

**A Short Guide to AutoCAD Drawing Primitives for 3D
Computer Graphics Models and the Walkthrough
AutoCAD-to-Polygon Conversion Program**

John C. Alspaugh
22 April 1992

Abstract: Intended for those who want to build models for 3D computer graphics as painlessly as possible, this document introduces the reader to those AutoCAD drawing primitives and AutoCAD structures that can be converted to polygons with the Walkthrough AutoCAD-to-polygon conversion program. It also discusses those primitives that are not converted and why they are not converted. In addition, this document introduces the reader to more advanced AutoCAD modelling techniques for efficient model-building and model-management. The final section tells how to run the AutoCAD-to-polygon conversion program: file locations, input file requirements and forms, and the format of the output file.

Contents

1 Entities	2
1.1 Entity creation.....	3
1.1.1 Usable entities.....	3
1.1.1.1 ARC.....	4
1.1.1.2 Lines: LINE and PLINE.....	7
1.1.1.3 CIRCLE.....	8
1.1.1.4 3DFACE	9
1.1.1.5 SOLID.....	10
1.1.1.6 Meshes	11
1.1.1.7 3-D.....	14
1.1.2 Degenerate entities	14
1.2 Entity modification	14
1.2.1 General commands for entity-modification	14
1.2.2 Entity-specific commands.....	16
1.3 Entity manipulation.....	17
2 Blocks and Layers.....	17
2.1 Block creation	18
2.2 Block modification.....	19
2.3 Block manipulation	20
3 Advanced Modelling	20
3.1 Blocks.....	23
3.2 Layers	24
4 Converting from AutoCAD Form to Polygons.....	25
4.1 Color-table.....	25
4.2 Running acadtopoly.....	26
4.3 Output	26
4.4 Correcting the Polygon Model.....	28
4.4.1 Backfacing polygons.....	28
4.4.2 Coincident polygons	30
4.4.3 Misoriented polygons.....	30
5 Conclusion.....	31

AutoCAD: *n.* a large, complex computer-aided design program that strikes fear into the hearts of even the most experienced graphics users.

You've heard this before. Not the actual words, and probably not in dictionary entry style, but you have gotten the word that AutoCAD is the ultimate in ugly programs. This is somewhat exaggerated; it is a large program, with its share of strange commands and methods, but it is also an adequate tool for building 3-dimensional, architectural-type models.

This document is to help you get around these two difficulties. It details how to draw with AutoCAD, while ignoring the commands for manipulating the drawing, the drawing files, and the requisite hardware. I will discuss the major object manipulation and modification commands, but not the details of how to *run* AutoCAD. More complete treatments of AutoCAD are to be found in the *AutoCAD Reference Manual*, which sits on the lower shelf of the bookcase in the Graphics Lab between the aquarium and Cezanne. The *AutoCAD Reference Manual* is not very readable, and there are other manuals on the same shelf that are easier to follow. If you need more information, any of these documents can provide it.

Section 1, Entities, and Section 2, Blocks, discuss the creation, modification, and manipulation of entities and blocks, the AutoCAD drawing primitives. In Section 3, I will go into some advanced modeling techniques and, in Section 4, how to convert an AutoCAD DXF file (called a Drawing Exchange File and produced using the DXFOUT command in AutoCAD) into a file of polygons that you can further convert to whatever format you need. We will do the conversion with an AutoCAD file parser called acadtopoly, built by the UNC Walkthrough project (for simplicity I will call the

AutoCAD-to-polygon conversion program “the AutoCAD parser” or simply “the parser” from now on).

For greater clarity, AutoCAD commands will be in small caps, new terms and concepts will be in italics, Walkthrough program commands will be in a different font, and familiar terms applied in new context will be put in quotation marks. Boldface is used to give special emphasis where special emphasis is due.

1 Entities

AutoCAD has a number of primitives -- things such as lines, circles, and solids -- that make up a drawing. These primitives tend to be 2-dimensional entities -- lines, circles, polylines -- that can be extruded perpendicular to the plane; for example, a circle drawn in the X-Y plane, when extruded along the Z-axis, becomes a cylinder. The amount of extrusion is called the entity’s *thickness*. Entities are simple objects that AutoCAD treats as single elements when creating, manipulating, or modifying them. Models are made by combining these primitives or collections of these entities called *blocks*. A block is a user-defined group of entities that can be treated as a single unit; I will discuss blocks at greater length in Section 2. An AutoCAD drawing is a list of blocks and primitives, each of which can be selected, manipulated, or modified without affecting the others. I will refer to these drawing primitives as *entities*, and will discuss their creation, manipulation, and modification in AutoCAD with attention to the way the parser interprets the AutoCAD commands.

I will discuss entity creation in greater detail than entity modification or entity manipulation, because the effects of entity modification are largely transparent to the parser and the effects of entity manipulation are entirely transparent. Modifying or manipulating an AutoCAD entity tends not to affect the parser’s polygonal representation for that entity, just its position, orientation, color, thickness, and so on. In Sections 1.2 and 1.3, I will discuss what side effects do exist.

1.1 Entity creation

I will not dwell on how to draw the simple entities; the *AutoCAD Reference Manual*, chapter 4 (pages 65-110) covers this adequately. Entities are created with commands of the same name as the entity; for example, ARC creates an arc, SOLID a solid, CIRCLE a circle. The majority of these commands are self-documented and intuitive, with each step in the command indicating the sort of input that is expected. However, there are exceptions; I will detail how to draw some of the more unusual objects, because their methods are unexpected.

One thing to remember when drawing with AutoCAD: use the Right-hand Rule (Right-hand Rule: if the fingers of the right hand curl in the direction of the vertices -- the first vertex at the base of the finger, the last vertex at the finger tips -- then the thumb indicates the front of the face). Entities drawn by selecting points in a counterclockwise order will produce correctly oriented polygons. Figure 1 shows a back-facing polygon and a correctly oriented (or front-facing) polygon to illustrate this idea. The numbers beside the vertices give the order in which the vertex was drawn.

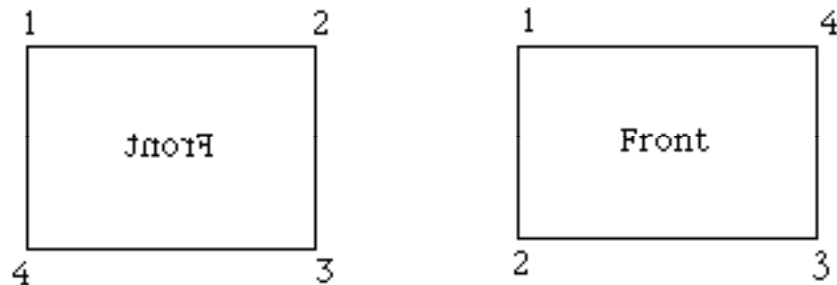


Figure 1: 3Dfaces. The face on the left is drawn with vertices chosen in clockwise order; the one on the right is drawn according to the Right-hand Rule.

There is, of course, an exception to this rule; the solid is drawn entirely differently (this will be discussed in Section 1.1.1.5).

1.1.1 Usable entities

The parser interprets some AutoCAD entities as you would expect it to, and others slightly differently. I will term *usable* entities those AutoCAD entities that the

parser correctly converts to polygons, with the polygons positioned as they were in the AutoCAD drawing. Those entities that the parser does not create polygons for, or does not make the polygons as they were in AutoCAD, will be termed *degenerate* entities and will be discussed throughout Sections 1.1.1 and 1.1.2.

1.1.1.1 ARC

The parser can produce two shapes from arcs; the resulting shape depends on whether the arc has a thickness (whether the arc is extruded perpendicular to the plane in which it is defined). A flat, 2-dimensional arc in the AutoCAD drawing becomes a flat, 2-dimensional surface: a surface enclosed by the curve and the chord across the end of it. It does not produce a pie shape, as one might expect. Figure 2 illustrates this with a conceptual drawing of two example arcs: an arc of less than 180 degrees, on the left, and an arc of more than 180 degrees, on the right. However, this is not how the actual polygons would be defined for such arcs. The parser would produce quadrilaterals **approximating** this shape, as in Figure 3, but not a single polygon **of** this shape (the number of quadrilaterals depends on the size of the angle).

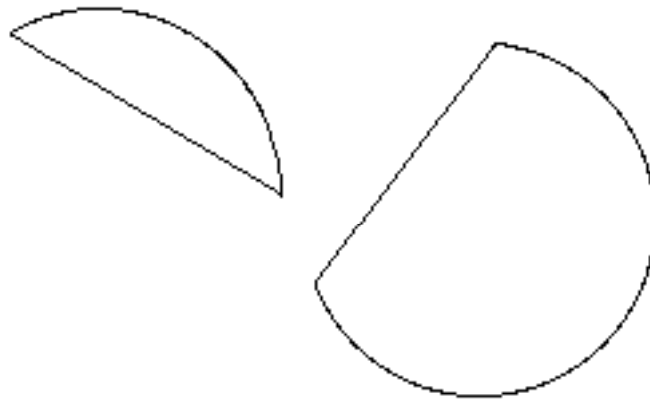


Figure 2: Conceptual Diagram of Faces Produced from Non-extruded Arcs

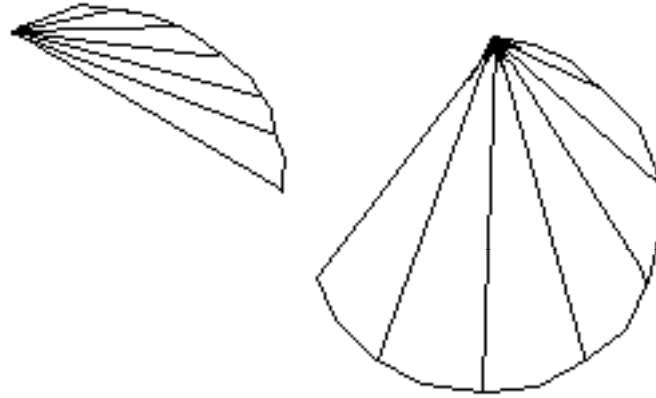


Figure 3: Quadrilaterals approximating the arcs of Figure 2

An extruded arc, one with thickness, produces polygons approximating a curved face around the axis of extrusion. It remains an arc, but is an arc with height. Figure 4 gives an example of how an extruded arc would appear after being converted by the parser, both conceptually and polygonally.

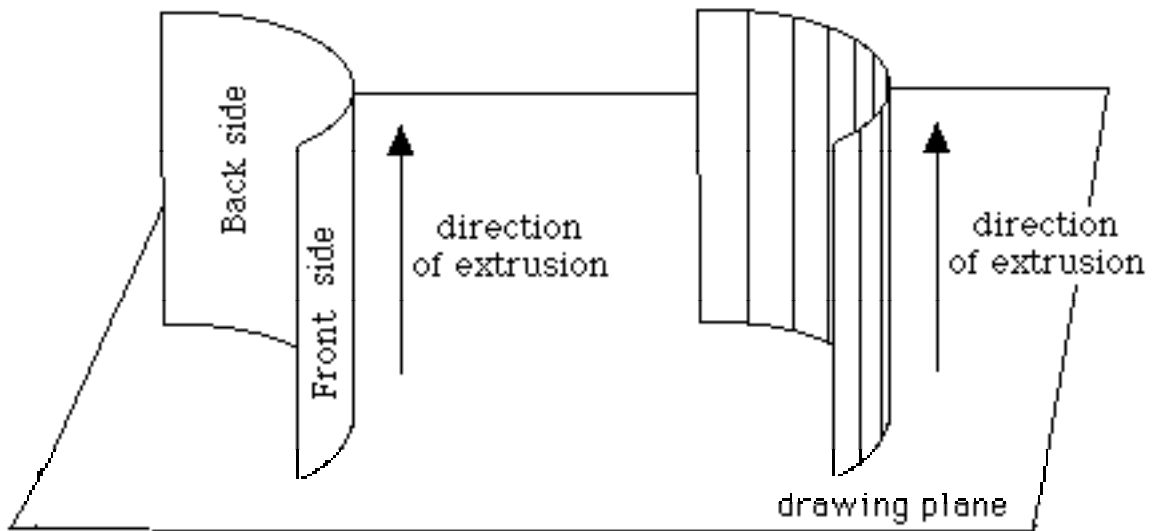


Figure 4: On the left, a conceptual diagram of an extruded arc, and on the right, the polygons produced by the parser. The view is at an oblique angle to the drawing plane.

The thickness can be set before creating the entity in AutoCAD (with the ELEV command) or, after the entity has been created, by modifying the entity with the CHANGE command (CHANGE THICKNESS). Entities that can be extruded (lines, arcs, polylines, and solids) are extruded perpendicularly to the plane in which they were

drawn. The 3Dfaces in Figure 1 and the arcs in Figure 2 (none of which are extruded) were drawn in the plane of this page. However, in Figure 4, the extruded arc, the drawing plane is skewed in relation to the plane of the page. If we were to view the extruded arc as it was drawn, we would see a curve with no discernable thickness. For clarity, views of extruded AutoCAD entities will be shown at an oblique angle, unless otherwise noted.

The sign of the thickness (negative or positive) influences the orientation of the polygons approximating the extruded arc. Figure 4, for example, has a positive thickness, as indicated by the direction of extrusion. Thus, the face in Figure 4 has its front side on the outside of the arc. A negative thickness would produce the reverse; the outside of the arc would be the back side of the face and the inside of the arc would be the front. This is illustrated in Figures 5 and 6. The line segment is the same in both figures; vertex 1 is the starting point of the segment, vertex 2 is the end point. The parser calculates the vertices of the polygon for this segment to be: vertex 1, vertex 2, vertex 2 + thickness, vertex 1 + thickness. A positive thickness means the polygon is defined as in Figure 5: the arrow inside the polygon indicates the order for the vertices, which is counterclockwise, and thus we are looking at the front of this polygon (by the Right-hand Rule). Likewise, in Figure 6, we are looking at the back side of the polygon because the vertices are listed in clockwise order. Extruded lines and polylines also share this property; the orientation for any segment of an extruded arc, line, or polyline is decided by the sign of the thickness.

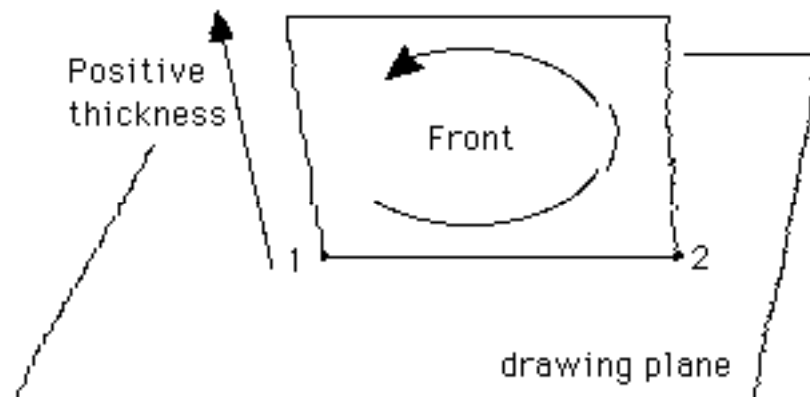


Figure 5: A Line Segment with Positive Thickness, viewed from an oblique angle. The axis of extrusion is coming out of the page.

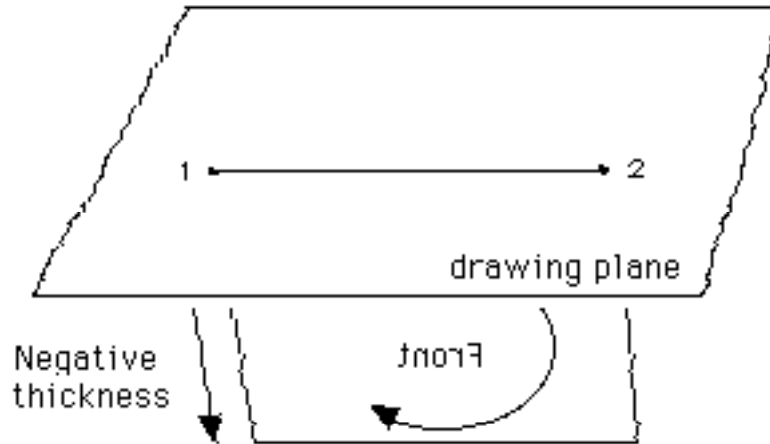


Figure 6: A Line Segment with Negative Thickness, viewed from an oblique angle. The axis of extrusion is going into the page.

1.1.1.2 Lines: LINE and PLINE

The LINE command creates useful entities only when given a thickness; a line without thickness is only a 2-dimensional object and cannot represent anything physical in 3-dimensional space. The parser thus ignores lines without thickness as degenerate entities. LINE with thickness produces a face in the direction of extrusion (lines are extruded perpendicularly to the the plane they were drawn in). Lines may be drawn as connected “segments,” with the first vertex of the new segment beginning at the last vertex of the previous segment. These segments are not actually components of a single entity; each segment is itself an independent line. AutoCAD automatically gives the option of making the end point of the last segment the start point of a new segment. Each segment is manipulated or modified independently of the rest. This makes the line ideal for drawing walls in architectural models: they are quickly drawn, easily modified, and readily make wall-like polygons.

Polylines (produced with the PLINE command) are compound entities. A polyline is made of line segments and arcs in a plane, connected at end points. Although -- as in the line -- each line segment or each arc or the polyline is specified as beginning where the last ended, the polyline -- unlike the line -- is treated as a single entity: the segments are not independent. To select a segment of a polyline is to select the entire polyline. This cohesiveness makes them convenient for defining 3-dimensional meshes, in which one might want a compound curve to approximate a complex surface. As with the line,

the parser interprets only the 2-dimensional, extruded polyline as polygons, ignoring the 2-D polylines without thickness.

As in the extruded arc, the orientation of a line or polyline depends on the extrusion vector (whether the line has a positive or negative thickness). A positive thickness has an extrusion vector pointing above the plane (“up”) the entity was defined in, and a negative thickness has the extrusion vector pointing below the plane (“down”). The parser interprets lines and the arcs and line segments of a polyline as depicted in Figures 4 and 5 of Section 1.1.1.1.

There is a 3-dimensional polyline called 3DPOLY, which can be defined in 3-dimensional space as opposed to a plane; however, the parser treats all cases of the 3DPOLY as degenerate and ignores them.

1.1.1.3 CIRCLE

The CIRCLE is similar to the arc; it can become either a flat face or an extruded shape. Without a thickness, it is a circular array of 12 triangles, as in Figure 7. Unlike previous figures, Figure 7 is not a conceptual diagram but a drawing of how the polygon approximation would look. The circular face is oriented with the front sides facing out of the screen as the circle was drawn. With positive thickness, the circle is interpreted as a hollow column, approximated by 12 polygons, with front faces facing outward, as in Figure 8. Figure 8 is also drawn as it is represented polygonally, rather than as a conceptual drawing.

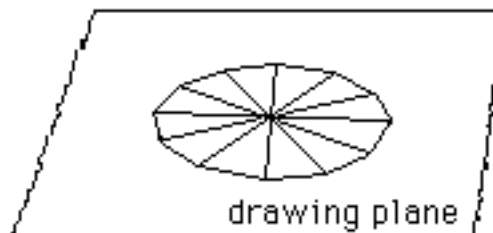


Figure 7: A Non-extruded Circle on a plane, viewed at an angle to demonstrate that it is without thickness.

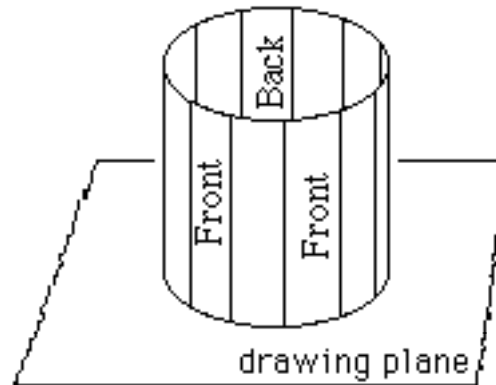


Figure 8: The Circle of Figure 7, extruded above the plane. A negative thickness would cause the circle to be extruded beneath the plane.

1.1.1.4 3DFACE

The 3dface, shown in Figure 1, is the AutoCAD entity most like the abstract idea of a polygon: it is a face, a polygon positioned in 3-dimensional space. It has, at most 4 vertices, and is selected a vertex at a time. It adheres to the Right-hand Rule and may be drawn in any 3-dimensional plane. This makes it ideal for sloping surfaces, such as roofs; it is also a good choice for drawing floor polygons. It cannot be extruded. 3Dfaces should be drawn according to the Right-hand Rule to ensure they are oriented correctly.

As far as the parser is concerned, 3Dfaces and extruded lines are frequently indistinguishable: both are converted into polygons. However, in the AutoCAD model, they are quite different. A line can only be extruded along the Z-axis, whereas a 3Dface can be oriented anywhere in 3-dimensional space. This implies that a polygon generated from an extruded line must be a rectilinear quadrilateral (two sets of parallel sides), because the extrusion is the same at both ends of the line. A 3dface is not constrained to have any parallel sides, or even four sides: you can, by hitting return instead of the fourth vertex, draw a triangle. But the 3dface is difficult to modify. The geometry can be changed only with the STRETCH command, which can be tedious in a complex model. It is much easier to change a line's thickness or start and end points (using CHANGE THICKNESS or TRIM and EXTEND, respectively). In the Walkthrough project models, lines are used for the walls and 3Dfaces for the window sills, roof, and furniture.

As a final note on lines versus 3Dfaces: it is possible to extrude lines other than along the Z-axis. There is a concept of “world” space, with a canonical set of axes, and within the world space, a user-defined set of axes where the user is drawing. This is the User Coordinate System (UCS), and it makes it possible to reorient the X-Y drawing plane in some arbitrary manner and then work there, as if in world space. UCS is beyond the scope of this primer; see p. 225 of the *AutoCAD Reference Manual* for instructions. UCS is not required for basic AutoCAD modeling, and can become very complicated to maintain. The parser does correctly position in world space entities drawn in a User Coordinate System.

1.1.1.5 SOLID

The solid is a quirky entity. Unlike most AutoCAD entities, which must be drawn counterclockwise to ensure proper orientation, solids are drawn in AutoCAD as in Figure 9. The numbers beside the vertices represent the order in which the vertices were drawn. Like the 3Dface, the solid has a maximum of 4 vertices (AutoCAD calculates the other vertices necessary to make a 3-D solid by adding the thickness to the Z value for each vertex. The X and Y for each vertex remains the same).

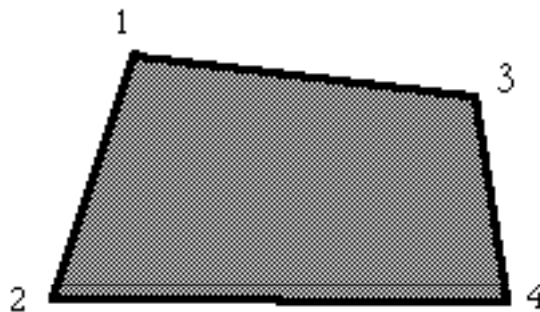


Figure 9: A Correctly Drawn Solid, as Seen from the Top. The drawing plane is parallel to this page.

A solid drawn according to the right-hand rule will produce a bow tie shape, as in Figure 10. This mistake cannot be corrected except by deleting the tie-shaped solid and redrawing the entity.

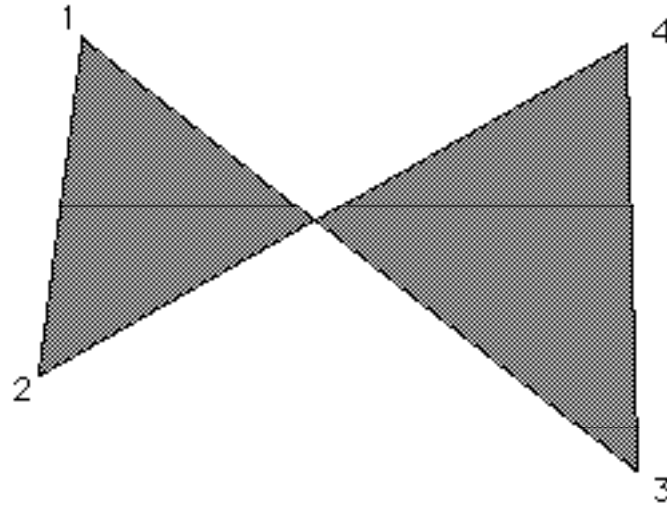


Figure 10: An Incorrectly Drawn Solid, Viewed from the Top. The drawing plane is parallel to this page.

The parser turns a solid into 6 polygons -- 4 sides, a top, and a bottom --that together define the same shape as in the solid, with the front sides on the exterior. Solids should be given a thickness; a solid without a thickness becomes two co-planar polygons, which is undesirable.

1.1.1.6 Meshes

Meshes are somewhat unusual entities, so I will go into more detail. There are four varieties of mesh-type entities, each doing something slightly different. The parser transforms the meshes into quadrilateral polygons almost exactly as one would expect, except when the mesh produces polygons that are not completely flat. These “bent” polygons are called **non-planar**, because they do not lie completely in any one plane. Non-planar polygons are undesirable (for example, the Walkthrough display programs will not load a non-planar polygon) and are automatically triangulated in the parser.

The four basic varieties of meshes are RULESURF, TABSURF, REVSURF, and EDGESURF. RULESURF, as in Figure 11, produces a mesh between two curves; TABSURF (Figure 12) makes a mesh from a curve defining a surface and an extrusion vector for that curve, extruding the curve the length of the vector in the direction the vector points; REVSURF (Figure 13) creates a surface of revolution by rotating a curve about an axis; EDGESURF (Figure 14) interpolates a mesh between four adjoining curves, which must meet at their end points. A curve in any of the surfaces can be an

ARC, LINE, PLINE, or 3DPOLY (3-dimensional polyline); these curves are separate from the mesh and, if not removed, will be interpreted by the parser as an arc, line, polyline, or 3D polyline, as is appropriate.

The number of gradations in the mesh are defined by the variables SURFTAB1 and SURFTAB2 (the default value for both variables is 6, and can be changed with the SETVAR <variable name> command). The parser translates meshes into polygons influenced by the SURFTAB variables; EDGESURF and REVSURF, for example, are converted into surfaces of SURFTAB1 times SURFTAB2 polygons. RULESURF and TABSURF are only SURFTAB1 number of polygons.

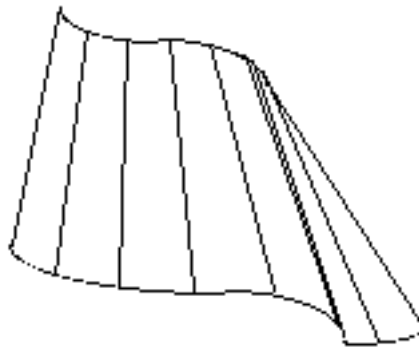


Figure 11: An Example of a RULESURF Mesh. The two curves are at top and bottom of the figure. The “curves” do not have to be formed from arcs; you can use circles, points, lines, polylines, arcs, or 3D polylines.

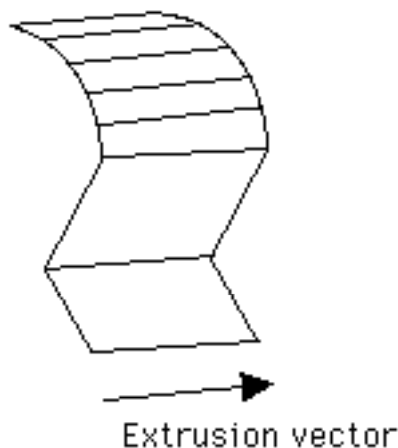


Figure 12: An Example of a TABSURF mesh. The defining curve is at the left side of the figure and is extruded to the right. The drawing plane for the curve is skew to the page and perpendicular to the extrusion vector (which is pointing into the page).

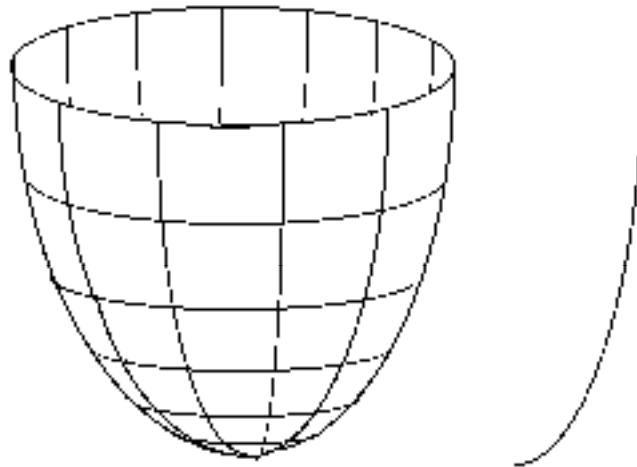


Figure 13: A Surface of Revolution. The curve used to generate the surface has been shifted to the right of the mesh; it was rotated about a line (not visible) inside the surface. The curve at right lies in the drawing plane.

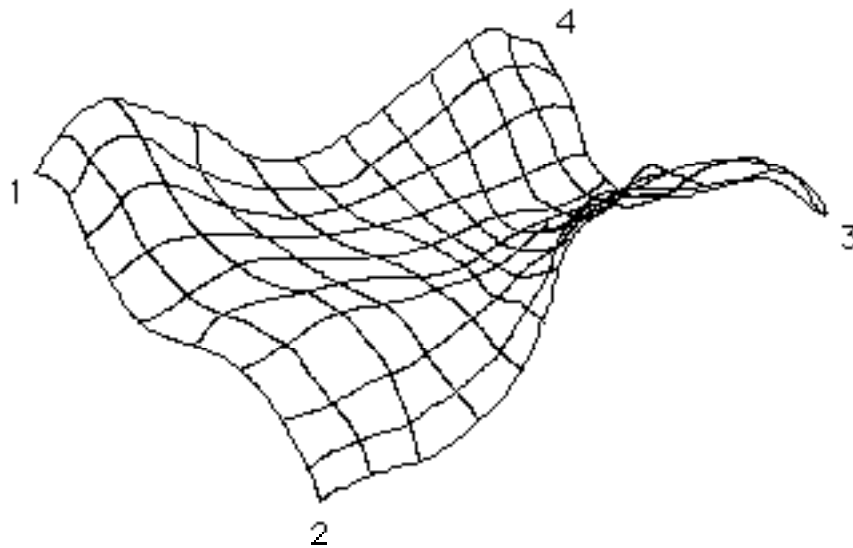


Figure 14: An Example of an EDGESURF. The four defining edges (in this case, 3-dimensional polylines) are from vertex 1 to vertex 2, from vertex 2 to vertex 3, from vertex 3 to vertex 4, and from vertex 4 to vertex 1.

For more details on the use of meshes, see pages 92 - 97 of the reference manual.

1.1.1.7 3-D

There is a special AutoCAD command menu of 3-dimensional shapes and objects called “3-D” accessible from the main AutoCAD menus. It includes the meshes of Section 1.1.1.6 and solids of Section 1.1.1.5, as well as wedges, tori, spheres, domes and bowls. These surfaces are not all well-designed, Spheres and domes are correctly oriented, but the tori and bowl are back-facing. The 3-dimensional pyramid in the script is not 3-dimensional, and so forth. The objects look all right in AutoCAD, but apparently were not designed to be really 3-dimensional. There may also be some more problems in the parser. I have not spent much time getting these objects to work correctly because there has been little need for them, and because all of the special entities can be made with other commands (for example, you can make tori, spheres, domes, and bowls with the REVSURF command). So use the 3-D menu at your own risk; the mesh commands work, but the other commands are not guaranteed.

1.1.2 Degenerate entities

As mentioned throughout Section 1.1.1, the parser does not convert some entities in certain cases. The entities listed here are not converted at all: POINT, LINE (no thickness), PLINE (no thickness), 3DPOLY (3-dimensional polyline), TEXT, SHAPE(a sort of primitive block; very quick to insert, but very primitive). These entities are not 3-dimensional, or are not necessary for 3-dimensional modeling, or are just things no one has gotten around to adding to the parser.

1.2 Entity modification

Entity modification commands fall into two categories: general commands and entity-specific commands. General commands work on all AutoCAD usable entities, and produce uniform -- though sometimes unexpected -- results. The entity-specific commands work on only a few entities, addressing special characteristics of those particular entities.

1.2.1 General commands for entity-modification

There are four important general entity-modification commands: SCALE, STRETCH, MIRROR, and CHANGE. The command names are self-explanatory, except

for the CHANGE command; for example, SCALE changes the size of an object, scaling it in relation to some user-specified point. STRETCH likewise does what it sounds like it should: you can "stretch" an entity by choosing a vertex or several vertices and then moving those vertices with respect to the unselected vertices. These commands do not adversely affect the way the parser interprets the entity; a stretched entity is turned into polygons with the same stretched shape as the entity, but those polygons are not otherwise altered.

MIRROR, on the other hand, alters entities in two ways, one obvious and one subtle. Applying MIRROR to an entity or collection of entities causes them to be reflected about a user-selected line (the user is prompted to enter a pair of points, not an actual line in the drawing), creating a reverse image of those entities. However, the mirrored entities will be flipped: every polygon in the reversed image will face in the opposite direction from its corresponding polygon in the original collection of entities, as demonstrated in Figure 15. The MIRROR command produces a mirror-image of the appearance of an object, but not of the polygon orientation.

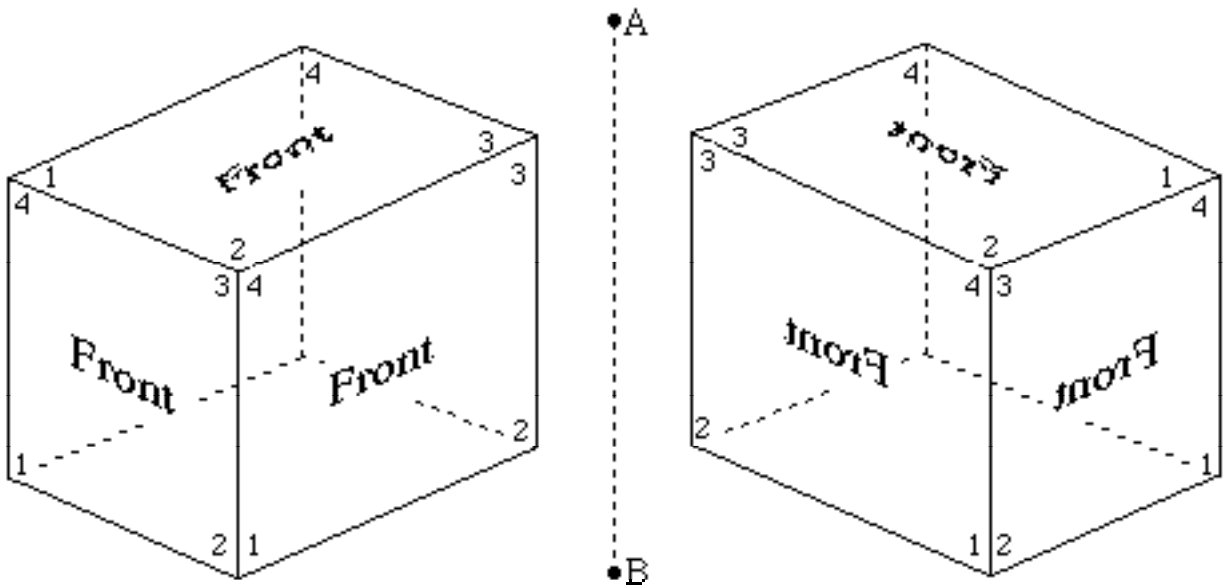


Figure 15: A Box of 3Dfaces Before and After the MIRROR Command. The original box is at left, the mirrored box at right. The points A and B define the line about which the box has been reflected. Polygons making up the original box -- each with vertices ordered $\langle 1,2,3,4 \rangle$ -- are reversed in the mirrored copy. The front sides are now on the interior of the box.

CHANGE is a very versatile command, modifying color, thickness, layer, elevation, and line type. CHANGE COLOR changes not only the selected entity's color in AutoCAD but also changes the color of the polygon produced by the parser (and the texture, if used with the Walkthrough texture information). This will be discussed further in Section 4. CHANGE THICKNESS changes an entity's thickness (the distance the entity is extruded), CHANGE LAYER changes the AutoCAD layer the entity is associated with, and CHANGE ELEVATION moves the object along the Z-axis (The CHANGE ELEVATION command is better classified as entity manipulation than as entity modification, since it moves the entity). And finally, CHANGE LINETYPE changes only the representation of the lines used to draw the object in AutoCAD; as far as the parser is concerned, it does nothing at all.

As a supplementary comment, it should be noted that CHANGE does not work on entities drawn in User Coordinate Systems. There is a modified CHANGE command, described on p. 129 of the *AutoCAD Reference Manual*, called CHPROP, that operates on entities even of differing coordinate systems. It operates on a subset of the CHANGE options (it cannot be used to change an entity's elevation), but the most used subset (color, layer, line type, and thickness).

1.2.2 Entity-specific commands

In addition to the foregoing commands, there are, for the line-based entities (line, arc, polyline), additional modification commands. Three that have proven useful in the architectural modeling I have done are the TRIM, EXTEND, and PEDIT commands. TRIM and EXTEND are complementary commands: EXTEND extends a line, arc, or polyline; TRIM clips those entities. In both commands, you select a line, arc, or polyline in the drawing that either should intersect or does intersect the entity you want to extend or trim; it is marked as a boundary. You then select the entity you want to extend or trim, and if it would intersect or if it does intersect the boundary, the entity is extended or trimmed, respectively. EXTEND is very useful for making walls meet (if you have used lines for walls), and TRIM is useful for clipping away parts of walls that are too long, cutting doors or windows into walls, and so on.

PEDIT, as its name might suggest, is for modifying polylines. There are many ways to modify a polyline with PEDIT; you can move vertices, add vertices, join two

polylines, close a polyline (by connecting the first and last vertices), open a polyline (by disconnecting the first and last vertices), fit curves to the vertices of the polyline, change the width of a polyline, or undo your last edit. It can also be used to modify polygon meshes, though for a large mesh it can be very tedious to edit all the vertices. However, if needed, the command is there.

The EXTEND and TRIM commands are transparent to the Walkthrough parser; if given a trimmed line and an equivalent untrimmed line as input, the parser will convert them to equivalent polygons. PEDIT is equally transparent, although the parser does not handle all PEDIT options (for example, the parser does not fit curves to points, though the AutoCAD PEDIT command does).

1.3 Entity manipulation

The basic AutoCAD entity manipulation commands, MOVE, COPY, ROTATE, CHANGE ELEVATION, and ERASE, perform the tasks of moving, copying, changing the elevation, and erasing entities. These commands are absolutely transparent to the parser; manipulating an AutoCAD entity will manipulate the polygons approximating it in the same way. More details on the use of these commands may be found in the *AutoCAD Reference Manual*.

2 Blocks and Layers

As your drawing becomes more complex, you will find that although one entity is easy to manipulate, collections of entities are less so. It is trivial to manipulate a floor, but somewhat difficult to manipulate an entire building. It is also tedious to copy whole groups of entities one-by-one, especially if you want to modify or manipulate the copies.

AutoCAD has two techniques to help manage the database: BLOCKS and LAYERS. A BLOCK is a collection of primitives that can be manipulated as a single entity; for example, one might want to have a BLOCK called **chair** to place in a model of a restaurant, rather than 50 copies of the polygons constituting the chair. The LAYERS help in a different way; entities (and BLOCKS) can be assigned to different layers that represent different aspects of the model. For example, in a building model, the

furniture might be put in a single layer, the first floor in another layer, and so on. Undesired LAYER can be frozen (using the LAYER command and FREEZE to turn off layers); the frozen layer is not included in the display list, speeding up REGEN commands, and not interpreted as part of the DXF file by the parser, creating a subset of the database.

2.1 Block creation

Blocks can be created in either of two ways: explicitly within a drawing or implicitly by inserting an existing drawing. The BLOCK command lets you pick a group of entities, give them a name and an origin, and then allows you to treat them as a single entity. The origin becomes the center of the block, used to position the block in the drawing (INSERT allows you to rotate the block about its origin after insertion). The block, once defined, disappears from the drawing (the individual entities vanish, and are not replaced by a copy of the block. You must INSERT the block to reposition it within the drawing). Blocks may be treated like entities and assigned an elevation, color, and layer (if there are default elevation, color, or layers set, the block will assume those values).

Blocks may also be made from existing drawings; if the INSERT command is given the name of a .dwg file, that drawing will be inserted as a block with its origin as its positioning point. Models can thus be modularly constructed; you can build up parts of a model at a time, building small objects such as furniture or windows in separate files, and then inserting them into a larger model. This serves a number of purposes. The parts can be modified and updated once, in the smaller file, and updated with

```
INSERT <old_block>=<new_block>
```

For example, to replace the block named “grand_piano” with the model in piano.dwg, you would type:

```
INSERT grand_piano=piano
```

You should replace <old_block> with the block name and <new_block> with the file name (without the file extension). Do not type any spaces about the “=” because AutoCAD will interpret the spaces as carriage returns. This form of the INSERT

command will replace the copies of the old block with the new, although they may not appear that way in AutoCAD immediately (AutoCAD doesn't always update its display list immediately).

The advantage in inserting files as blocks is twofold: you can save time by reusing existing blocks, and it makes the drawings easier to manipulate. Parts can be re-used; elements of one drawing can be put into a different drawing, saving drawing effort. Secondly, it is far easier to find and manipulate a single entity than it is to find and manipulate a dozen, especially in a large and complex drawing. It is convenient to build small parts of a design in separate drawings, and insert the pieces as blocks. Hierarchical models can be built from hierarchical blocks, and be manipulated as a single object.

The INSERT command has several options: insertion point, scale, rotation, and attributes. Obviously, the insertion point is where in the drawing the block should be inserted; AutoCAD will accept a point from the keyboard or a point selected with the cursor in either 2 or 3 dimensions. The scale options allow you to scale the block along the X, and Y, and -- if specially requested with the XYZ option -- Z axes; this can be convenient for bringing small blocks to scale in a drawing, but can be a grand pitfall if you ever need to modify the block, as will be covered under EXPLODE in the block modification section below. You can scale the block uniformly, by setting each scale factor the same, or you can scale differently along different axes. Blocks can be rotated about the insertion point, either under control of the cursor or by entering an angle at the keyboard. Lastly, there are the attributes, if present, that are defined when the block is inserted. Attributes are a more complicated item than really should be covered here; however, they are very useful for assigning special information to a block. For example, Walkthrough assigns lighting and texture information to special lighting and texture blocks. The parser can process only very specific types and formats of attributes, however, so their use is somewhat limited.

2.2 Block modification

Blocks can be modified similarly to the way entities can, with limitations. You cannot use TRIM, EXTEND, or PEDIT on a block, even if the block is made up of entities that can be trimmed, extended, or "peditied." Nor can you STRETCH a block; if you must STRETCH a block, you will need to try reinserting it with different scale factors to

get the effect. You can SCALE and CHANGE a block; SCALE functions the same as with entities but CHANGE works a little differently. If the entities that make up the block were given a color, then that color cannot be changed (except by changing the entity and then reinserting the block). If, on the other hand, the entities were not given a color or were given the color BYBLOCK, then whatever color is assigned to the block is assigned to the entities. You can even make blocks of some colored entities and some BYBLOCK colored entities; this has been used, for example, for wooden chairs with colored cushions, sinks for differently colored bathrooms (but all with the same metal fixtures), and lamps with a set shade color but a varying color for the base.

It is possible to turn a block back into its component entities with the EXPLODE command. An "exploded" block is replaced by the individual polygons that made up the block and the information assigned to the block (e.g., color) is lost. You can then STRETCH, TRIM, EXTEND, PEDIT, etc., the entities as much as you like. There are limitations; a block that was not scaled equally in all directions when inserted cannot be exploded, so if you scaled X by 5 and Y by 2, you cannot explode the block. Also, it is not possible to turn the exploded entities back into the original block, since you may have changed the entities (you can UNDO the EXPLODE command).

2.3 Block manipulation

Block manipulation is much more like entity manipulation than block modification is like entity modification. You can MOVE, COPY, CHANGE (with the limitations noted in Section 2.2), ROTATE, MIRROR, and ERASE blocks in the same way you moved, copied, changed, rotated, mirrored, and erased entities; the commands function exactly the same.

3 Advanced Modelling

Advanced AutoCAD modeling is dominated by the need to manage large and complex models. The modeling techniques and commands do not change; however, the model sizes do. Dramatically. Realistic models tend to be very complex and frequently large both in polygon count and in area; Walkthrough typically works with models of around 5000 - 8000 polygons representing buildings that are between 50 to 150 feet long on a side. For example, compare Figure 16: A Simple Church Hall with Figure

17: A Plot of the Orange United Methodist Church Model. Both model a building that is about 90 x 100 feet, but Figure 16 uses only 7 polygons whereas Figure 17 uses approximately 7500. Clearly, Figure 17 is the much more complicated drawing, and when compared with the real Orange United Methodist Church Fellowship Hall, the more accurate one. However, this complexity also makes it a more difficult file to work with.

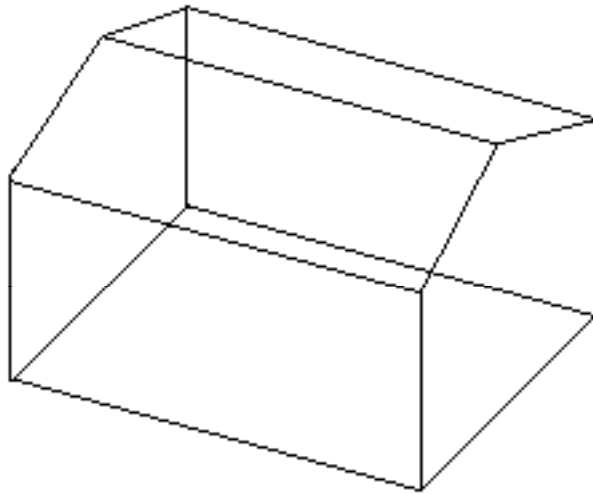


Figure 16: A Simple Church Hall

Figure 17: A Plot of the Orange County United Methodist Church Model

It is not difficult to change a single polygon in a simple model of 100 or fewer polygons; the model is small enough that you can keep track of its underlying structure and how the polygons in the model are related. In a complex model, this kind of knowledge of the structure is not possible: there are too many polygons, and the model represents too large an area. If we wanted to change the color of one polygon of Figure 17 -- granted that we could find it in such a view -- we would have a lot of difficulty picking the one polygon from the thousands on screen. You can't even tell which polygons belong to the upstairs and which belong to the basement. Also, it becomes difficult to get any sort of accuracy in picking points in the model -- much less figuring out what is what -- unless you zoom in on a very small section. Some form of order is needed to manage the polygon chaos, and although it is unlikely we will be able to reduce the total number of polygons in the model (and still maintain the same degree of realism), we can reduce the amount of complexity. By using blocks and layers, we can impose order on the model.

3.1 Blocks

Blocks are very useful in large models. Built separately from the main drawing, a block is generally a small set of polygons in a relatively simple figure, something that you can fit on the screen at a reasonable drawing scale. The main drawing -- the one the block will go in -- may be 100 feet on a side, which works out to about 10 feet in the model to one inch in the drawing window; it becomes very difficult to work accurately under these conditions. In the block, the scale tends to be less than 1 drawing-scale foot to the inch on the window. It is far easier to work in this scale.

It is also easier to manipulate the drawing of the block; with fewer polygons, it takes less time to regenerate a small drawing, to load a small drawing, to save a small drawing.

The best reasons for using blocks, however, have nothing to do with either of the above points; these are that blocks are reusable entities -- objects that can be moved, changed, and rotated as a single entity, inserted at will in any drawing. If you want to rearrange the furniture in a room, you need only move the chairs, the pianos, and the tables, not the polygons that make them up. Selecting any polygon in the block selects the block; you need not hunt for each polygon making up the chair. If you decide you

want a different chair, or a corrected version of the old chair, an INSERT <block>=<new block> will replace all instances of <block> with <new_block>; you need not erase every polygon of every chair. Creating a model becomes a quicker task: any repeated object -- door, window, chair, table, light fixture -- can be a block and need be drawn only once. Blocks can be inserted within blocks; a chair can be inserted in a house, the house in a neighborhood, the neighborhood in a city, and so on.

By using blocks, you make the drawing more modular and end up, consequently, with fewer entities that can be individually changed. There are still the same number of polygons in the drawing, of course; however, they are at least clustered. A further reduction in complexity can be made with the use of layers to order the drawing.

3.2 Layers

The advantages of using layers are twofold: you can reduce the number of things you see on the screen, and you can reduce the number of polygons produced by the Walkthrough parser. When you freeze a layer, you turn it off, in effect: it is not drawn, and it is not even considered to be part of the display list. Regenerating the screen is faster, since the entities in the layer are not considered to be part of the displayable drawing. Polygons that otherwise would be in the way are not displayed. So by putting the entities into different layers, you can control which parts of the drawing you will be working with: you can turn off the ones that will not be needed, working only with those sections that will be.

The LAYER command offers several options: ?, MAKE, SET, NEW, OFF, ON, COLOR, LINETYPE, FREEZE, and THAW. In AutoCAD, there is always a current layer. When an entity is drawn, it is assigned to whichever layer is current at the time of its creation. The entity's layer can be subsequently changed, but it is initially the current one. You can make a layer current with LAYER SET. You can create a new layer and make it the current layer by using LAYER MAKE, or by using LAYER NEW to create the layer and SET to make it current (the name of the current layer is displayed in the upper left-hand corner of the drawing at all times). LAYER ? displays a list of all existing layers. Layers can be turned off and on, or frozen and thawed. ON and THAW undo OFF and FREEZE; layers can only be turned on or thawed if they have been previously turned off or frozen. Layers turned off or frozen are not shown. Frozen layers are completely removed from the display list, making drawing regeneration times faster;

off layers are kept in the display list and culled at the last step, after transformations but before rendering.

The parser recognizes frozen layers, but does not distinguish off layers from ordinary layers. Layers frozen before a DXFOUT will be not be turned into polygons; all other layers (including off layers) will be transformed into polygons. This can be used to transform subsets of larger models; if you want a certain subset of the drawing, put the undesired entities into frozen layers and do a DXFOUT. The file produced from that DXF file will not include anything from the frozen layers.

4 Converting from AutoCAD Form to Polygons

Conversion from the AutoCAD format to the form I will refer to as a *poly* file is done with the Walkthrough AutoCAD-to-polygon parser, a program called acadtopoly. It accepts a .dxf file (so you will need to do a DXFOUT before running it) and a color-table file (.ctab file) as input and writes the polygons to the standard output. The .dxf file is ASCII representation of the model and can be treated as a black box: it is not necessary for the user to know anything about the contents of the file to convert it to polygons (Interested users can find out more about the format in Appendix C of the *AutoCAD Reference Manual*). On the other hand, the .ctab file, the parser, and the .poly file may require some user-interaction or modifications, and will be individually discussed.

4.1 Color-table

A color-table file has the form

x r g b (t)

where x is the index in the file, r g b is an rgb (red green blue) triplet, and the optional t is a texture number. The index is the same as the color numbers assigned in AutoCAD: the AutoCAD color is mapped to the color represented by the rgb triplet. AutoCAD allows up to 255 colors, so you can have up to 255 colors in your model. Producing the colors for the color table is not as easy as in AutoCAD; the rgb triplet as a way to specify a color is a difficult concept to grasp. Knowing that black is 0 0 0 and