

# 勉強会レジメ->画像処理

B4 魯陽

## 1. 画像処理

画像処理の目的：

- 1.1: 画像のデジタル化と符号化（圧縮も）
  - 1.2: 画像の強調と復元（ぼけ、雑音）画像認識
  - 1.3: 画像の分割と記述（特徴づけと特徴による分類）
- 入力と出力による分類：2D と 3D。

## 2.画像の形

2.1 数学的な定義： $z = f(x, y)$   $z$  は明度。

2.2 配列 array[] で表現する。（実際処理する対象）

2.3 チェインとしての曲線。

画像が数の少ない領域からでき、しかも濃度レベルも比較的小さいとき、大きな配列で表現するのは不経済である。領域の境界と濃度を規定できれば画像は完全に決定される。計算機で量子化された画像は、境界や曲線は短い線分からなる多角形で近似している。実際の曲線と違って、望むだけの精度によって、画像を再現しています。チェイン符号は正方形単位を扱っている。チェイン符号は傾きの系列で表現している。傾きは 4 5 の倍数 (0,45,90,135,180,225,270,315,) であるので、0,1,2,3,4,5,6,7 で表現できる。

例えば：10112 22222

概念：

ピクセル：フレームメモリに格納されるデジタル画像の 1 画素に対応し、データとして輝度や色を表すデジタル値を持つ。

## 3.数学的準備

3.1 フーリエ変換(1D):

$$F(k) = \mathcal{F}[f(x)] = \int_{-\infty}^{\infty} f(x)e^{-ikx} dx$$

は  $f(x)$  のフーリエ変換という。また

$$f(x) = \mathcal{F}^{-1}[F(k)] = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(k)e^{ikx} dk$$

は逆フーリエ変換という。

3.2 フーリエ変換(2D):

フーリエ変換:

$$F(k_1, k_2) = \mathcal{F}[f(x_1, x_2)] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x_1, x_2) e^{-i(k_1 x_1 + k_2 x_2)} dx_1 dx_2$$

逆変換：

$$f(x_1, x_2) = \mathcal{F}^{-1}[F(k_1, k_2)] = \frac{1}{(2\pi)^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(k_1, k_2) e^{i(k_1 x_1 + k_2 x_2)} dk_1 dk_2$$

で定義されている。

フーリエ変換を使うことによって高周波数と低周波数の画像を区別することができる。

#### 4.画像出力処理

- ディザ法 (シェーディングには二値画像を用いる)
 

二値画像から濃淡画像を再生成するには、人間の視覚システムの利点を利用しなければならない。われわれの眼は局所な輝度を平均化し、滑らかにする働きがある。このような処理を行う技術をディザ法 (**dither method**) と呼ぶ。
- 色変換 (加色、減色、色の不変性)
 

カラー画像を印刷する際には、表色系が必要となる。色の知覚は人間の内部における心理的現象であるため、色を物理的な測度に対応付けるのは非常に難しい。よって、状況や目的、出力デバイスの種類に応じた表色系に変換しなければならない。
- ステレオスコープ (2枚の画像を左右それぞれに表示して見せる)
 

また、3次元空間を直接人間に知覚するには、左右の目に、互いに視点が少しだけ違う視差画像をそれぞれの目に見せてやれば良い。このような技術はステレオスコープ (stereo scope) と呼ばれる。

描画アルゴリズム：

直線を描く Bresenham (ブレセンハム) のアルゴリズム：

Bresenham のアルゴリズムにおいては傾きの算出はせずに誤差値(error)をフラグとして x 値 y 値をそれぞれ増加させていく方法を取る。X 値を 1 を増やして、y の値を計算し、順番に画像を描く方法である。

Bresenham のアルゴリズムの価値は” 加算・減算のみで実数演算は無し” という部分が、現在のコンピューターを使った場合に既存の関数を利用するよりどれだけ処理を高速化できるか？という部分に注目される。

#### 5.画像入力処理

入力画像は二次元からの場合：

2次元平面の離散化、つまりピクセルに切り分けることを**サンプリング (sampling)**と呼び、切り分けのキメの細かさを**解像度 (resolution)**と呼ぶ。また、濃淡量を(例えば0から255までの)離散値にマッピングすることを**量子化 (quantization)**と呼ぶ。サンプリングも量子化もデジタル画像の質に影響する。

入力画像は三次元からの場合：

投影モデル：

[**並行投影モデル (parallell projection model)**]

並行投影モデルでは、太陽の光のように光源が無遠慮に設定されており、光は互いに並行に放射していると仮定する(図8)。光の方向に関する次元が省略可能なため、非常にシンプルな投影モデルといえる。この投影モデルの問題は、失われた次元の情報が一切残っていないという点である。例えば、画像中のオブジェクトのサイズはオブジェクトがカメラに近かろうと遠かろうと同じになる。

[**透視投影モデル (perspective projection model)**]

カメラは原理的にはピンホールカメラでモデル化できるのでカメラにより撮影された画像は、透視投影モデルにより獲得された画像となる。このモデルによる画像は、カメラから対象物までの距離についての情報を部分的に持っているため、画像中のオブジェクトが遠いか近いかを直観的に知ることができる。このモデルを説明するには、ピンホールのカメラモデルを考えれば良い。ピンホールカメラモデルでは、グローバル座標系での3次元座標  $(X, Y, Z)$  の点はレンズを通して画像平面上の1点  $(x, y)$  に投影される。座標軸は図9(a)のように設定する。相似三角形の特性により、スクリーン座標(画像上の座標)  $(x, y)$  は以下の式で導かれる。

$$\begin{aligned}x &= \frac{Xf}{Z} \\y &= \frac{Yf}{Z}\end{aligned}$$

ここで、 $f$  はレンズの焦点距離であり、これが画像平面の中心からレンズ中心までの距離となる。式から分かるように、スクリーン座標系上の座標  $(x, y)$  は座標  $(X, Y)$  だけでなく  $Z$  の値とレンズの焦点距離  $f$  にも依存する。これが、透視投影モデルによる座標  $(x, y)$  が距離  $Z$  に関する情報を持つ理由である。しかし数学的にはスクリーン座標  $(x, y)$  からだけでは  $Z$  の値を算出できないことは理解しておいて欲しい。

反射モデル：

光の反射要素として、一般的に**拡散反射 (diffuse reflection)**と**鏡面反射 (specular reflection)**の2種類がよく用いられる。

拡散反射とは、光が物体表面の微細構造の中に入り込み、ランダムな反射を繰り返したあと再び表面から外に向かっていく光の様子をモデル化したものである。今、入射光が方向

$\vec{e}$  で強さ  $I$  とし、物体の法線ベクトルを  $\vec{n}$  とする。ただしベクトルは全て単位ベクトルとする。

Lambertの余弦法則により、視点方向  $\vec{e}$  への反射光の強さ  $I_d$  は

$$I_d = -(\vec{i} \cdot \vec{n}) k_d I_l$$

である。

鏡面反射は 金属表面を観察したときに見えるハイライト部分の様子をモデル化したものである。

$$I_s = (\vec{s} \cdot \vec{n})^m k_s I_l \vec{s}' = \frac{\vec{e} - \vec{i}}{|\vec{e} - \vec{i}|}$$

現実世界で問題になるのは入射光がどこから来るかである。モデルで定義する明示的な平行光源や点光源は当然入射光の有力成分であるが、その他にもそれらの光源からの光が他の物体で反射した後で別の物体の表面に到達していることが考えられる。一般にこれを相互反射という。これを厳密に計算する様々な方法も提案されているが、これらはどれも膨大な計算コストを必要とする。

そこで、この相互反射を非常に粗い強引な近似で表現するのが環境光・環境反射である。

環境光の強さ  $I_e$  を次式で表す。

$$I_e = k_a I_a \quad 0 < k < 1$$

## 6.三次元グラフィックス

ポリゴン：三次元物体を描くための最小の単位であり、一般は数多くの小さい多角形で、三次元物体を近似する。この多角形は、ポリゴンという。

ラスタスキャン：点を高速に移動させる走査線 (raster) を用いて1枚の画像を表示する方式のディスプレイ。画面は走査線の並びでフレームとして表現される。スキャンしながら各点の色や輝度の情報をコンピュータからディスプレイに送って表示する。

隠面消去のスキャンライン法：オブジェクトを構成する各ポリゴンをスクリーン上の各スキャンライン (横方向のピクセルの集まり) ごとに分割し、これをセグメントと呼ぶ。同じスキャンラインを共有するセグメント同士の奥行き値を比較して、手前にあるほうだけを描画する。これをスクリーン上の全スキャンラインについて行なうことにより、陰面消去が行なえる。

隠面消去のZバッファ法：奥行き情報 (ピクセルのZ値のこと。X-Y-Z座標) を記録するためのバッファを使うことからこう呼ばれている。実際のハードウェアで実装している。膨大なメモリが必要である。

ボリューム・レンダリング手法: ボリューム・レンダリング手法は対象となる立体要素(ボクセル)の集合体と考えます。それぞれのボクセルは3次元座標上の点からサンプリングされたものです。またボクセルはもとの対象物について1つもしくは複数の観測値あるいは計算値(例えば、傾向強度、密度、流速、電荷、温度など)を保持しています。

並列処理: 3次元フレームメモリに3次元のボクセルデータを保持する。3次元フレームメモリを部分空間のボクセルデータを分散させ、並列に自己の部分空間に対する視点から見た2次元画像を生成し、それらの結果をマージして隠れ面処理を実行することにより、最終的な2次元画像を得る。

## 7.圧縮(情報理論)

一般データを圧縮する方法として、データ中に現れるパターンの発生確率を求め、多く現れるものには短い符号、あまり現れないものには長い符号を割り当てて符号化することで、圧縮する方法がある。Huffman符号化法は、理論上最短の符号化を実現できるものとして知られているが、あらかじめ元データをすべて調べて出現確率を求めて対応表を作る必要があり、実用上効率がよいとはいえない。

参考文献:

<http://www.mm.media.kyoto-u.ac.jp/education/DIP/>

<http://www.mm.media.kyoto-u.ac.jp/members/iiyama/lecture/2003-kyoto-le4cg/le4cg-html/le4cg-html.html>

デジタル画像処理: AZRIEL ROSENFELD

AVINASH C. KAK

長尾真 監訳