［技術報告］

３次元人体形状スキャナーの開発*

長谷川　辰雄**、中村　吉信***

多品種・適量生産の現状において、個人ごとの要求に合った製品が求められている。本研究では靴製造の新たな手法として、靴専用のCADを開発した。個人にフィットした靴の制作の現状は、数カ所の必要な寸法を計測することで行われている。この数カ所を自動計測する装置は現在開発中であるが、計測後の数カ所のデータから靴型のＣＡＤデータを制作するには、手作業により多大の時間を要する。そこで、必要な数カ所のデータから簡単に靴型を生成するＣＡＤを開発した。これによって、靴型のＣＡＤデータ制作の時間短縮が可能となった。
キーワード：靴型、CAD、自動計測

Development of 3D Scanner for Measurement of Human Body

HASEGAWA Tatsuo and NAKAMURA Yoshinobu

Various products that fits in individual are required in flexible manufacturing system. We developed a new CAD software for shoes design. The produce shoes that best-suited to individual size is that measuring necessary a few point of human feet. The 3D measurement instrument is under development. But, it has much time to design shoe-form with a few mesurement point from the 3D measurement instrument. In this reports, we described the development of the original three-dimensional CAD system for easy-to-puroduce shoe-form.

key words: shoe-form, CAD, Automatic measurement

１　緒　　言

一定規格の大量生産による低コストの製品が主流となっている現状において、靴、衣服、医療用具など、人が身につける物に関して、個人にフィットした製品が望まれている。特に装具などの医療福祉器具などは、自身の形状にフィットした製品が望まれている。このような現状から、人体の形状計測から製品製造までの工程を効率よく進める技術が重要である。現状の人体の形状計測は人手による計測が一般的であるが、人体用の自動計測器も市販されているが普及していない。理由は高額の他に、計測からＣＡＤデータを完成するまでに長時間を必要とすることにある。このことから、本研究では、計測からＣＡＤデータを生成する過程の中で、ＣＡＤの操作に注目し、より簡単に目的とする形状を描くことができないかを検討した。即ち、目的の形状を直接描くのではなく、基準となる線を動かすことで形状を変形させる手法を開発した。基準線は形状との幾何情報の関係を持つので、基準線の移動により形状が変形することから、始めから直接形状を描くよりも、基準線の移動だけで形状を描くことが可能となる。

２　開発方法とその構想

２－１　開発

開発するＣＡＤは、基準線と呼ばれる線と、描画する形状との間に連携情報を構築する。本ＣＡＤの基本構成は、以下に示す３つの基本機能からなる。

① Pattern ＣＡＤ

形状の基本原型と基準線の情報を格納したＣＡＤ情報データベース

② Proportion ＣＡＤ

数カ所の３次元計測値を元に、Pattern ＣＡＤから必要な形状を読み込み、測定器で計測した数カ所の３次元計測値をパラメータとして与えることで、目的の形状を生成するＣＡＤ

③ Gestalt ＣＡＤ

上記のPattern ＣＡＤとProportion ＣＡＤの機能を統合化するための環境

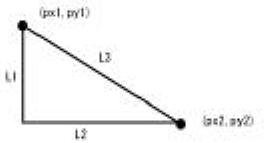
＊技術パイオニア養成事業
＊＊特産開発デザイン部
＊＊＊（株）でん

２－２　幾何計算プログラムの作成

プログラムはWindows系の言語に制限されないＤＬＬ(Dynamic Link Library)に基本的な幾何計算ルーチンを開発する。これによって、Visual BasicやVisual C++[1]の両方の言語から開発した幾何計算ルーチンを使うことができる。DLLは、メインプログラムからコールされて実行されるが、コールされた時点でコンピュータ上のメモリにロードされる。DLLを使用しないアプリケーションプログラムは、プログラムのすべてがコンピュータ上のメモリに常駐するために、使用されないプログラムもメモリ資源を使っていることになる[2]。このことから、DLLプログラムは、コンピュータのメモリ資源を有効に使用する手段となる[3]。開発したDLLプログラムを図１及び図２に示す。図１は平面及び立体の二点のピタゴラスの定理をDLLプログラム化したものである。Geometry DLL仕様の箇所は、関数pitago2d()と関数pitago3d()という名前のDLLプログラムの宣言と、その引数パラメータの宣言を行っている。pitago2d()は、２次元の点（px1,py1)と(px2,py2)の２点を入力引数とし、その距離を計算するプログラムである。ｐｉｔａｇｏ３ｄ（）は、３次元の点(px1,py1,pz1)と（px2,py2,pz2)の２点を入力引数とし、その距離を計算するプログラムである。図２は楕円上の二点から長径と短径を求めるDLLプログラムである。Geometry DLL仕様の箇所は、ellipse_r()という名前のDLLプログラムの宣言と、その引数パラメータの宣言を行っている。ellipse_r()は、楕円上の２点(x1,y1)と(x2,y2)を入力引数とし、楕円の長径及び短径を計算し、それぞれａ、ｂに値を返す。

```
Geometry DLL 仕様
  //平面の二点のピタゴラスの定理
  GEOMETRY_API double __stdcall pitago2d(double px1,double py1,double px2,double py2);
  //立体の二点のピタゴラスの定理
  GEOMETRY_API double __stdcall pitago3d(double px1,double py1,double pz1,
             double px2,double py2,double pz2);
```

```
//平面の二点のピタゴラスの定理
 GEOMETRY_API double __stdcall pitago2d(double px1,double py1,double px2,double py2)
 {
  double L1,L2,L1s,L2s,L3;
  L1=fabs(px1-px2);
  L2=fabs(py1-py2);
  L1s=L1*L1;
  L2s=L2*L2;
  L3=sqrt(L1s+L2s);
  return L3;
 }
```




```
//立体の二点のピタゴラスの定理
GEOMETRY_API double __stdcall pitago3d(double px1,double py1,double pz1,
           double px2,double py2,double pz2)
{
 double L1,L2,L3,L1s,L2s,L3s,L4,L5;
  L1=fabs(px1-px2);
  L2=fabs(py1-py2);
  L3=fabs(pz1-pz2);
 L1s=L1*L1;
 L2s=L2*L2;
 L3s=L3*L3;
 L4=sqrt(L1+L2);
 L5=sqrt(L4+L3s);
 return L5;
}
```

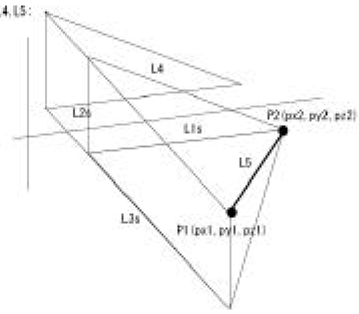


図１　ピタゴラスの定理計算

```
Geometry DLL 仕様
//楕円上の二点の座標から長径と短径を求める
  GEOMETRY_API void __stdcall ellipse_r(double x1,double y1,double x2,double y2,
                                        double *a,double *b);
```

```
//楕円上の二点の座標から長径と短径を求める
  GEOMETRY_API void __stdcall ellipse_r(double x1,double y1,double x2,double y2,
                                        double *a,double *b)
  {
   double as,bs;
   if(x1==0)
   {
    as=fabs(y1);
    bs=fabs(x2);
   }
   else if(y1==0)
   {
    as=fabs(x1);
    bs=fabs(y2);
   }
   else
   {
   as=sqrt((x2*y1-x1*y2)/(y1-y2));
   bs=sqrt((x1*y2-x2*y1)/(x1-x2));
   }

   *a=fabs(as);
   *b=fabs(bs);
  }
```

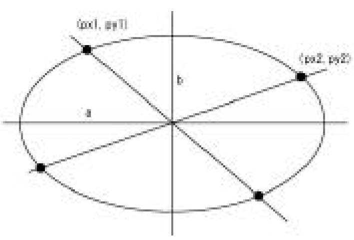


図２　楕円上の二点から長径と短径を求める

２－３　３次元計測用座標変換プログラムの作成

開発中の３次元計測器は、複数台のカメラを使い、三角測量によって計測する。このとき、カメラの位置と角度が既知でなければならない。カメラの位置及び角度を算出するために、指標となる寸法既知形状の対象物を撮影して行われる。この作業は、カメラ位置のキャリブレーションと呼ばれ、画像処理による計算で求められる。図３に、このキャリブレーションに必要な座標変換DLLプログラムを示す。convert_xy()は、撮影した指標（マーカ）からカメラまでの距離、角度を入力パラメータとし、カメラ原点で計算されている３次元計測値をマーカ原点に変換するプログラムである。

```
Geometry DLL 仕様
//二台のカメラで計測した結果を  world 座標に、座標変換する
  GEOMETRY_API void __stdcall convert_xy(int pnum,double p[][2],double cl1,
                                         double angle1,double rx,double ry,double rz,
                                          double ph[][2]);

for(int i=0;i<=pnum;i++)
{
 p[i][2]=p[i][2]-cl1;//カメラからマーカーに原点を移す
   //カメラ間の平面上、Z軸軸方向の回転変換
 rotate_p(0,0,p[i][0],p[i][1],angle1,&p[i][0],&p[i][1]);
 rotate_p(0,0,p[i][0],p[i][1],rz,&p[i][0],&p[i][1]);//Z軸方向の回転変換
 rotate_p(0,0,p[i][0],p[i][2],ry,&p[i][0],&p[i][2]);//Y軸方向の回転変換
 rotate_p(0,0,p[i][1],p[i][2],rx,&p[i][1],&p[i][2]);//X軸方向の回転変換
 ph[i][0]=p[i][0];
 ph[i][1]=p[i][1];
 ph[i][2]=p[i][2];
}
```

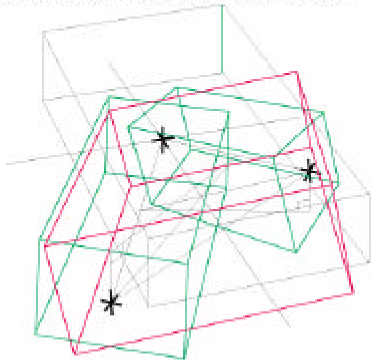


図３　座標変換プログラム

３　実験結果

開発したDLLプログラムの実行結果を図４に示す。図４は統合環境のためのPatternt CADの初期実行画面である。メインプログラムをVisual Basicを使って作成し、幾何計算のDLLプログラムを呼び出して実行した。ここで設計した対象物は靴型である。開発したＣＡＤ環境は、ワイヤーフレームデータとして構築されるが、面を生成する機能が無いため、市販ＣＡＤのLightWave3Dの機能で面を生成した。図５にその結果を示す。図６は、面を構成した足形の表示をシェーディング表示した結果である。

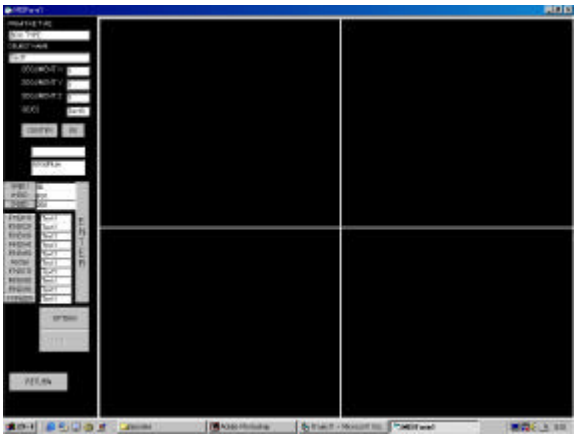


図４　開発CADの実行画面

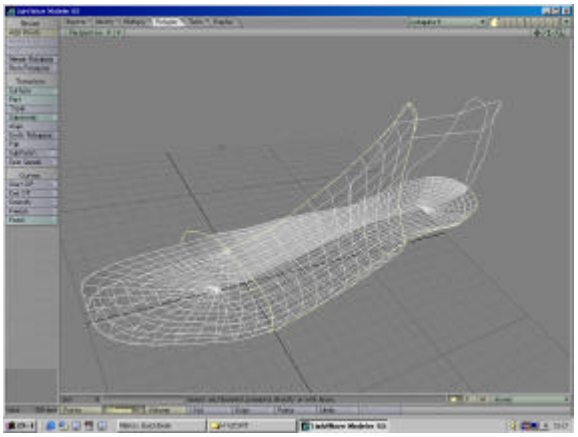


図５　靴型データの面の生成

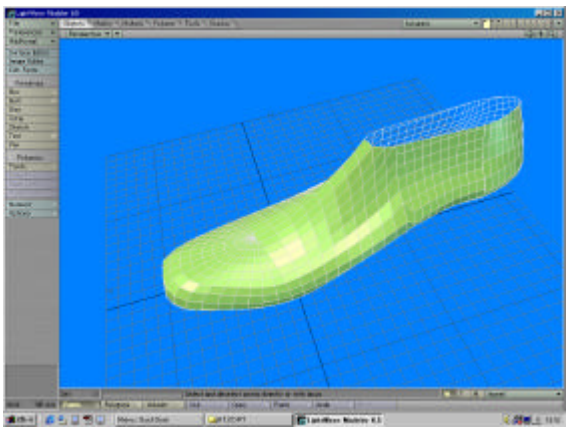


図６　靴型データのシェーディング表示

４　開発手法システムの特徴

開発したCADは、３次元のワイヤーフレームのデータを生成する。Gestalt CAD及びPatter CADは平面や曲面を生成する機能を備えていないため、 LightWave3Dの機能で面を構築した。本システムは、図７に示すような少数カ所をカメラで撮影して、その計測値を元に、開発したＣＡＤで靴型を設計する。既知の座標点数が少数である形状に平面や曲面を構築する作業は、ポリンゴン生成とよばれ、様々な手法が提案されているが、プログラミングに高度な技術を必要とする。開発したＣＡＤはポリゴン生成の機能を有していないが、必要に応じて実装する予定である。

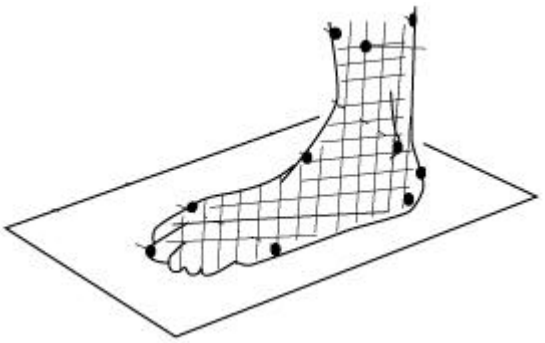


図７　靴型データのシェーディング表示

５　結　　言

コンピュータのメモリ資源を有効に活用するDLLプログラミングにより、独自のCADシステムを構築することができた。実験ではメインプログラムをVisual Basicで行っているが、Visual C++[4]でもDLLプログラムの実行を確認した。システム全体としては、３次元計測器からの計測データを、開発したＣＡＤに取り込んで設計し、ＣＡＭ（Computer Aided Manufacturing)で試作、加工を行う流れとなっている。測定器は現在開発中であり、今回開発したプログラムと統合化[5]してシステムを構築していく予定である。

文　　献

1) 桜田幸嗣，田口景介:Visual C++5.0プログラミング入門，アスキー， 1998
2) 横井与次郎:Visual C++5.0パワープログラミング ソフトバンク，1996
3) David J.kruglinski:Inside VisualC++ Version5，アスキー，1998
4) 林晴比古:新Visual C++ 5.0入門，ソフトバンク 1998
5) 石塚圭樹，横手靖彦:オブジェクト指向プログラミング，アスキー，1993