INTEGRATE 1.28: A SOFTWARE TOOL FOR VISUALIZING, ANALYZING, AND MANIPULATING THREE-DIMENSIONAL DATA (U)

Dennis Burnsides
Patrick Files

SYTRONICS, INC.
4433 DAYTON-XENIA ROAD
DAYTON, OH 45432

Jennifer Whitestone

HUMAN EFFECTIVENESS DIRECTORATE
CREW SYSTEM INTERFACE DIVISION
WRIGHT-PATTERSON AFB, OHIO 45433-7022

20011025 102
JANUARY 1998

INTERIM REPORT FOR THE PERIOD JANUARY 1996 TO MARCH 1997

Approved for public release; distribution is unlimited.
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 Air Force Research 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

DISCLAIMER
This Technical Report is published as received and has not been edited by the Air Force Research Laboratory, Human Effectiveness Directorate.

TECHNICAL REVIEW AND APPROVAL
AFRL-HE-WP-TR-2000-0100

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.

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

FOR THE COMMANDER

MARIS M. VIKMANIS
Chief, Crew System Interface Division
Air Force Research Laboratory
Title and Subtitle: Integrate 1.28: A Software Tool for Visualizing, Analyzing, and Manipulating Three-Dimensional Data

Performing Organization Name(s) and Address(es):
Sytronics, Inc.
4433 Dayton-Xenia Road, Bldg 1
Dayton, OH 45432-1949

Sponsoring/monitoring agency name(s) and address(es):
Air Force Research Laboratory
Human Effectiveness Directorate
Crew System Interface Division
Air Force Materiel Command
Wright-Patterson AFB, OH 45433-7022

Abstract:
This report documents the functionality available using Integrate 1.28, 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.
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.
INTEGRATE 1.28: 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 1.28, offers new functionality to meet the needs of current USAF engineering anthropometry challenges. With Version 1.28, additional commands have been added to provide virtual anthropometric tools using calipers and tape (measure), to allow registration of two objects having common surface areas using surf_reg, to reduce an object in size uniformly using shrink, and to determine the distance between two objects using clearance. Researchers from businesses and universities worldwide use earlier versions of INTEGRATE to test and evaluate new helmet systems, develop augmentative files such as landmark data or contour information for surface scan databases, record human-equipment interface geometries, extract measurements equivalent to traditional anthropometry for whole body image data, and prepare surface data for rapid prototyping systems (Daanen et al., 1996; Brunsman et al., 1996; Brunsman et al., 1996; 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" through at least the first two tutorials before beginning a new session. There are tutorials that are targeted for users of both head scans and whole body surface data. 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 13 million points (100 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 (13 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 Irix 6.2, 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: 101_53p
LandFile: 101_53p.ind
Active: 5  Lon Thin: 1  Lat Thin: 1
Left: 0  Right: 512  Lower: 0  Upper: 256
Angles: X: 13.4  Y: 72.9  Z: 357.3
Center: X: 0.0  Y: 200.1  Z: 0.0
Offset: X: -4.2  Y: -61.9  Z: 47.3
```

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.

```
Walls:  100:1400:1300
Eye: X:0 Y:0 Z:700 Dist:700
Data Path: new_tut/
Memory Use: 11% (1408K)
PICK OFF
```

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:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>This is the active object.</td>
</tr>
<tr>
<td>+</td>
<td>This object is on the screen now, not hidden.</td>
</tr>
<tr>
<td>W</td>
<td>Wireframe display is on for this object.</td>
</tr>
<tr>
<td>P</td>
<td>Point display is on for this object.</td>
</tr>
<tr>
<td>S</td>
<td>Surface display is on for this object.</td>
</tr>
<tr>
<td>T</td>
<td>The surface display for this object is transparent.</td>
</tr>
<tr>
<td>l</td>
<td>Landmark display is on for this object.</td>
</tr>
</tbody>
</table>
| 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:
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 (100 head scans=100%).
- Pick Mode: Pick mode is on or off.
- Orthogonal View: If O is displayed on line 1, orthogonal view is enabled.
- Intrplnd: If * is displayed next to pickmode, intrplnd is enabled.

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**, **point picking**, and toggle options.

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 up/down 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.

2.3.4 Toggle Options
Many INTEGRATE commands enable or disable screen features and operating modes. A toggle option
may be set by typing `<option command> on`, cleared by typing `<option command> off`, or toggled by
typing only the `<option command>`. Most toggle options are tied to function keys which is where the
toggle feature is most valuable.
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 1.28. 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 Helmet Clearance
- Tutorial_7 Manipulating Stereophotogrammetry Data
- Tutorial_8 Manipulating Whole Body Data (Cyberware WB4)
- Tutorial_9 Calipers/Tape Measure
- Tutorial_10 Establishing Joint Centers on Whole Body 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. Shown in Figure 4 is the image "010_53p", a scan of an unencumbered subject with blue dots indicating the location of anatomical landmarks (shown here in monochrome).

Figure 4: Scan of an unencumbered subject (shown in monochrome for this publication).
The files needed for this tutorial are:

010_53p
010_53p.rgb
010_53p.ind

<table>
<thead>
<tr>
<th>Steps</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Enter INTEGRATE and begin session.</td>
<td>integrate</td>
</tr>
<tr>
<td>2. Load the unhelmeted scan file of the subject.</td>
<td>cload 010_53p</td>
</tr>
<tr>
<td>3. Load the anatomical landmark file associated with this image.</td>
<td>iload 010_53p.ind</td>
</tr>
<tr>
<td>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.</td>
<td>rotate 0 75</td>
</tr>
<tr>
<td>5. Change the eyepoint to view the image from the right. Note that this does not MOVE the object, it only changes your viewpoint.</td>
<td>right</td>
</tr>
<tr>
<td>6. Change the eyepoint to view the image from the back.</td>
<td>back</td>
</tr>
<tr>
<td>7. Change the eyepoint to view the image from the left.</td>
<td>left</td>
</tr>
<tr>
<td>8. Change the eyepoint to view the image from the top.</td>
<td>top</td>
</tr>
<tr>
<td>9. Change the eyepoint to view the image from the bottom.</td>
<td>bottom</td>
</tr>
<tr>
<td>10. Change the eyepoint to view the image from the front.</td>
<td>front</td>
</tr>
<tr>
<td>11. Move the object 50 mm along the x axis. Notice that the object is moving relative to the screen.</td>
<td>move 50</td>
</tr>
<tr>
<td>12. Move the object 50 mm along the y axis. Notice that the object is moving relative to the screen.</td>
<td>move 0 50</td>
</tr>
<tr>
<td>13. Change the eyepoint to view the image from the right.</td>
<td>right</td>
</tr>
<tr>
<td>14. Move the object 50 mm along the z axis. Notice that the object is moving relative to the screen.</td>
<td>move 0 0 50</td>
</tr>
<tr>
<td>15. Move the object back to the original origin.</td>
<td>move -50 -50 -50</td>
</tr>
<tr>
<td>16. Change the eyepoint back to the front.</td>
<td>front</td>
</tr>
<tr>
<td>17. Trim away extraneous image noise at the top and bottom of the subject.</td>
<td>trim 0 0 0 -55 \ trim 0 0 0 55</td>
</tr>
<tr>
<td>Steps</td>
<td>Commands</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>18. Make a copy of the object and save it as object #2.</td>
<td>copy 1 2</td>
</tr>
<tr>
<td>19. Create voids in the image data.</td>
<td>ruin 1 2</td>
</tr>
<tr>
<td>20. Hide the second object.</td>
<td>hide 2</td>
</tr>
<tr>
<td>21. Fill in the voids on the image and the gap at the top of the</td>
<td>1</td>
</tr>
<tr>
<td>subject's head. Notice that “1” was first selected to perform</td>
<td>do fill</td>
</tr>
<tr>
<td>the operations on the first object.</td>
<td>toupee 200 205</td>
</tr>
<tr>
<td>22. Turn off the axes and the status windows.</td>
<td>axes</td>
</tr>
<tr>
<td></td>
<td>boxes</td>
</tr>
<tr>
<td>23. Turn the status windows back on and change the representation</td>
<td>boxes</td>
</tr>
<tr>
<td>of the landmark locations from crosshairs to “L#” with the</td>
<td>alt_land</td>
</tr>
<tr>
<td>number denoting the landmark number found in the landmark file.</td>
<td></td>
</tr>
<tr>
<td>See Appendix C.</td>
<td></td>
</tr>
<tr>
<td>24. Turn on the landmark list to view the active landmarks and their</td>
<td>landlist</td>
</tr>
<tr>
<td>coordinates in the object coordinate system.</td>
<td></td>
</tr>
<tr>
<td>25. Turn off the landmark list and turn on the help list.</td>
<td>landlist help</td>
</tr>
<tr>
<td>26. Turn off the help list and turn on the function keys listing.</td>
<td>help fkeys</td>
</tr>
<tr>
<td>Turn off the landmarks.</td>
<td></td>
</tr>
<tr>
<td>27. Turn on the landmarks.</td>
<td>land</td>
</tr>
<tr>
<td>28. Turn off the wireframe mode and apply the surface routine to</td>
<td>wireframe surface fullcolor 010_53p rgb</td>
</tr>
<tr>
<td>the object to show texture and color.</td>
<td></td>
</tr>
<tr>
<td>29. Calculate the volume and surface area of the object.</td>
<td>volume surface surface_area</td>
</tr>
</tbody>
</table>
3.2 Tutorial Two: Registration Techniques

This tutorial demonstrates registration techniques used to visualize a subject within a helmet system, as shown in Figure 5. This registration technique can be used for examination of a subject within any protective equipment item. As shown in Figure 5, an "x-ray" view is provided, allowing the designer to look inside the human-equipment interface.

Figure 5: 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 (*.Ind) 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.ind
010_53ph, 010_53ph.rgb, 010_53ph.ind
53psize5, 53psize5.rgb, 53psize5.ind

<table>
<thead>
<tr>
<th>Steps</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Enter INTEGRATE and begin session.</td>
<td>integrate</td>
</tr>
<tr>
<td>2. Load the unhelmeted scan file of the subject.</td>
<td>cload 010_53p</td>
</tr>
<tr>
<td>3. Rotate the subject around the Y axis so he is directly facing you.</td>
<td>rotate 0 75</td>
</tr>
<tr>
<td>Notice that a positive Y axis value causes counterclockwise rotation.</td>
<td></td>
</tr>
<tr>
<td>4. Trim away extraneous image noise at the top and bottom of the</td>
<td>trim 0 0 0 -55</td>
</tr>
<tr>
<td>subject.</td>
<td>trim 0 0 55</td>
</tr>
<tr>
<td>5. Fill in voids on the subject and the gap at the top of the subject’s head.</td>
<td>do fill</td>
</tr>
<tr>
<td></td>
<td>toupee 200 205</td>
</tr>
<tr>
<td>6. Load the anatomical landmark file associated with this image.</td>
<td>lload 010_53p.ind</td>
</tr>
<tr>
<td>7. Load the helmeted scan file of the same subject with his helmet</td>
<td>cload 010_53ph</td>
</tr>
<tr>
<td>donned.</td>
<td></td>
</tr>
<tr>
<td>8. Trim the noise from this image.</td>
<td>trim 0 0 0 -50</td>
</tr>
<tr>
<td></td>
<td>trim 0 0 45 0</td>
</tr>
<tr>
<td>9. Load the landmark file associated with this image.</td>
<td>lload 010_53ph.ind</td>
</tr>
<tr>
<td>Steps</td>
<td>Commands</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>10. Register the helmeted image with the unhelmeted image. Notice</td>
<td>register 2 1</td>
</tr>
<tr>
<td>that the second image is rotated and translated into the coordinate</td>
<td></td>
</tr>
<tr>
<td>system of the first image, and that register is used to align the</td>
<td></td>
</tr>
<tr>
<td>scans as the common landmarks are anatomical landmarks.</td>
<td></td>
</tr>
<tr>
<td>11. Change the viewpoint to view the images from the right.</td>
<td>right</td>
</tr>
<tr>
<td>12. Use the walls command to “slice” through the data to examine the</td>
<td>walls 695 699</td>
</tr>
<tr>
<td>alignment of profiles.</td>
<td>walls +5</td>
</tr>
<tr>
<td></td>
<td>walls full</td>
</tr>
<tr>
<td>13. Change the eyepoint back to the front.</td>
<td>front</td>
</tr>
<tr>
<td>14. Load the helmet scan (and landmark file) for registration with the</td>
<td>cload 53psize5</td>
</tr>
<tr>
<td>helmeted image file.</td>
<td>lload 53psize5.Ind</td>
</tr>
<tr>
<td>15. Register the helmet scan with the helmeted image file and view</td>
<td>zregister 3 2</td>
</tr>
<tr>
<td>the alignment. Notice that zregister is used to align the scans as</td>
<td>right</td>
</tr>
<tr>
<td>the common landmarks are auxiliary landmarks.</td>
<td>walls 698 699</td>
</tr>
<tr>
<td></td>
<td>walls +5</td>
</tr>
<tr>
<td></td>
<td>walls +5</td>
</tr>
<tr>
<td>16. Hide the helmeted scan and show only the subject and the scan of</td>
<td>hide 2</td>
</tr>
<tr>
<td>the helmet alone. This final configuration illustrates the position</td>
<td>walls full</td>
</tr>
<tr>
<td>of the subject within the helmet.</td>
<td></td>
</tr>
<tr>
<td>17. Change the subject file to a surface and the helmet scan to a</td>
<td>1</td>
</tr>
<tr>
<td>wireframe of lower resolution.</td>
<td>wireframe</td>
</tr>
<tr>
<td></td>
<td>surface</td>
</tr>
<tr>
<td></td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>thin 2 2</td>
</tr>
<tr>
<td>18. Change the helmet scan to a transparent surface.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>wireframe</td>
</tr>
<tr>
<td></td>
<td>surface</td>
</tr>
<tr>
<td></td>
<td>transparent</td>
</tr>
<tr>
<td>19. Change the subject file to represent the color information.</td>
<td>fullcolor 010_53p</td>
</tr>
<tr>
<td></td>
<td>rgb</td>
</tr>
<tr>
<td>Steps</td>
<td>Commands</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>20. View this configuration from different viewpoints.</td>
<td>front</td>
</tr>
<tr>
<td></td>
<td>left</td>
</tr>
<tr>
<td></td>
<td>back</td>
</tr>
<tr>
<td></td>
<td>front</td>
</tr>
</tbody>
</table>
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 D. If a new landmark picking order is required, the command **new_order** can be used to establish a order for picking the points. Figure 6 shows a head scan file with anatomical landmarks.

![Figure 6: Subject scan with color file (monochrome for this publication) and marked landmark locations.](image)
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 D.

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

<table>
<thead>
<tr>
<th>Steps</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Enter INTEGRATE and begin session.</td>
<td>integrate</td>
</tr>
<tr>
<td>2. Load the head scan file.</td>
<td>cload 010_53p</td>
</tr>
<tr>
<td>3. Trim away extraneous image noise at the top and bottom of the</td>
<td>trim 0 0 0 -50</td>
</tr>
<tr>
<td>subject.</td>
<td>trim 0 0 58</td>
</tr>
<tr>
<td>4. Fill in voids on the subject and the gap at the top of the subject’s head.</td>
<td>do fill</td>
</tr>
<tr>
<td></td>
<td>toupee 200 205</td>
</tr>
<tr>
<td>5. Change the surface from wireframe to color representation. Color mode clearly displays the color landmark dots.</td>
<td>wireframe</td>
</tr>
<tr>
<td></td>
<td>surface</td>
</tr>
<tr>
<td></td>
<td>fullcolor 010_53p</td>
</tr>
<tr>
<td></td>
<td>rgb</td>
</tr>
<tr>
<td>6. Rotate the subject to the right to prepare for landmark selection.</td>
<td>rotate 0 75</td>
</tr>
<tr>
<td>7. Begin landmarking session.</td>
<td>pick on</td>
</tr>
<tr>
<td></td>
<td>pickmode land</td>
</tr>
<tr>
<td>Steps</td>
<td>Commands</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>8. Use the mouse to put the cursor on the first landmark to be picked, the right trigion (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.</td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td><code>skip -1</code></td>
</tr>
<tr>
<td>10. Save the landmark locations to a landmark file.</td>
<td><code>lwrite land_010.lnd</code></td>
</tr>
<tr>
<td>11. Display the landmark list to view the landmark coordinates.</td>
<td><code>landlist</code></td>
</tr>
<tr>
<td>12. Turn off the landmark list and turn the function key display back on.</td>
<td><code>landlist fkeys</code></td>
</tr>
<tr>
<td>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.</td>
<td><code>cload 010_53p 010_53p.lnd right rotate 0 75</code></td>
</tr>
<tr>
<td>14. Change the original subject back to wireframe to compare the landmarks.</td>
<td><code>1 surface wireframe</code></td>
</tr>
<tr>
<td>15. Change the viewpoint to examine the scans from different views.</td>
<td><code>front left back front</code></td>
</tr>
</tbody>
</table>
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 7 illustrates two subjects wearing the same helmet.

![Figure 7: Two subjects wearing the same size and model helmet.](image)
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 8.

Figure 8: 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 (*.ind) 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.ind
100_53p, 100_53p.rgb, 100_53p.ind
101_53p, 101_53p.rgb, 101_53p.ind
104_53p, 104_53p.rgb, 104_53p.ind
105_53p, 105_53p.rgb, 105_53p.ind
100_53ph, 100_53ph.rgb, 100_53ph.ind
101_53ph, 101_53ph.rgb, 101_53ph.ind
102_53ph, 102_53ph.rgb, 102_53ph.ind
104_53ph, 104_53ph.rgb, 104_53ph.ind
105_53ph, 105_53ph.rgb, 105_53ph.ind

<table>
<thead>
<tr>
<th>Steps</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Enter INTEGRATE and begin session.</td>
<td>integrate</td>
</tr>
<tr>
<td>2. Load the helmet scan with the helmet reference landmarks.</td>
<td>cload 53psize5 53psize5.ind</td>
</tr>
<tr>
<td>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.</td>
<td>align xz z1 z5 z3 z3</td>
</tr>
<tr>
<td>4. Load the first subject's encumbered (helmeted) scan with landmark files containing both helmet reference landmarks and anatomical landmarks.</td>
<td>cload 100_53ph 100_53ph.ind</td>
</tr>
<tr>
<td>5. Register (align) this scan with the helmet scan using the common helmet landmarks.</td>
<td>zregister 2 1</td>
</tr>
<tr>
<td>Steps</td>
<td>Commands</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>6. Load the first subject’s unencumbered (bare head) scan with</td>
<td>cload 100_53p 100_53p.ind</td>
</tr>
<tr>
<td>anatomical landmarks.</td>
<td></td>
</tr>
<tr>
<td>7. Register (align) this scan with the encumbered scan. In effect,</td>
<td>Iregister 3 2</td>
</tr>
<tr>
<td>this aligns the subject with the helmet scan.</td>
<td></td>
</tr>
<tr>
<td>8. Copy the pupil landmark locations to the helmet scan. This</td>
<td>1</td>
</tr>
<tr>
<td>associates the pupil locations with respect to the global helmet</td>
<td>copyland l1 3 l34</td>
</tr>
<tr>
<td>system.</td>
<td>copyland l2 3 l38</td>
</tr>
<tr>
<td>9. Load the second subject’s encumbered (helmeted) scan with</td>
<td>cload 101_53ph 101_53ph.ind</td>
</tr>
<tr>
<td>landmark files containing both helmet reference landmarks and</td>
<td></td>
</tr>
<tr>
<td>anatomical landmarks.</td>
<td></td>
</tr>
<tr>
<td>10. Register (align) this scan with the helmet scan using the</td>
<td>zregister 4 1</td>
</tr>
<tr>
<td>common helmet landmarks.</td>
<td></td>
</tr>
<tr>
<td>11. Load the second subject’s unencumbered (bare head) scan with</td>
<td>cload 101_53p 101_53p.ind</td>
</tr>
<tr>
<td>anatomical landmarks.</td>
<td></td>
</tr>
<tr>
<td>12. Register (align) this scan with the encumbered scan. In effect,</td>
<td>Iregister 5 4</td>
</tr>
<tr>
<td>this aligns the subject with the helmet scan.</td>
<td></td>
</tr>
<tr>
<td>13. Copy the pupil landmark locations to the helmet scan. This</td>
<td>1</td>
</tr>
<tr>
<td>associates the pupil locations with respect to the global helmet</td>
<td>copyland l3 5 l34</td>
</tr>
<tr>
<td>system.</td>
<td>copyland l4 5 l38</td>
</tr>
<tr>
<td>14. Load the third subject’s encumbered (helmeted) scan with</td>
<td>cload 101_53ph 101_53ph.ind</td>
</tr>
<tr>
<td>landmark files containing both helmet reference landmarks and</td>
<td></td>
</tr>
<tr>
<td>anatomical landmarks.</td>
<td></td>
</tr>
<tr>
<td>15. Register (align) this scan with the helmet scan using the</td>
<td>zregister 6 1</td>
</tr>
<tr>
<td>common helmet landmarks.</td>
<td></td>
</tr>
<tr>
<td>16. Load the third subject’s unencumbered (bare head) scan with</td>
<td>cload 102_53p 102_53p.ind</td>
</tr>
<tr>
<td>anatomical landmarks.</td>
<td></td>
</tr>
<tr>
<td>17. Register (align) this scan with the encumbered scan. In effect,</td>
<td>Iregister 7 6</td>
</tr>
<tr>
<td>this aligns the subject with the helmet scan.</td>
<td></td>
</tr>
<tr>
<td>18. Copy the pupil landmark locations to the helmet scan. This</td>
<td>1</td>
</tr>
<tr>
<td>associates the pupil locations with respect to the global helmet</td>
<td>copyland l5 7 l34</td>
</tr>
<tr>
<td>system.</td>
<td>copyland l6 7 l38</td>
</tr>
<tr>
<td>Steps</td>
<td>Commands</td>
</tr>
<tr>
<td>---------------------------------------------------------------------</td>
<td>---------------------------------------------------------</td>
</tr>
<tr>
<td>19. Load the fourth subject’s encumbered (helmeted) scan with</td>
<td>cload 104_53ph 104_53ph.lnd</td>
</tr>
<tr>
<td>landmark files containing both helmet reference landmarks and</td>
<td></td>
</tr>
<tr>
<td>anatomical landmarks.</td>
<td></td>
</tr>
<tr>
<td>20. Register (align) this scan with the helmet scan using the common</td>
<td>zregister 8 1</td>
</tr>
<tr>
<td>helmet landmarks.</td>
<td></td>
</tr>
<tr>
<td>21. Load the fourth subject’s unencumbered (bare head) scan with</td>
<td>cload 104_53p 104_53p.lnd</td>
</tr>
<tr>
<td>anatomical landmarks.</td>
<td></td>
</tr>
<tr>
<td>22. Register (align) this scan with the encumbered scan. In effect,</td>
<td>iregister 9 8</td>
</tr>
<tr>
<td>this aligns the subject with the helmet scan.</td>
<td></td>
</tr>
<tr>
<td>23. Copy the pupil landmark locations to the helmet scan. This</td>
<td>1</td>
</tr>
<tr>
<td>associates the pupil locations with respect to the global helmet</td>
<td>copyland 17 9 134</td>
</tr>
<tr>
<td>system.</td>
<td>copyland 18 9 138</td>
</tr>
<tr>
<td>24. Load the fifth subject’s encumbered (helmeted) scan with</td>
<td>cload 105_53ph 105_53ph.lnd</td>
</tr>
<tr>
<td>landmark files containing both helmet reference landmarks and</td>
<td></td>
</tr>
<tr>
<td>anatomical landmarks.</td>
<td></td>
</tr>
<tr>
<td>25. Register (align) this scan with the helmet scan using the common</td>
<td>zregister 10 1</td>
</tr>
<tr>
<td>helmet landmarks.</td>
<td></td>
</tr>
<tr>
<td>26. Load the fifth subject’s unencumbered (bare head) scan with</td>
<td>cload 105_53p 105_53p.lnd</td>
</tr>
<tr>
<td>anatomical landmarks.</td>
<td></td>
</tr>
<tr>
<td>27. Register (align) this scan with the encumbered scan. In effect,</td>
<td>iregister 11 10</td>
</tr>
<tr>
<td>this aligns the subject with the helmet scan.</td>
<td></td>
</tr>
<tr>
<td>28. Copy the pupil landmark locations to the helmet scan. This</td>
<td>1</td>
</tr>
<tr>
<td>associates the pupil locations with respect to the global helmet</td>
<td>copyland 19 11 134</td>
</tr>
<tr>
<td>system.</td>
<td>copyland 20 11 138</td>
</tr>
<tr>
<td>Steps</td>
<td>Commands</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>29. Hide all objects except the helmet scan with the pupil feature envelopes.</td>
<td>hide 2</td>
</tr>
<tr>
<td></td>
<td>hide 3</td>
</tr>
<tr>
<td></td>
<td>hide 4</td>
</tr>
<tr>
<td></td>
<td>hide 5</td>
</tr>
<tr>
<td></td>
<td>hide 6</td>
</tr>
<tr>
<td></td>
<td>hide 7</td>
</tr>
<tr>
<td></td>
<td>hide 8</td>
</tr>
<tr>
<td></td>
<td>hide 9</td>
</tr>
<tr>
<td></td>
<td>hide 10</td>
</tr>
<tr>
<td></td>
<td>hide 11</td>
</tr>
<tr>
<td>30. View the pupil envelopes for the helmet from different viewpoints.</td>
<td>right</td>
</tr>
<tr>
<td></td>
<td>left</td>
</tr>
<tr>
<td></td>
<td>back</td>
</tr>
<tr>
<td></td>
<td>front</td>
</tr>
</tbody>
</table>
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 9 is an RDM of a subject’s head scan and a scan of his mask.

Figure 9: 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 (*.ind) 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

<table>
<thead>
<tr>
<th>Steps</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Enter INTEGRATE and begin session.</td>
<td>integrate</td>
</tr>
<tr>
<td>2. Load the head scan file.</td>
<td>cload face</td>
</tr>
<tr>
<td>3. Move the scan to the center of the screen.</td>
<td>move 0 200</td>
</tr>
<tr>
<td>4. Load the scan of the subject’s mask and move it to the center of the screen.</td>
<td>cload mask</td>
</tr>
<tr>
<td>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.</td>
<td>possub 1 2</td>
</tr>
<tr>
<td>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.</td>
<td>threshold 2 ge 1</td>
</tr>
<tr>
<td>7. Identify these radial values for the face scan. The results will be saved as the second object.</td>
<td>and 1 2</td>
</tr>
<tr>
<td>8. Perform these steps again using another scan file of the mask.</td>
<td>cload mask</td>
</tr>
<tr>
<td>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.</td>
<td>negsub 1 3</td>
</tr>
<tr>
<td>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.</td>
<td>threshold 2 ge 1</td>
</tr>
<tr>
<td>11. Identify these radial values for the face scan. The results will be saved as the third object.</td>
<td>and 1 3</td>
</tr>
<tr>
<td>Steps</td>
<td>Commands</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>12. Perform these steps again using another scan file of the mask.</td>
<td>clload mask</td>
</tr>
<tr>
<td></td>
<td>move 0 200</td>
</tr>
<tr>
<td>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.</td>
<td>possub 1 4</td>
</tr>
<tr>
<td>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.</td>
<td>threshold 4 lt 1</td>
</tr>
<tr>
<td>15. Identify these radial values for the face scan. The results will be saved as the fourth object.</td>
<td>and 1 4</td>
</tr>
<tr>
<td>16. Perform these steps again using another scan file of the mask.</td>
<td>clload mask</td>
</tr>
<tr>
<td></td>
<td>move 0 200</td>
</tr>
<tr>
<td>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.</td>
<td>negsub 1 5</td>
</tr>
<tr>
<td>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.</td>
<td>threshold 5 lt 1</td>
</tr>
<tr>
<td>19. Identify these radial values for the face scan. The results will be saved as the fifth object.</td>
<td>and 1 5</td>
</tr>
</tbody>
</table>
3.6 Tutorial Six: Helmet Clearance

This tutorial demonstrates an analysis of the fit of a PASGT helmet as actually worn by a subject. The fit criterion is that the inner surface of the helmet should be at least 12.5 mm from the head at all points. To test this criterion, first a helmeted scan is aligned with an unhelmeted scan, then the helmet alone is aligned with the helmeted scan. This brings the bare head and the helmet into the same relationship observed when the subject was wearing the helmet.

To approximate the inner surface of the helmet, we shrink a copy of the helmet outer surface 9.4 mm, which is the reported thickness of the PASGT helmet. This is not exact, since, among other things, the rivets that bulge on the outside of the helmet would also bulge inward on the inside of the helmet, which is not true of our approximated inner surface, but the analysis procedures remain the same when a more accurate inner surface model is available.

Note also the surface of the head is approximated by a cap placed over the hair to compress the hair and improve the visibility of the laser beam during scanning. This is also an approximation, and to be more exact, the depth of the cap and compressed hair should be determined and subtracted from the scanned head surface before checking the clearance.

After bringing the approximated helmet inner surface into the observed relationship with the approximated outer head surface, we begin our clearance check.

After the clearance check is complete, we have 2 products: a new object which represents all points on the head closer than the specified clearance criterion, and a histogram and list of the closest point on the helmet for each point on the head. The surface area of the new object can be determined as a measure of the fit of the helmet on the subject. Similarly, the histogram can be plotted to analyze the clearance distances between the helmet and head.
Figure 10: Picture of the helmet clearance with respect to the subject’s head scan.

In this example, the landmarks have been identified and saved to a landmark (*.ind) file for each image file. “head” files are of the unencumbered scans, “pasgt” files are of the helmet datasets, and “head.pasgt” are of the head in the helmet.
The following files are needed for this tutorial:

- head.g, head.lnd
- pasgt.g, pasgt.lnd
- head.pasgt.cdd, head.pasgt.lnd

Files produced by check_clearance script:

- head.results (list of closest points and histogram of closest distances)
- head.pasgt.fail.g (dataset with all head points failing clearance criterion)

<table>
<thead>
<tr>
<th>Steps</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Enter INTEGRATE and begin session.</td>
<td>integrate</td>
</tr>
</tbody>
</table>
| 2. Turn off the wireframe mode and turn on the points option for all subsequent loaded objects. | option wireframe off  
   option points on  
   eye 0 0 1200 |
| 3. Load unencumbered head scan. | gload head.g  
  lload head.lnd |
| 4. Load outer helmet surface. | gload pasgt.g  
  lload pasgt.lnd |
| 5. Load helmet head scan and hide this image. | cload head.pasgt.cdd  
  lload head.pasgt.lnd  
  hide |
| 6. Register the helmeted scan to the head scan and then register the helmet with the helmeted scan. | lregister 3 1  
  zregister 2 3 |
| 7. Approximate the inner helmet surface by creating another helmet object (object #4). | 2  
   movie_seg -  
   land off  
   shrink 9.4 0 0 0 |
| 8. Remove irrelevant points from head scan to speed things up and create another head object (object #5). | 1  
   movie_seg ly110  
   hide 1  
   refresh |
<table>
<thead>
<tr>
<th>Steps</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>9. Create object #6 which compares the area of the head (object #5)</td>
<td>clearance 4.5 12.5 head.results</td>
</tr>
<tr>
<td>within 12.5 mm of inner helmet surface (object #4) with a 0.5 mm</td>
<td></td>
</tr>
<tr>
<td>histogram bin size, histogram and intermediate results stored in</td>
<td></td>
</tr>
<tr>
<td>head.results. (This operation takes about an hour on a 133-Mhz SG</td>
<td></td>
</tr>
<tr>
<td>Indy. Plan accordingly.) NOTE: this may not be complete due to</td>
<td></td>
</tr>
<tr>
<td>missing data at top of head)</td>
<td></td>
</tr>
<tr>
<td>10. Store object containing all points failing clearance criterion.</td>
<td>gwrite head.pasgt.fall.g</td>
</tr>
</tbody>
</table>
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 11. Stereophotogrammetry was used in the 1970’s to acquire whole body surface data (McConville, et al., 1980). Appendix H 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 11. 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,

<table>
<thead>
<tr>
<th>Steps</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Enter INTEGRATE and begin session.</td>
<td>integrate</td>
</tr>
<tr>
<td>2. Turn the wireframe option off and the surface on.</td>
<td>option wireframe off</td>
</tr>
<tr>
<td></td>
<td>option surface on</td>
</tr>
<tr>
<td>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.</td>
<td>cloads -a m2.1c</td>
</tr>
<tr>
<td></td>
<td>cloads -a m2.2c</td>
</tr>
<tr>
<td></td>
<td>cloads -a m2.3c</td>
</tr>
<tr>
<td></td>
<td>cloads -a m2.4c</td>
</tr>
<tr>
<td></td>
<td>cloads -a m2.5c</td>
</tr>
<tr>
<td></td>
<td>cloads -a m2.6c</td>
</tr>
<tr>
<td></td>
<td>cloads -a m2.7c</td>
</tr>
<tr>
<td></td>
<td>cloads -a m2.8c</td>
</tr>
<tr>
<td></td>
<td>cloads -a m2.9c</td>
</tr>
<tr>
<td></td>
<td>cloads -a m2.10c</td>
</tr>
<tr>
<td></td>
<td>cloads -a m2.11c</td>
</tr>
<tr>
<td></td>
<td>cloads -a m2.12c</td>
</tr>
<tr>
<td></td>
<td>cloads -a m2.13c</td>
</tr>
<tr>
<td></td>
<td>cloads -a m2.14c</td>
</tr>
<tr>
<td></td>
<td>cloads -a m2.15c</td>
</tr>
<tr>
<td></td>
<td>cloads -a m2.16c</td>
</tr>
<tr>
<td></td>
<td>cloads -a m2.17c</td>
</tr>
<tr>
<td></td>
<td>cloads -a m2.18c</td>
</tr>
<tr>
<td></td>
<td>cloads -a m2.19c</td>
</tr>
<tr>
<td>Steps</td>
<td>Commands</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>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.</td>
<td>super link 2 1</td>
</tr>
<tr>
<td></td>
<td>super link 3 2</td>
</tr>
<tr>
<td>5. Link the right arm to the torso.</td>
<td>super link 7 8</td>
</tr>
<tr>
<td></td>
<td>super link 6 7</td>
</tr>
<tr>
<td></td>
<td>super link 3 6</td>
</tr>
<tr>
<td>6. Link the left arm to the torso.</td>
<td>super link 10 11</td>
</tr>
<tr>
<td></td>
<td>super link 9 10</td>
</tr>
<tr>
<td></td>
<td>super link 3 9</td>
</tr>
<tr>
<td>7. Link the right leg to the torso.</td>
<td>super link 14 15</td>
</tr>
<tr>
<td></td>
<td>super link 13 14</td>
</tr>
<tr>
<td></td>
<td>super link 12 13</td>
</tr>
<tr>
<td></td>
<td>super link 5 12</td>
</tr>
<tr>
<td>8. Link the left leg to the torso.</td>
<td>super link 18 19</td>
</tr>
<tr>
<td></td>
<td>super link 17 18</td>
</tr>
<tr>
<td></td>
<td>super link 16 17</td>
</tr>
<tr>
<td></td>
<td>super link 5 16</td>
</tr>
<tr>
<td>9. Link the lower torso to the upper torso.</td>
<td>super link 4 5</td>
</tr>
<tr>
<td></td>
<td>super link 3 4</td>
</tr>
<tr>
<td>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.</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>move 0 -80</td>
</tr>
<tr>
<td>11. Slightly bend torso at waist.</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>rotate -10</td>
</tr>
<tr>
<td>12. Rotate legs.</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>rotate -80</td>
</tr>
<tr>
<td></td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>rotate 90</td>
</tr>
<tr>
<td></td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>rotate -80</td>
</tr>
<tr>
<td></td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>rotate 90</td>
</tr>
<tr>
<td>Steps</td>
<td>Commands</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>13. Rotate arms.</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>rotate 0 30</td>
</tr>
<tr>
<td></td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>rotate -90</td>
</tr>
<tr>
<td></td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>rotate 0 -100</td>
</tr>
<tr>
<td></td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>rotate 0 -30</td>
</tr>
<tr>
<td></td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>rotate -90</td>
</tr>
<tr>
<td></td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>rotate 0 70</td>
</tr>
<tr>
<td>14. Turn the subject to the right and rotate</td>
<td>right</td>
</tr>
<tr>
<td>the entire subject 360 degrees.</td>
<td>rotate 0 30</td>
</tr>
<tr>
<td></td>
<td>(repeat this 11 times)</td>
</tr>
</tbody>
</table>
3.8 Tutorial Eight: Manipulating Whole Body Data (Cyberware WB4)

This tutorial demonstrates visualization and manipulation of whole body scan data. 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 12.

Figure 12: Full body scan data with the body segments separated from the torso.
The files needed for this tutorial are:

- dr.boff.g
- std.mtx

<table>
<thead>
<tr>
<th>Steps</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Clear the screen of axes, boxes, and function key commands.</td>
<td>axes off</td>
</tr>
<tr>
<td></td>
<td>boxes off</td>
</tr>
<tr>
<td></td>
<td>fkeys off</td>
</tr>
<tr>
<td>2. Expand the walls and change the eyepoint to allow full</td>
<td>walls 10 7000</td>
</tr>
<tr>
<td>visualization of the whole body image.</td>
<td>eyedist 3000</td>
</tr>
<tr>
<td>3. Move the eyepoint to the front.</td>
<td>front</td>
</tr>
<tr>
<td>4. Turn the wireframe off and the points on.</td>
<td>option wireframe off</td>
</tr>
<tr>
<td></td>
<td>option points on</td>
</tr>
<tr>
<td>5. Load the image file and a transformation matrix to position it in</td>
<td>gload dr.boff.g</td>
</tr>
<tr>
<td>the middle of the screen. From this viewpoint, the subject is viewed</td>
<td>mload std.mtx</td>
</tr>
<tr>
<td>from the side.</td>
<td></td>
</tr>
<tr>
<td>6. To view the object from the front, the “eye” command is used.</td>
<td>eye 3700</td>
</tr>
<tr>
<td>7. Turn on the pick mode and move the subject forward slightly so</td>
<td>pick on</td>
</tr>
<tr>
<td>that the entire object can be seen.</td>
<td>move 100 70</td>
</tr>
<tr>
<td>8. Turn the points off and the surface on.</td>
<td>points off</td>
</tr>
<tr>
<td></td>
<td>surface on</td>
</tr>
<tr>
<td>9. Turn the surface off and the points on.</td>
<td>surface</td>
</tr>
<tr>
<td>Notice that “off” and “on” are optional.</td>
<td>points</td>
</tr>
<tr>
<td>10. Segment the left arm. Note: object #1 is selected before</td>
<td>movie_seg uz-234 1</td>
</tr>
<tr>
<td>performing the next operation.</td>
<td></td>
</tr>
<tr>
<td>11. Segment the right arm.</td>
<td>movie_seg lz260 1</td>
</tr>
<tr>
<td>12. Segment the head.</td>
<td>movie_seg ly610 1</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>13. Segment the torso.</td>
<td>movie_seg ly120 uy609 lz-233 uz259 1</td>
</tr>
<tr>
<td>Steps</td>
<td>Commands</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>14. Segment the legs.</td>
<td>movie_seg uy119 lz-233 uz259</td>
</tr>
<tr>
<td>15. Hide object #1.</td>
<td>hide 1</td>
</tr>
</tbody>
</table>
| 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 |
3.9 Tutorial Nine: Calipers/Tape Measure

This tutorial demonstrates the use of virtual calipers and tape measure on the whole body data set. For this example, a whole body scan is loaded into INTEGRATE and the calipers are used to measure chest depth and the tape measure is used to record waist circumference at omphalion. Shown in Figure 13 is the whole body data with the calipers.

Figure 13: Full body scan data with the virtual calipers.
The files needed for this tutorial are:

- tsa_stda.ply, std2.mtx
- c50x25.g
- calipers.g,color, caliper.mtx, waist_circ.mtx

<table>
<thead>
<tr>
<th>Steps</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Enter INTEGRATE and begin session.</td>
<td>integrate</td>
</tr>
<tr>
<td>2. Establish general setup options.</td>
<td>rgb on</td>
</tr>
<tr>
<td></td>
<td>gouraud on</td>
</tr>
<tr>
<td></td>
<td>option wireframe off</td>
</tr>
<tr>
<td></td>
<td>option points off</td>
</tr>
<tr>
<td></td>
<td>option surface on</td>
</tr>
<tr>
<td>3. Load a whole body data set.</td>
<td>pload tsa_stda.ply -1000</td>
</tr>
<tr>
<td></td>
<td>mload std2.mtx</td>
</tr>
<tr>
<td>4. Load a caliper object and turn the calipers on.</td>
<td>gload c50x25.g</td>
</tr>
<tr>
<td></td>
<td>fullcolor calipers.g</td>
</tr>
<tr>
<td></td>
<td>calipers on</td>
</tr>
<tr>
<td>5. Orient the calipers for chest depth measurement on the whole body scan.</td>
<td>mload caliper.mtx</td>
</tr>
<tr>
<td></td>
<td>jaw 2 150</td>
</tr>
<tr>
<td>6. Display automatic calipers.</td>
<td>top</td>
</tr>
<tr>
<td></td>
<td>auto_jaws 1</td>
</tr>
<tr>
<td>7. Set walls to show calipers.</td>
<td>walls 640 643</td>
</tr>
<tr>
<td>8. Establish cyberware object to generate circumference measurement. <strong>Movie_seg</strong> is used to segment the torso from the whole body and <strong>resample</strong> is used to create a cyberware object which is compatible with commands such as <strong>cir3p</strong>. Note: the segmented torso object must be translated such that the y axis is centered in the middle of the torso. This is to ensure that the resampled object has a uniform resolution.</td>
<td>walls full</td>
</tr>
<tr>
<td></td>
<td>movie_seg uy40 ly-20</td>
</tr>
<tr>
<td></td>
<td>hide 1</td>
</tr>
<tr>
<td></td>
<td>hide 2</td>
</tr>
<tr>
<td></td>
<td>top</td>
</tr>
<tr>
<td></td>
<td>mload waist_circ.mtx</td>
</tr>
<tr>
<td></td>
<td>resample</td>
</tr>
<tr>
<td></td>
<td>hide 3</td>
</tr>
<tr>
<td>Steps</td>
<td>Commands</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>9. The function to pick points is now turned on and pickmode cir3p is</td>
<td>pick on</td>
</tr>
<tr>
<td>selected to establish a horizontal cut plane through the waist to</td>
<td>pickmode cir3p</td>
</tr>
<tr>
<td>establish a contour representative of waist circumference.</td>
<td></td>
</tr>
<tr>
<td>10. Use the mouse to put the cursor at the location of the first (of</td>
<td></td>
</tr>
<tr>
<td>three) landmarks to be picked. These are points, in this case, selected</td>
<td></td>
</tr>
<tr>
<td>by the user to represent the level at which the circumference will be</td>
<td></td>
</tr>
<tr>
<td>taken. Space the three landmarks horizontally across the abdomen.</td>
<td></td>
</tr>
<tr>
<td>Look in the global status window for the prompt that lists the next</td>
<td></td>
</tr>
<tr>
<td>landmark to be picked.</td>
<td></td>
</tr>
<tr>
<td>11. Change the view to see the contour shape.</td>
<td>top</td>
</tr>
<tr>
<td></td>
<td>surface</td>
</tr>
<tr>
<td>12. Determine the distance of the exact contour. Note: “2” in this</td>
<td>distance 2</td>
</tr>
<tr>
<td>case is the number associated with the contour. This number may</td>
<td></td>
</tr>
<tr>
<td>change and will be visible with the contour. (Hit “space bar” after</td>
<td></td>
</tr>
<tr>
<td>the measurement is displayed.)</td>
<td></td>
</tr>
<tr>
<td>13. Now determine the distance of the contour with the virtual tape</td>
<td>tape 2</td>
</tr>
<tr>
<td>measure. Notice the measurement is slightly smaller and more</td>
<td>distance 2</td>
</tr>
<tr>
<td>representative of an actual tape measure.</td>
<td></td>
</tr>
</tbody>
</table>
3.10 Establishing Joint Centers on Whole Body Data

This tutorial demonstrates the ability to create landmarks representing joint centers given a set of anatomical landmarks located on a whole body data set. The joint centers are estimates and are assumed to be derived from landmarks representing bony structures on the body. An example is establishing the kneew joint as the midpoint between the medial and lateral femoral condyles. A description of the joint center estimates is found in Appendix I. A landmark list complete with illustrations and descriptions is found in Appendix C. Shown in Figure 14 is the whole body with the estimated joint center locations.

Figure 14: Full body scan data with the estimated joint center locations.
The files needed for this tutorial are:

- chrisla.ply
- chrisla.ind, std.mtx

<table>
<thead>
<tr>
<th>Steps</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Enter INTEGRATE and begin session.</td>
<td>integrate</td>
</tr>
<tr>
<td>2. Load the whole body data set twice. (One object is to associate</td>
<td>pload chrisla.ply -1000</td>
</tr>
<tr>
<td>with the anatomical landmarks and the other is to associate with the</td>
<td>mload std.mtx</td>
</tr>
<tr>
<td>joint centers.)</td>
<td>pload chrisla.ply -1000</td>
</tr>
<tr>
<td></td>
<td>mload std.mtx</td>
</tr>
<tr>
<td>3. Load the anatomical landmarks for object #2.</td>
<td>lload chrisla.ind</td>
</tr>
<tr>
<td>6. Establish the head/neck joint center and copy it to object #1.</td>
<td>split l1 z72 z64</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>copyland l1 2 l1</td>
</tr>
<tr>
<td>7. Establish the neck/thorax joint center and copy it to object #1.</td>
<td>copyland l2 2 z42</td>
</tr>
<tr>
<td></td>
<td>add_to_land l2 0 -25 51</td>
</tr>
<tr>
<td>8. Establish the thorax/abdomen joint center and copy it to object #1.</td>
<td>copyland l3 2 z43</td>
</tr>
<tr>
<td></td>
<td>add_to_land l3 0 0 51</td>
</tr>
<tr>
<td>9. Establish the abdomen/pelvis joint center and copy it to object #1.</td>
<td>copyland l4 2 z46</td>
</tr>
<tr>
<td></td>
<td>add_to_land l4 0 0 51</td>
</tr>
<tr>
<td>10. Establish the right shoulder center and copy it to object #1.</td>
<td>copyland l5 2 z74</td>
</tr>
<tr>
<td></td>
<td>add_to_land l5 38 -38 0</td>
</tr>
<tr>
<td>11. Establish the right elbow joint center and copy it to object #1.</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>split l6 z52 z51</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>copyland l6 2 l6</td>
</tr>
<tr>
<td>12. Establish the right wrist joint center and copy it to object #1.</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>split l7 z21 z54</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>copyland l7 2 l7</td>
</tr>
<tr>
<td>Steps</td>
<td>Commands</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>13. Establish the left shoulder joint center and copy it to object #1.</td>
<td>copyland l8 2 z66</td>
</tr>
<tr>
<td></td>
<td>add_to_land l8 -38 -38 0</td>
</tr>
<tr>
<td>14. Establish the left elbow joint center and copy it to object #1.</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>split l9 z57 z58</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>copyland l9 2 l9</td>
</tr>
<tr>
<td>15. Establish the left wrist joint center and copy it to object #1.</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>split l10 z23 z60</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>copyland l10 2 l10</td>
</tr>
<tr>
<td>16. Establish the right hip joint center and copy it to object #1.</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>split l11 z24 z17</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>copyland l11 2 l11</td>
</tr>
<tr>
<td>17. Establish the right knee joint center and copy it to object #1.</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>split l12 z35 z36</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>copyland l12 2 l12</td>
</tr>
<tr>
<td>18. Establish the right ankle joint center and copy it to object #1.</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>split l13 z38 z78</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>copyland l13 2 l13</td>
</tr>
<tr>
<td>19. Establish the left hip joint center and copy it to object #1.</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>split l14 z24 z18</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>copyland l14 2 l14</td>
</tr>
<tr>
<td>20. Establish the left knee joint center and copy it to object #1.</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>split l15 z27 z28</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>copyland l15 2 l15</td>
</tr>
<tr>
<td>Steps</td>
<td>Commands</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>21. Establish the left ankle joint center and copy it to object #1.</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>split l16 z30 z70</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>copyland l16 2 l16</td>
</tr>
<tr>
<td>22. Hide object #2 and show object #1 with the estimated joint</td>
<td>2</td>
</tr>
<tr>
<td>centers.</td>
<td>hide</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>wireframe</td>
</tr>
<tr>
<td></td>
<td>surface</td>
</tr>
<tr>
<td></td>
<td>transparent</td>
</tr>
</tbody>
</table>
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

!  exit  pop
$  eye  possub
@  eyedist  print
abssub  fcmod  push
add  fWRITE  pWRITE
add_to_land  fill  readout
addobj  filter  recolor
align  fillseg  refresh
alt_land  fix_seam  remark
and  fkeys  resample
auto_jaws  front  rgb
avgland  fullcolor  right
axes  gcv  rotate
back  gload  ruin
balltest  gouraud  select
black  gwrite  set
bottom  help  shad
bottom_cap  hide  show
boxes  histogram  shrink
calipers  interpolve  side
cd  jaw  skip
center  jump  sleep
centroid  land  smooth
circumference  landlist  split
clearance  left  store
cloud  lload  subject
colors  lregister  super
comment  lwrite  surf_reg
conclose  man  surface
contour  median  surface_area
contours  merge  tape
copy  mload  text
copyland  modland  thin
copyseg  move  threshold
cursor  move_vertex  top
cwrite  movie  tops
cybermovie  movie_segment  toupee
delete  mwrite  transparent
delland  nameland  trim
delpnt  negsub  volume
delseg  new_order  walls
delta  newcenter  white
derive  newland  wireframe
diff  option  wload
dilate  ortho  wWRITE
displace  pause  xload
distance  pick  xWRITE
do fill  pickmode  zload
do smooth  planes  zregister
drawline  pload  zWRITE
erode  points
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.

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.

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).

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 \((8.3, 0, 31.2\text{ mm})\) 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.


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.

auto_jaws

This command automatically performs jaw closure on a calipers object (set as the active object) around a test object (object to be measured). It iteratively moves the jaws, then tests for intersection (see the jaw and intersect commands below). It requires that the calipers be positioned for the measurement, with the jaws definitely outside of (not
intersecting) the test object. It positions the jaws until they are within "tolerance" of the test object without intersecting. If "tolerance" is not specified, it defaults to .001. The "bounding_box" value is the distance outside the edges of the bounding box for the calipers in which points are still considered for the intersection test. Limiting this test increases the speed of the operation dramatically. If "bounding_box" is not specified, it defaults to 6.0 mm.

Usage: auto_jaws <test_object> [tolerance] [bounding_box]

Example: auto_jaws 2 0.1 4.0

Result: The calipers jaws are closed to within 0.1 mm of the surface of object 2. Only the object 2 points within 4 mm of the bounding box of the calipers are considered for the intersection test.

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.

This command fills in the bottom of the head of the active object. Note that this command works best when the object is positioned so that the (estimated!) lowest point on the head is centered about the Y axis. This command requires one parameter: the lowest latitude to establish the bottom plane. Check the object coordinates in the blue box on the lower left for the latitude to use for the bottom plane. The latitude must be within the current trim area.

Usage: bottom_cap (latitude to establish the plane)
Example: bottom_cap 50
Result: INTEGRATE fills in the bottom of the head scan trimmed at lower latitude 50

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.

This command is used to turn calipers mode on and off once a calipers object is loaded. Tutorial_9 demonstrates how calipers are used in conjunction with the caliper objects and commands such as jaw, auto_jaws, and
readout.
NOTE: A calipers object is a normal object representing 2 jaws and a slide bar. It also includes some special processing which is only active when calipers mode is on.

Usage: toggle command
Example: calipers
Result: Calipers will turn on or off.

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.

clearance

clearance - This command determines the clearance distances between every point on a test object and the closest corresponding point on a reference object. It produces a dataset made up of all points closer than a specified threshold, and a text file containing a histogram showing the distribution of clearance distances for the entire test object. The test object and the reference object must be mesh objects, and the resulting object is also a mesh object.

Clearance takes the following parameters:
ref_obj - the object which is tested against the active object to determine clearances
bin_size - the resolution of the clearance histogram
thresh - a distance below which a point is considered "too close". Points that are too close will be copied to a new object for further analysis. If thresh is not specified, no points are copied.
result - the name of a file in which to store the detail and histogram information. If result is not specified, no

62
histogram is created.

Usage: clearance ref_obj bin_size [thresh [result]]
Example: clearance 1 0.5 12.5 clearance.results
Result: The active object is compared to object 1. A histogram of clearance distances is produced in file clearance.results at 0.5 mm resolution. All points closer than 12.5 mm will be copied to a new object for separate analysis.

cload, clouds*

This command reads in a scan file. Cloud 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.ind
Result: INTEGRATE reads in a scan file called
headscan and its associated landmark file, headscan.Ind.

*clouds 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.
**concose**

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.

![Diagram of conclose](image)

Figure 15: 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]
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 11 2 12 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 delpt.

Usage: delpt (Z#/L#) or delpt <lon> <lat>
Example: delpt z24 or delpt 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 16 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 16: 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)`
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.

fcmod

This command separately removes all red, green, and blue color components which fall outside of the specified boundaries.

Usage: fcmod {UI} {RGB} <value> [...]  
Example: fcmod LR32 UR128  
Result: All red components below 32 or above 128 are set to 0, removing the red component from that surface point.

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: fwrite 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 cloud 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 17: 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.

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.

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.BUY file. Gwrite takes one argument, the name of the MOVIE.BUY 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.

**intrplnd**

This command enables or disables interpolated landmark mode. Normally, landmark picking selects the point most representing the average of the points in the pick region, but with intrplnd mode enabled, the actual average (when pickmode average is selected) is set as the landmark point.

Usage: toggle command
Example: intrplnd
Result: Interpolated landmark mode is enabled. Landmarks will be positioned at the average coordinates of the points within the pick window (cursor boundaries) if PICKMODE AVERAGE is active.

**jaw**

This command only works when the active object is a calipers object. It moves either jaw 1 or jaw 2 forward or
backward along the slide bar. Positive movement is along the direction from jaw 1 to jaw 2. Negative movement is along the direction from jaw 2 to jaw 1.

Usage: jaw [1|2] <distance>
Example: jaw 2 -10
Result: If jaw 2 is more than 10 mm from jaw 1, it will be moved 10 mm closer to jaw 1 along the slide bar. If it is less than 10 mm from jaw 1, it will be moved to adjoin jaw 1.

**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.ldd r
Result: INTEGRATE writes the landmark file new_landmarks.ldd 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
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.

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.

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

89
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 132
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_center

The command changes the center point of an object to the specified coordinates or landmark.

Usage: newcenter {X Y Z | Ln | Ln}
Example: newcenter z2
Result: The new center of rotation of the active object will be at auxiliary landmark z2.

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, delptn, 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.

*readout*

This command only works with a calipers object which is the active object. It produces the measured distance between the 2 jaws of the calipers. A parameter indicates whether the measurement is for the outside of the object (readout outside) (for example, measuring head breadth), or the inside of the object (readout inside) (for example, measuring the distance between a hand and the body).

Usage: readout [insideloutside]
Example: readout outside
Result: The distance between the inner surfaces of the jaws is reported in a screen message.

*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 \leq 256 \Rightarrow \text{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.

shrink
This command is used to reduce the radial values of a head scan in the movie.byu or .ply format uniformly and spherically. An example use of shrink is to approximate the inner surface of a hollow object such as a helmet.

Usage: shrink (amount in mm)
Example: shrink 10
Result: The head scan will shrink by 10 mm uniformly and spherically.

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:
\[
\frac{(\text{weight}_1 \cdot \text{coordinates}_1) + (\text{weight}_2 \cdot \text{coordinates}_2)}{\text{weight}_1 + \text{weight}_2}
\]

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.

surf_reg

This command attempts to improve a rough registration between 2 similar objects by iteratively computing distances and angles between the 2 surfaces, then correcting for the observed errors. Often a rough alignment can be accomplished using 3 or more common landmarks and using lregister or zregister to register one object with another.

This command accepts six parameters:
test_object - object to be adjusted based on surface matching
ref_object - object to be matched against
thresh - maximum distance between matched points to limit accidental matching with points on the far side of an object. Should be less than half the minimum diameter of the smallest object, but more than the maximum distance between the two surfaces in the matching region. The default for thresh is 50.0 mm.
nTries - the number of trials between asking the operator for input. After nTries trials, the operator is given the choice of: stopping the registration (N), continuing the registration (Y), or adjusting the number of probes being used (1, 2, 4, 8, 0). The number of probes is the number specified by nProbes, divided by a speedup factor of 1, 2, 4, 8, or 16. The speedup factor makes each loop correspondingly faster, but may reduce the accuracy of the match, so the operator is given control to balance the time spent vs. the accuracy. The default for nTries is 50.
nProbes - the maximum number of probes in each trial. The actual number of probes may be less than nProbes if a speedup factor has been specified. The probes are randomly selected from the entire set of points in the test object. For each probe point, a corresponding point on the reference object is determined by finding the point where the normal to the probe point intersects the reference object. The matched points are then used in the same alignment algorithm used by lregister and zregister. The default for nProbes is 1/10 of the points in the test object.
rotateAxis - sometimes it is desirable to limit the freedom of the rotations and translations which can take place in the matching process. For instance, objects may be known to be positioned at the same height with the same vertical axis, so
limiting the displacements to horizontal translations and rotations might be expected to produce a more-exact result. By specifying the vertical axis (usually Y in INTEGRATE) as the rotateAxis, extraneous displacements can be avoided. The default is unconstrained rotations and translations.

Usage: surf_reg test_obj ref_obj [thres [nTries [nProbes [rotateAxis]]]]
Example: surf_reg 5 4 25.0 15 500
Result: Object 5 will be surface_registered with object 4. Points in object 4 that are more than 25 mm away from a test point in object 1 will be disqualified for surface point matching. Every 15 loops the command will pause for operator action. If object 5 has more than 500 points, the command will choose 500 randomly-selected points for surface matching. RotateAxis is not specified, so rotation and translation are not constrained.

**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.

This command converts a surface contour/circumference into a tape-measure-equivalent by eliminating any concave curves in the surface. The resulting distance of the modified contour should then approximate the distance measured with a tape measure.

This command has two parameters: the contour to be modified and has an optional parameter which indicates whether the tape measure completely wraps around the object. This command might not work well for very complex curves, since it uses the center of mass of the contour points as a reference point. If the center of mass falls outside the curve, results are unpredictable.

Usage: tape (contour#) [wrap/norowrap]
Example: tape 2
Result: The contour is converted into a completely convex curve. Since the wrap parameter is not specified, the end points are excluded from the removal algorithm.

This command will print a text string on the screen.

The parameters are x coordinate, y coordinate, size, and the text string.

Usage: text xcoord ycoord size string
Example: text 10 300 4 my text
Result: The text string “my text” will show up on the lower left hand corner of the screen.
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.

Object before thinning

Object after executing thin 2 2

Figure 18: Thinning an object.

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 19: 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.
white

Figure 20: With walls set to 698 702, only a cross section of the object appears.

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.

\textbf{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.

\textbf{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:

   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:

<table>
<thead>
<tr>
<th>iaudit (with no options)</th>
<th>Lists the INTEGRATE sessions by date and time.</th>
</tr>
</thead>
<tbody>
<tr>
<td>-a</td>
<td>Lists every command that was executed in the session or sessions.</td>
</tr>
<tr>
<td>-b</td>
<td>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.</td>
</tr>
<tr>
<td>-d</td>
<td>Deletes sessions from the session list.</td>
</tr>
<tr>
<td>-t</td>
<td>Creates a batch file that does not include any batch file commands, not even the @(batch file name) commands that launch batch files. It does include all other commands, whether typed in or read from a file.</td>
</tr>
</tbody>
</table>

5. To **list all the commands** in sessions 5 through 10, type a command that looks like this:
   
   \[\text{iaudit} \ -a \ 5 \ 10\]

6. To **create a batch file** from the commands in sessions 24, type a command that looks like this:
   
   \[\text{iaudit} \ -b \ 24>\text{newbatch}\]
   
   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:
   
   \[\text{iaudit} \ -d \ 10 \ 20\]

8. To **create a batch file** from the commands in session 12, **merging any secondary batch file commands**, type a command that looks like this:
   
   \[\text{iaudit} \ -t \ 12>\text{newbatch}\]
   
   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


APPENDIX A

TUTORIALS: IMAGE DATA AND SCRIPT FILES
FILES NEEDED FOR TUTORIAL_1

<table>
<thead>
<tr>
<th>SCRIPT FILE</th>
<th>IMAGE FILE(S)</th>
<th>COLOR FILE(S)</th>
<th>LANDMARK FILE(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td>tutorial_1</td>
<td>010_53p</td>
<td>010_53p.rgb</td>
<td>010_53p.lnd</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>SCRIPT FILE</th>
<th>IMAGE FILE(S)</th>
<th>LANDMARK FILE(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td>tutorial_2</td>
<td>010_53p</td>
<td>010_53p.lnd</td>
</tr>
<tr>
<td></td>
<td>010_53ph</td>
<td>010_53ph.lnd</td>
</tr>
<tr>
<td></td>
<td>53psize5</td>
<td>53psize5.lnd</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Script File</th>
<th>Image File(s)</th>
<th>Color File(s)</th>
<th>Landmark File(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>tutorial_3</td>
<td>010_53p</td>
<td>010_53p.rgb</td>
<td>010_53p.lnd</td>
</tr>
</tbody>
</table>

### Script File for Tutorial_3

load 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
<table>
<thead>
<tr>
<th>SCRIPT FILE</th>
<th>IMAGE FILE(S)</th>
<th>LANDMARK FILE(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td>tutorial_4</td>
<td>53psize5</td>
<td>53psize5.lnd</td>
</tr>
<tr>
<td></td>
<td>100_53ph</td>
<td>100_53ph.lnd</td>
</tr>
<tr>
<td></td>
<td>100_53p</td>
<td>100_53p.lnd</td>
</tr>
<tr>
<td></td>
<td>101_53ph</td>
<td>101_53ph.lnd</td>
</tr>
<tr>
<td></td>
<td>101_53p</td>
<td>101_53p.lnd</td>
</tr>
<tr>
<td></td>
<td>102_53ph</td>
<td>102_53ph.lnd</td>
</tr>
<tr>
<td></td>
<td>102_53p</td>
<td>102_53p.lnd</td>
</tr>
<tr>
<td></td>
<td>104_53ph</td>
<td>104_53ph.lnd</td>
</tr>
<tr>
<td></td>
<td>104_53p</td>
<td>104_53p.lnd</td>
</tr>
<tr>
<td></td>
<td>105_53ph</td>
<td>105_53ph.lnd</td>
</tr>
<tr>
<td></td>
<td>105_53p</td>
<td>105_53p.lnd</td>
</tr>
</tbody>
</table>

**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 11 3 134
copyland 12 3 138
cload 101_53ph
lload 101_53ph.lnd
zregister 4 1
cload 101_53p
lload 101_53p.lnd
lregister 5 4
1
copyland 13 5 134
copyland 14 5 138
cload 102_53ph
lload 102_53ph.lnd
zregister 6 1
cload 102_53p
lload 102_53p.lnd
zregister 7 6
1
copyland 15 7 134
copyland 16 7 138
cload 104_53ph
lload 104_53ph.lnd
zregister 8 1
cload 104_53p
lload 104_53p.lnd
lregister 9 8
1
copyland 17 9 134
copyland 18 9 138
cload 105_53ph
lload 105_53ph.lnd
zregister 10 1
cload 105_53p
lload 105_53p.lnd
lregister 11 10
1
copyland 19 11 134
copyland 110 11 138
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

<table>
<thead>
<tr>
<th>SCRIPT FILE</th>
<th>IMAGE FILE(S)</th>
<th>COLOR FILE(S)</th>
<th>LANDMARK FILE(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td>tutorial_5</td>
<td>face</td>
<td>face.rgb</td>
<td></td>
</tr>
<tr>
<td></td>
<td>mask</td>
<td>mask.rgb</td>
<td></td>
</tr>
</tbody>
</table>

### SCRIPT FILE FOR TUTORIAL_5

```plaintext
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

<table>
<thead>
<tr>
<th>SCRIPT FILE</th>
<th>IMAGE FILE(S)</th>
<th>LANDMARK FILE(S)</th>
<th>MATRIX FILE(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td>tutorial_6</td>
<td>head.g</td>
<td>head.lnd</td>
<td></td>
</tr>
<tr>
<td></td>
<td>pasgt.g</td>
<td>pasgt.lnd</td>
<td></td>
</tr>
<tr>
<td></td>
<td>head.pasgt.cdd</td>
<td>head.pasgt.lnd</td>
<td></td>
</tr>
</tbody>
</table>

### SCRIPT FILE FOR TUTORIAL_6

```plaintext
option wireframe off
option points on
eye 0 0 1200
gload head.g
lload head.lnd
gload pasgt.g
lload pasgt.lnd
cload head.pasgt.cdd
lload head.pasgt.lnd
hide
lregister 3 1
zregister 2 3
2
movie_seg -
land off
shrink 9.4 0 0
1
movie_seg ly110
hide 1
clearance 4.5 12.5 head.results
gwrite head.pasgt.fail.g
```
### FILES NEEDED FOR TUTORIAL_7

<table>
<thead>
<tr>
<th>SCRIPT FILE (S)</th>
<th>IMAGE FILE(S)</th>
<th>COLOR FILE(S)</th>
<th>MATRIX FILE(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td>tutorial_7</td>
<td>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</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>link1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sit1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### SCRIPT FILE FOR TUTORIAL_7

```bash
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
```

134
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

135
**FILES NEEDED FOR TUTORIAL_8**

<table>
<thead>
<tr>
<th>SCRIPT FILE</th>
<th>IMAGE FILE(S)</th>
<th>COLOR FILE(S)</th>
<th>MATRIX FILE(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td>tutorial_8</td>
<td>dr_boff.g</td>
<td></td>
<td>std.mtx</td>
</tr>
<tr>
<td>fbsetup</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**SCRIPT FILE FOR TUTORIAL_8**

```plaintext
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_9

SCRIPT FILE (S)       IMAGE FILE(S)       COLOR FILE(S)       MATRIX FILE(S)
tutorial_9            tsa_stda.ply       calipers.g.color    std2.mtx, waist_circ.mtx
c50x25.g              caliper.mtx

SCRIPT FILE FOR TUTORIAL_9

rgb on
gouraud on
option wireframe off
option points off
option surface on
pload tsa_stda.ply -1000
mload std2.mtx
gload c50x25.g
fullcolor calipers.g
calipers on
mload caliper.mtx
jaw 2 150
top
auto_jaws 1
walls 640 643
walls full
right
movie_seg uy40 ly-20
hide 1
hide 2
top
mload waist_circ.mtx
resample
hide 3
pick on
pickmode cir3p
top
surface
distance 2
tape 3
distance 2
FILES NEEDED FOR TUTORIAL_10

<table>
<thead>
<tr>
<th>SCRIPT FILE (S)</th>
<th>IMAGE FILE(S)</th>
<th>LANDMARK FILE(S)</th>
<th>MATRIX FILE(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td>tutorial_10</td>
<td>chrisla.ply</td>
<td>chrisla.ind</td>
<td>std.mtx</td>
</tr>
</tbody>
</table>

SCRIPT FILE FOR TUTORIAL_10

```
pload chrisla.ply -1000
mload std.mtx
pload chrisla.ply -1000
mload std.mtx
lload chrisla.ind
split 11 z72 z64
  1
copyland 11 2 11
copyland 12 2 z24
  add_to_land 12 0 -25 51
copyland 13 2 z43
  add_to_land 13 0 0 51
copyland 14 2 z46
  add_to_land 14 0 0 51
copyland 15 2 z74
  add_to_land 15 38 -38 0
  2
split 16 z52 z51
  1
copyland 16 2 16
  2
split 17 z21 z54
  1
copyland 17 2 17
copyland 18 2 z66
  add_to_land 18 -38 -38 0
  2
split 19 z57 z58
  1
copyland 19 2 19
  2
split 110 z23 z60
  2
split 111 z24 z17
  1
copyland 111 2 111
  2
split 112 z35 z36
  1
copyland 112 2 112
  2
```
split 113 z38 z78
 1
copyland 113 2 113
 2
split 114 z24 118
 1
copyland 114 2 114
 2
split 115 z27 z28
 1
copyland 115 2 115
 2
split 116 z30 z70
 1
copyland 116 2 116
 2
hide
 1
wireframe
surface
transparent
APPENDIX B

HEAD AND FACE 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.
*MIM = Midlateral Inframandibular
APPENDIX C

WHOLE BODY ANATOMICAL LANDMARKS:
DESCRIPTIONS AND ILLUSTRATIONS

<table>
<thead>
<tr>
<th>#</th>
<th>Segment</th>
<th>Landmarks</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Front</td>
<td>1) Head</td>
<td>Sellion</td>
<td>scanned</td>
</tr>
<tr>
<td></td>
<td>2) Infraorbitale (L)</td>
<td>scanned</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3) Tragion (L)</td>
<td>scanned</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4) Gonion (L)</td>
<td>scanned</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5) Supramenton</td>
<td>scanned</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6) Infraorbitale (R)</td>
<td>scanned</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7) Gonion (R)</td>
<td>scanned</td>
<td></td>
</tr>
<tr>
<td></td>
<td>8) Tragion (R)</td>
<td>scanned</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9) Thorax Suprasternale</td>
<td>scanned</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10) Clavicale (L)</td>
<td>scanned</td>
<td></td>
</tr>
<tr>
<td></td>
<td>11) Clavicale (R)</td>
<td>scanned</td>
<td></td>
</tr>
<tr>
<td></td>
<td>12) Acromion (R)</td>
<td>scanned</td>
<td></td>
</tr>
<tr>
<td></td>
<td>13) Acromion (L)</td>
<td>scanned</td>
<td></td>
</tr>
<tr>
<td></td>
<td>14) Anterior Axilla Ref. Point (L)</td>
<td>scanned</td>
<td></td>
</tr>
<tr>
<td></td>
<td>15) Thelion/Bustpoint (L)</td>
<td>scanned</td>
<td></td>
</tr>
<tr>
<td></td>
<td>16) Substernale</td>
<td>scanned</td>
<td></td>
</tr>
<tr>
<td></td>
<td>17) Thelion/Bustpoint (R)</td>
<td>scanned</td>
<td></td>
</tr>
<tr>
<td></td>
<td>18) Anterior Axilla Ref. Point (R)</td>
<td>scanned</td>
<td></td>
</tr>
<tr>
<td></td>
<td>19) Tenth Rib (R)</td>
<td>scanned</td>
<td></td>
</tr>
<tr>
<td></td>
<td>20) Tenth Rib (L)</td>
<td>scanned</td>
<td></td>
</tr>
<tr>
<td></td>
<td>21) Pelvis Illocristale (R)</td>
<td>scanned</td>
<td></td>
</tr>
<tr>
<td></td>
<td>22) Illocristale (L)</td>
<td>scanned</td>
<td></td>
</tr>
<tr>
<td></td>
<td>23) ASIS (R)</td>
<td>scanned</td>
<td></td>
</tr>
<tr>
<td></td>
<td>24) ASIS (L)</td>
<td>scanned</td>
<td></td>
</tr>
<tr>
<td></td>
<td>25) Trochanterion (R)</td>
<td>scanned</td>
<td></td>
</tr>
<tr>
<td></td>
<td>26) Trochanterion (L)</td>
<td>scanned</td>
<td></td>
</tr>
<tr>
<td></td>
<td>27) Crotch</td>
<td>calculated</td>
<td></td>
</tr>
<tr>
<td></td>
<td>28) Wrist Radial Styloid (R)</td>
<td>scanned</td>
<td></td>
</tr>
<tr>
<td></td>
<td>29) Hand Metacarpal-Phalangeal II</td>
<td>scanned</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30) Dactyliion (R)</td>
<td>scanned</td>
<td></td>
</tr>
<tr>
<td></td>
<td>31) Wrist Radial Styloid (L)</td>
<td>scanned</td>
<td></td>
</tr>
<tr>
<td></td>
<td>32) Hand Metacarpal-Phalangeal II (L)</td>
<td>scanned</td>
<td></td>
</tr>
<tr>
<td></td>
<td>33) Dactyliion (L)</td>
<td>scanned</td>
<td></td>
</tr>
<tr>
<td></td>
<td>34) Thigh Lateral Femoral Epicondyle (L)</td>
<td>scanned</td>
<td></td>
</tr>
<tr>
<td></td>
<td>35) Medial Femoral Epicondyle (L)</td>
<td>scanned</td>
<td></td>
</tr>
<tr>
<td></td>
<td>36) Medial Femoral Epicondyle (R)</td>
<td>scanned</td>
<td></td>
</tr>
<tr>
<td></td>
<td>37) Lateral Femoral Epicondyle (R)</td>
<td>scanned</td>
<td></td>
</tr>
<tr>
<td></td>
<td>38) Ankle Medial Malleolus (R)</td>
<td>scanned</td>
<td></td>
</tr>
<tr>
<td></td>
<td>39) Sphyrion (R)</td>
<td>scanned</td>
<td></td>
</tr>
</tbody>
</table>
40) Foot Metatarsal-Phalangeal I (R) scanned
41) Phalange II (Foot) (R) scanned
42) Metatarsal-Phalangeal V (R) scanned
43) Ankle Medial Malleolus (L) scanned
44) Sphyrrion (L) scanned
45) Foot Metatarsal-Phalangeal I (L) scanned
46) Phalange II (Foot) (L) scanned
47) Metatarsal-Phalangeal V (L) scanned
48) Back Nuchale scanned
49) Cervicale (Spine I) scanned
50) Thorax Posterior Axilla Ref. Point (L) scanned
51) Posterior Axilla Ref. Point (R) scanned
52) Tenth Rib Midspine (Spine II) scanned
53) Pelvis PSIS (L) scanned
54) PSI Midspine (R) calculated
55) Preferred Waist Posterior scanned
56) Arm Lateral Humeral Epicondyle (L) scanned
57) Radiale (L) scanned
58) Olecranon (L) scanned
59) Medial Humeral Epicondyle (L) scanned
60) Medial Humeral Epicondyle (R) scanned
61) Olecranon (R) scanned
62) Lateral Humeral Epicondyle (R) scanned
63) Radiale (R) scanned
64) Wrist Ulnar Styloid (R) scanned
65) Metacarpal-Phalangeal V (R) scanned
66) Hand Ulnar Styloid (L) scanned
67) Metacarpal-Phalangeal V (L) scanned
68) Hand Knee Crease (L) scanned
69) Knee Crease (R) scanned
70) Ankle Lateral Malleolus (R) scanned
71) Foot Posterior Calcaneous (R) scanned
72) Posterior Calcaneous (R) scanned
73) Posterior Calcaneous (L) scanned
74) Ankle Lateral Malleolus (L) scanned

-Note these landmarks are subject to change.
APPENDIX D

LANDMARK FILES: ANATOMICAL AND AUXILIARY LANDMARKS
FOR THE HEAD AND FACE
<table>
<thead>
<tr>
<th>Subject ID</th>
<th>010.53p</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scan Type</td>
<td>NO TYPE</td>
</tr>
<tr>
<td>Study Name</td>
<td>*** NO STUDY ***</td>
</tr>
<tr>
<td>Land Study</td>
<td>New Study</td>
</tr>
<tr>
<td>Std Land</td>
<td>42</td>
</tr>
<tr>
<td>Aux Land</td>
<td>0</td>
</tr>
<tr>
<td>Standard</td>
<td></td>
</tr>
<tr>
<td>1 371 116</td>
<td>72.66   -11.54 181.31 -71.74 Right Tragion</td>
</tr>
<tr>
<td>2 343 124</td>
<td>74.03   -35.69 193.81 -64.86 Right Zygion</td>
</tr>
<tr>
<td>3 344 118</td>
<td>72.86   -34.34 184.43 -64.26 Right Infra Zygion</td>
</tr>
<tr>
<td>4 346 78</td>
<td>55.40   -24.91 121.91 -49.49 Right Gonion</td>
</tr>
<tr>
<td>5 283 67</td>
<td>68.70   -64.97 104.72 -22.35 Right Mid Infra Mandibular</td>
</tr>
<tr>
<td>6 299 153</td>
<td>83.38   -72.03 239.14 -41.99 Right Frontotemporale</td>
</tr>
<tr>
<td>7 297 146</td>
<td>86.20   -75.52 228.20 -41.57 Right Zygofrontale</td>
</tr>
<tr>
<td>8 280 110</td>
<td>90.75   -86.84 171.93 -26.35 Right Infra Malar</td>
</tr>
<tr>
<td>9 288 137</td>
<td>80.96   -74.80 214.13 -30.99 Right Ectocanthus</td>
</tr>
<tr>
<td>10 259 138</td>
<td>75.82   -75.77 215.69 -2.79 Right Endocanthus</td>
</tr>
<tr>
<td>11 281 129</td>
<td>84.46   -80.52 201.63 -25.51 Right Infraorbitale</td>
</tr>
<tr>
<td>12 237 161</td>
<td>89.27   -86.86 251.64 -20.62 Glabella</td>
</tr>
<tr>
<td>13 237 143</td>
<td>93.29   -90.76 223.51 -21.55 Sellion</td>
</tr>
<tr>
<td>14 233 122</td>
<td>117.98  -113.31 190.69 32.85 Pronsale</td>
</tr>
<tr>
<td>15 234 111</td>
<td>102.96  -99.23 173.49 27.46 Subnasale</td>
</tr>
<tr>
<td>16 234 75</td>
<td>106.66  -102.80 117.22 28.44 Promenton</td>
</tr>
<tr>
<td>17 231 65</td>
<td>92.67   -88.35 101.60 27.98 Menton</td>
</tr>
<tr>
<td>18 257 93</td>
<td>96.21   -96.20 145.36 -1.18 Right Cheilion</td>
</tr>
<tr>
<td>19 234 96</td>
<td>106.02  -102.18 150.05 28.27 Stomion</td>
</tr>
<tr>
<td>20 216 94</td>
<td>99.29   -87.57 146.92 46.80 Left Cheilion</td>
</tr>
<tr>
<td>21 224 53</td>
<td>58.81   -56.68 82.84 15.68 Submandibular</td>
</tr>
<tr>
<td>22 196 130</td>
<td>84.98   -62.97 203.19 57.07 Left Infraorbitale</td>
</tr>
<tr>
<td>23 215 138</td>
<td>75.36   -66.02 215.69 36.34 Left Endocanthus</td>
</tr>
<tr>
<td>24 186 137</td>
<td>80.62   -52.66 214.13 61.04 Left Ectocanthus</td>
</tr>
<tr>
<td>25 182 70</td>
<td>68.10   -41.90 109.41 53.69 Left Mid Infra Mandibular</td>
</tr>
<tr>
<td>26 176 154</td>
<td>84.04   -46.69 240.70 69.88 Left Frontotemporale</td>
</tr>
<tr>
<td>27 175 148</td>
<td>85.42   -46.58 231.32 71.60 Left Zygofrontale</td>
</tr>
<tr>
<td>28 189 110</td>
<td>91.67   -62.39 171.93 67.16 Left Infra Malar</td>
</tr>
<tr>
<td>29 132 123</td>
<td>77.00   -3.78 192.25 76.91 Left Zygion</td>
</tr>
<tr>
<td>30 133 117</td>
<td>76.22   -4.68 182.87 76.08 Left Infra Zygion</td>
</tr>
<tr>
<td>31 128 81</td>
<td>61.23   0.00 126.60 61.23 Left Gonion</td>
</tr>
<tr>
<td>32 108 118</td>
<td>74.80   18.17 184.43 72.56 Left Tragion</td>
</tr>
<tr>
<td>33 495 93</td>
<td>85.38   83.53 145.36 -17.68 Nuchale</td>
</tr>
<tr>
<td>34 271 139</td>
<td>80.98   -79.62 217.26 -14.83 Right Pupil</td>
</tr>
<tr>
<td>35 286 148</td>
<td>88.59   -82.65 231.32 -31.89 Right SupraEctocanthus</td>
</tr>
<tr>
<td>36 269 153</td>
<td>86.10   -85.01 239.14 -13.68 Right SupraPupil</td>
</tr>
<tr>
<td>37 256 150</td>
<td>87.00   -87.00 234.45 0.00 Right SupraEndocanthus</td>
</tr>
<tr>
<td>38 203 140</td>
<td>84.67   -67.39 218.82 51.27 Left Pupil</td>
</tr>
<tr>
<td>39 192 150</td>
<td>88.79   -62.79 234.45 62.78 Left SupraEctocanthus</td>
</tr>
<tr>
<td>40 205 153</td>
<td>89.22   -72.31 239.14 52.26 Left SupraPupil</td>
</tr>
<tr>
<td>41 219 150</td>
<td>87.33   -78.48 234.45 38.30 Left SupraEndocanthus</td>
</tr>
<tr>
<td>42 234 87</td>
<td>104.15  -100.38 135.98 27.78 SupraMenton</td>
</tr>
<tr>
<td>Aux</td>
<td></td>
</tr>
<tr>
<td>End</td>
<td></td>
</tr>
</tbody>
</table>
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
10 -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 Sillon
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 =

153
APPENDIX E

LANDMARK FILES: ANATOMICAL LANDMARKS
FOR THE WHOLE BODY
SUBJECT_ID = vstemplate.ply
SCAN_TYPE = NO TYPE
STUDY_NAME = *** NO STUDY ***
LAND_STUDY = New Study
STD_LAND = 42
AUX_LAND = 80
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 -999 -999 0.00 0.00 0.00 0.00
7 -999 -999 0.00 0.00 0.00 0.00
8 -999 -999 0.00 0.00 0.00 0.00
9 -999 -999 0.00 0.00 0.00 0.00
10 -999 -999 0.00 0.00 0.00 0.00
11 -999 -999 0.00 0.00 0.00 0.00
12 -999 -999 0.00 0.00 0.00 0.00
13 -999 -999 0.00 0.00 0.00 0.00
14 -999 -999 0.00 0.00 0.00 0.00
15 -999 -999 0.00 0.00 0.00 0.00
16 -999 -999 0.00 0.00 0.00 0.00
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 -999 -999 0.00 0.00 0.00 0.00
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 -999 -999 0.00 0.00 0.00 0.00
27 -999 -999 0.00 0.00 0.00 0.00
28 -999 -999 0.00 0.00 0.00 0.00
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 9 1120 760.65 -83.94 -72.12 756.00 => Glabella
2 9 223 742.99 -85.97 -73.61 738.00 Sellion
3 8 1190 765.58 -92.41 -78.25 646.00 Supramenton
4 8 1532 720.05 -93.12 -44.83 714.00 Rt Infraorbit.
5 8 1534 716.11 -54.99 -93.33 714.00 Lt Infraorbit.
6 8 8413 552.65 -26.85 -28.16 552.00 => Neck Base
7 8 5221 506.61 -24.87 -25.82 506.00 Suprasternale
8 8 5742 313.90 -44.11 -4.06 512.00 Rt Clavicale
9 8 5755 512.16 -12.85 -55.16 512.00 Lt Clavicale
10 7 187 313.17 -75.23 -62.18 304.00 => Substernale
11 7 6193 375.15 -118.33 23.95 356.00 Rt Thelion
12 7 6229 356.04 -5.53 -138.36 356.00 Lt Thelion
13 6 4307 230.59 -118.18 35.77 198.00 Rt Tenth Rib
14 6 3182 189.26 -21.82 -126.45 188.00 Lt Tenth Rib
15 5 9603 143.82 -99.34 103.25 104.00 => Rt Illocrist.
16 5 10711 125.86 53.34 -121.67 114.00 Lt Illocrist.
17 5 1623 148.37 -143.93 32.80 36.00 Rt ASIS
18 5 1668 39.65 -16.60 -142.01 36.00 Lt ASIS

155
APPENDIX F

COMMANDS: FUNCTIONALITY FOR HEAD AND WHOLE BODY IMAGE DATA
<table>
<thead>
<tr>
<th>COMMAND</th>
<th>Works With Cyberware?</th>
<th>Works With .g/.ply/.obj?</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>$</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>@</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>abssub</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>add</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>add_to_land</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>addobj</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>align</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>alt_land</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>and</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>avgland</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>axes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>back</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>balltest</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>black</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>bottom</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>bottom_cap</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>boxes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>cd</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>center</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>centroid</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>circumference</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>clearance</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>cload</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>colors</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>comment</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>conclose</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>contour</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>contours</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>copy</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>copyland</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>copyseg</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>cursor</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>cwrite</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>cybervmovie</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>delete</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>delland</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>delpnt</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>delseg</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>delta</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>derive</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>diff</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>dilate</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>displace</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>distance</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>do</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>drawline</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>erode</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>exit</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>eye</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>eyedist</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>fwrite</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>fill</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>filter</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>filtseg</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>fix_seam</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>fkeys</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>front</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Command</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>--------------</td>
<td>-----</td>
<td>----</td>
</tr>
<tr>
<td>fullcolor</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>gcv</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>gload</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>gouraud</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>gwrite</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>help</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>hide</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>histogram</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>interpolate</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>intrplnd</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>jump</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>land</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>landlist</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>left</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>lload</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>lregister</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>lwrite</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>man</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>median</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>merge</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>mload</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>modland</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>move</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>move_vertex</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>movie_segment</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>mwrite</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>nameland</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>negsub</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>new_order</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>newland</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>option</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ortho</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>pause</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>pick</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>pickmode</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>planes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>pload</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>points</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>pop</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>possub</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>print</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>push</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>pwrite</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>recolor</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>refresh</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>remark</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>resample</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>rgb</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>right</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>rotate</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ruin</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>select</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>set</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>shade</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>shrink</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>show</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>side</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>skip</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>sleep</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>smooth</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>split</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>store</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>subject</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>super</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>surface</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>surface_area</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>surf_reg</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>tape</td>
<td>Yes</td>
<td>Yes*</td>
</tr>
<tr>
<td>thin</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>thres</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>top</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>tops</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>toupee</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>transparent</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>trim</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>volume</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>walls</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>white</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>wireframe</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>wload</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>wwrite</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>xload</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>xwrite</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>zload</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>zregister</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>zwrite</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

* contours can only be created on Cyberware format, but zload can be used on movie.byu format after creating them on Cyberware objects.
APPENDIX G

FILE FORMATS: DESCRIPTION OF IMAGE DATA FILES SUPPORTED
BY INTEGRATE VERSION 1.28
I. Cyberware Scanner Format (new, all except WB-series scanners)

ASCII header with lines of the form <keyword>=<value>
Header terminates with DATA=
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 Vertices (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

162
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.

IV. 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
   <x y z> [coordinates for 1st point in contour]
   ...
   <x y z> [coordinates for last point in contour]

VI. Old CARD Lab Landmark Files Format

5 1 1
<xlon lat radius y z> for standard landmark 1
...
<NO> 1
<xlon lat radius y z> for standard landmark ‘N’
-1 -1 -1 [end of file tag]

VII. New CARD Lab Landmark File Format

SUBJECT_ID = <subject identifier>
SCAN_TYPE = <type identifier>
STUDY_NAME = <acquisition study identifier>
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 =

--- 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]
APPENDIX H

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 specified 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 {mlf}<subj#>.<segment#>.c which contains cylindrical coordinates for each point in the original dataset, and {mlf}<subj#>.<segment#>.cs, which contains a centerpoint for each slice in the cylindrical coordinate file. Example: f2.1c is radius information for the head segment of female subject 2, which 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 {mlf}<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 System, 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

166
After Stereo_Slice has been used to extract and convert the data for a Stereo 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 ((mlf)<subj#>.<segment#>), but also the centers file ((mlf)<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 way, with all segments responding appropriately to movement of other segments. There are several INTEGRATE script files which simplify the job 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 I

DEFINITIONS FOR DETERMINATION OF JOINT CENTERS
DEFINITIONS FOR DETERMINATION OF JOINT CENTERS

Ankles, right and left: midpoint between Lateral Malleolus and Sphyrion

Knees, right and left: midpoint between Lateral Femoral Condyle and Medial Femoral Condyle

Hips, right and left: 1.) midpoint between Anterior Superior Iliac Spine (ASIS) and Symphysion  
                           2.) translate to x coordinate (fore-aft) of the Trochanterion  
                           3.) translate 15 mm down in the z direction (vertical)

Pelvic Joint:  1.) use Posterior Superior Iliac Midspine coordinates  
                  2.) translate 51 mm in the x direction (forward)

Abdomen Joint: 1.) use 10th Rib Midspine coordinates  
                     2.) translate 51 mm in the x direction (forward)

Thorax Joint:  1.) use Cervicale coordinates  
                        2.) translate 51 mm in the x direction (forward)  
                        3.) translate 25 mm down in the z direction (vertical)

Head/Neck Joint: midpoint between right and left Tragions

Shoulder, right and left: 1.) use Acromiale coordinates  
                                 2.) translate 38 mm toward the body (y direction)  
                                 3.) translate 38 mm down in the z direction (vertical)

Elbow, right and left: midpoint between Medial and Lateral Human Epicondyles

Wrist, right and left: midpoint between Radial and Ulnar Styloid Processes