

教育用3次元CADシステムの開発

新津 靖^{†1}

東京電機大学・情報環境学部

Development of 3D-CAD System for the Education of Descriptive Geometry

YASUSHI NIITSU^{†1}

School of Information Environment, Tokyo Denki University

In a class of the design and drawing, almost University's classes adopt the 3D-CAD system as an educational tool. Many Universities wish to use 3D-CAD for the descriptive geometry education. However, there is no 3D-CAD system for educational use, and they use 3D-CG or 3D-CAD of the professional production use. The author has been developing the 3D-CAD system for educational use. In this paper, the author introduces developed new software (3D-CAD) with many new functions.

1. はじめに

コンピュータの進歩に伴い、図学の後にくる製図教育では、多くの大学でコンピュータを使用した CAD 教育に移行しており、図学教育もコンピュータを援用した教育に移行する必要性が叫ばれている。しかし、広く一般の大学で使用される図学教育用のソフトウェアはない。著者は、大学1年次生の図学教育用3次元CADを開発してきた。本研究では、従来のものに、多くの機能を追加した新規ソフトウェアを開発した。^{1),2),3)}

著者は1993年より図学教育用ソリッドモデラーの開発をスタートし、1998年より東京電機大学工学部の1年次生の授業で使いはじめた。2001年にはソリッドインタプリタ Version 3 として販売を開始した。このソフトウェアは約40種類のコマンドを並べたテキスト形式のファイルを作り、そのコマンド列の記述にそって3Dモデルの生成とモデルの動作および表示を行うものである。BASIC言語のようなインタプリタに近い動作であることから「ソリッドインタプリタ」と命名した。コマンドの中には繰り返しや条件分岐などの実行制御コマンドもある。しかしながら、Version 3 では変数や式の記述などに制限があり、多くの改良の余地があった。そこで、2000年からVersion 4の開発に着手し、これらの問題点の解決を試みてきた。本報告では開発中の教育用3D-CADの概要と機能強化の内容を示す。^{4),5)}

2. データ構造とコマンドスクリプト

2-1 立体モデルのデータ構造

最近の3D-CADの多くはデータ構造としてHalf Edge構造(B-Reps)を採用するようになっているが、本教育用

3D-CADは立体モデルのデータ構造としてWinged Edge構造を採用している。^{1),2),3)}1本の稜線(Edge)の往復操作を前後の稜線番号と左右のループ番号の関係で分けるため、ループ検索が若干複雑になり実行速度がHalf Edge構造に比べ劣るとされている。しかしながら、メモリ効率が良いことや稜線と対応するデータが1対1に対応するなどわかりやすい面もある。立体オブジェクトが有する代表的な立体表現用データを以下に示す。

- (a) 頂点 : プロパティ, 頂点座標, 法線ベクトル
- (b) 面 : プロパティ, 親ループ
- (c) ループ : プロパティ, 子ループリスト
- (d) 稜線 : プロパティ, Winged Edge 構造
- (e) 面グループ, 稜線グループ

各要素データのもつプロパティは、色や生成時の条件、曲面・曲線の違いなどを意味する。穴のある面を表現するために面とループデータを分けており、1つの面に複数のループを設定することで、穴のループを表現している。また、面と稜線はグループ化しており、生成時の面や稜線のデータを分類している。Fig.1に円柱生成における面と稜線のグループ化の例を示す。円柱の場合、面および稜線は3グループに分類される。Fig.1の例では円柱の側面を構成する

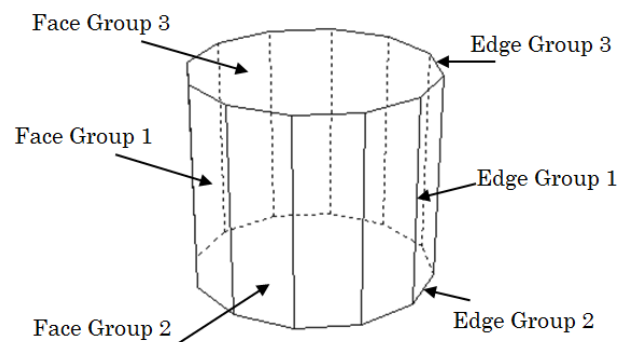


Fig. 1 Grouping of the faces and edges of Cylinder Object.

^{†1} 東京電機大学・情報環境学部
School of Information Environment, Tokyo Denki University

12 面が曲面として FaceGroup1 に所属し、底面と上面は 1 面でそれぞれ FaceGroup2,3 に所属する。稜線は縦の 12 本の稜線が EdgeGroup1 に、底面の 12 本の稜線が EdgeGroup2 に、上面の 12 本が EdgeGroup3 に所属する。

立体を構成する要素が番号・ポインタにより参照されるのに対し、立体オブジェクト、グループオブジェクト、軸オブジェクトおよびカメラオブジェクトは名前で参照される。グループオブジェクトは立体や子グループをまとめる機能であり、軸オブジェクトは立体やグループに移動や回転動作を設定するオブジェクトである。カメラオブジェクトは表示用の視点や注視点位置、表示モードなどのデータを格納するオブジェクトである。

2-2 コマンドスクリプト

本教育用 3D-CAD はテキスト形式のコマンドを順次実行することで立体モデルの生成、グループ化、表示などを行うことができる。コマンドは「基本立体生成」、「立体操作」、「表示」、「その他」の 4 種類に分類でき、基本立体生成だけでも 17 種類あり、各コマンドはオプション指定で複数の種類の立体を生成することができる。たとえば球体を生成する「Sphere」コマンドでは半円の回転掃引による球体以外に正 12 面体、正 20 面体、32 面体、80 面体の 5 種類の球状体を生成できる。Table 1 に基本立体生成コマンドの書式と機能を、Fig.2 に 15 種のコマンドで生成される基本立体の例を示す。Version 3 では 13 種類であったが、Version 4 で「CubeStep」、「ConeStep」、「RoundStep」、「Plate」の 4 種類を追加した。Plate コマンドは数式で立体を生成するコマンドである。「立体操作」コマンドには立体の移動と回転、アフィン変換、集合演算などの操作と、コピーやグループ化などがある。「表示」コマンドにはカメラオブジェクトの生成、画面分割、立体表示、軸表示などのコマンドがある。「その他」のコマンドには、繰り返しや条件分岐などのプログラミング的要素のコマンドやファイル出力コマンドなどがある。コマンドは全部で 46 種類あり、オプション指定により、1 つのコマンドに多くの機能が付加されている。

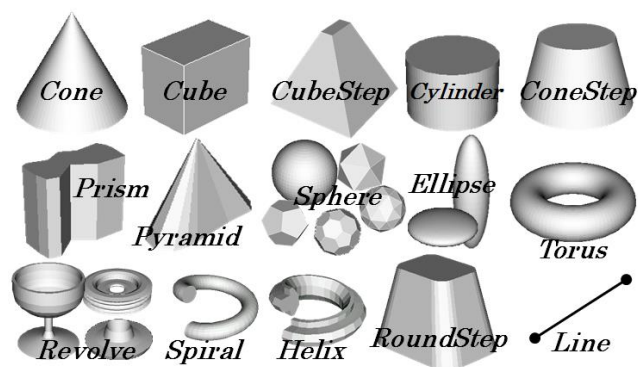


Fig.2 Primitive models made by 15 kinds of commands.

Table 1 Primitive creation command list.

生成立体	コマンド	主なパラメータ
円錐	Cone	<op>立体名, 分割数, 半径, 高さ
直方体	Cube	<op>立体名, X,Y,Z 長さ
円柱	Cylinder	<op>立体名, 分割数, 半径, 高さ
直方台	CubeStep	立体名, X1,Y1, X2, Y2,Z 長さ
円錐台	ConeStep	立体名, 分割数, 半径 1,2, 高さ
滑らかな台	RoundStep	立体名, X,Y,Z 長さ, 半径
三角形面	Triangle	立体名, A 座標, B 座標, C 座標
回転楕円体	Ellipse	立体名, 分割数, XY 半径, XZ 半径
直線	Line	立体名, 始点座標, 終点座標
多角柱	Prism	<op>立体名, 高さ, 座標点列
多角錐	Pyramid	<op>立体名, 高さ, 座標点列
回転体	Revolve	<op>立体名, 分割数, 座標点列
球体	Sphere	<op>立体名, 分割数, 半径
円環体	Torus	立体名, 分割数, 大半径, 小半径
螺旋体	Spiral	立体名, 大半径, 小半径, ピッチ
螺旋体	Helix	立体名, ピッチ, 座標点列
曲面・曲線	Plate	<op> 立体名, 厚さ, 式数, x=数式; y=...; z=...;

3. 立体モデルの生成と操作方法

3-1 立体モデルの生成

立体モデルはテキスト形式のコマンドを実行して、基本立体を生成し、さらに生成した基本立体に移動や回転やアフィン変換を施すことで位置や形を変えることができる。また、複数の立体を集合演算で合成することもできる。集合演算には和集合、差集合、積集合の 3 種類があり、非常に多様な立体モデルを生成することができる。2 つの円柱に 3 種の集合演算を施した例を以下に示す。「#」以降はコメントである。

```

new                                # 初期化
Eye eye0 100,100,82, 0,0,0, 3, 1,4,0,0    # 視線生成
Cylinder y0 60, 10, 30                # 円柱 y0 の生成
Move y0 0, 0, -15                     # 円柱の移動
y1 = y0                                # 円柱のコピー
Rotate y1 0,0,0, 1,0,0, 90            # 円柱 y1 の回転
Xcolor y0 255, 255, 255                # 着色 (白)
Xcolor y1 255, 128, 0                  # 着色 (青)
Obj1 = y0 + y1                          # 集合演算(和)
Obj2 = y0 - y1                          # 集合演算(差)
Obj3 = y0 * y1                          # 集合演算(積)
Move Obj2 -25,-25,0                    # 立体の移動
Move Obj3 25, 25,0                      # 立体の移動
Display eye0 Obj1,Obj2,Obj3            # 表示
..... (1)

```

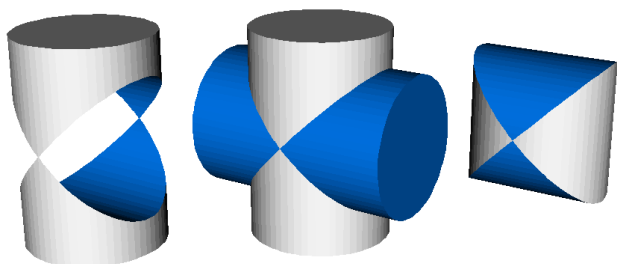
上述のスキプトの実行により Fig.3 のような3つの立体 (Obj1, Obj2, Obj3)が生成され表示される。集合演算により面や稜線の数は増え、同時に面グループや稜線グループも増える。集合演算は面や稜線のプロパティを継承するため、Fig.3 に示すように、2つの立体の混色立体が生成される。集合演算は2つの立体間の演算であり、複数の立体を含むグループ間の集合演算はできない。

3-2 立体モデルのグループ化

立体モデルのグループ化は Group コマンドと範囲指定で実行できる。子グループ、孫グループとフォルダ構造と同様の構造を生成することができる。また、グループは、コピー、移動や回転、アフィン変換も可能である。以下のコマンドリスト(2)を実行することで Fig.4 に示すようなメンタルローテーション問題を生成することができる。

```
New #初期化
Eye eye0 100, 180, 81, 0, 0, 20, 3,1,6,0,1 #視線生成
group Model1 { #Model1 の生成
  Cube -c q0 10, 10, 10 #立方体 q0 の生成
  Xcolor q0 64, 32, 64 #色設定
  Fcolor q0 224, 255, 200 #面の色設定
  q1=q0; move q1 0, 0, 10 #コピーと移動
  q2=q1; move q2 0,0,10 #コピーと移動
  q4=q2; move q4, 0, 10, 0 #コピーと移動
  q5=q4; move q5 0, 10, 0 #コピーと移動
  q6=q5; move q6 0, 0, 10 #コピーと移動
  q7=q6; move q7 0, 0, 10 #コピーと移動
  q8=q5; move q8 -10, 0, 0 #コピーと移動
} #Model1 の終了
Model2 = Model1 #グループのコピー
Affine Model1 0,0,0; -1,0,0; 0,1,0; 0,0,1 #アフィン変換
rotate Model2 0, 15, 25, 1, 1, 0, 50 #回転
rotate Model2 0, 15, 25, 0, 0, 1, 70 #回転
move Model2 30, 0, 8 #移動
move Model1 -30, 0, 0 #移動
Display eye0 ** #すべて表示
```

(2)



(a) Subtract (b) Add(Union) (c) Intersection

Fig. 3 Three 3D-models made by boolean operations.

リスト(2)の 16 行目がアフィン変換コマンドであり、Fig.4 の正解は「Mirror and Rotate」になる。15 行目は Model1 を Model2 にコピーするコマンド行である。この実行により新しい Model2 グループが生成され、その中に Model1 と同じ名前の立体がコピーされる。すなわちグループが複製される。ただし、グループ直下の立体がコピーされ、子グループや孫グループはコピーされない。子グループや孫グループも含めた複製は、テキストコピーを使って行う必要がある。

3-3 曲面および曲線の立体モデル生成

Table 1 の曲面および曲線の立体モデルは数式を使って生成する。数式には四則演算に加えて三角関数や指数関数などが使用できる。A~Z と a~z の一文字が変数として使用でき、変数 x, y, z は曲面や曲線の座標を表現するのに使われる変数である。Plate コマンドのパラメータ書式を以下に示す。

-f/-l 立体名, 式数,厚さ/太さ, x=数式; y=数式; z=数式; ...

Fig.5 および Fig.6 は Plate コマンドにより生成した立体モデルの例を示している。Fig.5(a)は 5 式による表現で、(b)は 3 式による単純なパラメータ設定である。-f オプションの面生成では最後の 2 式が、範囲指定を表しており、3つの数値(初期値, 最終値, 増分)をコンマで区切って表現する。Fig.5 の2つのモデルのスキプト記述を以下に示す。(Fig.5a)では、変数 r, t を媒介変数として使用しており、式の数は 5 である。はじめの3式は x,y,z の定義でなければならない。最後の2式は曲面モデルの場合は掃引範囲指定でなければならない。

(Fig.5a) Plate -f Body 5, 0.02; z = 2.4*(exp(-r*r/0.6));
x = r*cos(t*2); y = r*sin(t*2);
r = 0.01, 2.5, 0.12; t = 0, 2.5, 0.08

(Fig.5b) Plate -f Body 3, 0.001; z = 0.6*(sin(x*2) * cos(y*3));
x = -1.6, 1.6, 0.08; y = -1.6, 1.6, 0.08

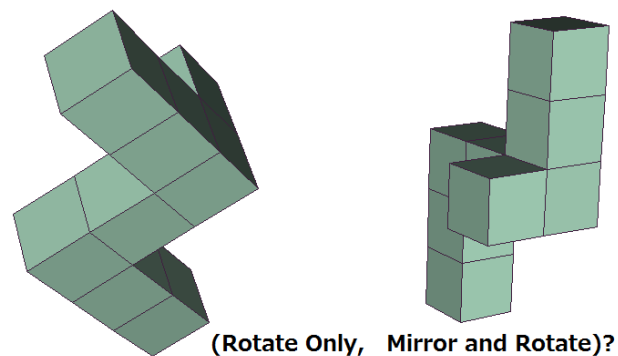


Fig.4 Sample of grouping operation and affine transfer.

一方、曲線モデル生成では、最後の1式のみが掃引範囲指定でなければならない。Fig.6の2つの曲線モデルのコマンドを以下に示す。曲面モデルでは式の数に3～5であるのに対し、曲線モデルでは3～4である。

(Fig.6a) Plate -l XBody 3, 0.8;
z = 10*(exp(-x*x/60)*cos(1.0*x));
y = 0; x= -20, 20, 0.2

(Fig.6b) Plate -l XBody 4, 0.01; z = t;
x = (3-t)*cos(t*6); y = (3-t)*sin(t*6);
t= -2, 2, 0.02

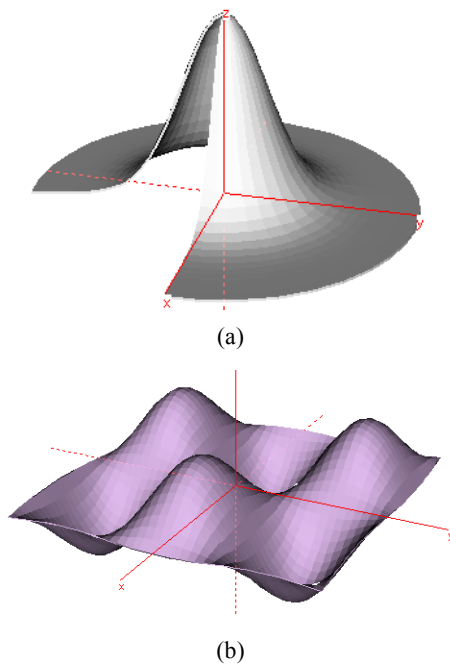


Fig. 5 Solid model with curved faces.

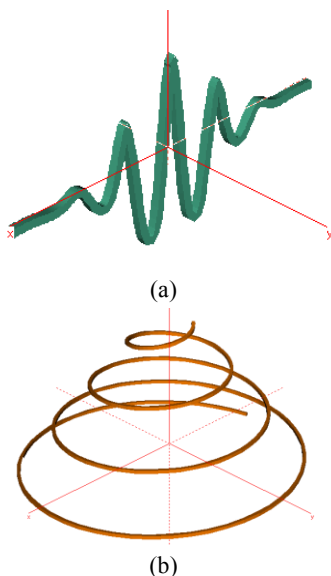


Fig. 6 Solid models of curves.

3-4 変数および数式を用いた立体モデル生成と操作

前節で曲面および曲線モデルの生成を示したが、モデル生成におけるコマンドスクリプトに変数と数式を使うことができる。Version 3では単純な四則演算のみ可能であったが、Version 4では条件式も含めて書式を通常のC言語に合わせた。さらに、コマンドの数値パラメータ部分のほとんどの場所に数式を使うことができるようにした。ただし、変数は1文字だけで、A～Z, a～zの52個で、すべて倍精度浮動小数点数である。「New」初期化コマンドでゼロに初期化されている。1文字のトークンを変数としているため、立体名やグループ名などには2文字以上の識別名を指定する必要がある。たとえば、「A」という名前の立体を作ることはできない。また、数式ではタブやスペースを式の間に挿入できるため、Version 3のようにスペースをコマンドパラメータの区切り記号として使えない。このため、Version 4ではコマンドの区切り記号を「,」とした。なお、「;」は行の区切りである。(Version 3はスペースまたは「,」を区切り記号としていた)以下にパラメータに数式を使用したConeコマンドとMoveコマンドの記述例を示す。

```
R = 20 ; V = V + 0.1 ; T = T + 0.8
Cone body 12, R*(1+cos(T*3.141592/180.0)), 0.1*T
Move body R*sin(V), R*cos(V), T
```

3-5 繰り返しや条件分岐を用いた立体モデル生成と操作

従来の「Repeat」コマンドに加えて「For」および「While」コマンドを追加し、より柔軟な立体モデルの生成と操作を可能にした。以下に「For」コマンドの例を示す。多重繰り返しの例である。

```
for (j=0; j<5; j++) {
  for (i=0; i<5; i++) {
    Cylinder yy(j)_i 16, 4, (i+1)*(j+1)
    Move yy(j)_i -20+j*10, -20+i*10, 0
    c = 90+i*j*10
    Xcolor yy(j)_i 0,0,0; Fcolor yy(j)_i c, c, c
  }
}
```

このコードの実行により25個の円柱が生成される。Version 4では名前フィールドの書式・解析を設定した。上述のCylinder, Move, Xcolorコマンドの生成・操作立体名に「yy(j)_i」の書式を指定している。コマンド解析ルーチンでは、「(変数)」部を整数文字列に変換する書式を採用した。すなわち、上記のコードの実行により次の25個の名前の立体が生成される。

```
yy0_0, yy0_1, yy0_2, yy0_3, yy0_4
yy1_0, yy1_1, yy1_2, yy1_3, yy1_4
yy2_0, yy2_1, yy2_2, yy2_3, yy2_4
yy3_0, yy3_1, yy3_2, yy3_3, yy3_4
yy4_0, yy4_1, yy4_2, yy4_3, yy4_4
```


これらの立体を表示すると Fig.7 の画像が得られる。立体の色も変数で設定されている。

For コマンドの繰り返し条件にあるようにC言語の通常の比較演算と論理演算が式に使用できる。While コマンドや条件分岐である if-else コマンドでも同様に条件式に、「==, !=, <, >, <=, >=, && ||」の比較演算と論理演算が使える。繰り返しや条件分岐に、数値入力を行う Input コマンドやリアルタイムキーションを行う Key 関数を使うことで、インタラクティブなモデル生成と操作を行うことができるようになった。

3-6 特殊な関数とコマンド

本来はソリッドモデラーや 3D-CAD には不要であるが、本ソフトウェアを使用した実習では、学生は 3D アニメーションを課題作品として制作することが多い。そこで、より柔軟なモデル操作ができるように、4 種類の関数と 2 つのコマンドを加えた。それらの機能を Table 1 と Table 2 に示す。

Table 2 Introduced special functions.

関数名	記述	機能
乱数生成	random(n)	0~n-1 の整数の乱数を返す
乱数生成	rand(x)	0.0~x の乱数を返す
キー入力	key(k)	k 値のキーが押されていれば 1 をそうでなければ 0 を返す
日時取得	time(N)	N=0 のとき経過時間を N=1 のとき経過日数を返す

Table 3 Introduced special commands.

コマンド	機能
Sound	WAV ファイルの読み込みと音声出力の制御 Sound -L N(0~31), FileName : 読み込み Sound -P N : 再生開始, Sound -S : 停止
Input	数値入力フォームが表示されて入力できる 以下は変数 X,Y,Z に入力される例 Input (-n) 1; X Input (-n) 1; "X Value?", X //メッセージ付 Input (-n) 3; X; Y; Z

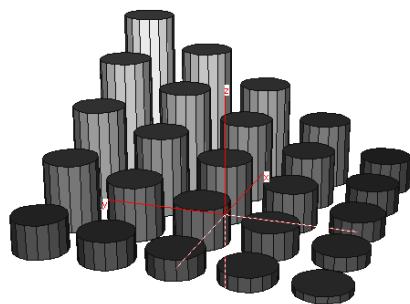


Fig. 7 Solid models made by FOR command.

4. 立体モデルの表示機能

4-1 面表示

本ソフトウェアは約 20 年前から開発をしているもので、当時は Windows もなく、グラフィック表示も 2D 表示のみであった。そのため、当初は面表示がなく、稜線表示のみであった。Windows95 に対応した時点で、Canvas クラスのポリゴン表示を用いて面表示は行っている。投影変換と色計算も独自のコードで行っており、照明設定などは必要としない簡易の表示方法である。ポリゴン表示のため、穴のある面の表示は 1 つの穴まで可能で、2 つ以上は表示エラーとなる。Fig.8 に複数の穴のある立体を表示した例を示す。稜線表示が正確に表示しているのに対し、面表示では 5 つの穴のある面を表示すると表示エラーが発生する。この問題は、OpenGL のテッセレーション機能で解決できるが、現在コードを開発中である。さらに、面表示では面を遠い順に表示しているため、面の重なりを正確に表示することができない場合がある。

4-2 稜線表示と陰線処理

完全な稜線表示と陰線処理は非常に複雑な計算を要する。処理時間も面表示に比べて長くかかる。本 3D-CAD では、複数の陰線表示機能を有している。複数の立体も含めた完全な陰線処理がもっとも多くの計算処理を必要とする。簡易陰線処理は面方向のみを調べ、視線に向いている面を表示する。計算量は少なく済むが、複数の立体や凹曲面などの表示では表示エラーが生じる。Fig.9(a)が簡易陰線処理モードによる表示例である。Fig.9(b)は同じ立体を完全陰線処理モードで表示した図である。同じく Fig.9(c),(d)も完全陰線処理モードの表示例である。Fig.9(d)は曲面上の稜線も表示するモードである。穴の形状や奥にある円錐の稜線も正確に陰線表示できていることがわかる。

4-3 三面図表示

CAD では図面表示に 3 面図を使用する機会が多い。視線情報には、3 面図で表示するモードがある。Fig.10 はロボットの胸部から頭部を 3 面図+透視投影図で表示した例である。3 面図表示に対しては、陰線表示だけでなく面表示も可能であり、表示方法は 140 種類以上のバリエーションがある。

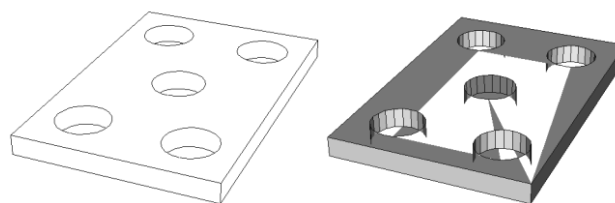


Fig.8 Display error of face display mode

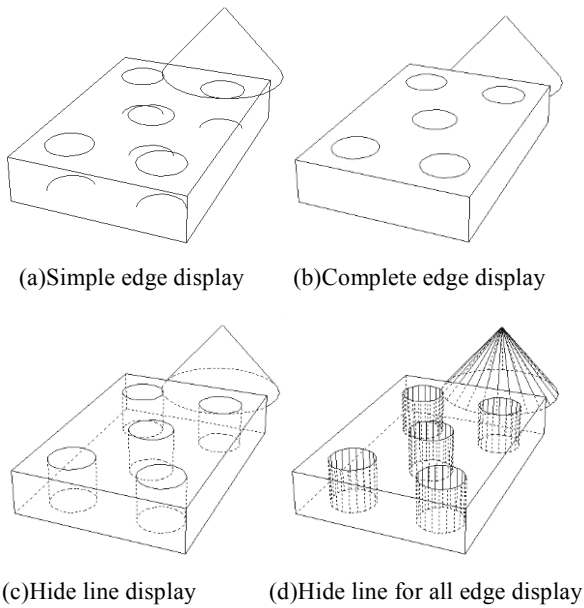


Fig. 9 Image by edge display modes

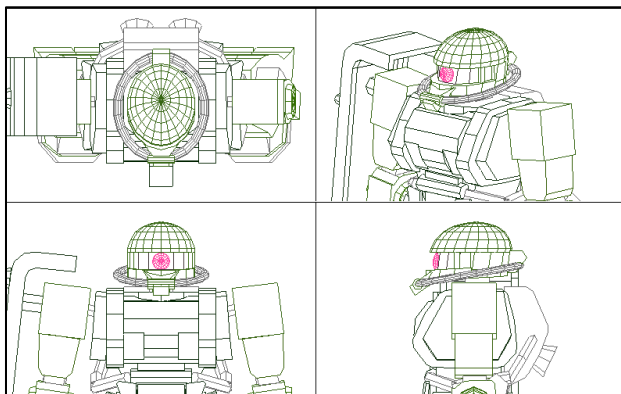


Fig.10 Three directional views and perspective view.

5. 実行画面とツール群

5-1 実行画面

本ソフトウェアを実行するとコマンドスクリプト入力用のエディタが起動する．ここにコマンドをキー入力し実行ボタンを押すことで，モデル生成や表示機能を実行でき

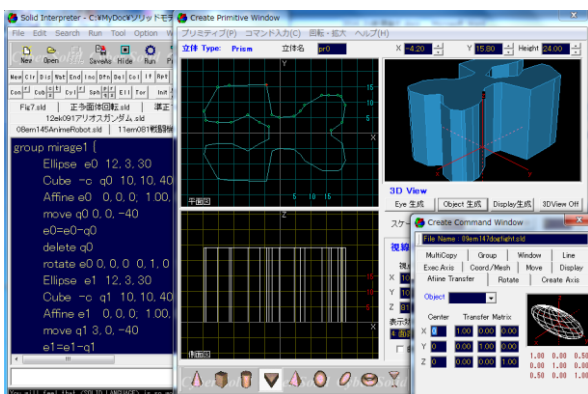


Fig.11 Executive state of 3D-CAD

る．コマンドを直接キー入力して実行することもできるが，複数のパラメータやオプションを間違いなく入力することは困難である．特に初心者にとってはキー入力自体が困難である．そこで，本ソフトウェアでは，コマンド入力をビジュアルにサポートする次の3つのツールを用意している．

- (a) プリミティブ生成フォーム (Fig.12)
- (b) コマンド生成フォーム (Fig.13(a))
- (c) オブジェクトインスペクター (Fig.13(b))

5-2 プリミティブ生成フォーム

このツールは基本立体をビジュアルに生成できるツールである． Fig.12 に任意底面の角柱を作成中の画面表示を示す．左側の上下の画面上をマウス操作することで，ほとんどの種類の基本立体を生成することができる．下部には作成したい基本立体の選択ボタンがあり，右上画面には制作途中の立体の表示が自動的になされる．この右上の画面は右下で指定した視線情報に合わせて表示されるようにできている．立体のパラメータは数字で入れることもでき，その値は3つの各画面に反映される．コマンド生成用のボタンを押すことで，画面で確認した基本立体を生成するコマンドがメインフォームのテキストエディタに転送される．

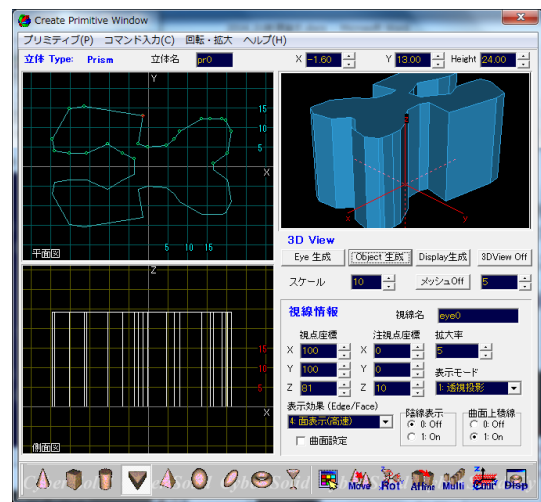
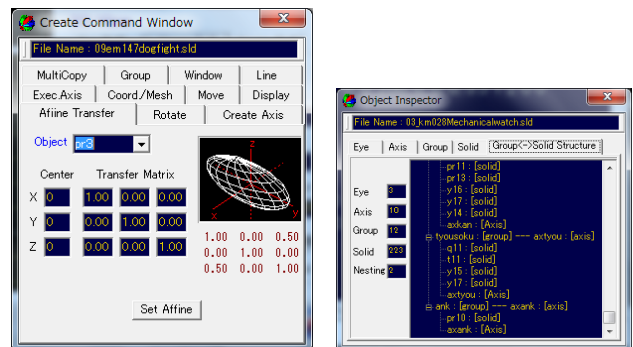


Fig.12 View of primitive creation tool.



(a) Command creation form (b) Object inspector

Fig.13 Command creation form and Object inspector.

5-3 コマンド生成フォームとオブジェクトインスペクター

コマンド生成フォームは基本立体生成コマンド以外のコマンドを生成するためのツールである。Fig.13(a)はアフィン変換コマンドを生成するタブを開いており、立体名やマトリクスデータを入力しボタンを押すと、指定したパラメータのアフィン変換コマンドをメインフォームのエディタに転送できる。回転コマンド、グループコマンドや軸コマンドなど多くのオプションを有するコマンドの生成には必須のツールである。

オブジェクトインスペクターは、コマンドスクリプトを実行して生成した立体モデルの状態を表示するツールである。Fig.13(b)は、グループ構造をツリー状に表示している画面である。他のタブを選択すると、立体や視線情報の現在の状態を数値で見ることができる。たとえば立体の場合は、要素の数や親グループや軸設定の状態だけでなく、体積や表面積も見ることができる。

これらのツールだけでなく、メモリ状態をチェックしたり設定するフォームや CAD データ出力を設定する多くのフォームがある。

6. サンプルモデル

著者の所属する大学の1年次生に対してワークショップと呼ばれる授業において半期の半分である6週（週1回2コマ）の実習授業を実施してきた。図形に対する基本的な事項の解説と描画実習に1コマ、4コマを本3D-CADの使い方と紙に書く実習を行い、最後に自由課題の発表会を行っている。Fig.14に発表会で提出された立体モデルの一部を示す。モデルはアニメーションとして動作するものがほとんどである。



Fig.14 Sample models made by university students.

7. おわりに

本ソフトウェア開発は、2001～2003年の経済産業省未踏ソフトウェア創造事業の一つとして開発が進められた。

Solid Interpreter という商品名で販売されたが、十分な売り上げが得られず、著者が主に使用してきた。この間、モデラーのエンジンを使用したツールの製作なども行ってきたが、今後は、フリーソフトとして多くの教育機関で使ってもらうことを目標に再度開発を進めていくことにした。特に、初等・中等教育向けのより簡易なソリッドモデラーあるいは3D-CADを開発していく予定である。

参考文献

- 1) 新津 靖, "図学およびコンピュータグラフィックス教育のための3次元立体生成ソフトウェアの開発", 情報教育方法研究, Vol.2, No.1 (1999), pp.7-12.
- 2) Yasushi Niitsu, "Solid Modeling with Polygon Data of Mathematica", Proc. of 4th Int. Mathematica Symp. (Symbolic Computation), (2001), pp.369-374.
- 3) 鳥谷浩志・千代倉弘明, "3次元CADの基礎と応用", 共立出版, 1991.
- 4) 新津 靖, "教育用3次元CADシステムのSTL出力機能", 日本図学会中部講演会, 2013-9.
- 5) 新津 靖, "高機能教育用ソリッドモデラーの開発", 日本図学会秋季講演会, 2013-11.