

## パーソナルコンピュータによる第三角法図面から 立体投影図への変換処理法について

山 田 健 治

### Method of Transforming from a Plane Figure to a Solid Figure with a Parsonal Computer

Takeharu YAMADA

This paper explains the graphic processing method adapted to produce three dimensional displays from objects represented by only two dimensional planes.

The x-y values at each vertice of the plain figure in front, plane and side elevation are input by computer key.

The data are checked for errors and reduced to three dimensional x-y-z values in the computer.

As a result, this method enables the personal computer to be used for more educational work in machine design.

#### 1. はじめに

最近、機械科系を有する高校や大学ではパーソナルコンピュータ（以下パソコンという）を用いて、簡単なCAD教育を実施しているところも増えてきている現状である。

本論文では、図形情報処理教育の一つとしてこのパソコンを用いて、機械製図で用いられている第三角法によって描かれた三面図（正面図・平面図・側面図）より、図を構成する頂点の二次元座標データをキーボードにて読み取らせ、これらの座標をさらに補正を行い、三次元空間上に立体を表現する簡易図形情報処理法を試みたので、この図形処理法のアロリズムおよびプログラムを報告する。

#### 2. 図形処理装置及びデータ入力法

図形処理装置としては図1に示すように、使用したパソ

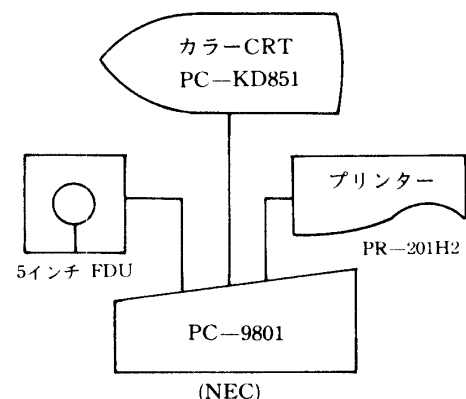


図1 処理装置のハードウェア

コンはNEC PC-9801 (CPU 16bit・ROM 96kB・RAM 640kB) と14インチカラーディスプレイおよびドットプリンターの周辺機器を使用した。また、プログラムに使用した言語は、N88-BASICである。三面図からの二次元座標データは、テンキーによる入力法を採用した。

### 3. 三次元化処理法

一般に三次元物体を構成している面には、平面・曲面さらにこの両者の組合せによるものから成る。ここで扱う物体の構成面は平面で構成する物体のみとする。

直方体は平面の集合体である。従って、平面は稜線によって囲まれ、さらに稜線はその両端の点(頂点)によって与えられる。よって、これらの組合せデータは階層配列型データ構造となる。

また、現在のところ、三次元の形状をコンピュータが理解できる三次元モデルには、ワイヤフレームモデル・サーフェイスモデル・ソリッドモデルの三つがあるが、ここでは最も簡易な図形、ワイヤーフレームモデルを採用する。図2は図形処理を行なうフローチャートを示す。

### 4. 図形処理のアルゴリズム

図3は直方体の三面図を図示したものであるデータ入力は正面図、側面図および平面図の各頂点に独立した番号を付記しておき、その頂点の二次元座標データ( $x-y$ )を読み取る。

各稜線の始点および終点に関する三次元座標データを求めるアルゴリズムについて以下に説明する。

正面図について

図3に示す線分1.2について

- (1) 始点1の座標データ( $x, z$ )の $z$ 座標値に対し、側面図よりその値と等しい $z$ 座標値の頂点を探す。始点1の座標データは(50, 50)であり、この $z$ 座標値(50)である点を探すとP5(120, 50)とP6(90, 50)の二点を得られる。
- (2) 始点1の座標データ( $x, z$ )の $x$ 座標値に対し、平面図よりその値と等しい $x$ 座標値の頂点を探す。平面図より $x$ 座標値(50)である点を探すと、P6(50, 120)とP12(50, 90)の二点を得られる。

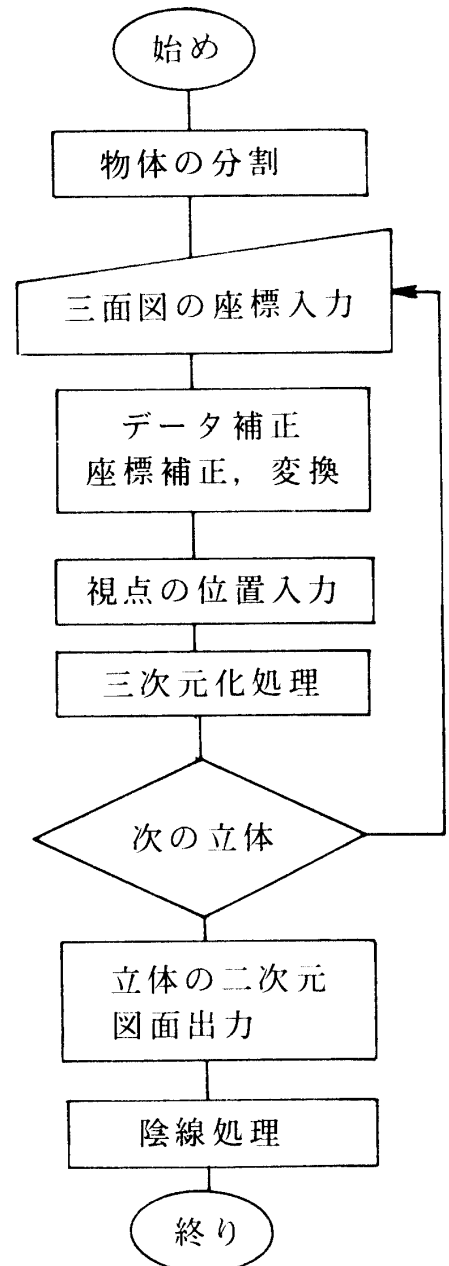


図2 図形処理フローチャート

(3) (1), (2)で見出だされた頂点に対し, 側面図と平面図から  $y$  座標値の等しい点の組合せを求める。 $y$  軸の支点の座標は三面図の座標系より  $(90, 90)$  なので, まず側面図から得られた値を補正する。補正值は側面図支点  $(90, 0)$  より,  $P5$   
 $(y, z) = (120, 50) - (90, 0)$   
 $= (30, 50)$ ,  $P6 (y, z) = (90, 50) - (90, 0) = (0, 50)$

次に平面図から得られた値を補正する。

平面図支点は  $(0, 90)$  であるから,  $P9$   
 $(x, y) = (50, 30)$ ,  $P12 (x, y) = (50, 0)$  となる。これを  $x-y$  座標について

まとめると,  $P9-P12 = (0, 30)$  となり  $y-z$  座標は  $P5-P6 = (30, 0)$  となる。よって  $y$  座標値は  $(30)$  である。これより始点 1 から得られた各点の三次元座標  $(x, y, z)$  は次のようになる。 $P5 = P9 (50, 30, 50)$ ,  $P6 = P12 (50, 0, 50)$

同様にして線分 1・2 の終点 2 に対して得られた各点の三次元座標は  $P5 = P10 (0, 30, 50)$ ,  $P6 = P11 (0, 0, 50)$  となる。

同様にして他の線分についても求める。

線分 2・3 の  $P2$  に対応する座標値  $P5 = P10 (0, 30, 50)$ ,  $P6 = P11 (0, 0, 50)$ ,  $P3$  に対応する座標値  $P8 = P10 (0, 30, 0)$ ,  $P7 = P11 (0, 0, 0)$

線分 3・4 の  $P4$  に対応する座標値  $P8 = P9 (50, 30, 0)$ ,  $P7 = P12 (50, 0, 0)$

線分 4・1 の  $P1$  に対応する座標値  $P5 = P9 (50, 30, 50)$ ,  $P6 = P12 (50, 0, 50)$  となる。

側面図・平面図も同様にして線分の始点, 終点について対応する三次元座標値を求める。

## 5. 立体図の線分の選索

二次元座標から三次元座標へ変換して得られたデータの 8 点を, 前述の正面図から得られたデータ順に番号 ( $P_31$  から  $P_38$ ) を付けると図 4 のようになる。この点  $P_31$  から  $P_38$  が三面図上の対応している点を正面図の点より探す。

$P_31 (50, 30, 50)$  は三面図の線分 1・2 の始点 1 に対する三面図上の点は,  $P5, P9$  である。他の点についても同様に求める。

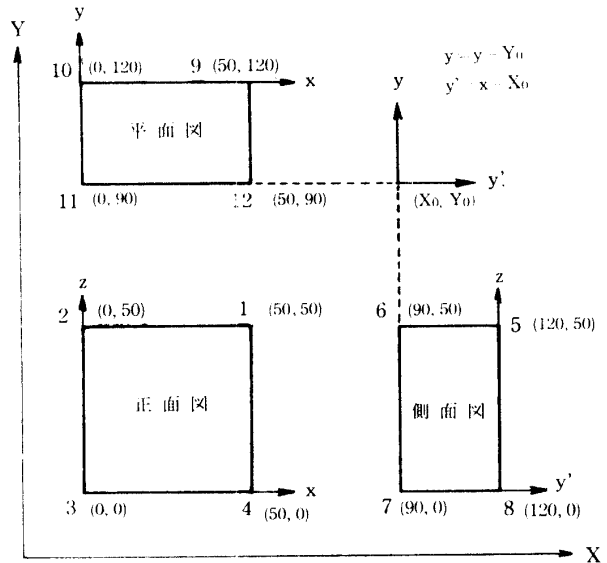


図 3 三面図の座標系

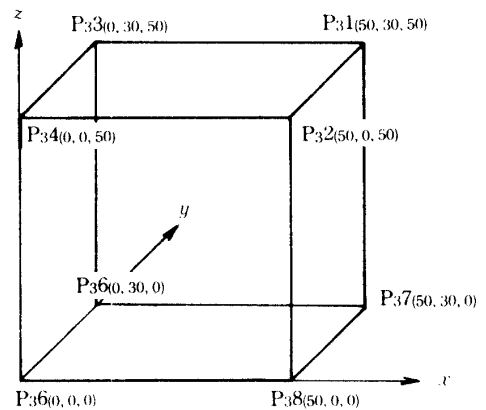


図 4 正面図方向から見た直方体の三次元座標値

次に $P_3,1$ から $P_3,8$ までの点から見出だされた三面図上の点を他の $P_3,1$ から $P_3,8$ までの点から同じであるものを捜し出す。たとえば $P_3,1$ から見出だされた $P_5$ と $P_9$ については、 $P_5$ は $P_3,3(0, 30, 50)$ 、 $P_9$ は $P_3,7(50, 30, 0)$ となる。

以下同様にして求めた結果をまとめると表1のようになる。

表1 正面図から見た各点の対応点

点番号	三面図上の対応点	点番号との終点对応点	点番号の始点と終点
$P_3,1$	$P_5, P_9$	$P_3,3, P_3,7$	$P_3,1-P_3,3, P_3,1-P_3,7$
$P_3,2$	$P_6, P_{12}$	$P_3,4, P_3,8$	$P_3,2-P_3,4, P_3,2-P_3,8$
$P_3,3$	$P_5, P_{10}$	$P_3,1, P_3,5$	$P_3,3-P_3,1, P_3,3-P_3,5$
$P_3,4$	$P_6, P_{11}$	$P_3,2, P_3,6$	$P_3,4-P_3,2, P_3,4-P_3,6$
$P_3,5$	$P_8, P_{11}$	$P_3,7, P_3,3$	$P_3,5-P_3,7, P_3,5-P_3,3$
$P_3,6$	$P_7, P_{11}$	$P_3,8, P_3,4$	$P_3,6-P_3,8, P_3,6-P_3,4$
$P_3,7$	$P_8, P_9$	$P_3,5, P_3,1$	$P_3,7-P_3,5, P_3,7-P_3,1$
$P_3,8$	$P_7, P_{12}$	$P_3,6, P_3,2$	$P_3,8-P_3,6, P_3,8-P_3,2$

点の座標値を線で結ぶと立体図が描ける事になる。

ただし、画面表示する場合は三次元座標を二次元上に変換しなければならないので、さらに各頂点の座標値および各頂点を結ぶ稜線の始点と終点の配列表が必要になる。これらの配列データを用い三次元の図形表示をするわけであるが、三次元の形状をディスプレイを通して人間が見ようとするとき三次元から二次元への投影図が必要になる。ここでの投影図法は透視投影法を採用した。視点の原点を立体の原点( $P_3,1$ )に置き、ディスプレイ中央点より左右方向(X)、前後方向(Y)、上下方向(Z)に視点を移動したときの立体の投影図を描くようにした。陰線処理は図形によって幾通りにも解釈できるので形状によっては複雑で非常に困難な場合がある。今回は単純な立方体の物体を構成する面、面の一部、稜線を消す陰線処理法を用いた。

## 6. 処理結果

図5に今回作成した直方体の図形処理のプログラムリストを示す。

配列型データ構造の情報を基に、正面・側面・平面図より各点の二次元座標を入力し、始点・終点およびその時の三次元座標を求める。その出力結果を図6に示す。その各頂点を線で結び立体図形表示を行なう。ただし、図形表示する前に図形の移動、回転表示等のデータはテンキーにより入力し、図形表示後さらに隠線処理を行なう。

ショウメン <sup>*</sup>		X	Y	Z	
シテン	9	50	30	50	1
シテン	12	50	0	50	2
シテン	10	0	30	50	3
シテン	11	0	0	50	4
シュウテン	8	50	30	0	7
シュウテン	7	50	0	0	8
シュウテン	8	0	30	0	5
シュウテン	7	0	0	0	6
シテシ	5	50	30	50	1
シテシ	6	50	0	50	2
シテシ	8	50	30	0	3
シテシ	7	50	0	0	4
シュウテン	10	0	30	50	7
シュウテン	11	0	0	50	8
シュウテン	10	0	30	0	5
シュウテン	11	0	0	0	6

図6 正面図から見た始点終点と三次元座標値

パーソナルコンピュータによる第三角法図面から立体投影図への変換処理法について

```

100 'save "CAD
110 WIDTH 80,28:CONSOLE 0,28,0,0:CLS 3
120 PI=3.14159:DEG=PI/180
130 -----
140 LINE(0,0)-(839,199):B,B,SHF99F
150 LINE(0,100)-(839,100):S,SH8888
160 LINE(320,0)-(320,199):S,SH8888
170 -----
180 DIM MAT(4,4):P(100,2),A(1,100),B(2,100)
190 DIM PX(1,100),PY(1,100),PZ(1,100)
200 DIM X(100),Y(100),Z(100)
210 -----
220 GOSUB=XYZDATA
230 GOSUB=MENU
240 -----
250 ON NO GOTO *NC1,*NC2,*NC3
260 *NC1:GOSUB *CODE1:GOTO 290
270 *NC2:GOSUB *CODE2:GOTO 290
280 *NC3:GOSUB *CODE3:GOTO 290
290 KP=0:KS=0:KP=0
300 FOR P=1 TO PA
310 FOR J=1 TO Z
320 ON J GOTO *Z1,*Z2
330 *Z1:C=PONT(1,P)
340 FOR L=1 TO PA:PPS(J,L)=PHASE(1,L):NEXT L:GOSUB *XYZ:GOTO 370
350 *Z2:C=PONT(2,P)
360 FOR L=1 TO PA:PPS(J,L)=PHASE(2,L):NEXT L:GOSUB *XYZ
370 NEXT J
380 ON NO GOTO *NP1,*NP2,*NP3
390 *NP1:PPS1=CKYZ(1,2):PPS2=CKYZ(2,1):GOSUB *CHECK:GOTO 420
400 *NP2:PPS1=CKYZ(1,1):PPS2=CKYZ(2,1):GOSUB *CHECK:GOTO 420
410 *NP3:PPS1=CKYZ(1,2):PPS2=CKYZ(2,2):GOSUB *CHECK:GOTO 420
420 NEXT P
430 GOSUB *AB1
440 GOSUB *AB2
450 GOSUB *AB3:GOTO 230
460 *XYZ:E=0
470 FOR I=1 TO PA
480 IF PPS(I,1)<C THEN 530
490 ON J GOTO *S1,*S2
500 *S1:E=E+1:XYZ1(J,K)=WANT1(1,1):XYZ2(J,K)=WANT2(1,1):MO(J,K)=NODE(1,1)
510 GOTO 530
520 *S2:E=E+1:XYZ1(J,K)=WANT1(2,1):XYZ2(J,K)=WANT2(2,1):MO(J,K)=NODE(2,1)
530 NEXT I
540 ON J GOTO *T1,*T2
550 *T1:FOR I=1 TO K:XYZ2(J,1)=XYZ2(J,1)-XYZ0(J):NEXT I
560 XYZ2(J,1)=XYZ1(J,1)-XYZ1(J,2):XYZ2(J,2)=XYZ2(J,1):GOTO 590
570 *T2:FOR I=1 TO K:XYZ2(J,1)=XYZ2(J,1)-XYZ0(J):NEXT I
580 XYZ2(J,1)=XYZ1(J,1)-XYZ1(J,2):XYZ2(J,2)=XYZ2(J,1)-XYZ2(J,2)
590 RETURN
600 *CHECK:IF PPS1>PPS2<0 THEN 650
610 IF PPS1=PPS2 THEN UAL(1)=PPS1
620 UAL(2)=0:GOTO 680
630 IF PPS1=PPS2 THEN UAL(2)=PPS1
640 UAL(1)=0:GOTO 680
650 IF PPS2<0 THEN UAL(2)=ABS(PPS2):SWAP MO(2,1),MO(2,2)
660 UAL(1)=ABS(PPS2):UAL(2)=0
670 IF PPS1<0 THEN UAL(2)=ABS(PPS1):SWAP MO(1,1),MO(1,2)
680 UAL(1)=ABS(PPS1):UAL(2)=0
690 ON NO GOTO *NO1,*NO2,*NO3
700 *NO1:GOSUB *POUT1:GOTO 730
710 *NO2:GOSUB *POUT2:GOTO 730
720 *NO3:GOSUB *POUT3:GOTO 730
730 RETURN
740 *CODE1
750 FOR I=1 TO PA
760 PONT(1,1)=XP(1):PONT(2,1)=ZP(1)
770 PHASE(1,1)=XP(1):PHASE(2,1)=ZS(1)
780 WANT1(1,1)=XP(1):WANT2(1,1)=YP(1)
790 WANT1(2,1)=YS(1):WANT2(2,1)=ZS(1)
800 NODE(1,1)=NOP(1):NODE(2,1)=NOS(1)
810 NEXT I:RETURN
820 *CODE2
830 FOR I=1 TO PA
840 PONT(1,1)=YS(1):PONT(2,1)=ZS(1)
850 PHASE(1,1)=YP(1):PHASE(2,1)=ZP(1)
860 WANT1(1,1)=XP(1):WANT2(1,1)=YP(1)
870 WANT1(2,1)=XP(1):WANT2(2,1)=ZP(1)
880 NODE(1,1)=NOP(1):NODE(2,1)=NOP(1)
890 NEXT I:RETURN
900 *CODE3
910 FOR I=1 TO PA
920 PONT(1,1)=XP(1):PONT(2,1)=YP(1)
930 PHASE(1,1)=XP(1):PHASE(2,1)=YS(1)
940 WANT1(1,1)=XP(1):WANT2(1,1)=ZP(1)
950 WANT1(2,1)=YS(1):WANT2(2,1)=ZS(1)
960 NODE(1,1)=NOP(1):NODE(2,1)=NOS(1)
970 NEXT I:RETURN
980 *POUT1:FOR I=1 TO K:KP=I
990 LPRINT MO(1,1),MO(2,1),XYZ1(1,1),UAL(1),XYZ2(2,1),KP
1000 A(1,KP)=MO(1,1)
1010 B(2,KP)=MO(2,1)
1020 PX(1,KP)=XYZ1(1,1):PY(1,KP)=UAL(1):PZ(1,KP)=XYZ2(2,1)
1030 NEXT I:RETURN
1040 *POUT2:FOR I=1 TO K:KS=I
1050 LPRINT MO(1,1),MO(2,1),UAL(1),XYZ2(1,1),XYZ2(2,1),KS
1060 A(1,KS)=MO(1,1)
1070 B(2,KS)=MO(2,1)
1080 PX(1,KS)=UAL(1):PY(1,KS)=XYZ2(1,1):PZ(1,KS)=XYZ2(2,1)
1090 NEXT I:RETURN
1100 *POUT3:FOR I=1 TO K:KP=I
1110 LPRINT MO(1,1),MO(2,1),XYZ1(1,1),XYZ1(2,1),UAL(1),KP
1120 A(1,KP)=MO(1,1)
1130 B(2,KP)=MO(2,1)
1140 PX(1,KP)=XYZ1(1,1):PY(1,KP)=XYZ1(2,1):PZ(1,KP)=UAL(1)
1150 NEXT I:RETURN
1160 -----
1170 *XYZDATA:CLS
1180 INPUT "正交法の点の数 PF=":PF
1190 INPUT "斜交法の点の数 PS=":PS
1200 INPUT "平面法の点の数 PP=":PP
1210 FOR I=1 TO PF:INPUT "XF,ZF,NOF=":XF(1),ZF(1),NOF(1):NEXT I: 'ショールズ
1220 FOR I=1 TO PS:INPUT "YS,ZS,NOS=":YS(1),ZS(1),NOS(1):NEXT I: 'ソルメス
1230 FOR I=1 TO PP:INPUT "XP,YP,NOP=":XP(1),YP(1),NOP(1):NEXT I: 'ヘイス
1240 FOR I=1 TO PF:PRINT XF(1),ZF(1),NOF(1):NEXT I: 'ショールズ
1250 FOR I=1 TO PS:PRINT YS(1),ZS(1),NOS(1):NEXT I: 'ソルメス
1260 FOR I=1 TO PP:PRINT XP(1),YP(1),NOP(1):NEXT I: 'ヘイス
1270 RETURN
1280 -----
1290 *MENU:CLS
1300 LOCATE 30,4:PRINT "①: 正 面 図 "
1310 LOCATE 30,8:PRINT "②: 側 面 図 "
1320 LOCATE 30,8:PRINT "③: 平 面 図 "
1330 LOCATE 30,10:PRINT "④: 次 の 立 体 "
1340 LOCATE 30,12:PRINT "⑤: 除 破 処 理 "
1350 LOCATE 30,13:PRINT " (立方体の場合)"
1360 LOCATE 30,18:PRINT "⑥: 終 わ り "
1370 LOCATE 30,17:INPUT NO=":NO
1380 ON NO GOTO *NL1,*NL2,*NL3,*NL4,*NL5,*NL6
1390 *NL1:LOCATE 20,19:INPUT "XYZ(1),XYZ(2)=":XYZ0(1),XYZ0(2)
1400 PA=PF:PRINT:PRINT " 正 面 図 ":PRINT:GOTO 1480
1410 *NL2:LOCATE 20,19:INPUT "XYZ(1),XYZ(2)=":XYZ0(1),XYZ0(2)
1420 PA=PS:PRINT:PRINT " 側 面 図 ":PRINT:GOTO 1480
1430 *NL3:LOCATE 20,19:INPUT "XYZ(1),XYZ(2)=":XYZ0(1),XYZ0(2)
1440 PA=PP:PRINT:PRINT " 平 面 図 ":PRINT:GOTO 1480
1450 *NL4:GOTO 220
1460 *NL5:GOTO 2930
1470 *NL6:CLS:END
1480 RETURN
1490 -----
1500 *AB2:JJ=M-(M/4)+1
1510 FOR I=1 TO M
1520 IF I=3 THEN GOTO 1640 ELSE 1640
1530 A1(1,JJ)=A(1,1):A2(1,1)=A(1,JJ)
1540 PX1(1,JJ)=PX(1,1):PY1(1,JJ)=PY(1,1):PZ1(1,JJ)=PZ(1,1)
1550 PX2(1,1)=PX(1,JJ):PY2(1,1)=PY(1,JJ):PZ2(1,1)=PZ(1,JJ)
1560 B1(2,JJ)=B(2,1):B2(2,1)=B(2,JJ)
1570 A(1,1)=A(1,1):A(1,JJ)=A(1,1)
1580 PX(1,1)=PX(1,1):PY(1,1)=PY(1,1):PZ(1,1)=PZ(1,1)
1590 PX(1,JJ)=PX(1,JJ):PY(1,JJ)=PY(1,JJ):PZ(1,JJ)=PZ(1,JJ)
1600 B(2,1)=B(2,1):B(2,JJ)=B(2,1)
1610 LPRINT A(1,1),B(2,1),PX(1,1),PY(1,1),PZ(1,1),I
1620 GOTO 1740
1630 IF I=4 THEN GOTO 1650 ELSE 1730
1640 A1(1,JJ)=A(1,1):A2(1,1)=A(1,JJ)+1
1650 PX1(1,JJ)=PX(1,1):PY1(1,JJ)=PY(1,1):PZ1(1,JJ)=PZ(1,1)
1660 PX2(1,1)=PX(1,JJ):PY2(1,1)=PY(1,JJ):PZ2(1,1)=PZ(1,JJ)+1
1670 B1(2,JJ)=B(2,1):B2(2,1)=B(2,1)
1680 A(1,1)=A(1,1):A(1,JJ)=A(1,1)
1690 PX(1,1)=PX(1,1):PY(1,1)=PY(1,1):PZ(1,1)=PZ(1,1)
1700 PX(1,JJ)=PX(1,JJ):PY(1,JJ)=PY(1,JJ):PZ(1,JJ)=PZ(1,JJ)
1710 B(2,1)=B(2,1):B(2,JJ)=B(2,1)
1720 LPRINT A(1,1),B(2,1),PX(1,1),PY(1,1),PZ(1,1),I
1730 NEXT I:RETURN
1740 -----
1750 *-----作 図-----
1760 *AB1:CLS:II=0:K=0
1770 LOCATE 28,11:INPUT "立体図の頂点の数 PO=":M
1780 G=M-PA
1790 *-----main-----
1800 *-----
1810 *-----
1820 GOSUB 2080: ' initialize matrix
1830 GOSUB 2500: ' set matrix
1840 *-----
1850 FOR I=1 TO M
1860 X(I)=PX(1,I):Y(I)=PY(1,I):Z(I)=PZ(1,I)
1870 GOSUB 2430: ' set point
1880 NEXT I
1890 *-----line-----
1900 FOR I=1 TO G
1910 PRINT " ",A(1,I),X(I),Y(I),Z(I),I
1920 NEXT I
1930 *-----line-----
1940 FOR K=1 TO G
1950 FOR I=1 TO G
1960 II=I+G
1970 IF A(1,II)=A(1,K) THEN 1980 ELSE 2020
1980 RE(2,I)=B(2,II)
1990 SX=320-P(X,1):SY=100-P(X,2)*.4495
2000 EX=320-P(I,1):EY=100-P(I,2)*.4495
2010 LINE (SX,SY)-(EX,EY),2
2020 NEXT I
2030 NEXT K
2040 RETURN
2050 *-----initialize matrix-----
2060 FOR I=1 TO 4
2070 FOR J=1 TO 4
2080 IF I=J THEN MAT(I,J)=1 ELSE MAT(I,J)=0
2090 NEXT J
2100 NEXT I
2110 MAT(3,3)=1
2120 RETURN
2130 *-----translate matrix-----
2140 MAT(4,1)=MAT(4,1)+TX
2150 MAT(4,2)=MAT(4,2)+TY
2160 MAT(4,3)=MAT(4,3)+TZ
2170 RETURN
2180 *-----rotation X-----
2190 S1=SIN(RX/DEG):CO=COS(RX/DEG)
2200 FOR I=1 TO 4
2210 TEMP=MAT(1,2)+CO*MAT(1,3)+S1
2220 MAT(1,3)=MAT(1,2)+S1*MAT(1,3)+CO
2230 MAT(1,2)=TEMP
2240 NEXT I
2250 RETURN
2260 *-----rotation Y-----
2270 S1=SIN(RY/DEG):CO=COS(RY/DEG)
2280 FOR I=1 TO 4
2290 TEMP=MAT(1,3)+CO*MAT(1,1)+S1
2300 MAT(1,1)=MAT(1,3)+S1*MAT(1,1)+CO
2310 MAT(1,3)=TEMP
2320 NEXT I
2330 RETURN
2340 *-----rotation Z-----
2350 S1=SIN(RZ/DEG):CO=COS(RZ/DEG)
2360 FOR I=1 TO 4
2370 TEMP=MAT(1,1)+CO*MAT(1,2)+S1
2380 MAT(1,2)=MAT(1,1)+S1*MAT(1,2)+CO
2390 MAT(1,1)=TEMP
2400 NEXT I
2410 RETURN
2420 *-----set point-----
2430 *-----
2440 X1=X(I)+MAT(1,1)+Y(I)+MAT(2,1)+Z(I)+MAT(3,1)+MAT(4,1)
2450 Y1=X(I)+MAT(1,2)+Y(I)+MAT(2,2)+Z(I)+MAT(3,2)+MAT(4,2)
2460 Z1=X(I)+MAT(1,3)+Y(I)+MAT(2,3)+Z(I)+MAT(3,3)+MAT(4,3)
2470 P(1,1)=INT(X1/Y1*100)
2480 P(1,2)=INT(Z1/Y1*100)
2490 RETURN
2500 *-----set matrix-----
2510 *-----loop menu-----
2520 LOCATE 28,11:INPUT "Enter Rotation Axis And Angle : ",AX,RA
2530 IF AX="X" OR AX="x" THEN RX=RA:GOSUB 2180:GOTO 2510
2540 IF AX="Y" OR AX="y" THEN RY=RA:GOSUB 2260:GOTO 2510
2550 IF AX="Z" OR AX="z" THEN RZ=RA:GOSUB 2340:GOTO 2510
2560 IF AX="Q" AND AX="q" THEN 2510
2570 *-----translate menu-----
2580 LOCATE 28,12:INPUT "Enter Translate Dx,Dy,Dz : ",TX,TY,TZ
2590 GOSUB 2130: ' translate matrix
2600 RETURN
2610 *-----
2620 *AB3
2630 CLS
2640 *-----
2650 FOR I=1 TO M
2660 X(I)=PX(1,I):Y(I)=PY(1,I):Z(I)=PZ(1,I)
2670 GOSUB 2660 SET POINT
2680 NEXT I
2690 *-----line-----
2700 FOR I=1 TO PA
2710 LPRINT " ",B(2,I),X(I),Y(I),Z(I),I
2720 NEXT I
2730 *-----line-----
2740 FOR K=1 TO PA
2750 FOR I=1 TO G
2760 II=I+PA

```

図5 プログラム リスト

```

2770 IF B(2,1)=B(2,K) THEN 2780 ELSE 2820
2780 A(1,1)=A(1,1)
2790 SX=320-P(K,1):SY=100-P(K,2)*.4495
2800 EX=320-P(1,1):EY=100-P(1,2)*.4495
2810 LINE (SX,SY)-(EX,EY),2
2820 NEXT K
2830 NEXT I
2840 RETURN
2850 '----- set point -----
2870 X1=X(1)*MAT(1,1)+Y(1)*MAT(2,1)+Z(1)*MAT(3,1)+MAT(4,1)
2880 Y1=X(1)*MAT(1,2)+Y(1)*MAT(2,2)+Z(1)*MAT(3,2)+MAT(4,2)
2890 Z1=X(1)*MAT(1,3)+Y(1)*MAT(2,3)+Z(1)*MAT(3,3)+MAT(4,3)
2900 P(1,1)=INT(X1/Y1*100)
2910 P(1,2)=INT(Z1/Y1*100)
2920 RETURN
2930 '--- 図形描画 ---
2940 FOR I=1 TO M
2950 PRINT "A,B,X,Y,Z=",A(1,1),B(2,1),X(1),Y(1),Z(1),1
2960 IF X(1)=0 AND Y(1)=0 AND Z(1)<>0 THEN GOSUB 3080 :GOTO 3040 :I=8
2970 IF X(1)<>0 AND Y(1)=0 AND Z(1)=0 THEN GOSUB 3100 :GOTO 3040 :I=4
2980 IF X(1)=0 AND Y(1)<>0 AND Z(1)=0 THEN GOSUB 3120 :GOTO 3040 :I=5
2990 IF X(1)<>0 AND Y(1)<>0 AND Z(1)=0 THEN GOSUB 3140 :GOTO 3040 :I=7
3000 IF X(1)<>0 AND Y(1)=0 AND Z(1)<>0 THEN GOSUB 3160 :GOTO 3040 :I=1
3010 IF X(1)=0 AND Y(1)<>0 AND Z(1)<>0 THEN GOSUB 3180 :GOTO 3040 :I=2
3020 IF X(1)<>0 AND Y(1)<>0 AND Z(1)=0 THEN GOSUB 3200 :GOTO 3040 :I=3
3030 IF X(1)=0 AND Y(1)=0 AND Z(1)<>0 THEN GOSUB 3220 :GOTO 3040 :I=6
3040 NEXT I
3050 GOSUB 3080
3060 GOTO 230
3070 RETURN
3080 EX8=320-P(1,1):EY8=100-P(1,2)*.4495
3090 RETURN
3100 EX4=320-P(1,1):EY4=100-P(1,2)*.4495
3110 RETURN
3120 EX6=320-P(1,1):EY6=100-P(1,2)*.4495
3130 RETURN
3140 EX7=320-P(1,1):EY7=100-P(1,2)*.4495
3150 RETURN
3160 EX1=320-P(1,1):EY1=100-P(1,2)*.4495
3170 RETURN
3180 EX2=320-P(1,1):EY2=100-P(1,2)*.4495
3190 RETURN
3200 EX3=320-P(1,1):EY3=100-P(1,2)*.4495
3210 RETURN
3220 EX5=320-P(1,1):EY5=100-P(1,2)*.4495
3230 RETURN
3240 '-----
3250 '-----
3260 *BC:CLS
3270 LINE(EX8-1,EY8-1)-(EX4+1,EY4-1),0,BF
3280 IF EY8<EY7 THEN 3290 ELSE 3310
3290 LINE(EX8-1,EY7)-(EX5-1,EY5),0,BF
3300 GOTO 3320
3310 LINE(EX8-1,EY8-1)-(EX5-1,EY5),0,BF
3320 IF EX8>EX1 AND EY8>EY1 THEN 3330 ELSE 3350
3330 LINE(EX8-1,EY8-1)-(EX1-1,EY1),0,BF
3340 GOTO 3370
3350 IF EX8=EX1 AND EY8<EY1 THEN 3360
3360 LINE(EX8-1,EY8-1)-(EX1,EY1-1),0
3370 RETURN

```

これらの結果を基に、直方体の立体的図形表示処理結果を図示したものが図7、図8である。  
処理結果は、CRTに表示された図形をプリンタ出力させたものである。

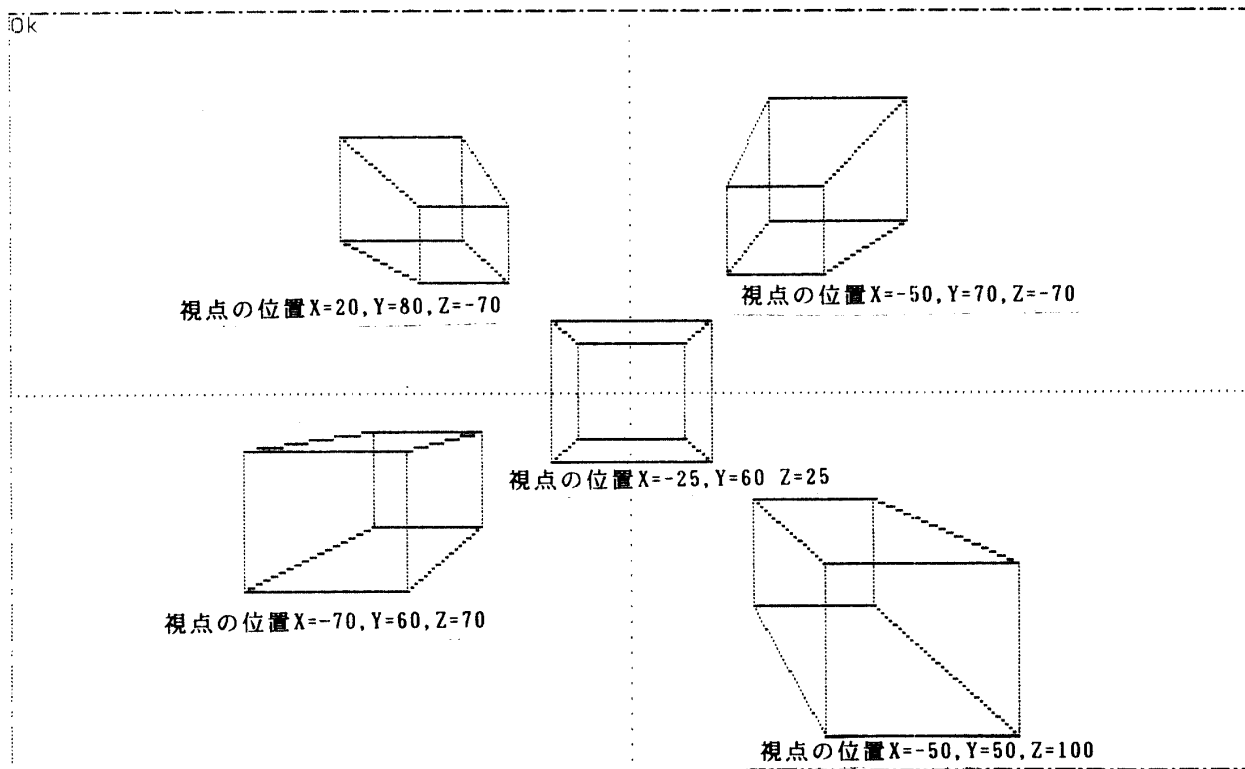


図7 三面図からの直方体の処理結果

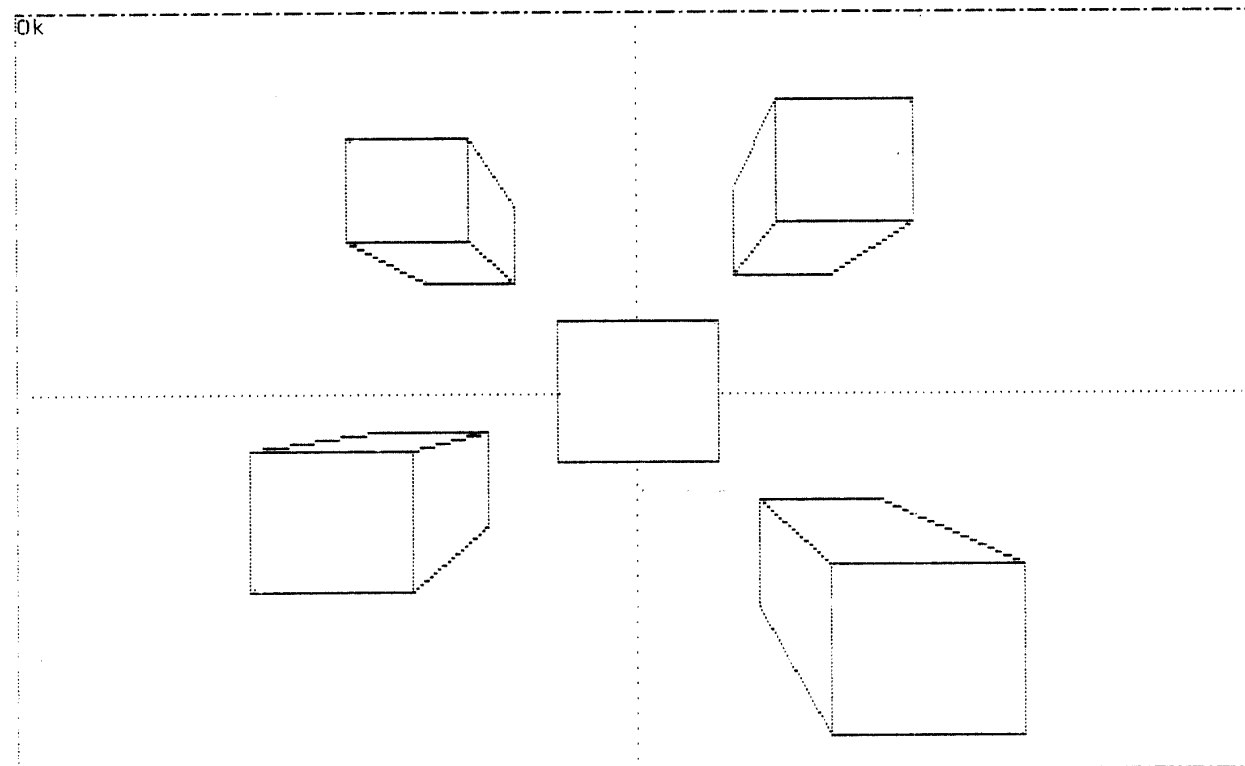


図8 直方体の陰線処理結果

## 7. ま と め

工科系高校、大学では近年、設計製図関連の教育にパソコンを用いたCAD/CAMなどを取り入れ、市販のソフト等によって実施しているところが増えてきている。筆者はパソコンをベースにして機械製図で利用する三面図の座標データを読み取り、テンキー入力によってその物体を三次元化処理するための考究とその応用を試みた。

その結果として以下の結論を得ることが出来た。

1. 直方体を例にあげてきたが、この処理法を用いて必要最小限の形状分割施すと、平面の比較的多い多面体についても処理出来ることがわかった。
2. この三次元化処理法は機械系のCAD学習の初歩として充分役立つことがわかった。

最後に本研究に協力された62年度卒研究生揚原英義君外3名に深謝する。

## 参 考 文 献

- 1) 望月・光成, 設計者のためのCAD/CAM, 産業図書, (1985-3)
- 2) 朝野・今井, 計算とアルゴリズム, オーム社, (1986-7)
- 3) 加藤・川元, 日本製図学会誌, (1987-5) p 29
- 4) 福永・柴田, パソコンによる作図の基礎, 培風館, (1987-1)