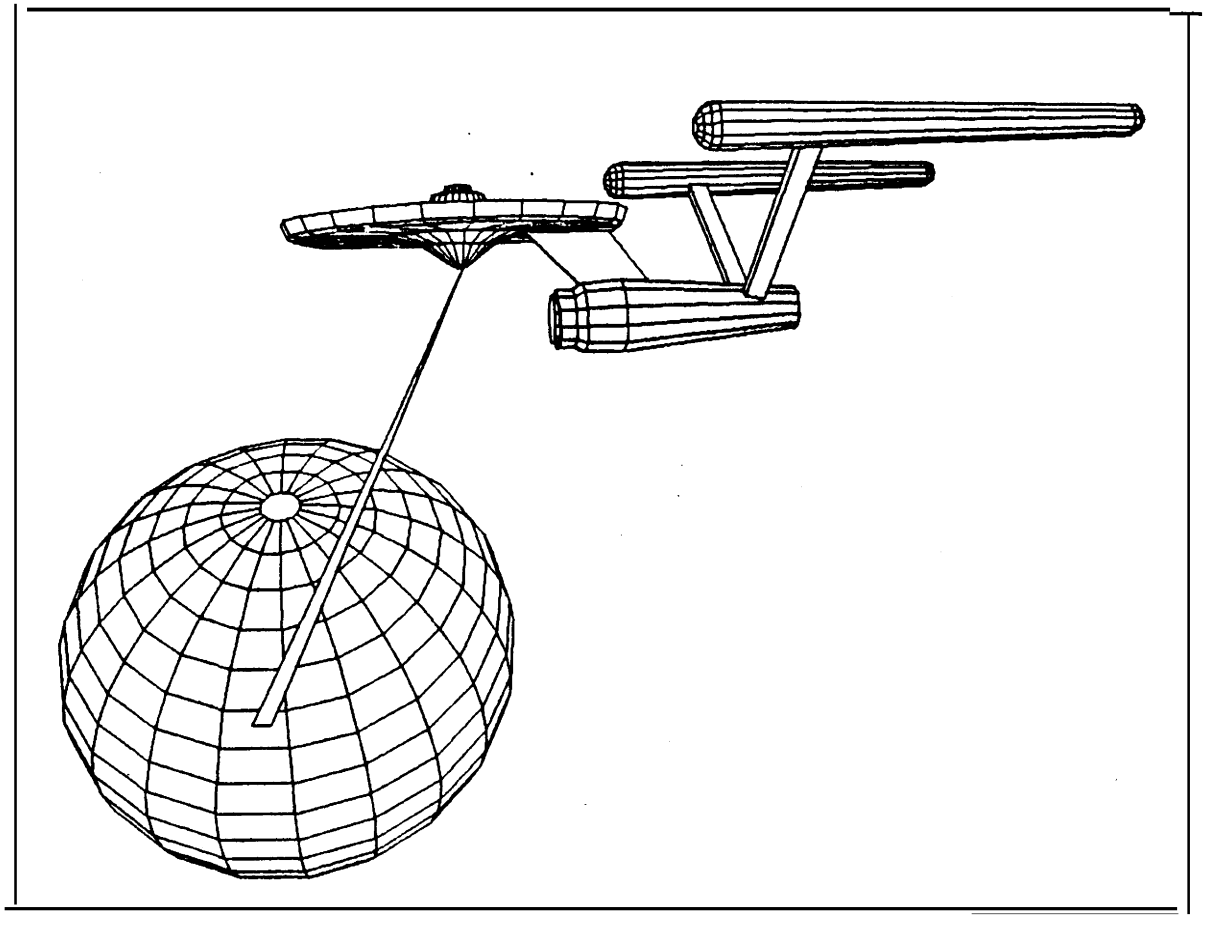
GEOMED



Geometric Editor

Supported by
Advanced Research Projects Agency
ARPA Order 2495

STANFORD ARTIFICIAL INTELLIGENCE LABORATORY
MEMO AIM-232
COMPUTER SCIENCE DEPARTMENT REPORT ND. CS-414

MAY 1974

GEOMED - A GEOMETRIC EDITOR.

Bruce G. Baumgart

ABSTRACT:

GEOMED is a system for doing 3-D geometric modeling; used from a keyboard, it is an interactive drawing program; used as a package of SAIL or LISP accessible subroutines, it is a graphics language. With CEOMED, arbitrary polyhedra can be contructed; moved about and viewed in perspective with hidden lines eliminated. In addition to polyhedra; camera and image models are provided so that simulators relevant to computer vision, problem solving, and animation may be constructed.

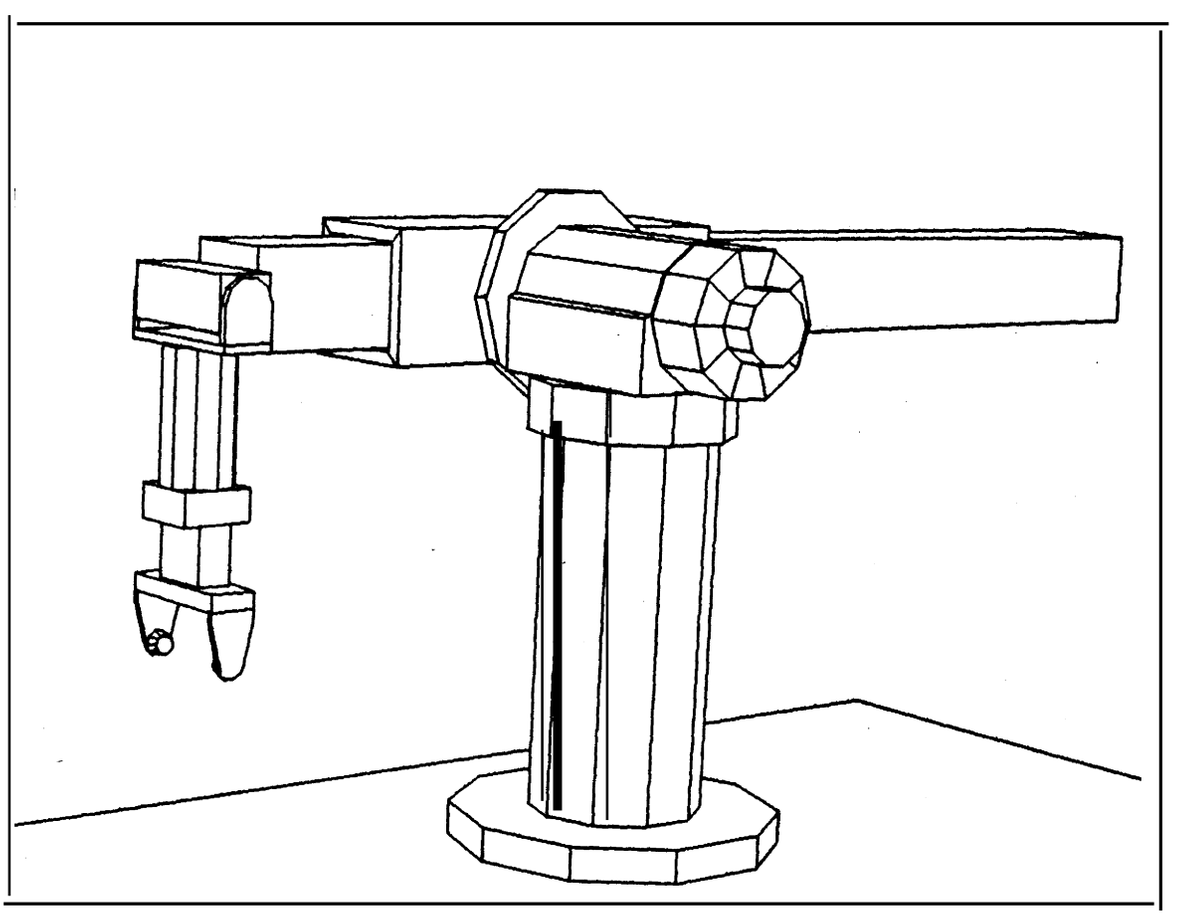
This research was supported in part by the Advanced Research Projects Agency of the Office *of the Secretary of Defense under Contract DAHC15-73-C-0435. The views and conclusions contained* in *this document are those of the author and should not be interpreted as necessarily representing the* official *policies, either expressed* or *implied, of the Advanced Research Projects Agency or the United States Government.*

Reproduced in *the U.S.A.* Available from the National *Technical Information Service,* Springfield, Virginia 22151.

CONTENTS.

GEOMED - A GEOMETRIC EDITOR.

1. PRIMER.....PAGE 1.
2. EXAMPLES.....PAGE 8.
3. GEOMETRIC COMMANDS.....PAGE 16.
4. TOPOLOGICAL COMMANDS.....PAGE 21.
5. INPUT/OUTPUT COMMANDS.....PAGE 28.
6. EDITOR CONTROL COMMANDS.....PAGE 30,
7. DISPLAY CONTROL COMMANDS.....PAGE 35.
a. ADDENDA.....PAGE 40.



DETAILED CONTENTS.

PAGE SECTION CONTENTS

1. SECTION 1. PRIMER.

8. SECTION 2. EXAMPLES.

16. SECTION 3. GEOMETRIC COMMANDS.

17. 3.1 EUCLIDEAN TRANSFORMATION KEYS.
Translation, « Rotation, β Dilation, ‹ Reflection.
";" Transform Minus X-Axis. ":" ...Plus X-Axis.
"(" Transfora Minus Y-Axis. ")" ...Plus Y-Axis.
--- Transfora Minus Z-Axis. "@" ...Plus Z-Axis.
"!" Translation Defwlt. ● e- Rotation Defwlt.

17. 3.2 STRENGTHS OF TRANSFORMATION.
3.2.1 "/" Halve a Transformation Strength.
"\" Double a Transformation Strength.
18. 3.2.2 "λ<expr>" Set Translation Strength. TDEL.
"π<expr>" Set Rotation Strength. RDEL.
"$<expr>" Set Dilation Strength. DDEL.
3.2.3 -<digit>- Set Transform Strength Immediate.

18. 3.3 EUCLIDEAN SWITCHES.
3.3.1 "F" Step Frnae Switch Forunrds.
"βF" Step Fraae Switch Backwards.
19. 3.3.2 "O" Toggle Frame Origin Switch.
3.3.3 "βA" Step Axis Counter.
3.3.4 "V" Enable All Body Motions.
"‹V" Disable Frame Motion.
"βV" Disable Vertex Motion.
"‹V" Disable Parts notion.

19. 3.4 THE ITERATION COUNTER.
-<digit>- Accumulate Iteration Count.
-<return>- Reset Iteration Count to Zero.

20. 3.5 DIRECT EUCLIDEAN COMMANDS.
3.5.1 "U" Unmove a Body.
3.5.2 "X PLACE <X><Y><Z>"
3.5.3 "X ORIENT <PAN><TILT><SWING>"

21. SECTION 4. TOPOLOGICAL COMMANDS.

22. 4.1 FIVE EASY CREATIONS.
4.1.1 "●" Instant Cube.
4.1.2 "‹●" Instant Torus.
4.1.3 "X CUBE <ox> <DY> <DZ>"
4.1.4 "X CYLW <RADIUS> <N-SIMS> <DZ>"
4.1.5 "X BALL <RADIUS> <M-LONGITUDES> <N-LATITUDES>"

23. 4.2 COPY, KILL AND DUAL.
4.2.1 "C" COPY ·
4.2.2 "K" Kill.
4.2.3 "‹K" Edge-Vertex Kill.
4.7.4 "βD" Dual.

24. 4.3 WEEP COMMANDS.
4.3.1 "S" Sweep
4.3.2 "‹5" Pyramid.
4.3.3 "R" Rotation Solid Coapletlan.
4.3.4 "X SILHOUETTE <ZMIN><ZMAX>"
4.3.5 "X PRISMOID"

25. 1.4 THE EUI ER COMMANDS.
4.4.1 "V" Make Vertex Body.
1.4.2 "E" Make Edge and Vertex.
4.4.3 "M" Midpoint an Edge.
4.4.4 "J" Join Vort ices.
4.4.5 "G" Glue Faces.

26. 1.5 BODY INTERSECTION COMMANDS.
4.5.1 "‹∩" Body Intersection.
4.5.2 "‹U" Body Union.
4.5.3 "‹-" Body Subtract ton.
4.5.4 --- Evert Body Surface Orientation.
4.5.5 " |" Invert Edge Linear Orientation.
4.5.6 "$" Make Convex.
4.5.7 "‹$" Edgr Slurp.

27. 4.6 CUT COMMANDS.
"X cur <x> <Y><Z>" Cut Bodies, Faces and Edger.
"‹X CUT <x> <Y><Z>" Cut Faces and Edges.
"βX CUT <X><Y><Z>" Cut Edges.
"‹X CUT" Kill Temporaries (Uncut).

27. 4.7 PARTS TREE COMMANDS.
-a- Attach. "D" Detach.

PAGE SECTION CONTENTS

28. SECTION 5. INPUT/OUTPUT COMMANDS,
"I" input B3D file. "O" output 830 file.
"‹I" input CAM file. "‹O" output CAM file.
"βI" input CRE f ilm. "βO" output V2D file.
"‹I" input GEM file. ● eO- output GEM file.
29. "β2" take commands from GEO file.
● P- output PLT file of current display buffer.

30. SECTION 6. EDITOR CONTROL COMMANDS,
31. 6.1 PROGRAM CONTROL.
"‹E" Exit GEOMED. "‹W" Enter DDT.
31. 6.2 STACK COMMANDS.
6.2.1. "=" Swap 1st and 2nd.
"‹=" Swap 1st and 3rd.
"β=" Swap 1st and Last.
"‹=" Swap 2nd and 3rd.
6.2.2 "↓" Duplicate Push Stack Down.
"↑" Discard Pop Stack.
6.2.3 "‹↓" Rotate the Stack Down.
"‹↑" Rotate the Stack Up.
6.2.4 "●" Push Universe Node Into the Stack.
● Push the Now Camera into the Stack.
32. 6.3 BODY NAMING AND RETRIEVING.
6.3.1 "N" Name Body.
6.3.2 "B" Body Get.
6.3.3 "‹B" Body Retrieval by Name or by Numeral.
32. 6.4 FACE AND VERTEX PERIMETER TRAVELING.
. . Fetch Counter Clockwise.
"," Fetch Clockwise.
"+" Fetch Other.
"‹V<n>" Fetch Nth Vertex of a Body.
33. 6.5 LINK FOLLOWING COMMANDS.
● <- NFACE Link. ">" PFACE Link.
"ζ" NEED Link. "ζ" PED Link.
● -· NVT Link. "^" PVT Link.
"n" DAD link. "U" SON Link.
"c" BRO Link. "⊃" SIS Link.
"←" ALT Link. "→" ALT2 Link.
● ,- CW Link. "." CCY Link.

35. SECTION 7. DISPLAY CONTROL COMMANDS.
36. 7.1 STATUS DISPLAY.
7.1.1 "≡" Status Display Toggle.
"‹≡" Toggle Now Window's Darken Bit.
"≡" Clear the page printer.
7.1.2 "∂" Datum Display Toggle.
"‹∂" Datum Format Toggle.
7.1.3 "L" Toggle FEV Lights Switch.
"‹L" Tossle Body Lights Switch.
"βL" Toggle Frame Lights Switch.
37. 7.2 HELP DISPLAY.
"?" Information Prefix.
"H" Help.
"‹H" Unhelp.
37. 7.3 DISPLAY MODES.
"space" Display Refresh. GEODPY.
"ALT" GEODPY uith hidden lines ● liminated.
"‹ALT" GEODPY uith back faces eliminated.
"βALT" GEODPY everything visible.
● tALT- GEOOPY with OCCULT diagonostics.
"_" Sticky Alt.
"‹_" St icky ‹ALT.
"β_" St icky βALT.
"‹_" St icky ‹ALT.
37. 7.4 EDGE DISPLAY BITS.
7.4.1 "‹D" Darken.
"‹D" Undarken.
7.4.2 ● X NSHARP"
38. 7.5 FACE COLORING.
● X COLOR <n>RED <n>GRN <n>BLU <n>ALB"
38. 7.6 CAMERA COMMANDS.
"‹C" Push now camera Into the stack.
"βC" Make neu camera.
"β}" Step now camera forunrds.
"β{" Step nou camera backwards.
"‹F" Set Camera's Focal Plane distance.
39. 7.7 WORLDS AND WINDOWS.
"W" flake Uindou In Now Display Ring.
● Make Window in New Display Ring.
"βW" Hake New Uorld.
"{" and "}" Step Now Display.
"‹{" and "‹}" Step Now Uorld.
33. 7.8 IMAGE RINGS.
"‹I" Make a Simulated laage Body.
● and "+" Step Perceived Image Ring.
"‹+" and ● e- Step Predicted Image Ring.

SECTION 1

PRIMER

1.1 Purpose.

GEOMED is for making drawings, synthetic video images, and 3-D geometric models for the sake of computer vision. This document explains the external command language of GEOMED as an interactive drawing program.

1.2 Running GEOMED.

The system copy of GEOMED is started by typing "R GEOMED" <carriage return> at a CRT vector display console, rather than at a video console. The program will display a rectangle, type an asterisk and await single character commands. Typing additional carriage returns will yield more asterisks showing that GEOMED is alive and listening. At Stanford the CRT vector display consoles were made by Information International Inc. and are called "III" (pronouced "triple ● ye"), while the video display consoles wore made by Data Disc and are called "Data Discs"; inspite of local jargon I wish to refer to the two kinds of display devices by generic names: CRT or vector display and TV or video display.

1.3 Instant Cube and Translation

Typing the character "∞", calls a macro that makes a cube. Adjust the pots on the display console so that the cube looks right. The cube may be moved about by typing the character ":;)(*-". The character ":" moves the cube right (positive x axis); the character ";" moves the cube left (negative x axis); the character ")" moves the cube up (positive y axis); the charactor "(" moves the cube down (negative y axis); the charactor "*" moves the cube nearer (positive z axis); and the character "-" moves the cube away (negative z axis). Clearly these command characters are not mnemonics, they were chosen because they do not require the TOP key and are conveniently grouped in pairs under one's right hand. The distance the cube is moved on each command is called the strength of translation or TDEL. The value of TDEL is displayed in the upper right corner of the screen and is initially one foot, The strength of translation is halved by typing the command character "/" or doubled by typing "\". (The unit of length can be switched to meters or centimeters by the "X METER" and "X CM" commands respectively).

1.4 World Frame of Reference.

The direction the cube moved was with respect to the world frame of reference which is a right handed coordinate system. The initial camera is positioned looking down the Z axis towards the world origin. The world origin is in the center of the display screen a simulated sixteen feet away; and the positive world X axis is to your right; and the positive world Y axis is upwards, Coordinate axes can be displayed by toggling the "βL" switch command.

1.5 Control, Meta, and Meta-Control Keys.

Notice that the Stanford A.I. keyboard has four "shift" keys labeled SHIFT, TOP, META & CONTROL. SHIFT only determines whether an alphabetic letter is upper or lower case; GEOMED converts lower case letters into upper so that the SHIFT key has no effect. The TOP key allows two ASCII character codes to be on each physical key; this doubling up on the physical keys is not important to GEOMED since "TOP-E" can always be referred to as "@". Finally, of great importance, the CONTROL and META keys add two extra bits to the 7-bit ASCII code, so that a 9-bit charactor is formed. In this document the characters "∝", "β" and "€" will be used as prefix abbreviations for CONTROL, META and META-CONTROL keying of a character. Furthermore, the command scanner accepts the characters "∝", "β" and "€" as prefix modifiers that will add the appropriate control and meta bits to the next character typed. The fourth (and final) prefix character is the question mark. Typing a "?" followed by any character will cause a one line reminder to be printed of what commands are invoked by that character.

1.6 Rotation.

The rotation command characters are the same as for translation except you must hold the CONTROL key down or prefix the command with an "∝". Try rotating the instant cube about the X-axis with "∝:". The positive direction of rotation is counter-clockwise. The negative direction of rotation is clockwise. The strength of rotation is named RDEL. The value of RDEL is displayed in two formats in the upper right hand corner of the screen: as a pi fraction and in degrees, minutes, seconds. The initial value of RDEL is n/4, 45 degrees. Analogous to translation, the strength of rotation is halved or doubled by "∝/" and "∝\" respectively. Since a sequence of rotations is common, there is a way to make the CTRL key be sticky. The Euclidean transformation default is translation world frame; but after typing the "@" command character, the Euclidean default is rotation world frame. Translation default is restored by typing "!". The state of the Euclidean transformation default is also displayed in the status in the upper right hand corner of the screen. (The "≡" command disables the status display; "≡" toggles).

1.7 The Stack.

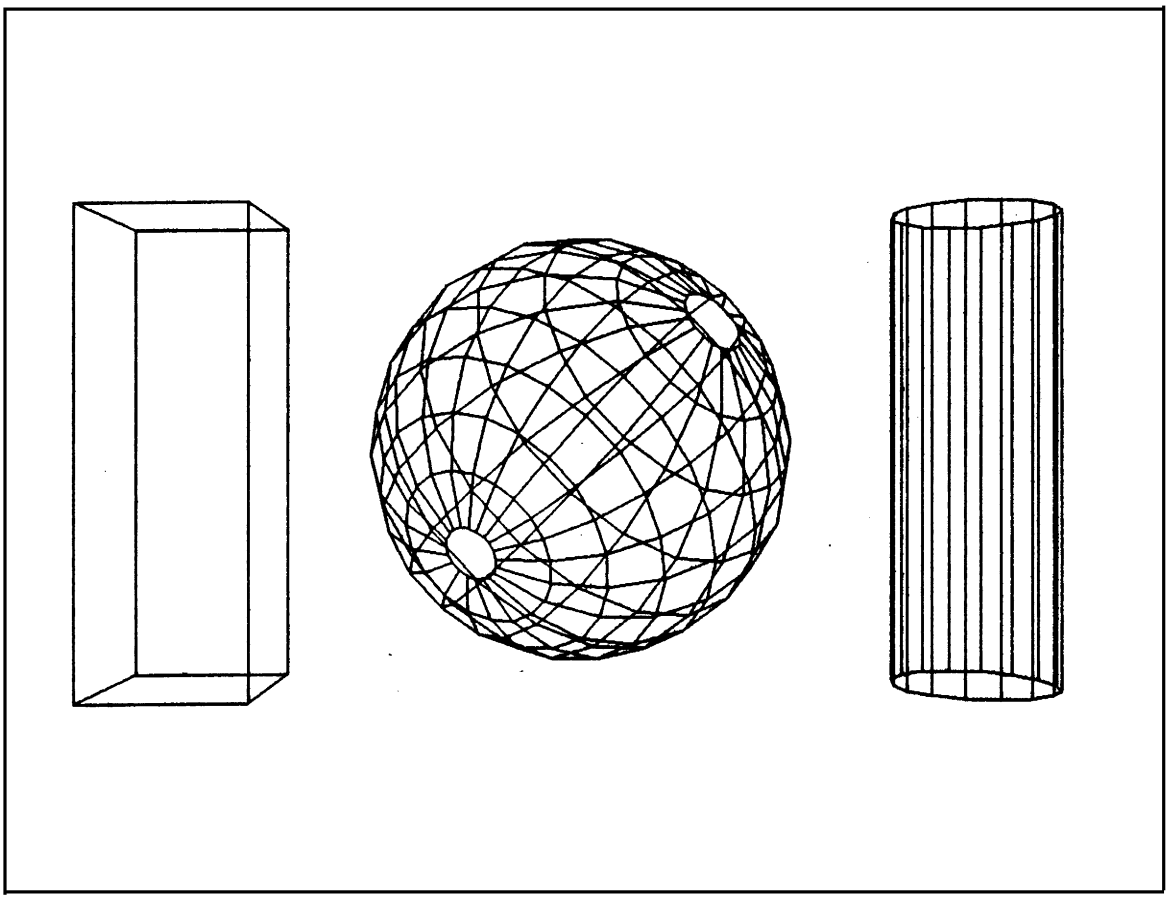
GEOMED commands take their arguments from and leave their results in a push down stack of bodies, faces, edges and vertices. The contents of the stack are displayed on the left hand side of the display screen. Every kind of node has a default print identifier, for the first body created a 'B1' is displayed in the stack. Making a second cube will push a 'B2' into the stack. To retrieve 'B1' use the "↑" stack pop command or the "∝↑" stack rotate up or "∝↓" stack rotate down or "↔" the swap top of stack command, The "↓" command pushs the stack down and places in the new top the entity that was previously top of stack.

1.8 The Extended Command Scanner.

The X-commands are executed by typing "X" followed by the first three letters of a mnemonic (further letters are ignored) followed by any necessary arguments. The command is terminated by • RETURN. The arguments may be separated by any reasonable **delimiters: space,** comma, etc. **The** following three "X" commands provide easy creation of simple polyhedra:

X CUBE <DX **width>,<DY height>,<DZ** depth>
X BALL **<radius>,<M longitudes>,<N** latitudes>
X CYLN **<radius>,<N sides>,<DZ length>**

The CUBE command makes a right rectangular **prism** with width, height and depth as given, **The** BALL command makes **a** polyhedron that approximates a **sphere. The** CYLN command **makes** a **right** prism that **approximates** a circular cylinder.



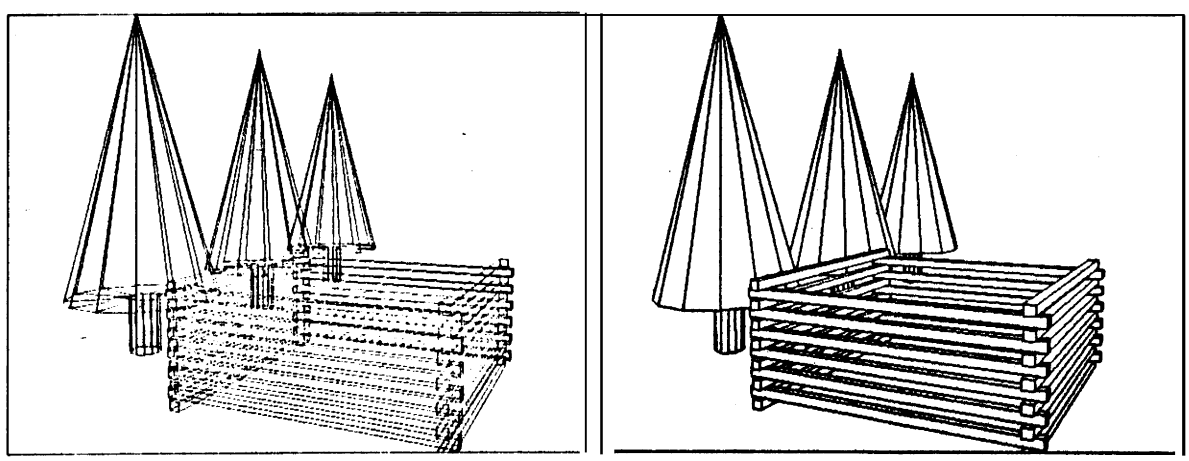
1.9 The Face, Edge and Vertex Structures.

A polyhedron consists of three circular double linked lists (called rings); there is a ring of faces, a ring of edges and a ring of vertices. The head of each ring is the body node. The rings can be traveled by using the commands "<" and ">" to go forwards and back through the face ring, the commands "≤" and "≥" for the edge ring, and the commands "v" and "^" for the vertex ring. Notice that when a face, edge or vertex node is in the top of the stack the corresponding entity is intensified in the drawing.

In addition to the three body rings, the faces, edges and vertices have pointers among themselves. Faces and vertices always have a pointer to one of their edges called the "prime edge" or "PED". The PED of a face or vertex in the top of the stack may be retrieved by using the "≥" command. On the other hand, edges have eight pointers; each edges points at its two faces, its two vertices and its four neighboring edges (clockwise and counterclockwise) in each of its two faces. These last four pointers are the so called "wings" for which the representation is named. The wings are *oriented so that a traverse of a face (or vertex) perimeter will appear oriented counter clockwise as viewed from the exterior side of the polyhedral surface. Perimeter traveling is explained in subsection 6.4; it suffices now only to be aware that there exist oriented perimeters that can be easily traveled. The contents of the non-wing pointers of an edge can be brought into the stack by the "v" and "A" commands for the two vertices of the edge, and by the "<" and ">" commands for the two faces of the edge.

1.10 Hidden Line Elimination.

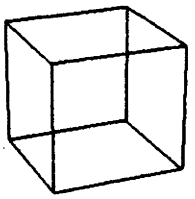
Typing ALT-MODE causes your drawing to be displayed with its hidden lines eliminated; as is illustrated by the two pictures of a log cabin in the woods. To stay in hidden line display mode use the "_" underbar command; the "β_" returns the display mode default to display all the lines. The hidden line eliminator, named OCCULT, does not display concave faces correctly; and for the sake of speed it does not check for them. The faces of a polyhedra can be forced convex by applying the "§" command to a body or to a face.



1.11 Making a Cube Explicitly.

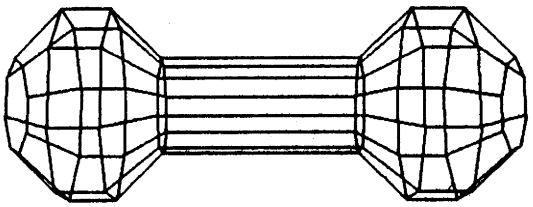
After **"∞"** and **"X** CUBE", a third way to draw a cube will be given to illustrate the primitive topology commands. Starting with a fresh copy of **GEOMED,** type the command letters in the **second** column to get the advertised results:

1.	v	Vertex body creation.
2.	**:)**	Position the vertex into the first quadrant.
3.	s;;	Make an edge and vertex and move left.
4.	**S((**	**Edge** vertex down.
5.	s::	Edge vertex right.
6.	**J↑**	Join the first and last vertices of the wire.
7.	*****	Pull **the** face lamina towards you.
8.	**S**	Sweep the square face into a very thin solid.
9.	**--↑**	Move the face away from you giving the cube depth.
10.	**@/):!**	Rotate the cube to a more familiar position.



Three commands that have not been mentioned yet are: **"V"** vertex body creation, **"S"** sweep and **"J"** join two vertices with a new edge. The **"V"** command takes no arguments and pushs a new body, face and vertex into the stack. This degenerate point polyhedron satisfies the Euler equation **F-E+V=2** (1 **-0+1=2)** and is always placed at the world origin. The first three sweep commands in the **example** sweep a vertex polyhedron into a wire polyhedron by adding a now **vertex** and a new edge (incidentally preserving Euler's equation as do all **GEOMED** commands). The wire-sweep **takes** a vertex argument from the stack, and returns the **new vertex to the stack.** The new vertex has the same locus as the argument vertex. After three sides of a square have been formed, the **"J" command** creates a new face and a new edge between the first and last vertex of the wire face. The wire-join **case** of **the "J"** command takes a vertex argument and identifies it as the end of a wire and **know6 enough to fetch** the other end of that wire, as its second argument.

1.12 Making a Solid of Rotation.



A solid of rotation can be **made** by sweaping a wire face into a sheet (of rotation) and by closing the sheet with the **"R"** command (Rotation Completion. For example a dumbbell shaped object is formed by typing the following commands:

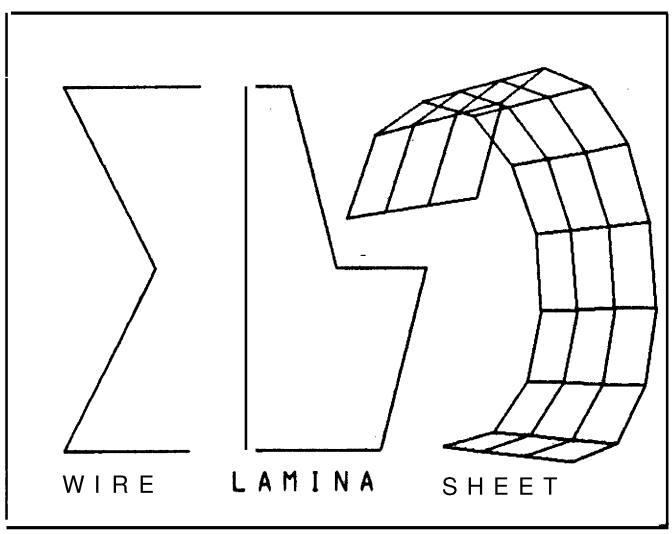
1.	**V(((((:**	Get Vertex **to starting position.**
2.	**S:)S)S;)S))))**	Lower bell.
3.	**S:)S)S;)↑**	Upper Bell.
4.	**7@S(R!↑**	Rotate wire about Y-axis and complete solid.

1.13 Wire, Lamina, Sheet, Euler and Solid Polyhedra.

The entities being created and altered by GEOMED are representations of polyhedra. As already mentioned, a polyhedron is comprised of a single body node which is connected to a ring of face nodes, a ring of edge nodes and a ring of vertex nodes. The nodes are fixed size et twelve words per node and all the topological, geometric and photometric data fits in the nodes. In particular, there are no variable length structures such as face perimeter lists; thus the core space required for a polyhedron is **(B+F+E+V)*12** words of memory.

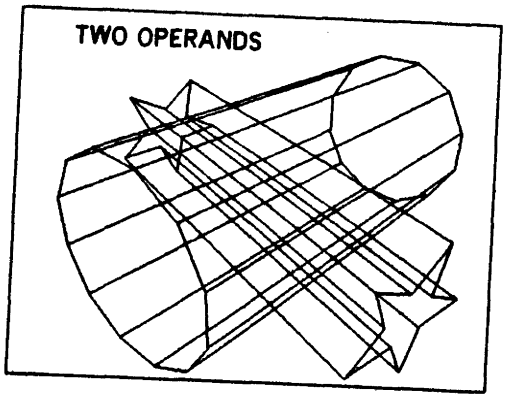
In GEOMED, several classes of polyhedra are handled. The most general class allowed are Euler polyhedra which satisfy the relation **F-E+V=2*B-2*H;** where **B,** F, **E,** and V are the number of bodies, faces, edges and vertices in the model and H is the genus of the surface of the polyhedron (which is the same as the number of handles or holes). The editor is based on primitives which maintain the Euler relation so that a polyhedron that violates the Euler relation can not be generated. The most **restricted** polyderon class are called solid convex-face polyhedra; these polyhedra may be concave but all their faces must be convex and **planar;** all their vertices and faces **must** have a valence (number of edges) of three or more; and no edge may intersect a face or vertex' to which it is not topologicelly linked. Solid convex-faced polyhedra are required for efficient and accurate operation of the hidden **line eliminator.**

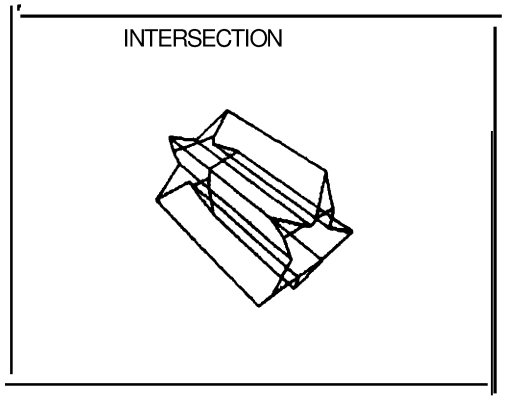
Intermediate between **Eulerian** and solid, are the wire, lamina and sheet polyhedra which can be made by sweep commands. A wire polyhedron is Eulerian, one faced with linear topology; a lemina is a two faced polyhedron with no interior edges or dangling wire; and finally a sheet is an array of Irmine. With the exception of sweeping ruled surfaces, commands for mrnipuleting and folding shoots are non-existent.

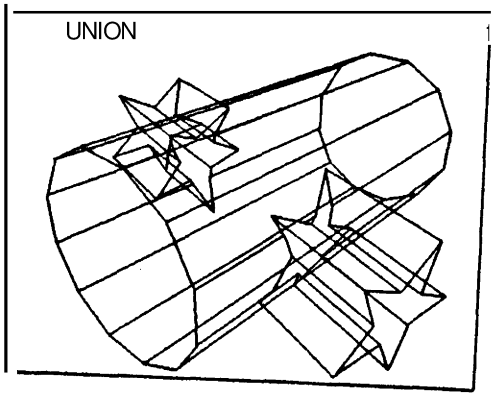


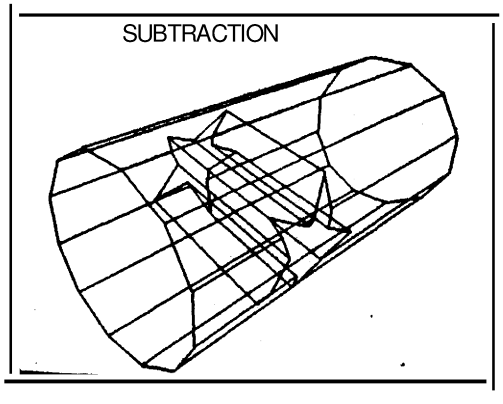
1.14 Body Intersection, Union and Subtraction,

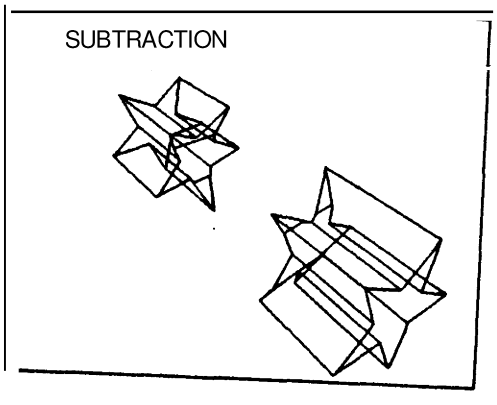
Given two solid polyhedra in the top of the stack, the polyhedron (or polyhedra) enclosing the space of their interesection, union or set difference can be obtained by typing the "∝∩", "∝U" or "∝-" commands respectively, As illustrated in the lower right figure, a polyhedral model of a plastic toy horse can be made by applying the intersection command to cones of silhouette contour6 of video images taken from two camera positions.

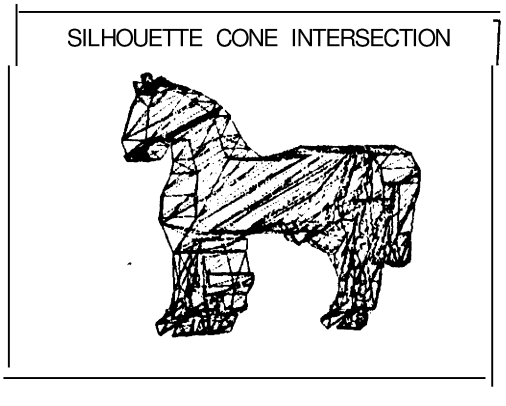










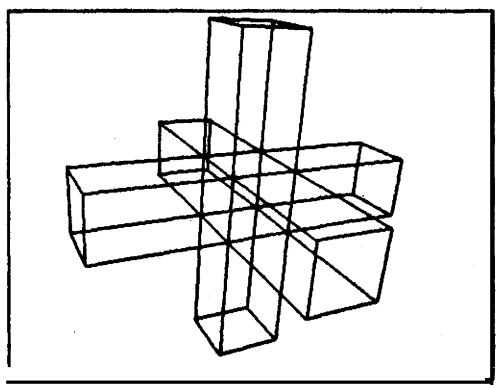


EXAMPLES

The ● xrmplas can be used as exercises in t wo ways: the energetic reader can glance at the figures and attempt to reproduce the results on hir own; the casual reader can copy the answers into the computer and follow the action a step at a time. In ● ithor ● vent, one should look at sections 3 through 7 for commands that have not yet boon ● xplrinod.

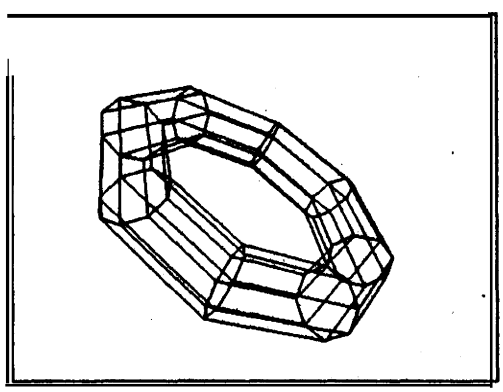
2.1 Jack.

```
XCUB 2 2 2
\\FF >S* >S* >S* >S* >S* >S* B
```



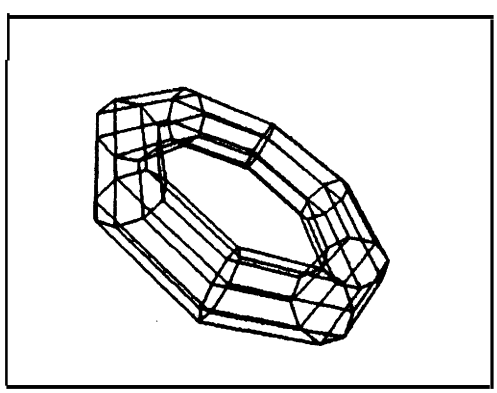
2.2.1 Torus.

V:@	First Vortex.
S*S*S*	Seven more vertices.
S*S*S*S*	
Jt	Form Lamina.
!\\:	Position the lamina.
@S)S)S)S)	Sweep the face around the Y-axis.
S)S)S)S)	
↓>Gt	Glue the ends of the torus together.



2.2.2 Torus using iterated sweeps.

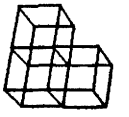
V:7@S*J	Make lamina.
!t8/:\@S)	Swoop lamina into ring.
↓>Gt!	Glue the ends.



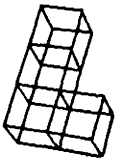
2.3 Soma Cubes

An introduction to both Some Cubes and the Platonic solids can be found in Martin Gardner's Second Book of Mathematical Puzzles and Diversions, Simon and Schuster, New York, 1961. The objects are shown rotationally displaced from the position in which they are constructed. If you do one object after another, then the "FF" should not be reported.

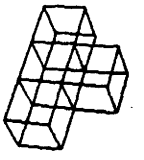
FFXCUB 1 1 1
>S*>>S*BNSOMA 1



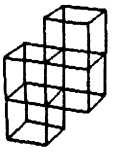
FFXCUB 1 1 1
>S*S*>>S*BNSOMA2



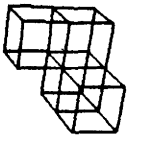
FFXCUB 1 1 1
>S* >S* >S*BNSOMA3



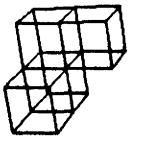
FFXCUB 1 1 1
<S* <S*↓¿.•↑ S*BNSOMA4



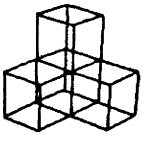
FFXCUB 1 1 1
>>S*>>>>S*↓¿.•↑S*BNSOMA5



FFXCUB 1 1 1
>>S*>>>>S*↓¿.•↑S*€;¬BNSOMA6



FFXCUB 1 1 1
<S*<<<S*<S*BNSOMA7



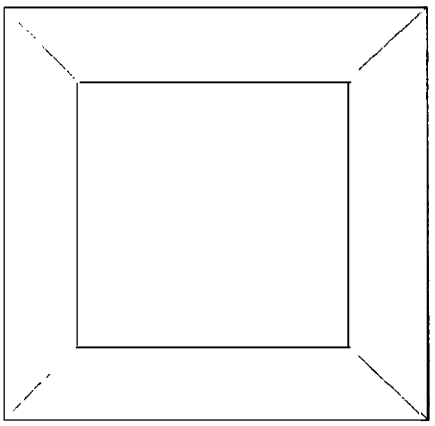
2.4 The Platonic Solids.

Regular Tetrahedron, edge length of SQRT(2).

```
XCUB 1 1 1'
^↓^^↓B v↓vv        Push opposing vertices into the stack,
SK↑SK↑SK↑KB        Lop off the opposing vertices.
```

Another Regular Tetrahedron, edge length of SQRT(3).

```
XCY L 6 3 6*1.414  Three sided cylinder
>KB                Pyramid one of the ends.
```



Regular Hexahedron, edge length of 1.

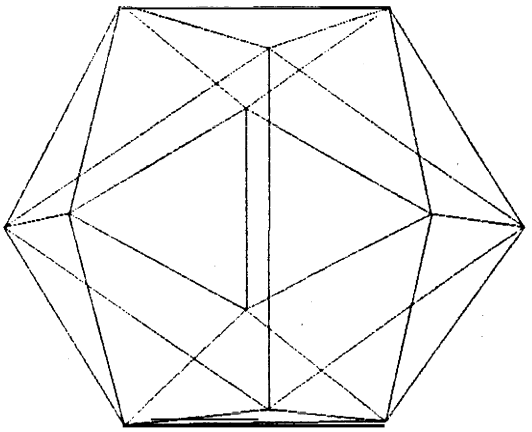
```
XCUB 1 1 1         Very easy.
```

Regular Octahedron, edge length of SQRT(1/2).

```
XCUB 1 1 1
βD                 Face-Vertex Dual of a cube
```

Regular Icosahedron, edge length of 6*SIN(π/5).

```
Xcyl 3 5 0         Pentagonal lamina
λ.507*3
>*XPRISM           Prismoid face sweep
--ππ/5
∝*λ.6 18x3         Reciprocal of golden mean.
↓>∝s*↑∝s-B
```



Regular Dodecahedron, edge length of 6/PHI.

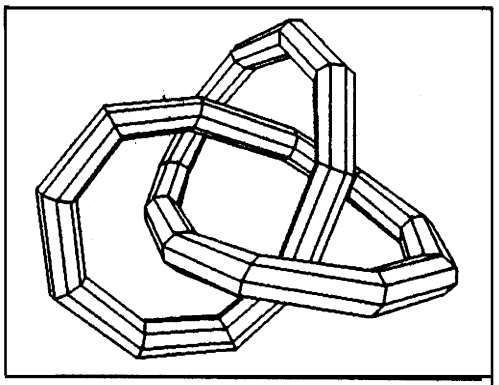
(after Euclid's classical construction:
Elements, Book XIII, Proposition 17).

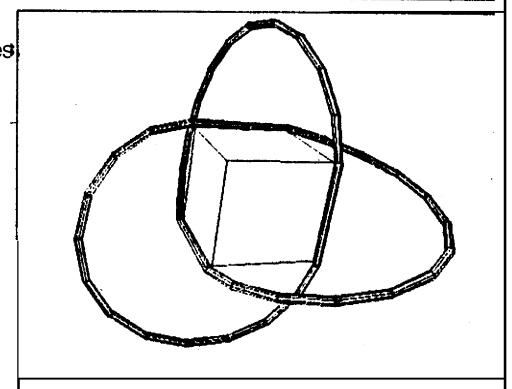
```
XCUB 6 6 6         Start with a cube.
β5λ3               Midpoint DDEL, Midedge TDEL.
>>S:↔B<<<<<        Edge sweeps to midface.
S:↔<S(↔↓M↔
<<↓M↔<<S)∝↑
↓M↔<M∝↑Jβ↔J        Join the two pairs of midpoints
X6 1.8034          Reciprocal of the golden mean, PHI.
λ0.309017*6              SQRT(PHI↑2 - 1/4 - (1-PHI/2)↑2)
β;)↑β-;↑β(-↑        Shrink and translate the six new edges.
β(*↑β-:↑β;(B
```

2.5 A Knot.

The knot was developed by visualizing the connection of three non-parallel and non-coplanar edges of a **cube**, as in the second knot **figure**.

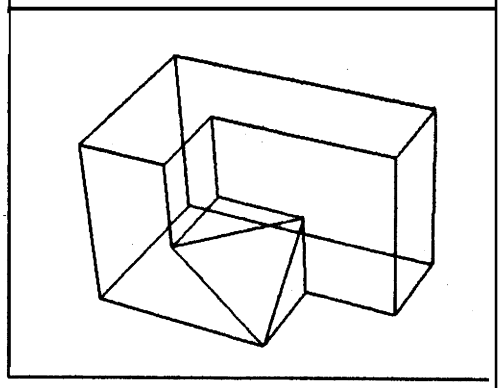
```
XCYL 1 8 6          Straight section.
X6
Q;/->>λ 1
S∝))S∝))S∝))        Three quarters of a turn.
S∝))S∝))S∝))Bλ3
*:(@9C              Symetry on a cube
:-C-(!
>↓>∝↓               Fetch three pairs of faces.
>↓>∝↓
>↓>∝↓
G↑G↑G               Glue the pairs of faces.
```





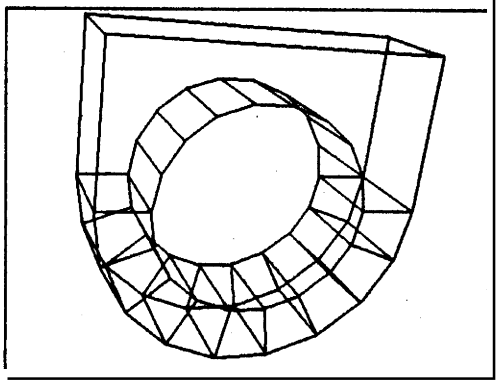
2.6 An Odd Shaped Block (illustrating kill commands).

```
XCUB 3 2 1
%100/3
≥↓≥≥
M↔↓>↔M
E(E:↔↑J<S*B
≥↓≥↓↓v..
↔K↑
↔K↑K↑↓v↔K↑KB
∝V12
SKB$
```



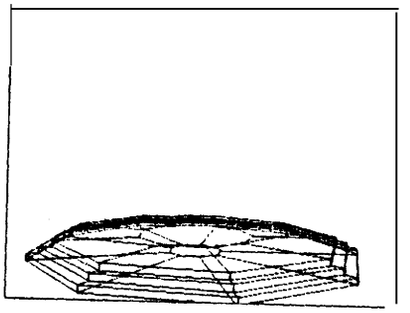
2.7 **Yet** Another Torus.

```
V\:ππ/9
9@S*!/S;S)))
S::::::S(((J↑
-S**>>↓>>>>9@S*!
G↑$
```



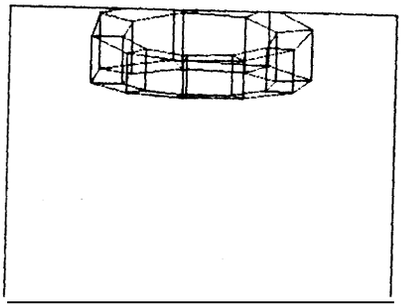
2.8 The Pulgas Water Temple.

Ten miles north of Stanford, the Pulgas Water Temple marks the termination of an aqueduct that brings water from the Sierra Nevada to San Francisco. This example illustrates how a complicated object is built up of simple polyhedra.



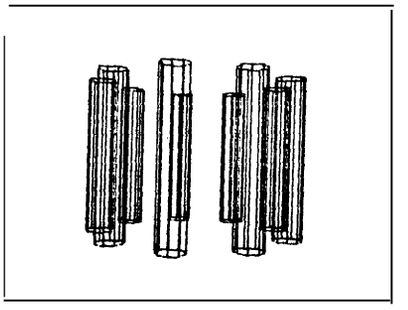
Base of Temple.

```
V:λ5                    Foundation of temple.
S:λ0.2
s )S;;S)S;;S)Sλ3        Jaggies forming steps.
;↑π 40'                 Set RDEL to 40 degrees.
8@S)R↑¬λ3.6 1           Sweep out a solid of rotation.
!(
```



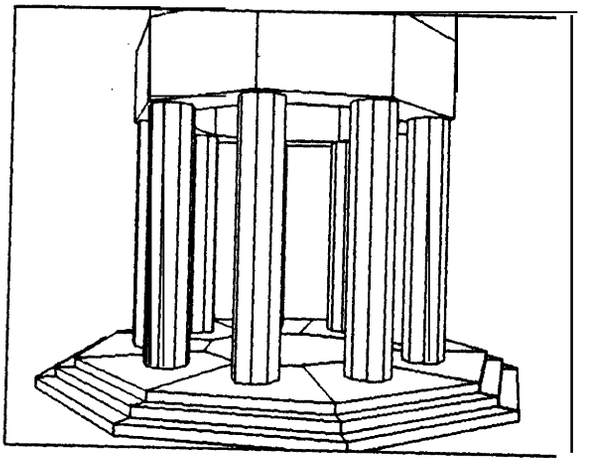
Top of Temple.

```
!Vλ0.6
:S)))S;;S(((J           Make rectangular lamina
↑λ3.5
:π40'
9@S)↓>G↑/)              Sweep lamina.
!λ2.72
)
```



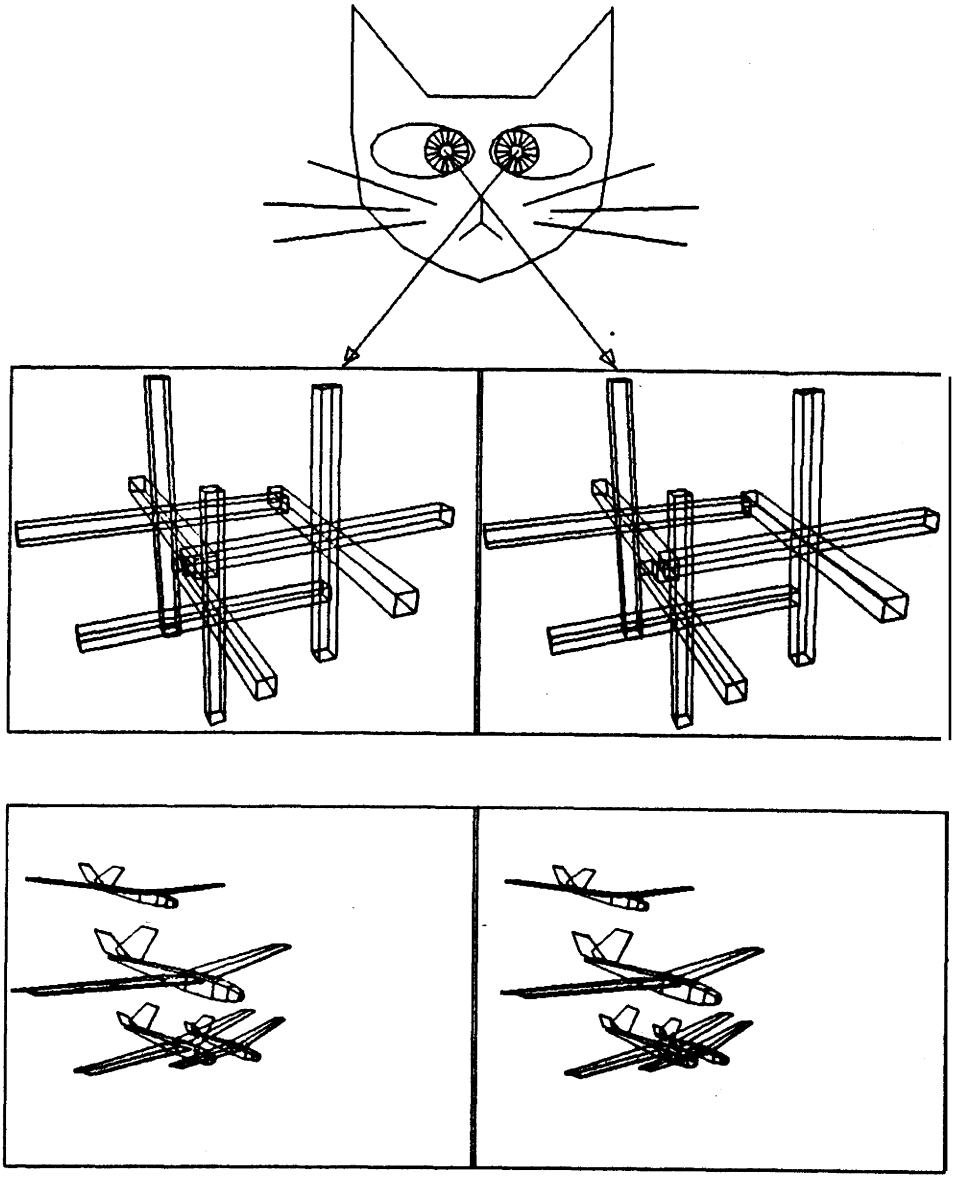
Columns of Temple,

```
XCYL 0.5 9 6            Make first column,
@9:!λ3.5                posit ion column.
>Aβ9β(<:π20'            Taper the top of the column.
Q@)\
C)C)C)C)                Make eight copies of the column,
C)C)C)C)
```



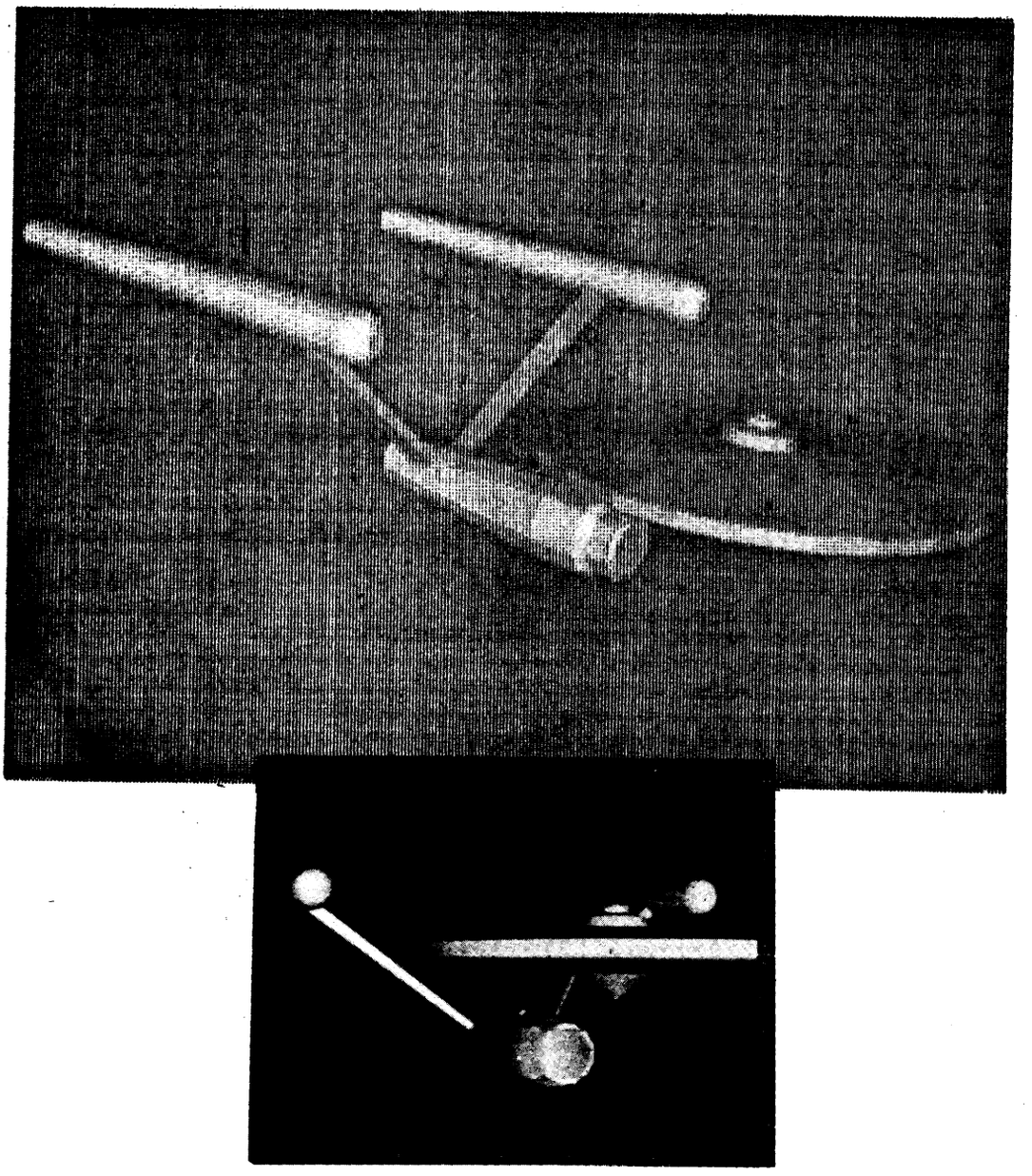
2.9 Cross-Eyed Stereo Pairs.

Models may be viewed as a stereo pair by creating an additional camera and display window; this is done automatically by using the "/8∞" macro. Although stereo pair prisms are availabla and can be used to view the display screen, the stereo macro command swaps the usual left and right images so that the depth may be seen by looking at the screen cross-eyed as is illustrated by the Siamese cat below. That is in order to see stereo depth, look at the display cross-•yod so that you can see three windows; next, concentrate on focusing on the middle window. With practice the displayed objects cm be seen in depth at a glance with no special equipment, which is a great help in positioning things.



2.10 Video Synthesis.

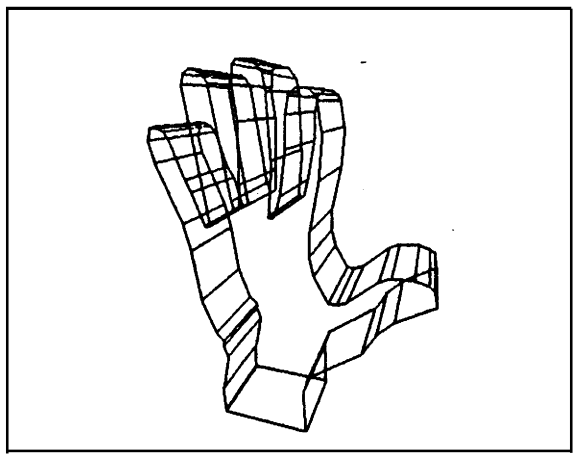
The "/BO" command will do a hidden line elimination and output a 2-D vector based image format, V2D file, which in turn is suitable input to a program called MKVID. MKVID creates, shades, colors, and dejaggies television pictures. The only online video hardcopy device is the Xerox Graphics Printer. A quick, but low quality XGP video image (such as the large one below of the star ship Enterprise) is obtained by typing "R XIP;LO,0;*<FILENAME>" or "R XAP;LO,0;*<FILENAME>" to the monitor; the smaller picture was made by taking a Polotiod picture of a video display screen. XIP and XAP are underground Xerox document formatting programs which will someday be superceded by the establishment supported program named PUB. XIP is three times faster than XAP, but uses twice as much core. This document was formated and printed using XIP.



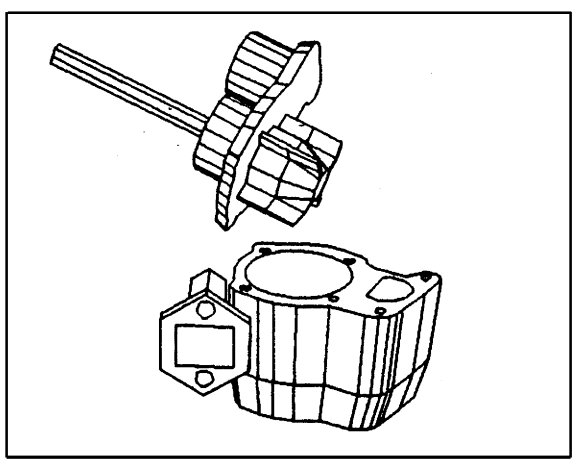
2.1 1 Video Derived Polyhedra.

One way to make a polyhedron from a video image is to sweep the silhouette of an intensity contour from the program CRE (a video contouring program). The CRE commands **"T"** for **take, "C"** for contour and **"∝O"** for output contoured image will yield a disk file **suitable** for input to GEOMED. Using GEOMED commands: **"βI"** will input a CRE file and create a **face,** edge, vortex data structure on **the** now camera node that corresponds to the contoured image;

0.	TC 1 **0∝OHAND.CRE**	CRE Commands to take, contour and output.
1.	**βIHAND.CRE**	GEOMED input from CRE file.
2.	**∝C...**	Locate a suitable contour body node.
3.	**XSIL <ZMIN> <ZMAX>**	**Sweep** out silhouette polyhedron.



The CRE program is documented in Stanford A.I. Memo **#199,** titled 'Image Contouring and Comparing', by Baumgart. A more elaborate example of video assisted drawing is demonstrated by the water pump below; the shape of the base and the holes in the **base** were derived from a **video image.**



SECTION 3

GEOMETRIC COMMANDS

3.1 EUCLIDEAN TRANSFORMATION KEYS.

Translation, ∝ Rotation, β Dilation, € Reflection.

";" Transform Minus X-Axis.	":" ...Plus X-Axis.
"(" Transform Minus Y-Axis.	")" ...Plus Y-Axis.
"-" Transform Minus Z-Axis.	"*" ...Plus Z-Axis.
"!" Translation Default.	"@" Rotation Default.

3.2 STRENGTHS OF TRANSFORMATION.

32.1 W/W Halve a Transformation Strength.
W\" Double a Transformation Strength.

3.2.2 "λ<expr>" Set Translation Strength, **TDEL.**
"π<expr>" Set Rotation Strength, RDEL.
"%<expr>" Set Dilation Strength, DDEL.

3.2.3 "<digit>" Set Transform Strength Immediate

3.3 EUCLIDEAN SWITCHES.

3.3.1 "F" Step Frame Switch Forwards.
"βF" Step Frame Switch Backwards.
3.3.2 "Q" Toggle Frame Origin Switch.
3.3.3 "βA" Step Axis Counter.
3.3.4 "∀" Enable All Body Motions.
"∝∀" Disable Frame Motion.
"β∀" Disable Vertex Motion.
"€∀" Disable Parts Motion.

3.4 THE ITERATION COUNTER.

"<digit>" **Accumulate** Iteration Count.
"<return>" Reset Iteration Count to Zero.

3.5 DIRECT EUCLIDEAN COMMANDS,

3.5.1 "U" Unmove a Body.
3.5.2 "X PLACE <X> <Y> <Z>"
3.5.3 "X ORIENT <PAN> <TILT> <SWING>"

3.1 EUCLIDEAN TRANSFORMATIONS.

Translation, ∝ Rotation, β Dilation, € Reflection.
";" Transform Minus X-Axis. ":" ...Plus X-Axis.
"(" Transform Minus Y-Axis. ")" ...Plus Y-Axis.
"-" Transform Minus Z-Axis. "*" ...Plus Z-Axis.
"!" Translation Default. "@" Rotation Default.

The Euclidean geometric transformations are translation, rotation, dilation and reflection. The entity in the top of the stack is transformed by typing one of the six characters: colon, semicolon, loft parenthesis, right parenthesis, minus sign or asterisk. The characters colon, left parenthesis and minus sign transform the object in the negative direction with respect to the X, the Y, and the Z axes respectively. The characters semicolon, right parenthesis and asterisk transform the object in the positive direction with respect to the X, the Y, and the Z axes respectively. The particular transformation is selected by keying the control bits: none, control, meta and meta-control which respectively select translation, rotation, dilation and reflection. Finally, the no-control-bits case can be forced to be rotation by the "@" rotation default command, or translation by the "!" translation default command.

Translation moves the top entity in the stack in the direction specified by one unit of translation strength, TDEL. Rotation rotates the entity about the axis specified by one unit of rotational strength, RDEL. Positive rotations are counter clockwise and negative rotations are clockwise.

Dilations and reflections refer to a three axis count selector. State 1 causes dilation (reflection) to be done on the specified axis, dilation state 2 causes dilation (reflection) on the two axes not indicated, and state 3 causes dilation (relfection) on all the axes. The axis count selector is advanced by typing "βA". The state of the selector is displayed as a digit 1, 2 or 3 just to the right of the dilation strength's per cent sign in the editor status in the upper right of the display screen.

When the dilation strength,. DDEL, is less than 1007, a positive dilation will scale the entity by 1/DDEL and a negative dilation will scale the entity by DDEL. Positive and negative keying8 make no difference in the execution of a reflection.

3.2 STRENGTHS OF TRANSFORMATION.

3.2.1 "/" Halve a Transformation Strength.
"\" Double a Transformation Strength.

The strength of a Euclidean transformation can be halved or doubled by keying the transform's control bits and by striking slash or back slash respectively.

3.2.2 **"λ<real** expression>" Set Translation Strength, TDEL.
"π<real expression>" Set Rotation Strength, RDEL.
"%<real expression>" Set Dilation Strength, DDEL.

The strengths of the Euclidean transformations can be entered numerically by typing **"λ", "π"** or **"%"** followed by an arithmetic expression of numerical constants. The simple expression scanner can **take "+", "-", "*", "/"** and parenthesis in the usual precedence order; the scanner also evaluates the pi character, **"π",** to 3.1415927; numbers suffixed with the inch mark double quote **(")** are divided by twelve; numbers suffixed with a left single quote **(')** are multiplied by **1.745329E-2** which converts degrees into radians.

3.2.3 "<digits>" Set Transform Strength Immediate.

The strength of a Euclidean transformation can be set by keying the Transform's control bits and by striking a digit from zero to nine. Keying **"€<digit>"** sets the strength of translation; a digit without **meta-control** bits contributes to the iteration count. For Translation: **"€0"** sets TDEL to **1/16** of a foot, **"€4"** sets TDEL to one Foot, and **"€9"** sets **TDEL** to 32 Feet. That is, **€<digit>** sets TDEL to **2.0†(<digit>-4)** feet. For Rotation: **"∝9"** sets RDEL to n/2, **"∝8"** sets RDEL to n/4, and so on by **halves.** For Dilation: **"β1"** sets DDEL to 1 **0%, "β2"** sets DDEL to **20%** and so on.

3.3 EUCLIDEAN SWITCHES.

3.3.1 **"F"** Step frame switch forwards.
"βF" Step frame switch backwards.

There are four frames of reference: world frame, body frame, relative frame and camera frame. The world frame is that in which all coordinates are stored; the world frame is the natural direct frame of reference, all the other frames being represented in world frame coordinates by a frame node which contains the origin location and axes orientation of secondary frames. In particular, each body and camera has a frame node, which determine a camera or a body frame of reference that translates and rotates when Euclidean transformations are applied to that body or camera. The relative frame mode is a catch all; when a body is in the top of the stack, its relative frame is that of the body of which it is a part; when a face is in the top of the stack, its relative frame is a special face frame with Z-axis parallel to the face's outward pointing normal. The **"F"** command steps the frame switch selector forwards, **"βF"** steps the frame switch selector backwards. The state of **the** frame switch selector is displayed in the status in the upper right of the display screen.

3.3.2 **"Q" Toggle** Frame Origin Switch.

Euclidean transformations in world frame can be done with **repect** to **the** world origin or with **repect** to the origin of the entity being **transformed** depending on **the state** of **the** FRMORG switch. In particular, FRMORG **affects** world frame rotations: FRMORG true **causes** rotation about **a world axis thru** the world origin; FRMORG false causes rotation about an axis **parallel** to a world axis, but passing **thru** the body origin. The FRMORG switch is flipped by the **"Q"** command. **The** state of FRMORG is **indicated** by an asterisk in the status display; the asterisk is present **when** FRMORG is true **and absent when** FRMORG is false.

3.3.3 **"βA"** Step Axis Counter.

The three-state switch named AXECNT affects dilations and reflections. **State #1** indicates dilation (reflection) only on the specified axis. State **#2** indicates dilation (reflection) on the two **axes** not specified. State **#3** indicates dilation (reflection) on all three axes. **The** state of the AXECNT switch is indicated by a digit to the right of the dilation strength's_percent sign **in the status display. AXECNT is cycled forward** by the **"βA"** command.

3.3.4 **"∀"** Enable All Body Motions.
"∝∀" Disable Frame Motion.
"β∀" Disable Vertex Motion.
"∈∀" Disable Parts Motion.

Applying a Euclidean transformation to a body means applying the tranform to the vertices, frame and parts of that body. Each of these three phases of body transformation can be individually disabled by the **"∀"** command with the appropriate combination of **meta-control** bits. With no **meta-control** bits, the **"∀"** command resets all the body disable bits. Transforming a fully **disabled body** is a no-operation.

3.4 THE ITERATION COUNTER.

"<digit>" Accumulate Iteration Count.
"<return>" Reset Iteration Count to Zero.

Digits typed without control keys are accumulated into an iteration counter. Tho iteration count is cleared by typing a carriage return. The count applies **to** Euclidean transformations and **sweeps.** The ability to iterate and to do macros is not developed because of **the** existence of GEOMED **imbedded** in LISP which provides better interactive programming facilities than would be possible' under **the** present character command scanner. (It is my design philosophy that **interactive geometric editing can in fact** be kept distinct from interactive programming).

3.5 DIRECT EUCLIDEAN COMMANDS.

3.5.1 **"U"** Unmove a Body.

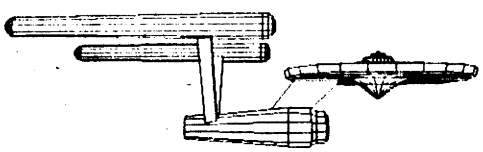
This command applies the inverse of the given body's frame of reference to the body itself. This has the effect of undoing any translations and rotations that have been applied to the body; that is the body's frame is brought back to be coincident with the world frame of reference, Thus a polyhedral body can be constructed in a standard orientation; then rotated randomly for inspection; and then be brought back to its original orientation for further editing.

3.5.2 **"X** PLACE <X> <Y> **<Z>"**

The location of the body, camera or vertex in the top of the stack is placed at the given world coordinates.

3.5.3 **"X** ORIENT <pan> <tilt> <swing>"

The orientation of the given body or camera is set to the specified values,



SECTION 4

TOPOLOGICAL COMMANDS.

4.1 FIVE EASY CREATIONS.

- 4.1.1 **"∞"** Instant Cube.
- 4.1.2 **"∝∞"** Instant Torus.
- 4.1.3 **"X** CUBE **<DX>** <DY> **<DZ>"**
- 4.1.4 **"X** CYLN <RADIUS> <N-SIDES> **<DZ>"**
- 4.1.5 **"X** BALL <RADIUS> <M-LONGITUDES> <N-LATITUDES>"

4.2 COPY, KILL AND DUAL.

- 4.2.1 **"C"** **Copy**.
- 4.2.2 **"K"** Kill.
- 4.2.3 **"∝K"** Edge-Vertex Kill.
- 4.2.4 **"βD"** Dual.

4.3 SWEEP COMMANDS.

- 4.3.1 **"S"** Sweep
- 4.3.2 **"∝S"** Pyramid.
- 4.3.3 **"R"** Rotation Solid Completion.
- 4.3.4 **"X** SILHOUETTE **<ZMIN><ZMAX>"**
- 4.3.5 **"X PRISMOID"**

4.4 THE EULER COMMANDS.

- 4.4.1 **"V"** Make Vertex Body.
- 4.4.2 **"E"** Make Edge and Vertex.
- 4.4.3 **"M"** Midpoint an Edge.
- 4.4.4 "J" Join Vertices.
- 4.45 **"G"** Glue Faces.

4.5 BODY INTERSECTION COMMANDS.

- 4.5.1 **"∝∩"** Body Intersection.
- 4.5.2 **"∝U"** Body Union.
- 4.5.3 **"∝¬"** Body Subtraction.
- 4.5.4 **"¬"** **Evert** Body Surface Orientation.
- 4.5.5 "|" Invert Edge Linear Orientation.
- 4.5.6 "$" Make Convex.
- 4.5.7 **"∝$"** Edge Slurp,

4.6 CUT COMMANDS.

"X CUT **<X><Y><Z>"**	Cut Bodies, Faces and Edges.
"∝X CUT <X> <Y> **<Z>"**	Cut Faces and Edges.
"βX CUT **<X>** <Y> **<Z>"**	Cut Edges.
"∈X CUT"	Kill Temporaries (Uncut).

4.7 PARTS TREE COMMANDS.

- "A" Attach.
- **"D"** Detach.

4.1 FIVE EASY CREATIONS.

4.1 .1 "∞" Instant Cube.

The instant cube command places a command string for making a particular cube into the teletype buffer. The purpose of this command is to make a demonstration object,

4.1.2 "∝∞" Instant Torus.

The instant torus command places a command string for making a particular torus into the teletype buffer. The purpose of this command is to make a demonstration object.

4.1.3 "X CUBE <DX> <DY> <DZ>"

The CUBE command makes a right rectangular prism with width height and depth as given in the three arguments. The body node of the cube is pushed into the stack, and the cube is located at the world origin. The absolute value of the arguments is taken so that the cube is always solid. A zero <DZ> returns a rectangular lamina rather than a cube.

4.1.4 "X CYLN <RADIUS> <N-SIDES> <DZ>"

The CYLN makes a right prism that approximates a circular cylinder. A zero <DZ> argument returns a circular lamina rather than a cylinder. The <N-sides> argument is forced to be at least three,

4.1.5 "X BALL <RADIUS> <M-LONGITUDES> <N-LATITUDES>"

The BALL command makes a polyhedron that approximates a sphere. The <M-Longitudes> and <N-Latitudes> are forced to be at least three and two respectively.

4.2 COPY, KILL AND DUAL.

4.2.1 "C" Copy.

This command copies the body or face from the **top of stack** in a reasonable manner and pusht it down. The copy of a body includes all the parts of that body. The copy of a **face, creates** a **two-faced** body lamina coincident with the given face.

4.2.2 "K" Kill.

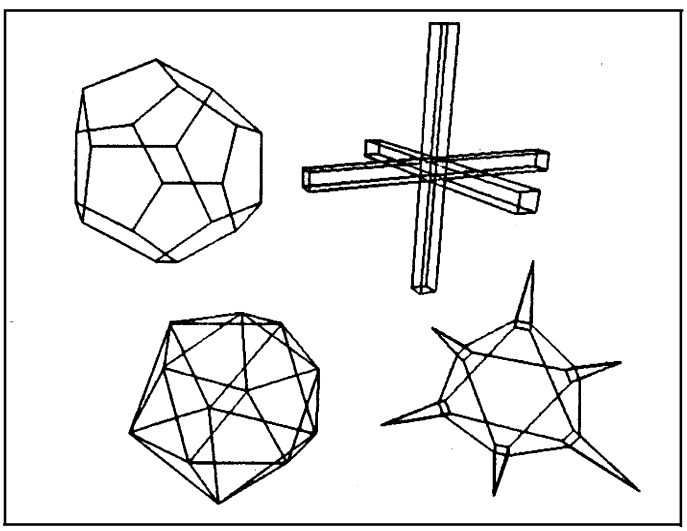
This command takes the top body, face, edge or vertex and attempts to delete it in a reasonable manner. Kill of a body deletes it and all its parts. Kill of a face removes the given face and all its edges and vertices but one vertex, which is placed in the center of the given face and is returned to the stack. Kill of an edge removes the given edge and its negative face, **NFACE(E).** Kill of a vertex removes the vertex and all its edges and faces but one face, which is returned to the stack. The kill operation is not defined in cases which would leave one-sided faces or dangling edges;. however the necessary trihedral restrictions are not imposed so that certain illegal kills **are** possible **and other** illegal kills are fatal.

4.2.3 "∝K" Edge-Vertex Kill.

The **"∝K"** kill of an edge removes the edge and its negative vortex, **NVT(E);** the positive **vertex, PVT(E),** is repositioned to be at the center of the original **edge. The PVT(E)** vertex is returned to **the** stack.

4.2.4 "βD" Dual.

The **"βD"** command makes the face/vertex dual of a body; that is all the faces become vertices and all the vertices become faces. This command will turn dodecahedrons into icosahedrons (or jacks into horny octahedrons) as illustrated.



4.3 SWEEP COMMANDS.

4.3.1 "S" Sweep.

"/3S" Sweep Cylinder, Edges Not Sharp."

This command sweeps a face into either a sheet or a solid depending on the NCNT of the face. If the **NCNT** is zero, then **"S"** returns a new face coincident with the given face and connected to it by new sweep edges and rectangular sweep faces forming a prism. When the NCNT is non-zero, only the first **NCNT** edges of the argument face's perimeter are swept into new faces, for the sake of creating solids of rotation, The **"S"** command applied to a vertex sweeps out a new vertex and a new edge. The **"/3S"** command is the same as "S" except that the new sweep edges are marked with the NSHARP (not sharp) bit.

4.3.2 "∝S" Sweep Pyramid.

"∈S" Sweep Pyramid, Edges Not Sharp,

Given a face, **"∝S"** will create a peak vertex and place it in the canter of the face connected by an edge to each of the face's original vertices. The peak vertex is returned to the stack, Given a vertex, the pyramid command forces all the faces of that vertex to be triangular (as if the given vertex were a peak vertex of a pyramid).

4.3.3 "R" Rotation Solid Completion.

After a wire has been swept around, you have a shell of rotation. To change a shell into a solid all that remains to be done is do a series of J commands on the leading and lagging vertices of the shell. The R command, using the NCNT to step around the pole caps, calls the J-command for all the necessary pairs of vertices. The sweep face is taken as an argument and nothing is returned.

4.3.4 "X SILHOUETTE <ZMIN> <ZMAX>"

This command sweeps a silhouette lamina body into a cone between -<zmin> and -<zmax> on the Z-axis of the now camera. The minus Z half space is the ono that contains the objects visible from the camera; that is, the camera's principle ray is the minus Z-axis ray.

4.3.5 "X PRISMOID"

This command is li ke the face sweep command except that the diagonal edges are provided across ail the new sweep face rectangles forming a prismoid of triangles rather than a prism of rectangles. (See, the icosahedron example, 2.4).

4.4 THE EULER COMMANDS.

4.4.1 "V" Make Vertex Body.

"βV" Make Body node.

This command creates a polyhedron consisting of one vertex and one face. The vertex is placed at the world origin. The V command takes no arguments and leaves the new body, face and vertex on the stack with the vertex on top. The **"βV"** command will make a naked body node, that is a body with no faces, edges or vertices.

4.4.2 "E" Make Edge and Vertex.

This command creates a new edge and a new vertex starting from an old vertex. The E command requires two arguments, the top argument is the old vertex and the second argument must be one of the faces to which the vertex belongs. The-new edge and vertex become associated with the given face.

4.4.3 "M" Midpoint an Edge.

This command takes an edge argument and creates a new edge and a new vertex. The new mid vertex is positioned along the edge in proportion to the value of DOEL. Type the commands **"€5M"** in order to make a midpoint that is geometrically in the middle of the given edge.

4.4.4 "J" Join Vertices.

This command creates a new face and a new edge. The **"J"** command has two forms. The general form of the **"J"** command expects two vertices of the same face to be given on the top of the stack, and it returns in their place a new edge and a new face with the edge on top. The wire form of the **"J"** command applies only to a face that doesn't have a closed perimeter. The wire **"J"** returns the other end of the wire in the top of stack and leaves the second argument unchanged.

4.4.5 "G" Glue Faces.

This command takes two faces from the -top of the stack. The faces should be of opposite orientation, they should have the same number of vertices, and the vertices should be nearly coincident. The two faces are deleted along with all the edges and vertices of the second one of them. If there were two bodies before, then one will remain; as a body glued to itself forms a hole (or handle) yielding a polyhedral surface of genus one greater than before. In either case, the glue command returns the body of the faces so formed.

4.5 BODY INTERSECTION COMMANDS.

4.5.1 "∝∩" Body Intersection.

Given two different bodies from the top of the stack, if the intersection of the space enclosed by their surfaces is not empty a new polyhedron will be formed to represent the surface of the space of intersection; and the given bodies will be killed.

4.5.2 "∝∪" Body Union.

Given two different bodies from the top of the stack, if the union of the space enclosed by their surfaces is simply connected (that is a'la de Morgan: the intersection of the space enclosed by their surfaces is not empty); then a new polyhedron will be created representing the surface of the union of the enclosed spaces of the given bodies. The given bodies will be killed.

4.5.3 "∝¬" Body Subtraction.

Subtract the top of the stack body from the body second in the stack (where subtraction means union of the evert of the top body with the second body).

4.5.4 "¬" Evert Body Surface Orientation.

GEOMED polyhedra have an inside and an outside irrespective of enclosing a finite positive volume; that is, a thing that appears to be a cube can either be a solid in space or a space in a solid. The evert command turns solids into holes and holes into solids. Holes are defined as polyhedra with negative volume.

4.5.5 "|" Invert Edge Linear Orientation.

Edges are directed vectors with a negative vertex and a positive vertex. The "|" command flips the linear orientation of an edge by swapping its vertices, faces and wings (edge wings are discussed in part II).

4.5.6 "$" Make Convex.

This command may be applied to a face or a body. All the faces referred to will be split into triangles.

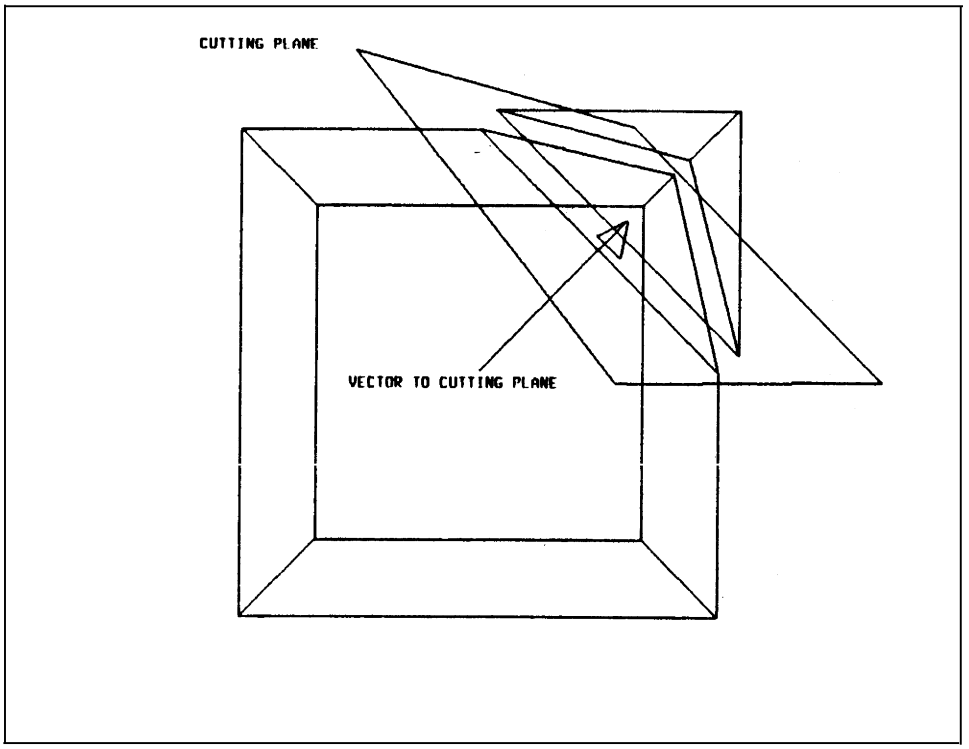
4.5.7 "∝$" Edge Slurp.

This command will attempt to remove or darken the visually unaesthetic edges that arise from the "$" make convex command. The edges slurped are the ones with dihedral angles of almost pi, and with wing angles that sum to less than pi at each vertex.

4.6 CUT COMMANDS.

"X CUT <X> <Y> <Z>"	Cut Bodies, Faces and Edges.
"∝X CUT <X> <Y> <Z>"	Cut Faces and Edges.
"βX CUT <X> <Y> <Z>"	Cut Edges.
"εX CUT"	Kill Temporaries (Uncut).

The cut command argument specifies a cutting plane in terms of a vector from the world origin; the direction of the vector is normal to the desired plane, and the magnitude of the vector is the distance of the origin from the desired plane.



4.7 Parts Tree Structure - The Attach and Detach Commands.

"A" ATTACH
"D" DETACH

The parts' tree commands "A" attach and and "D" detach, allow bodies to be connected into a tree structure without affecting their faces, edges or vertices. The "A" command links the body in the top of the stack to the body in the second stack position as a sub-part (or child). The detach command unlinks a body from its parent. Certain operations such as the Euclidean transformations apply to a body and all its descendents. If the top entities of the stack are not bodies, then these commands are no-operations. Because of the implementation, the parts structure is always kept as a tree; circularities and incest are prevented.

SECTION 5

INPUT/OUTPUT COMMANDS.

"I"	input B3D file.	"O"	output B3D file.
"∝I"	input CAM file.	"∝O"	output CAM file.
"βI"	input CRE film.	"βO"	output V2D file.
"εI"	input GEM file.	"εO"	output GEM file.

"βZ" take commands from GEO file.
"P" output PLT file of current display buffer.

5.1 I/O COMMANDS.

The "I" and "O" are the commands for input and output. There are two file formats for polyhedra named B3D and GEM. The plain "O" command will output a body (and all its parts) from the top of the stack into a B3D file. The plain "I" command will input a body (and all its parts) from a B3D file and will push it on the stack. The "∝I" and "∝O" input and output CAM files to and from the node of the "now" camera. The final simple I/O command is "P" which creats a plot file of the current display.

The "εI" and "εO" commands are nearly identical to the "I" and "O" commands, except that they read and write more of the contents of each body, face, edge and vertex node for the sake of GEOMES (SAIL) and GEOMEL (LISP) users. Specifically, words 0 and 8 of each node is included in the GEM format. Online GEOMES and GEOMEL documentation can be found under *.WRU[GEM,HE]@SAIL.

The meta I/O commands provide an indirect method of getting video images into and out of GEOMED. On the input side, a television image must first be converted into a line drawing using the program CRE which creates CRE files; on the output side, line drawings, V2D, are converted into televsion pictures using the program MKVID. The "βI" command inputs a CRE film of images and places the images on the now-camera's node. The "βO" command does a hidden line elimination and a photometric simulation (shines the sun on all the visible faces) in order to create a V2D file (vectors 2-D) for the sake of MKVID. Further details on CRE and MKVID are available from the system's HELP command.

5.2 ONLINE PLOT FILE HARDCOPY: PLTVEC, XAP AND XIP.

There are two online hardcopy plotting devices: the Xerox Graphics Printer, XGP and the Calcomp plotter. A plot file (which is a display buffer) can be plotted on the Calcomp by running the program PLTVEC. PLTVEC first asks for point vector default: REGULAR, DASHED or VECTOR; type the letter **"R"** or just type a carriage return. Next PLTVEC asks for a scale size; type a real number between 0.1 and 2.5. A scale 1 .O will yield a plot approximately the size of the physical display screen, 10 **inchs** on a side. Next PLTVEC asks for the plot file name, after which the plot begins. When the plot is completed, the question "move pen" appears and can be answered with the letters **"L"** for left, **"R"** for right, **"U"** for up and **"D"** for down. On the other hand, in order to get **an XGP** copy of a plot (or even a video file); type **"R XAP;L0,0;*<file>;"** or **"R XIP;L0,0;*<file>;". Futher** details on XAP and XIP are available from the system HELP command.

5.3 VIDEO IMAGE DISPLAY: DDVID.

After a **V2D** file is converted into a television file by the program MKVID; the picture, which is in standard Hand/Eye format, may be displayed on the video synthesizer via the Data Disc using the program DDVID. To display **a** television image using DDVID, type **"R** DDVID <return> E <return> **I<filename>** <return> **R<return>". DDVID** will type a list of its commands in response to a question **mark.**

5.4 GEOMED COMMAND FILES.

Contrary to its design philosophy, GEOMED will accept commands from a file. The proper design philosophy encourages the would be geometric programmer to use the LISP embedded or the SAIL embedded version of GEOMED notation rather than the syntactically weak single-character jump-table notation, A command file is executed by the **"βZ"** command which will ask for a file name. The **file** should have no line numbers. Comments may be included among the commands prefixed by **"3"** and terminated by a carriage-return and line-feed. A **"∝3"** comment prefix will cause the comment to **be** typed out; the **"β3"** comment prefix will cause GEOMED to wait for the operator to type a **character; and** the **"∈3"** comment prefix will both print the comment and wait.

SECTION 6

EDITOR CONTROL COMMANDS.

6.1 PROGRAM CONTROL.

6.1.1 "∈E" Exit GEOMED.
6.1.2 "∝#" Enter **DDT.**

6.2 STACK COMMANDS.

6.2.1. "↔" Swap First and Second Elements of Stack.
"∝↔" Swap First and Third Elements of Stack.
"β↔" Swap First and Last Element6 of Stack.
"∈↔" Swap Second and Third Element6 of Stack.
6.2.2 "↓" Duplicate Push Stack **Down.**
"↑" Discard Pop Stack.
6.2.3 "∝↓" Rotate the Stack Down.
"∝↑" Rotate the Stack Up.
6.2.4 "⊗" Push Universe Node into the Stack.
"∝C" Push the Now Camera into the Stack.

6.3 BODY NAMING AND RETRIEVING.

6.3.1 "N" Name Body.
6.3.2 "B" Body Get.
6.3.3 "∝B" Body Retrieval by Name or by Numeral.

6.4 FACE AND VERTEX PERIMETER TRAVELING.

" " Fetch Counter Clockwise.
"," Fetch **Clockwise.**
"." Fetch Other.
"∝V<n>" Fetch Nth Vertex of a Body.

6.5 LINK FOLLOWING COMMANDS.

"<" NFACE Link.
"≤" NED Link.
"∨" NVT Link.
"∩" DAD Link.
"⊂" BRO Link.
"←" ALT Link.
"," CW Link.
">" PFACE Link.
"≥" PED Link.
"A" PVT Link.
"U" SON Link.
"⊃" SIS Link.
"→" ALT2 Link.
"." CCW Link.

6.1 PROGRAM CONTROL.

6.1.1 "∈E" Exit GEOMED.

Exit the GEOMED editor. Returns control to monitor level or to user program level. When embedded, GEOMED returns the entity at the top of its stack to its caller as its value.

6.1.2 "∝*" Enter DDT.

Enters DDT if it exists. The SYS versions of GEOMED never have DDT; the versions GEOMED.DMP[GEM,HE] and G.DMP[GEM,BGB] will usually have DDT (which is actually a Swinehart RAID); and a GEOMES core image has DDT depending on how the compile and load was done. A DDT "∝P" command will continue execution in the GEOMED listen loop.

6.2 STACK COMMANDS.

6.2.1 "↔" Swap First and Second Elements of the Stack.

"∝↔" Swap First and Third Elements of the Stack.
"β↔" Swap First and Last Elements of the Stack.
"∈↔" Swap Second and Third Elements of the Stack.

GEOMED commands take their arguments from and leave their results in a pushdown stack of bodies, faces, edges and vertices. The contents of the stack are displayed on the left hand side of the display screen. The swap commands exchange the indicated elements of the stack if they exist. The stack can hold one hundred entities, although only the top thirty elements will be displayed.

6.2.2 "↓" Duplicate Push Stack Down.

"t" Discard Pop Stack.

The "↓" command push the stack down and places a duplicate of the previous top of stack into the top of stack. The "↑" command pops the stack discarding the top element.

6.2.3 "∝↓" Rotate the Stack Down.

"∝↑" Rotate the Stack Up.

The rotate stack commands preserve the number of elements in the stack. Rotate down pushs the stack and moves the last element to the top. The rotate up pop the stack and moves the previous top to the bottom.

6.2.4 "⊗" Push Universe Node into the Stack.

"∝C" Push the now camera into the Stack.

These two commands require no arguments, but merely push the Universe node or the current camera into the stack, so that node linking can be started.

6.3 BODY NAMING AND RETRIEVING.

6.3.1 "N" Name Body.

This command accepts a string of up to ten characters typed as a name for the body in the top of the stack; any characters can be used in the name, the named is terminated by a carriage return.

6.3.2 "B" Body Got.

This command will replace the edge, face, or vertex in the top of the stack with the body to which it belongs.

6.3.3 "∝B" Body Retrieval by Name or by Numeral.

Each body has a numeral corresponding to its position in its world's body ring. The "∝B" command will accept a name or numeral and will push the indicated body into the stack.

6.4 FACE AND VERTEX PERIMETER TRAVELING.

" "	Fetch Next Edge Counter Clockwise about Face (Vortex).
","	Fetch Next Edge Clockwise about Face (Vertex).
"."	Fetch Other Face (Vertex) of the Edge.
"∝V<n>"	Fetch Nth Vertex of a Body.
"≥"	Fetch First Edge of a Face or Vertex.

When the top two positions of the stack contain an edge and a face (or an edge and a vertex) that are connected; then both are intensified and face (vertex) perimeter traveling is possible using the "," or the l'." commands to obtain the next edge about the face (vertex) from the given edge, going in the clockwise or counter clockwise direction (as seen from the exterior surface of the polyhedron). It is also possible while perimeter traveling to cross to the other face (vertex) of the given edge by means of the "." command. Remember that whenever there is a face, edge or vertex in the top of the stack, the body to which it belongs can be obtained by using the "B" command. Mastering these commands alleviates the urge to have a light pen, or to take prolonged trips around edge rings. The "∝V<n>" retrieves the Nth vertex around the body ring of the body in the top of the stack; the vertex numerals are displayed when the "∝L" switch is toggled on. Trivia experts will be happy to know that "∝." and "∝," fetch the next vertex (face), counter clockwise and clockwise respectively, from the current edge with respect to the current face (vertex).

6.5 LINK FOLLOWING COMMANDS.

The **GEOMED** data structure consists of twelve word nodes which contain links and data. The following command characters fetch particular links of the entity in the top of the stack:

WORD1 :	"<" NFACE Link.	">" PFACE Link.
WORD2:	"≤" NED Link.	"≥" PED Link.
WORD3:	"∨" NVT Link.	"A" PVT Link.
WORD4:	"∩" DAD Link.	"∪" SON Link.
WORDS:	"⊂" BRO Link.	"⊃" SIS Link.
WORD6:	"←" ALT Link.	"→" ALT2 Link.
WORD7:	"," CW Link.	"." CCW Link.

The root of the data structure is the UNIVERSE node, which is unique, and which can **be pushed** into the stack by executing the "⊗" command. Directly accessible from the UNIVERSE node are the display ring and the world ring:

"∩" of the UNIVERSE	returns the "now" WORLD.
"∪" of the UNIVERSE	returns the "first" WORLD.
"." of the UNIVERSE	returns the "now" display.
"," of the UNIVERSE	returns the "first" display.

A display is not a node, but is rather a ring of windows. The refresh subroutine **GEODPY, refreshs** the windows of the "now" display ring.

"⊂" and "⊃" of a WINDOW travels the Window Ring of a display.
"." **and** "," of a WINDOW travels the Display Ring.

Each world **has** at least one camera, one sun, and a ring of bodies.

"←" of a WORLD	returns the "first" SUN.
"∩" of a WORLD	returns the "now" CAMERA.
"∪" of a WORLD	returns the "first" CAMERA.

"." and "," of a WORLD travels the World's Body Ring.
"⊂" and "⊃" of a WORLD travels the World Ring of the UNIVERSE.

Each camera points back to the world to which it belongs "∪"; and belongs to that world's camera ring , "⊂" and "⊃" ; and has potentially two-rings of images: the synthetic image ring, ".", of images from the hidden line eliminator, OCCULT; and the perceived image ring, ",", of Images from the video edge finder, CRE.

6.5 LINK FOLLOWING COMMANDS (continued).

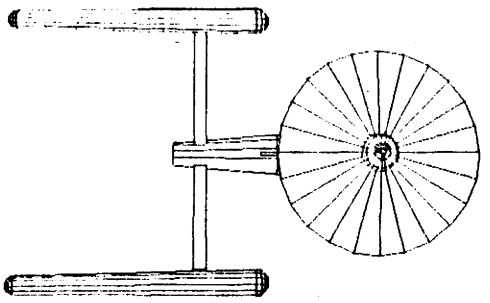
A polyhedral surface is composed of faces, edges and vertices. Each particular face, edge, or vertex belongs to one and only one body; and in turn each body has a ring of faces, a ring of edges and a ring of vertices:

"<" and **">"** of a body travels the Face Ring of that body.
"≤" and **"≥"** of a body travels the Edge Ring of that body.
"v" and "A" of a body travels the Vertex Ring of that body.

When the **"∝L"** switch is toggled ON, a numeral is displayed at each vertex of the body in the top of the stack, The N'th vertex of a body can be brought into the stack by typing **"∝V<n>"**. Besides belonging to rings; vertices and faces have one additional link, the PED or first edge link which points to one of the edges of the vertex or face. **The** PED of a vertex or **face** in the top of the stack can be obtained by typing **"≥"**.

Besides belonging to an edge ring, each edges points at its two faces, its two vertices and its four neighboring edges in each of its two faces; these last four pointers are the so called "wings" of the edge and are used to travel the perimeters of faces and vertices. Thus with an edge in the top of the stack, either of its two faces or two vertices can be obtained by typing one of the four characters **"<>vA"** respectively.

The arcane link positions in words -3, -2 and -1 of a node can be retrieved by the control **"<>≤≥vA"** respectively; while **meta "<>≤≥vA"** will retrieve from words 6, 7 and 8. No important links are stored in the arcane positions.



SECTION 7

DISPLAY CONTROL COMMANDS.

7.1 STATUS DISPLAY.

7.1.1 "≡" Status Display Toggle.
"∝≡" Toggle Now Window's Darken Bit.
"#" Clear the page printer.

7.1.2 "∂" Datum Display Toggle.
"∝∂" Datum Format Toggle.

7.1.3 "L" Toggle FEV Lights Switch.
"∝L" Toggle Body Lights Switch.
"βL" Toggle Frame Lights Switch.

7.2 HELP DISPLAY.

"?" Information Prefix.
"H" Help.
"∝H" Unhelp.

7.3 DISPLAY MODES.

"space" Display Refresh, **GEODPY.**,
"ALT" GEODPY with hidden lines **eliminated.**
"∝ALT" GEODPY with back side faces eliminated.
"βALT" GEODPY everything visible
"∈ALT" GEODPY with OCCULT **diagonostics.**
"_" Sticky **Alt.**
"∝_" Sticky **∝ALT.**
"β_" Sticky **βALT.**
"∈_" Sticky **∈ALT.**

7.4 EDGE DISPLAY BITS.

7.4.1 "∝D" Darken,
"∈D" Undarken.

7.4.2 "X NSHARP"

7.5 FACE COLORING.

"X COLOR <n>RED <n>GRN <n>BLU <n>ALB"

7.6 CAMERA COMMANDS.

"∝C" Push now camera into the stack.
"βC" Make new camera.
"β}" Step now camera forwards.
"β{" Step now. camera backwards.
"∝F" Set Camera's Focal Plane distance

7.7 WORLDS AND WINDOWS.

"W" Make Window in Now Display Ring.
"∝W" Make Window in New Display Ring.
"βW" Make New World.
"{" and "}" Step Now Display.
"∝{" and "∝}" Step Now World.

7.8 IMAGE RINGS.

"∝T" Make a Simulated Image Body.
"←" and "→" Step Perceived CRE Image Ring.
"∝←" and "∝→" Step Predicted OCCULT Image Ring.

7.1 STATUS DISPLAY.

The execution of nearly every GEOMED command is followed by the two display refresh routines called GEODPY and STADPY. STADPY refreshes the stack display, the editor status display, the node contents display, and the top of stack brighteners called "the lights" which are governed by the "L" command, GEODPY scans the "now" display ring of the universe node, and refreshes a piece of glass for each window having a camera.

7.1.1 "≡" Status Display Toggle.

"∝≡" Toggle Now Window's Darken Bit.

"*" Clear the page printer.

The STADPY (status display) is turned on and off by the "≡" command. The "∝≡" command will turn the window border on and off, (that rectangle). The "*" command types four form feeds, for clearing the page printer of visible characters.

7.1.2 "∂" Datum Display Toggle.

"∝∂" Datum Format Toggle.

The "∂" command causes the contents of the node at the top of the stack to be displayed in the lower right hand corner of the screen. Frame, body and camera nodes have a special display which indicates their location and orientation; the special display format can be toggled on and off by means of the "∝∂" command. In the regular format node display, the appropriate link following commands for the different words of the node are indicated.

7.1.3 "L" Toggle FEV Lights Switch.

"∝L" Toggle Body Lights Switch.

"βL" Toggle Frame Lights Switch.

When the "L" switch is on; faces, edges and vertices in the top of the stack are intensified. When the "∝L" switch is on, numerals are displayed for all the vertices of the body in the top of the stack. The vertex numerals can be used for obtaining a particular vertex using the "∈V" command. When the "βL" switch is on, three vectors are displayed indicating the direction of the axes of the current frame of reference; the vectors are labeled respectively by a letter "X", "Y", or "Z" followed by a number between -1 .O and + 1 .O indicating whether the vector is going away or coming towards the current camera.

7.2 HELP DISPLAY.

"?" information Prefix.
"H" Help.
"∝H" Unhelp.

Typing a question mark "?" followed by any character will type a one line reminder of what commands are invoked by that character. The "H" command allows the user to read this document a page at a time while using GEOMED. Help starts by displaying page-4 and waits for other desired page numbers; typing a carraige return will exit the help with the display persisting, control carraige return will exit the help with the display cleared. The "∝H" command clears the HELP display glass.

7.3 DISPLAY MODES.

"space"	Display Refresh, GEODPY.
"ALT"	GEODPY with hidden lines eliminated.
"∝ ALT"	GEODPY with back side faces eliminated.
"βALT"	GEODPY everything visible.
"ϵ ALT"	GEODPY with OCCULT diagonostics.
"_"	Sticky Alt.
"∝_"	Sticky ∝ALT.
"β_"	Sticky βALT.
"ϵ_"	Sticky ϵALT.

There are two basic display modes: with and without the hidden lines eliminated. Initially the display refresh displays all the lines that are visible from the simulated cameras of the windows of the now display ring. The "ALT" commands do one refresh in the mode selected by the control bits. An easy hidden line elimination takes less than a second; the hardest take up to thirty seconds. The underbar commands, "_", make the selected display refresh mode sticky, all refreshes are done in that mode until the next underbar. The spacebar command redoes the last refresh.

7.4 EDGE DISPLAY BITS.

7.4.1 "∝D" Darken. "ϵD" Undarken.

Given a body or an edge in the top of the stack; these two commands set or reset the darken bit all the specified edges. Darkened edges will not be intensified by the display refresh.

7.4.2 "X NSHARP"
"X GOURAUD"

Given a body or an edge in the top of the stack, these two extended scanner commands set the "not sharp" and the "Gouraud" bits, respectively all the edges of the body or of the given edge. The two bits specify different kinds of intensity smoothing to the "β0" command. The "not sharp" bit causes not sharp and not folded edges to be not visible after a hidden line elimination, which yields line drawings with a rounded appearance.

7.5 FACE COLORING.

"X COLOR <n>RED **<n>GRN** <n>BLU **<n>ALB"**

The coloring command applied to a body or a face sets the bytes of words 4 and 5 of all the faces given. The <n> arguments represent percentage values between 0 and 100; the argument must be suffixed by a letter: R, **G,** B or A standing for red, green, blue and albedo. Combining the colors red and green gives yellow; red and blue gives lavender; blue and green give a light sky blue. Lowering the albedo turns red into crimson, yellow into brown, green into dark green, sky-blue into slate, blue into deep blue, and lavender into purple.

In order to get the so called "I and **C"** required by **DDVID's** color command, "C"; suffix the filename you type to **MKVID** with a **"/C"** switch. To display a color television picture on the color synthesizer; you must remove a BNC terminator from the color synthesizer, which is located in the kludge bay above the video switch cable fan out; you must place the toggle on the upper left of the color television set in the up position; and you must- give the following commands to **DDVID: "C<filename>** <return> **E<return> R<return>"**.

The color synthesizer requires twelve data disc channels, which are usually not available in the day, The intensity channels are 36, 35, 34, 33, 37 and 32; the color channels are 31, 24, 23, 27, 30 and 25. The system information programs DDUSE and FINGER will type information on who is using which channels and where particular users are located.

7.6 CAMERA COMMANDS.

"∝C" Push now camera into the stack.
"∝F" Set Camera's Focal Plane distance.
"ßC" Make new camera.
"ß}" Step now camera forwards.
"ß{" Step now camera backwards.

There are two basic camera commands: **"∝C"** and **"∝F"**. The **"∝C"** command will push the now camera node into the stack. The **"∝F<distance>"** command will set the now camera's focal plane distance to the given value, the distance is assumed to be in units of millimeters. The initial focal plane distance is 12.5 mm. The additional camera commands allow the creation and use of a ring of cameras, The **"ßC"** command will create a new camera at the end of the camera ring of the now world. The **"ß}"** and **"ß{"** commands will advance or retreat the now camera around the camera ring.

7.7 WORLDS AND WINDOWS.

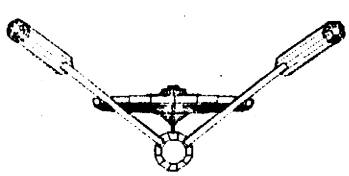
"W"	Make Window in Now Display Ring.
"∝W"	Make Window in New Display Ring.
"βW"	Make New World.
"{" and **"}"**	Step Now Display.
"∝{" and **"∝}"**	Step Now World.

Although, the data structure below GEOMED allows many worlds and windows; these commands are not intended for general use. The **"W"** command will make a new display window, such as the one **made** by the crosseyed stereo pairs' macro, **"β∞"**. Windows can be scaled and moved about the **screen** using the Euclidean transformations, one translation unit corresponds **to 100 display units. The "∝W"** command will make a new window in a new display refresh ring; and the **"{","∝{","∝}"** and **"}"** will as stated advance the display or world rings. When there is more than one camera (or one world) the node identification of the now camera (or now world) is displayed in the status,

7.8 IMAGE RINGS.

"∝T"	Make a Simulated Image Body.
"∝←" and "∝→"	Step Predicted OCCULT Image Ring.
"β←" and "β→"	Step Perceived CRE Image Ring.

The results of a hidden line elimination can be used to create a special kind of polyhedron that approximates the image. An image polyhedron is like a photographic print with faces, edges and vertices representing the details of the image that is on the print. The **"∝T** command (take a picture) **creates** such an image polyhedral body and attachs it to the OCCULT image ring of the noew camera. **Similar** to **"∝T",** the **"βI"** command inputs an image polyhedral bodies from a CRE file and **attachs them to** the CRE image ring of the now camera. The **"∝T"** command images are synthetic and the **"βI" command** images are perceived; the display refresh includes the first CRE and the first OCCULT image of the now camera ring (as well as its usual view of its simulated world). The control and **meta,** left and right arrow, commands will step the CRE and OCCULT rings as stated above. The creation of structures for both synthetic and perceived images brings us to the start of computer vision and the conclusion of GEOMED.



TEXT COMMANDS OF GEOMED 1973,

Toward the end of 1973, text nodes and text commands were added to GEOMED to demonstrate its potential value for mechanical drawing. This work was done by Tovar Mock, and the resulting alternate version of GEOMED is available on the system as the program named "G".

Text nodes are created (or edited) by applying the "T" command with a vertex in the top of stack. The header of a text node list is called a Y-node and can be reached from a text node by the "≤" command; Y-nodes carry their own 3-D space locus and so can be moved about using the Euclidean transformation. The resulting data structures can only be saved and restored by means of the "∝O" and "cl" commands, which dump and restore D3D files. The D3D format I/O doesn't exist in present GEOMED. The "T" command accepts text by means of an editor resembling the system's editors TV and E, which use the system line edit commands. The particular line edit mode commands of G are:

<CR> Forward 1 line
<VT> Backward 1 line

∝> Forward 4 lines
∝< Backward 4 lines
∝≥ Forward 16 lines
∝≤ Backward 16 lines

cD Delete a line
∝I Enter line insert mode
∝Z Concatenate next line with current line

∝E Return to GEOMED
∝V Update display
∝/ Shrink character size.
∝\ Expand character size.

The best way to get a labeled drawing into hardcopy is to "P" plot it out of G into a plot file, ".PLT"; and then-to use PLTVEC to output the drawing on the Calcomp Plotter or to use XAP to output the drawing to the XGP, Xerox Graphics Printer. See section 5.2, for details on how to make hardcopy drawings.

COMMAND INDEX BY ASCII ORDER.

PAGE	ASCII	CHARACTER	COMMAND
-	000	NULL	No operation.
31	001	↓	Push stack down.
31		∝↓	Rotate stack down.
2	002	∝	Control key prefix.
2	003	β	**Meta** key prefix.
33	004	∧	PVT, **Positive** vertex link.
26	005	¬	**Evert** Body.
26		∝¬	Body Subtraction.
2	006	∈	**Meta-Control** keys prefix.
1	007	π	Set rotation **streng8th,** RDEL.
18	010	λ	Set translation strength, TDEL.
-	011	TAB	No operation.
-	012	LF	No operation.
-	013	VT	No operation.
-	014	FF	No operation.
19	015	CR	Reset iteration count.
1	016	∞	Instant cube.
8		∝∞	Instant torus.
13		β∞	Crosseyed stereo.
36	017	∂	Datum display Toggle.
36		∝∂	Datum display format Toggle.
33	020	⊂	BRO, parts tree link.
33	021	⊃	SIS, parts tree link.
33	022	∩	DAD, parts tree link.
26		∝∩	Body Intersection.
33	023	U	SON, parts **tree** link.
26		∝U	Body Union.
19	024	∀	Enable all body motions.
19		∝∀	Disable frame motion.
19		β∀	Disable vertex motion.
19		∈∀	Disable parts motion.
29	025	∃	Comment prefix.
29		∃	Print comment.
29		β∃	Wait for character.
29		∈∃	Print comment and wait.
31	026	⊗	Push universe node into stack.
31	027	↔	Swap stack elements 1 and 2.
31		∝↔	Swap stack elements 1 and 3.
31		β↔	Swap stack elements 1 and last.
31		∈↔	Swap stack elements 2 and 3.
37	030	_	Sticky ALT display refresh.
37		∝_	Sticky ∝**ALT** display refresh.
37		β_	Sticky β**ALT** display **refresh.**
37		∈_	Sticky ∈**ALT** display refresh.
33	031	→	ALT2, link fetch.
-	032	TILDE	No operation.
-	033	≠	No operation.
33	034	≤	NED, **link** fetch.
33	035	≥	PED, link fetch.
36	036	≡	Toggle status display enable.
36		∝≡	Toggle window display enable.
33	037	∨	NVT, link fetch.
37	040	SPACE	Display refresh.
19	041	!	Set translation default.
-	042	"	No operation.
36	043	#	Clear page printer.
31		∝#	Enter **DDT.**

PAGE	ASCII	CHARACTER	COMMAND
26	044	$	Make convex.
26		∝$	Edge slurp.
18	045	%	Set DDEL, dilation strength.
-	046	&	No operation.
	047	'	No operation.
17		(	Euclidean transformation minus Y.
17	050	)	Euclidean transformation plus Y.
17	052	*	Euclidean transformation plus Z.
32	053	+	Fetch other face or vertex.
32	054	,	Fetch clockwise.
17	055	-	Euclidean transformation minus Z.
32	056	.	Fetch counter clockwise.
17	057	/	Halve strength of transformation.
19	060-071		DIGITS 0 THRU 9
17	072	:	Euclidean transformation plus X.
17	073	;	Euclidean transformation minus X.
33	074	<	NFACE, link fetch.
	075	=	No operation.
33	076	>	PFACE, link fetch.
2	077	?	Information prefix.
17	100	@	Set rotation default.
27		A	Attach.
19	101	βA	Cycle Axis count.
32	102	B	Body of a face, edge or vertex.
32		∝B	Retrieve body by numeral or by name.
23	103	C	Copy.
13		∝C	Push now camera into the stack.
27	104	D	Detach.
37		∝D	Darken.
37		€D	Undarken.
25	105	E	Make edge and vertex.
31		€E	Exit GEOMED.
18	106	F	Step frame selector forwards.
38		∝F	Accept focal plane distance.
18		βF	Step frame selector backwards.
25	107	G	Glue face-face,
37	110	H	Help display.
28	111	I	Input B3D.
28		∝I	Input CAM.
28		βI	Input CRE.
28		€I	Input GEM.
25	112	J	Join vertex-vertex.
23	113	K	Kill entity.
23		∝K	Kill edge and vertex.
36	114	L	Face, edge, vertex lights toggle.
36		∝L	Body lights toggle.
36		βL	Frame of reference lights toggle.
25	115	M	Midpoint and edge.
32	116	N	Name a body.
28	117	O	Output B3D.
28		∝O	Output CAM.
28		βO	Output V2D.
28		€O	Output GEM.
29	120	P	Plot file output.
19	121	Q	Toggle frame origin switch.
24	122	R	Rotation completion.

PAGE	ASCII	CHARACTER	COMMAND
24	123	S	Sweep cylinder.
24		∝S	Sweep pyramid.
24		βS	Sweep cylinder edges not sharp.
24		ϵS	Sweep pyramid edges not sharp.
39	124	∝T	Take a simulated picture.
20	125	U	Unmove a body.
25	126	V	Make vertex body.
32		∝V	Retrieve Nth vertex of a body.
25		βV	Make body node.
39	127	W	Make window in now display ring.
39		∝W	Make window in a new display ring.
39		βW	Make new world at end of world ring.
-	130	X	Extended command scanner:
21			PLACE <x> <y> <z>
21			ORIENT <pan> <tilt> <swing>
22			CUBIC <X width> <Y height> <Z depth>
22			BALL - <radius> <M longitudes> <N latitudes>
22			CYLN <radius> <N sides> <Z length>
27			CUT <x> <y> <z>
38			COLOR <n>RED <n>BLU <n>GRN <n>ALBEDO
37			NSHARP
24			PRISMOID
24			SILHOUETTE <zmin> <zmax> .
-	131	Y	No operation.
29	132	βZ	Read commands from GEO text file.
	133	[	No operation.
1;	134	\	Double strength of transformation.
-	135	|	No operation.
31	136	↑	Pop stack.
31		∝↑	Rotate stack upwards.
33	137	←	ALT link fetch.
39		∝←	Step thru simulated image ring of now camer
-	140	'	No operation,
	141 THRU 172		Lower case letters same as upper case.
3-S	173	{	Step now display backwards.
39		∝{	Step now world backwards.
39		β{	Step now camera backwards.
26	174	|	Invert edge linear orientation.
37	175	ALT	Display refresh with hidden lines eliminated.
37		∝ ALT	Display refresh with backside faces eliminate
37		βALT	Display refresh with everything visible.
37		ϵ ALT	Display refresh with OCCULT diagonostics.
39	176	}	Step now display forwards.
39		∝}	Step now world forwards.
39		β}	Step now camera forwards.
-	177	RUBOUT	No operation.