

This is to certify that the

thesis entitled

3-DIMENSIONAL MODEL OF THE KINEMATICS
OF THE HUMAN SPINE

presented by

Tsang Yeou Chen

has been accepted towards fulfillment
of the requirements for

Masters degree in Elect. Engr.

A handwritten signature in black ink that reads "Michael Shanblatt". The signature is written in a cursive, flowing style.

Major professor

Date Feb. 27, 1985

Michael Shanblatt



3 1293 01066 4781

L



RETURNING MATERIALS:

Place in book drop to
remove this checkout from
your record. FINES will
be charged if book is
returned after the date
stamped below.

AUG 06 1984

--	--	--

3-DIMENSIONAL MODEL OF THE KINEMATICS
OF THE HUMAN SPINE

By

Tsang Yeou Chen

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Department of Electrical Engineering and System Science

1985

ACKNOWLEDGEMENTS

I wish to express my sincere gratitude to Dr. Richard Hallgren for his guidance and assistance throughout the course of my studies. This thesis could not have been finished without his support and encouragement.

I am also grateful to Dr. Michael Shanblatt for his valuable guidance and painstaking review of this thesis.

I would also like to thank my parents whose support and love have encouraged me to make this thesis possible.

TABLE OF CONTENTS

Chapters	Page
I. INTRODUCTION	1
II. BACKGROUND AND MOTIVATION	3
2.1 Literature Review	3
2.2 Impetus	5
2.3 Statement of Problem	7
III. SYSTEM DESCRIPTION	8
3.1 System Organization	8
3.2 Collection of Vertebra Morphological Data . .	10
3.3 Three-Dimensional Resconstruction of Morphological Data	12
3.4 Tektronix 4112 Graphic Terminal Description .	12
3.5 Description of 4112 Terminal Driver Routines	19
3.6 Apple Controller	26
IV. MATHEMATICS OF KINEMATICS	30
3.1 Coordinate Systems	30
3.2 Static Kinematics	31
3.3 Dynamic Kinematics	42
V. CONCLUSIONS	48

APPENDICES

APPENDIX A	50
APPENDIX B	55
APPENDIX C	57
APPENDIX D	59
LIST OF REFERENCES	61

LIST OF FIGURES

Figure	Page
1	System For Generating 3-Dimensional Model of Human Spine 9
2	CAT scan Radiographs of L5 11
3	Topological Map of L5 13
4	3-Dimensional Model of L3 14
5	3-Dimensional Model of L4 15
6	3-Dimensional Model of L5 16
7	Block Diagram of Connections Between PRIME Computer, Apple Computer, Tektronix 4112 Terminal and Videotape Recorder For Automating Videotape Production 29
8	Appearance of Balls on Terminal Screen 33
9	3-Dimensional Model of L3, L4 and L5 41
10	3-Dimensional Model of L3, L4 and L5 45
11	3-Dimensional Model of L3, L4 and L5 46
12	3-Dimensional Model of L3, L4 and L5 47

ABSTRACT

3-DIMENSIONAL MODEL OF THE KINEMATICS
OF THE HUMAN SPINE

By
Tsang Yeou Chen

This thesis presents a method for constructing a high resolution, 3-dimensional model of human spine using medical and computer techniques. Medical techniques were employed first to obtain a set of cross-sectional images of vertebra and another set of data describing the motion of human spine. By processing the image data a set of contours corresponding to the outlines of vertebra slices were obtained. The construction of a 3-dimensional model of human spine was achieved by processing the contour data followed by a series of mathematical operations and axis transformations. The constructed model can be displayed on graphic display devices. By referencing the data describing the spinal motion and then manipulating the constructed model, the motion of human spine can be simulated.

CHAPTER I

INTRODUCTION

The human spine, which consists of 24 mobile bones called vertebra, is a very complex structure. Seven of these 24 vertebrae are in the cervical, or neck region; twelve in the thoracic, or chest region; and the other five in the lumbar, or lower back region. These vertebrae are symbolically designated from top to bottom as C1 - C7, T1 - T12, and L1 - L5. The spinal impairment is always concerned with many common diseases to adults such as neck pains and back trouble.

A knowledge of the detailed structure of the human spine is necessary to the study of both normal function and pathological conditions. Because of its structural complexity a simplified model is often used and subsequent simulations are performed on the model to facilitate the study of the human spine. Consequently a definitive biomechanical model of the human spine would serve as an invaluable research and instructional aid and would improve our understanding of the complex motion of the spine. Many models have been developed, most of them idealizing the vertebral column as a three-dimensional collection of rigid bodies (vertebrae) interconnected by deformable elements (intervertebral discs

and ligaments). These models inevitably contain a lot of mathematical equations that are not only hard to be analyzed but sometimes difficult to be understood by individuals not schooled in mathematics.

Computer graphic presentations have the capability of conveying information that can be quickly and easily comprehended and retained by the human mind. While the motion of the human spine though can be represented as a collection of mathematical equations, these data can be better appreciated and utilized by the medical profession when they are presented as a 3-dimensional object moving through space.

The goal of this thesis was to construct a high resolution, 3-dimensional model for three lumbar vertebrae (L3,L4,L5) using medical and computer techniques. After the individual models of each vertebra were constructed a series of mathematical manipulations were performed on them and finally a 3-dimensional model of these three vertebrae was constructed. This model depicts the real structure of the lumbar spine as they are in human body. Through the use of a set of general purpose computer graphics programs developed at Brigham Young University (MOVIE.BYU) this model can be displayed on graphic devices and graphical simulation can be accomplished.

CHAPTER II

BACKGROUND AND MOTIVATION

2.1 Literature Review

Spinal kinematics concerns the kinds and amount of motion the human spine undergoes during its normal physiologic movements. Because of the clinical value in the management of spinal impairment, the kinematic analysis and simulation study of intervertebral motion of human spine are of great interest for research engineers and physicians. The lumbar region in a human spine is a critical zone which bears high load. The lumbar zone has been the topic of intensive research by many investigators.

Mathematical analogs are often used to study biological systems. Complex systems which are difficult to investigate using conventional experimental techniques can be modeled, and their behavior may then be studied in great detail by conducting experiments with the model. Due to the complexity of the structure and motion of the human spine the study of spinal kinematics will be cumbersome without a definitive biomechanical model of the human spine.

White and Panjabi [1,2] analyzed the motion of the human spine using mathematical approach. They treated the spine as

a system of rigid bodies (vertebrae) connected in series by deformable elements (intervertebral discs) and used mathematical expressions to describe the following three spinal kinematic aspects:

1. The relative motion and translation of each vertebra with respect to each increment of motion.
2. Three-dimensional motion of spine.
3. Comparision of general relative motion of vertebrae of different sizes and configurations.

Schultz and galante [3] also treated the spine as a system similar to that used by White and Panjabi and constructed a mathematical model of the spine. Their model comprised a collection of equations expressing the mechanical relationship that must be obeyed among its elements, each of which represented a segment of a real spine. Some degree of descriptive reality has been inevitably sacrificed to gain simplicity in approximating the behavior of the spine.

Soni et al. [4] designed a spine-apparatus consisting of spine fixture and linkage transducer and presented a methodology to investigate kinematic behavior of the lumbar motion segments subject to a combined load of shear and bending. The spinal kinematics was described by large amount of numerical data. Patwardhan et al. [5] proposed a

mathematical motion simulation model to examine the motion characteristic of a human spine segment. The equivalent motion characteristic of the intervertebral joint was described by a spherical joint which was located for an intervertebral joint.

2.2 Impetus

A comprehensive description of spinal kinematics has to at least include the following five aspects:

1. Range of motion.
2. Coupling characteristic of vertebrae.
3. Pattern of motion.
4. Instantaneous axes of rotation.
5. Helical axis motion.

Using mathematical approach to describe the spinal kinematics, five different sets of mathematical expressions have to be included to represent the five individual aspects separately. The apinal kinematics though cae be analyzed mathematically the motion of the spine still incomprehensible. Five individual sets of mathematical

expressions by no means can describe the correlation between five different aspects of spinal kinematics. The spinal kinematics can be described mathematically but it can be easily comprehended when it is shown as a three-dimensional solid object moving through space.

Due to the availability of general purpose computer programs, the graphic ability no longer be patent to computer aided design systems. A general purpose computer can be given the graphic ability by including a graphic package to its system software and hooking up a graphic terminal to it.

Three-dimensional graphic representation of human motion has been performed by many investigators [6,7,8]. The conventional approach is to use a simplified representation of the object to be modeled.

The invention of computer-assisted tomography facility has made the derivation of the data describing the cross sectional images of vertebra an easy process. Consecutive data sets of cross sectional image slices implies real morphology of vertebra can be processed by computer and a three dimensional solid object representing the vertebra can be shown on graphic terminal and be used to model the vertebrae. The model can be displayed as detail as possible. The three-dimensional motion of human spine can be inspected by performing computer simulation with the model.

2.3 Statement of Problem

To construct a three-dimensional model of human spine using graphic approach the following procedures have to be gone through:

1. Collection of morphological data of vertebrae.
2. Collection of vertebral motion data.
3. Development of a interface program between a general purpose computer graphic program and a specific graphic terminal.
4. Three-dimensional reconstruction of morphological data.
5. Graphic simulation of vertebral motion.

CHAPTER III

SYSTEM DESCRIPTION

3.1 System Organization

A block diagram of the overall system and procedures to generate a 3-dimensional model of L3,L4 and L5 is shown in Figure 1. Excised and cleaned human vertebrae were processed through a computer-assisted tomography facility (CAT scan) to obtain a collection of data sets describing the cross sectional image of vertebrae. These data were passed to the PRIME 750 computer and the vertebrae morphological data were generated. Anatomical pointmark data obtained from the System Anthropometry Laboratory (SAL) describing the spatial relation of vertebrae were also passed to the PRIME 750. A series of correlation and mathematical manipulations were performed by the PRIME and finally the morphological data of three vertebrae were transformed into a coordinate system that represents the real spatial relation of vertebrae as they were in human body. The morphological data of each vertebra in the same coordinate system could then be merged together and used as input to a general purpose graphics program. This program would generate the 3-dimensional model of vertebra, providing animation if desired, and display on the Tektronix

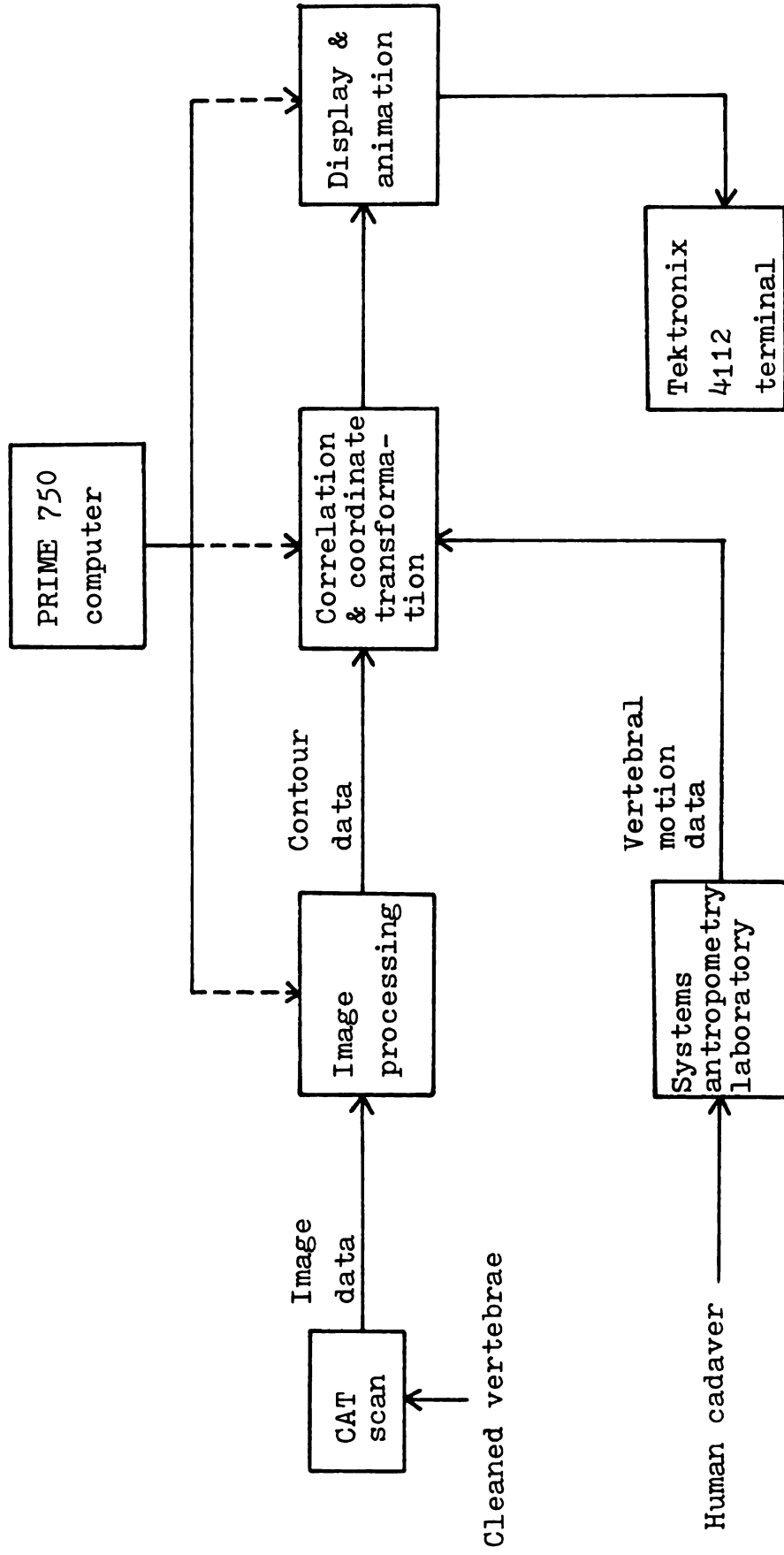


Figure 1. System for generating 3-dimensional model of human spine .

4112 graphic terminal.

3.2 Collection of Vertebra Morphological Data

Excised and cleaned vertebrae were individually mounted on a block of foam, placed on the traveling bed of the CAT scan, and incremented through the machine while collecting cross sectional density data. Each data set was made up of a 512x512 matrix that describes the cross sectional density of a 2 mm slice through the vertebra. Approximate 30 such slices were obtained for each vertebra. Four slices, representative of the type of radiographic data that was obtained from the CAT scan, are included in Figure 2. The complete density data set for each vertebra was stored on magnetic tape and then transferred to a PRIME 750 computer. By testing the density data and detecting the large density change that occurs at air/bone interface and then following this interface around the perimeter of the vertebra, a set of X and Y coordinates that describe the outline of each vertebra can be determined.

Since the contours obtained from the individual vertebra are very irregular, they will occupy large quantities of memory if we want to describe the contours precisely. In order to save memory needed to accomodate the contour data, the contour line is approximated by straight line segments.

In performing the CAT scan, the axis of movement through the machine is perpendicular to successive cross sections of each vertebra and defines the Z axis along which the contours

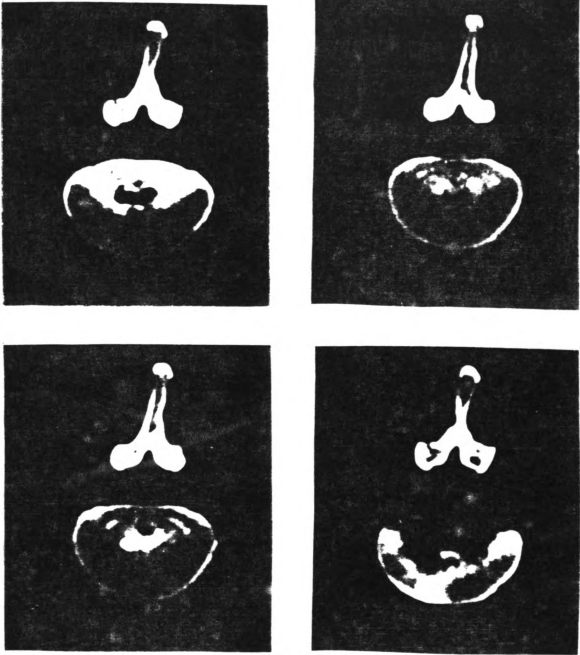


Figure 2. CAT scan radiographs of L5.

are stacked. The location of each contour on the Z axis was determined by recording the position of the traveling bed when a scan was completed. A partial set of morphological contours are shown in Figure 3.

3.3 Three-Dimensional Reconstruction of Morphological Data

Three-dimensional representation of vertebral data is accomplished through the use of a set of general purpose computer graphics programs (MOVIE.BYU). These programs are unique, allowing data describing spatial attributes of the vertebrae to be maintained in a highly structured list form. The MOVIE.BYU software provides a program (MOSAIC) that takes the contour data as input and generates panels between adjacent contours. These panels constitute a planar approximation of the surface of the vertebra. Once the data set describing the vertebral surface has been obtained, this data can then be read into another MOVIE.BYU program (DISPLAY) and the vertebra can be displayed as a three-dimensional object on the Tektronix 4112 graphic terminal. The 3-dimensional models of L3, L4 and L5 are shown in Figures 4, 5 and 6 respectively.

3.4 Tektronix 4112 Graphic Terminal Description [9,10]

The Tektronix 4112 terminal is a programmable computer display terminal. It receives commands from a host computer

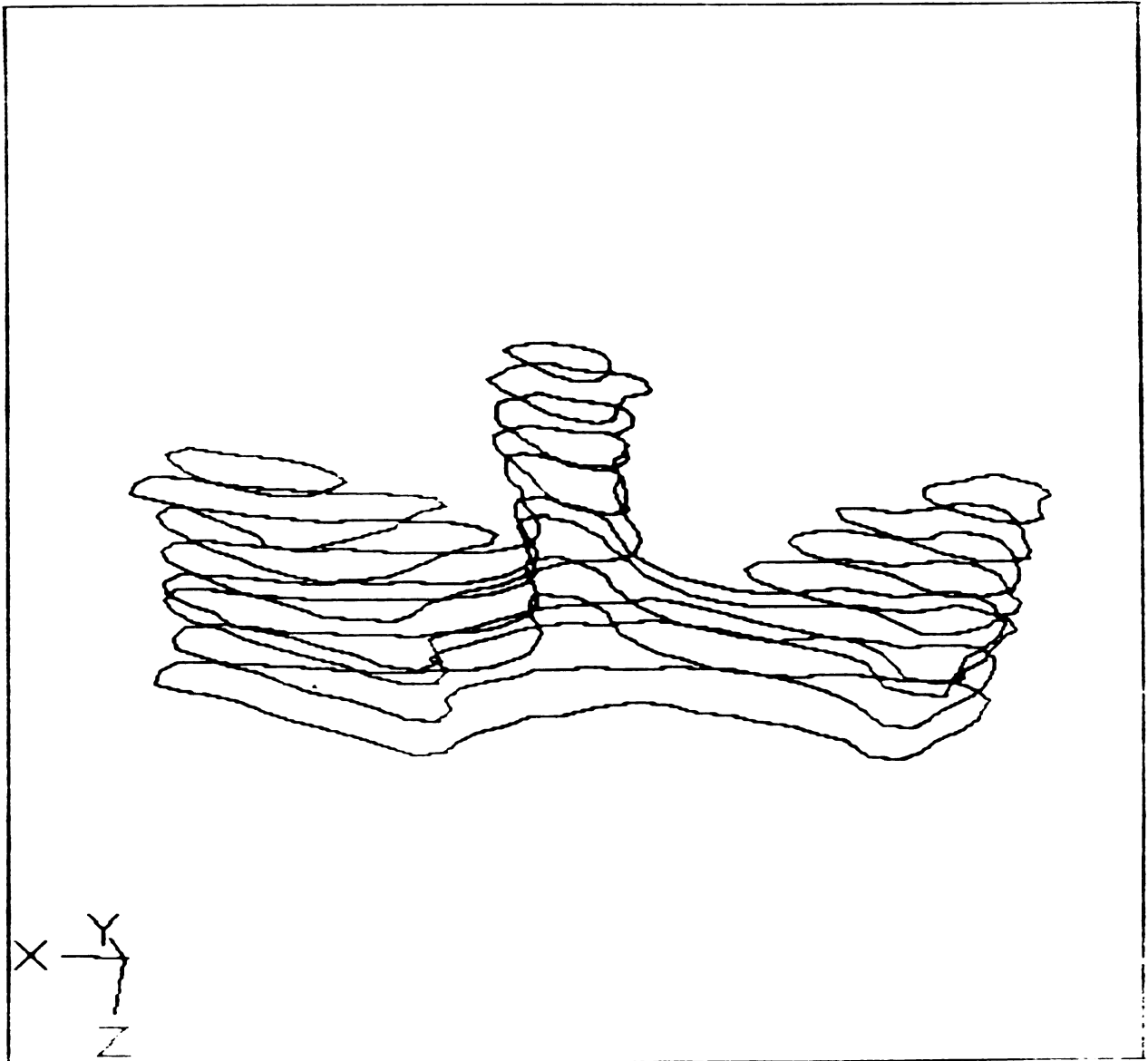


Figure 3. Topographical map of L5.

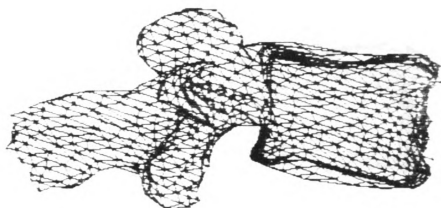


Figure 4. Three-dimensional model of L3.

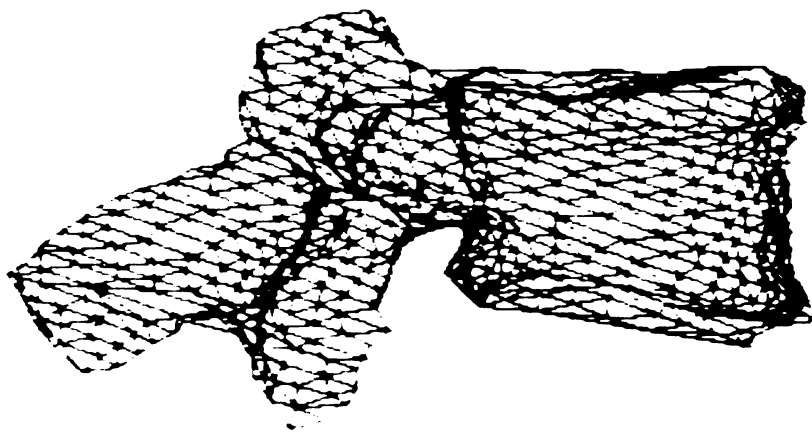


Figure 5. Three-dimensional model of L4.

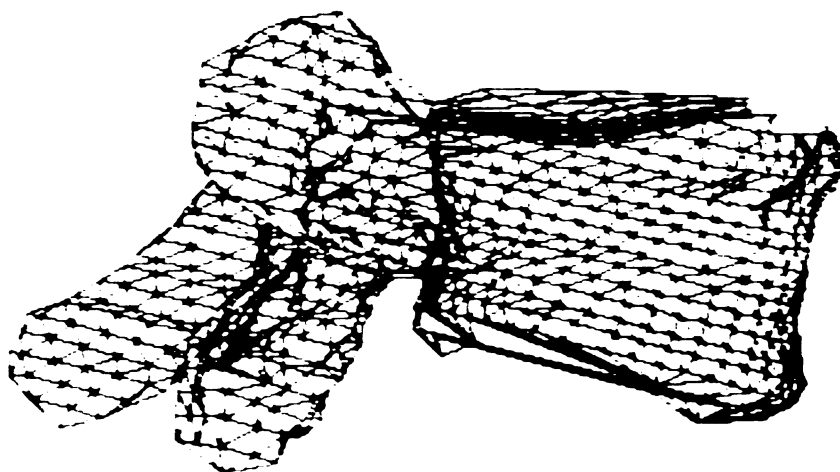


Figure 6. Three-dimensional model of L5.

and performs different operations according to the commands. This terminal's commands appears as a sequence of individual ASCII characters having a specific syntax. In order to use the terminal properly, a package of host computer subroutines had to be written to communicate with the terminal.

3.4.1 Command Syntax

Before describing the terminal driver routines it is necessary to describe the representation of command syntax briefly.

3.4.1.1 Individual ASCII characters

Individual ASCII characters sent to or from the terminal are enclosed in parentheses.

Example:

(A) The ASCII "A" character.

(ESC) "Escape" control character.

3.4.1.2 Sequence of ASCII characters

Expressions enclosed in "angle brackets" represent sequence of ASCII characters which might be sent to or from the 4112 terminal.

Example:

<xy> A pair of X and Y coordinates as encoded for transmission to the terminal as part of a command.

3.4.1.3 Specific values

To represent specific values of commands, specific values are included between the angle brackets.

Example:

<xy:0,1000> The coordinate pair X=0, Y=1000 as encoded for transmission to the terminal.

3.4.1.4 Parameters

Each command has its own op code consisting of one or more ASCII characters. After the op code there may be one or more parameters. The parameters have many types with each data type having its own syntax and coding scheme. The host computer has to send the parameters according to their type or the terminal will not be able to recognize them. Three of the parameter types that are most frequently used are:

<int> A sequence of characters representing an integer number.

<int-array> An array of <int> parameter, including a "count" for the array sent as an <int> parameter at the beginning.

<xy> A sequence of characters representing X- and Y- coordinates for some location on the 4112's screen.

3.5 Description of 4112 Terminal Driver Routines

The driver routines can be divided into three parts. One is for initialization status of the terminal which is called every time the DISPLAY program in MOVIE.BYU is executed and the Tektronix 4112 terminal is used as the graphic display device. Another is called when a line is to be drawn on terminal. It receives coordinates and light intensity values of the start and end points of a line that is to be drawn from the main program and issues appropriate commands to the terminal. The third part of the driver routine is a subroutine which is called whenever a picture is going to be sent from the host computer to the terminal. The purpose of this routine is to clear the screen and draw a square box around the graphic area on the screen.

3.5.1 Initialization Routine

When the Tektronix 4112 terminal is selected as the

graphic display device at the beginning of the execution of the DISPLAY program, many 4112 terminal status parameters must be set so that it can work consistantly with the host computer.

The terminal has eight different line styles: solid, dashed, etc. One of these must be selected before drawing a line on the screen can begin. Since the MOVIE.BYU software always draws solid lines, the computer must send a command sequence to set the line style to 0. The following command sequence accomplishes this.

```
(ESC)(M)(V)<int:0>
```

The 4112 has three "bit planes" of display memory. Each bit plane holds one binary bit for each pixel in the raster memory space. With 3 bits per pixel, a pixel's gray-indices can be any number from 0 to 7. Different gray index in raster memory will show up as different intensity on screen. With 1 bit per pixel, a pixel's gray-index can assume only two values black or white. The terminal's display area can be defined as 3 surfaces, each with one bit plane or as one surface with three bit planes. In the MOVIE.BYU software, only one surface is needed but each pixel should be able to assume more then two intensity levels for continuous tone image displays to be accomplished. So we define one surface with 3 bit planes by issuing the following command.

```
(ESC)(R)(D)(1)(3)
```

The 4112 terminal uses fraction numbers between 0 and 1 to represent gray-level. Gray-level 0 represents transparent and 1 represents black. For each gray-index we assume a gray-level to it according to Table 3.1.

Gray-index	Gray-level
0	transparent
1	25 %
2	30 %
3	45 %
4	60 %
5	70 %
6	85 %
7	90 %

Table 3.1 Gray-index-to-gray-level assignments.

The gray-level for each gray-index can be defined by issuing the following command.

```
(ESC)(R)(G)(1)<int-array:25,30,45,60,70,85,90>
```

The background of the display area of the 4112 can also

be assigned one of the eight gray-indices. For the purpose of flexibility in viewing the displayed object on the screen we wanted the background gray-index to be selectable. So, in the initialization routine the host computer asks the user to enter the background gray-level. The host computer then reads the gray-index entered by the user and issues a command to set the background gray-index as follows:

```
(ESC)(L)(I)(7)(1)<int:i><int:i>
```

Where i is the background gray-index entered by user.

The 4112 display area contains two parts: one is for conversational dialogs between the operator and the host computer called dialog area, the other is for displaying graphics called graphic area. The dialog area can be any of the three surfaces. Since there is only one surface in the application of MOVIE.BYU the dialog area is overlapped with the graphic area. The following command should be issued to set the dialog area to be on surface one.

```
(ESC)(L)(I)(1)
```

The dialog area can be set visible or invisible on the surface. Since we want to be able to see messages sent by the host computer at all times, we should set the dialog area to be visible by issuing the following command.

```
(ESC)(L)(V)(1)
```

The last step in the initialization is to set the size of the graphic area on screen. The screen of the Tektronix 4112 terminal consists of 4096-by-4096 pixels. The graphic area can be viewed as a window on the screen. Since we want the graphic area to be as large as possible, we set the window size to its maximum value 4096x3072. This can be done by issuing the following commands.

```
(ESC)(R)(W)<xy:0,0><xy:4096,3072>
```

```
(ESC)(R)(V)<xy:0,0><xy:4096,3072>
```

3.5.2 Line Drawing Routine

In MOVIE.BYU an object can be displayed as a line drawing or as a continuous tone shaded image. In the line drawing mode, when a line is to be drawn the main program tells the driver routine the coordinate of the start and end points of the line. The driver routine then issues a command to the terminal to move the beam to the start point, then issues another command to draw a line from the current beam position to the end point of the line. In continuous tone mode, besides telling the driver routine the coordinates of the start and end point of the line, the main program also specifies the light intensity of these two points. In MOVIE.BYU the light intensity is composed of three components

red, blue and green. The intensity of each component is represented as integers ranging from 0 for darkest to 255 for brightest. In 4112 there are only 8 gray-levels ranging from gray-index 0 for darkest to gray-index 7 for brightest. So, the gray-index of a point on screen can be computed by the following equation,

$$\text{Gray-index} = (\text{SR} + \text{SG} + \text{SB}) \times 1/3 \times 8/255 \quad (3.1)$$

where

SR : Intensity of red component of the point;

SB : Intensity of blue component of the point;

SG : Intensity of green component of the point.

Since SR, SB and SG in equation (2.1) are known for the start and end points of the line, the gray-indices of these two points can be computed. The gray-indices of every point in between the start and end points can then be computed using interpolation as shown in the equation below,

$$G_i = G_s + (G_e - G_s) \times d/D \quad (3.2)$$

where

G_i : Gray-index of a point i in between start and end points of a line;

G_s : Gray-index of start point of the line;

G_e : Gray-index of end point of the line;

- d : Distance from start point to point i;
- D : Distance between start and end point.

When drawing a line in continuous tone mode the driver routine first moves the beam to the start point of the line and sets the line-index to the gray-index of the start point, then the line is drawn point by point until it reaches the end point. Before a point is drawn its gray-index is computed by equation (3.2), if it is different from current line-index then the line-index will be set to gray-index of the point to be drawn. To set the line index the host computer sends a command as follows:

(ESC)(M)(L)<int:Gi>.

3.5.3 Clear Screen Routine

This routine is called before a picture is to be displayed on the screen. It first asks the user if the previous picture on screen should be erased before the new picture is going to be displayed by sending a prompt as "<OVERLAY>". If the answer is "YES" then nothing will be done. Otherwise the screen will be erased by sending a "PAGE" command to terminal and a square box will be drawn around the graphic area of the screen by sending 4 "Draw line" commands.

3.6 Apple Controller

The MOVIE.BYU software provides a program named DISPLAY which takes a data set describing the surface of an object as input and displays a 3-dimensional representation on graphic devices such as the Tektronix 4112 computer terminal. The object can be translated in any direction and/or rotated about any one of the X,Y or Z axis. DISPLAY also provides a command called "Animate" which allows the user to define a sequence of pictures and then have the computer display them. If the output is to the user's terminal, the program will display the first frame and then wait for a carriage return before displaying the next frame. Further carriage returns will display the rest of the sequence.

Once the data set of X,Y and Z coordinates describing the outline of vertebrae is available, it can be read by DISPLAY and then the vertebrae can be displayed on 4112 terminal. By using the "Animate" command and recording the frames displayed to videotape, the continuous motion of vertebrae can be shown. Because of the large amount of data that is transferred from host computer to terminal, the animation sequence is a time consuming process. For example, using a 1200 baud communication channel, it will take about 40 minutes to display an object described by 1500 surface points on the 4112 terminal. Under this condition it will take about 10 hours to complete an animation sequence consisting of 15 frames. Due to the necessity of interaction

between host computer and user in the animation sequence, a constant supervision is required in the production of video tape. In order to eliminate the need for constant supervision of the videotape production process an Apple IIe computer was employed to take over the interaction between user and host computer in the animation sequence. The connection between the Apple computer, host computer, Tektronix 4112 terminal and videotape recorder is shown in Figure 7.

The Apple computer is connected in parallel with the Tektronix 4112 terminal. During the animation sequence the Apple computer neglects the graphic commands sent by the host computer, however it keeps on monitoring the characters from the host. After the completion of a frame the host computer always send a specific sequence of characters. By detecting this sequence of characters the Apple computer will be able to know when a frame is finished. Once the completion of a frame is detected, the Apple computer activates a signal to tell the videotape recoder to record the frame. Having recorded the current frame, the Apple computer sends a signal to tell the recorder to stop the recording and then send a carriage return to the host computer to activate the sending of the next frame. After the animation sequence is completed the Apple computer halts its monitoring operation and acts just as a computer terminal so that the next animation sequence can be set. The Apple computer resumes its monitoring operation after the next animation sequence begins. By using the Apple computer, the production of

videotape can be automated, eliminating the need for constant human supervision. The assembly language program of Apple IIe for automating the production of videotape is in shown Appendix A.

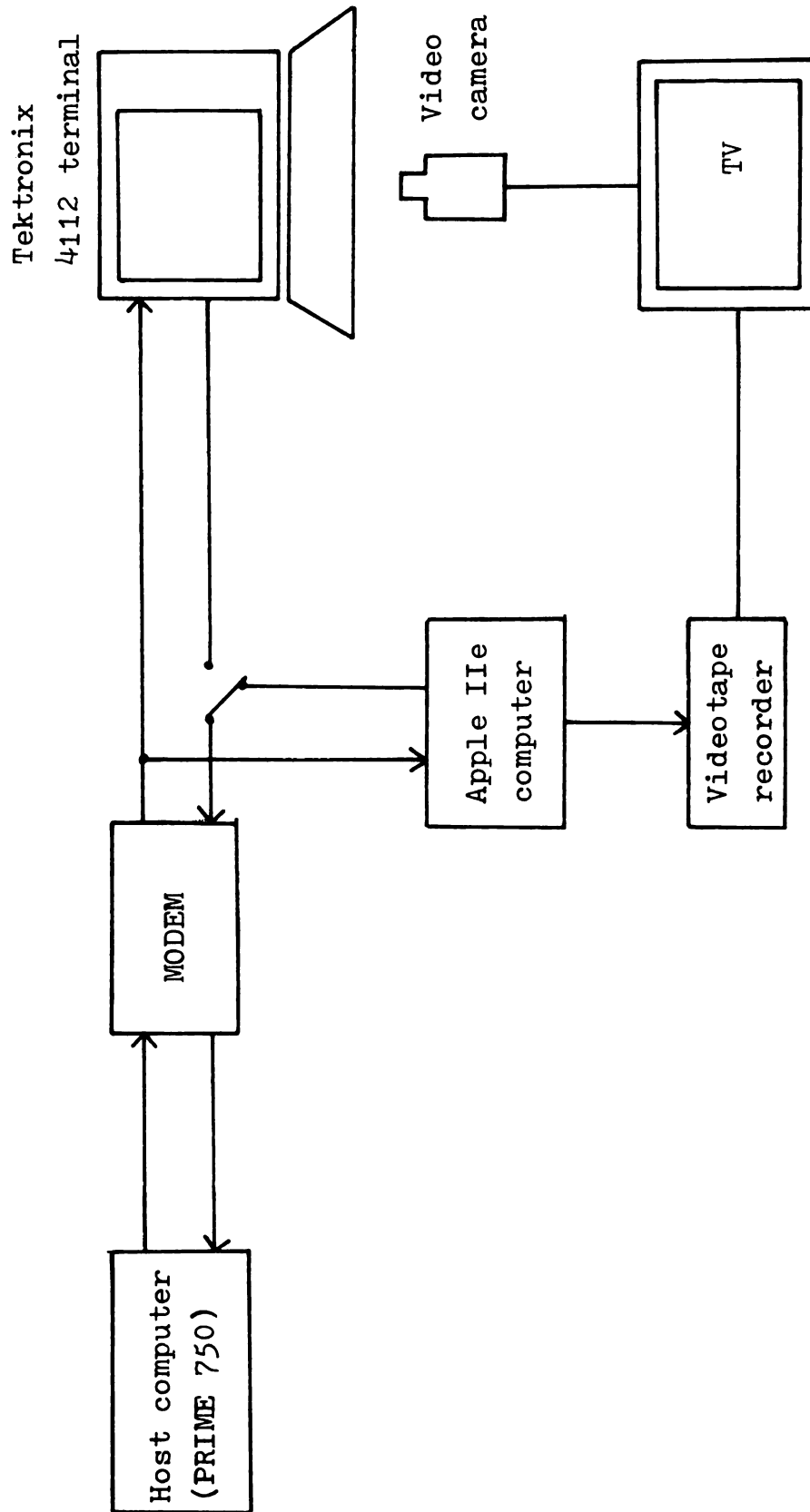


Figure 7. Block diagram of connections between PRIME computer, Apple IIe computer, Tektronix 4112 terminal and videotape recorder for automating videotape production.

CHAPTER IV

MATHEMATICS OF KINEMATICS

4.1 Coordinate Systems

Before the discussion of kinematics it will be helpful to define the following three coordinate systems for the vertebrae and the tungsten-carbid balls mounted on them.

(1) World coordinate System

The world coordinates of a ball refer to the locations of balls measured in the System Anthropometry Laboratory (SAL), representing ball coordinates in the human body. Units are in centimeters.

(2) Local Screen Coordinate

By processing the data derived from the CAT scan, the set of data that describe the topological surface of a vertebra and the location of balls mounted on it can be obtained. These data then can be read into a graphic program (MOSAIC) where the vertebrae will be reconstructed as a three-dimensional object on the screen. When performing the

CAT scan, each vertebra is placed arbitrarily and consequently these data do not represent the actual spatial position of each vertebra as they are in the human body. Each vertebra has its own data set and axis system that describe its shape, hence we say that these data are in the local screen coordinate system.

(3) Global Screen Coordinate

A series of coordinate transformations can be performed that transform the local screen coordinates of each vertebra into a global screen coordinate system. In this coordinate system, vertebrae can be represented as three-dimensional solid objects with the same spatial position as they are in the human body.

4.2 Static Kinematics

4.2.1 Coordinates of Balls in Local Screen Coordinate System

A Fortran program has been written which uses the data obtained from the CAT scan as input and determines the set of X and Y coordinates of the outline of each vertebra. The contours of these X and Y coordinates that form closed loops can be displayed on the Tektronix 4112 terminal. In running this program several parameters such as minimum number of points per loop, minimum and maximum distance in pixel

between two adjacent points, threshold level of intensity, etc. can be defined. By properly setting these parameters the balls mounted on vertebrae can be detected, appearing as small loops as shown in Figure 8. The center of these small loops can be defined as the centers of the balls in local screen coordinate can be approximated as follows:

$$X_c = 1/2(X_{\max} + X_{\min}) \quad (4.1)$$

$$Y_c = 1/2(Y_{\max} + Y_{\min}) \quad (4.2)$$

where

X_c = X coordinate of the ball in local screen coordinate system;

Y_c = Y coordinate of the ball in local screen coordinate system;

$X_{\max}(\min)$ = maximum(minimum) X coordinate of the small loop correspond to that ball;

$Y_{\max}(\min)$ = maximum(minimum) Y coordinate of the small loop correspond to that ball.

The Z coordinates of the balls in local screen coordinate can be approximated as the Z coordinates of these loops. Using the approximation described above, the local screen coordinates of balls detected can easily be determined. Care must be taken in detecting balls mounted on

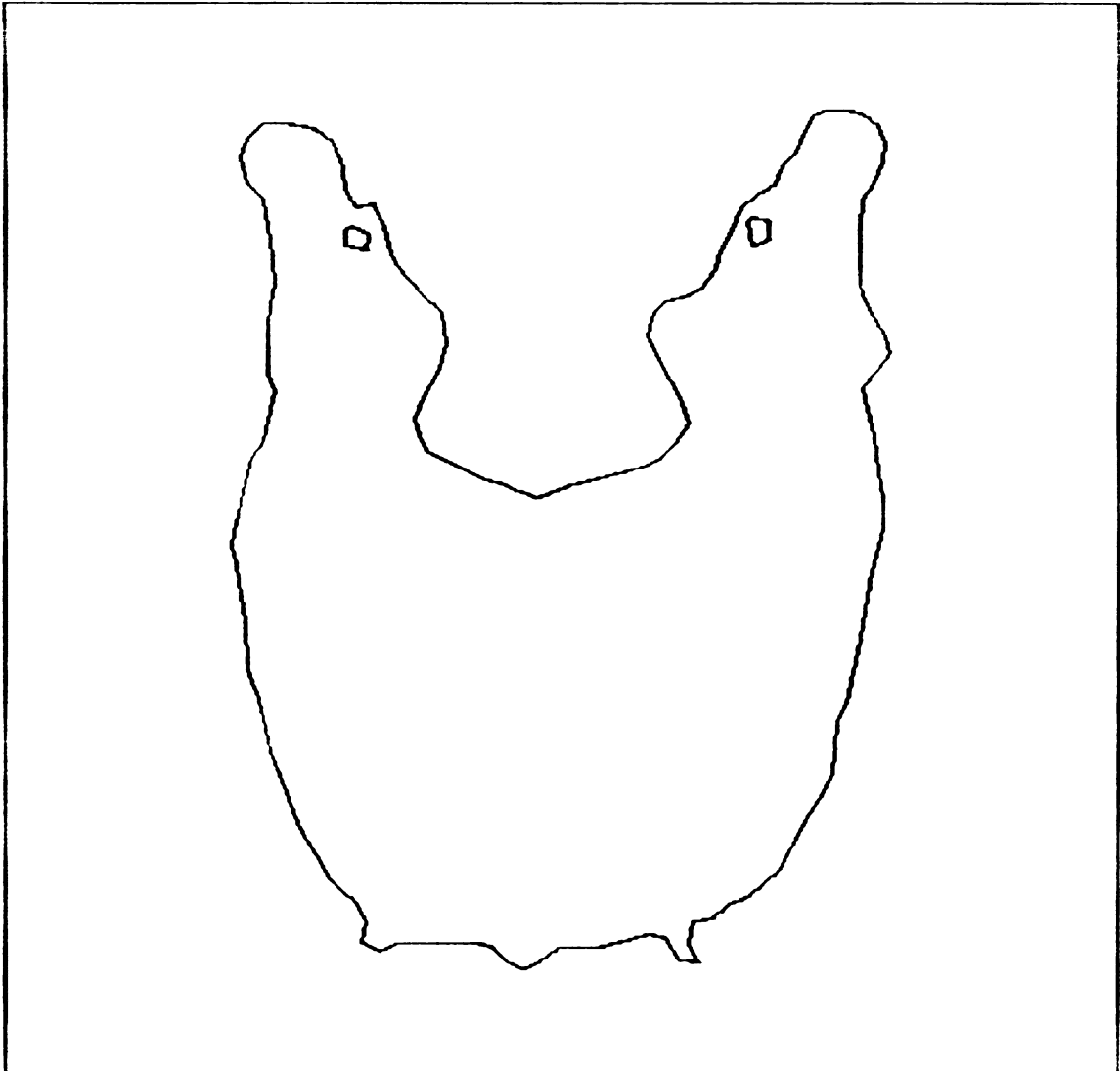


Figure 8. Appearance of balls on terminal screen.

vertebrae because some of the voids within the vertebrae will also appear as small loops on screen which might be misunderstood as balls. In order to prevent this kind of ambiguity, a procedure of verification is necessary. This can be done by computing the distances between any two balls in local screen coordinates and compare them with the distances in world coordinates. Since a length of 17.23 cm in world coordinates will occupy 512 pixels in local screen coordinates, the distance between two points in local screen coordinates should be 512/17.23 times as long as the distance between those two points in world coordinates. A Fortran program has been written (Appendix B) which uses the world and local screen coordinates of balls as input and then compute the distances between balls in world coordinates and local screen coordinates. The distances in local screen coordinates is compared with the theoretical distances in local screen coordinates (i.e. distance in world coordinate $\times 512/17.23$) and the error between them is reported. Those balls with the largest error in distance should be discarded. An example of the output from this program with balls in L5 is shown in Appendix C. We see that the balls with numbers 5, 8 and 9 always have larger error in distance than the others. The local screen coordinates of these three balls are inaccurate and should be discarded, leaving only balls with number 1, 2, 3, 4, 6 and 7 having accurate local screen coordinates.

4.2.2 Coordinate Transformation

A point X in a three-dimensional space can be represented as a homogeneous coordinate form as: (11)

$$X = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (4.3)$$

and the transformation from one coordinate system to another which always include rotation, scaling and translation can be represented as a matrix multiplication as follows:

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} & x_1 \\ m_{21} & m_{22} & m_{23} & m_{24} & y_1 \\ m_{31} & m_{32} & m_{33} & m_{34} & z_1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \quad (4.2)$$

or

$$X_2 = M * X_1 \quad (4.3)$$

where

X_1 = the coordinate of a point in coordinate system 1;

X_2 = the coordinate of the same point in coordinate system 2;

M = transformation matrix.

The transformation matrix M can be computed if the angle of rotation, scaling factor and amount of translation are known. Once this transformation matrix is known, every point in coordinate system 1 can easily be mapped into coordinate system 2.

4.2.3 Transformation From Local Screen Coordinate System to Global Screen Coordinate System

Once the coordinates of balls in the local screen coordinate system and the global screen coordinate system are known, we can find a transformation matrix for each vertebra that transforms the position of the balls from the local screen coordinates into global screen coordinates. These matrices then can be used to transform the 3-dimensional data of each vertebra in local screen coordinate system into global screen coordinate system. The transformed 3-dimensional data of each vertebra in the global screen coordinate system then represents the spatial position of each vertebra as they are in human body. Multiple vertebra can then be merged together and displayed on the graphics terminal.

4.2.4 The Derivation of Transformation Matrices

The transformation between two coordinate systems always include axis rotation, scaling and translation. The transformation matrix can be defined explicitly if the parameters associated with rotation, scaling and translation are known. Sometimes it is difficult to know these parameters. Sometimes only the coordinates of some common points in both coordinate systems are known. Simple arithmetic operations can then be used to determine the transformation matrix M .

From equation (4.2) we know that

$$M = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} . \quad (4.4)$$

There are 12 unknowns in matrix M . Each point with coordinates in both coordinate systems yields 3 linear equations.

$$x_2 = m_{11}x_1 + m_{12}y_1 + m_{13}z_1 + m_{14} \quad (4.5)$$

$$y_2 = m_{21}x_1 + m_{22}y_1 + m_{23}z_1 + m_{24} \quad (4.6)$$

$$z_2 = m_{31}x_1 + m_{32}y_1 + m_{33}z_1 + m_{34} \quad (4.7)$$

So, 4 such points will be enough to determine the 12

unknowns m_{ij} . However, if more than 4 points are known no definite solution will exist and a least square error method [12,13] should be used to determine the 12 unknowns with minimum error. For k points with coordinates in both coordinate system 1 and system 2 known, we can write 3 matrix equations as follows:

$$\begin{bmatrix} x_{21} \\ x_{22} \\ . \\ . \\ . \\ x_{2k} \end{bmatrix} = \begin{bmatrix} x_{11} & y_{11} & z_{11} & 1 \\ x_{12} & y_{12} & z_{12} & 1 \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ x_{1k} & y_{1k} & z_{1k} & 1 \end{bmatrix} \begin{bmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \end{bmatrix} \quad (4.8)$$

$$\begin{bmatrix} y_{21} \\ y_{22} \\ . \\ . \\ . \\ y_{2k} \end{bmatrix} = \begin{bmatrix} x_{11} & y_{11} & z_{11} & 1 \\ x_{12} & y_{12} & z_{12} & 1 \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ x_{1k} & y_{1k} & z_{1k} & 1 \end{bmatrix} \begin{bmatrix} m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \end{bmatrix} \quad (4.9)$$

$$\begin{bmatrix} z_{21} \\ z_{22} \\ . \\ . \\ . \\ z_{2k} \end{bmatrix} = \begin{bmatrix} x_{11} & y_{11} & z_{11} & 1 \\ x_{12} & y_{12} & z_{12} & 1 \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ x_{1k} & y_{1k} & z_{1k} & 1 \end{bmatrix} \begin{bmatrix} m_{31} \\ m_{32} \\ m_{33} \\ m_{34} \end{bmatrix} \quad (4.10)$$

where

x_{ij} = x coordinate of point j in coordinate system i;

y_{ij} = y coordinate of point j in coordinate system i;

z_{ij} = z coordinate of point j in coordinate system i.

Since all x_{ij} , y_{ij} and z_{ij} for $i=1,2$ and $j=1,2,3,\dots,k$ are known, $[m_{11} \ m_{12} \ m_{13} \ m_{14}]$ can be obtained by solving equation (4.8) using the least square error technique. Similarly $[m_{21} \ m_{22} \ m_{23} \ m_{24}]$ and $[m_{31} \ m_{32} \ m_{33} \ m_{34}]$ can be obtained by solving equation (4.9) and (4.10) respectively. Lawson [14] has written a Fortran IV program to solve the least square error problems. This program has been modified to a FORTRAN V program and used in deriving the transformation matrices.

In the global screen coordinate system, all the vertebrae will be located in positions as they are in the human body. If we transform the local screen coordinate system of L3 and L4 into local screen coordinate system of L5, then the relative position of these three vertebrae in local screen coordinate system of L5 will look like they are in global screen coordinate system. We can choose the local screen coordinate system of L5 as the global screen coordinate system. Since the world coordinates and local screen coordinates of balls in L5 are known, the transformation matrix M for transforming world coordinates

into local screen coordinates for balls in L5 can be obtained using the least square error technique. The matrix M also can be used to transform the world coordinates of balls in L3 and L4 into their global screen coordinates. Since the local screen coordinates of balls in L3 and L4 are known, once the global screen coordinate of these balls are derived the least square error technique can be applied again and two matrices M_{lg3} and M_{lg4} can be obtained that transform the balls from local screen coordinates into global screen coordinates.

The 3-dimensional image of the vertebrae are only sets of data that describe the shape of vertebrae in their local screen coordinate system. These data also can be transformed to global screen coordinate system using the transformation matrices for transforming balls.

For example, if we multiply matrix M_{lg4} with the local screen coordinate of every points in 3-dimensional model of L4 $[x_i \ y_i \ z_i \ 1]^T$, we will get a new set of data that represent the 3-dimensional model of L4 in global screen coordinate system. A Fortran program for transforming the 3-dimensional model of vertebra from a coordinate system to another coordinate system is in Appendix D.

By merging the data representing the 3-dimensional model of each vertebra in global screen coordinate system together, a new set of data that represent the real 3-dimensional model of vertebrae is constructed and can be shown on the screen. The 3-dimensional model of L3, L4 and L5 in global screen coordinate system is shown in Figure 9.

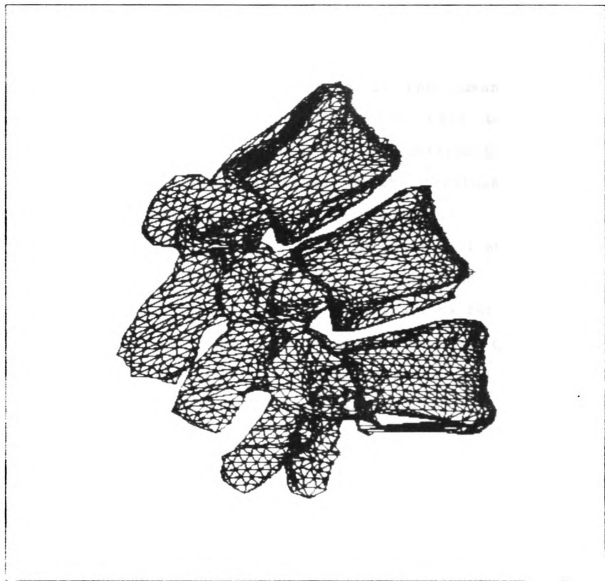


Figure 9. Three-dimensional model of L3, L4 and L5.

4.3 Dynamic Kinematics

In the System Antropometry Laboratory many sets of world coordinates of balls corresponding to the human body in different positions were obtained. For each position a 3-dimensional model of vertebrae can be constructed by going through the procedures described in the previous section. However a much simpler method can be employed.

Assume two sets of world coordinates of ball X_{w1} and X_{w2} that correspond to human body in position 1 and 2 respectively are available. Using position 1 as reference X_{w2} can be viewed as a rotated/translated version of X_{w1} . A transformation matrix T satisfying equation (4.11) can be obtained using least square error technique.

$$X_{w2} = T * X_{w1} \quad (4.11)$$

For X_{w1} and X_{w2} a transformation matrix M can be found that transform the balls from world coordinates to global screen coordinates.

$$X_{g1} = M * X_{w1} \quad (4.12)$$

$$X_{g2} = M * X_{w2} \quad (4.13)$$

where

$X_{g1(2)}$: Global screen coordinates of balls correspond

to human body position 1(2).

Substituting (4.11) into (4.13) yields

$$X_{g2} = M * T * X_{w1} \quad (4.14)$$

From (4.12) we know

$$X_{w1} = M^{-1} * X_{g1} \quad (4.15)$$

Combining (4.14) and (4.15) we get

$$X_{g2} = M * T * M^{-1} * X_{g1} \quad (4.16)$$

Knowing matrices M and T and the 3-dimensional models of vertebrae corresponding to human body in position 1, the 3-dimensional models of vertebrae corresponding to the human body in a new position can be obtained by equation (4.16).

While collecting world coordinates of balls in SAL, the human body was placed in many different positions and for each position a set of coordinates were recorded. The sequential change in position of the human body yields sequential sets of world coordinates of balls that in turn can be used to construct a series of 3-dimensional models of vertebrae. The motion of the human vertebrae can be investigated by showing the 3-dimensional model of vertebrae for different human body positions. Figures 10, 11 and 12

show 3-dimensional model of L3, L4 and L5 for different human body positions.

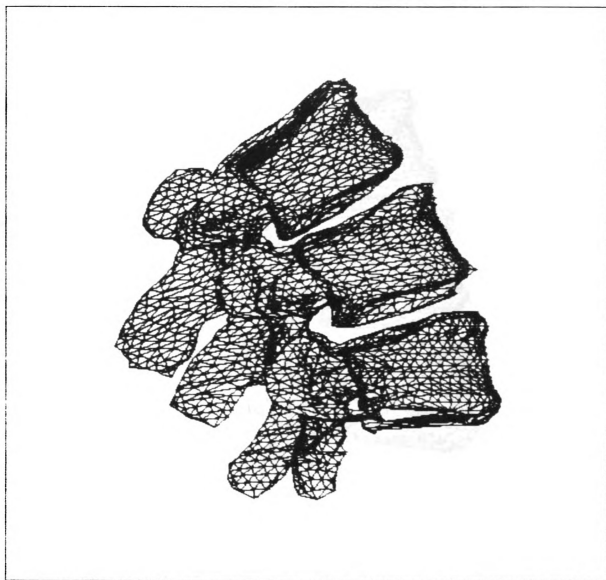


Figure 10. Three-dimensional model of L3, L4 and L5.

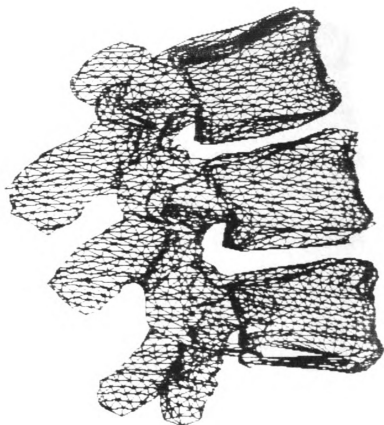


Figure 11. Three-dimensional model of L3, L4 and L5.

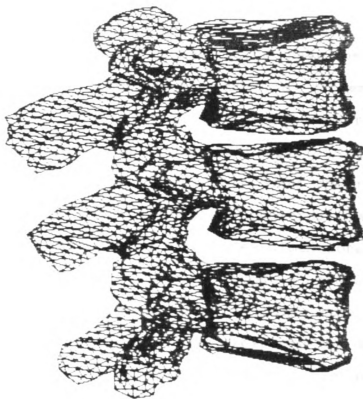


Figure 12. Three-dimensional model of L3, L4 and L5.

CHAPTER V

CONCLUSIONS

Many models of the human spine have been developed, however none are committed to both a mathematically and visually correct presentation. The model constructed in this thesis not only represents the real spatial relationship of vertebrae but can also dynamically show the motion of the human spine. The model combines fundamental research in the area of human biomechanics with three-dimensional solid image generation technology. Use of such a model will allow individuals to expand upon research being conducted in such areas as impact testing, human response to vibration and analysis of forces developed during athletic activities. In addition, the model will serve as a sophisticated educational tool to assist in the understanding of complex relationship between soft tissues and body structure of the spine as a function of position and time.

Although the model, includes only L3, L4 and L5, the number of vertebrae can be extended to include the whole human lumbar vertebrae or even the whole spine. However, since the graphic area of the Tektronix 4112 terminal is fixed, increasing the number of vertebrae will reduce the resolution of the model. A possible remedy to this problem is

to upgrade the graphic display devices. The other impact in increasing the number of vertebrae in the model is that the size of data of the model will be increased proportionally with the number of vertebrae. One way to increase the number of vertebrae without increasing the amount of data dramatically is to decrease the resolution of the model.

Motion of the spine can also be investigated. A series of images for each vertebra, for sequentially changes in the human body position, can be shown on the 4112 terminal and then recorded on videotape as described in chapter 2. The real time motion of lumbar vertebrae can then be presented to an audience by playing the videotape.

APPENDICES

APPENDIX A

6502 ASSEMBLY PROGRAM FOR AUTOMATING VIDEOTAE RODUCTION

CH	EPZ	\$24
CV	EPZ	\$25
BASL	EPZ	\$28
BASH	EPZ	\$29
BAS2L	EPZ	\$2A
BAS2H	EPZ	\$2B
TEMP	EPZ	\$5E
PROMPT	EPZ	\$33
INDX	EPZ	\$5F
BUFFER	EPZ	\$60
COUNT	EPZ	\$5C
STORE	EPZ	\$5C
FLAG	EPZ	\$5D
SETNORM	EQU	\$FE84
INIT	EQU	\$FB2F
SETKBD	EQU	\$FE89
BELL1	EQU	\$FBDD
CLREOZ	EQU	\$FC9C
CONTROL	EQU	\$7FA
COUT1	EQU	\$FDF0
BASCALC	EQU	\$FBC1
DIRECT	EQU	\$77A
KBD	EQU	\$C000
KBDSTRB	EQU	\$C010
SEROUT	EQU	\$C205
STATUS	EQU	\$C0AE
DATA	EQU	\$C0AF
RESET	EQU	\$FF59
WAIT	EQU	\$FCA8
BS	EQU	\$FC10
ON	EQU	\$C059
OFF	EQU	\$C058
	ORG	\$1000
	LDA	#\$03
	STA	STATUS
	JSR	SETNORM
	JSR	INIT
	JSR	SETKBD
	JSR	BELL1
	CLD	
	LDA	#0

```

        STA  CH
        STA  OFF
        STA  FLAG
        LDA  17
        STA  CV
        JSR  CLREOZ
        LDA  #$09
        STA  STATUS
        LDA  #$80
        STA  DIRECT
        LDA  #$DF
        STA  PROMPT
        LDA  PROMPT
        JSR  COUT1
L1      LDY  CH
        LDA  (BASL),Y
        PHA
        AND  #$3F
        ORA  #$40
        STA  (BASL),Y
KEYIN   BIT  KBD
        BPL  TEST
        LDA  KBD
        BIT  KBDSTRB
LL      AND  #$7F
        JSR  SEROUT
TEST    LDA  STATUS
        LSR
        BCC  KEYIN
        PLA
        STA  (BASL),Y
        LDA  DATA
        JSR  PROCESS
        JMP  L1
PROCESS CMP  #$1D
        BEQ  SCAN
PROCES1 CMP  #$0D
        BNE  DISP
        JMP  CR
DISP    ORA  #$80
        CMP  #$88
        BNE  LLL
        JMP  BS
LLL     CMP  #$A0
        BCC  RETD
        CMP  #$E0
        BCC  UPPER
        AND  #$DF
UPPER   LDY  CH
        STA  (BASL),Y
        INC  CH
        LDA  CH
        CMP  #$28
        BCC  RETD

```

```

      JMP      CR
RETD   RTS
SCAN   LDA     #$00
      STA     FLAG
SCAN0  LDX     #$00
LOOPS  LDA     STATUS
      LSR
      BCC     LOOPS
      LDA     DATA
      STA     STORE
      CMP     TABLE,X
      BNE     TST
      INX
      CPX     #$05
      BNE     LOOPS
      LDA     FLAG
      BNE     BEEP
      INC     FLAG
      BNE     SCAN0
TST    LDA     FLAG
      BEQ     RETS
      BNE     SCAN0
RETS   LDA     STORE
      ORA     #$80
      JSR     DISP
      RTS
BEEP   LDA     STATUS
      LSR
      BCC     BEEP
      LDA     DATA
      CMP     #$0D
      BEQ     BL
      JSR     DISP
      JMP     BEEP
BL     LDA     #$00
      STA     FLAG
      JSR     CR
      LDA     #$28
      STA     COUNT
BLOOP  LDA     STATUS
      LSR
      BCC     BL1
      LDA     DATA
      JSR     PROCES1
BL1    LDA     #$28
      JSR     WAIT
      DEC     COUNT
      BNE     BLOOP
      JSR     BELL1
      LDA     #$A0
      JSR     WAIT
      JSR     BELL1
      LDA     ON
      LDA     #$80

```

```

LLOOP      STA  COUNT
           LDA  #$FF
           JSR  WAIT
           DEC  COUNT
           BNE  LLOOP
           LDA  OFF
           LDA  #$0D
           JSR  SEROUT
           RTS

CR          LDA  #$00
           STA  INDX
           PHA
           STA  CH
           STA  BASL
           LDA  #$04
           STA  BASH
           CLC

SCRL1      LDA  BASL
           STA  BAS2L
           LDA  BASH
           STA  BAS2H
           LDY  #$28
           DEY
           PLA
           CLC
           ADC  #$01
           CMP  #$18
           BCS  SCRL3
           PHA

SCRL2      JSR  BASCALC
           LDA  (BASL),Y
           STA  (BAS2L),Y
           DEY
           BPL  SCRL2
           LDA  STATUS
           LSR
           BCC  SCRL1
           LDX  INDX
           LDA  DATA
           STA  BUFFER,X
           INC  INDX
           BCS  SCRL1

SCRL3      JSR  CLREOZ
           LDA  #$D0
           STA  BASL
           LDA  #$07
           STA  BASH
           LDA  #$00
           STA  TEMP

LOOP1      LDA  INDX
           BNE  L2
           RTS

L2          LDX  TEMP
           LDA  BUFFER,X

```

	JSR	DISP
	DEC	INDX
	INC	TEMP
	BNE	LOOP1
TABLE	BYT	\$37,\$7F,\$20,\$40,\$16
	END	

APPENDIX B

PROGRAM FOR COMPARING DISTANCES BETWEEN BALLS IN WORLD AND SCREEN COORDINATE SYSTEMS

```

PROGRAM DISTAN
DIMENSION WCOOR(20,3),SCOOR(20,3)
CHARACTER*80 IPFILE,OPFILE
10 PRINT *,'<COOR. FILE NAME>'
   READ(*,'(A)',ERR=10) IPFILE
   OPEN(FILE=IPFILE,UNIT=5,ERR=10)
   READ(5,*) N
   IF (N.GT.20.OR.N.LT.1) THEN
       PRINT *,'<INVALID NUMBER>'
       CLOSE(5)
       STOP
   END IF
90 PRINT *,'<OUTPUT FILE NAME>'
   READ(*,'(A)',ERR=90) OPFILE
   OPEN(FILE=OPFILE,UNIT=6,ERR=90)
   WRITE(6,110)
   DO 20 I=1,N
       READ(5,*) (WCOOR(I,J),J=1,3),(SCOOR(I,J),J=1,3)
       WRITE(6,120) I,(WCOOR(I,J),J=1,3),(SCOOR(I,J),J=1,3)
20  CONTINUE
   WRITE(6,150)
   DO 30 I=1,N-1
       DO 30 J=I+1,N
           WDIST=(WCOOR(I,1)-WCOOR(J,1))**2+(WCOOR(I,2)-
1WCOOR(J,2))**2+(WCOOR(I,3)-WCOOR(J,3))**2
           SDIST=(SCOOR(I,1)-SCOOR(J,1))**2+(SCOOR(I,2)-
1SCOOR(J,2))**2+(SCOOR(I,3)-SCOOR(J,3))**2
           WDIST=SQRT(WDIST)
           SDIST=SQRT(SDIST)
           TDIST=WDIST*512./17.23
           ERROR=SDIST-TDIST
           WRITE(6,100) I,J,WDIST,SDIST,TDIST,ERROR
30  CONTINUE
       CLOSE(5)
       CLOSE(6)
       STOP
100 FORMAT(/,T8,I2,4X,I2,T20,4F10.2)
110 FORMAT(/,T26,'LOCATION OF BALLS',/,T14,'WORLD COOR.
1 (CM)',T40,'SCREEN COOR. (PIXEL)',/,T4,'BALL #',T15,
2 'X',T23,'Y',T32,'Z',T42,'X',T50,'Y',T58,'Z')

```

```
120 FORMAT(/,5X,I2,T10,3F8.2,T37,3F8.2)
150 FORMAT(/,3X,'DISTAN BETWEEN',T27,'CM',T34,'PIXEL',T44,
1 'THEO.',T55,'ERROR',/,T4,'BALL I AND J')
END
```

APPENDIX C

AN EXAMPLE OF OUTPUT FROM PROGRAM IN APPENDIX B

BALL #	LOCATION OF BALLS					
	WORLD COOR. (CM)			SCREEN COOR. (PIXEL)		
	X	Y	Z	X	Y	Z
1	-0.99	-0.79	19.80	269.50	377.50	41.27
2	1.13	1.62	19.58	330.50	309.00	60.50
3	1.50	-2.62	19.55	204.00	311.50	60.50
4	4.50	-2.83	19.49	186.00	229.50	85.97
5	6.25	-0.31	19.04	264.5	165.5	92.15
6	1.53	1.46	22.09	322.00	319.00	137.46
7	2.11	-2.32	22.07	211.00	316.00	137.46
8	4.80	-2.96	21.64	180.00	240.00	154.53
9	6.61	0.16	21.32	264.00	178.00	166.93

DISTANCE BETWEEN BALL I AND J		CM	PIXEL	THEO.	ERROR
1	2	3.22	93.72	95.60	-1.89
1	3	3.10	94.95	92.12	2.83
1	4	5.86	175.71	174.28	1.43
1	5	7.30	218.08	216.79	1.28
1	6	4.08	124.22	121.28	2.94
1	7	4.14	128.28	122.89	5.39
1	8	6.45	199.36	191.70	7.65
1	9	7.81	235.84	232.03	3.80
2	3	4.26	126.52	126.48	0.05
2	4	5.58	166.88	165.90	0.98
2	5	5.50	161.09	163.38	-2.30
2	6	2.55	78.07	75.68	2.39
2	7	4.76	142.31	141.53	0.78
2	8	6.22	190.40	184.83	5.57
2	9	5.93	181.41	176.28	5.14
3	4	3.01	87.73	89.38	-1.65
3	5	5.31	161.18	157.69	3.49
3	6	4.81	141.08	142.82	-1.74
3	7	2.61	77.41	77.56	-.015
3	8	3.92	120.54	116.51	4.03
3	9	6.08	180.97	180.69	0.28
4	5	3.10	101.47	92.14	9.33
4	6	5.83	170.76	173.23	-2.48
4	7	3.55	103.72	105.60	-1.88

4	8	2.17	69.62	64.62	5.00
4	9	4.09	123.66	121.58	2.07
5	6	5.89	166.03	163.73	2.29
5	8	3.99	128.77	118.43	10.34
5	9	2.36	75.82	70.00	5.82
6	7	3.82	111.04	113.64	-2.60
6	8	5.52	163.39	163.93	-0.54
6	9	5.30	155.29	157.49	-2.21
7	8	2.80	83.84	83.15	0.68
7	9	5.19	150.74	154.30	-3.56
8	9	3.62	105.14	107.61	-2.47

APPENDIX D

PROGRAM FOR TRANSFORMING 3-DIMENSIONAL MODEL OF VERTEBRA FROM A COORDINATE SYSTEM TO ANOTHER COORDINATE SYSTEM

```

PROGRAM MAPP
COMMON/MAP/P,IP
DIMENSION P(3,8000),NPL(2,27),IP(240000)
DIMENSION C(4,4)
CHARACTER*80 IPFILE,OPFILE
10 PRINT *,'<ENTER TRANSFORMATION MATRIX FILE NAME>'
   READ(*,'(A)',ERR=10) IPFILE
   OPEN(FILE=IPFILE,UNIT=5,ERR=10)
   DO 20 I=1,3
      READ (5,*) (C(I,J),J=1,4)
20 CONTINUE
   CLOSE(5)
   C(4,1)=0.
   C(4,2)=0.
   C(4,3)=0.
   C(4,4)=1.
30 PRINT *,'<ENTER INPUT GEOM FILE NAME>'
   READ(*,'(A)',ERR=30) IPFILE
   OPEN(FILE=IPFILE,UNIT=5,ERR=30)
40 PRINT *,'<ENTER TRANSFORMED GEOM FILE NAME>'
   READ(*,'(A)',ERR=40) OPFILE
   OPEN(FILE=OPFILE,UNIT=6,ERR=40)
   READ(5,100) NP,NJ,NPT,NCOM,NTEST
   READ(5,100) ((NPL(I,J),I=1,2),J=1,NP)
   READ(5,110) ((P(I,J),I=1,3),J=1,NJ)
   READ(5,100) (IP(K),K=1,NCOM)
   DO 60 I=1,NJ
      P1=P(1,I)
      P2=P(2,I)
      P3=P(3,I)
      P(1,I)=C(1,1)*P1+C(1,2)*P2+C(1,3)*P3+C(1,4)
      P(2,I)=C(2,1)*P1+C(2,2)*P2+C(2,3)*P3+C(2,4)
      P(3,I)=C(3,1)*P1+C(3,2)*P2+C(3,3)*P3+C(3,4)
60 CONTINUE
   WRITE(6,100) NP,NJ,NPT,NCOM,NTEST
   WRITE(6,100) ((NPL(I,J),I=1,2),J=1,NP)
   WRITE(6,110) ((P(I,J),I=1,3),J=1,NJ)
   WRITE(6,100) (IP(I),I=1,NCOM)
   CLOSE(5)
   CLOSE(6)

```

```
      STOP  
100  FORMAT(16I5)  
110  FORMAT(6E12.5)  
      END
```

REFERENCES

REFERENCES

1. M. M. Panjabi and A. A. White, A Mathematical Approach for Three-Dimensional Analysis of the Mechanics of the Spine, J. Biomechanics, 14:203:211, 1969.
2. A. A. White and M. M. Panjabi, The Basic Kinematics of the Human Spine, Spine, Vol 3, No. 1, pp. 12-29, 1978.
3. A. B. Schultz and J. O. Galante, A Mathematical Model for the Study of the Mechanics of the Human Vertebral Column, J. Biomechanics, 3:405-416, 1970.
4. A. H. Soni, J. A. Sullivan, A. G. Patwardhan, M. R. Gudavalli and J. Chitwood, Kinematic Analysis and Simulation of Vertebral Motion Under Static Load - Part I : Kinematic Analysis, Transaction of the ASME, 104:105-111, 1982.
5. A. G. Patwardhan, A. H. Soni, J. A. Sullivan, M. R. Gudavalli and V. Srinivasan, Kinematic Analysis and Simulation of Vertebral Motion Under Static Load - Part II : Simulation Study, Transaction of the ASME, 104:112-118, 1982.
6. N. I. Badler, Human Body Models and Animation, IEEE Computer Graphics and Applications, 2(9):6-7, 1982.

7. T. W. Calvert, J. Chapman and A. Patla, Aspects of the Kinematic Simulation of Human Movement, IEEE Computer Graphics and Applicationa 2(9):41-50, 1982.
8. W. A. Fetter, A Progression of Human Figures Simulated by Computer Graphics, IEEE Computer Graphics and Applications, 2(9):9-13, 1982.
9. Tektronix, 4112 Computer Display Terminal Operator's Manual, Tektronix, 1981.
10. Tektronix, 4112 Computer Display Terminal Host Programmer's Manual, Tektronix, 1981.
11. David F. Rogers, J. Alan Adams, Computer Graphics, McGraw-Hill Book Co., N. Y., 1979
12. W. C. Rhienboldt, J. M. Ortega, Interactive Solution of Nonlinear Equations in Several Variables, Academic Press, N. Y., 1970. Chapter 5.
13. J. Stoer, R. Bulirsch, Introduction to Numerical Analysis, Springer-Verlag Inc, New York, 1980.
pp. 199-213.
14. Charles L. Lawson, Richard J. Handson, Solving Least Squares Problems, Prentice-Hall, Englewood Cliffs. N. J., 1974.