

3次元モデリングとアセンブリツールソフトウェアの開発

Development of a Tool Software for 3D Modeling and Assembling

高 三徳(いわき明星大)

Sande GAO

Iwaki Meisei University, 5-5-1 Iino, Chuonndai, Iwaki, Fukushima

Key Words: 3D Modelling, Assembling, Software developing, Visual C++, OpenGL

1. 研究の目的と内容

現在, 設計や開発では汎用 CG/CAD/CAE/CAM ソフトが使用されている. しかし, 機能などによって, その入出力データのフォーマットが異なり, ソフト間の直接データ転送ができない場合がある. また, ソフト自体が高額であり, 少しの機能を求めている場合に購入するというのは難しい. このような問題を解決するために, 本研究では Visual C++および OpenGL 使用して, 3次元モデリングソフトの開発と応用の研究を行い, 低価格のモデリングとアセンブリツールソフトの実現を目標とした.

主な研究内容は, ソフトの枠組みであるインターフェースの作成, 描画に関わるモデル作成と編集機能の作成, モデルのデータを保存・読み込み機能の作成であった. 具体的には, ①インターフェース: Visual C++を用いてツールバー, メニューバー, ダイアログボックス, モデルの体積とモーメントの表示機能を作成する. ②基本モデル: ボックス, 円筒, パイプ, 円錐, 円台, 穴あき円台, 球, トーラス, 多角柱を作成する. ③モデルの操作: 移動, 回転, スケーリング, 編集を行う. ④レンダリング: OpenGLを用いてワイヤフレームモデルまたはサーフェイスマodelでの表示を行う. ⑤物性値: モデルの質量, 回転モーメントの計算を行う. ⑥データの保存: 独自のtxt形式でモデルデータを保存する. ⑦データの読み込み: 保存したtxt形式のデータファイルを読み込み, モデルを描画する.

2. モデルのデータ構造と描画アルゴリズム

モデル (Object) のデータ構造を次のように定義した.

```
#define OBJTB struct objtb
OBJTB //モデルのリンクテーブル
{
    short type; //タイプ
    bool selflag; //0:選択していない; 1:選択している
    float tran[4][4]; //幾何変換マトリックス
    float color[3]; //カラー (3原色)
    float *dat; //形状の基本データ
    float v; //体積
    float mo[3]; //モーメント
    OBJTB *objpre; //pointer to the previous object
```

```
OBJTB *objnxt; //pointer to the next object
};
```

形状の基本データおよび幾何変換マトリックスから描画データを計算し, OpenGLの描画関数を利用してモデルを表示する. 即ち, 複数のポイントを作成し, ポイント同士を結ぶ事でモデルを構築し描画する. 例として, 円筒面の場合, 半径, 高さ, 位置座標および方向角から, 図1に示すように, 底面および上面の圆心および円上点P1 P2 P3およびP1' P2' P3'の座標を計算し, 三角形ポリゴンP1 P2 P3とP1' P2' P3'および四角形ポリゴンP2 P3 P3' P2'を回転コピーして描画すると, 円筒面が現れる. また, ポリゴンの3つの頂点座標から法線を計算し, OpenGLの照明関数によりモデル表面のシェーディング描画を行う.

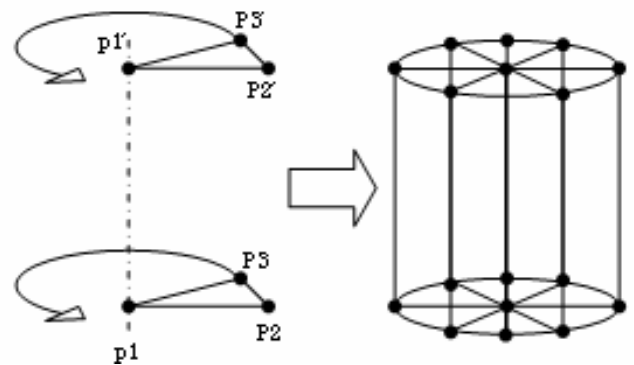


図1 円筒面の描画アルゴリズム

3. モデルの操作のアルゴリズム

①ピッキングアップ

モデルをマウスでクリックするとモデルが選択される機能を作成した. ポリゴンが三角形である場合, 図2に示すように, ポリゴンの画面への投影頂点 A, B, C を結んだ三角形の面積を S とし, 各頂点の投影点 A, B, C とマウスカーソルの点 mp を結んだ三角形の面積を S_1, S_2, S_3 としたとき, 次式の関係

$$s - s_1 - s_2 - s_3 \doteq 0 \quad (1)$$

になればマウスポイントがモデルの投影範囲内にあり, モデルが選択されたと判断される. ポリゴンが四角形である場合, 図3に示すように, 四角形を2つの三角形に分けて, 三角形の面積の関係が式 (2) になればモデルが選択されたと判断される.

$$s_1 + s_2 - s_3 - s_4 - s_5 - s_6 \doteq 0 \quad (2)$$

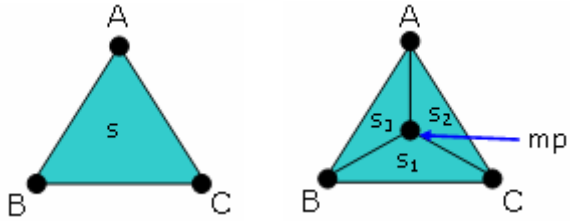


図2 三角形ポリゴンピッキングのアルゴリズム

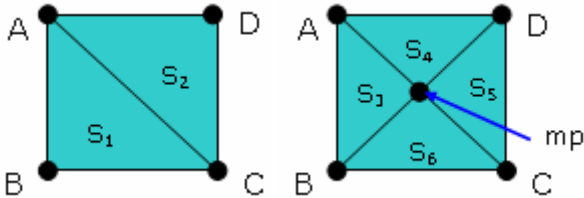


図3 四角形ポリゴンピッキングのアルゴリズム

②追加と削除

モデル追加の場合は、図4のようにTailモデルと追加モデルをNxtとPreのポイントで結ぶと同時に、Headモデルと追加モデルをNxtとPreのポイントで結ぶ。また追加したモデルがTailとなり最後のモデルとなるようにする。モデル削除の場合は、図5のように、まず、選択したモデルの前後のモデル同士のNxtとPreポイントを結ぶ。そして選択したモデルを削除する。

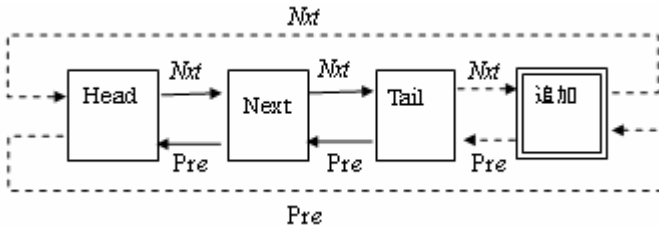


図4 モデル追加操作のアルゴリズム

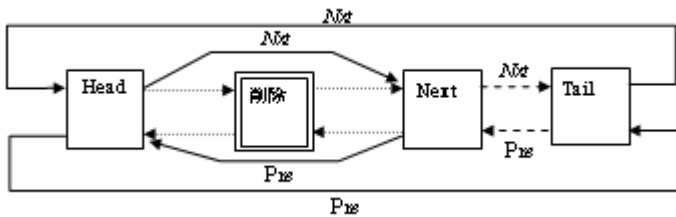


図5 モデル削除操作のアルゴリズム

③寸法修正とアセンブリ操作

形状の基本データを変更することによりモデルの寸法修正を行う。幾何変換マトリックスを修正することによりモデルの移動、回転、スケールなどのアセンブリの操作を行う。

4. モデリングおよびアセンブリの例

ボックス、円筒、パイプ、円錐、円台、穴あき円台、球、トラス、多角柱の9種類の基本モデルのデータ入力ボックスに形状の基本データおよび位置座標を入力して、作成したワイヤフレームモデルおよびサーフェスモデルを図6および7に

示す。また、アセンブリモデルの例を図8示す。

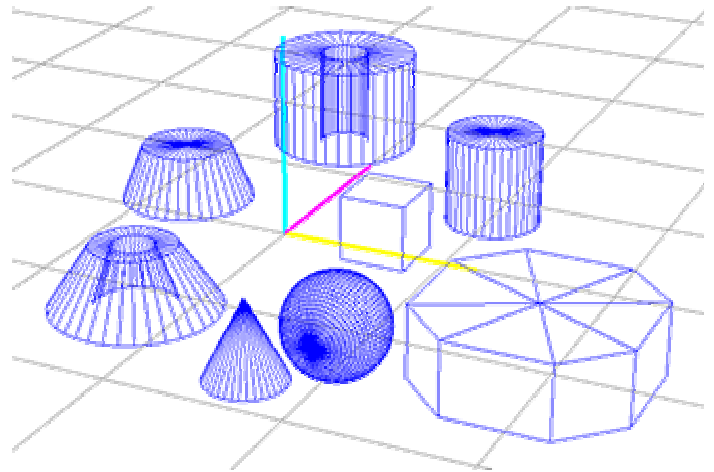


図6 基本モデルのワイヤフレーム表示

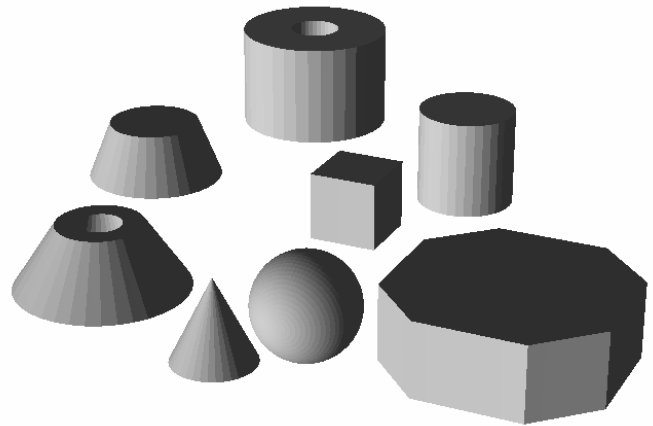
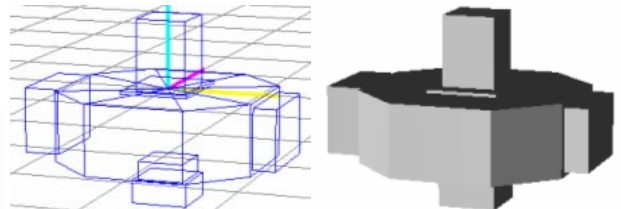


図7 基本モデルのサーフェスシェーディング表示



(a) ワイヤフレームモデル (b) サーフェスモデル

図8 アセンブリモデルの例

5. 結論と今後の課題

モデリング及びアセンブリツールソフトウェアの基礎的な部分を開発することが出来た。この開発をとおして Visual C++とOpenGL、ポインタ、リンクテーブルを使用したモデリングについて理解できた。リンクテーブルのデータ構造を使用することによりモデルのピッキング、追加、削除などの操作の便宜性およびコンピュータメモリー使用・管理の合理性が向上された。今後の課題として、ソフト動作の不安定さを解消すること、容易にモデル同士の配置を整えるなどのモデリング機能の向上、また、数値計算ソフトと連動するなど、モデリングソフトとしてだけでなく、解析の補助ソフトとしての機能を持たせ、実用化させることが期待される。