〔非公開〕

ＴＲ－Ｃ－００７１

３次元画像モデルデータベースのアクセス・指示法の研究

垣谷　恵三
Keizo KAKITANI

１９９１．１１．１４

ＡＴＲ通信システム研究所

３次元画像モデルデータベースのアクセス・指示法の研究

垣谷　恵三

ＡＴＲ通信システム研究所

知能処理研究室

目次

１．まえがき

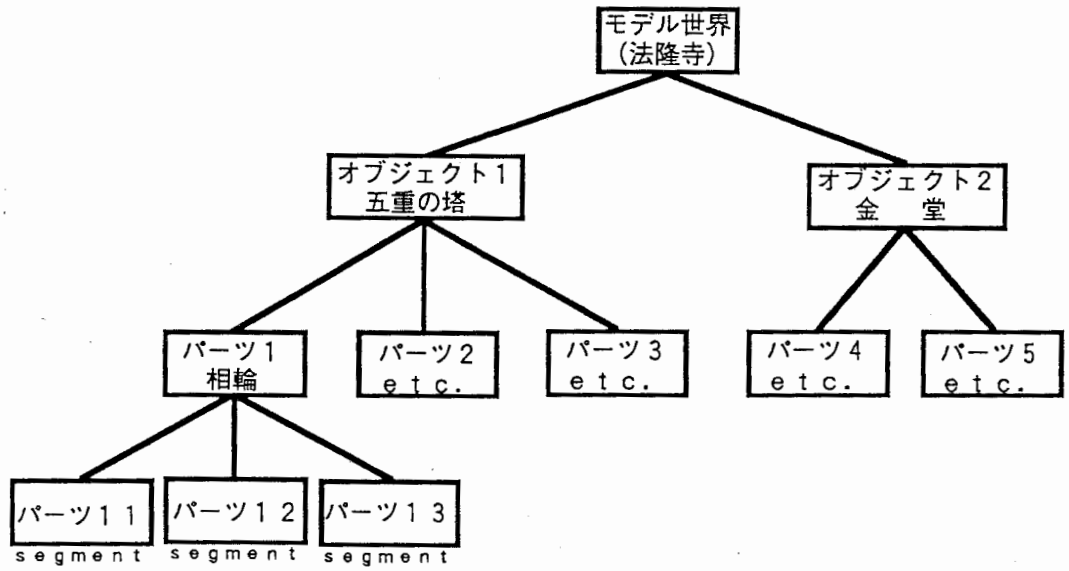
臨場感通信会議 [1] 等の高度な通信システムにおいては，協調作業や遠隔地点とを結んだ臨場感の発生が将来の重要なテーマである。ここでは協調作業の実現に必須の３次元画像モデルデータベースへのアクセス・指示法が重要な問題となると考えられる。特に画像中の個々のオブジェクトに対するアクセス・指示法や，複数の作業者間との通信・アクセス手法を検討していく必要がある。画像中へのアクセス・指示法の研究としては，手形状・手の位置が検出でき直観的な操作が可能なデータグローブ^{TM} による仮想空間操作 [1][2] や，データグローブ^{TM} と言語指示を併用して地図上の建物等の名称を検索させる地図案内システムＩＭＡＧＥ [3] の提案がある。さらに言語処理による画像へのアクセス・指示法の研究としては，自然言語テキスト入力から対象世界の再構成を行う実験システムＳＰＲＩＮＴ [4]、ニューラルネットを利用した画像・図形生成と言語のインターフェース [5]，確信度関数を用いて２次元空間中で指示言語を定量化した画像検索システムＳＰＡＤＥ [6]，３次元モデル世界における言語によるシーン探索・記述システム３Ｄ－ＳＰＡＲＳや３次元モデル世界生成システムを構成する提案 [7][8]，等がある。

本レポートにおいては，画像中のオブジェクトへのより抽象的なアクセス・指示法と複数端末を用いた協調作業の実現をねらった３次元モデル中のオブジェクトの移動手法の検討を行った。具体的にはＳＵＮワークステーションにインストールされた市販の３次元ＣＧソフトウエア（ＨＯＯＰＳ）を利用したアプリケーションプログラムを作成し、問題点の抽出を図った。

なお、本研究は１９９１年１０月７日より１１月１日までの約１ヶ月間に、同志社大学工学部からの学外実習として ATR 通信システム研究所で行なったものである。

２．画像中のオブジェクトへのアクセス・指示法の検討

ＣＧソフトウエアＨＯＯＰＳで表示されたモデル世界のオブジェクトのパーツを指示し、指示されたパーツのみを移動・回転等の操作を行うことを目的とした。特に抽象的な言語指示の可能性についての検討をおこなった。図１にオブジェクト構成の例を示す。このような階層構造で任意のパーツを指示すると、指示されたパーツの下位のパーツをすべて含んだセグメント群が選択される。例えば法隆寺の五重の塔の「相輪」と指示することで、下位のパーツ群１１～１３が包括的に指示される。



ｆｉｇ．１　オブジェクトの階層構成例

３．　ＨＯＯＰＳによるプログラムの作成

今回作成したプログラムのソースファイルはabstruct.cとteleacc.cの２つのプログラムで，いずれもＨＯＯＰＳのＣ言語仕様で書かれている。各々のプログラムについて以下に説明する。

３．１　プログラム *⟨abstruct.c⟩*

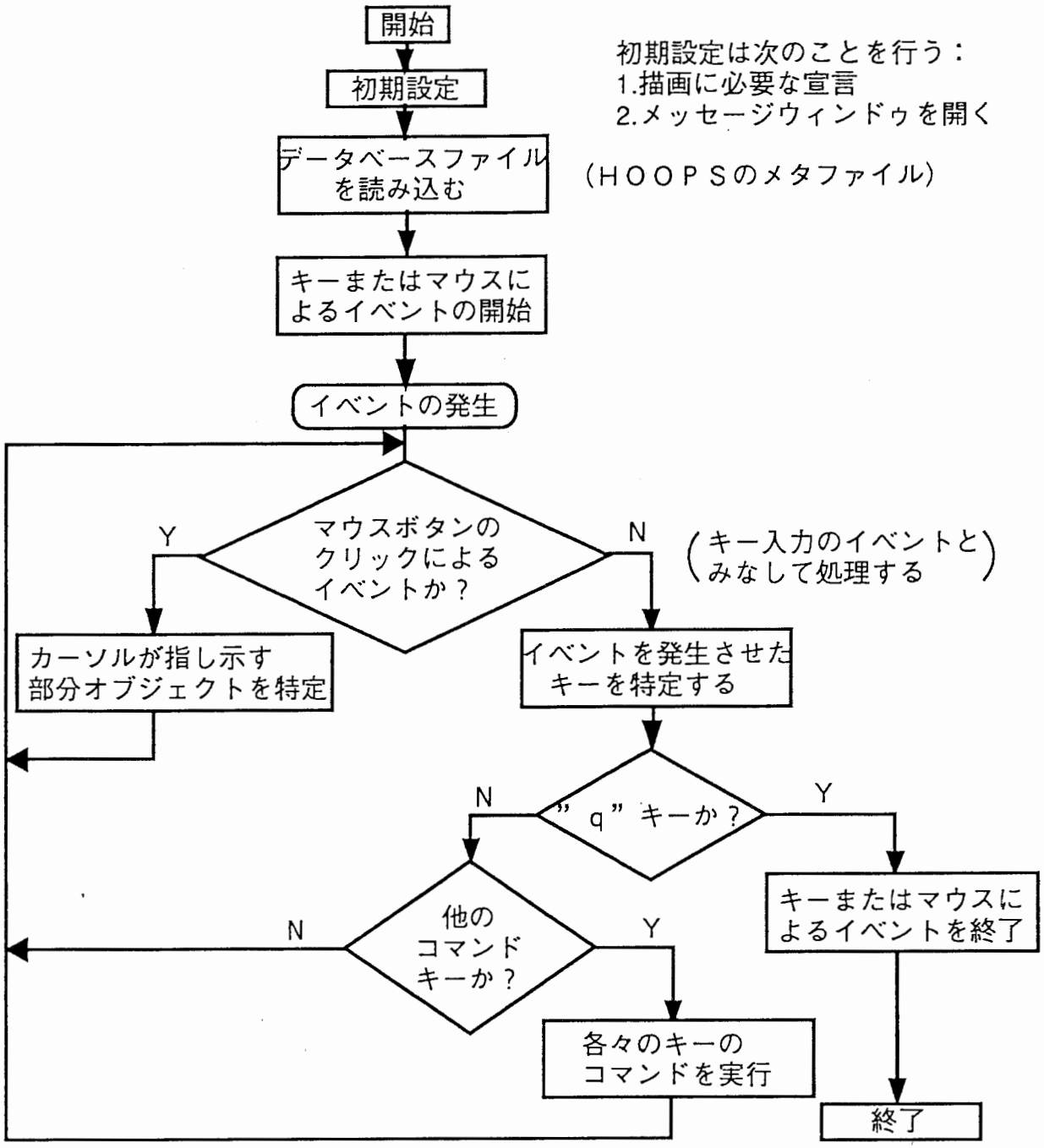
３．１．１　概要

このプログラムは表示されたモデル世界中のオブジェクトの一部または全体をキーボード、またはマウスからの入力命令によって、移動・回転などをさせることを目的としている。（フローチャートは図２、実験例は図３、ソースファイルはＡｐｐｅｎｄｉｘを参照）

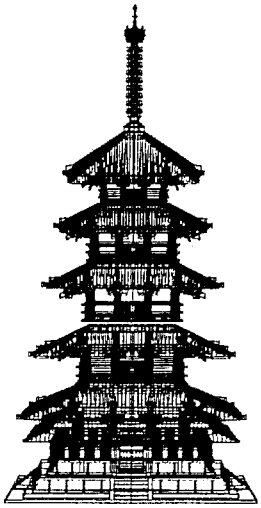
３．１．２　ファイル構成

ソースファイルは４つのプログラムに分けられる。各々のプログラムはおもに次のような役割・機能を持つ。

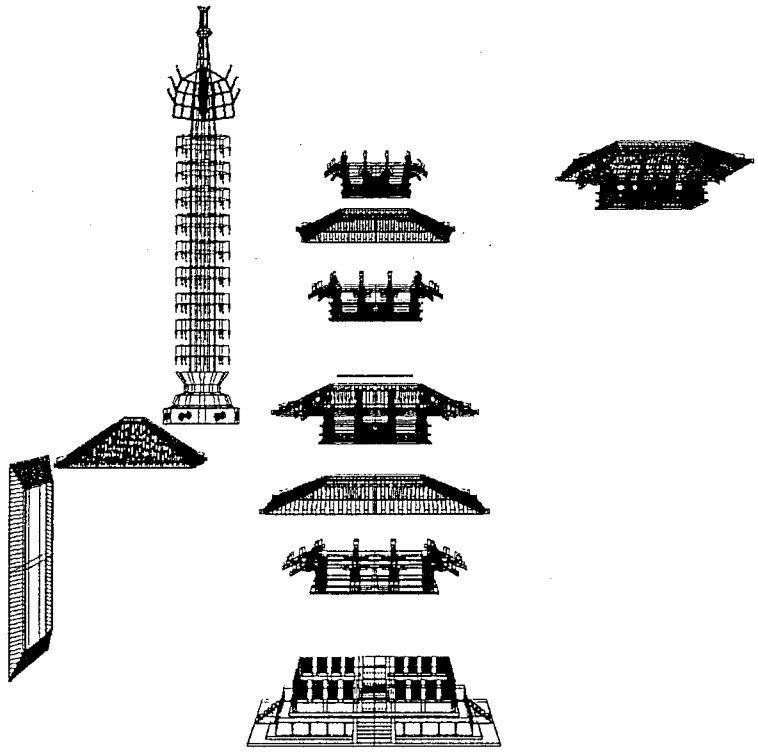
（１） main() ：主プログラムとして、初期設定・コマンド入力制御・終了設定を行う。
（２） *Get_mouse*() ：マウスによって指示されたセグメントの情報を取り込む。
（３） *Doit_subr*() ：キー入力によるコマンドを制御する。
（４） *Cancel_red*() ：赤くハイライトしたポリゴンのエッジを初期化する。



ｆｉｇ.2　ａｂｓｔｒｕｃｔ.ｃのフローチャート



（ａ）五重の塔の表示



（ｂ）各セグメントのパーツを移動させた例

ｆｉｇ．３　ａｂｓｔｒｕｃｔ．ｃによる実験例

３．１．３　コマンド

キー入力できるコマンド群は表１の通りである。

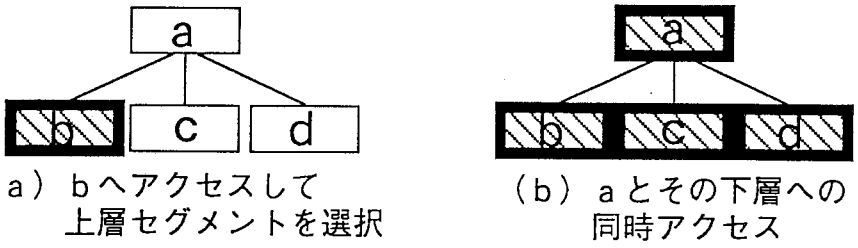
表１　　入力可能なコマンド群 ⟨*abstruct.c*⟩

命令	コマンドの効用
u	対象とするオブジェクトを画面上方（ｚ方向正）に移動
d	画面下方（ｚ方向負）
l	画面左方（ｙ方向正）
r	画面右方（ｙ方向負）
i	画面前方（ｘ方向正）
o	画面後方（ｘ方向負）
[*CTRL*]+	大きく移動量を調節
[*SHIFT*]+	小さく移動量を調節
↑ (up arrow)	画面の横方向を軸として上方回転
↓ (down arrow)	下方回転
→ (right arrow)	画面の縦方向を軸として右方回転
← (left arrow)	左方回転
b	１８０°回転
. (period)	画面の法線方向を軸として時計回り
, (comma)	反時計回り
m	対象とするオブジェクトを拡大する
[*SHIFT*]+	回転角を小刻みにできる
n	縮小する
c	対象とするオブジェクトを全体<ーー>部分で切り替える
p	対象とするオブジェクトの親セグメントに対象を移す
z	対象とする部分のオブジェクトを解除する
q	プログラムを終了させる
マウス	カーソルで指し示したいオブジェクトの位置に置いて任意のボタンをクリックすると、その部分のエッジが赤くハイライトされてそれが部分オブジェクトとなったことを示す

３．１．４　主な機能・特徴

(1) 移動や回転の対象となるオブジェクトを部分的に指示してアクセスすることができる。
(2) 単一の部分オブジェクトのあるセグメントから親セグメントに対象を移すことができる。（図４参照）
(3) 移動・回転のキーコマンドにCTRLキー、SHIFTキーを併用することにより移動量を３段階、回転角を２段階に調節することができる。
(4) 部分オブジェクトを回転させる際に、そのオブジェクトの中心を通る回転軸を基準にして回転

する。
(5) 画面下方の2枚のメッセージウィンドゥにより現在動かせるオブジェクト、指示した部分オブジェクトを知ることができる。



（a） bへアクセスして
上層セグメントを選択

（b） aとその下層への
同時アクセス

ｆｉｇ．４　親セグメントの呼び出し例

３．２　プログラム ⟨*teleacc.c*⟩

３．２．１　概要

このプログラムは2つの端末より同一マシンの同一ディレクトリに入りこのプログラムを走らせると、データベース上のオブジェクトを各々からアクセスさせて移動等をおこなうことを目的としている。（図５参照、なおフローチャートは図６、ソースファイルはＡｐｐｅｎｄｉｘを参照）

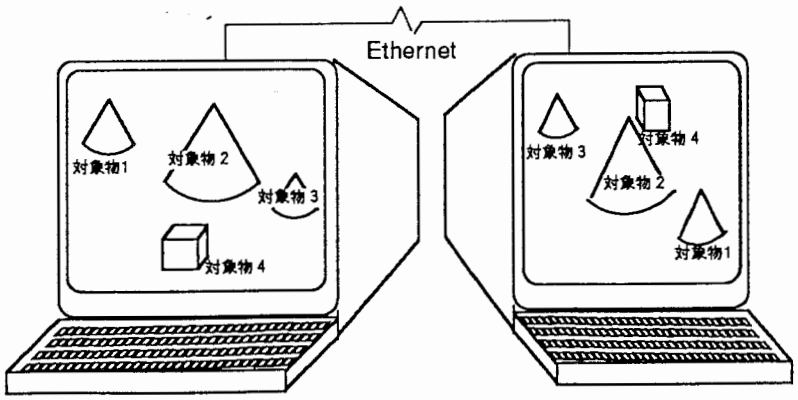
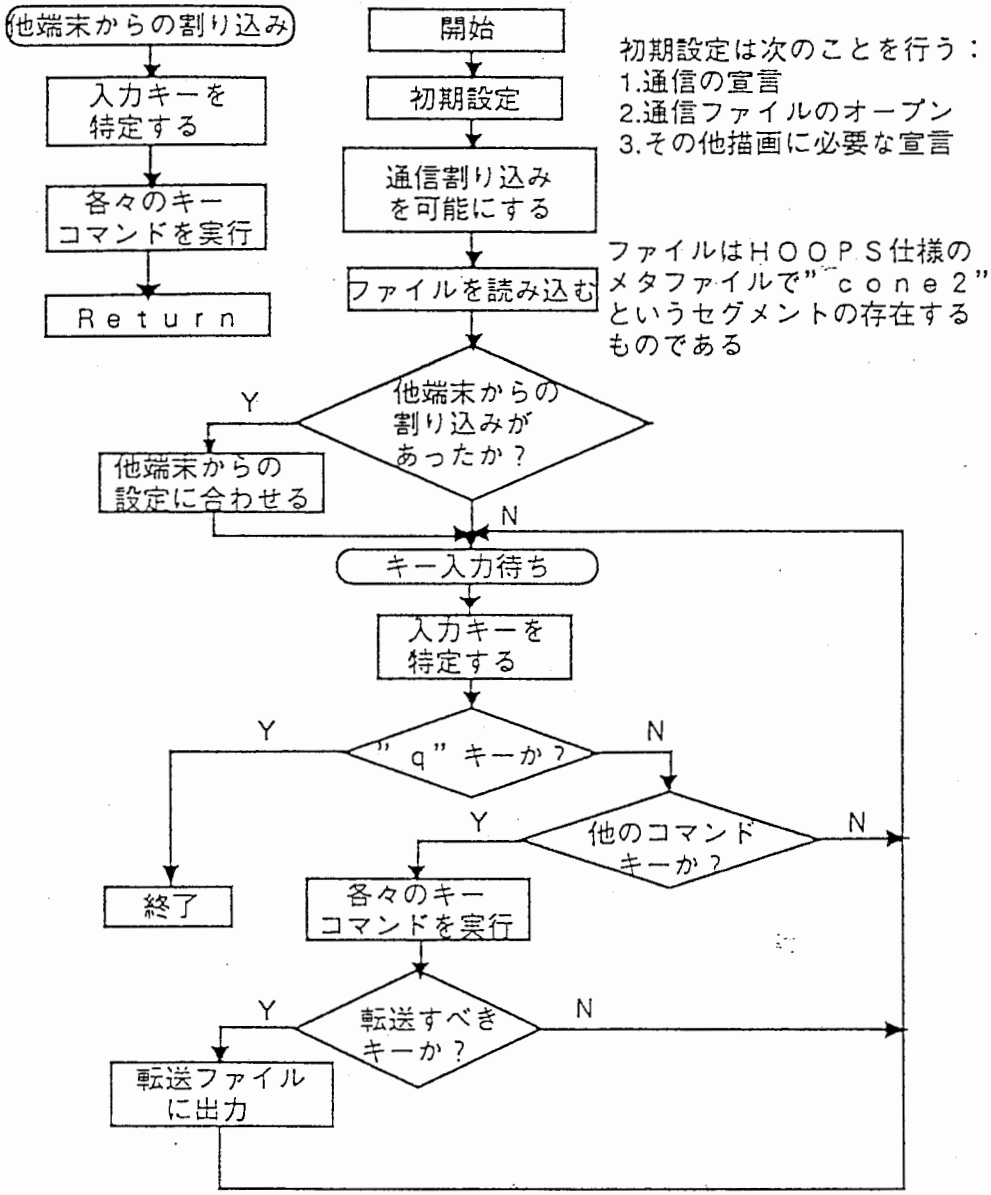


図5　二端末間通信の概念図



ｆｉｇ．６　ｔｅｌｅａｃｃ．ｃのフローチャート

３．２．２ ファイル構成

ソースファイルは次の３つのプログラムから構成される。以下に各プログラムの役割・機能を示す。

（１） main()：主プログラムとして初期設定・割り込みの設定・キー入力の設定等をおこなう。
（２） irrupt()　：２端末間通信に必要な割り込み処理を行う。
（３） *Doit_subr*()：キー入力の処理を行う。

３．２．３　コマンド

キー入力のうち、特にデータベースの変更に影響を与えるコマンドに絞って以下に挙げる。いずれも一方の端末において投入されたコマンドが記録されたファイルから読み出され実行されていく。（表. ２を参照）

表. ２　　転送可能なコマンド群 ⟨*teleacc.c*⟩

命令	コマンドの効用
Z	画面の縦方向（Z 軸）に関して、円錐が半球の回りを反時計回りに回る
W	時計回りに回る
F	円錐が半球の中心から遠ざかる方向に移動する
G	中心に近づく
U	円錐が画面上方に移動（未完成）
D	画面下方に移動（未完成）
A	画面後方に移動（未完成）
B	画面前方に移動（未完成）
L	画面左方に移動（未完成）
R	画面右方に移動（未完成）

３．２．４　起動方法

２つの端末より同一マシンの同一ディレクトリにはいり、そこでこのプログラムを別個に起動させる。後に起動させた方は、先に起動させてデータベース変更を行った属性を受け入れるので、起動は時間的なずれがあってもよい。視点はそれぞれの端末で設定出来る。以下に起動方法の一例を示す。file1 と file2 はそれぞれのプログラムのコマンド書き込み／読みだし用ファイルであり、file3 は Work File である。＃１の端末で投入されたコマンドは＃１のプログラムにより file1 に書き込まれ、書き込み event が＃２のプログラムに渡されると、＃２のプログラムで file1 が読み込まれた後、file1 の内容は消去される。＃２の端末で投入されたコマンドも同様に file2 に書き込まれ、＃１のプログラムで読み込まれる。この様にして、両プログラムのモデル世界中の対象物を同じ様に操作する。

(1) 端末＃１での起動コマンド

teleacc file1 file2 file3

(2) 端末# 2での起動コマンド

teleacc file2 file1 file3

３．２．５　主な機能・特徴

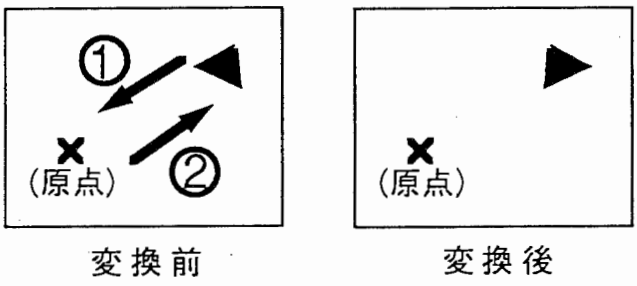
一方の端末から指示されたのオブジェクトの移動・回転の結果できたオブジェクト世界がそのまま他端末に転送されて表示される。２端末から同時にキー入力のアクセスをすることができる。後にプログラムを起動した側は、さきに起動した側のオブジェクトの属性データをそのまま引き継ぐことができる。（オブジェクトが配置されている座標など）

３．３　ＨＯＯＰＳの使用に際しての留意点

ＨＯＯＰＳはＣ言語で組み込めるＣＧツールであり、多面体等で構成される対象物を処理するためのコマンドを多数用意している。そこでマニュアルやチュートリアル等の添付ソフトで書かれていないより高度な使用をするために、いくつか留意すべき点について以下に述べる。

(1) オブジェクトの回転

ＨＯＯＰＳではオブジェクトを回転させる際は、原点を通る軸に関して回転するコマンドしかないのでxy, yz, xz 平面からずれた位置にあるオブジェクトは原点に対称に人工衛星軌道を描いて回転してしまう。この対策としてあらかじめ *Rotate_Object* コマンドで原点にオブジェクトを置いてから回転させ、その後元の位置に戻すようにした。（図. ７参照）



ｆｉｇ．７　オブジェクトを一旦原点に
移動させてから回転させる

(2) オブジェクトの位置

Show_Modelling_Matrix コマンドでその属性を検索できるが、初期配置時は位置が定義されていない場合があるので、あらかじめ *Translate_Object* で往復移動させて、位置を定義させた。ソースファイル中の *Doit_Subr*() に一見無意味な移動コマンドがあるのはそのためである。なお, この場合は *Show_Existence* コマンドで *modelling_matrix* の存在を検出することは、位置の定義までは確かめることができないので、ここでは意味を持たないこととなる。

(3) 視点の移動

Orbit_Camera,Dolly_Camera といったカメラ（視点）を移動するコマンドがある。ひとたびこのコマンドで部分オブジェクトを移動させた後、親セグメントに制御を移したときにその部分オブジェクトは別個の存在として解釈され、親セグメントと共に移動・回転をしなくなるのである。この解決法として　今回は *Translate_Object* 等のオブジェクトを直接移動させるコマンドのみを使用した。

４．　考察および検討

４．１　データベースモデルの部分的アクセスとその指示法

今回はＨＯＯＰＳによってモデルの部分的アクセスを行ってきたのだが、それに際して克服すべき問題として、面の重なりの判定がある。例えば２つの物体を接合する際は２つのオブジェクトが同じ空間を共有して重なってはならない。しかしＨＯＯＰＳでは、２つのオブジェクトの重なりを判定・禁止するといった機能を備えていない。そのため自らがオブジェクトの頂点を抽出して、面や線を構成してその判定するといったアルゴリズムを開発しなければならない。そしてそれは構成されるオブジェクトの種類・ポリゴンの数が大きくなればなるほど膨大な回数の重なりの判定を行わなくてはならないので、処理時間のなどの面で不利である。そこで各々のオブジェクトを含んだセグメント情報に、あらかじめ頂点の座標とオブジェクトの大きさの属性を与える。そしてあるオブジェクトを移動・回転させた際に、他のオブジェクトは大きさの属性内にいま動かしたオブジェクトの頂点が入り込んでいないかを判定するわけである。こうすればただ漠然と全てのポリゴンにおいて重なり判定を行うよりは、その判定の回数を減らすことができる。このようにセグメントに重なり判定のための属性を与えることは非常に有効であるので、この方法を提案し、また今後の検討課題としたい。

４．２　複数の端末を用いた協調作業

今回はオブジェクトとして半球と円錐からなる簡単なデータベースを用いたのであるが、abstruct.c の様な、あらゆるメタ・ファイル仕様のデータベースアクセスや部分オブジェクトをアクセスすることを可能にしたプログラムも組み込むことができる。また２端末に限らず、３端末以上での多重アクセスへと拡張することもできるが具体的なプログラムは作成していない。しかし問題点として以下のことが考えられる。

(1) ２端末間でのアクセス状態から一方が終了して、再びプログラムを立ち上げると３次元の対象物モデル世界がそれぞれの端末で異なる場合がある。これは、モデル世界の初期値からコマンドによる差分でオブジェクトの移動を行なっているためである。

(2) 複数の同時に発生したコマンドに対する処理の方法も検討する必要があるように思える。２端末で同時にコマンドを投入すると、さきに受け付けた側のコマンドが優先され、他方のコマンドはそのコマンド処理が終わるまで待機状態となる。その場合、たとえば待機状態にある側の端末で何らかの不都合が生じたとき、その不都合を解消しようとキー割り込みをかけても，そのキーは待機状態であるためすぐさま実行できない等の不都合を生じることになる。その対策として通信割り込み中から直ちに復帰できる優先度の高いコマンドを用意することが考えられる。

５．まとめ

今回は画像中のオブジェクトへのアクセス・指示法と、協調作業を可能にするための複数端末間の共有データベースアクセスの検討を行った。その具体例として3次元CGソフトウエアHOOPSを利用したプログラムを作成した。このプログラムにより、画像モデルのオブジェクトパーツへのアクセス・指示が可能になり、また共有データベースを2端末間においてアクセスすることが今後可能になると考えられる。

謝辞

今回の学外実習のお世話をしてくださった寺島社長、岸野室長、終始指導に御尽力くださった望月主任研究員、その他ATR通信システム研究所の皆様に深く感謝いたします。

参考文献

1) 岸野:“臨場感通信会議,” PIX・EL '91, 1991
2) 岸野:“臨場感通信会議,” 計測と制御,Vol.30, No.6, pp485,1991
3) 高橋, 岸野:“手振りの認識方法とその応用”, 電子情報通信学会論文誌, D-II, Vol.J73-D-II, No.12, pp1985, 1990
4) 山田, 西田, 堂下:“情景描写文からの対象世界の復元”, 情報処理学会研究報告, Vol.89, No.2(AI-62), pp.62.3.1-62.3.10, 1989
5) 曽我, 豊田:“画像・図形と自然言語のインターフェースについての一考察,”人工知能全国大会論文集, Vol.3, No.Pt.2, pp519, 1989
6) 高橋, 島, 岸野:“位置情報を手がかりとする画像検索法”, 情報処理学会論文誌, Vol.31, No11, pp1636, 1990
7) 望月, 岸野:“言語による3次元画像検索・記述法の提案”, 第6回ヒューマンインターフェースシンポジウム, 2222, pp375, 1990
8) 望月、岸野:“言語と画像メディアを統合した3次元対象物モデル世界生成システムの提案,” 電子情報通信学会, HC91-20, pp1, 1990

Appendix. A
abstruct. cのソースファイル

```
 ****************************************************************/

#include <stdio.h>
#include <math.h>

#define STREQ(a,b)    (strcmp(a,b) == 0)

char    cur_seg[128] = "¥0",
        str_seg[128] = "¥0",
        str_buf[128] = "¥0",
        segname[32]  = "¥0";

static int eins       = 0,
           segment_up = 0;

main(argc, argv)
   int     argc;
   char  **argv;
{
   char   button[30], cmd[128], etype[32];

   HC_Define_Alias("?root","/driver/x11/cs01:0.0");

   HC_Open_Segment("?root/flame1");
      HC_Set_Window (-1.0, 1.0, -1.0, -0.9);
   HC_Close_Segment ();

   HC_Open_Segment("?root/flame2");
      HC_Set_Window (-1.0, 1.0, -0.9, -0.8);
   HC_Insert_Text (0.0, 0.0, 0.0,"Main Object Mode");
   HC_Close_Segment ();

   HC_Open_Segment("?root/object");
      HC_Set_Driver_Options
         ("double-buffering,subscreen=(-1, 1,-1, 1)");
      HC_Set_Camera_Position(10.0 , 0.0 , 0.0);
      HC_Set_Camera_Up_Vector(0.0 , 0.0 , 1.0);
      HC_Set_Heuristics ("no hidden surfaces");
      HC_Set_Color ("edges = black,lines = blue");
      HC_Set_Visibility ("faces = off, lights = off");
      HC_Set_Window (-1.0, 1.0, -0.8, 1.0);
        HC_Set_Line_Weight( 3.0 );
        HC_Insert_Line(0.1, 0.0, 0.0,  -0.1, 0.0, 0.0);
        HC_Insert_Line(0.0, 0.2, 0.0,   0.0,-0.2, 0.0);
      if (argc < 2)
         sprintf(cmd, "read, name = dece.hmf");
      else
         sprintf(cmd, "read, name = '%s'",argv[1]);
      HC_Set_Metafile (cmd);

      HC_UnSet_Metafile ();
      HC_Set_Selectability ("everything = on");
```

```
   HC_Enable_Button_Events ("?Keyboard", "anything");
   HC_Enable_Selection_Events ("?Locater","?root");
/***** Event Occur *****/


   while (1)
   {
      HC_Await_Event ( etype );
        if (STREQ (etype, "selection"))  Get_mouse();
        else
        {
           HC_Show_Button ( button );
           if (!Doit_subr ( button )) break;
        }
      HC_Flush_All_Events();
    }

   HC_Disable_Button_Events("?Keyboard", "anything");
   HC_Disable_Selection_Events ("?Locater","?root");
/***** Event End *****/

   HC_Close_Segment();

   HC_Reset_System();
}

/*********************************************************************/

Get_mouse()
{
      if (!eins )  eins = !eins;
      else if(strlen(str_seg))  Cancel_red();
         HC_Show_Selection_Pathname( cur_seg );
         strcpy (str_seg, cur_seg);
         HC_Flush_Segment ("?root/flame1");
         HC_Show_Selection( segname );
         HC_QInsert_Text ("?root/flame1"
                                  , 0.0, 0.0, 0.0, segname);
         HC_QSet_Color ( str_seg , "edges = red");
     if ( segment_up ) segment_up = !segment_up;
}

/*********************************************************************/

Doit_subr( button )
   char  *button;
{
   int                 i, j, n;
   float             mp,mtx[4][4];
   static int segment_at  = 1;

   i = strlen(button);  mp = 1.0; /* mutiple command */
   if ( i == 9 )
      {
         mp = 5.0;  sprintf(button,"%c",button[8]);
      }
   else if ( button[0] >= 'A'  &&  button[0] <= 'Z' )
      {
         mp = 0.2;  sprintf(button,"%c",(button[0]-'A'+'a'));
       }
   else if (i == 3  ||  STREQ(button, "<")  ||  STREQ(button, ">"))
         mp = 0.1111111111;
```

```
if ( segment_at )  strcpy(str_buf, "?root/object");
else  strcpy (str_buf, str_seg);
   HC_QTranslate_Object(str_buf, 1.0, 1.0, 1.0);
   HC_QTranslate_Object(str_buf, -1.0, -1.0, -1.0);
HC_QShow_Modelling_Matrix(str_buf, mtx);
HC_QTranslate_Object(str_buf,
                -mtx[3][0], -mtx[3][1], -mtx[3][2]);

        if (STREQ(button, "u"))
      HC_QTranslate_Object(str_buf, 0.0  ,   0.0, 1.0*mp);
   else if (STREQ(button, "d"))
      HC_QTranslate_Object(str_buf, 0.0  ,   0.0, -1.0*mp);
   else if (STREQ(button, "r"))
      HC_QTranslate_Object(str_buf, 0.0  , -1.0*mp,   0.0);
   else if (STREQ(button, "l"))
      HC_QTranslate_Object(str_buf, 0.0  ,  1.0*mp,   0.0);
   else if (STREQ(button, "i"))
      HC_QTranslate_Object(str_buf, 1.0*mp,   0.0,   0.0);
   else if (STREQ(button, "o"))
      HC_QTranslate_Object(str_buf, -1.0*mp,   0.0,   0.0);

   else if (STREQ(button, "up arrow")  ||  STREQ(button, "kp8"))
      HC_QRotate_Object (str_buf,  0.0, 90.0*mp,  0.0);
   else if (STREQ(button, "down arrow") || STREQ(button, "kp2"))
      HC_QRotate_Object (str_buf,  0.0, -90.0*mp,  0.0);
   else if (STREQ(button, "right arrow") || STREQ(button, "kp6"))
      HC_QRotate_Object (str_buf,  0.0,  0.0, -90.0*mp);
   else if (STREQ(button, "left arrow") || STREQ(button, "kp4"))
      HC_QRotate_Object (str_buf,  0.0,  0.0, 90.0*mp);
   else if (STREQ (button, "b"))
      HC_QRotate_Object (str_buf,  0.0,  0.0, 180.0);
   else if (STREQ (button, ">")  ||  STREQ(button, "."))
      HC_QRotate_Object (str_buf, 90.0*mp,  0.0,  0.0);
   else if (STREQ (button, "<")  ||  STREQ(button,"comma"))
      HC_QRotate_Object (str_buf, -90.0*mp,  0.0,  0.0);

   else if (STREQ (button, "m"))
      HC_QScale_Object (str_buf, 1.5*mp, 1.5*mp, 1.5*mp);
   else if (STREQ (button, "n"))
      HC_QScale_Object (str_buf, 1/(1.5*mp), 1/(1.5*mp), 1/(1.5*mp));
   HC_QTranslate_Object(str_buf,
                     mtx[3][0], mtx[3][1], mtx[3][2]);

if (STREQ (button, "c"))  /*  Change main<-->each  */
  {
     if (!eins )
          HC_QInsert_Text ("?root/flame1",
                           0.0, 0.0, 0.0,"not defined");
     else
       {
          HC_Flush_Segment ("?root/flame2");
            if ( segment_at )
                 HC_QInsert_Text ("?root/flame2",
                       0.0, 0.0, 0.0,"Each Segment Mode");
            else
                 HC_QInsert_Text ("?root/flame2",
                       0.0, 0.0, 0.0,"Main Object Mode");
          segment_at = !segment_at;
       }
  }
```

```
    else if (STREQ (button, "p"))
        {
           if (!eins )
              {
                 HC_QInsert_Text ("?root/flame1",
                                  0.0, 0.0, 0.0,"not defined");
                 return(1);
              }
           if (!segment_up )
             {
                Cancel_red();
                sprintf(str_buf,"%s",str_seg);
                HC_Show_Owner (str_buf, str_seg);



                sprintf(str_buf,"%s/*",str_seg);
                HC_QSet_Color ( str_buf, "edges = red");
                HC_Flush_Segment("?root/flame1");
                n = strlen( str_seg );                  /*get segname*/
                while ( str_seg[n] != '/' ) { --n; }
                i = 0;
                while ( str_seg[n] != '¥0' )
                  {
                     n++; segname[i] = str_seg[n]; ++i;
                  }                                     /* complete! */
                if ( !strcmp(segname,"object") )
                   segment_up = !segment_up;
                HC_QInsert_Text ("?root/flame1",
                                 0.0, 0.0, 0.0,segname);
             }
        }

    else if (STREQ (button, "z")) /*  Canceling each segment  */
        {
           if ( eins ) eins = !eins;
           Cancel_red();
           strcpy(str_seg,"¥0");
           HC_Flush_Segment("?root/flame1");
           HC_QInsert_Text ("?root/flame1",
                            0.0, 0.0, 0.0,"not defined");
           HC_Flush_Segment("?root/flame2");
           HC_QInsert_Text ("?root/flame2",
                            0.0, 0.0, 0.0,"Main Object Mode");
           if (!segment_at )  segment_at = !segment_at;
        }

    else if (STREQ (button, "q"))  return(0);

    return(1);
}
```

```
/*****************************************************************/

Cancel_red()
{
   HC_QUnSet_Color ( str_seg );
   HC_QSet_Color (str_seg, "lines = blue");
      if ( strcmp(str_seg, cur_seg) )
         {
            sprintf(str_buf,"%s/*",str_seg);
            HC_QUnSet_Color ( str_buf );
            HC_QSet_Color (str_buf, "lines = blue");
         }
}

/*****************************************************************/
```

Appendix. B
teleacc. cのソースファイル

```
/*****************************************************************/
#include        <stdio.h>
#include        <math.h>
#include        <math.h>
#include        <sys/types.h>
#include        <sys/ipc.h>
#include        <signal.h>

#define         STREQ(a,b)      (strcmp(a,b) == 0)
/*****************************************************************/
static char  **wargv;
static int   pid1, pid2, semid;
static char  button[31];
static float sita,fai,obj_sita,obj_r;
static float x, y, z;
static FILE  *fp3, *fp4;

main(argc, argv)
    int     argc;
    char  **argv;
{
    int    i, irrupt();
    char   event[100], cmd[128];
    FILE   *fopen(), *fp1, *fp2;

    wargv = argv;
    signal(SIGUSR1, irrupt);
    fp1 = fopen (*(wargv+1), "w");
    fp2 = fopen (*(wargv+2), "r");
    fp3 = fopen (*(wargv+3), "w");
    fp4 = fopen (*(wargv+4), "w");
    pid2 = 0;
    pid1 = getpid();
    semid = semget(1, 1, IPC_EXCL | IPC_CREAT);
    semop(semid, -1, 1);
    fprintf (fp1, "%d,%s¥n", pid1, "dummy");
    fclose (fp1);
    semop(semid, 1, 1);
```

```
HC_Open_Segment("?Picture");
    HC_Set_Driver_Options("subscreen=(-0.5,0.5,-0.5,0.5)");
    HC_Set_Camera_Position(10.0, 0.0, 0.0);
    HC_Set_Camera_Up_Vector(0.0, 0.0, 1.0);
    HC_Set_Heuristics ("no hidden surfaces");
    HC_Set_Color ("faces=white, edges=black");
    HC_Open_Segment ("light red");
        HC_Set_Color ("lights = red");
        HC_Insert_Distant_Light (0.5, 0.5, 0.5);
    HC_Close_Segment ();
    HC_Open_Segment ("light green");
        HC_Set_Color ("lights = green");
        HC_Insert_Distant_Light (0.5, 0.5, 0.5);
    HC_Close_Segment ();
    HC_Open_Segment ("light blue");
        HC_Set_Color ("lights = blue");
        HC_Insert_Distant_Light (0.5, 0.5, 0.5);



        HC_Close_Segment ();
    HC_Close_Segment ();

    HC_Open_Segment ("?Picture/object");
        HC_Set_Camera_Position(10.0, 0.0, 0.0);
        HC_Set_Camera_Up_Vector(0.0, 0.0, 1.0);
        HC_Set_Color ("faces = white, edges = black");
        if (argc < 2)
            sprintf (cmd, "read, name = z.hmf");
        else
            sprintf (cmd, "read, name = '%s'", wargv[1]);
        HC_Set_Metafile ("read, name = '../hmf/subj-31.hmf'");
        HC_UnSet_Metafile();

        HC_Set_Visibility ("polygons=on,lights=on");
        HC_Set_Selectability ("everything=on");
        HC_Set_Heuristics
        ("hidden surfaces, no polygon handedness,backplane cull");
        HC_Scale_Object (0.75, 0.75, 0.75);

        sita=0.0; fai=0.0; obj_sita=0.0; obj_r=3.0;
        x = 3; y = 0; z = 0;
        if (fp2 != NULL) {
           semop(semid, -1, 1);
           for (;fscanf(fp2, "%d,%s", &pid2,button) != EOF;) {
            Doit_subr (2);
            fprintf(fp3,"button=%s sita=%f fai=%f o-sita=%f o-radius=%f¥n",
            button,sita,fai,obj_sita,obj_r);
           }
           fclose(fopen(*(wargv+2), "w"));
           semop(semid, 1, 1);
        }
        if (pid2 != 0 )
          kill(pid2, SIGUSR1);
        for (;;) {
            HC_Get_Button (button);
            if (!Doit_subr (1)) break;
            fprintf(fp3,"button=%s sita=%f fai=%f o-sita=%f o-radius=%f¥n",
            button,sita,fai,obj_sita,obj_r);
```

```
        }
    HC_Close_Segment ();
    HC_Reset_System ();
}
/*****************************************************************/
/***                interrupt function                         ***/
/*****************************************************************/
irrupt()
{
     FILE    *fp;

     semop(semid, -1, 1);
     fp = fopen (*(wargv+2), "r");
     fscanf (fp, "%d,%s¥n", &pid2, button);
     fclose (fp);
     fp = fopen (*(wargv+2), "w");
     fclose (fp);
     semop(semid, 1, 1);
     Doit_subr (2);
     fprintf(fp3,"button=%s sita=%f fai=%f o-sita=%f o-radius=%f¥n",
     button,sita,fai,obj_sita,obj_r);
     signal(SIGUSR1, irrupt);
}

/*****************************************************************/
/***                command analysis function                  ***/
/*****************************************************************/


Doit_subr (type)

    int    type;
{
    static float mtxx[4][4]={{1.0, 0.0, 0.0, 0.0},{0.0, 1.0, 0.0, 0.0},
                             {0.0, 0.0, 1.0, 0.0},{3.0, 0.0, 0.0, 1.0}};
    static float pai=3.14159265;
    float d_x,d_y,d_z;
    float xx, xy, xz, yx, yy, yz, zx, zy, zz;
    float ux, uy, uz;
    FILE    *fp;
    static int    edges_on= 1,
                  faces_on= 1,
                  perspective_on= 1,
                  light1= 1,
                  light2= 1;


    if (STREQ (button, "right arrow"))
       {
         HC_Orbit_Camera(0.0,-fai);
         sita=sita+10.0;
         printf("fai = %f¥n", fai);
         HC_Orbit_Camera(10.0,fai);}
    else if (STREQ (button, "left arrow"))
       {
         HC_Orbit_Camera(0.0,-fai);
         sita=sita-10.0;
         printf("sita = %f¥n", sita);
         HC_Orbit_Camera(-10.0,fai);}
    else if (STREQ (button, "down arrow"))
       { fai=fai+5.0;
```

```
        printf("fai = %f¥n", fai);
        HC_Orbit_Camera(0.0,5.0);}
    else if (STREQ (button, "up arrow"))
       { fai=fai-5.0;
        printf("fai = %f¥n", fai);
        HC_Orbit_Camera(0.0,-5.0);}
/* ------------------------------------------------*/
    else if (STREQ (button, "Z")){
              obj_sita=obj_sita+5.0;
              x = obj_r*cos(obj_sita*pai/180);
              y = obj_r*sin(obj_sita*pai/180);
              HC_QRotate_Object ("?Picture/object/cone2", 0.0, 0.0, 5.0);}
    else if (STREQ (button, "W")){
              obj_sita=obj_sita-5.0;
              x = obj_r*cos(obj_sita*pai/180);
              y = obj_r*sin(obj_sita*pai/180);
              HC_QRotate_Object ("?Picture/object/cone2", 0.0, 0.0, -5.0);}
/* ------------------------------------------------ */
    else if (STREQ (button, "F")){
              d_x=0.5*cos(obj_sita/180.0*pai);
              d_y=0.5*sin(obj_sita/180.0*pai);
              x = x + d_x;
              y = y + d_y;
              obj_r=obj_r+0.5;
              HC_QTranslate_Object ("?Picture/object/cone2", d_x, d_y, 0.0);}
    else if (STREQ (button, "G")){
              d_x=0.5*cos(obj_sita/180.0*pai);
              d_y=0.5*sin(obj_sita/180.0*pai);
              x = x + d_x;
              y = y + d_y;
              obj_r=obj_r-0.5;
              HC_QTranslate_Object ("?Picture/object/cone2", -d_x, -d_y, 0.0);}
/* ------------------------------------------------ */
    else if (STREQ (button, "U")  ||   /*  up        */


             STREQ (button, "D")  ||   /*  down      */
             STREQ (button, "L")  ||   /*  left      */
             STREQ (button, "R")  ||   /*  right     */
             STREQ (button, "A")  ||   /*  backward  */
             STREQ (button, "B")) {    /*  foward    */
 /*         HC_Show_Net_Camera_Up_Vector (ux, uy, uz);  */
            xx = cos(fai*pai/180)*cos(sita*pai/180)/2;
            xy = sin(sita*pai/180)/2;
            xz = sin(fai*pai/180)*cos(sita*pai/180)/2;
            yx = -cos(fai*pai/180)*sin(sita*pai/180)/2;
            yy = cos(sita*pai/180)/2;
            yz = -sin(fai*pai/180)*sin(sita*pai/180)/2;
            zx = -sin(fai*pai/180)/2;
            zy = 0;
            zz = cos(fai*pai/180)/2;
/*          printf ("real up-vector = %f,  %f,  %f¥n", ux, uy, uz);
            printf ("cal  up-vector = %f,  %f,  %f¥n", yx, yy, yz);  */
            if (STREQ (button, "U")) {
               HC_QTranslate_Object ("?Picture/object/cone2", zx, zy, zz);
               x += zx;
               y += zy;
               z += zz;
               fprintf(fp4,"op = U fai = %f, sita = %f, dx = %f, dy = %f, dz = %
```

```
            z += yz;
            fprintf(fp4,"op = R fai = %f, sita = %f, dx = %f, dy = %f, dz = %
f¥n",
                    fai, sita, yx, yy, yz);
         }
         obj_r = sqrt(x*x + y*y + z*z);}
   /* ------------------------------------------------ */


f¥n",
                    fai, sita, zx, zy, zz);
         }
         if (STREQ (button, "D")) {
            HC_QTranslate_Object ("?Picture/object/cone2", -zx, -zy, -zz);
            x -= zx;
            y -= zy;
            z -= zz;
            fprintf(fp4,"op = D fai = %f, sita = %f, dx = %f, dy = %f, dz = %
f¥n",
                    fai, sita, -zx, -zy, -zz);
         }
         if (STREQ (button, "A")) {
            HC_QTranslate_Object ("?Picture/object/cone2", -xx, -xy, -xz);
            x -= xx;
            y -= xy;
            z -= xz;
            fprintf(fp4,"op = A fai = %f, sita = %f, dx = %f, dy = %f, dz = %
f¥n",
                    fai, sita, -xx, -xy, -xz);
         }
         if (STREQ (button, "B")) {
            HC_QTranslate_Object ("?Picture/object/cone2", xx, xy, xz);
            x += xx;
            y += xy;
            z += xz;
            fprintf(fp4,"op = B fai = %f, sita = %f, dx = %f, dy = %f, dz = %
f¥n",
                    fai, sita, xx, xy, xz);
         }
         if (STREQ (button, "L")) {
            HC_QTranslate_Object ("?Picture/object/cone2", -yx, -yy, -yz);
            x -= yx;
            y -= yy;
            z -= yz;
            fprintf(fp4,"op = L fai = %f, sita = %f, dx = %f, dy = %f, dz = %
f¥n",
                    fai, sita, -yx, -yy, -yz);
         }
         if (STREQ (button, "R")) {
            HC_QTranslate_Object ("?Picture/object/cone2", yx, yy, yz);
            x += yx;
            y += yy;
```

```
else if (STREQ (button, "s"))
    HC_Rotate_Object ( 10.0, 0.0, 0.0);
else if (STREQ (button, "a"))
    HC_Rotate_Object ( -10.0, 0.0, 0.0);

else if (STREQ (button, "z"))
    HC_Rotate_Object (0.0, 0.0, -10.0);
else if (STREQ (button, "w"))
    HC_Rotate_Object (0.0, 0.0, 10.0);

else if (STREQ (button, "j"))
    HC_Rotate_Object (0.0, -10.0, 0.0);
else if (STREQ (button, "k"))
    HC_Rotate_Object ( 0.0, 10.0, 0.0);
/*****/
else if (STREQ (button, "i"))
    HC_Zoom_Camera (1.5);
else if (STREQ (button, "o"))
    HC_Zoom_Camera (1/1.5);

else if (STREQ (button, "H")) {
    printf ("¥n Postscript Fileout .... ¥n¥n");
    HC_Open_Segment ("?Driver/postscript/sample.ps");
        HC_Set_Visibility ("windows=on");
        HC_Include_Segment ("?picture");
        HC_Update_Display();
        HC_Set_Visibility ("windows=off");
        HC_Flush_Segment (".");
    HC_Close_Segment ();
}

else if (STREQ (button, "e")) {
    if (!edges_on)
        HC_Set_Visibility ("edges = on");
    else {
        HC_Set_Visibility ("edges = off");
        if (!faces_on) {
            HC_Set_Visibility ("faces = on");
            faces_on = 1;
        }
    }
    edges_on = !edges_on;
}

else if (STREQ (button, "f")) {
    if (!faces_on)
        HC_Set_Visibility ("faces = on");
    else {
        HC_Set_Visibility ("faces = off");
        if (!edges_on) {
            HC_Set_Visibility ("edges = on");
            edges_on = 1;
        }
    }
    faces_on = !faces_on;
}

else if (STREQ (button, "r")) {
    HC_QSet_Visibility ("?Picture/light red", light1 ? "off" : "on");
```

```
        light1 = !light1;
    }
    else if (STREQ (button, "b")) {
        HC_QSet_Visibility ("?Picture/light blue", light2 ? "off" : "on");
        light2 = !light2;
    }
    else if (STREQ (button, "p")) {
        if (perspective_on)
            HC_Set_Camera_Projection ("orthographic");
        else
            HC_Set_Camera_Projection ("perspective");
        perspective_on = !perspective_on;
    }
    else if (STREQ (button, "control c") ||
             STREQ (button, "control d") ||
             STREQ (button, "control z") ||
             STREQ (button, "q")){return 0;}

    else if (STREQ (button, "X")) {
            HC_Set_Modelling_Matrix (mtxx);

    }

    if ((STREQ (button, "Z") ||
         STREQ (button, "W") ||
         STREQ (button, "F") ||
         STREQ (button, "U") ||
         STREQ (button, "D") ||
         STREQ (button, "L") ||
         STREQ (button, "R") ||
         STREQ (button, "A") ||
         STREQ (button, "B") ||
         STREQ (button, "G")) && (type == 1)) {
       semop(semid, -1, 1);
       fp = fopen (*(wargv+1),"a");
       fprintf (fp, "%d,%s¥n", getpid(), button);
       fclose (fp);
       semop(semid, 1, 1);

       if (pid2 != 0)
          kill(pid2, SIGUSR1);
    }
    return 1;
}
/*****************************************************************/
```