

3 次元形状測定と幾何形状処理

三谷純(mitani@cs.tsukuba.ac.jp)

1. はじめに

本実験課題は 3 次元形状を実際の物体から 3 次元スキャナで測定し、そのデータを加工処理する実験を行う。手順は大きく分けて次の 4 段階で行う。

- (1) 3 次元形状の形状測定
- (2) 3 次元形状表示用アプリケーションの作成
- (3) 3 次元形状データのデータ量削減
- (4) 幾何学的な形状データからの形状パラメータの抽出

2. 実験の流れ

実験の流れは次に示す通りである。

- (1) 実験室 3C315 室の 3 次元形状測定機 NextEngine を用いて 3 次元形状を計測する。
- (2) NextEngine に付属のソフトウェアを用いて 3 次元形状データを構築する。
- (3) 3 次元形状データを OBJ 形式で保存し、自身のアカウントのディスクに転送する。
- (4) OBJ 形式のデータを画面に表示するアプリケーションを作成する。
- (5) OBJ 形式で表現された 3 次元形状データ（一般の形状）に対してデータ量削減のプログラムを作成する。
- (6) 幾何学的な曲面の形状データから形状パラメータを抽出することを試みる。

本実験を通じて 3 次元形状表現手法と形状データ処理のアルゴリズムを学ぶことを目的とする。

【注意】

本実験は COINS の計算機環境で行うことを前提としているが、自身で自由に使える環境が別にあるのであれば、任意の環境で行って構わない。プログラムの開発は C,C++,Java など自分の得意な言語を用いてよい。

なお、毎回 3C113 室に置いてある出席表にサインを記入すること。

3. 実験のスケジュール

実験は全 15 週であり、以下のスケジュールを目安に行う。

第 1 週：ガイダンス

第 2 週：3 次元形状測定（測定したい対象物を持参する。可能であれば、データを保存する USB メモリも持参する）

第 3～6 週：3 次元形状表示アプリケーションの開発(中間レポート提出)

第 7～10 週：データ量削減プログラムの開発(中間レポート提出)

第 11～14 週：形状パラメータ抽出プログラムの開発

第 15 週：最終レポートとりまとめ

（詳しくは Web ページ参照のこと）

4. レポートの提出について

上記のスケジュールに従って実験を行い（多少前後しても構わない）、中間レポート 2 回と最終レポートをあわせて、計 3 回のレポート提出をする。

レポートには以下の内容を含めた PDF ファイルと、開発したプログラムコード、測定データ（初回のみ）を zip でまとめたものを指定された期日までにメールで提出すること。

- (1) 氏名、学籍番号
- (2) 各実験日に何を行ったかの報告
- (3) 開発したプログラムの内容説明（適宜スクリーンショットを含める）
- (4) 感想やコメント

5. 3 次元スキャナによる形状測定

実験室 3C315 室の 3 次元形状測定機(NextEngine)を用いて、対象物の形状データを取得する。測定機は PC から制御して行う。

基本的には TA が操作をサポートするが、Web ページ上のマニュアルを参照して基本操作を理解すること。マニュアル（閲覧のためのパスワードは実験初日に伝える）は接続されている PC でも参照できる。計測用コンピュータのパスワードは実験初日の説明の時に伝える。

計測機の原理

微弱なレーザー光線を対象物に走査しながら照射し、このスポット反射光を別の角度からビデオカメラで撮影する。スポット像の位置から三角測量の原理で奥行き量を含めた 3 次元的位置情報を取得する。このとき、レーザー光のスポット像ばかりでなく、物体形状、物

体上のテクスチャも同時に撮影する。したがって、後で立体像を生成したときに、テクスチャを載せることも可能である。ただし、本実験では形状のみを扱い、テクスチャは利用しないことにする。

測定対象物に関する注意

本実験で使う被測定対象物は2種類あり、一般的な形状と幾何学的な形状（球、円筒、円錐）である。一般的な形状は各自で準備すること。計測対象物として不適当なものを以下に示す。これらの特徴を持つ形状は避けること。

- ・大きさが極端に大きいもの(20cm 立方以上)、または極端に小さいもの(5cm 立方未満)
- ・色が極度に濃い(黒い)物体(反射するレーザ光を撮影して位置計測するので、色が濃いと反射光が撮影できなくなる)
- ・鏡面反射する物体(反射したレーザ光がカメラの方向に戻ってこないため)
- ・表面の位置を識別する模様やテクスチャ、しわが全くない状態(複数画像の位置合わせができないため)
- ・レーザ光が陰となって見えなくなるような、深くくぼみ、穴のある取手、入り組んだ形状等

その他の注意

- ・測定機は1台なので交代で使うこと。
- ・測定機を使う順番がまわって来ない間は自分のデータが得られないので、その間はサンプルデータを使用して、先のステップに進んでおく。

6. 3次元形状の表示

データの形式

3次元の形状を面の集合で表現するためのデータには、多面体の幾何情報を表す各頂点の3次元座標値(x,y,z)と、面がどのような頂点で構成されているかを表す位相情報が含まれる。一般に多面体をコンピュータ内で表現するためには、頂点、稜線、面を区別し、それらの接続関係を記述した位相情報を用いる。さらに、頂点の3次元座標値(幾何情報)も別途保持しており、相互に参照できるデータ構造をもつ。

図1の三角柱の例で、位相情報がどのように表現されているかを説明する。

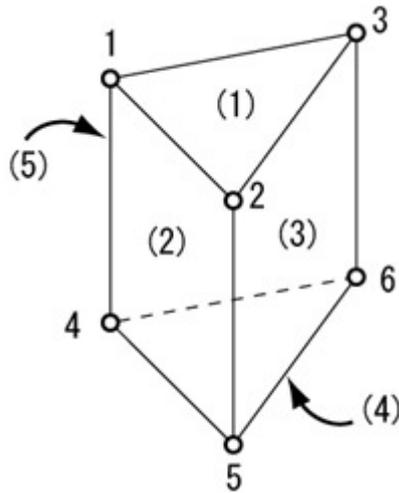


図1 三角柱

図1では頂点に1から6までの番号を振って識別している。面は()で囲まれた数字(1)～(5)で表わす。3次元形状を表現するファイル形式の一つとして、OBJ形式というものがある。OBJ形式では、頂点座標値が1番の頂点から順に記述されている。そして、稜線の記述は無く、各面がどの頂点で構成されているかを表す頂点番号のリスト(頂点ループリストと呼ぶことにする)を保持している。表1を参照すると、1番の面(上面の三角形)は頂点1,2,3の順で面が構成されていることがわかる。2番の面は四角形であるので、頂点は4つとなる。

面番号	面を構成する頂点ループ			
(1)	1	2	3	
(2)	1	4	5	2
(3)	2	5	6	3
(4)	4	6	5	
(5)	1	3	6	4

表1 三角柱のOBJ表現

実際のデータは次のように1行ごとに頂点または面の情報が記述されている。

```
v 0.000000 100.000000 100.000000
v 86.602547 100.000000 -49.999989
v -86.602539 100.000000 -50.000004
```

```
v 0.000000 -100.000000 100.000000
v 86.602547 -100.000000 -49.999989
v -86.602539 -100.000000 -50.000004
f 1 2 3
f 1 4 5 2
f 2 5 6 3
f 4 6 5
f 1 3 6 4
```

行頭の記号「v」と「f」で頂点情報と面情報を区別し、「v」で始まる行には頂点の x,y,z 座標が、「f」で始まる行には、面を構成する頂点の番号が空白区切りで並べられる。頂点の番号は、ファイル中に登場する頂点情報を上から順番に見てきた時の順番を表す 1 以上の整数である。

ここで重要なことは、頂点ループの順番である。頂点ループは面の表から見て、左回りに順番に頂点番号を並べていることである。これを右回りに記述すると面の向きが逆である（物体の内側から面を見ている）と解釈される。

アプリケーションの開発

上記のフォーマットである OBJ ファイルを読み込み、3次元形状を表示するアプリケーションを作成する。その際に、資料として提供されている OBJ ビューワーの実装（「CG 基礎」の授業で提供されているもの）と、ポリゴンモデルのデータ構造などに関する資料を参考に、次の機能を有するものを開発すること。

- ・指定した OBJ ファイルを読み込み、内部的には Halfedge 構造でデータを保持する
- ・接続する面が 1 つしかない稜線（立体表面の境界線に位置する稜線）を太い赤線で強調表示する。

7. 3次元形状データ処理：データ量削減

ネットワーク上で 3次元形状データを扱うときは、記憶容量や転送スピードを考慮すると、可能な限り少ないデータ量が望ましい。そこで、本実験ではデータ量の削減をメインテーマとしてプログラムを作成することを課題とする（以下のアルゴリズムを開発する段階では、大量の実験データでは処理時間がかかりデバッグの効率が悪くなるため、三角柱などの簡単なデータ構造を手作業で作成して頂点座標値、および頂点ループの表を用いると便利な場合がある）。

面の削減

データ量を減らす方針として、面の数を減らすことを行う。面を削除するには全体の形状を大きく変化させないように気をつける必要がある。

面の削減方針

面を削減するために、図 2 のように隣接する面の共通稜線 AC に注目して、その長さを無限小に縮めるという操作を行う。これによって頂点 C、稜線 AC、2 面($\triangle ABC, \triangle ACD$)の削減を行うことができる。このように稜線を削除することで 2 つの面を同時に削減する方法を **Edge collapse** という。OBJ 形式のデータの場合、データの中に稜線の情報は明記されていないが、面を構成する頂点のループから稜線を決定することができ、また同一の稜線を共有する面が 2 つあれば、その 2 つの面は互いに隣接関係にあることがわかる。

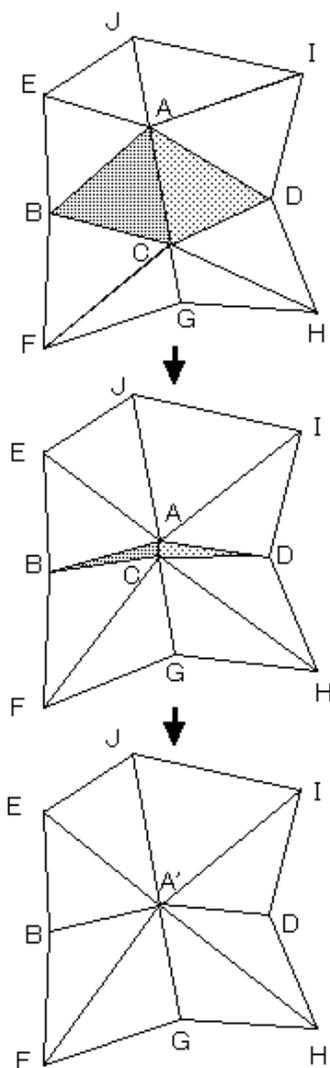


図 2 稜,面,頂点の削減

削除する稜線を選ぶ

上記の Edge collapse を繰り返すことで面の数を減らすことができる。しかし、ランダムに削除するのではなく、元の形がなるべく維持されるような戦略を考える必要がある。例えば、削減される面に隣接した近傍の面が、削減される面とほとんど同じ方向を向いていればそのうちの一つの面を消去して残りの面の面積を大きくして、おきかえても全体の形状の変化は少ないだろうと考える。

図2の例では $\triangle ABC$ 、 $\triangle AEB$ 、 $\triangle BFC$ の3つの面がほぼ同じ向きで、一方、 $\triangle ACD$ 、 $\triangle ADI$ 、 $\triangle DCH$ の3つの面が互いにほぼ同じ向きであれば(このとき、 $\triangle ABC$ と $\triangle ACD$ は必ずしも同じ向きである必要はない) $\triangle ABC$ 、 $\triangle ACD$ を図2の一番下の図のように削減しても、全体の形状の変化は小さいはずである。また、稜線の長さが短いものを選ぶ、または面積の小さい面を消すようにを選ぶ、というような戦略も有効である。どの稜線を優先的に削除すると効果的であるかは、複数のケースで実験を行い比較すること。

この Edge collapse のステップを他の部分にも適用していくことで、全体の面数、頂点数を削減してゆく。

特殊なケース

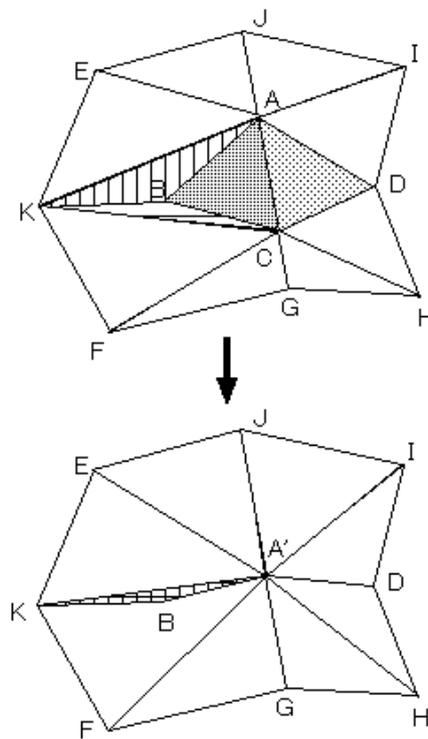


図3 稜・面消去時に余分な2面 ($\triangle A'KB$, $\triangle A'BK$)が残る例

面の接続関係が特殊な図 3 のような場合は、稜線 AC の長さを 0 にして 2 面 $\triangle ABC$ 、 $\triangle ACD$ を消去しても、 $\triangle AKB$ 、 $\triangle BKC$ は消去されずに、重なって $\triangle A'KB$ 、および、 $\triangle A'BK$ となって存在することになる。一般的に、1 つの稜線が属する面は 2 つ(境界部分では 1 つ) であるが、この場合は 4 つの面に属することになってしまう。このままだと後の処理で問題を起こすことがあるため、物体からはみ出したこれらの 2 つの面も消去しておく必要がある(または、そもそもこのような特殊なケースでは **Edge collapse** を行わない、という方法もある)。

各面の法線方向

複数の面が同じ向きになっているかどうかを調べるには、面の単位法線ベクトルを求め、それらの内積が 1 に近くなれば、ほぼ同じ向きになっていると見なすことができる。一般に 3 次元空間内にある 3 個の頂点 $P_i(x_i, y_i, z_i) (i=0,1,2)$ からなる三角形の法線方向を $n(n_x, n_y, n_z)$ とすると、 n は次式で表される。

$$\begin{aligned} n_x &= y_0 \cdot z_1 - y_1 \cdot z_0 + y_1 \cdot z_2 - y_2 \cdot z_1 + y_2 \cdot z_0 - y_0 \cdot z_2 \\ n_y &= z_0 \cdot x_1 - z_1 \cdot x_0 + z_1 \cdot x_2 - z_2 \cdot x_1 + z_2 \cdot x_0 - z_0 \cdot x_2 \\ n_z &= x_0 \cdot y_1 - x_1 \cdot y_0 + x_1 \cdot y_2 - x_2 \cdot y_1 + x_2 \cdot y_0 - x_0 \cdot y_2 \end{aligned}$$

(注意：この法線ベクトルは単位ベクトルではないので、改めて単位ベクトルにする必要がある。)

隣接する面の平面度の計算

各面の単位法線方向ベクトルから、面の相互の傾き具合を調べる。相互の傾き角が少ない(同一平面内に近い)と各面の単位法線ベクトルは近い方向にあるので、内積が 1 に近くなる。この内積を平面度と定義する。

2 つのベクトル $m(m_x, m_y, m_z)$ 、 $n(n_x, n_y, n_z)$ の内積は

$$m_x \cdot n_x + m_y \cdot n_y + m_z \cdot n_z$$

で求まる。

面の削減

平面度がある閾値より大きいとほぼ同一平面内にあると見なす。従って、注目する面側の片側の 3 つの面($\triangle ABC$ 、 $\triangle EBA$ 、 $\triangle BFC$)がほぼ同一平面内にあり、他の片側の 3 つの面($\triangle ACD$ 、 $\triangle ADI$ 、 $\triangle DCH$)もほぼ同一平面内にあることがわかれば、面対($\triangle ABC$ 、 $\triangle ACD$)を消去しても形状の変化は少ないはずであるから消去する。

面を消去する手続きは次の通りである。

稜線 AC の長さを 0 にする操作であるから、点 A、点 C のどちらか一方を残して、他方を

消去することにする。一方を消去後、残された頂点は元の 2 頂点 A、C の中点とするのが適当である。

仮に点 C を消去とした場合、面を定義する頂点ループのリストの中で、頂点 C を参照している部分を探して、全て頂点 A を参照するように変更する。

削減の繰り返し

隣接する面対を選択して消去する操作を繰り返すことで、面と頂点のデータ量を削減する。このとき、形状全体を考慮して選択する稜線を探索する場所が均等に散らばるように工夫が必要である。消去する面対が局所的に偏ると全体としての形状がひずむ場合がある。

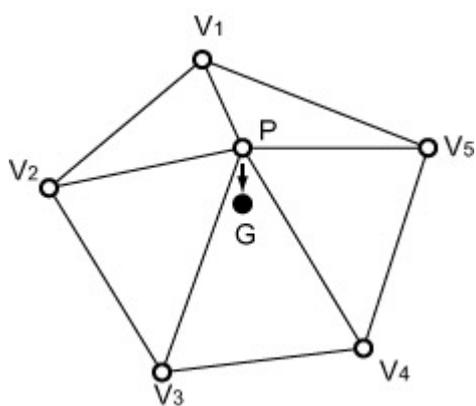
前処理としての形状平滑化

計測機で取得されるデータは、一般的にノイズを多く含み、滑らかでない（激しい凹凸を含む）形が得られることが多い。そこで、データ量削減の前処理として、形状を滑らかにする「平滑化处理」を行うことを考える。

この平滑化は、面の削減などは行わず（位相変化を伴わない）、単に頂点を移動させることだけで実現する「ラプラシアンスムージング」の手法を用いることとする。このアルゴリズムの次のように極めて単純なものである。

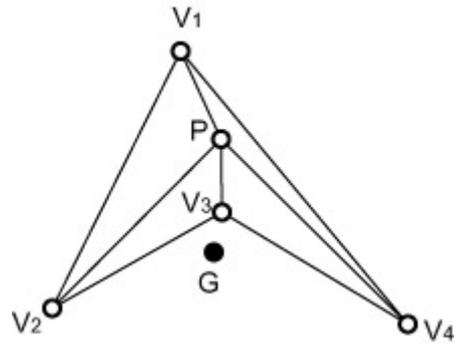
「各頂点 P を、P の 1 近傍頂点の重心に移動する」

これは下図のように、点 P に着目した場合、点 P から 1 つの稜線を介して隣接する位置にある点 V(1~5)の重心 G に点 P を移動させる操作を、全ての頂点に対して行う、ということである。



平滑化

ラプラシアンスムージングは極めて単純なアルゴリズムであるため、次のようなケースでは面が反転して好ましくない結果になることがある。さらに余力がある場合は、この問題への対処方法も考慮すること。



8. 幾何学的な形状の形状パラメータの抽出

球、円柱、円錐のような2次曲面が含まれる幾何学的な形状は、半径や、高さ、頂角などのパラメータがわかれば、形状データは計算で容易に求められる。そこで、逆に測定した形状データから球か、円柱か、円錐かを区別し、さらにこれらのパラメータを抽出することに挑戦するのが、ここでの課題である。