

# CG 基礎数学



# 第1章 デジタルカメラモデル

## 1.1 座標と座標系

座標系 (coordinate system)

直交座標系 (rectangular coordinate) またはデカルト座標系 (Cartesian coordinate)

座標系で指定される 2 つの値の組 = 座標 (coordinates)

デカルト (Rene Descartes)(1596-1650)

Cartesian : Descartes の形容詞形

方法序説 (1637 年)

「われ思う、ゆえにわれ在り」

コギト・エルゴ・スム (Cogito Ergo Sum)

## 1.2 幾何学的変換

### 1.2.1 平行移動

点  $(x, y)$  を、 $x$  軸方向に  $t_x$ 、 $y$  軸方向に  $t_y$  だけ平行移動した点を  $(x', y')$  とすると、

$$x' = x + t_x$$

$$y' = y + t_y$$

である。

### 1.2.2 拡大・縮小

図形を  $x$  軸方向を  $s_x$  倍に、 $y$  軸方向を  $s_y$  倍に拡大すると、元の点  $(x, y)$  は、変換後に、

$$x' = s_x x$$

$$y' = s_y y$$

となる。

### 1.2.3 回転

角の測り方

度数法    1 回転 =  $360^\circ$

弧度法    1 回転 =  $2\pi$  ラジアン

#### 回転

点  $P(x, y)$  を、原点  $O$  を中心に角  $\theta$  回転した点を  $P'(x', y')$  とする。更に、 $\overline{PO} = \overline{OP'} = r$  とすると、

$$x = r \cos \phi$$

$$y = r \sin \phi$$

また、

$$\overline{OQ} = r \cos \theta, \quad \overline{P'Q} = r \sin \theta$$

なので、

$$\begin{aligned} x' &= \overline{RS} \\ &= \overline{QS} - \overline{QR} \\ &= \overline{OQ} \cos \phi - \overline{P'Q} \sin \phi \\ &= r \cos \theta \cos \phi - r \sin \theta \sin \phi \\ &= x \cos \theta - y \sin \theta \end{aligned}$$

また、

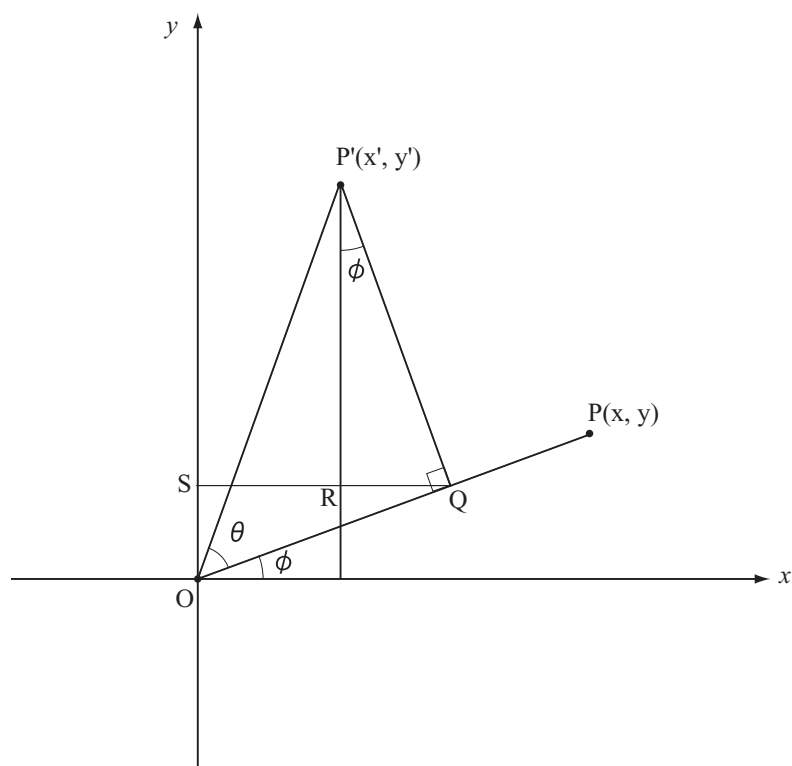
$$\begin{aligned} y' &= \overline{P'R} + \overline{SO} \\ &= \overline{P'Q} \cos \phi + \overline{OQ} \sin \phi \\ &= r \sin \theta \cos \phi + r \cos \theta \sin \phi \\ &= x \sin \theta + y \cos \theta \end{aligned}$$

すなわち、

$$x' = x \cos \theta - y \sin \theta$$

$$y' = x \sin \theta + y \cos \theta$$

を得る。



また、

$$x' = r \cos(\theta + \phi), \quad y' = r \sin(\theta + \phi)$$

とも書けることから、三角関数の加法定理

$$\cos(\theta + \phi) = \cos \theta \cos \phi - \sin \theta \sin \phi$$

$$\sin(\theta + \phi) = \sin \theta \cos \phi + \cos \theta \sin \phi$$

が導かれる。

#### 1.2.4 鏡映

$x$  軸に関する鏡映  $x' = x, \quad y' = -y$

$y$  軸に関する鏡映  $x' = -x, \quad y' = y$

$y = x$  に関する鏡映  $x' = y, \quad y' = x$

### 1.2.5 スキュー (剪断)

$x$  軸方向のスキュー  $x' = x + ay, \quad y' = y$

$y$  軸方向のスキュー  $x' = x, \quad y' = y + bx$

### 1.2.6 合成変換

点  $P(x, y)$  が 点  $Q(x_Q, y_Q)$  を中心として  $\theta$  回転する場合

- (1) 回転の中心  $(x_Q, y_Q)$  が原点に来るように、点  $(x, y)$  を平行移動する。すなわち、

$$\begin{aligned} x_t &= x - x_Q \\ y_t &= y - y_Q \end{aligned}$$

と置く。

- (2)  $(x', y')$  を原点を中心に  $\theta$  回転を行う。

$$\begin{aligned} x_r &= x_t \cos \theta - y_t \sin \theta \\ &= (x - x_Q) \cos \theta - (y - y_Q) \sin \theta \\ y_r &= x_t \sin \theta + y_t \cos \theta \\ &= (x - x_Q) \sin \theta + (y - y_Q) \cos \theta \end{aligned}$$

- (3) 原点が、 $(x_Q, y_Q)$  に戻るように平行移動をする。

$$\begin{aligned} x' &= x_r + x_Q \\ &= (x - x_Q) \cos \theta - (y - y_Q) \sin \theta + x_Q \\ y' &= y_r + y_Q \\ &= (x - x_Q) \sin \theta + (y - y_Q) \cos \theta + y_Q \end{aligned}$$

## 1.3 投影変換

投影とは、3次元図形をディスプレイモニタ画面など2次元画面に表示することである。代表的な投影法として、透視投影と平行投影がある。

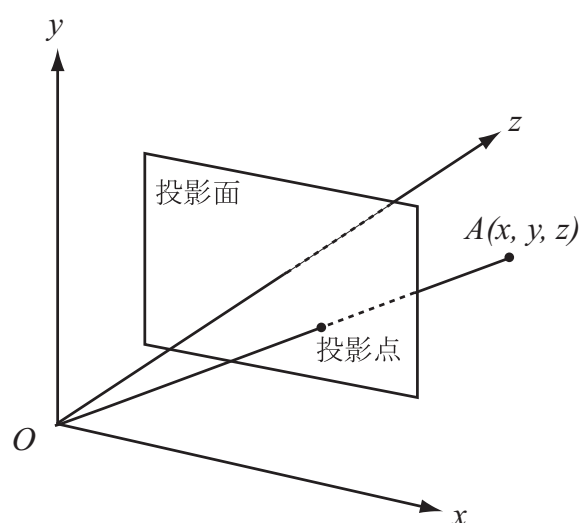
### 1.3.1 投影の原理

#### 透視投影 (perspective projection)

視点 (目やカメラの位置)

投影面 (projection plane)

3次元図形の各点から視点に向かって線 (投射線) を引く。投影面上に投射線との交点の集まりとしての投影図が形作られる。



1. 視点を原点に取る。 $(x, y, z) = (0, 0, 0)$
2. 投影面を  $z = 1$  とする。
3. 物体の点 A を  $(x, y, z)$  とする。

#### 点 A と投影面における座標 $(x', y', 1)$ の関係

原点 O と点 A を結ぶ直線の方程式は、

$$\ell(t) = (tx, ty, tz), \quad -\infty < t < +\infty$$

で与えられる。ここ直線が投影面をよぎる点  $t$  の値は、

$$(x', y', 1) = (tx, ty, tz)$$

を満たす。 $z$ 座標の値を比較して、すなわち、 $tz = 1$  より  $t = 1/z$  を得る。これから、

$$x' = \frac{x}{z}, \quad y' = \frac{y}{z}$$

を得る。

または、比例関係

$$x' : y' * 1 = x : y : z$$

から、同様の結果を得る。

我々の物の見方は、透視投影と考えられる。写真に近い図、映画やゲームなどに用いられる。平行線 (例えば、鉄道のレールなど) が平行ではなく、歪んで表示されるために、物の形を正確に把握するのには適さない。

透視投影 — **遠近法**とも言う。

平行な直線群が1点に集中する。この点を**消点** (消失点) と言う。

消点の個数による分類

**1 点透視**

**2 点透視**

**3 点透視**

## 平行投影 (parallel projection)

3次元の各点から投射線を $z$ 軸に平行に投影面に下ろし、投影図を形作る。

1. 視点は $z$ 軸のマイナス無限遠方  $(x, y, z) = (0, 0, -\infty)$
2. 投影面を $z = k$  ( $k$ は任意) とする。
3. 物体の点 $A$ の座標を $(x, y, z)$  とする。

点 $A$ と投影面における座標 $(x', y', k)$ との関係は、

$$x' = x, \quad y' = y$$

である。物の形を正確に表現するために用いられる。設計図、グラフ描画などに用いられる。

ここまで扱ってきた投影法—投影方向が投影面に垂直な平行投影。これを**直投影**という。投影方向が投影面に垂直でない場合を**斜投影**と言う。斜投影は物体の正面と側面の両方を表示したい場合に用いられる。

**三面図 (第三角法)** 正面図、側面図、平面図

**アイソメ図 (等測投影図)** 等速投影 (isometric projection) 立方体の3軸の長さが等しくなるような投影法。



### ビューボリュームとクリッピング

- 実際の投影では、投影面が有限な長方形である。これをウィンドウ (window) と言う。
- 視点からウィンドウをカバーする角度を 画角 (視野角) と言う。
- 奥行きに関しては、前後に有限の範囲を考え、その範囲内の物だけを投影する。  
視点に近いものを**前方クリッピング面** (front clipping plane)  
視点から遠いものを**後方クリッピング面** (back clipping plane)  
と言う。clipping ; はさみで切ること、刈り込み、切り取ったもの
- 図形が描かれる範囲は、四角錐台 (六面体) となる。これを**ビューボリューム** (view volume) と呼ぶ。  
volume : 体積、量、かさ ; 容量、容積
- ビューボリュームからはみ出す図形の削除処理を**クリッピング**と言う。

## 1.4 いろいろな座標系と変換

CGにおいては、被写体をさまざまな方向から表示することができる。このような場合、さまざまな座標系を処理に応じて使い分け、その間の変換を行う。

**モデリング座標系・物体座標系** 物体の形状を定義しやすいように設けられた座標系

**ワールド座標系** CGの世界全体を扱う際の座標系

**カメラ座標系** CGでワールド座標系に配置された図形を2次元の画像で表示する場合のカメラ固有の座標系。

ワールド座標系からカメラ座標系への変換 — **視野変換**

**投影変換** 投影変換では、3次元図形を2次元図形として表す。2つの図形が重なった場合には、重なり部分の遠方図形を消去する。これを**陰面消去**という。

**ウィンドウとビューポート**

ウィンドウ — 3次元CGにおいて、投影面上で実際に表示される部分

ビューポート — ディスプレイ画面に設けられたデバイス座標系上で画像が表示される部分

ウィンドウの範囲がちょうどビューポートに収まるように行う変換を**ビューポート変換**と言う。

### 投影図の生成と解釈

図形が定義され、変換を受けて、最終的に表示されるまでの一連の過程を、**ビューイングパイプライン**と呼ぶ。

- (1) モデリング座標系からワールド座標系へのモデリング変換
- (2) ワールド座標系からカメラ座標系への視野変換
- (3) カメラ座標系から投影変換、ビューポート変換を行う。

## 1.5 ビジュアル情報処理の光学的モデル

### 1.5.1 光と色

#### 人間の視覚

外界からの光は**角膜**を通り、さらに**水晶体**を通して眼球内に入り、**網膜**上にその像を結ぶ。網膜には、2種類の視細胞がある。長波長 (L)・中波長 (M)・短波長 (S) に最も反応する3つの錐体と弱い光に感受性が強い桿体がある。

白色光を7色に分類したのは、ニュートンである。光学 (1704 年)

c.f. ゲーテ「色彩論」

虹の7色は、波長の長い順に、赤、橙、黄、緑、青、藍、紫

#### 光の3原色 (RGB)

**L 錐体** 赤 (Red) に対する感受性

**M 錐体** 緑 (Green) に対する感受性

**S 錐体** 青 (Blue) に対する感受性

3原色を混ぜ合わせて別の色を生成することを**混色**という。3原色の光が混ざると白色になる。このような色の生成を**加法混色**と言う。R,G,B の3色の光で色をきじゅつする体系を**RGB カラーモデル**と呼ぶ。

一方、絵具は、色を混ぜるごとに暗くなる。これを**減法混色**と言う。印刷などで用いられる3原色としては、C(シアン)、M(マゼンタ)、Y(イエロー) の**CMY**表示系が用いられる。

#### 知覚に基づく表色系

米国の画家アルバート・マンセル (1858-1918) によって作り出された。

#### マンセル表色系

色を**色相 (Hue)**、**明度 (Intensity)**、**彩度 (Saturation)** の3つの属性で分類。

3つの属性を立体的に表現したもの — **マンセルの色立体**

マンセル表色系を、定量的に扱いやすくしたものが、**HSI カラーモデル**である。

## ビジュアル情報処理の光学的モデル

物体が見えるのは、物体に光が当たってそれを反射するためである。反射した光が我々の目に入って、我々は物体の存在を認識するのである。CGでは、これをモデル化して行う。CGで陰影を付ける処理を**シェーディング (shading)**と言う。  
shade 陰： shading：陰にすること。陰影画法

物体を照射する光

**直射光** 光源から物体に直接当たる光

【例】太陽光，ライト 太陽は遠く離れているために、光源からの光の方向も強さも一定になる。すなわち、**平行光線**とみなせる。

**間接光** 光源から出た光が他の物体で反射してから物体に当たったもの。

## 光の反射

入射角と反射角は等しい。(スネルの法則)

ところが、ざらざらした物体では、ミクロの反射面がさまざまな方向を向いているので、反射光方向も様々である。このような面を**拡散反射面**、その光を**拡散反射光**と言う。

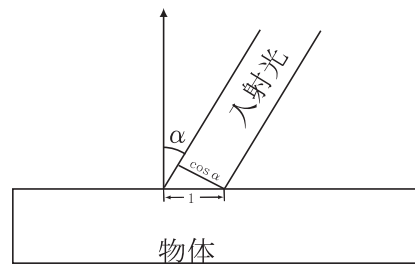
拡散反射：diffuse reflection

## 反射光の強さ

入射光の強さを  $I_i$ 、面の垂直方向ベクトル (**法線ベクトル**) と光源方向のなす角を  $\alpha$  とすると、物体の単位面積当たりの入射光の強さは、 $I_i \cos \alpha$  となる。拡散反射率を  $k_d$  とすると、反射光の強さ  $I$  は、

$$I = k_d I_i \cos \alpha$$

となる。 $k_d$  が大きいと物体は明るくなり、小さいと暗くなる。



## 1.6 デジタル画像

### 1.6.1 画像の標本化と量子化

デジタル画像 - 実際の映像を等間隔に並んだ格子状に並んだ個々の投影単位素子でのみ記録される。

**標本化 (sampling)** 連続的に変化する画像データを一定の間隔において有限個の点 (標本点) で測定すること。

**標本値** 標本化によって得られた値。

**量子化** 色の濃淡や明るさなど、連続的に変化する量を離散的な値に置き換えること。

### 1.6.2 デジタル画像

デジタル画像の個々の標本点を、**画素 (ピクセル)** と言う。

ピクセル (pixel) - picture cell または picture element の略

アナログ画像に対して標本化と量子化を行う **アナログ/デジタル (A/D) 変換**  
デジタル信号をアナログ信号に変換する **デジタル/アナログ (D/A) 変換**

### 1.6.3 階調と解像度

明るさの程度 (明度) を 1 バイト=8 ビットで表す (8 ビット量子化) の場合、明るさは、 $2^8 = 256$  段階に分けて表される。RGB カラーの場合、各色の濃淡を 1 バイトで表現すると、3 色で 3 バイトとなる。

コンピュータによる数の表現は、2進法 (binary system) である。  
8桁までの数表現 00000000~ 11111111 2進数 11111111 を10進数で表すと、

$$\begin{aligned} &11111111 \\ &= 1 \times 2^7 + 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\ &= 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 \\ &= 255 \end{aligned}$$

**階調 (gradation)** 色の濃淡の変化のこと、または濃淡変化の滑らかさのこと、各色を何段階に分けたかを数字で表す。たとえば、8ビット量子化による色の表現を256階調と言う。

**解像度** どの程度まで細かく画像を再現できるかを表す尺度。標本化間隔が狭くなると、解像度は上がる。標本化間隔は、単位長さ当りの標本点数で表す。

【例】dpi(dots per inch) は、1インチ (=2.54 cm) 当りの標本点数。  
5インチ×5インチの写真を150dpiで入力した場合の画素数

$$(150 \times 5) \times (150 \times 5) = 750 \text{ 画素} \times 750 \text{ 画素} = 562,500 \text{ 画素}$$

画像全体のデータ量

大きさが640×480画素のモノクロ画像を8ビット(256階調)のグレースケールで表すと、データ量は、

$$640 \times 480 \times 8 = 2,457,600 \text{ ビット} = \frac{2,457,600}{8} \text{ バイト} = 307,200 \text{ バイト}$$

これを、RGBカラーで表現すると、各色ごとに、8ビットが必要となるので、グレースケールの場合の3倍となる。

#### 1.6.4 ラスタ化による図形の描画

デジタル画像のデータ表現

- (1) ラスタ表現
- (2) ベクタ表現

## [1] 線分のラスタ化

画素に線分を書き込む方法

2点  $A(x_A, y_A)$ ,  $B(x_B, y_B)$  を両端点とする線分のラスタ化  
線分の方程式

$$y = \frac{\Delta y}{\Delta x}(x - x_A) + y_A$$

ただし、 $\Delta x = x_B - x_A$ ,  $\Delta y = y_B - y_A$

ここで、 $x_A, y_A, x_B, y_B$  は、すべて整数。簡単のために、 $0 \leq \frac{\Delta y}{\Delta x} \leq 1$  かつ  $x_A < x_B$  とする。 $A$  から  $B$  まで線分を描くには、 $x_A$  から  $x_B$  まで、 $x$  の値を1ずつ増やす。各整数  $x$  における  $y$  の値を上の直線の方程式より計算し、小数点以下を四捨五入して整数化する。こうして求めた点  $(x, y)$  の画素を塗りつぶす。

線分のラスタ化を高速に行う方法 — ブレゼンハム (Bresenham) のアルゴリズム

## [2] ポリゴンのラスタ化

ポリゴンをラスタ化するには、水平方向の線 (スキャンライン) に沿って、左側の稜線から右側の稜線までを塗りつぶせば良い。

## 1.6.5 エイリアシングとアンチエイリアシング

画像のラスタ化における問題点

- 斜めの直線がギザギザの階段状になってしまう (ジャギー jaggy)。  
jagged - のこぎりの歯のような、ぎざぎざの
- 細かい点や線の消去、分離。または、実際には存在しない縞模様の出現。これをエイリアシング (aliasing) と言う。  
alias - 別名、偽名

## [1] エイリアシングが生じる理由

細かな縞模様を、その周期の  $1/2$  以上の間隔で標本化すると、擬似的な縞模様が現れる。

**標本化定理 (シャノン)** 一般に、縞模様 (サインカーブ) の周期の  $1/2$  未満の間隔で標本化しなければ、元の縞模様は正しく表せない。

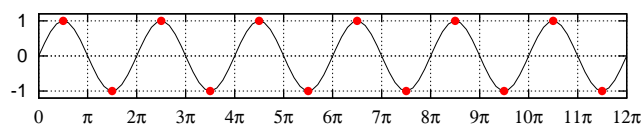


図 1.1: 標本化間隔が周期の  $1/2$  で、これはうまくいく。

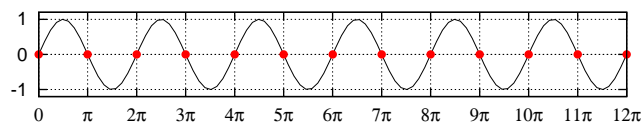


図 1.2: 標本化間隔が同じ周期の  $1/2$  でも、これはうまくいかない。

【仮定】元の連続信号が帯域制限されている = ある最大周波数以下の周波数成分しか含まない。

## [2] 画像処理でのアンチエイリアシング

**アンチエイリアシング** エイリアシングの影響を低減する処理

アナログ信号(元の画像)のフーリエ変換を行った時、サンプリング周波数の  $1/2$  より短い短い周波数は、これを取り除く。これは、画像をぼかすことに相当する。この処理により、サンプリング周波数/2 より細かな縞模様は、見えなくなる。

## [3] CG でのアンチエイリアシング

線分のアンチエイリアシング - 背景と図形の中間の色の画素を図形の境界部分に配置する。

## 1.6.6 いろいろな画像

### [1] 2 値画像

白または黒の 1 ビットの画素値を持つ画像



### グレースケール画像

画像を白から黒までの明暗だけで表現し、色の情報は含まない「モノクロ」のこと。灰色を何階調で表現するかをビット数によって表す。8ビットなら(白と黒を含めて)256階調、16ビットなら 65536階調の灰色で表現する。

### カラー画像

R, G, B 表現色で表すのが一般的。

各色を8ビットの濃淡レベルで表した画像を**フルカラー画像**と呼ぶ。

## 1.7 画像処理の分類と役割

### 1.7.1 画像を出力する処理

#### [1] 画像の幾何学的変換

デジタル画像上の2次元直交座標系 (**画像座標系**) では、画素の座標値が整数のところに格子点 (画素) が置かれる。このような座標系でも幾何学的変換を考えることができる。

デジタル画像の幾何学的変換 - 個々の画素に幾何学的変換を適用する。

#### [2] 画像の再標本化と補間

デジタル画像の標本化位置は、変換前も返還後も等間隔の格子点上にある。

変換後の標本点 (整数の座標値) から返還前の標本点を逆変換で求めると、得られた標本点の座標値は一般には整数でなくなる。この整数ではない座標値の画素値を求めることを**画像の再標本化**と言う。

画像の再標本化は、周囲の格子点における画素値を用いて行う。これを**補間**という。

#### 代表的な補間方法

##### ニアレストネイバー (最近隣内挿法)

補間したい点に最も近い格子点の画像データを求める点の画像データとする。

$$P = P_{i,j}$$

$$\text{ただし、} i = [x' + 0.5], j = [y' + 0.5]$$

$i, j$  は、それぞれ、 $x', y'$  に最も近い整数。

##### バイリニア補間 (双一次補間)

求めたい点の画像データを、その周囲4つの格子点の画像データの一次式によって求める方法。

$$\begin{aligned} P &= ((j+1) - y') \left[ ((i+1) - x') P_{i,j} + (x' - i) P_{i+1,j} \right] \\ &\quad + (y' - j) \left[ ((i+1) - x') P_{i,j+1} + (x' - i) P_{i+1,j+1} \right] \\ &= ((i+1) - x') ((j+1) - y') P_{i,j} \\ &\quad + ((i+1) - x') (y' - j) P_{i,j+1} \\ &\quad + (x' - i) ((j+1) - y') P_{i+1,j} \\ &\quad + (x' - i) (y' - j) P_{i+1,j+1} \end{aligned}$$

ただし、 $i \leq x' \leq I + 1$ ,  $j \leq y' \leq j + 1$ .

バイナリ補間により平滑化が行われる。。

### [3] イメージモザイク

複数の画像を接合して1枚の画像を作成することを、**イメージモザイク**という。そのためには、**位置合わせ**と**色合わせ**が必要となる。

## 1.7.2 画像から情報を抽出する処理

**3次元再構成** 実写画像から被写体の3次元形状を求める手法。  
これは、一般的には解けない問題である。

**陰影からの復元** 拡散反射面の陰影から形状を復元する手法

**ステレオビジョン** 異なる2方向から見た画像を用いた3次元復元

## 1.7.3 入出力の伝送・蓄積のための処理

画像圧縮フォーマット PNG, GIF, JPEG など、  
現在、広く使われている画像圧縮フォーマットは JPEG である。



## 第2章 画像の濃淡変換とフィルタリング処理

### 2.1 画像の性質を表す諸量

#### 2.1.1 ヒストグラム

デジタル画像 - 画素値の集合

画素値の分布 - 画像の濃淡ヒストグラム、単にヒストグラム  
s コントラストの低い画像 - 分布が狭い範囲にかたまる。

コントラストの高い画像 - 分布が広範囲にわたる。

#### 2.1.2 画像の統計量

$M \times N$  画素の大きさの画像において、位置  $(i, j)$  の画素値を  $f(i, j)$  とおくと、平均値  $\mu$  と分散  $\sigma^2$  は、

$$\mu = \frac{\sum_{i,j} f(i, j)}{MN}$$

$$\sigma^2 = \frac{\sum_{i,j} (f(i, j) - \mu)^2}{MN}$$

で与えられる。

【注】確率の平均と分散

確率変数  $x$  の従う確率分布関数を  $p(x)$  とすると、 $x$  の平均  $\mu$  と分散  $\sigma^2$  は、

$$\mu = \sum_x x p(x)$$

$$\sigma = \sum_x (x - \mu)^2 p(x)$$

で与えられる。そこで、 $M \times N$  画素の標本においては、画素値  $x = f(i, j)$  の出現確率を  $1/MN$  とみなす、すなわち、 $p(x) = \frac{1}{MN}$  と置くことにより、画素の平均と分散を得る。

## その他の統計量

**最小値** 標本の最小値

**最大値** 標本の最大値

**中央値 (メディアン)** その値より小さい観測値の数と大きな観測値の数が等しくなるような値

**最頻値 (モード)** 最も多く現れる観測値

**【例1】**  $4 \times 4 = 16$  の画素値の頻度と統計量

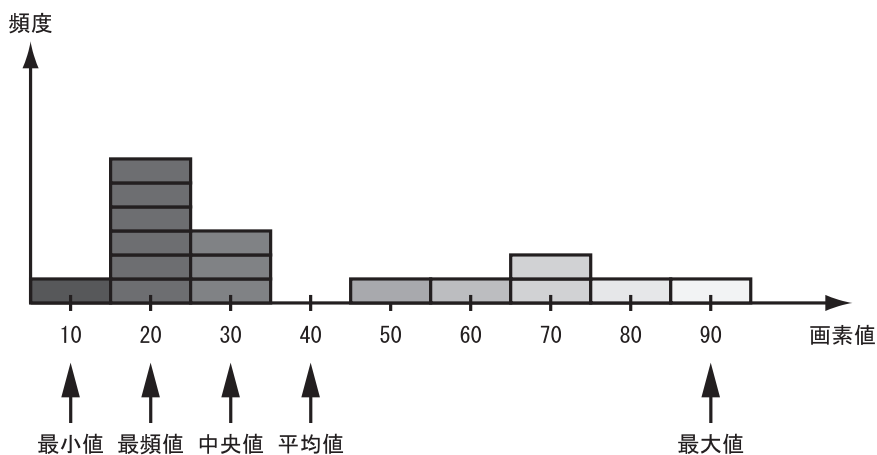
画素値	10	20	30	40	50	60	70	80	90
頻度	1	6	3	0	1	1	2	1	1

平均

$$\mu = \frac{10 + 20 \times 6 + 30 \times 3 + 50 + 60 + 70 \times 2 + 80 + 90}{16} = 40$$

分散

$$\begin{aligned} \sigma^2 &= \frac{1}{16} \left\{ (10 - 40)^2 + 6 \times (20 - 40)^2 + 3 \times (30 - 40)^2 + (50 - 40)^2 \right. \\ &\quad \left. + (60 - 40)^2 + 2 \times (70 - 40)^2 + (80 - 40)^2 + (90 - 40)^2 \right\} \\ &= 625 \end{aligned}$$



【例 2】  $5 \times 5 = 25$  画素の統計量 (教科書 p.134 図 5.5)

画素値	30	40	50	60	70	80
頻 度	1	7	4	6	5	2

平均

$$\begin{aligned}\mu &= \frac{30 \times 1 + 40 \times 7 + 50 \times 4 + 60 \times 6 + 70 \times 5 + 80 \times 2}{25} \\ &= 55.2\end{aligned}$$

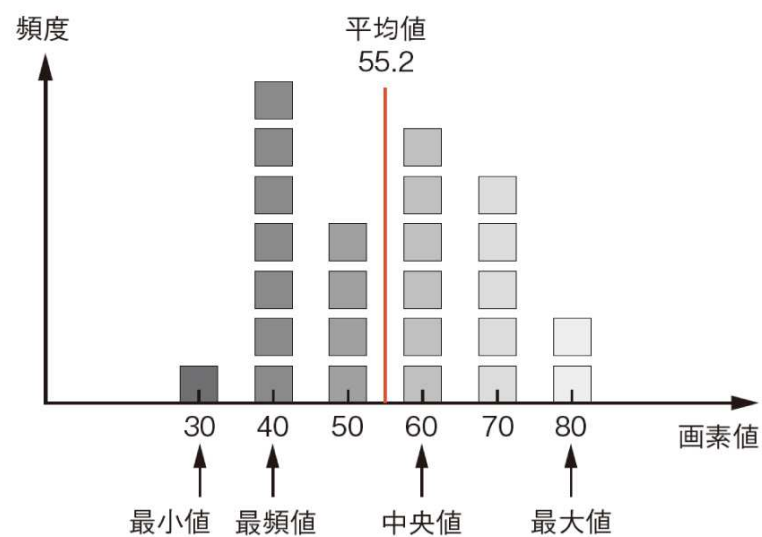
分散

$$\begin{aligned}\sigma^2 &= \frac{1}{25} \left\{ 1 \times (30 - 55.2)^2 + 7 \times (40 - 55.2)^2 + 4 \times (50 - 55.2)^2 \right. \\ &\quad \left. + 6 \times (60 - 55.2)^2 + 5 \times (70 - 55.2)^2 + 2 \times (80 - 55.2)^2 \right\} \\ &= \frac{4824}{25} = 192.96\end{aligned}$$

標準偏差

$$\sigma = \sqrt{\sigma^2} = 13.89100 \dots$$

■図5.5——ヒストグラムと統計量



## 2.2 画素ごとの濃淡変換

### 2.2.1 トーンカーブ

画像の濃淡の変換

入力画像の画素値  $\leftarrow$  対応関係  $\rightarrow$  出力画像の画素値

対応関係を与える関数 - 階調変換関数

その関数のグラフ - トーンカーブ

### 2.2.2 折れ線型のトーンカーブ

図 2.7 画像が全体的に明るくなり、黒くつぶれていた部分が見えやすくなった。

図 2.8 トーンカーブの傾きが 1 より大きい範囲でコントラストが上がった。

折れ線型トーンカーブでは、水平な部分では、出力画像の画素値がすべて一定になるために、入力画像のその範囲の濃淡変化は完全に失われる。

### 2.2.3 指数対数型トーンカーブ

折れ線型のトーンカーブの欠点を補うことができる。

入力画像の画素値を  $x$ 、出力画像の画素値を  $y$  とすると、トーンカーブは、

$$y = 255 \left( \frac{x}{255} \right)^{\frac{1}{\gamma}}$$

このような変換をガンマ変換またはガンマ補正と言う。

### 2.2.4 S字トーンカーブ

画素値中央に偏ったコントラストの低い画像では、画素値が分散されるため、コントラストが上がる。

### 2.2.5 ヒストグラムの平坦化

画像のヒストグラムで、その分布が集中している部分を引き延ばす作業を自動化する方法に、ヒストグラム平坦化処理がある。



### 2.2.6 濃淡の反転

ネガ・ポジ反転とも呼ばれる。

### 2.2.7 ポスタリゼーション

階段状のトーンカーブ 出力画像の階調を落とすこと。階調変更。  
とくに、出力画像の画素値が2段階に制限された処理を**2値化**と言う。

### 2.2.8 ソラリゼーション (solarisation)

画像の濃淡の一部を反転させることにより、ネガ画像とポジ画像が混ざり合った特殊な効果を出させること。

### 2.2.9 カラー画像の変換

R, G, B の各色に対応する3つのトーンカーブにより変換を行う。通常は、3つのトーンカーブを同一にする。

### 2.2.10 擬似カラー

グレースケール画像 - R, G, B 各色の画素値がすべて等しい。

R, G, B の各色について異なるトーンカーブを用いることで、グレースケール画像に色づけを行う。これを**擬似カラー**と言う。

### 2.2.11 色相、彩度、明度の変化

色相、彩度、明度に変換したうえで、それぞれの属性を調整する方法もよく用いられる。

## 2.3 領域に基づく濃淡変換

### 2.3.1 空間フィルタリング

入力画像のある領域内の画素の値を用いて、出力画像の画素値を求める処理。

#### 線形フィルタ

出力画像  $(i, j)$  の画素値を求める場合、フィルタの原点と入力画像の  $(i, j)$  の位置を合わせる。フィルタの係数と対応する入力画像値を掛け算した結果をすべて足し合わせる (積和演算)。得られた値が出力画像の  $(i, j)$  における画素値となる。

非線形フィルタ 線形フィルタ以外のもの

### 2.3.2 平滑化

画像の処理によって濃淡の変化を滑らかにする処理のこと。

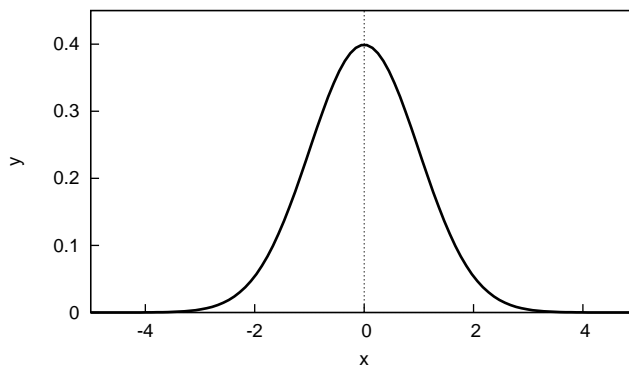
平均化フィルタ フィルタによって覆われる領域内の平均値を求めるもの

原点に近いほど大きな重みを付ける **重み付き平均**を求めるフィルタもある。

ガウシアンフィルタ 重みをガウス分布に近づけたもの  
ガウス分布

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

ここに、平均は  $\mu$ 、分散は  $\sigma^2$  である。



平滑化を特定の方向に限って行うフィルタもある。

**メディアンフィルタ** 領域内の中央値(メディアン)を出力するフィルタ。非線形フィルタの一種。スパイク状のノイズ除去に効果的。

### 2.3.3 エッジ抽出

#### 微分フィルタ

注目画素とその右隣の画素との差を出力するフィルタ

位置  $(i, j)$  の画素値を  $f(i, j)$  とする。位置  $(i, j)$  における出力画像の画素値  $g(i, j)$

0	0	0
0	-1	1
0	0	0

表 2.1: 微分フィルタ

$f(i-1, j+1)$	$f(i, j+1)$	$f(i+1, j+1)$
$f(i-1, j)$	$f(i, j)$	$f(i+1, j)$
$f(i-1, j-1)$	$f(i, j-1)$	$f(i+1, j-1)$

表 2.2: 入力画像の画素値

は、

$$g(i, j) = f(i+1, j) - f(i, j) \equiv \Delta_x f(i, j)$$

となる。

微分

$$\frac{df(x)}{dx} = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$

に対し、

$$\Delta f(i) = f(i+1) - f(i)$$

を差分という。

#### ラプラシアンフィルタ

2階微分  $\frac{d^2 f(x)}{dx^2}$  に相当する2階差分は、

$$\begin{aligned} \Delta^2 f(i) &= [f(i+1) - f(i)] - [f(i) - f(i-1)] \\ &= f(i+1) - 2f(i) + f(i-1) \end{aligned}$$

連続2変数の場合のラプラシアンは、

$$\frac{\partial^2}{\partial x^2} f(x, y) + \frac{\partial^2}{\partial y^2} f(x, y)$$

離散的な場合のラプラシアンは、

$$\begin{aligned} \nabla^2 f(i, j) &= \Delta_i^2 f(i, j) + \Delta_j^2 f(i, j) \\ &= f(i+1, j) - 2f(i, j) + f(i-1, j) + f(i, j+1) - 2f(i, j) + f(i, j-1) \\ &= f(i+1, j) + f(i-1, j) + f(i, j+1) + f(i, j-1) - 4f(i, j) \end{aligned}$$

これはラプラシアンフィルタを、

0	1	0
1	-4	1
0	1	0

とおくことに相当する。

### 2.3.4 鮮鋭化

入力画像の濃淡を残したままエッジを強調するフィルタを**鮮鋭化フィルタ**という。4近傍鮮鋭化フィルタは、入力画像  $f(i, j)$  から出力画像

$$f(i, j) - \nabla^2 f(i, j)$$

を作り出す。これは、これはフィルタを、

0	-1	0
-1	5	-1
0	-1	0

とおくことと同じである。

## 2.4 そのほかの処理

### 2.4.1 画像構成要素の置き換え

画素 - 画像構成要素の最小単位

正方形または長方形に並んだ画素、任意の多角形を形成する画素の集合などを構成要素と考えることができる。

**モザイク処理** 画素のブロックをその中の平均値で置き換えたもの

### 2.4.2 画像間演算

2枚またはそれ以上の入力画像を用いて、それぞれの画像の同じ位置にある画素ごとに、ある算術演算や論理演算をおこなうもの。

#### アルファブレンディング (alpha blending)

2枚の入力画像の対応する画素値を  $f_1$ ,  $f_2$  としたとき、

$$g = \alpha f_1 + (1 - \alpha) f_2$$

と  $\alpha (0 \leq \alpha \leq 1)$  で重みを付けて表したものを言う。

$\alpha$  の値を時間的に変化させたシーンの切替を **ディゾルブ (dissolve)** または **オーバーラップ (overlap)** と言う。

dissolve 分解する。溶かす。

overlap 重ねる。

**エンボス (emboss)** 浮彫にする。

1. 入力画像のネガ・ポジ反転画像の生成
2. 作成した画像を適当な方向に数画素平行移動する。
3. 入力画像  $f_1$  と移動後の画像を  $f_2$  より、以下の演算で出力画像  $g$  を作成する。

$$g = f_1 + f_2 - 128$$

ただし、画素値は  $0 \sim 255$  とする。 $128 = 256/2$ 。 $g$  が  $0$  より小さいときは  $0$  に、 $255$  より大きいときは  $255$  とする。

### 2.4.3 マスク処理 (mask processing)

2枚の画像と合成して1枚の画像を作成するさいに、画素ごとに1, 0で指定した画像 (**マスク画像**) をあらかじめ作成し、それに基づいて画像作成を行う方法。

#### クロマキー (chroma key)

chroma 色 color (ギリシア語の色から)

key 鍵、手がかり

各画素をその色 (chroma) を鍵 (key) として、単一色の背景を切り分ける処理

## 第3章 モデリング

### 3.1 形状モデル

#### 3.1.1 ワイヤフレームモデル (wireframe model)

直線や曲線の稜線だけを記録することによって立体の形状を表現

**利点**

直線要素の多い立体を高速に表示

**欠点**

どこに面があるのか、どこが立体の内部か分からない。立体どうしの干渉計算、隠面消去、陰線消去などの処理ができない。

#### 3.1.2 サーフェスモデル (surface model)

立体の面データだけを表現

**利点**

隠面消去、陰線消去、陰影表示ができる。

**欠点**

どこが物体の内部か分からない。

体積の計算、物体どうしの集合演算ができない。

#### 3.1.3 ソリッドモデル

中身が詰まった立体としての完全なデータとして表現

**利点**

レンダリングや陰影表示、集合演算が可能

**欠点**

データ量が多い。

### 3.1.4 形状モデルと表示

一般に、物体の表示結果から、その物体がどの立体モデルとして表現されているかを判定することは困難である。

## 3.2 ソリッドモデルの形状表現

### 3.2.1 境界表現

**B-reps**(boundary representation) と呼ばれる。

頂点 (vertex)、稜線 (edge)、面 (face) のデータを数値によって表しそれらの接続関係をグラフで示すことによって立体を表現。

**利点**

どんな立体であっても表現することが可能

**欠点**

多量の情報が必要

### 3.2.2 CSG 表現

**CSG** — Constructive Solid Geometry(空間領域構成法)

立体を基本立体の集合演算で表現

基本立体 (primitive) — 直方体、円柱、多面体、錐体、球

集合演算

和 (union)                     $\cup$      $A \cup B$

積 (intersection)     $\cap$      $A \cap B$     (共通部分とも言う。)

差 (difference)         $-$      $A - B$

**利点**

少ない情報量で立体の表現が可能

**欠点**

表現できる形状は基本立体の種類に依存



### 3.2.3 スイープ表現 (sweep representation)

平面図形を定められた軌道 (trajectory) に沿って移動したときの軌跡として表現  
利点

少ない情報量で立体の表現が可能

欠点

表現できる形状が限定される。

### 3.2.4 オイラー操作

ソリッドモデルで扱う立体 — 二多様体 (two-manifold)

二多様体でないもの — 非多様体 (non-manifold)

多面体に対するオイラー (Euler) の公式

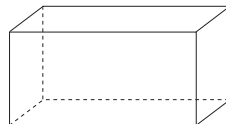
単純な (穴の無い) 多面体の頂点の数を  $V$ , 稜の数を  $E$ , 面の数を  $F$  とすると、常に、

$$V - E + F = 2$$

である。

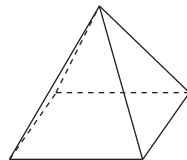
**例 1** 直方体

$V = 8, E = 12, F = 6$  だから  $V - E + F = 8 - 12 + 6 = 2$



**例 2** 四角錐

$V = 5, E = 8, F = 5$  だから  $V - E + F = 5 - 8 + 5 = 2$



オイラーの公式を保持する変形操作を**オイラー操作** (Euler operation) と言う。  
すなわち、

$$\Delta V - \Delta E + \Delta F = 0$$

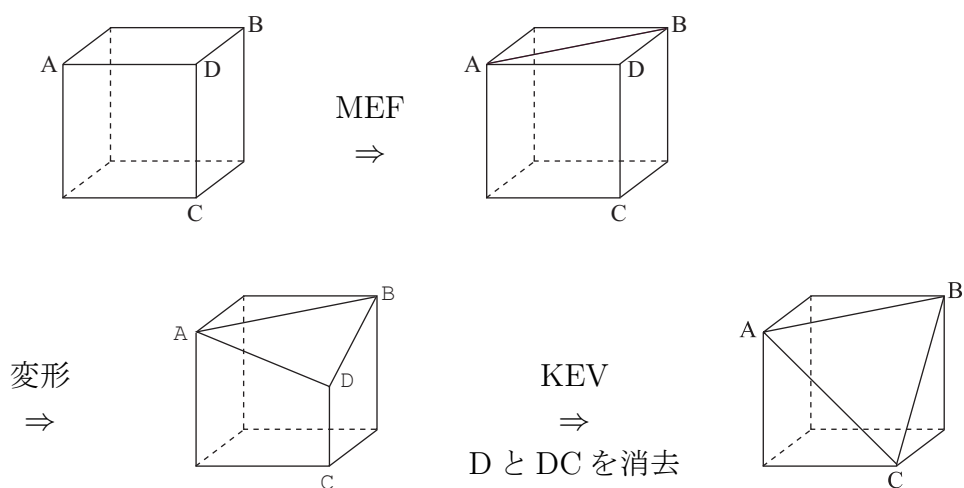
を満たす操作である。

- 新しい頂点と稜線を追加 : MEV(Make Edge Vertex)  
 $\Delta V = 1, \Delta E = 1, \Delta F = 0.$
- 頂点と稜線の削除 : KEV(Kill EdgeVertex)  
 $\Delta V = -1, \Delta E = -1, \Delta F = 0.$
- 新しい稜線と面の追加 (面の分割) : MEF(Make Edge Face)  
 $\Delta V = 0, \Delta E = 1, \Delta F = 1.$
- 稜線と面の削除 : KEF(Kill Edge Face)  
 $\Delta V = 0, \Delta E = -1, \Delta F = -1.$

【例】図のように、立方体から三角錐 ABCD を切り取る。



手順



立方体  $V = 8, E = 12, F = 6$  で  $V - E + F = 8 - 12 + 6 = 2$

隅を切り取った図形  $V = 7$ ,  $E = 12$ ,  $F = 7$  で  $V - E + F = 7 - 12 + 7 = 2$

$V$ ,  $E$ ,  $F$  の変化  $\Delta V = -1$ ,  $\Delta E = 0$ ,  $\Delta F = 1$  だから、 $\Delta V - \Delta E + \Delta F = 0$

### オイラーの公式の証明

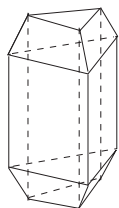
多面体の面の一つを取り除く。その面の一つが除かれた多面体を平面に広げる。そのようにして得られた2次元図形は、面の数が一つ減っただけで、頂点の数と稜線の数は変わらない。得られた図形を三角分割する。これは、稜が1つ、面が一つ増えるだけなので、 $V - E + F$ は変わらない。外側の三角形を一つずつ取り除く。その場合にも、 $\Delta V - \Delta E + \Delta F = 0$  が成り立つ。最後に三角形が一つ残る。これは、 $V = 3$ 、 $E = 3$ 、 $F = 1$  だから、 $V - E + F = 1$  である。 $V - E + F$ は変わらないので、元の平面図形も  $V - E + F = 1$  である。したがって、面を一つ除いた多面体も  $V - E + F = 1$  である。結局、完全な多面体は、面が一つ多いので、 $V - E + F = 2$  となる。

### 穴のある場合のオイラーの公式

穴の数を  $g$  とすると、

$$V - E + F = 2 - 2g$$

【例題】図のように、真ん中に穴の開いた立体を考える。 $V = 12$ 、 $E = 24$ 、 $F = 12$ 、 $g = 1$  だから、 $V - E + F = 0 = 2 - 2 \times 1$  となり、確かに成り立っている。



## 3.3 曲線・曲面

### 3.3.1 曲線の表現形式

#### [1] 代数表現

陽関数形式 (explicit function form)  $y = f(x)$    【例】放物線  $y = x^2$

1つの  $x$  の値に対して1つの  $y$  の値しか表現できない。

陰関数形式 (implicit function form)  $f(x, y) = 0$    【例】円  $f(x, y) = x^2 + y^2 - r^2 = 0$

同一の  $x$  の値にたいして、複数の  $y$  の値が対応する。

## [2] パラメトリック表現

パラメータ (parameter - 媒介変数) を用いた表現

【例】円  $x = r \cos t, y = r \sin t \quad (0 \leq t \leq 2\pi)$

## 3.3.2 2次曲線

陰関数形式  $ax^2 + by^2 + c + 2dxy + 2ex + 2fy = 0$

2次曲線は円錐曲線とも呼ばれる。

## 種類と標準形

楕円 (ellipse)

$$\text{陰関数形式} \quad \frac{x^2}{a^2} + \frac{y^2}{b^2} - 1 = 0 \quad (a, b > 0)$$

$$\text{パラメータ形式} \quad x = a \cos \theta, \quad y = b \sin \theta$$

放物線 (parabola)

$$\text{陰関数形式} \quad y^2 - ax = 0$$

双曲線 (hyperbola)

$$\text{陰関数形式} \quad \frac{x^2}{a^2} - \frac{y^2}{b^2} - 1 = 0 \quad (a, b > 0)$$

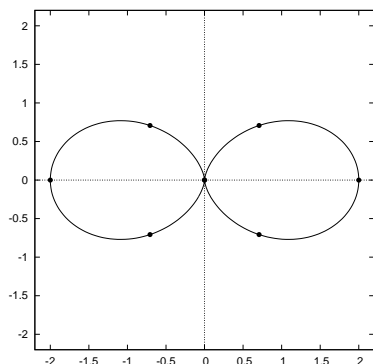
$$\text{パラメータ形式} \quad x = a \sec \theta, \quad y = b \tan \theta$$

$$\text{または、} \quad x = a \cosh t, \quad y = b \sinh t$$

## 3.3.3 極方程式で表された曲線

【例】  $r = 1 + \cos 2\theta$  の表す曲線

$\theta$	0	...	$\frac{\pi}{4}$	...	$\frac{\pi}{2}$	...	$\frac{3\pi}{4}$	...	$\pi$	...	$\frac{5\pi}{4}$	...	$\frac{3\pi}{2}$	...	$\frac{7\pi}{4}$	...	$2\pi$
$r$	2		1		0		1		2		1		0		1		2



### 3.3.4 パラメトリック曲線

#### [1] ベジエ曲線

**ベジエ曲線** (Bézier curve) とは、**制御点** (control point) と呼ばれる複数の点  $\{\mathbf{P}_i\}$  に基づいて定義される多項式曲線である。

バーンスタイン (Bernstein) 多項式 (バーンスタイン基底関数)

$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i} \quad \text{ここで、} \binom{n}{i} = {}_n C_i \text{ は、二項係数}$$

バーンスタイン多項式は、次の関係式を満たす。

$$\begin{aligned} \sum_{i=0}^n B_i^n(t) &= \sum_{i=0}^n \binom{n}{i} t^i (1-t)^{n-i} \\ &= (t + (1-t))^n \\ &= 1 \end{aligned}$$

#### $n$ 次ベジエ曲線

$$\mathbf{P}(t) = \sum_{i=0}^n B_i^n(t) \mathbf{P}_i \quad (0 \leq t \leq 1)$$

$n$  次のベジエ曲線を定義するためには、 $n+1$  コの制御点が必要である。

## 3 次ベジエ曲線

$$\mathbf{P}(t) = \mathbf{P}_0 B_0^3(t) + \mathbf{P}_1 B_1^3(t) + \mathbf{P}_2 B_2^3(t) + \mathbf{P}_3 B_3^3(t)$$

ここで、

$$\begin{aligned} B_0^3(t) &= \binom{3}{0} t^0 (1-t)^3 = (1-t)^3 \\ B_1^3(t) &= \binom{3}{1} t^1 (1-t)^2 = 3t(1-t)^2 \\ B_2^3(t) &= \binom{3}{2} t^2 (1-t)^1 = 3t^2(1-t) \\ B_3^3(t) &= \binom{3}{3} t^3 (1-t)^0 = t^3 \end{aligned}$$

特に、

$$\begin{aligned} B_0^3(0) &= 1, \quad B_1^3(0) = B_2^3(0) = B_3^3(0) = 0 \\ B_3^3(1) &= 1, \quad B_0^3(1) = B_1^3(1) = B_2^3(1) = 0 \end{aligned}$$

だから、

$$\mathbf{P}(0) = \mathbf{P}_0, \quad \mathbf{P}(1) = \mathbf{P}_3$$

ベジエ曲線の両端点は両端の制御点に一致する。

$$\begin{aligned} \frac{dB_0^3(t)}{dt} &= -3(1-t)^2, & \frac{dB_0^3(0)}{dt} &= -3, \quad \frac{dB_0^3(1)}{dt} = 0 \\ \frac{dB_1^3(t)}{dt} &= 3(1-t)^2 - 6t(1-t), & \frac{dB_1^3(0)}{dt} &= 3, \quad \frac{dB_1^3(1)}{dt} = 0 \\ &= 3t(1-t)(1-3t) \\ \frac{dB_2^3(t)}{dt} &= 6t(1-t) - 3t^2, & \frac{dB_2^3(0)}{dt} &= 0, \quad \frac{dB_2^3(1)}{dt} = -3 \\ &= 3t(2-3t) \\ \frac{dB_3^3(t)}{dt} &= 3t^2, & \frac{dB_3^3(0)}{dt} &= 0, \quad \frac{dB_3^3(1)}{dt} = 3 \end{aligned}$$

ゆえに、

$$\begin{aligned}\frac{d\mathbf{P}(0)}{dt} &= 3(\mathbf{P}_1 - \mathbf{P}_0) \\ \frac{d\mathbf{P}(1)}{dt} &= 3(\mathbf{P}_3 - \mathbf{P}_2)\end{aligned}$$

接ベクトルは端の制御点とその1つ隣の制御点を結んだ向きになる。

曲線は4つの制御点を含む最小の凸多角形の中に含まれる (凸包性)。これは、細分割法 (ド・カステリョのアルゴリズム) より明らか。

一般に、

$$\mathbf{P} = \sum_{i=0}^3 a_i \mathbf{P}_i \quad \text{で} \quad \sum_{i=0}^3 a_i = 1 \quad \text{かつ} \quad a_i \geq 0$$

ならば、 $\mathbf{P}$  は、 $\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3, \mathbf{P}_4$  で作られる凸四角形の中に含まれる。

【証明】

1.  $\mathbf{P} = a_0 \mathbf{P}_0 + a_1 \mathbf{P}_1$  で、 $a_0 + a_1 = 1$ , かつ  $a_0 \geq 0, a_1 \geq 0$   
ならば、 $\mathbf{P}$  は、 $\mathbf{P}_0$  と  $\mathbf{P}_1$  を内分する点だから、 $\mathbf{P}_0$  と  $\mathbf{P}_1$  の間にある。
2.  $\mathbf{P} = a_0 \mathbf{P}_0 + a_1 \mathbf{P}_1 + a_2 \mathbf{P}_2$  で、 $a_0 + a_1 + a_2 = 1$ , かつ  $a_0 \geq 0, a_1 \geq 0, a_2 \geq 0$   
ならば、

$$\mathbf{P} = a_0 \mathbf{P}_0 + (1 - a_0) \left( \frac{a_1 \mathbf{P}_1 + a_2 \mathbf{P}_2}{a_1 + a_2} \right) = a_0 \mathbf{P}_0 + (1 - a_0) \mathbf{Q}$$

と書ける。 $\mathbf{Q} = \frac{a_1 \mathbf{P}_1 + a_2 \mathbf{P}_2}{a_1 + a_2}$  は、線分  $\mathbf{P}_1 \mathbf{P}_2$  を内分する点で、 $\mathbf{P}$  は、線分  $\mathbf{PQ}$  を内分する点である。ゆえに、 $\mathbf{P}$  は、 $\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2$  で作られる3角形の内部の点である。

3.  $\mathbf{P} = \sum_{i=0}^3 a_i \mathbf{P}_i$  で  $\sum_{i=0}^3 a_i = 1$  かつ  $a_i \geq 0$   
ならば、

$$\begin{aligned}\mathbf{P} &= a_0 \mathbf{P}_0 + a_1 \mathbf{P}_1 + a_2 \mathbf{P}_2 + a_3 \mathbf{P}_3 \\ &= a_0 \mathbf{P}_0 + (1 - a_0) \left( \frac{a_1 \mathbf{P}_1 + a_2 \mathbf{P}_2 + a_3 \mathbf{P}_3}{a_1 + a_2 + a_3} \right) \\ &= a_0 \mathbf{P}_0 + (1 - a_0) \mathbf{R}\end{aligned}$$

と書ける。 $\mathbf{R}$  は三角形  $\mathbf{P}_1 \mathbf{P}_2 \mathbf{P}_3$  の中の点で、 $\mathbf{P}$  は  $\mathbf{P}_0 \mathbf{R}$  を内分する点である。ゆえに、 $\mathbf{P}$  は、 $\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3$  から作られる凸4角形の内部の点である。



3 次ベジエ曲線  $\mathbf{P}(t) = \mathbf{P}_0 B_0^3(t) + \mathbf{P}_1 B_1^3(t) + \mathbf{P}_2 B_2^3(t) + \mathbf{P}_3 B_3^3(t)$  は、  
 $\sum_{i=0}^3 B_i^3(t) = 1$ , かつ  $B_i^3(t) \geq 0$  ( $i = 0, \dots, 3$ ) なので、凸包性を満たす。

### 細分割法 (ド・カステリヨ (de Casteljau) のアルゴリズム)

$$\begin{aligned} \mathbf{P}(t) = & (1-t) \left\{ (1-t) \left[ (1-t)\mathbf{P}_0 + t\mathbf{P}_1 \right] + t \left[ (1-t)\mathbf{P}_1 + t\mathbf{P}_2 \right] \right\} \\ & + t \left\{ (1-t) \left[ (1-t)\mathbf{P}_1 + t\mathbf{P}_2 \right] + t \left[ (1-t)\mathbf{P}_2 + t\mathbf{P}_3 \right] \right\} \end{aligned}$$

**【例】** 制御点  $\vec{P}_0 = (1, 1)$ ,  $\vec{P}_1 = (0, 0)$ ,  $\vec{P}_2 = (1, -1)$  によって定義される 2 次ベジエ曲線

- (1) 2 点  $\vec{P}_0, \vec{P}_1$  を  $t : (1-t)$  に内分する点を  $\vec{P}_a$  とすると、  
 $\vec{P}_a = (1-t)\vec{P}_0 + t\vec{P}_1 = (1-t, 1-t)$
- (2) 2 点  $\vec{P}_1, \vec{P}_2$  を  $t : (1-t)$  に内分する点を  $\vec{P}_b$  とすると、  
 $\vec{P}_b = (1-t)\vec{P}_1 + t\vec{P}_2 = (t, -t)$
- (3) 2 点  $\vec{P}_a, \vec{P}_b$  を  $t : (1-t)$  に内分する点を  $\vec{P}(t)$  とすると、これが、求める 2 次ベジエ曲線である。  
 $\vec{P}(t) = (1-t)\vec{P}_a + t\vec{P}_b = ((1-t)^2 + t^2, (1-t)^2 - t^2) = (1-2t+2t^2, 1-2t)$   
すなわち、 $x(t) = 1-2t+2t^2$ ,  $y(t) = 1-2t$
- (4)  $x(t)$  を  $y(t)$  の関数として表す。 $t = (1-y)/2$  を  $x(t) = 1-2t+2t^2$  に代入して、  
 $x = \frac{1}{2}(y^2 + 1)$  を得る。

### [2]B スプライン曲線

B-spline : base-spline の略

制御点  $\{\mathbf{P}_i\}$  とノット列 (ノットベクトル : knot vector)  $\{t_i\}$  によって定義された曲線

**ベジエ曲線** 大域的に定義されている。制御点の位置の変更が曲線全体に及ぶ。

**B スプライン曲線** 局所的に定義されている。制御点が及ぼす影響の範囲で局所的に定義

$t_i < t < t_{i+1}$  をセグメントと言う。 $n$  次の B スプライン曲線が 1 セグメントを定義するためには、 $n+1$  コの制御点が必要である。

セグメント	コントロールポイント	$t$ の範囲
1 コ	$\mathbf{P}_0, \mathbf{P}_1, \dots, \mathbf{P}_n$	$t_n \leq t \leq t_{n+1}$
2 コ	$\mathbf{P}_0, \mathbf{P}_1, \dots, \mathbf{P}_n, \mathbf{P}_{n+1}$	$t_n \leq t \leq t_{n+2}$
3 コ	$\mathbf{P}_0, \mathbf{P}_1, \dots, \mathbf{P}_n, \mathbf{P}_{n+1}, \mathbf{P}_{n+2}$	$t_n \leq t \leq t_{n+3}$
$\vdots$	$\vdots$	
$L$ コ	$\mathbf{P}_0, \mathbf{P}_1, \dots, \mathbf{P}_n, \mathbf{P}_{n+1}, \mathbf{P}_{n+2}, \dots, \mathbf{P}_{n+L-1}$	$t_n \leq t \leq t_{n+L}$

$L$  コのセグメントから構成される  $n$  次の B スプライン曲線は、制御点  $\mathbf{P}_0, \mathbf{P}_1, \dots, \mathbf{P}_{n+L-1}$  に基づいて、

$$\mathbf{P}(t) = \sum_{i=0}^{n+L-1} \mathbf{P}_i N_i^n(t) \quad N_i^n(t) \neq 0 \text{ となるのは、} t_0 < t < t_{n+1}$$

で定義される。 $N_i^n(t) \neq 0$  となるのは、 $t_i < t < t_{n+i+1}$  である。すなわち、

$$N_0^n(t) \neq 0 \text{ となるのは、} t_0 < t < t_{n+1}$$

$$N_1^n(t) \neq 0 \text{ となるのは、} t_1 < t < t_{n+2}$$

$$\vdots$$

$$N_n^n(t) \neq 0 \text{ となるのは、} t_n < t < t_{2n+1}$$

$$\vdots$$

$$N_{L-1}^n(t) \neq 0 \text{ となるのは、} t_{L-1} < t < t_{n+L}$$

$$\vdots$$

$$N_{n+L-1}^n(t) \neq 0 \text{ となるのは、} t_{n+L-1} < t < t_{2n+L}$$

ゆえに、 $t$  の範囲は、 $t_n \leq t \leq t_{n+L}$  である。

**【例】** 3 コのセグメントから構成される 3 次 B スプライン曲線

コントロールポイント :  $\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3, \mathbf{P}_4, \mathbf{P}_5$

パラメータ  $t$  の範囲 :  $t_3 \leq t \leq t_6$

### 3.3.5 パラメトリック曲面

曲面のパラメトリック表現 : 3 次元空間内の曲面上の点の座標値  $x, y, z$  を、2 つのパラメータ  $u, v$  の関数として表す。

## [1] ベジエ曲面

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m \mathbf{P}_{i,j} B_i^n(u) B_j^m(v)$$

ここで、 $\mathbf{P}_{i,j}$ ,  $B_i^n(u)B_j^m(v)$  は、それぞれ、 $n$  次と  $m$  次のバーンスタイン基底関数である。

## [2] B スプライン曲面

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m \mathbf{P}_{i,j} N_i^n(u) N_j^m(v)$$

## 3.4 ポリゴン曲面

ポリゴン (polygon) : 多角形

コンピュータの高性能化によって、形状設計の段階から曲面を多数の細かいポリゴン (多角形) として扱うことができるようになった。

## 3.4.1 ポリゴン曲面

ポリゴンの集合として表現された曲面

利点 : 曲面を繋げる際に、複雑な計算が要らない。

## 3.4.2 細分割曲面

ある多面体からそれぞれの面を再帰的に分割し、形状をなめらかにすることで得られる曲面

## 3.4.3 平滑化処理

計測機器で実際の物体の形状を測定する際に生じるノイズを除去する方法

### 3.4.4 詳細度制御

コンピュータで扱うポリゴン数を調整して形状の詳細度を調整する手法

### 3.4.5 パラメータ化

ポリゴン曲面に2次元画像を貼り付ける場合に、曲面上の位置と画像上の位置の対応関係を定めなければならない。このとき、ポリゴン曲面上の各点をパラメータ  $u, v$  と対応付ける必要がある。このような対応関係を構築することを**パラメータ化**とよぶ。

## 3.5 そのほかの形状生成手法

### 3.5.1 ボクセル

小さな立方体である**ボクセル**を格子状に配置して3次元物体を表現する手法  
ボクセル (voxel) : volume pixel

### 3.5.2 八分木

正方形の各辺を2分割：四分木  
立方体の各辺を2分割：八分木  
立方体を8つの小さな立方体に再帰的に分割して物体の形状を表現する手法

### 3.5.3 フラクタル

山や入り組んだ海岸線：部分を拡大すると全体に一致する。

↓

自己相似性 (フラクタル (Fractal))

Fractal : マンデルブロの造語 (1975)

#### コッホ曲線のフラクタル次元

元の図形を3倍に拡張した図形は、元の図形4コからできている。

$$4 = 3^D$$

だから、

$$D = \frac{\ln 4}{\ln 3} = 1.26185 \dots$$

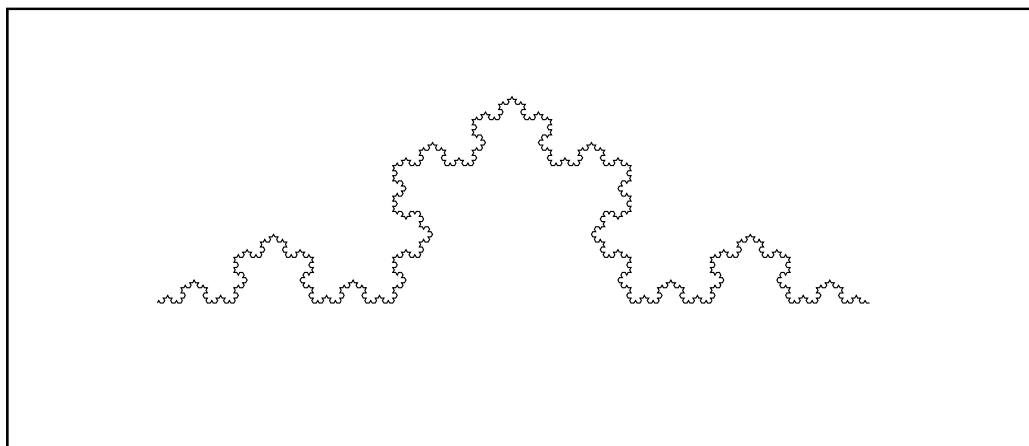


図 3.1: コッホ曲線