

# バーチャル・フラワーアレンジメント 「仮想現実いけばな」の制作

近藤 泰允

Yasumitsu KONDO

名古屋文理大学 情報文化学部 情報文化学科 はせがわ研究室  
HASEGAWA Laboratory, Department of Information Culture, Nagoya Bunri University

平成16年1月30日 提出

## 要旨

最近の3次元コンピュータグラフィックス(3D-CG)の目覚ましい進歩により、3D-CGで本物に近く見えるほどの物体の表現ができるようになった。3D-CGで描かれた仮想物体を操作できるようにすれば、本物に触れなくとも様々なことをコンピュータ上で体験できる仮想現実世界が構築でき、様々な分野への応用が期待できる。その中でも今回は、『いけばな』をテーマに選んでインタラクティブな仮想現実体験システムの構築を目指した。現実の「生け花」は、時間が経つと花が枯れてしまったり流儀や作法の知識が要求されたりするため、素人が個人で気軽に楽しむのは容易でない。今回、本格的に「生け花」を習得するのではなく流儀にとらわれずにデザインだけを考えて作品を作ることができれば、専門知識のない素人でも楽しめるのではないかと考え、3D-CGで『花』と『器』を表現し、パソコン上で気軽に『いけばな』ができる仮想現実システムを開発した。制作した『仮想現実いけばな(バーチャル・フラワーアレンジメント)』は、あたかも本当に生け花をするかのように、コンピュータ上で花を器に配置するなどのデザインを楽しむものである。開発には、Borland CとOpenGLを用いた。『仮想現実いけばな』を使えば、CGで描かれた花を3次元空間内の平行移動と回転移動によってインタラクティブに動かして器に配置し、自分が思い描いたとおりのデザインができる。これにより、誰でも気軽に花のCGを使った3D-CGのデザイン作品を生み出すことができる。『仮想いけばな』は、単に現実の「生け花」を疑似体験するというだけでなく、現実の「生け花」とは異なる新しいタイプの芸術分野にもなりうると考えられる。

## 1. はじめに

近年、3次元コンピュータグラフィックス(3D-CG)の目覚ましい進歩により、CGで描かれた物が本物に近いほどに見える様になっ

た。本物に触れなくとも、インタラクティブな3D-CGにより、本物に近い見た目でも色々なことを仮想現実の世界で体験できる。

一方、現実世界を考えると、たとえば現実

の「生け花」では、生きている本物の花を使うため、素材が高価であったり、時間が経つと花が枯れてしまったりする。花への水の与え方や保存の仕方についての知識が必要とされ、器の使い方や花の配置についても流儀や専門知識が必要である。そのため、素人が個人で気軽に楽しむのは容易でない。そこで、今回、本格的に「生け花」を習得するのではなく、流儀や専門知識にとらわれずに、デザインだけを楽しんで作品を作ってみたいと考え、3D-CGで『花』と『器』を表現し、PC上のバーチャル世界で気軽に『いけばな』が体験できる様にする事を目指した。

「いけばな」には、日本の「生け花」の他に、西洋の「フラワーアレンジメント」がある<sup>1)</sup>が、開発した『仮想現実いけばな(フラワーアレンジメント)』システムを使えば洋の東西や流儀を問わず自由な発想でデザインを楽しむことができる。以下に、まず、開発したシステムの概要を述べ、次に、プログラム開発の内容を説明した後、実際にシステムを利用して作成した、仮想『いけばな』作品の例を紹介して考察する。

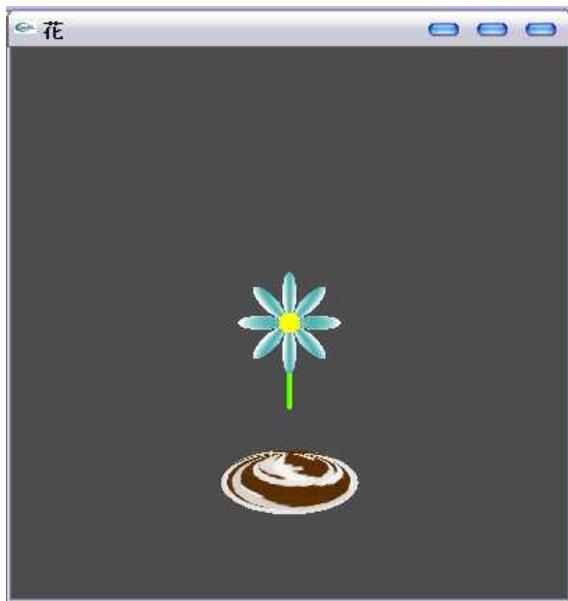


図1. 『仮想現実いけばな』の初期画面

## 2. 開発したシステムの概要

開発したシステムの概要と利用法について以下に順に述べる。

### 2.1. 動作環境

『仮想現実いけばな』は、1つの実行ファイル(ikebana.exe)と合計20個の画像データファイル(次項の表1・表2参照)から成り、OpenGLのGLUT(OpenGL Utility Toolkit)<sup>2)・5)</sup>(詳細は後述3.1.項参照)のDLL(Dynamic Link Library)がインストールされたMS-Windows上で作動する。

### 2.2. 起動方法

プログラムの起動は、コマンドプロンプトからコマンドラインで行う。起動の際にコマンドライン引数として、花の数と器の柄をそれぞれ1~10の番号で指定する。たとえば、>ikebana 3 1と入力して起動すれば、花の数は3本、器の柄は1番を指定したことを意味する。引数を与えずに起動した場合は、利用法が「ikebana(花の数)(器番号)」であることが表示されるようにしてある。引数を指定して起動すれば、ウィンドウが開き、花と器が表示される。

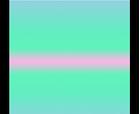
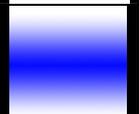
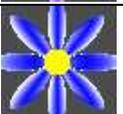
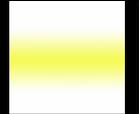
図1に、『仮想現実いけばな』を起動した初期画面の例を示す。花が複数ある場合でも、初期画面ではすべて同じ位置に重なっているため図1のように1本だけのように見える。

### 2.3. 花の本数と柄および器の柄

起動時に設定する花の本数と器の柄は、それぞれ表1と表2に示すとおりである。現バージョンでは、花、器とも、形は1種類で、柄はそれぞれ10種類利用できる。

花の本数は1~10の数で指定する。たとえば花の数を3と指定すれば、花番号1~3の3本の花が利用できる。表1に示すように、花番号1~10の花ごとに異なる柄の画像データが割り当てられてテクスチャマッピング

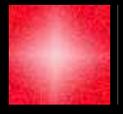
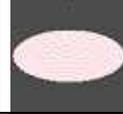
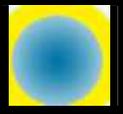
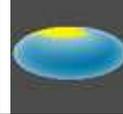
表 1 . 花番号と花びらの柄

花番号	ファイル名	画像データ	表示される花
1	hana000.bmp		
2	Hana001.bmp		
3	Hana002.bmp		
4	Hana003.bmp		
5	Hana004.bmp		
6	Hana005.bmp		
7	Hana006.bmp		
8	Hana007.bmp		
9	Hana008.bmp		
10	Hana009.bmp		

(texture mapping)される。花番号 1 ~ 10 の各花の柄は、それぞれ、データファイル名 hana000.bmp ~ hana009.bmp の 64×64dot サイズの 24bit 色ビットマップファイルをマッピングしたものとなる。

器は、現バージョンでは形は楕円版状の平型の 1 種類であるが、表 2 のような 10 種類の柄を指定することができる。器番号 1 ~ 10 の器の柄は、花の場合と同様に、それぞれファ

表 2 . 器番号と器の柄

器番号	ファイル名	画像データ	表示される器
1	utuwa000.bmp		
2	utuwa001.bmp		
3	utuwa002.bmp		
4	utuwa003.bmp		
5	utuwa004.bmp		
6	utuwa005.bmp		
7	utuwa006.bmp		
8	utuwa007.bmp		
9	utuwa008.bmp		
10	utuwa009.bmp		

イル名 utuwa000.bmp ~ utuwa009.bmp の画像データによって決まる。

なお、花・器とも、64×64dot サイズの 24bit ビットマップファイル形式の別の画像データを用意して、それぞれ表 1 または表 2 に示されたファイル名で保存すれば、対応する花または器の柄を変更することができる。たとえば、図 2 のように hana000.bmp の画像データ (テクスチャ) を変更すれば、それに応じ

て花番号 1 の花びらの柄が変更される。

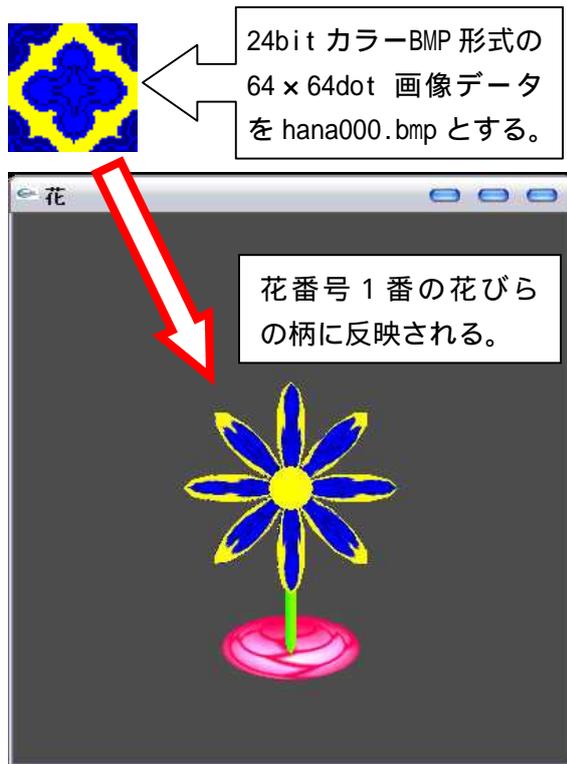


図 2 . 花の柄 (テクスチャ) の変更例

#### 2 . 4 . 花の移動

『仮想現実いけばな』では、図 3 のように物体 (花や器) は X 軸・Y 軸・Z 軸を設定した空間座標上に表現される。

現バージョンでは、移動はすべてキーボード操作で行う。

各花は、それぞれ独立に、平行移動と回転移動によって自由に配置できる。平行移動は、左右 (X 軸方向の + - )・上下 (Y 軸方向の + - )・前後 (Z 軸方向の + - ) (図 4 ) であり、回転移動は X 軸・Y 軸・Z 軸についてそれぞれ原点から見て右回りである (図 5 )。回転軸の原点は各花の花びらの中心核の位置である。

花を操作するための操作キーを、表 3 にまとめる。ユーザは、" 1 " キーを押すことで移動対象とする花を選び、" W " " S " " Q " " E " " A " " D " の各キーで、移動対象の花の平行移動、" X " " C " " Z " の各キーで回転移動を実行する。

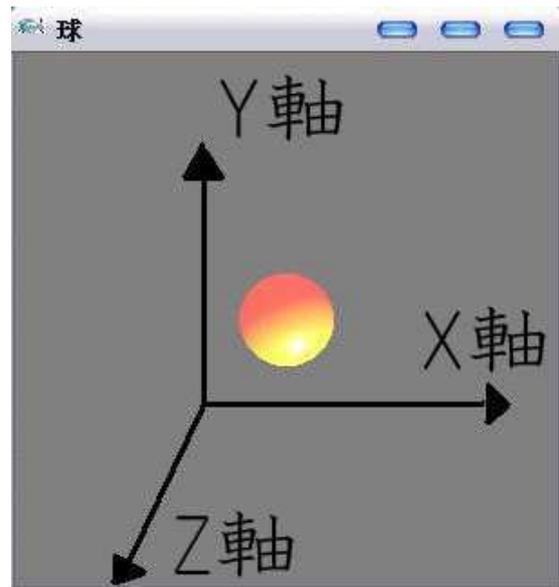


図 3 . 立体オブジェクトと 3 次元空間座標

表 3 . 花の移動のキー操作一覧

キー	動作
" 1 "	移動する花の切替え (花番号を + 1 )
" W "	上に移動 (Y 軸の + 方向に平行移動)
" S "	下に移動 (Y 軸の - 方向に平行移動)
" Q "	前に移動 (Z 軸の + 方向に平行移動)
" E "	後に移動 (Z 軸の - 方向に平行移動)
" A "	右に移動 (X 軸の + 方向に平行移動)
" D "	左に移動 (X 軸の - 方向に平行移動)
" X "	X 軸回転 (X 軸の周りに右回転)
" C "	Y 軸回転 (Y 軸の周りに右回転)
" Z "	Z 軸回転 (Z 軸の周りに右回転)

表 4 . 視点 (全物体) の回転のキー操作一覧

キー	動作
" B "	X 軸回転 (X 軸の周りに右回転)
" N "	Y 軸回転 (Y 軸の周りに右回転)
" V "	Z 軸回転 (Z 軸の周りに右回転)

#### 2 . 5 . 視点 (全物体) の回転

花ごとの移動とは別に、全物体 (すべての花と器) を同時に回転することもできる。回転は、表 4 に示す X 軸・Y 軸・Z 軸回転の 3 通りである。この場合の座標軸の原点は物体の位置に関わらず画面の中心、すなわち初期

画面（図 1）で表示される花の核の位置に固定されている。

画面の中心を原点に全物体を同時に回転することは、視点を回転することに相当する。この操作によって、空間内での花の配置を把握することや、デザインされた『いけばな』を視点を変えて鑑賞することができる（図 6）。

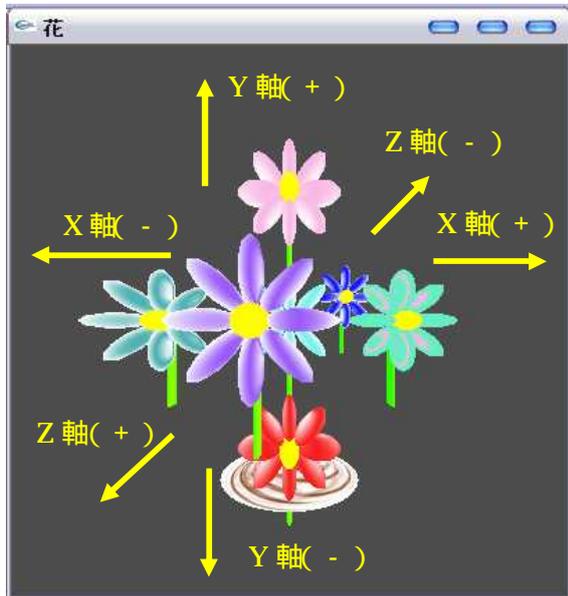


図 4 . 花の平行移動



図 5 . 花の回転移動



図 6 . 視点（全物体）の回転

### 3 . システムの開発について

今回、『仮想現実いけばな』システムがどのように開発されたかについて述べる。

#### 3 . 1 . 開発言語について

開発言語は C を使い、Windows 上で開発した。コンパイラは Borland C++ 5.5 を使った。このコンパイラは Borland 社のホームページ<sup>6)</sup>から無料でダウンロードできる。

また、3D-CG の表現には、OpenGL の GLUT を用いた。これにより、ほとんどの OS で実行できる（ソースレベル互換の）プログラムとなり、インタラクティブな 3D-CG が実現できた。プログラムは、モデリング、テクスチャマッピング、キーの受け付け、物体の移動などの複雑な処理を含むが、約 350 ステップで記述できた。図 7・図 8 にソースプログラムの一部を示す。なお、GLUT は、SGI 社のホームページなど<sup>7) 8)</sup>から無料で入手できる。GLUT は、`glut.h`、`glut32.lib`、`glut32.dll` の 3 つのファイルから成り、それぞれ所定の場所に置いて利用する。ただし、`glut32.lib` は、Borland C++ 用に生成したものをを使う必要がある<sup>2) 8)</sup>。

### 3.2. キー入力の受け付け

キー入力は、図7に一部を示す key\_press

関数で受け付ける。ここで前出の表3と表4

のキーのすべてを受け付けて処理している。

```
#include<GL/glut.h>
< 中略 >
int rotx[MAXFLNUM],roty[MAXFLNUM],rotz[MAXFLNUM]; //各花の角度
double dx[MAXFLNUM], dy[MAXFLNUM], dz[MAXFLNUM]; //各花の中心座標
int rotxAll, rotyAll, rotzAll; //全物体の角度
< 中略 >
//*****
// 移動（平行移動・回転）に関する関数群
//*****
void xrotAll(void){ rotxAll+=12; rotxAll%=360; draw(); } //X軸視点（全体）回転
void yrotAll(void){ rotyAll+=12; rotyAll%=360; draw(); } //Y軸視点（全体）回転
< 中略 >
//*****
// キー操作
//*****
void key_press(unsigned char key) //キー入力受け付け
{
    if(key=='b') xrotAll(); //X軸視点（全体）回転
    if(key=='n') yrotAll(); //Y軸視点（全体）回転
}
```

図7. ソースプログラムの一部（キー入力の受け付けと視点の回転）

```
//*****
// 全物体の描画
//*****
void draw(void)
{
    int i;

    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glPushMatrix();
    glColor3d(0.0,0.0,0.0);
    glTranslated(0.0,0.0,0.0);
    glPopMatrix();
    for (i = 0 ; i < flwrnum ; i++) { //花の数だけ花を描く
        glPushMatrix();
        glRotated((double)rotxAll,1.0,0.0,0.0); //全体の回転
        glRotated((double)rotyAll,0.0,1.0,0.0);
        glRotated((double)rotzAll,0.0,0.0,1.0);
        glTranslated(dx[i],dy[i],dz[i]); //各花の平行移動
        glRotated((double)rotx[i],1.0,0.0,0.0); //各花の回転
        glRotated((double)roty[i],0.0,1.0,0.0);
        glRotated((double)rotz[i],0.0,0.0,1.0);
        glColor3d(0.0,0.0,0.0);
        glTranslated(0.0,0.0,0.0);
        kaku(); //花の核を描く
        hiras(i); //花びらを描く
        kuki(); //花の茎を描く
        glPopMatrix();
    }
    glPushMatrix();
    glRotated((double)rotxAll,1.0,0.0,0.0); //全体の回転
    glRotated((double)rotyAll,0.0,1.0,0.0);
    glRotated((double)rotzAll,0.0,0.0,1.0);
    utuwa(); //器を描く
    glPopMatrix();
    glutSwapBuffers();
    glFlush();
}
```

図8. ソースプログラムの一部（オブジェクトの描画）

### 3.3. モデリング

花などの物体は、ソリッドモデル (solid model) で表現してある。図8に見える kaku 関数の中では、花の核となる部分を GLUT の glutSolidSphere 関数で球を描いて表現している。同じように、図8の hiras 関数では、花びらの部分を、球を描く関数を用いて表した。まず glTranslated 関数で Y 軸方向の位置を指定し、glScalef 関数で球を Y 軸方向に伸ばして一つの花びらの形(楕球)を作った。そして、glRotated 関数で核の周りに 45° ずつ回転しながら 8 つの楕球を配置して花びらを作った。茎の部分は、図8の kuki 関数の中で、gluCylinder 関数で筒を 3 つ作って表現した。器は、図8の utuwa 関数の中で、球をもとに glScalef 関数で球を平らに伸ばして表現した。

### 3.4. テクスチャのマッピング

モデリングした物体のうち、花びらと器では、前述のように、画像データ(テクスチャ)を BMP ファイルから読み込んでマッピングして柄を表現した(前掲の表1・表2・図2を参照)。

### 3.5. 花の数

図7に見える MAXFLNUM は、システムであつかう花数の最大値を意味し、現バージョンでは「#define MAXFLNUM 10」のように最大 10 本の花を表示できる仕様となっている。利用できる花の数 (MAXFLNUM) は、起動時にコマンドライン引数で指定され、その値はグローバル変数 flwrnum(図8中段参照)に格納されている。

### 3.6. 移動操作

図7上段の行で定義され、図8中頃の行で利用されているように、花番号 i の花の位置は、( dx[i], dy[i], dz[i] ) で表される。平行移動の操作では、対応するキーが 1 回押される

たびに、それぞれ X または Y または Z 座標の + または - の方向へ 10 だけ移動する。平行移動は、dx[i] または dy[i] または dz[i] の値を + または - 10 だけ増減して行う。また、回転移動では、回転角の値は X, Y, Z ごとに (rotx[i], roty[i], rotz[i]) (番号 i の花) および (rotxAll, rotyAll, rotzAll) (視点) で表され(図7) キーが 1 回押されるたびに、12° だけ回転する(たとえば図7中段参照)。

## 4. 『仮想現実いけばな』システムを使った『いけばな』作品の例

現実の「いけばな」では、日本の「生け花」には大きな特徴として造形の非対称性がある。それに対して、西洋のフラワーアレンジメントでは均衡がとれた対称形が基本となる<sup>1)</sup>。

『仮想現実いけばな』では、こうした現実の「いけばな」を意識しながら、現実の場合よりも気軽に『いけばな』作品を作ることができる。また、『仮想現実いけばな』では、現実の「いけばな」とは無関係に、空間内に自由に花を配置して、仮想世界ならではの『いけばな』を楽しむこともできる。

以下に、『仮想現実いけばな』で作った仮想『いけばな』作品を 3 例示す。それぞれ、(1) 日本の生け花を意識した非対称型、(2) 西洋フラワーアレンジメント風の対称型、(3) 現実と無関係な超現実型、の 3 種類である。

#### (1) 非対称型『いけばな』(図9)

日本の「生け花」をイメージして、図9のように 6 本の花を非対称な配置にしてみた。ここでは、花を互い違いに配置し、なおかつあと 1 本を手前に配置してみた。

#### (2) 対称型『いけばな』(図10)

図10の作品では、西洋のフラワーアレンジメントをイメージして花を配置した。10 本の花がツリー状に配置され、整った均衡による美しさを生み出している。

(3) 超現実型『いけばな』(図11)

『仮想現実いけばな』では、現実世界では不可能な場所に花を配置することも可能である。図11の作品では、花を空中に配置して、ひとつのデザインとして表現した。さらに、『仮想現実いけばな』では、花どうしや花と器を重ねて同じ空間内に置くこともできる。

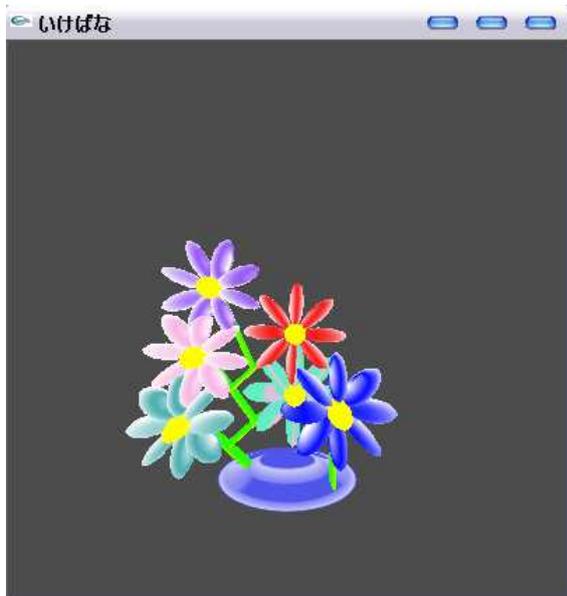


図9.『いけばな』の作品例(非対称型)



図10.『いけばな』の作品例(対称型)



図11.『いけばな』の作品例(超現実型)

## 5. 考察

今回、インタラクティブな3D-CGによってパソコン上で誰でも気軽に『いけばな』を楽しめる仮想現実システムを作成し、実際に使用して作品を作ってみた。この『仮想現実いけばな』では、生け花の知識がなくても現実世界の疑似体験ができるというだけでなく、現実の「いけばな」ではできないような作品も作ることができる。花のCGを自由に配置して構成したデザイン作品は、デジタル画像なので、そのままインターネット上に公開することも容易であり、さらに加工して各種のデザインに利用することもできる。このように、仮想『いけばな』は、新しい芸術分野にもなる可能性があるといえる。

今後、このようなインタラクティブ3D-CGによる仮想現実体験システムは、さらに様々な分野で応用が進むと考えられる。

## 6. 今後の課題

現バージョンの『仮想現実いけばな』システムに残された課題を挙げる。(1)物体の移動に物理的な制限を加え、花が器を突き抜けたら花どうしが重なったりしないようにして

リアリティーの向上を図る。( 2 )現状では棒の様な茎を、現実の花の様に曲げたり切ったりできるようにする。なお、現プログラムでも茎の部分は3つの筒 ( cylinder ) を重ねてモデリングしてある。つなぎ目の部分で折ったり切ったりできるようにする事が課題である。( 3 )花や器の形を増やして選択できるようにする。( 4 )現状ではキーボードでの操作しかできないが、マウスなどで操作ができる様にしてユーザインターフェースを改善する。

これらの課題を改善すれば、さらに利用性の高いプログラムとなると思われる。

[ 参考文献 ]

- 1 ) 高木隆司 : 「かたちの事典」, 丸善 ( 2003 )
- 2 ) 関根詮明・安居院猛 : 「応用 OpenGL グラフィックス」, 森北出版 ( 2003 )
- 3 ) 安居院猛・関根詮明 : 「入門 OpenGL グラフィックス」, 森北出版 ( 2001 )
- 4 ) エドワード・エンジェル : 「OpenGL 入門」, ピアソン・エデュケーション ( 2002 )
- 5 ) シリコングラフィックス社 ( SGI ) ホームページ : OpenGL に関する情報など ,  
<http://www.sgi.com/software/opengl/>  
<http://www.opengl.org/>
- 6 ) Borland 社ホームページ :  
Borland C++ Compiler 5.5 ダウンロード ,  
<http://www.borland.co.jp/cppbuilder/freecompiler/>
- 7 ) Nate Robins ホームページ : Windows 用 GLUT ( ただし Microsoft Visual C++ 用 ) のダウンロード ,  
<http://www.xmission.com/~nate/opengl.html>
- 8 ) 森北出版ホームページ :  
Borland C++ 用の GLUT のダウンロード ,  
<http://www.morikita.co.jp/soft/OpenGL/>