N> [number of auxiliary landmarks in dataset]
STANDARD =
 1 <lon> <lat> <radius> <x> <y> <z>
. . .
42 <lon> <lat> <radius> <x> <y> <z>
AUX =
 1 <lon> <lat> <radius> <x> <y> <z> <auxiliary landmark 1 name>
. . .
[N] <lon> <lat> <radius> <x> <y> <z> <auxiliary landmark N name>
END =
```

VIII. CARD Lab matrix file format

```
<original file name>
<original subject id or duplicate of file name>
<x1> <y1> <z1> <t1> [displacement matrix]
<x2> <y2> <z2> <t2>
<x3> <y3> <z3> <t3>
<x4> <y4> <z4> <t4>
THIN <lon thin> <lat thin> [sub-sampling intervals on lon and lat]
TRIM <low lon> <high lon> <low lat> <high lat> [include bounds on lon and lat]
CENTER <x> <y> <z> [displacement(s) from original object center]
```

negative vertexes (implying separate polygon definitions).

Example: 1 2 -3 defines a triangle using vertices 1, 2, and 3

III. Wavefront .obj format

Refer to Wavefront documentation/description.

Several variants of Stanford .ply format

This complex format is [no longer?] fully documented at the Stanford Web Site.

V. CARD Lab ASCII point format (grid and non-grid)

---

1. Grid format:

```
GRID <total_points> <nlon> <nlat> <lon1>:<lonn> <lat1>:<latn>
<lon lat radius x y z>
. . .
<lon lat radius x y z>
```

2. Non-Grid format:

```
CART <total_points>
<x y z>
. . .
<x y z>
```

3. CARD Lab contour file format:

```
CONT <total_points> 0 0 0:0 0:0
<x y z> [coordinates for 1st point in contour]
. . .
<x y z> [coordinates for last point in contour]
```

VI. Old CARD Lab Landmark File Format

```
5 1 1
<lon lat radius x y z> for standard landmark 1
. .
<N> 1
<lon lat radius x y z> for standard landmark 'N'
-1 -1 -1 [end of file tag]
```

VII. New CARD Lab Landmark File Format

```
SUBJECT_ID = <subject identifier>
CAN_TYPE   = <type identifier>
TUDY_NAME  = <acquisition study identifier>
```

File Formats Supported By INTEGRATE Version 1.25

I. Cyberware Scanner Format (new) (all(?) except WB-series scanners)

ASCII header with lines of the form <keyword>=<value>\n
Header terminates with DATA=\n
Binary cylindrical data, 2 bytes per radius. First radius is lon0 lat0. 2nd radius is lon0 lat1. Typically 512 longitudes and 256 latitudes, but header (NLG, NLT) is final authority. Each radius is multiplied by 2 ** RSHIFT (value from header). RSHIFT is typically either 3 or 5. Resulting radius value is in microns, so for an RSHIFT of 3, the radius is multiplied by .008 to get millimeters. Longitude proceeds clockwise (viewed from top) and latitude goes from bottom to top.

II. Movie.byu .g format

ASCII file with 4 sections:

A. Counts Line
<npart nvert npoly nedge>

npart - number of different parts in scene,
nvert - number of vertex points in scene,
npoly - number of unique polygons in scene,
nedge - total number of polygon edges in scene

B. Part-scene Definitions
<poly1first poly1last>
poly1first - index of first polygon in first scene part
(lowest = 1)
poly1last - index of last polygon in first scene part
(highest = npoly)
. . .
<polyNfirst polyNlast> (N = npoly)
There are a total of npart lines in this section,
or 2 * npart indexes.

C. Vertex Point Coordinates (x, y, z),
3 coordinates per Vertex, 1 or more Vertices per line.
There are a total of nvert Vertexs (or nvert * 3 coordinates) listed in this section.

D. Polygon Definitions
Polygons are defined by their vertexes. Edges are implied between adjacent vertexes in the list, and between the last vertex of a polygon and the first vertex in the polygon. The index of the last vertex of a polygon is indicated by negating the index. The lowest vertex index is 1. The largest is nvert. There are a total of nedge vertexes listed in this section. There should be a total of npoly

APPENDIX F

STEREOPHOTOGRAMMETRY: USERS' MANUAL FOR STEREO_SLICE

Information concerning the use of "Stereo People" data with INTEGRATE

I. Users' Manual for STEREO_SLICE

Stereo_Slice is a program which extracts datasets from the CARD Lab "stereo people" datasets (cstereo.mal and cstereo.fem) and creates datasets which are compatible with INTEGRATE. It creates 2 files for each segment in the pecified dataset. There are 19 segments in each dataset, as follows:

1. head
2. neck
3. thorax
4. abdomen
5. pelvis
6. right upper arm
7. right lower arm
8. right hand
9. left upper arm
10. left lower arm
11. left hand
12. right upper thigh
13. right lower thigh
14. right lower leg
15. right foot
16. left upper thigh
17. left lower thigh
18. left lower leg
19. left foot

The 2 files that are created for each segment are {m|f}<subj#>.<segment#>c hich contains cylindrical coordinates for each point in the original dataset, nd {m|f}<subj#>.<segment#>cs, which contains a centerpoint for each slice n the cylindrical coordinate file. Example: f2.1c is radius information for he head segment of female subject 2, while m5.2cs contains the slice centers for the neck segment for male subject 5. This format allows horizontal slices, which correspond to the original dataset points, but allows non-vertical center axes for the cylinder. The cylindrical data (c) is created in Cyberware Head Scanner or Cyberware Digitizer Data format, while the centerpoint data (cs) is an ASCII list of centerpoint coordinates, 1 per slice in the segment of the original dataset.

Usage of STEREO_SLICE: Stereo_Slice requires 2 parameters: the gender and subject number of the desired dataset, in the form {m|f}<subj#>, and the number of longitudes to create for each slice. Stereo_Slice also requires that the 2 stereo data files (cstereo.mal and cstereo.fem) be either in the local directory, or in a directory specified by an optional 3rd parameter. On CARD Lab systems, the cstereo files are in /home/code/stereo.

Example: "stereo_slice m2 32 /home/code/stereo" creates an INTEGRATE-compatible dataset for male subject 2, and generates 32 points for each slice of data.

II. Use of Stereo Data With INTEGRATE

ᶠter Stereo_Slice has been used to extract and convert the data for a ereo dataset into an INTEGRATE-compatible format, several special considerations are needed for INTEGRATE to use the stereo data.

First, INTEGRATE needs to load not only the radius files ({m|f}<subj#>.<segment#>c), but also the centers file ({m|f}<subj#>.<segment#>cs). There is a slightly different form of the CLOAD command (CLOADS) which indicates that the centers file is also required. The CLOADS command works identically to CLOAD, except that it also loads the centers file.

Second, in order to articulate the body segments, the segments must be linked together using the "SUPER LINK" command. The SUPER LINK command connects a segment to another segment at the approximately correct anatomical point, and sets the center of rotation of the subordinate segment at the connection point. The result is that the subordinate segment moves along with the owning segment when the owning segment is moved, and the subordinate segment rotates around the connection point when it is moved individually. Except for the lack of definition of anatomical rotation axes and limits, a properly linked set of segments can be articulated in a fairly natural ᵓy, with all segments responding appropriately to movement of other ments. There are several INTEGRATE script files which simplify the ɔ of loading and linking the segments of a stereo dataset. The CS file performs the CLOADS command for each of the 19 body segments for the specified subject. The LINK1 script links the body segments in a hierarchy starting at the thorax, assuming that the segments were loaded as INTEGRATE objects 1-19. Similarly, the LINK20 script links segments assuming that they were loaded as objects 20-38. The SIT1 and SIT20 scripts demonstrate articulation by rotating a subject into a sitting position, again assuming the segments are objects 1-19 or 20-38.

APPENDIX G

CYBERWARE WB4 DATA: CONVERTING WHOLE BODY SCANNER DATA INTO AN INTEGRATE-COMPATIBLE FORMAT

Converting Whole Body Scanner Data Into An INTEGRATE-Compatible Format

---

Preliminary Steps

0. At some earlier time, make an alignment file (per CyPie operating instructions) which synchronizes the images from the 4 scan head cameras. Name this file "best.align".
1. Perform the scan according to standard procedures.
2. Select the "Tesselate", "Normals", and "Texture Map" options from the Cyberware Process menu (creates polygons and color info for CyPie).
3. Click on the "Create" button. Wait until the Create button indicates done.
4. Select the "Save CyPie" Processing option and choose a file name when asked. (creates the CyPie input files)

Conversion/Retention Steps

5. Rename the resulting .color files to make them compatible with CyPie (The "colorname" script file will rename all relevant files automatically. Note that minor changes to the script may be required.)
6. Activate CyPie using the new scan name(s) and the best.align file per CyPie operating instructions.
7. Merge the scans and write them out per CyPie operating instructions. (Steps 6 and 7 can be performed semi-automatically using the "merge_scans" script file. Note that minor changes to the script may be required.) (Steps 5, 6, and 7 can be performed semi-automatically using the "fbproc" script file. Note that minor changes to the script may be required.)

9. The new .ply file is now directly useable by INTEGRATE using the PLOAD command.

COLORNAME script file - USAGE: colorname <base name of scan>
Example: colorname my_scan

```
#rename scanner color files
mv $1-000.color $1-000.ply.color
mv $1-075.color $1-075.ply.color
mv $1-180.color $1-180.ply.color
mv $1-255.color $1-255.ply.color
```

MERGE_SCANS script file - USAGE: merge_scans <base name of scan>
Example: merge_scans my_scan

```
#activate CyPie with correct options
cypie -c -p,best.align $1*.ply
#rename output files to prevent overwriting
mv out.ply ../zippered/$1.ply
```

FBPROC script file - USAGE: fbproc <base name of scan>
Example: fbproc my_scan

```
-----------------------------------------------------
#!/bin/csh
if ($#argv < 1) then
  echo "usage: `basename $0` <input file>"
  exit(0)
endif
#rename color files
colorname $1
#activate CyPie with correct options
merge_scans $1
#when # below removed, convert .ply into .g format (optional)
#plymovie $1.ply $1.g 4000
exit(0)
-----------------------------------------------------
```

U.S. Government Printing Office 1996 750-071/40020