

コンピュータグラフィックス

試験によく出るまとめ

佐藤証 西9-613

akashi.satoh@uec.ac.jp

2次元座標變換



同次座標

- 平行移動, 拡大・縮小, 回転等の基本変換は組み合わせてることが多く, 全ての変換の統一的な述が望ましい
- 座標の列ベクトル $(x, y)^T$ の行列演算で表そうとするとき, 拡大・縮小と回転は行列との積, 平行移動はベクトルの和になってしまう

拡大・縮小
$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

回転
$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

平行移動
$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

同次座標

- 実数 $w \neq 0$ を用いて (wx, wy, w) と表す座標
 - 通常座標 $(2,3)$ の同次座標 $(2,3,1)$ と $(4,6,2)$ は同じ
 - 簡単のため普通は $w=1$ を用いる
- 全ての幾何学的変換を行列の積で表す

拡大・縮小

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = S(s_x, s_y) \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

回転

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = R(\theta) \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

平行移動

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = T(t_x, t_y) \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

鏡映

x軸に関する鏡映

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

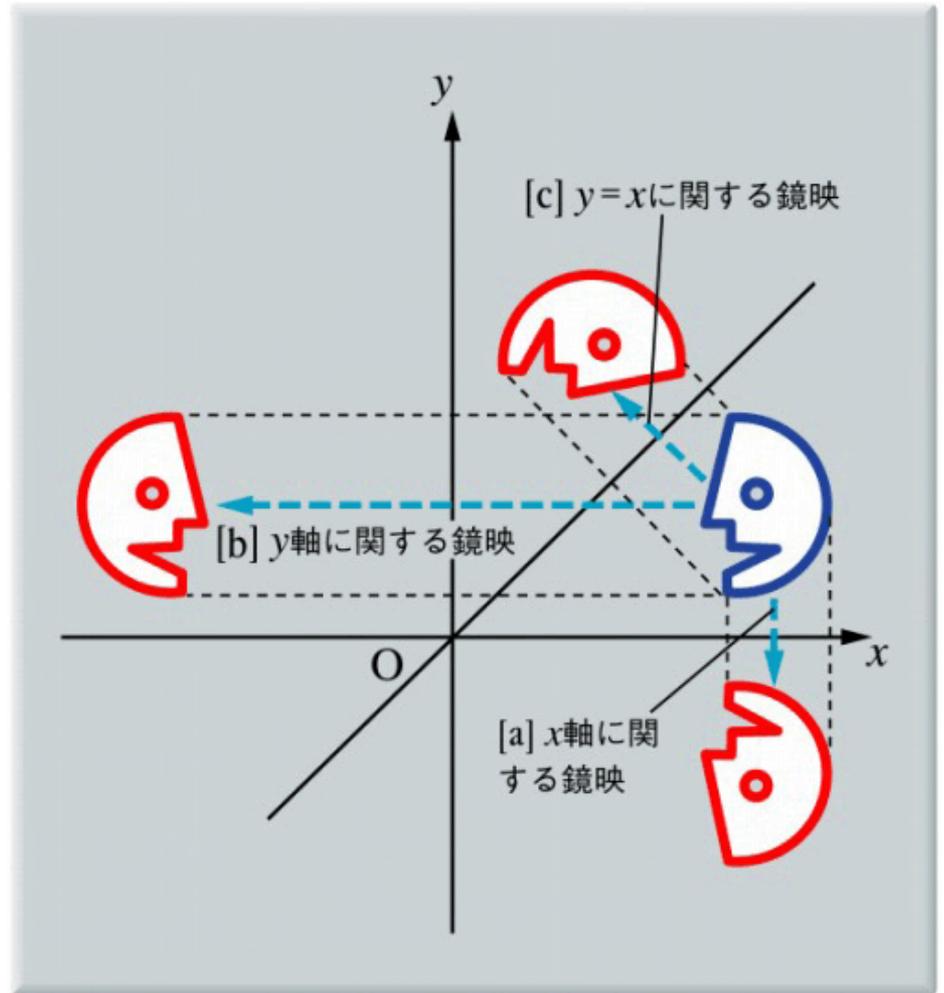
y軸に関する鏡映

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

直線 $x = y$ に関する鏡映

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

x座標とy座標の交換



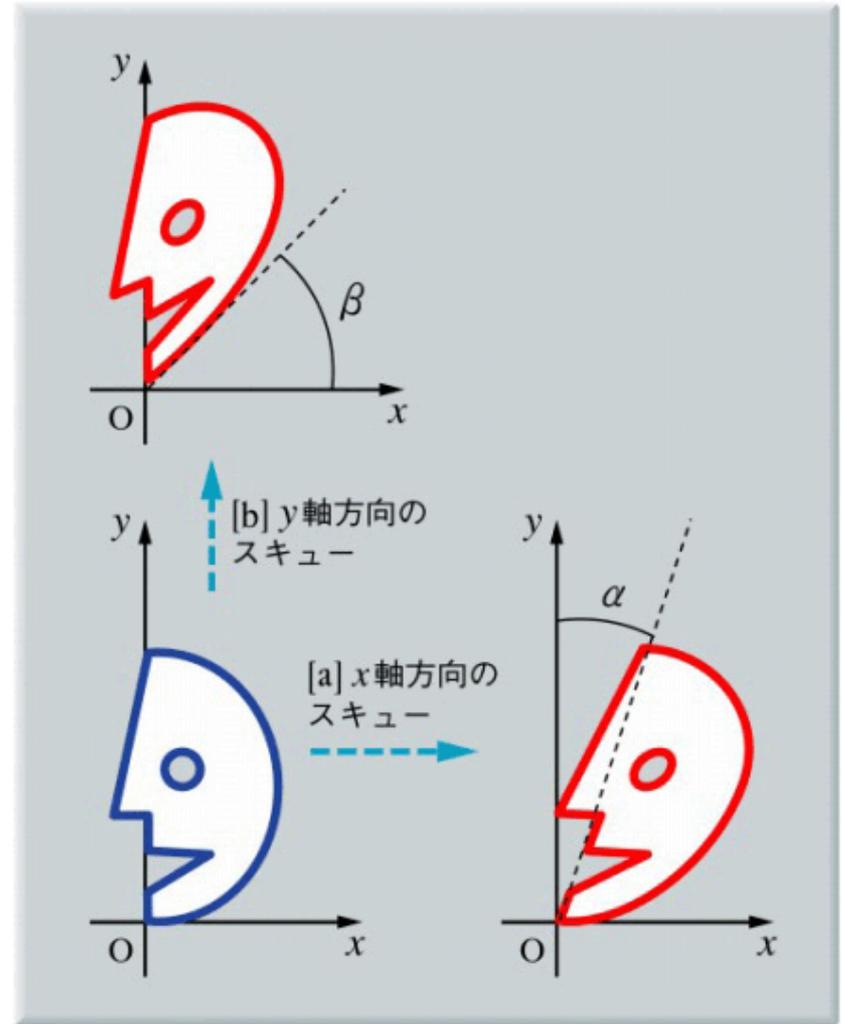
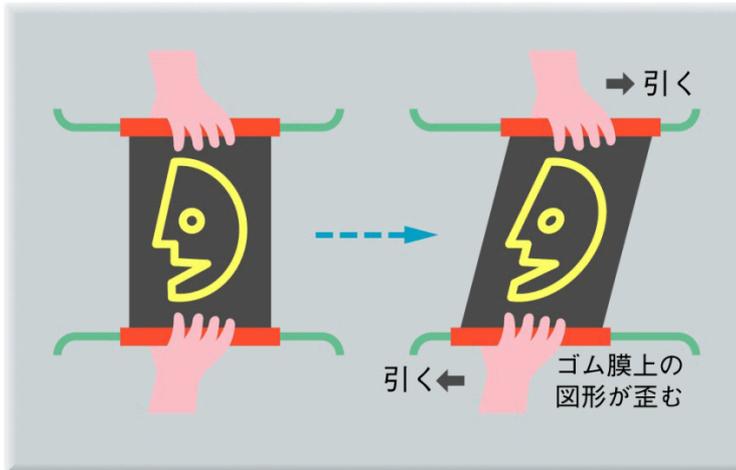
スキュー(せん断)

y方向のスキュー

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ \tan \beta & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

x方向のスキュー

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & \tan \alpha & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

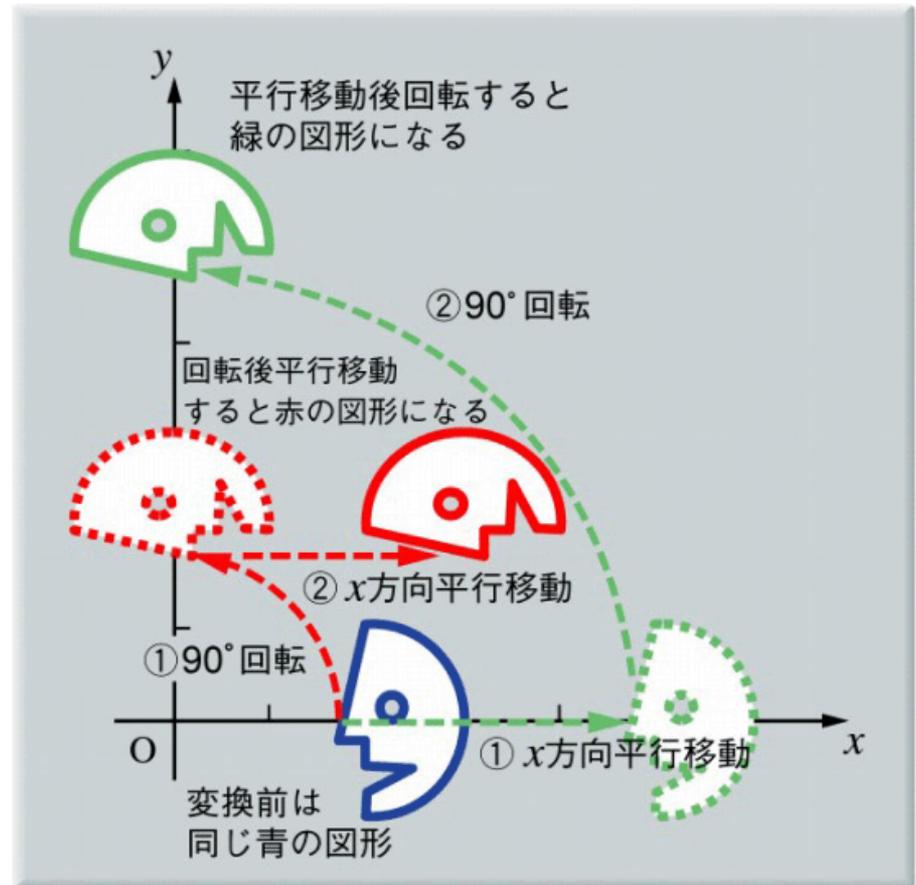


2次元座標系における合成変換

- 合成変換では変換の順序を入れ替えると一般には同じ変換にならない ($A_1A_2 \neq A_2A_1$) ことに注意

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = R(\theta)T(x_0, y_0) \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = T(x_0, y_0)R(\theta) \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

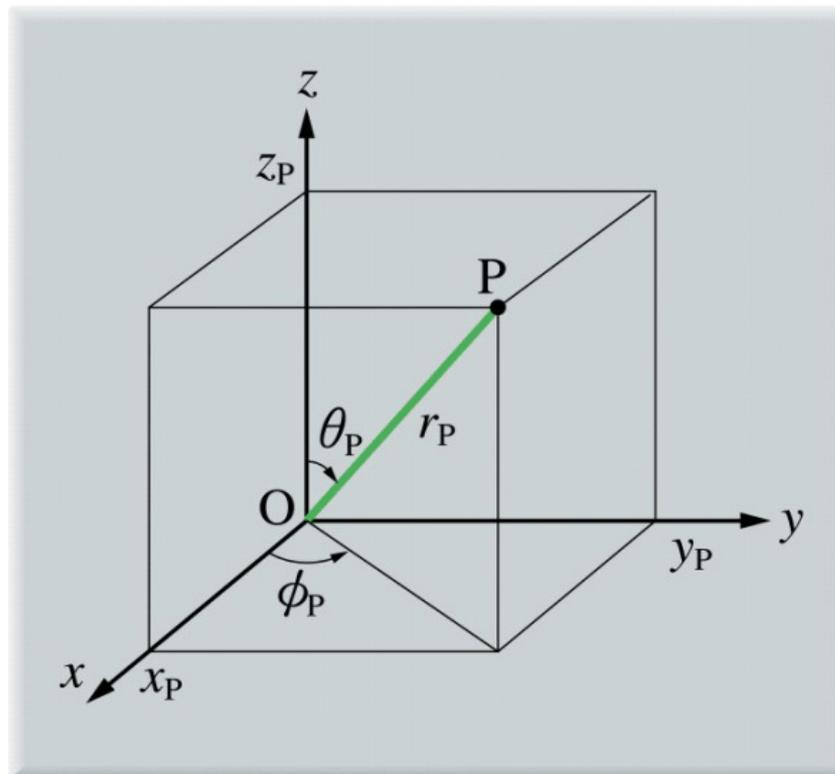


3次元変換と投影



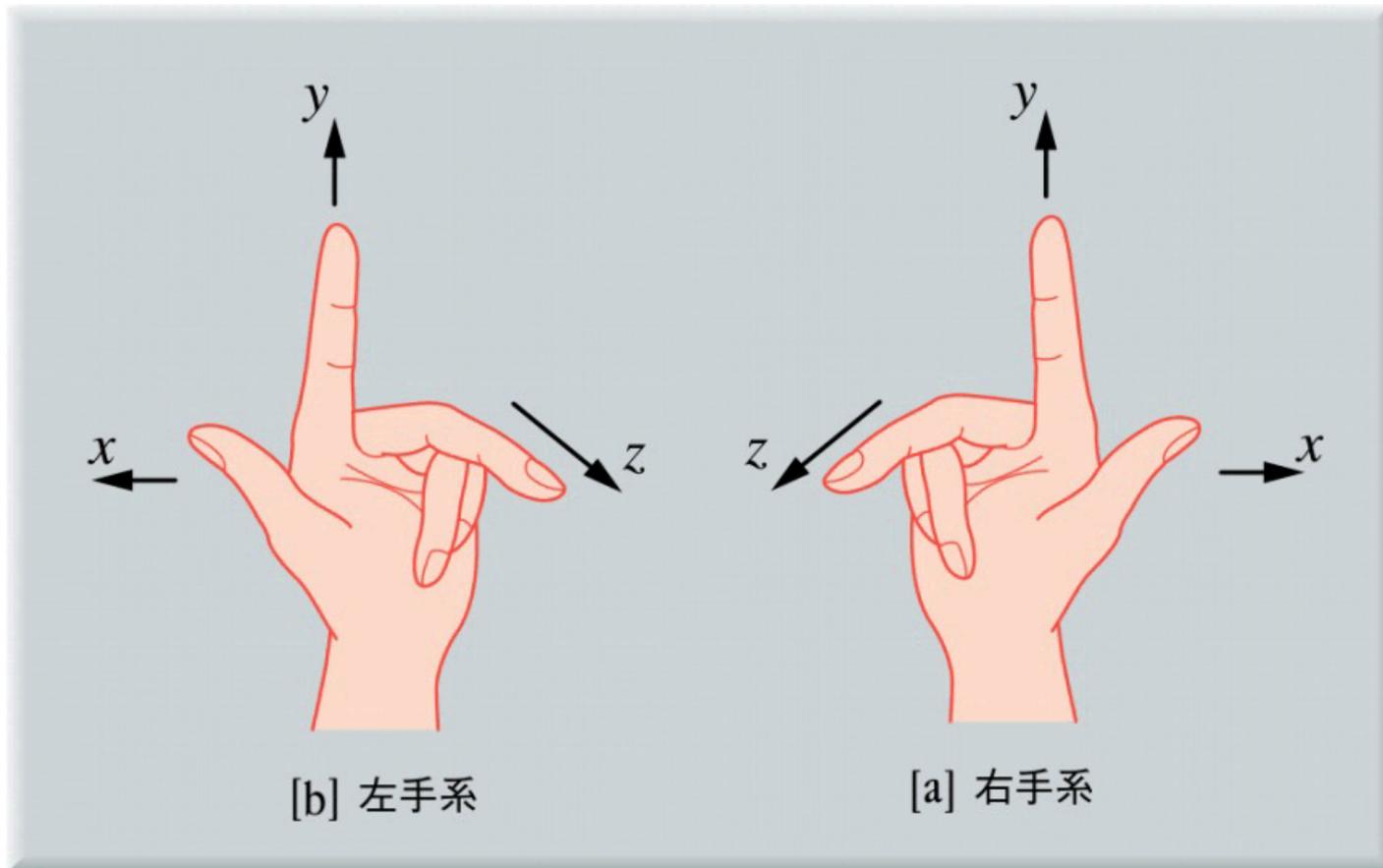
3次元直行座標系と極座標系

- 3次元直行座標系で点Pの座標は x, y, z の3軸により (x_P, y_P, z_P) と一意に表現される
- 極座標系では原点Oから点Pまでの距離 r_P と z 軸からの角度 θ_P 、 x 軸からの角度 ϕ_P で (r_P, θ_P, ϕ_P) と表現される



右手系と左手系

- x 軸, y 軸, z 軸の向きによって右手系と左手系に分けられる

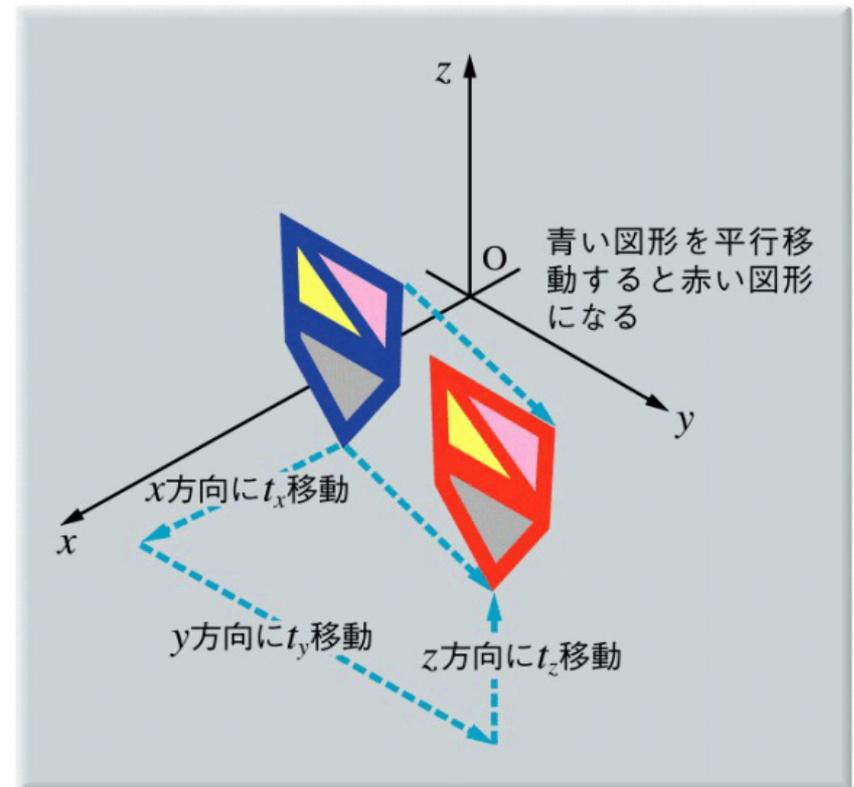


同次座標と3次元の基本変換

- 2次元と同様に、実数 $w \neq 0$ を用いて座標を (wx, wy, wz, w) と表すことで全ての幾何学変換を行列の積で表す
 - 簡単のため普通は $w=1$ を用いる

平行移動

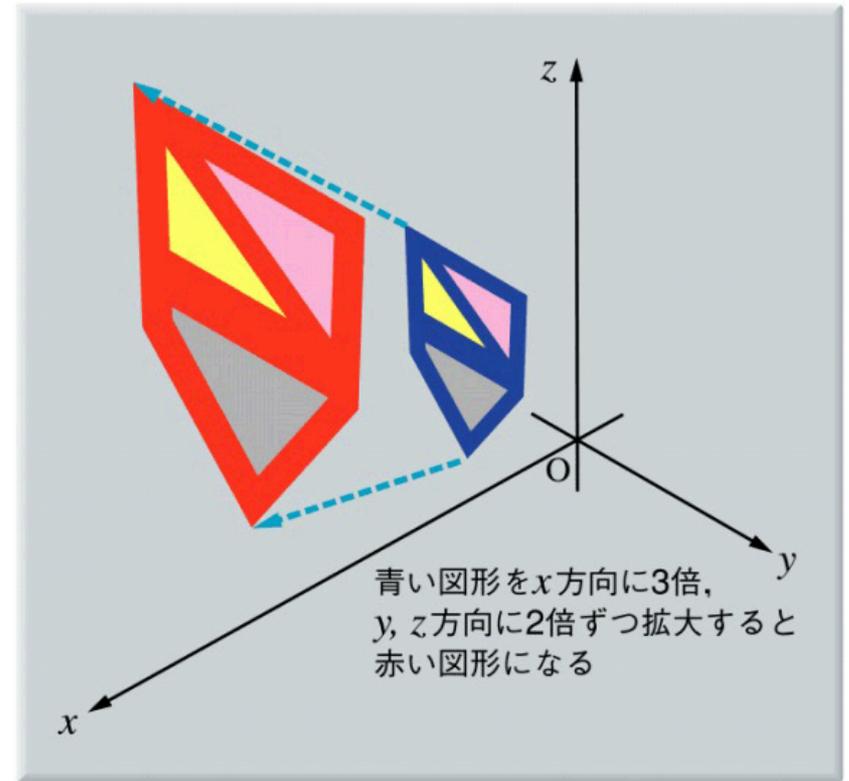
$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$
$$= T(t_x, t_y, t_z) \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$



拡大・縮小

- x, y, z 軸方向にそれぞれ s_x, s_y, s_z と拡大または縮小する変換

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$
$$= S(s_x, s_y, s_z) \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$



回転

- x, y, z の回転軸によって変換が異なる

x軸まわりの回転

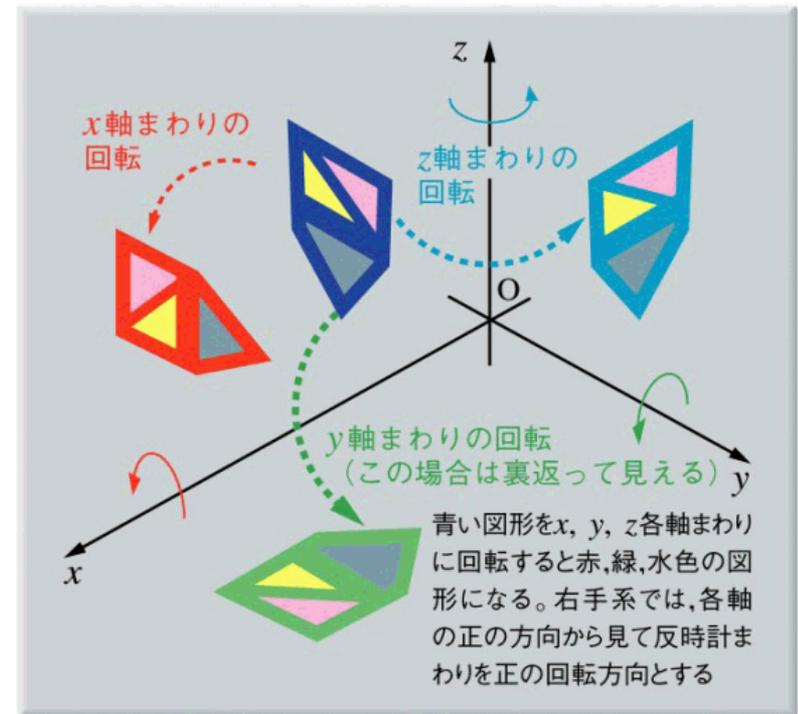
$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = R_x(\theta) \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

y軸まわりの回転

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = R_y(\theta) \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

z軸まわりの回転

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = R_z(\theta) \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

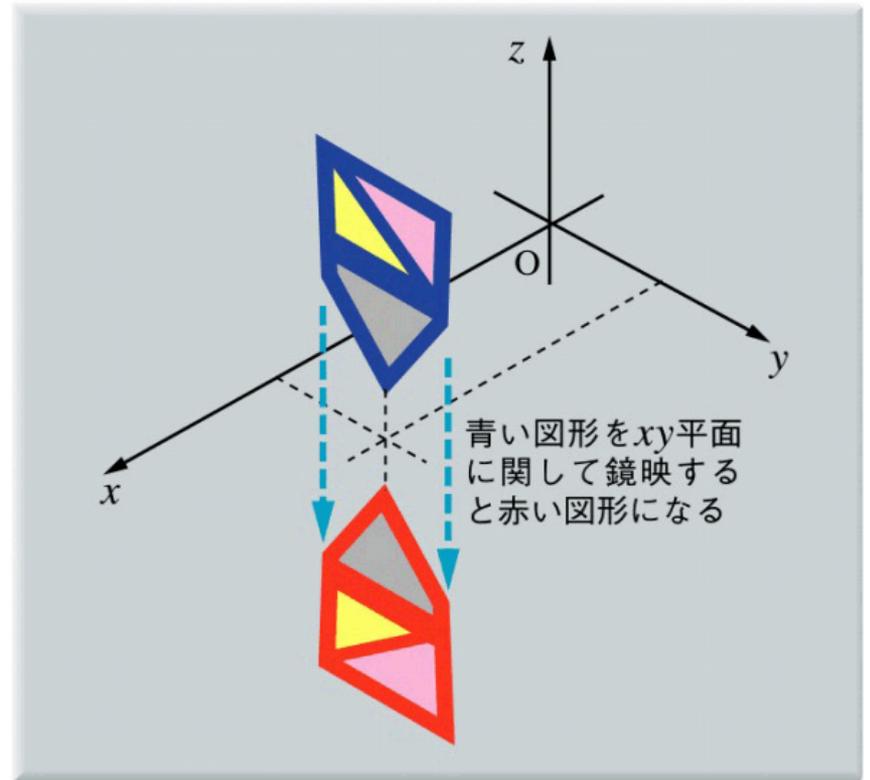


鏡映

- 3次元の鏡映は直線ではなく、平面に関して対称な位置への移動

xy平面に関する鏡映

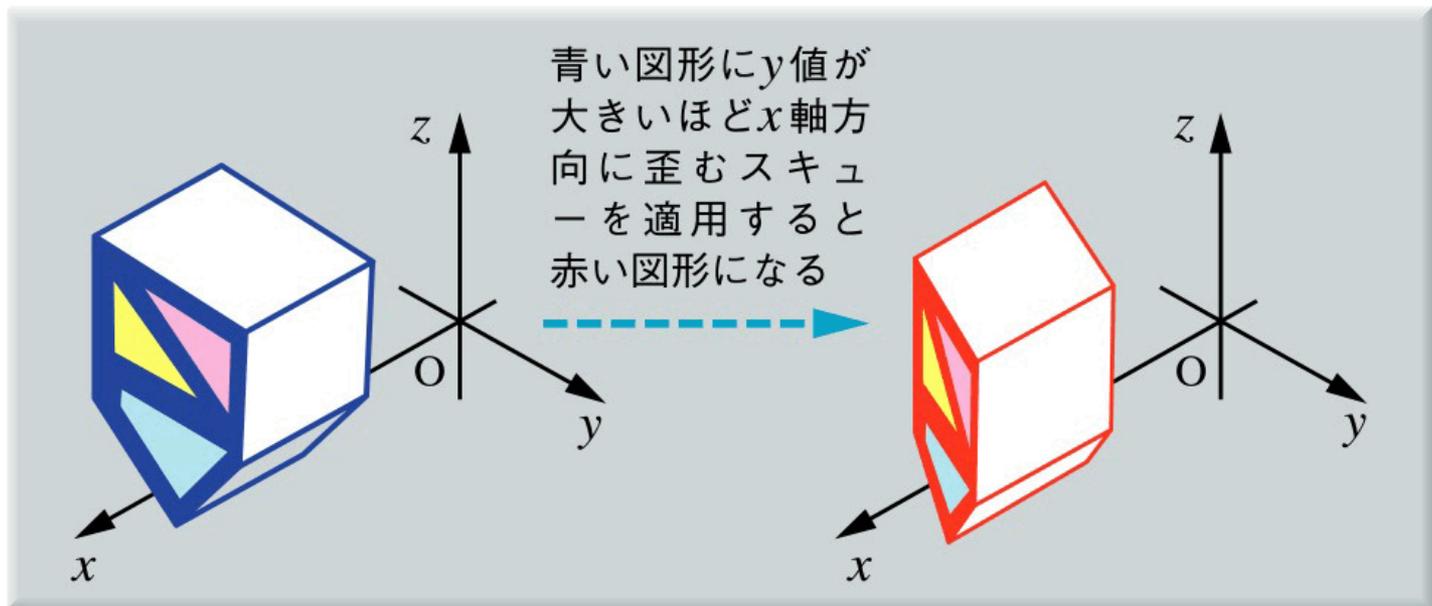
$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$



スキュー

- 一般に、行列の*i*行*j*列の要素 $a_{ij}(i \neq j)$ が0でない場合、その要素は第*j*軸の座標値が大きいほど、第*i*軸方向に大きく歪む (座標値の変化量大きい)

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & a_{12} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$



投影

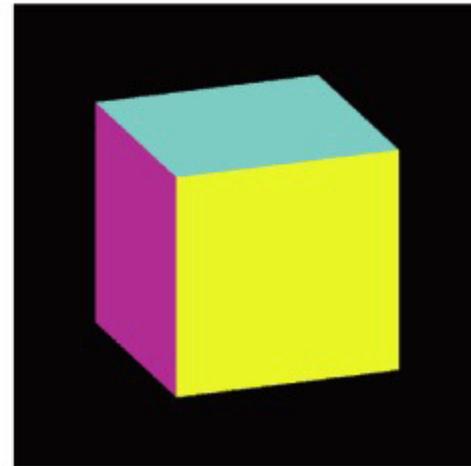
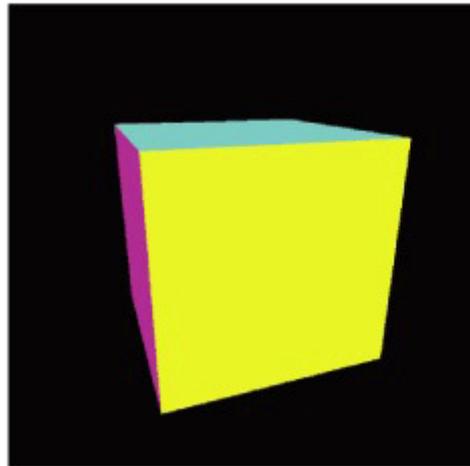
- 3次元図形をディスプレイモニタの画面や紙等の二次元平面上に表示するために、2次元図形に変換する処理

透視投影

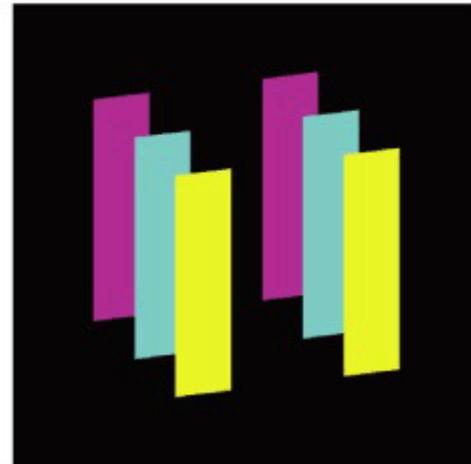
平行投影

エッシャー

立方体

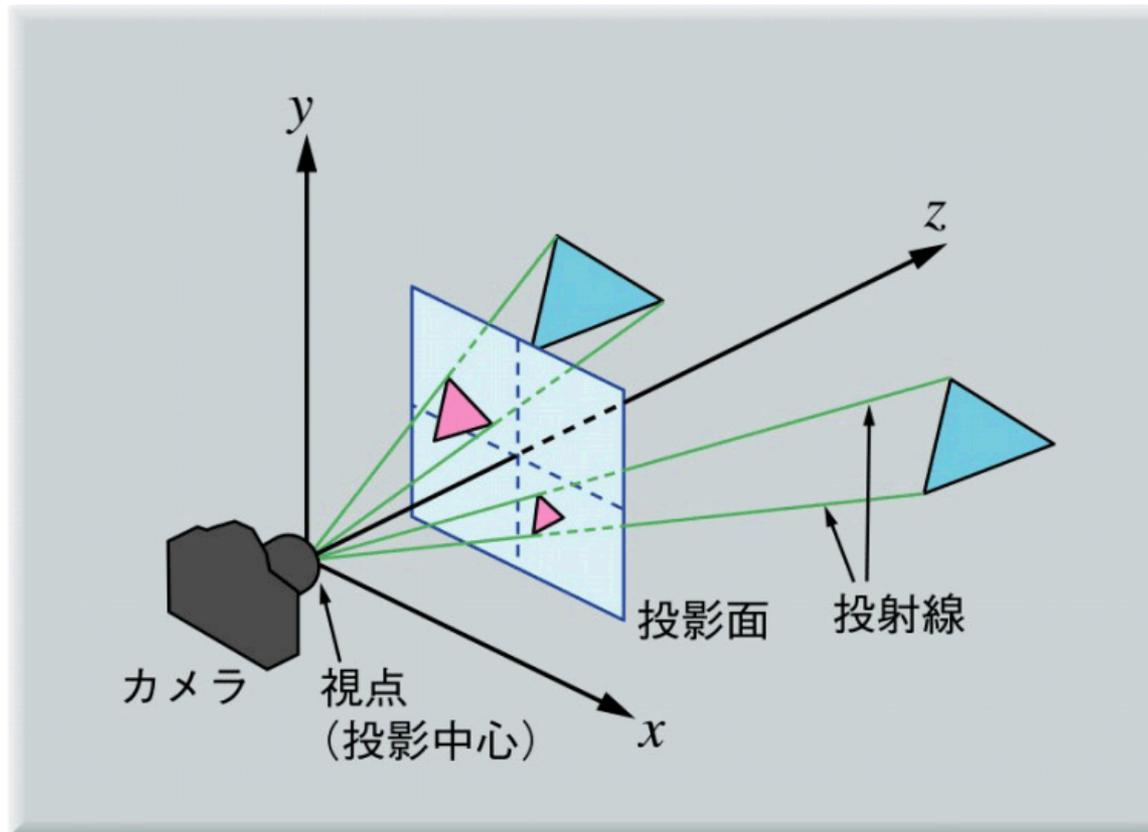


ポリゴン



透視投影

- 3次元空間に視点(投影中心)と図形を投影するための投影面を置く
- 3次元図形の各点から視点に向かって投射線を引くと、投影面との交点の集まりとして投射図形が得られる

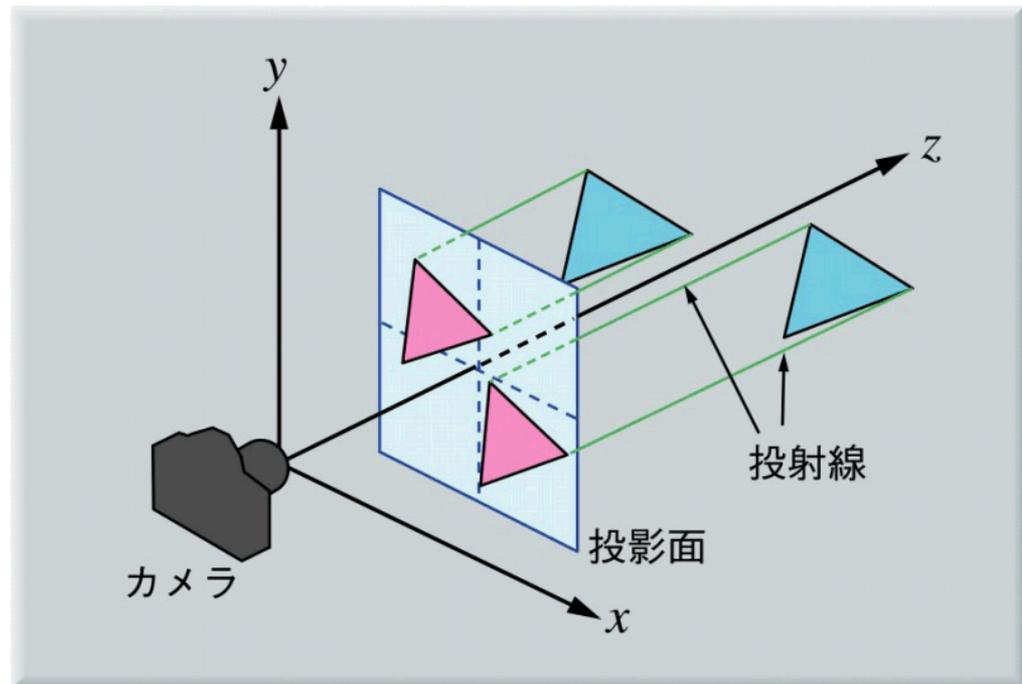


平行投影

- 3次元図形の各点からの当社線を平行に投影面におろす
- 視点を無限遠点に置いた投影
- 遠くのものと同近のものと同じ大きさで描画
- 写実的な画像の作成や映画・ゲーム等には利用されない
- 平行な線は投影面上でも平行で歪みがなく、形を正確に描画する設計図やグラフに利用される

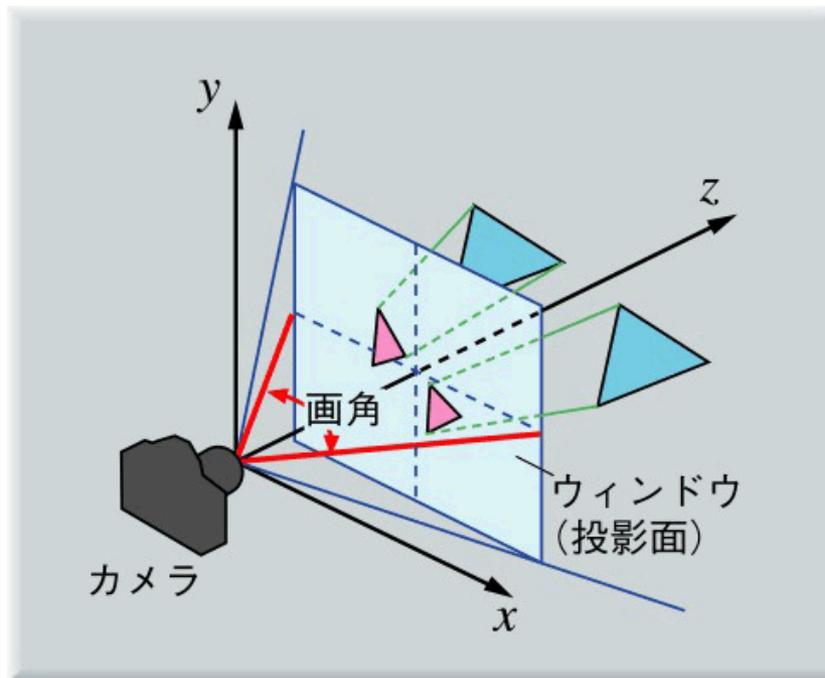
投影面 ($z=k$: 任意)
への変換

$$\begin{cases} x' = x \\ y' = y \end{cases}$$

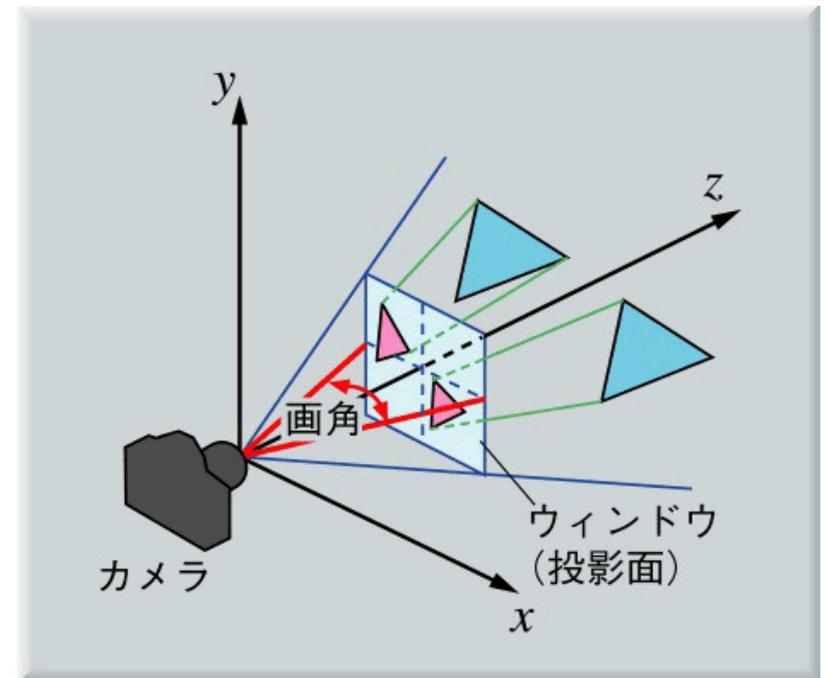


ビューボリュームと投影

- 実際の計算では、投影面上に長方形のウィンドウを考え、この範囲に投影される図形だけを描く
- 視点からウィンドウをカバーする角度を画角とよぶ
- 画角が大きいと物体が小さく写る広角レンズの効果
- 画角が小さいと物体が大きく写る望遠レンズの効果



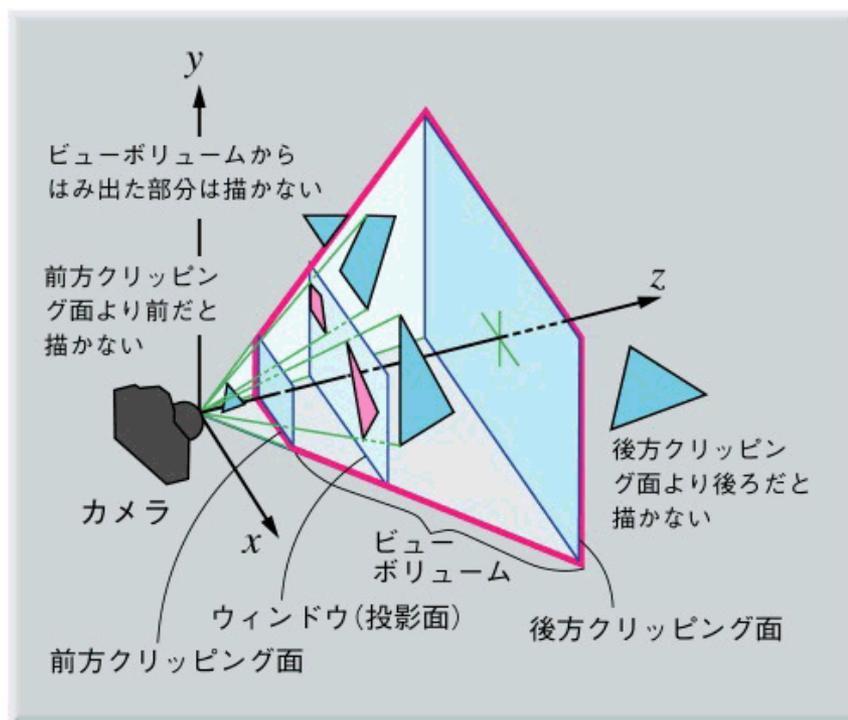
[a] 画角が大きい場合



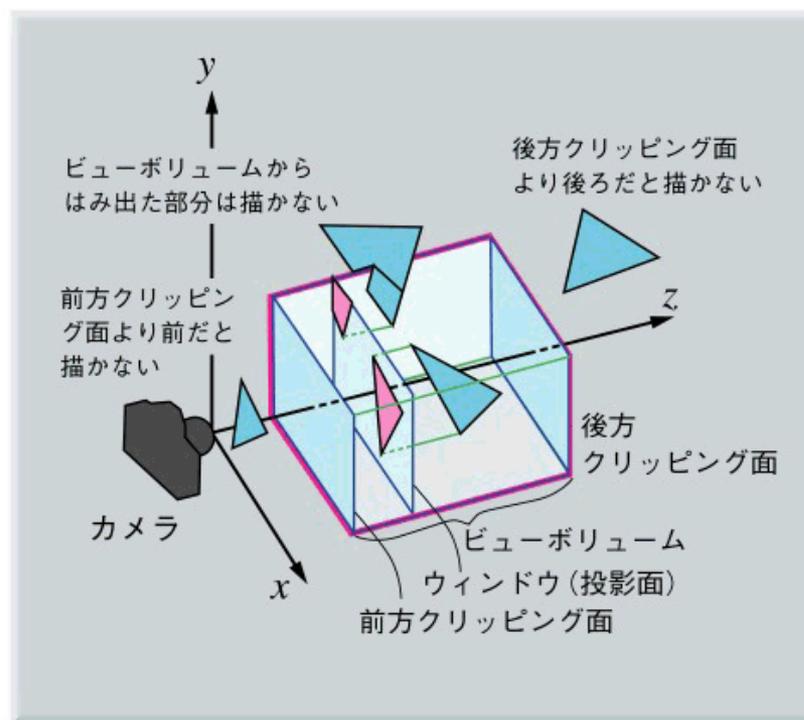
[b] 画角が小さい場合

ビューボリュームとクリッピング

- 投影面に平行な前後に有限の範囲のみ計算を行う
 - 前方クリッピング面・後方クリッピング面
- 図形が描かれる範囲をビューボリュームと呼ぶ
- 透視投影では四角錐台、平行投影では長方形となる



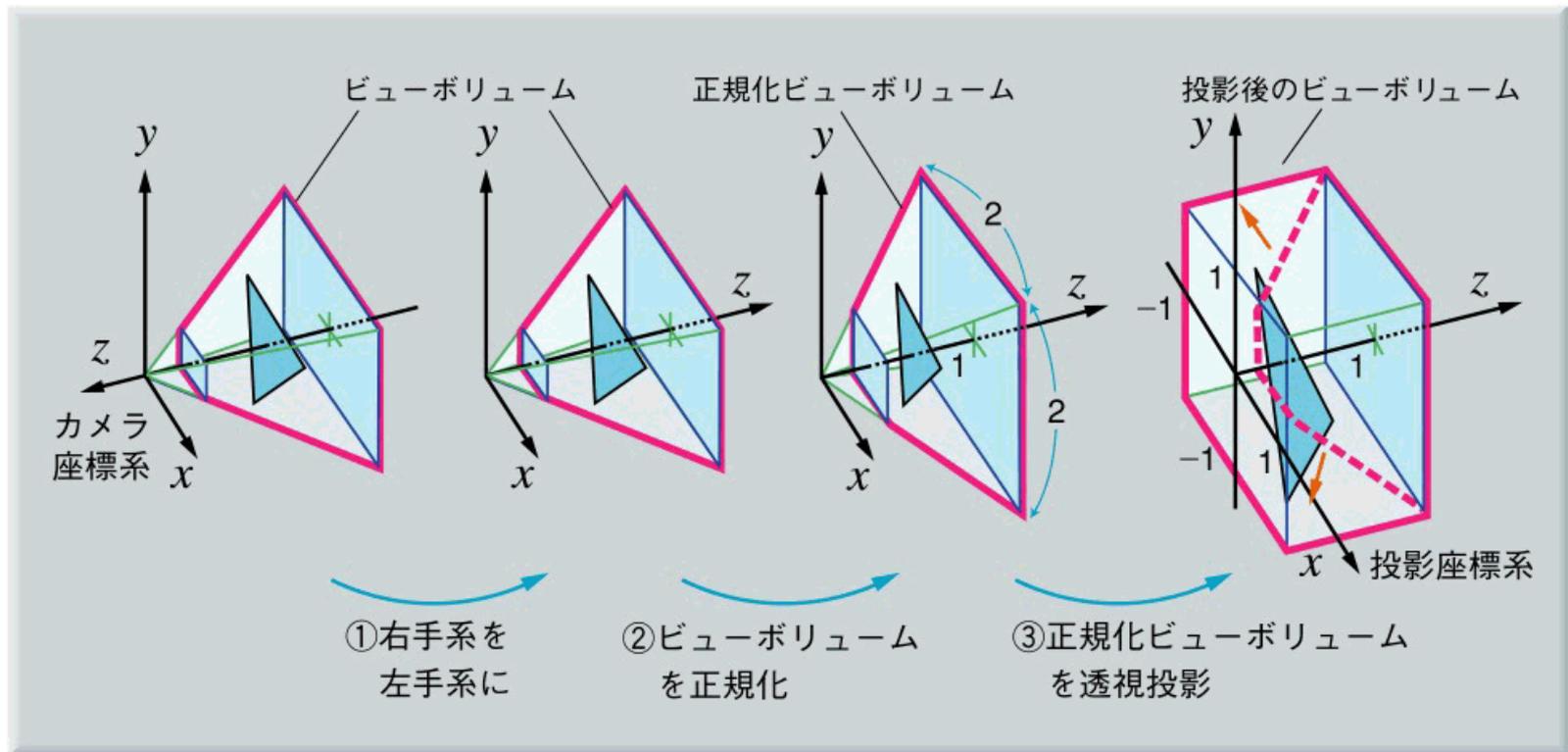
[a] 透視投影の場合



[b] 平行投影の場合

透視投影の計算法

- ① カメラ座標系を右手系から左手系に変換
 - xy平面に対する鏡映変換 $S(1,1,-1)$
- ② ビューボリュームを正規化ビューボリュームに変換
 - 後方リッピング面が $z=1$ に、 $z=1$ でのビューボリュームの範囲が $-1 \leq x \leq 1$ 、 $-1 \leq y \leq 1$ の正方形になるよう拡大・縮小



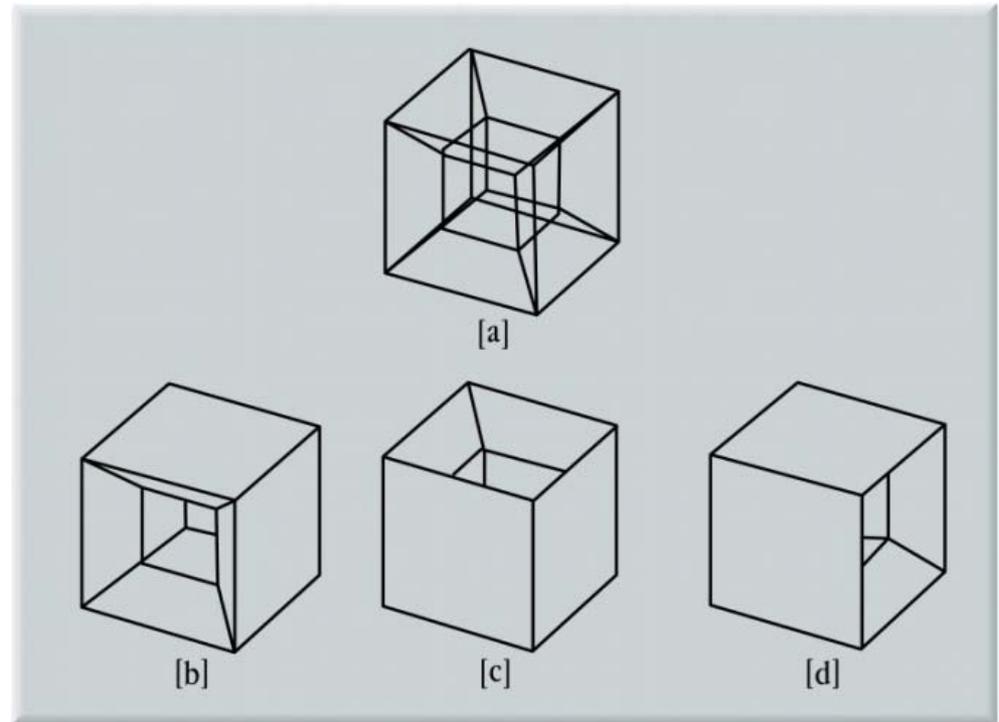
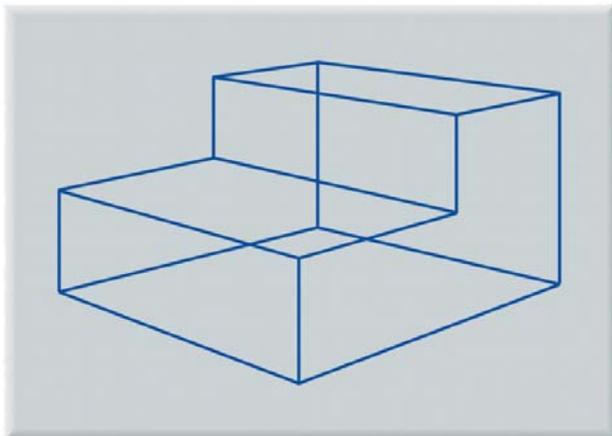
モデリング1

－ 直線・平面 －



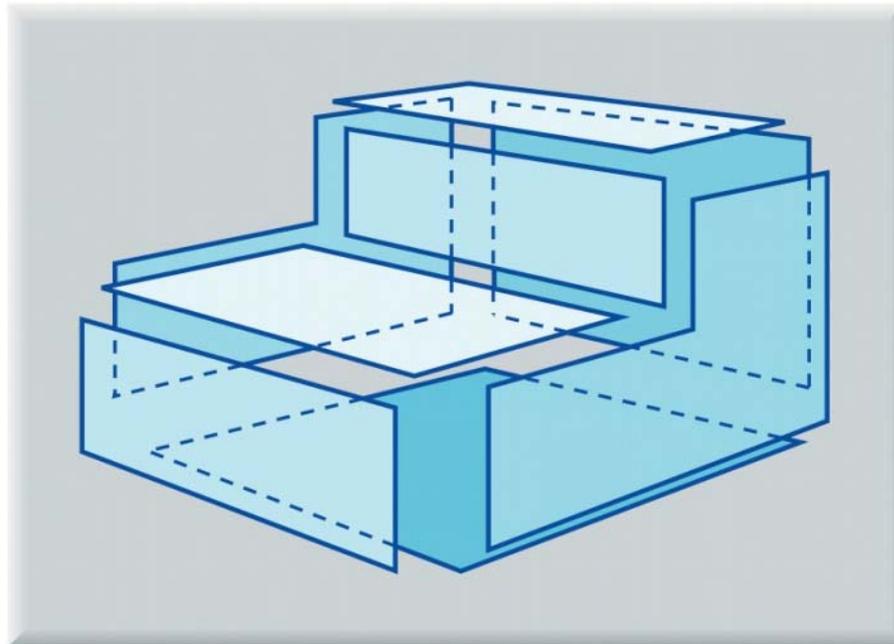
ワイヤースケッチモデル

- 稜線だけを記録することで立体を表現したモデル
- 頂点の接続関係を簡単に表現でき処理が高速
- どこが立体の内部化不明で、立体同士の干渉計算や陰線消去・陰面消去ができない。



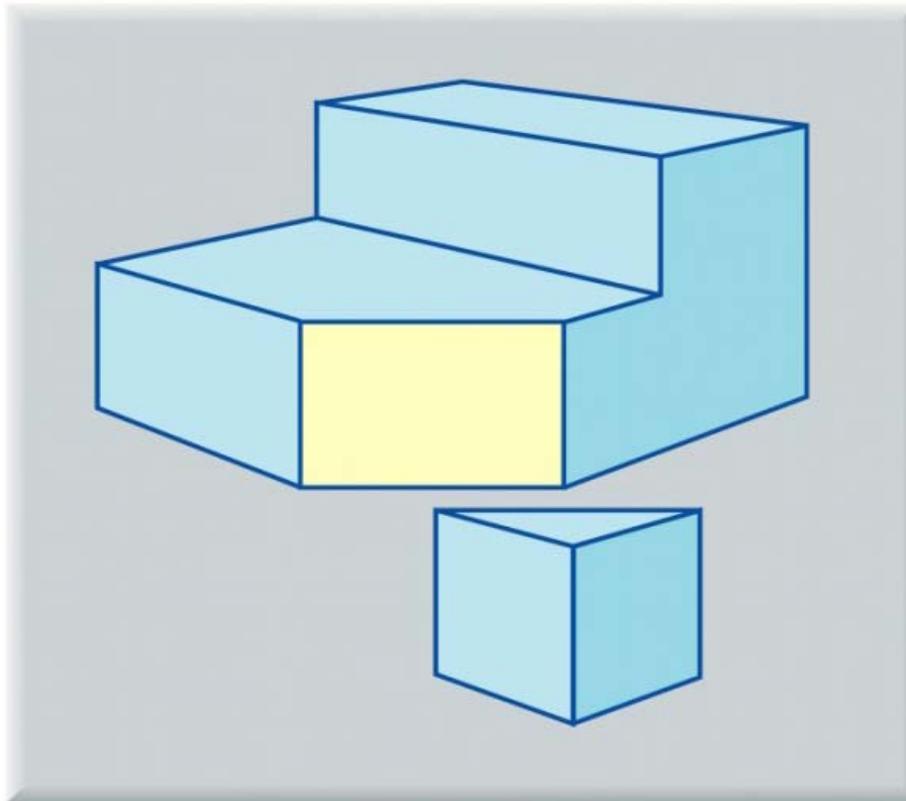
サーフェスモデル

- ワイヤーフレームモデルに面の情報を付加
- 陰線消去、陰面消去、面の陰影が表現できる
- 物体内部の情報を持たないため、体積計算や物体同志の集合計算はできない



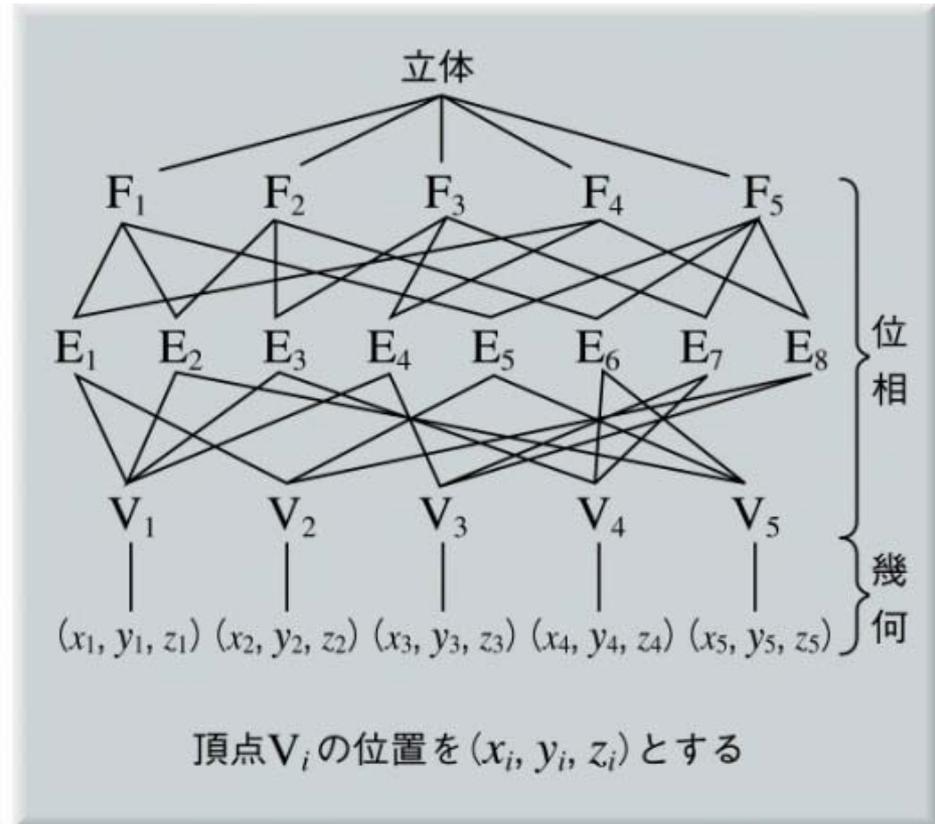
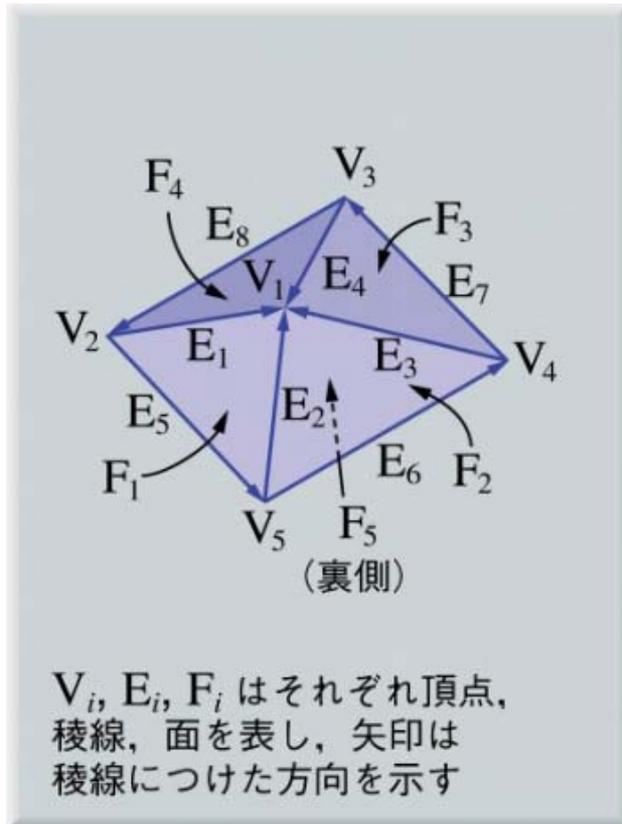
ソリッドモデル

- サーフェースモデルに物体の内部用法を付加
- 物体の和・積・差の集合演算や、体積計算、物体同志の厳密な干渉計算が可能
- 有限要素法による様々な解析が可能



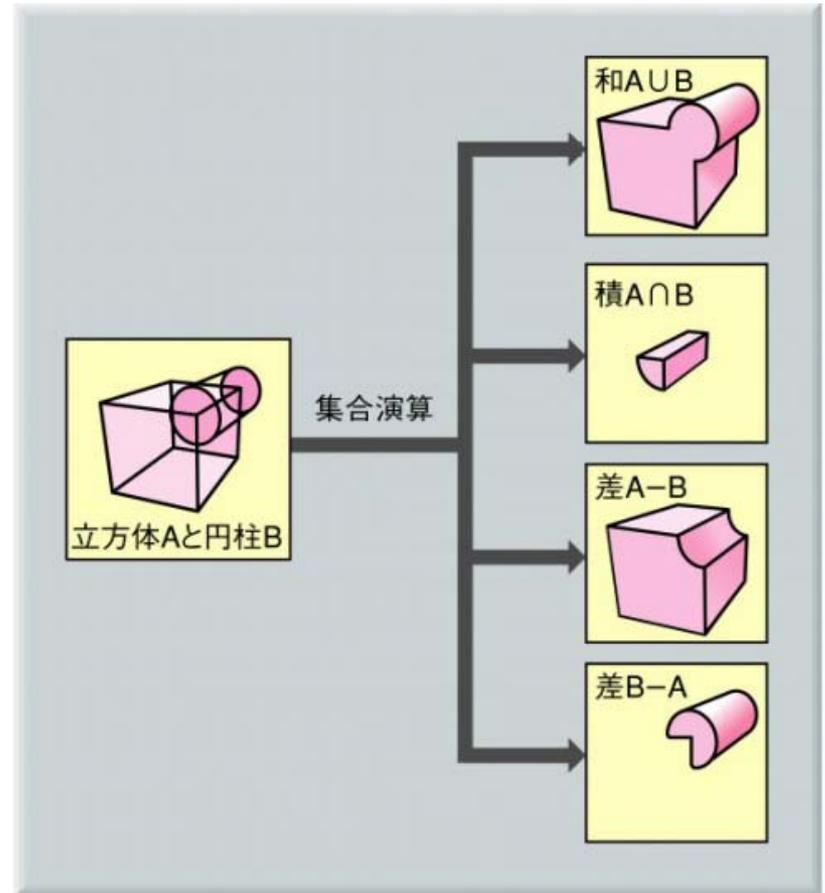
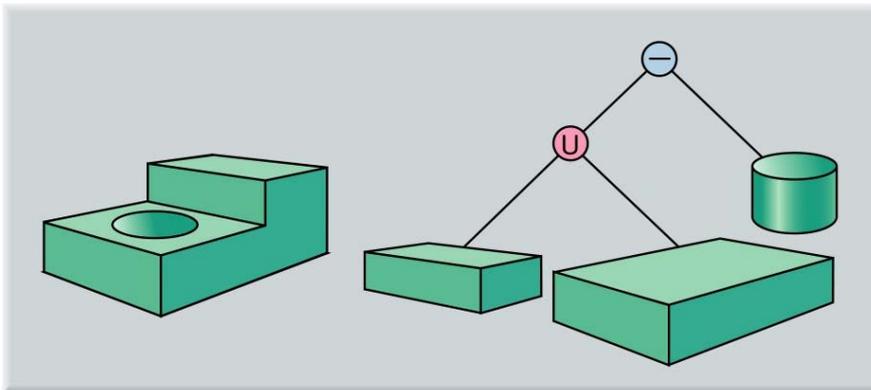
境界表現

- 境界表現はB-repとも呼ばれ、頂点、稜線、面のデータとその接続関係をグラフで保持して立体を表現する
- 要素の接続関係を表す部分を位相、頂点の座標値の部分を幾何と呼ぶ



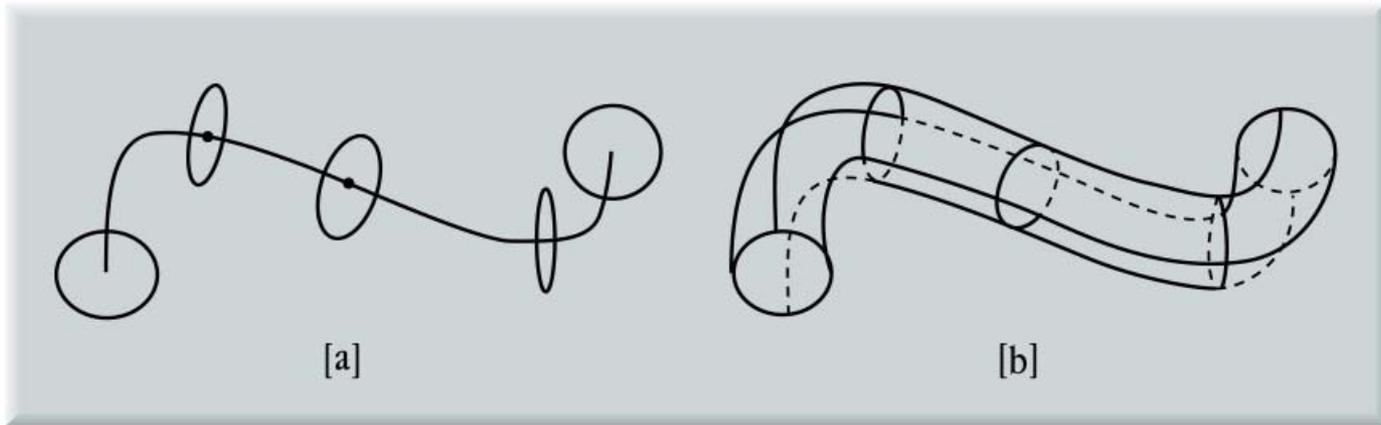
SCG表現

- SCG (Constructive Solid Geometry)とはプリミティブと呼ぶ基本立体を集合演算の木構造によって表現
 - 基本立体：直方体、円柱、多面体、錐体、級、半空間
 - 集合演算：和、積、差
- 同じ形を表すためのCSG表現は一つとは限らない

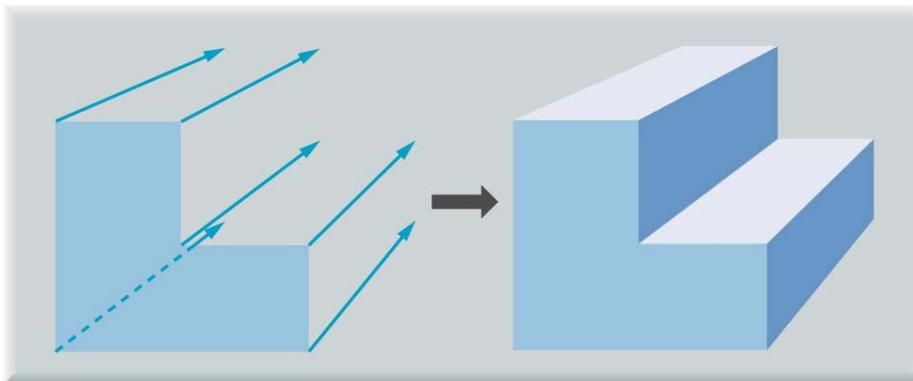


スイープ表現

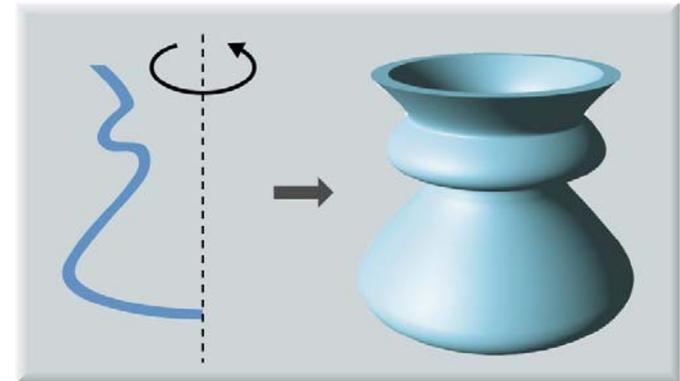
- 立体の断面の2次元図形を軌道に沿って移動したときの軌跡として形状を表現



スイープ表現による立体の生成



平行移動スイープ



回転スイープ

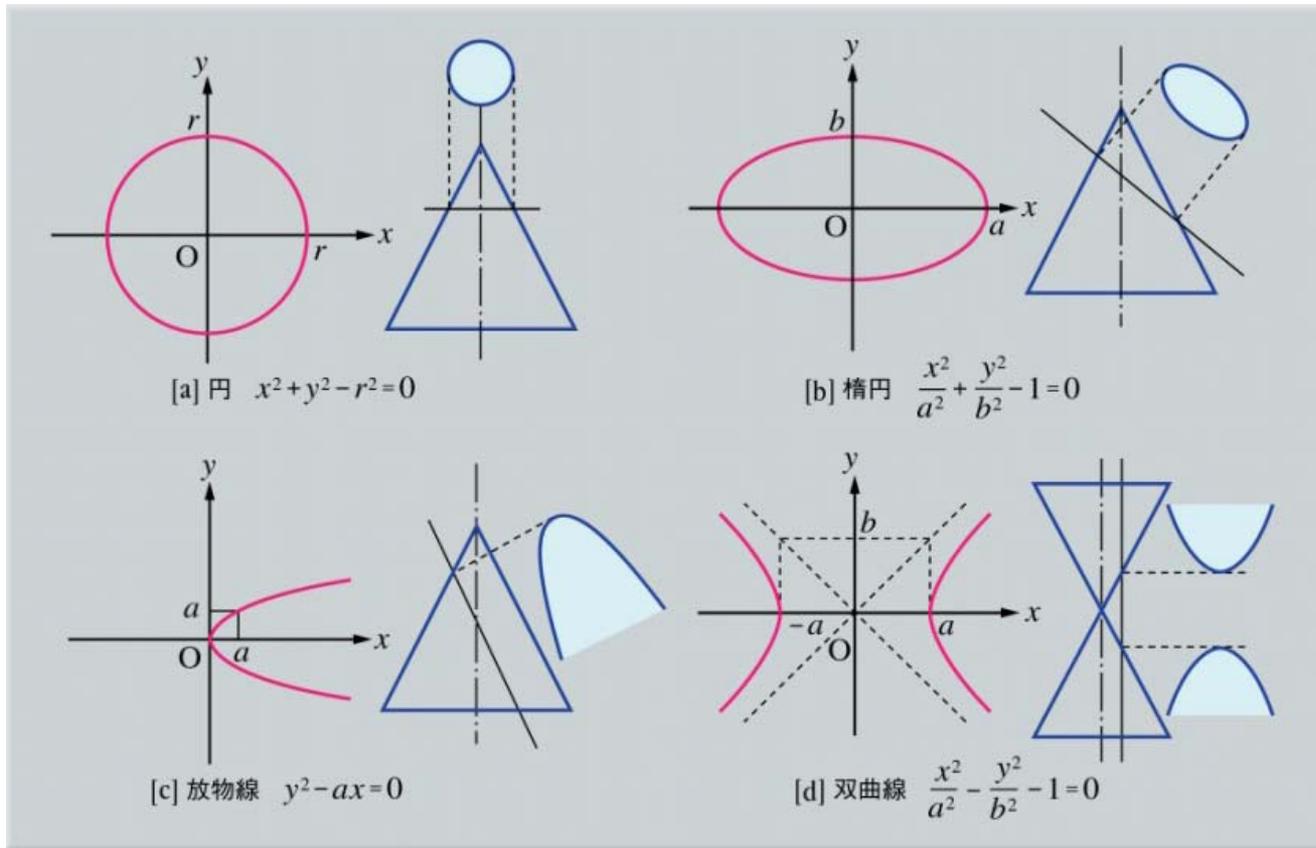
モデリング2

－ 曲線・曲面 －



2次曲線

- 2次多項式を用いた陰関数形式による代数曲線
 - $ax^2+by^2+c+2dxy+2ex+2fy=0$
- 楕円、放物線、双曲線の3種類に分類される



円錐の断面と2次曲線

パラメトリック曲線

- 各座標がパラメータ t の陽関数 $C=F(t)$ と表現された曲線
 - 多項式曲線： F が t の多項式
 - 有理式曲線： F が t の有理式
 - 複合曲線：複数の曲線の部分(セグメント)を一本に合わせたもの
- 曲線の性質
 - 凸包性：曲線が制御点の凸包内に存在
 - 変動減少性：曲線が制御点列の形状をなめらかな形にしたもの
 - 局所性：制御点を移動したとき曲線で変形する部分が限定される
 - 直線再現性：制御点が直線上にあれば得られる曲線が直線になる
 - アフィン不変性：曲線にアフィン変換を施したものと、アフィン変換を施した制御点から得られる曲線が一致
 - 2次曲線の再現性：円や楕円、双曲線を正確に表現できること
 - 射影不変性：曲線に射影変換を施した結果と、射影変換を施した制御点から得られる曲線が一致すること

パラメトリック曲線

曲線の性質

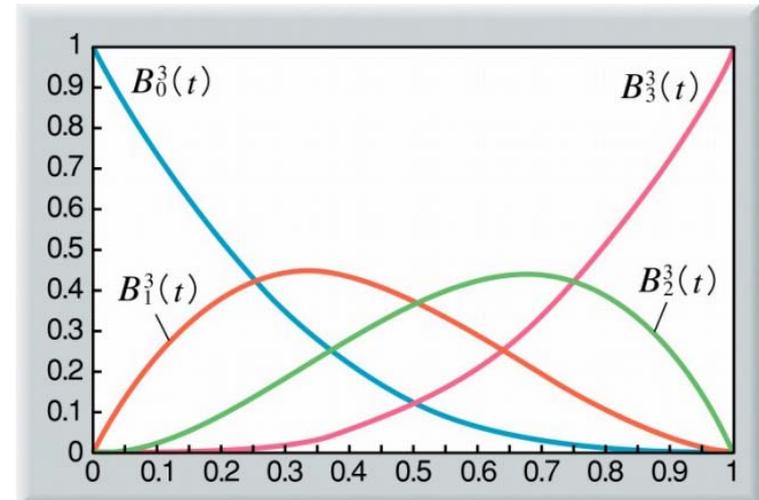
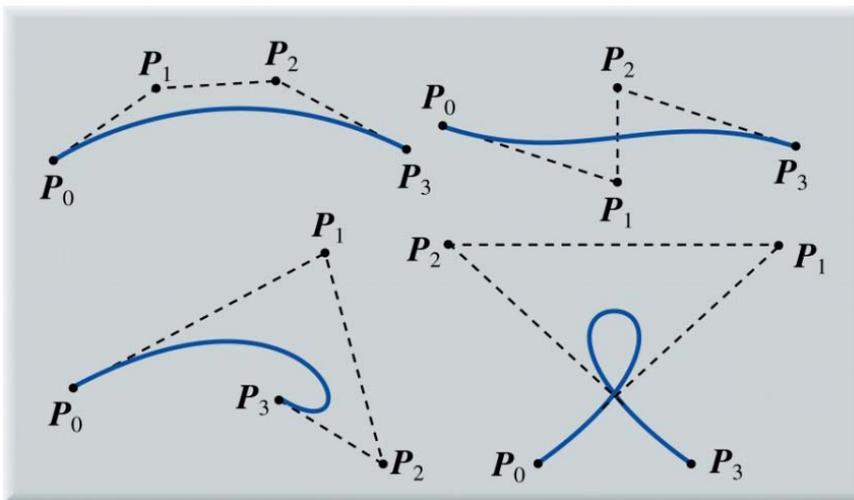
| 性質 \ 種類 | ベジエ曲線 | Bスプライン曲線 | 有理ベジエ曲線 | NURBS 曲線 |
|----------|-------|----------|---------|----------|
| 凸包性 | ○ | ○ | ○ | ○ |
| 変動減少性 | ○ | ○ | ○ | ○ |
| 局所性 | × | ○ | × | ○ |
| 直線再現性 | ○ | ○ | ○ | ○ |
| アフィン不変性 | ○ | ○ | ○ | ○ |
| 2次曲線の再現性 | × | × | ○ | ○ |
| 射影不変性 | × | × | ○ | ○ |

制御点を用いて形を定める曲線の特徴

| | 1セグメント | 複合曲線 |
|---------|---------|----------|
| 局所性 | なし | あり |
| 2次曲線再現性 | ベジエ曲線 | Bスプライン曲線 |
| なし | 有理ベジエ曲線 | NURBS曲線 |
| あり | | |

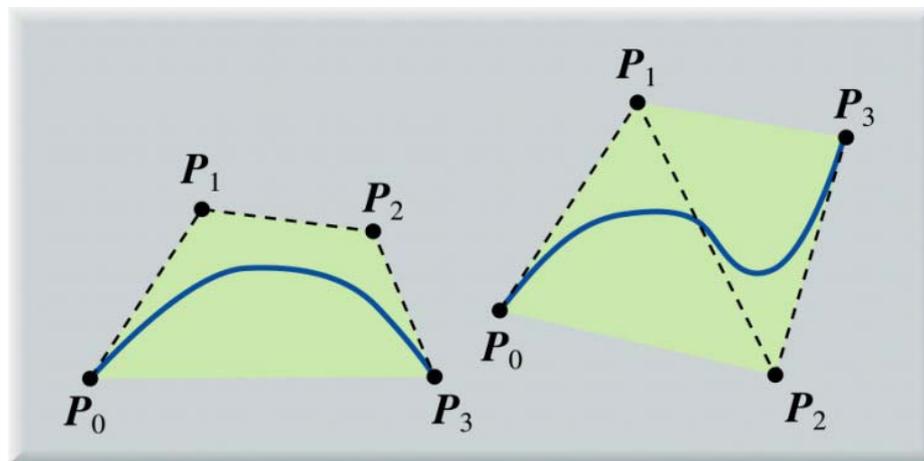
ベジエ曲線

- 複数の制御点 $\{P_i\}$ によって定義される
- 3次バーンスタイン基底関数 $B_i^3(t)$ をかけて足し合わせた形
- n 次のベジエ曲線は $n+1$ 個の制御点が必要
- CGでは3次ベジエ曲線が良く用いられる
 - $P(t) = P_0 B_0^3(t) + P_1 B_1^3(t) + P_2 B_2^3(t) + P_3 B_3^3(t)$, $0 \leq t \leq 1$
 - $B_0^3(t) = (1-t)^3$
 - $B_1^3(t) = 3t(1-t)^2$
 - $B_2^3(t) = 3t^2(1-t)$
 - $B_3^3(t) = t^3$



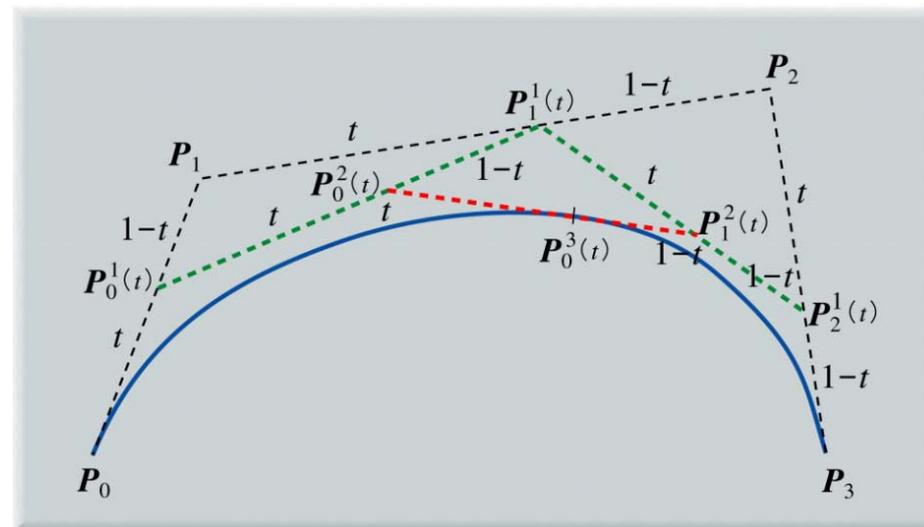
ベジェ曲線

- 制御点によって定義される凸包の中に曲線が完全に含まれる



ベジェ次曲線の凸包性

- ド・カステリヨのアルゴリズムによりベジェ曲線は新たな2つのベジェ曲線に分割できる



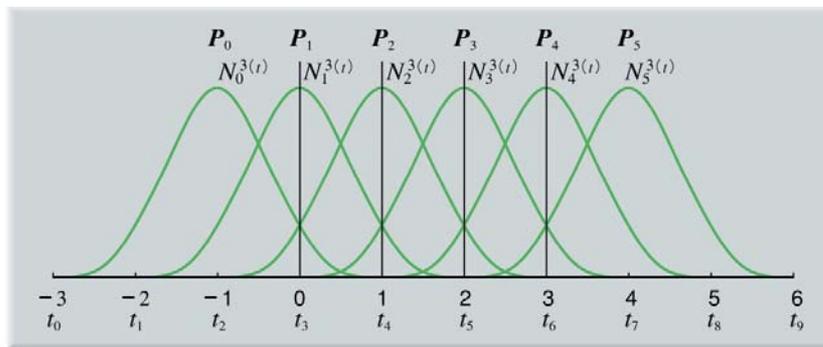
ド・カステリヨのアルゴリズム

Bスプライン曲線

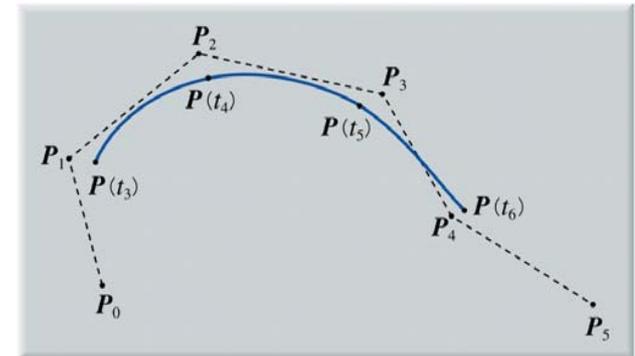
- 制御点 $\{P_i\}$ とノット列 $\{N_i\}$ で指定された接続点のパラメータにより複数の曲線(Bスプライン既定関数 N_i^n) を1本に接続
 - 一様Bスプライン曲線：ノット値が一定間隔のもの
 - 非一様Bスプライン曲線：ノット値が一定間隔でないもの
- L 個のセグメントから構成される n 次のBスプライン曲線は次式で定義される

$$P(t) = \sum_{i=0}^{n+L-1} P_i N_i^n(t), t_n \leq t \leq t_{n+L}$$

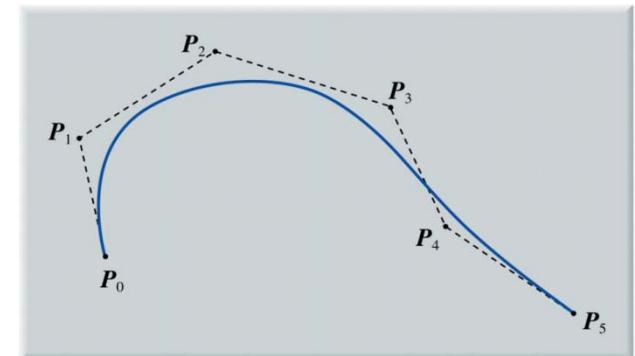
- 実用的には3次が多く用いられる



一様3次Bスプライン既定関数



一様3次Bスプライン曲線($L=3$)



非一様3次Bスプライン曲線

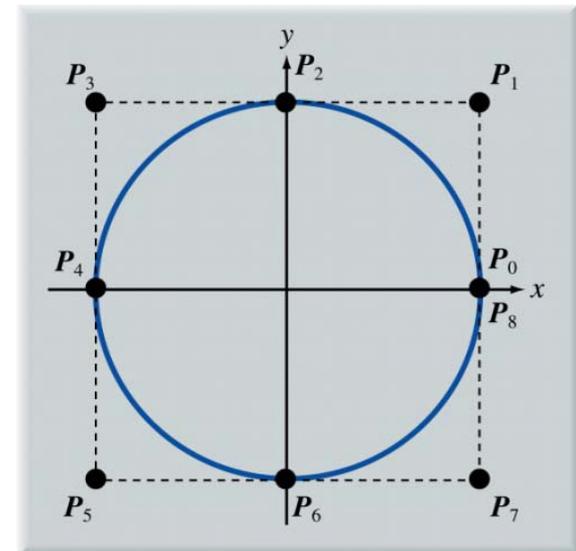
NURBS曲線

- 制御点 $\{P_i\}$ とそれらに対する重み $\{w_i\}$ 、ノット列 $\{N_i\}$ で定義する区分別有理式曲線
- 全ての重みを等しくするとBスプライン曲線になる

$$P(t) = \frac{\sum_{i=0}^{n+L-1} w_i P_i N_i^n(t)}{\sum_{i=0}^{n+L-1} w_i N_i^n(t)}, \quad t_n \leq t \leq t_{n+L}$$

- 2次のNURBS曲線でノット列を $(0,0,0,1,1,1)$ とすると、2次の有理ベジエ曲線になる
 - 2次曲線を正しく表現できる

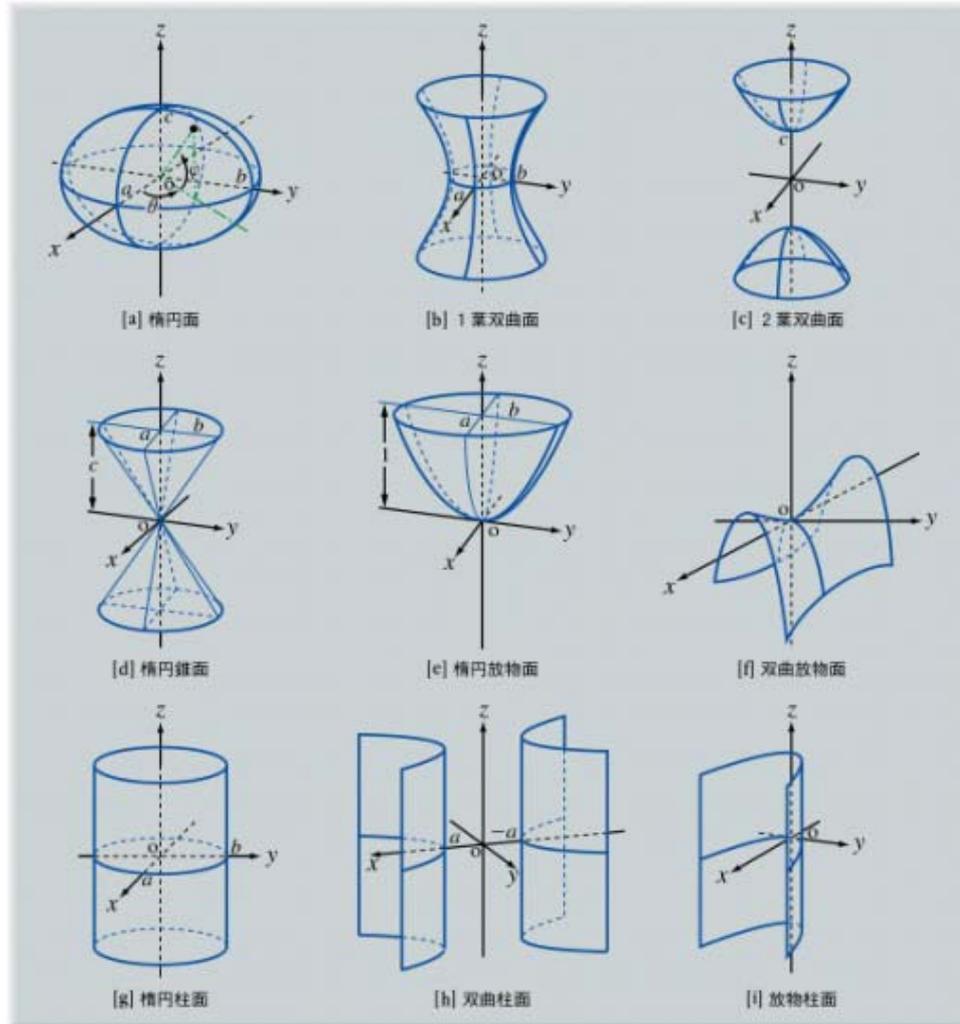
NURBSによる円周の表現



2次曲面

- 2次多項式によって表現される代数曲面

$$ax^2 + by^2 + cz^2 + d + 2eyz + 2fzx + 2gxy + 2hx + 2iy + 2jz = 0$$



パラメトリック曲面

- 各座標がパラメータ u, v の陽関数 $S=F(u, v)$ と表現された曲面
 - 多項式曲面： F が u と v の多項式
 - 有理式曲面： F が u と v の有理式
 - 複合曲面：複数の曲面の部分(パッチ)をつなぎ合わせたもの
 - 双 n 次曲面： u と v の次数が共に n の曲面
- 曲面の性質
 - 凸包性：曲面線が制御点の凸包内に存在
 - 変動減少性：曲面が制御点列の形状をなめらかな形にしたもの
 - 局所性：制御点を移動したとき曲面で変形する部分が限定される
 - 平面再現性：制御点が平面上にあれば得られる曲面が平面になる
 - アフィン不変性：曲面にアフィン変換を施したものと、アフィン変換を施した制御点から得られる曲面が一致
 - 2次曲面の再現性：球面や楕円面、双曲面を正確に表現できること
 - 射影不変性：曲面に射影変換を施した結果と、射影変換を施した制御点から得られる曲面が一致すること

パラメトリック曲面

曲面の性質

| 性質 \ 種類 | ベジエ曲面 | Bスプライン曲面 | 有理ベジエ曲面 | NURBS 曲面 |
|----------|-------|----------|---------|----------|
| 凸包性 | ○ | ○ | ○ | ○ |
| 変動減少性 | ○ | ○ | ○ | ○ |
| 局所性 | × | ○ | × | ○ |
| 平面再現性 | ○ | ○ | ○ | ○ |
| アフィン不変性 | ○ | ○ | ○ | ○ |
| 2次曲面の再現性 | × | × | ○ | ○ |
| 射影不変性 | × | × | ○ | ○ |

制御点を用いて形を定める曲面の特徴

| | | パッチ | 複合曲面 |
|---------------|----|---------|----------|
| 局所性 \ 2次曲面再現性 | なし | ベジエ曲面 | Bスプライン曲面 |
| | あり | 有理ベジエ曲面 | NURBS曲面 |

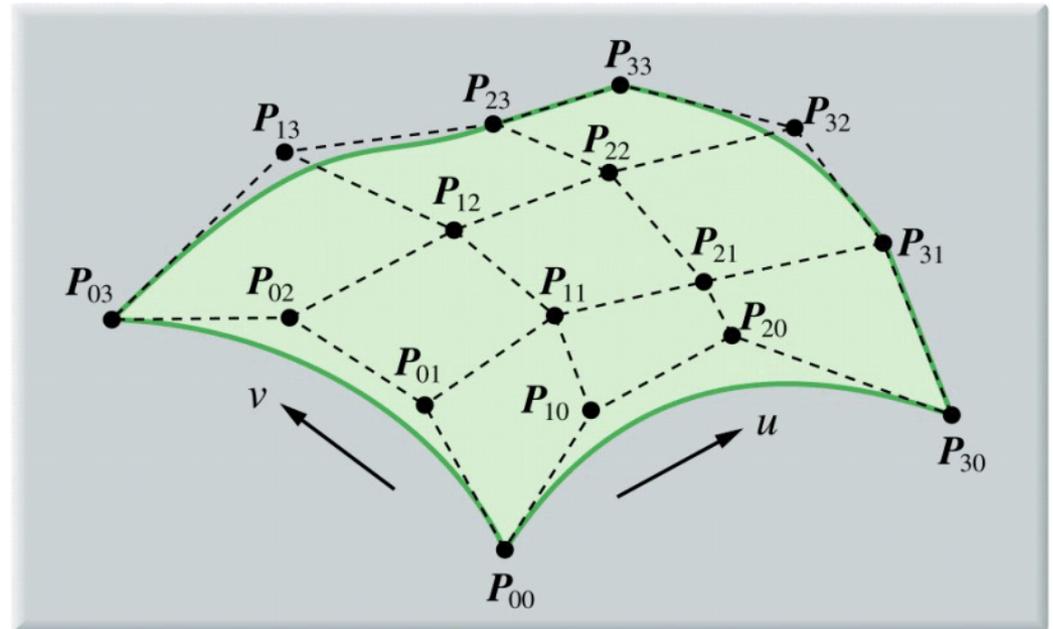
ベジエ曲面

- $(n+1) \times (m+1)$ 個の格子状に並べた制御点 $\{P_{ij}\}$ で定義
- 制御点 $P_{00}, P_{n0}, P_{0m}, P_{nm}$ が四隅を指定し、他が外形を定める
- 四辺は3次のベジエ曲線となる
- CGでは次数が3次までのベジエ曲面が広く用いられる
- 双3次ベジエ曲面は双3次クーンズ曲面と一致する

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

$$B_i^n(u) = \binom{n}{i} u^i (1-u)^{n-i}$$

$$B_j^m(v) = \binom{m}{j} v^j (1-v)^{m-j}$$



双3次ベジエ曲面

Bスプライン曲面

- 制御点 $\{P_i\}$ 、 u 方向と v 方向のノット列 $\{u_i\}$, $\{v_i\}$ で定義される
- Bスプライン曲線が複数の区間から構成されるのと同様に、いくつかのパッチから構成される
- $n \times m$ 次のBスプライン曲面を構成する各パッチ $(n+1) \times (m+1)$ 個の制御点で定められ、この他の制御点の移動はそのパッチに影響しない → 局所性

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

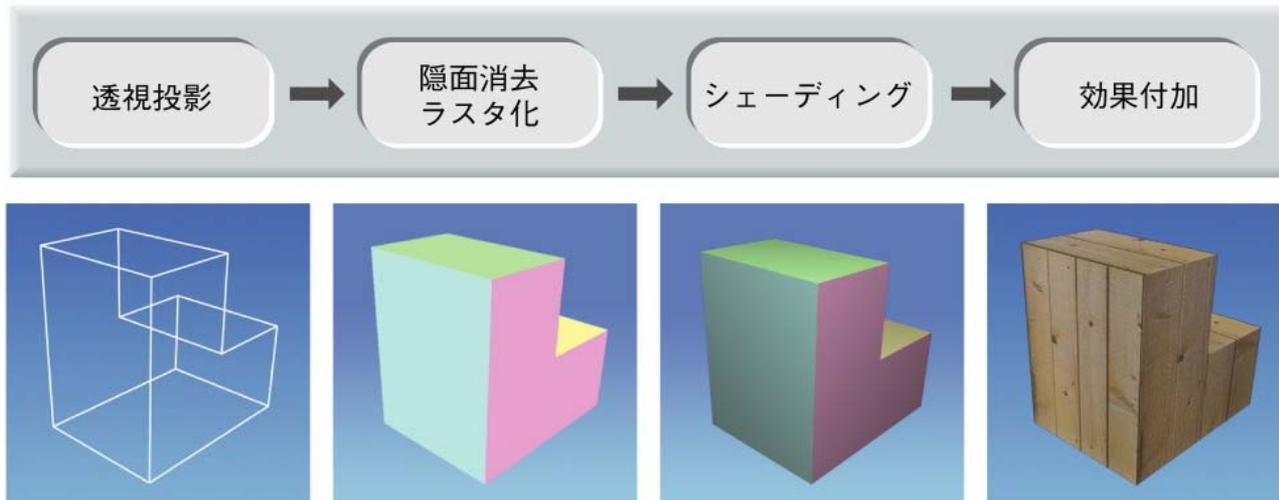
レンダリング1

－ 陰面消去 －



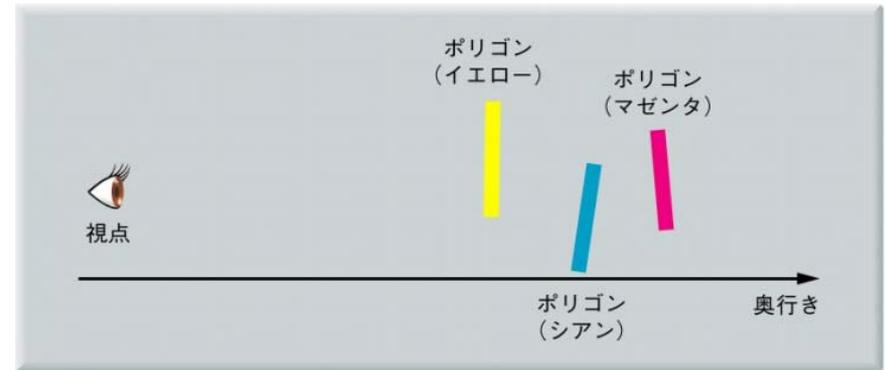
実写的表現法

- 3次元レンダリングは以下のような処理で構成される
 - 透視投影：3次元空間で定義された形状モデルを2次元の投影面(スクリーン)に投影
 - 陰面消去・ラスタ化：手前の面で隠された線や面の一部を消去し、その過程で線分によりベクタ表現されていた図形が画素の集まりのラスタ表現に変換
 - シェーディング：光の物理的な性質に基づいて表示対象の輝度(画素のR,G,B値)を計算
 - 効果付加：テクスチャマッピング等様々な効果を付加



優先順位アルゴリズム

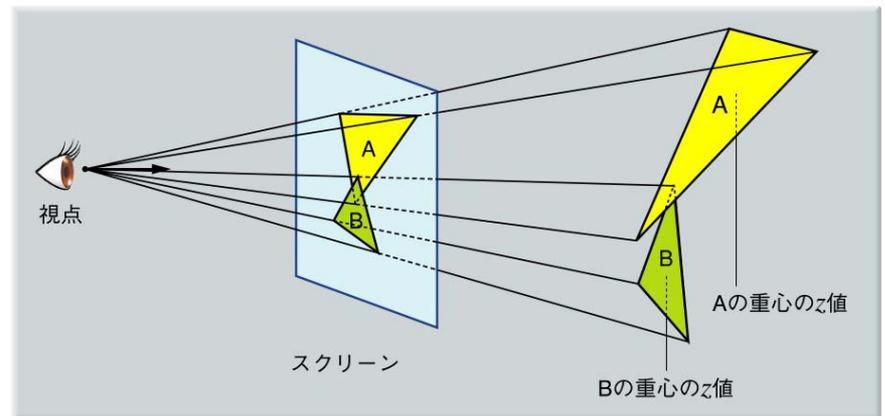
- 画素の色を格納するメモリのフレームバッファに対して奥にある面を先に、手前にある面を先に描くことで陰面消去を行う
- 優先順位を視点とポリゴンの代表点(重心位置等)との距離で決める単純な方法では陰面消去に失敗する場合があります。



[a] ポリゴンの配置



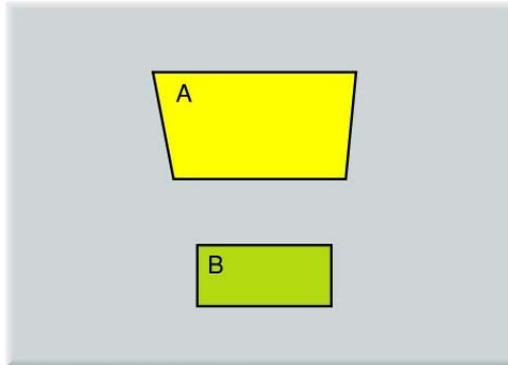
奥行による優先順位のソート



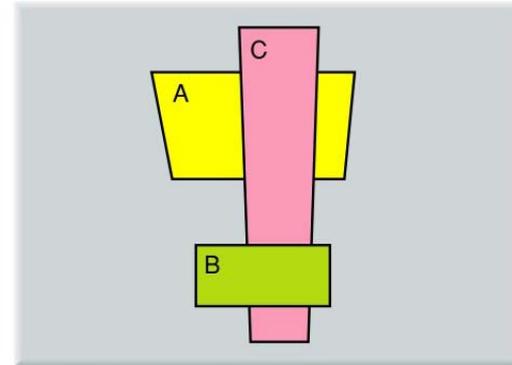
重心によるソートで陰面消去に失敗した例

優先順位アルゴリズム

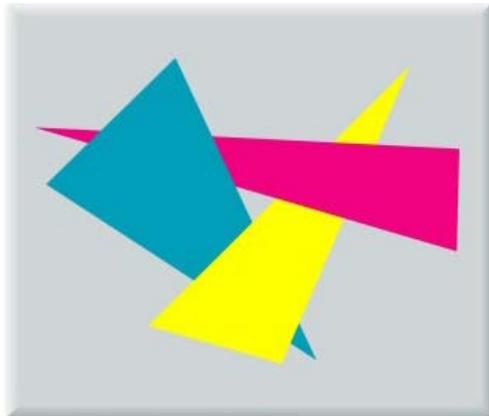
- 2つのポリゴンでも順序を決められない場合が多々ある
 - ポリゴンを分割することで解決する場合もある
- 優先順位の決定に必要な情報を視点が与えられる前に計算して奥方法も開発されている



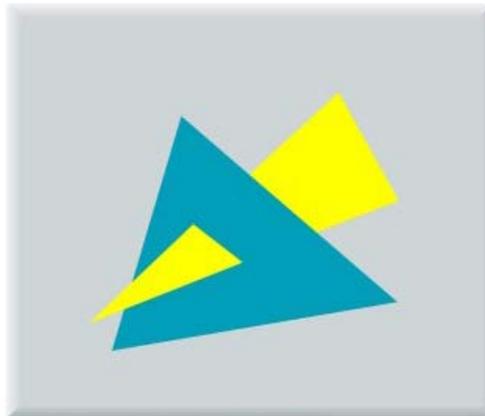
[a] ポリゴンが2枚の場合



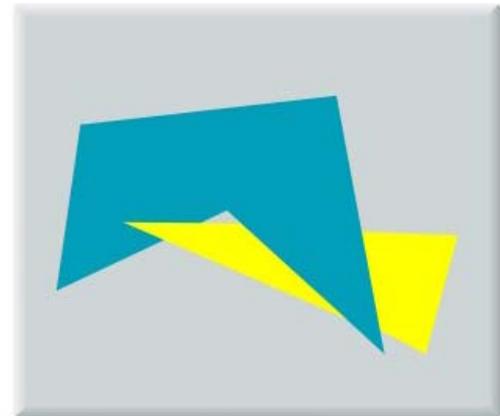
[b] ポリゴンが3枚に増えた場合



[a] 三すくみの場合

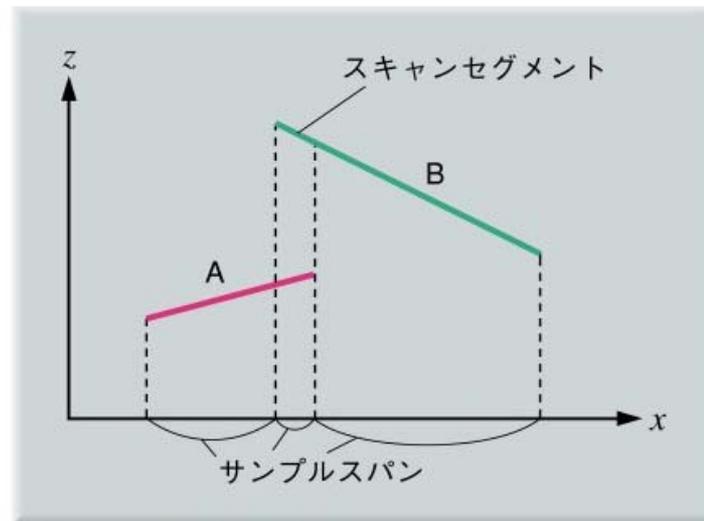
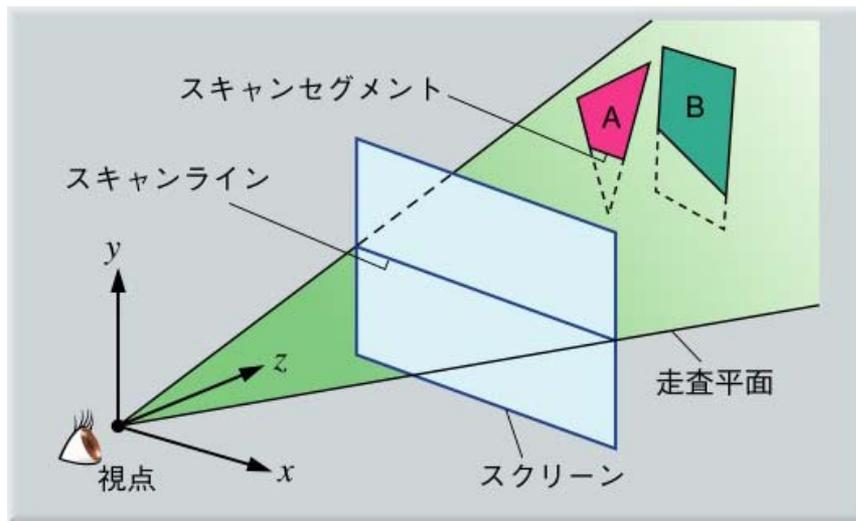


[b] 貫通している場合



[c] 凹ポリゴンの場合

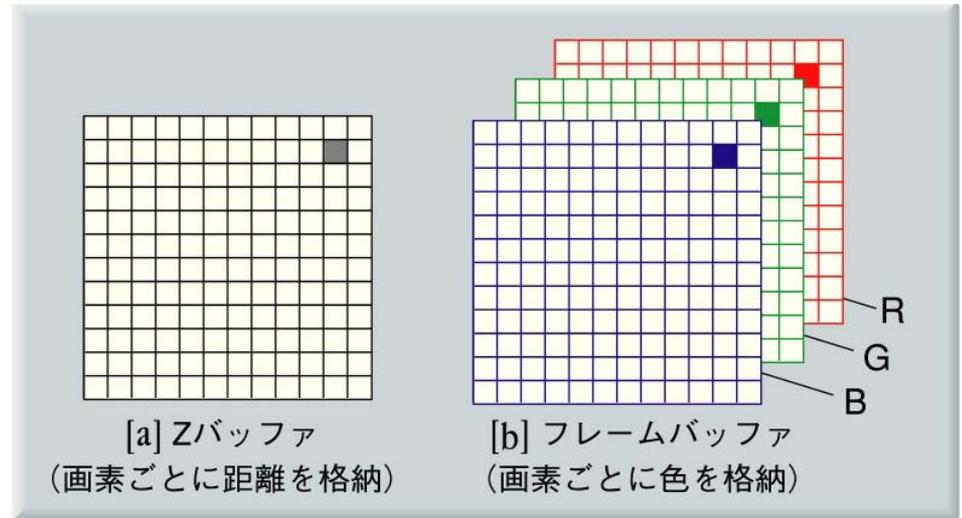
スキャンライン法



- 作業メモリはスキャンライン一本分で済むが，光の反射や映り込み等の表現ができない

Zバッファ法

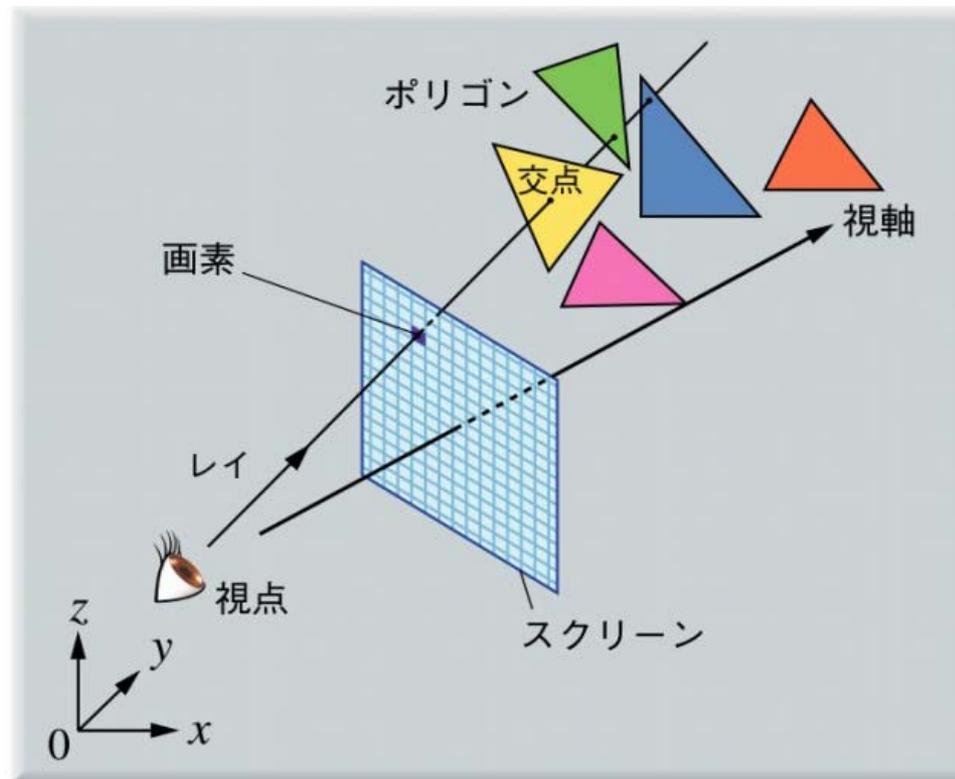
- フレームバッファと同じ解像度で、画素ごとに奥行き (z値) を格納するZバッファを用意して、交点のz値を比較
- ポリゴン以外の曲面にも適用できる
- アルゴリズムが簡単でハードウェア化しやすいためGPU(Graphics Processing Unit)で採用されている



Zバッファ法の描画過程

レイトレーシング法

- ワールド座標系で視点からスクリーンの画素に向うレイと物体の交点を計算
- レイを交差した物体の反射や透過・屈折方向に分けて処理を繰り返す
- 各画素の処理が独立しているので複数のCPUで並列化可能



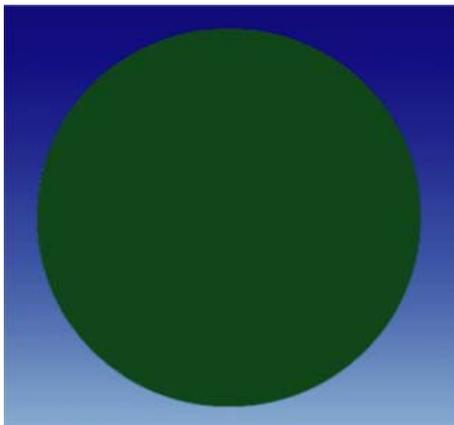
レンダリング2

－ シェーディング －



シェーディングモデル

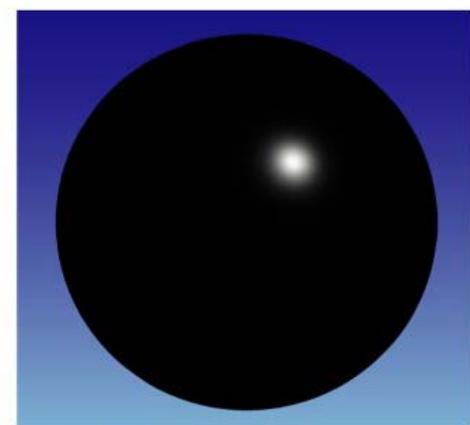
- 直接光に関して照明計算を行う局所照明モデルと間接光まで考慮する大域照明モデルに分類される
- 光源は太陽光の様な平行光線, 1点から放射状に広がる点光源, 長さを持つ線光源, 大きさを持つ面光源に分けられる
- 直接光は光源から出た光で, 間接光は光源から出た光が他の物と相互作用を起こしたものの
- 間接光は反射光(鏡面反射, 拡散反射), 物質を通過した透過光, 微粒子で散乱された散乱光に分けられる



[a] 環境光



[b] 拡散反射光



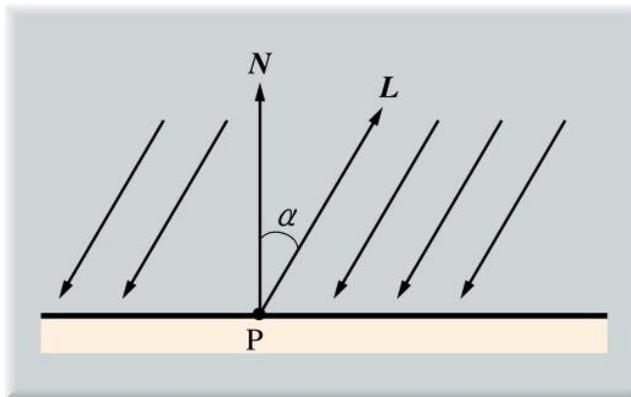
[c] 鏡面反射光

平行光線の拡散反射

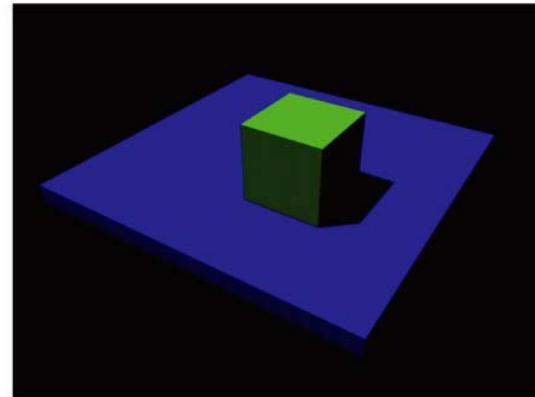
- どの方向から見ても物体面の輝度が一定となる反射を拡散反射と呼ぶ
- 入射角 α の平行光線で照射された拡散反射光の強さ I は

$$I = k_d I_i \cos \alpha = k_d I_i (\hat{N} \cdot \hat{L})$$

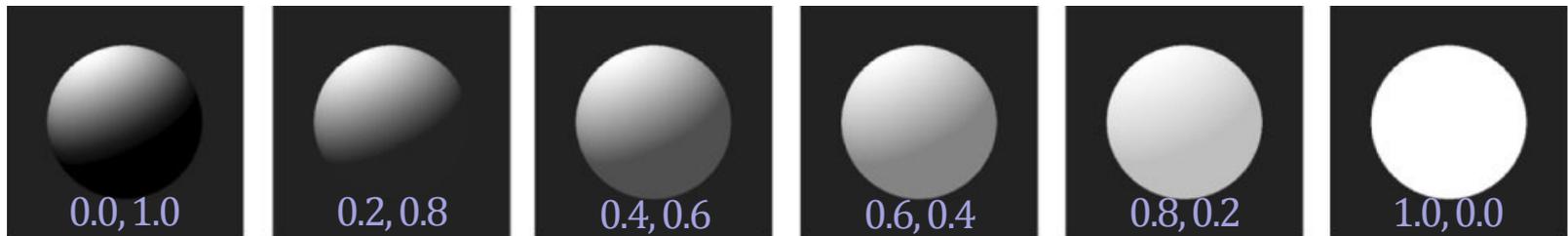
k_d : 拡散反射率
 I_i : 入射光の強さ



[a] 平行光線による拡散反射成分の計算



[b] 平行光線により照射された物体



環境光に対する反射率 k_a と拡散反射率 k_d の変化にともなう陰影の違い

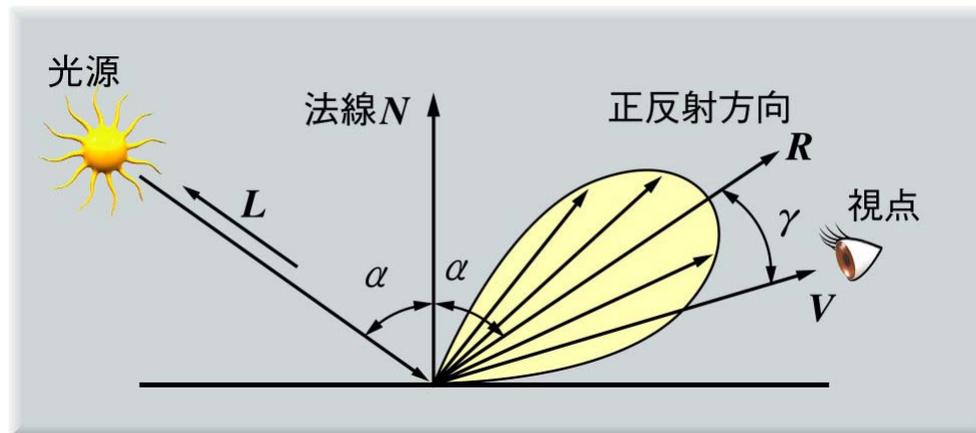
鏡面反射

- 拡散反射は物質内の浅い部分で多重散乱した光であるのに対し、鏡面反射は物質表面での直接反射で生じる
- フォンのモデルで強さ I_i の入射光に対する鏡面反射光の強さ I は視線方向と正反対方向のなす角 γ により

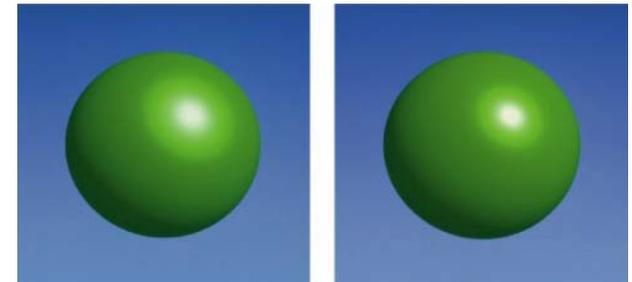
$$I = W(\alpha) I_i \cos^2 \alpha \cos^n \gamma = W(\alpha) I_i (\hat{\mathbf{R}} \cdot \hat{\mathbf{V}})^n$$

$W(\alpha)$: 鏡面反射率

n : ハイライト特性制御

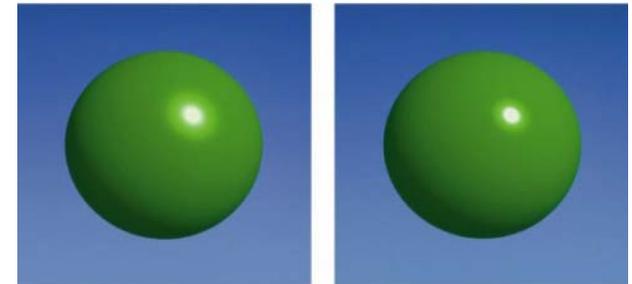


鏡面反射



[a] $n=2$

[b] $n=5$



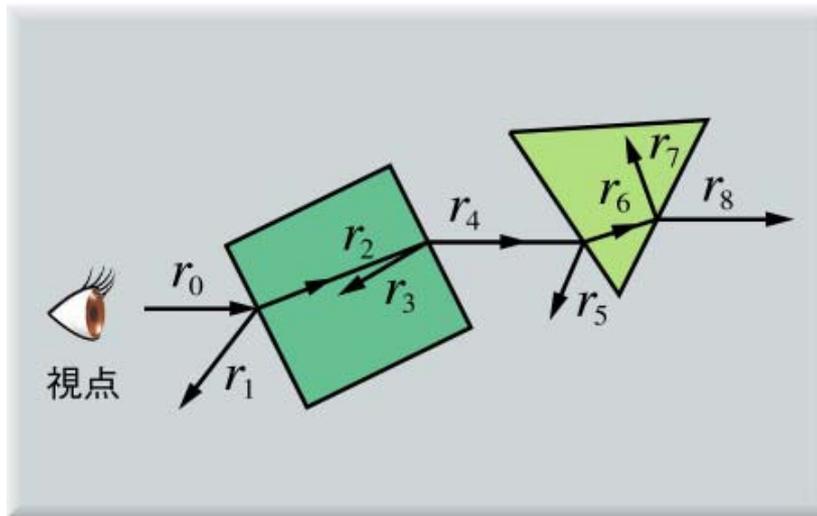
[c] $n=10$

[d] $n=20$

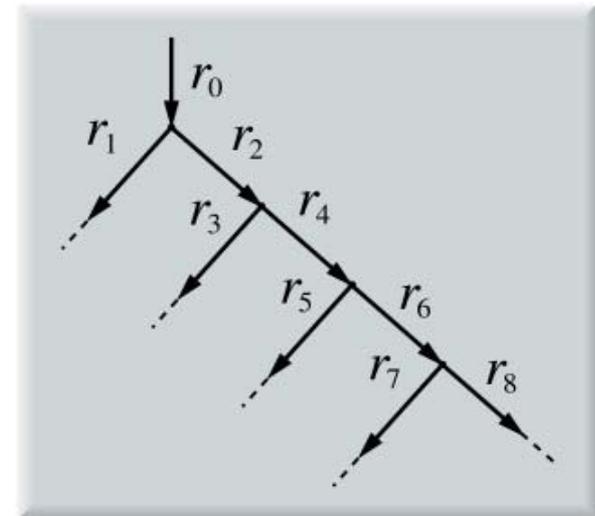
フォンのモデルによるハイライトの違い

レイトレーシング法による反射と透過・屈折

- レイトレーシング法は反射，透過・屈折の度にレイを分割し，拡散反射面に当たるか交差する物体がなくなるまで再帰的に繰り返す
- 計算に時間がかかるため，光の減衰やレイの分割回数に応じて追跡を中止する工夫が取られる



[a] 光の反射と屈折



[b] 光線の追跡過程の二分木表現

屈折方向ベクトル

- レイが屈折する効果の係数 k_f で屈折方向ベクトル T を表すと

$$T = k_f(\hat{N} + V') - \hat{N}$$

$k_f = 1$: レイが屈折率の同じ
媒質中を直進

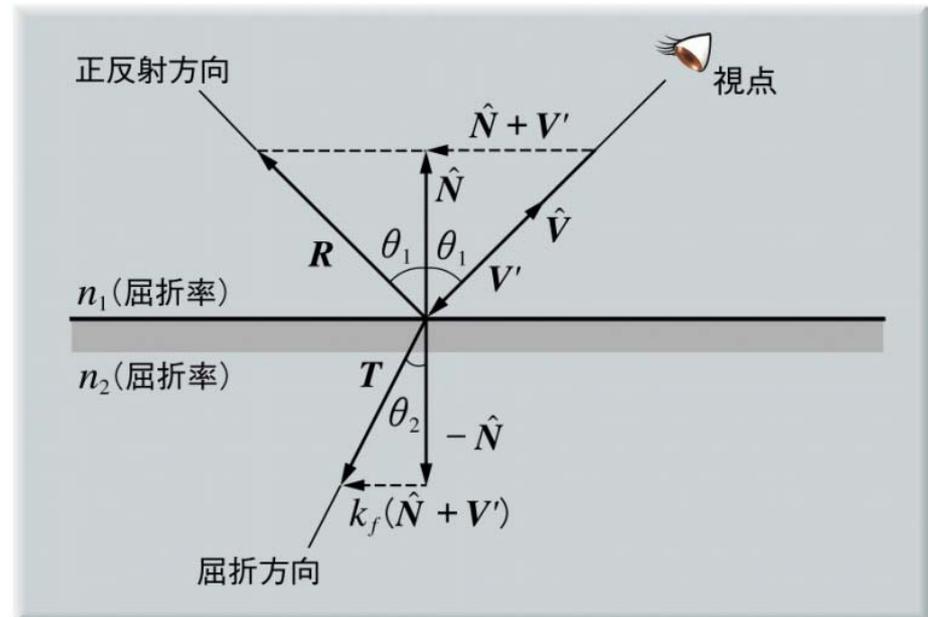
$k_f < 1$: 屈折率低 \Rightarrow 屈折率大

$k_f > 1$: 屈折率大 \Rightarrow 屈折率小

- 屈折率 n_1, n_2 の媒質の
相対屈折率 n は

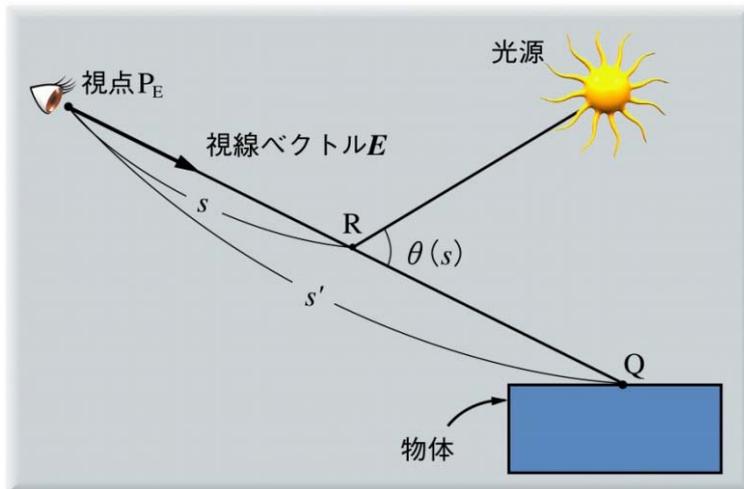
$$n = \frac{n_2}{n_1}$$

$$= \frac{\|k_f(\hat{N} + V') - \hat{N}\|}{k_f \|V'\|}$$



散乱・減衰

- 光が通過する媒体が散乱粒子を含むと散乱・減衰が生じる
- レイトレーシング法を拡張したレイマーチング法を用いる
 - レイに沿って微小距離ずつ移動しながら、視点に到達するまでの減衰を考慮して散乱光を加算
 - 視線方向 E と光源方向とのなす角 θ の関数で粒子の散乱特性を表す位相関数を用いて点 R の散乱を計算
 - 位相関数の分布形状は散乱粒子の大きさと光の波長に大きく依存



[a] 霧の表示



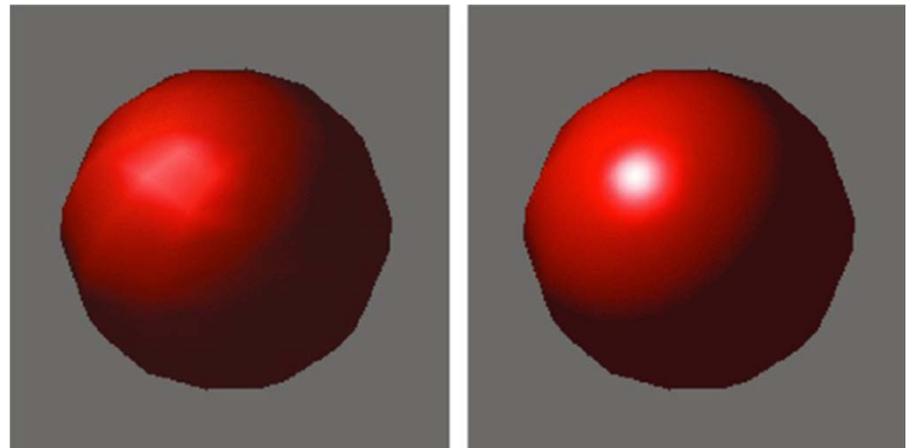
[b] 夕焼け空と夕焼けによって照らされた建物

スムーズシェーディング

- 代表点の輝度でポリゴンを表示するコンスタントシェーディングでは、輝度に敏感な目の特性(マツノバンド効果)のため近似したポリゴン形状が見えてしまう
- ポリゴン内の輝度を保管してなめらかな明るさ表示を行う方法をスムーズシェーディングと呼ぶ
 - グローの手法
 - フォンの手法
- 表示された物体の輪郭部分の形状は変わらない



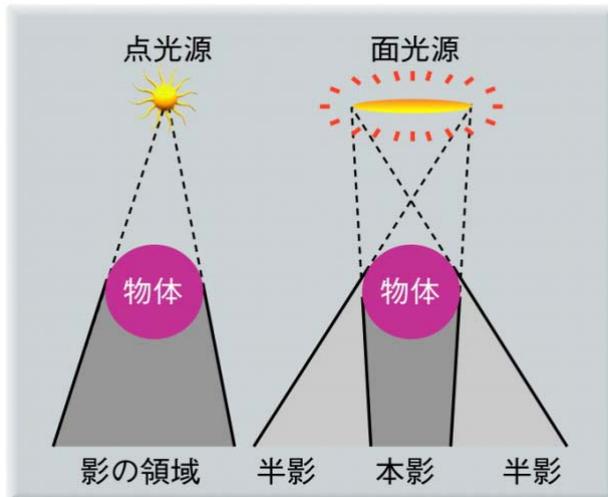
コンスタントシェーディングとスムーズシェーディング



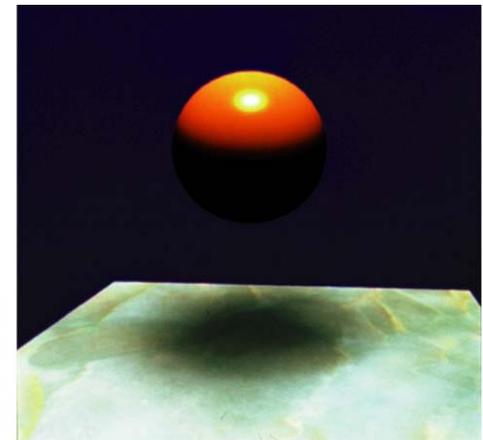
グローとフォンのスムーズシェーディング

本影と半影

- 平行線や点光源によって生じる影は，境界とそうでない部分がはっきり分かれるが，面光源は境界がぼやける
 - 本影：光源からの光がまったく届かない領域
 - 半影：光源からの光の一部が到達する領域
- 平行光源や点光源の影付けは比較的簡単に行える
 - 主な方法に①レイトレーシング法 ②スキャンライン単位の方法 ③物体空間における2段階法 ④シャドウポリゴン法 ⑤Zバッファ法を用いた2段階法 などがある



[a] 点光源



[b] 面光源(本影と半影ができる)

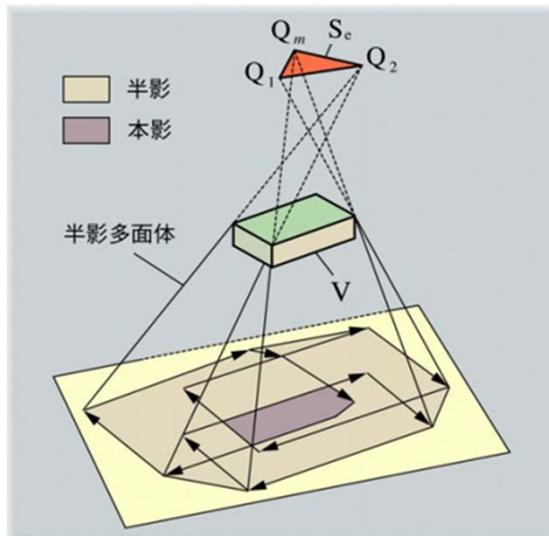
大きさをもつ光源による影

● 近似的に求める方法

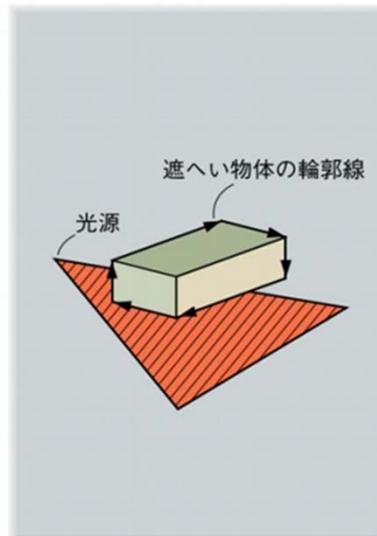
- 光源の表面上に複数の点光源を配置して近似する方法は点光源の密度が低い場合計算誤差が大きい
- 光源を球として考え、円錐と物体との交差領域で影の度合を近似

● 積分法により求める方法

- 半影内の点の明るさはそこから見た光源の可視領域の大きさに依存し、積分によって求めることが可能



[a] 本影と半影の領域



[b] 面光源の可視領域



[a] 円柱光源



[b] 曲面光源

ラジオシティ法

- パッチ同士のラジオシティ授受の割合を示すフォームファクタはパッチの形状や配置から計算され，光源や視点とは無関係で事前計算できる
- 拡散反射成分だけを考慮した相互反射光の計算はパッチの放射輝度は方向によらず一定
- 鏡面反射成分を考慮した計算では放射輝度が方向性を持つため処理が複雑となる



[a] 直射光のみの画像



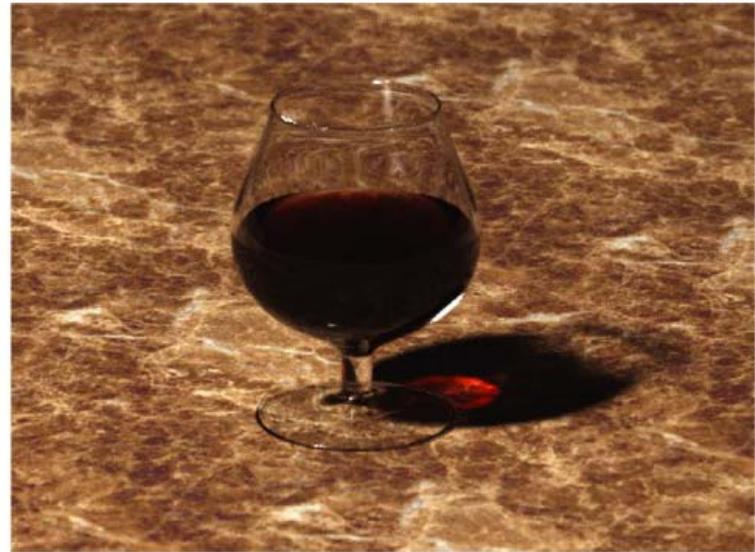
[b] 相互反射を考慮したラジオシティ法による画像

フォトンマップ法

- ラジオシティ法で鏡面反射や屈折を生じる透明な物体で生じる集光現象の表現は困難
- レイトレーシング法を拡張したフォトンマップ法は拡散反射面での光の相互反射と集光現象を統一的に扱え、パッチ分割も不要
- 2段階のレイトレーシングを行って光の相互反射を考慮したレンダリングを行う



[a] 集光現象および映り込みの表示



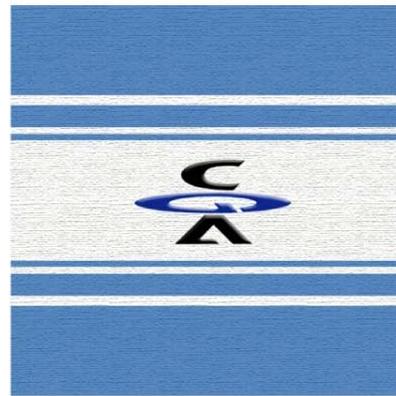
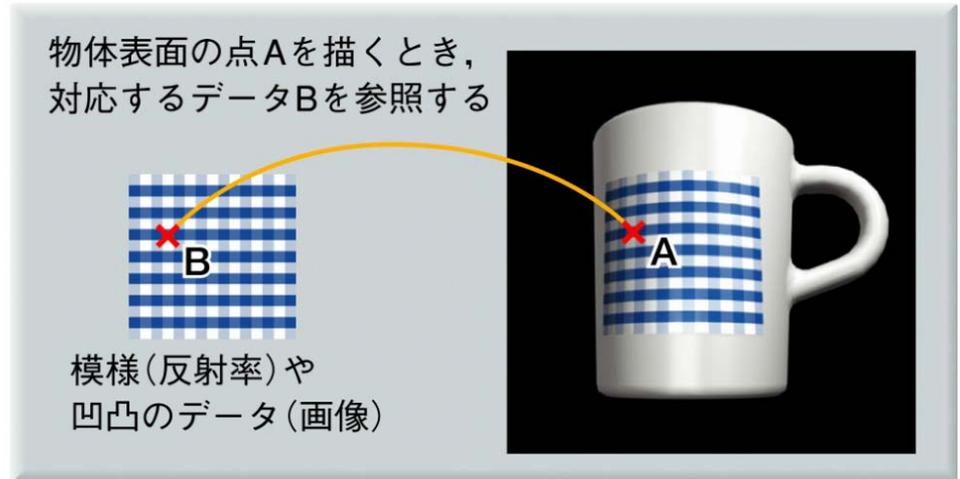
[b] グラスの中のコニャックによる集光現象の表示

レンダリング3 - マッピング -



マッピング

- 物体表面の模様や微小な凹凸を直接ポリゴン等でモデル化する代わりに画像等の張り付けで表現
 - テクスチャマッピング
画像・模様を張り付けて表示
 - バンプマッピング
凹凸を表示
 - 環境マッピング
反射による周囲の映り込みを表示



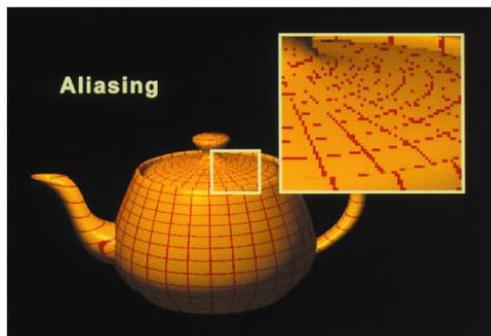
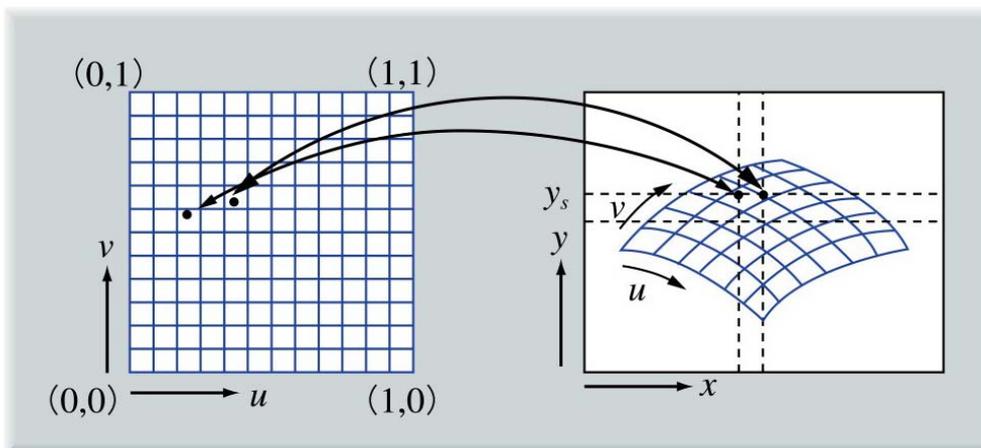
[a] テクスチャ



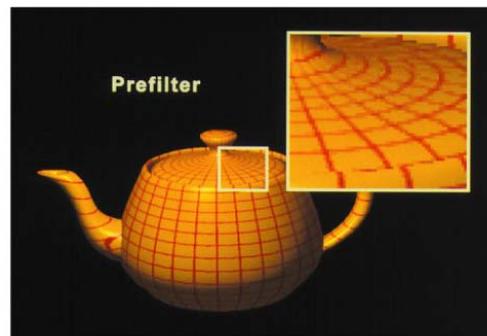
[b] 曲面上へのマッピング

アンチエイリアシング

- テクスチャマッピングではジャギー(エッジのギザギザ)やモアレ等のエイリアシングに注意が必要
- アンチエイリアシングはエイリアシングの影響を抑えて画質の劣化を防ぐ



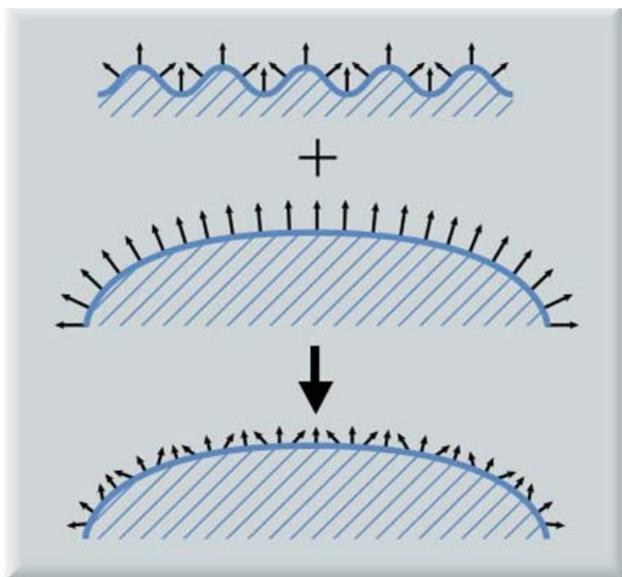
[a] エイリアシングの発生



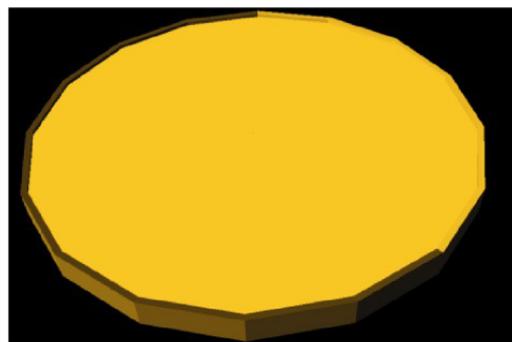
[b] アンチエイリアシング(プリフィルタリング法)を施した例

バンプマッピング

- 模様の代わりに凹凸を物体上にマッピングして表示
- 物体の表面形状を変えるのではなく、面の法線方向を変化させて凹凸を表現
- 物体の輪郭に凹凸は生じない
 - ディスプレイメントマッピングは物体の表面形状を変化させる



バンプマッピングの概念図



[a] 表面がフラットなコイン形状



[b] バンプマッピングを施した画像



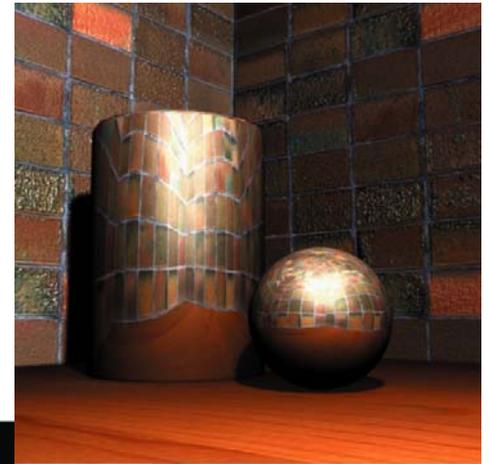
[a] バンプマッピング



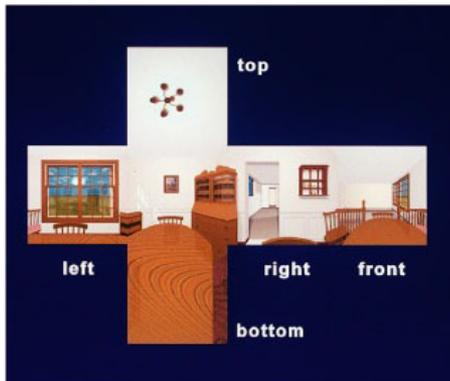
[b] ディスプレイメントマッピング

環境マッピング

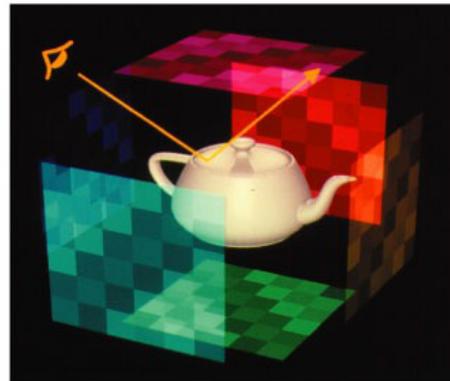
- 反射による周囲の映り込みをマッピングで疑似的に表現
- 表示する物体を囲む大きな球や立方体を仮想的に設置し、その内側に周囲の環境のテクスチャを張り付け、物体表面で反射したレイと仮想球/立方体との交点でのテクスチャの色を求める
- レイトレーシング法よりも少ない計算量で反射を疑似的に表現できるが物体同士が互いに写りこむようなシーンは表現できない



環境マッピングによる画像



6枚のテクスチャ



テクスチャの対応の算出



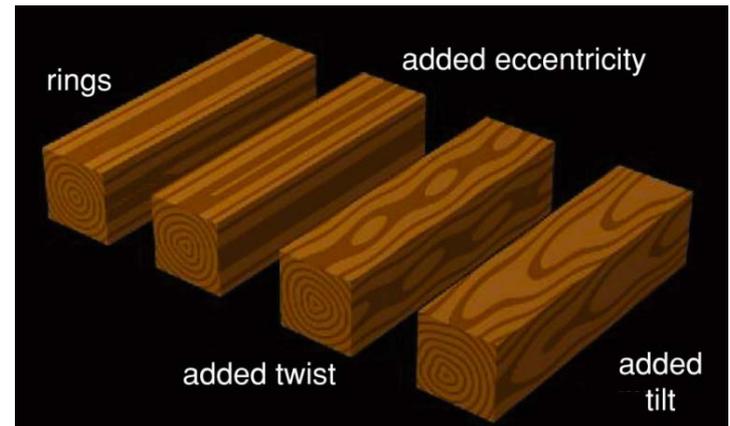
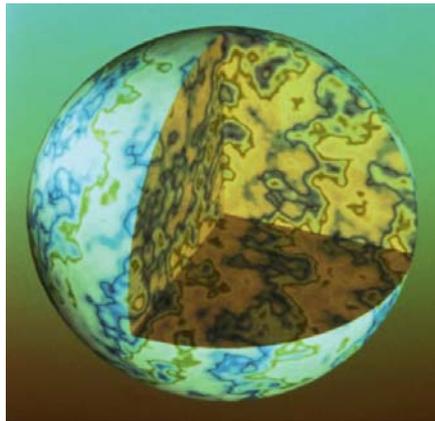
環境マッピングを施したティーポット

ソリッドテクスチャリング

- 大理石や木材のテクスチャを六面体の各面に施すと境界部分の模様を合わせるのが非常に困難
- ソリッドテクスチャリングは3次元空間でテクスチャを定義し、様々な形に切り出して表示する方法



各面ごとにテクスチャマッピングを施した例

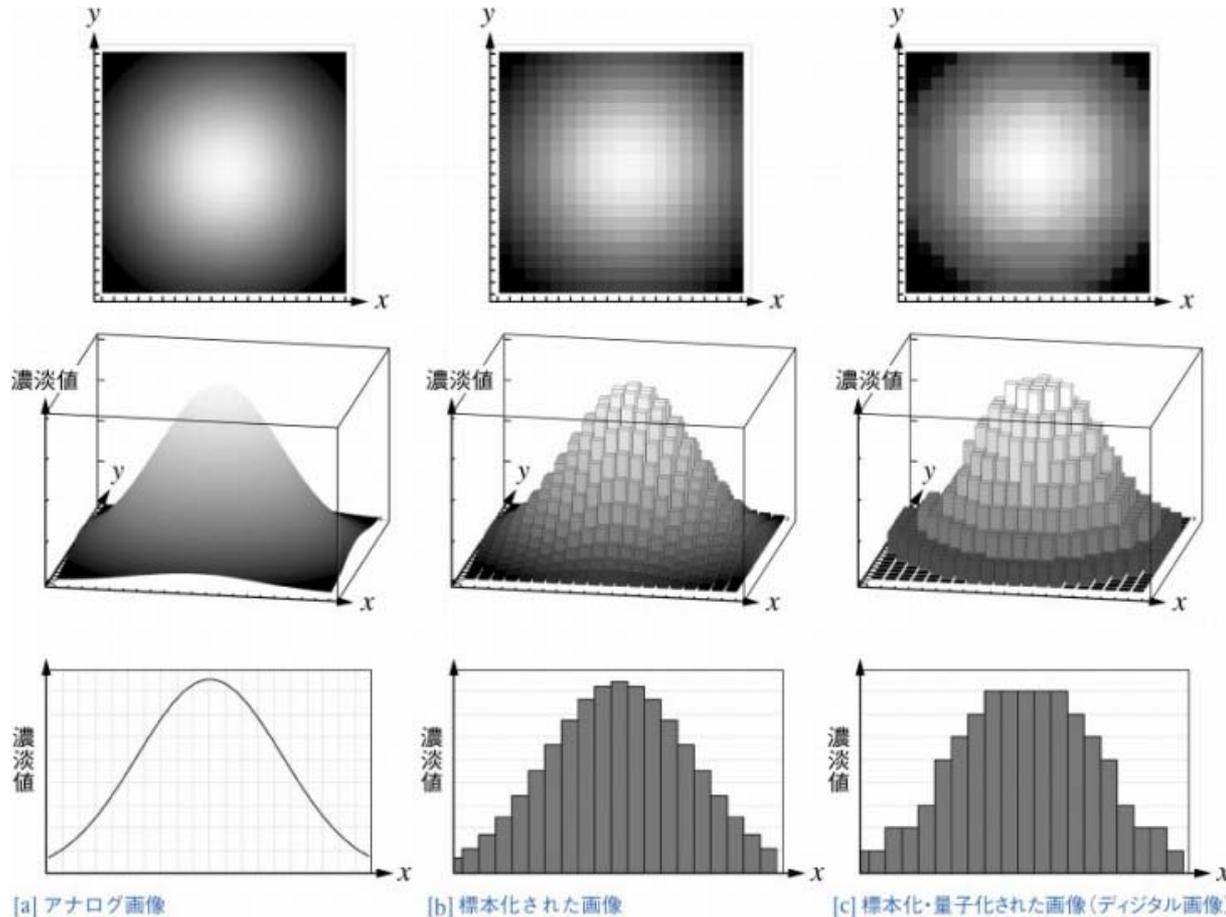


ソリッドテクスチャリングを施した例

画像処理 1

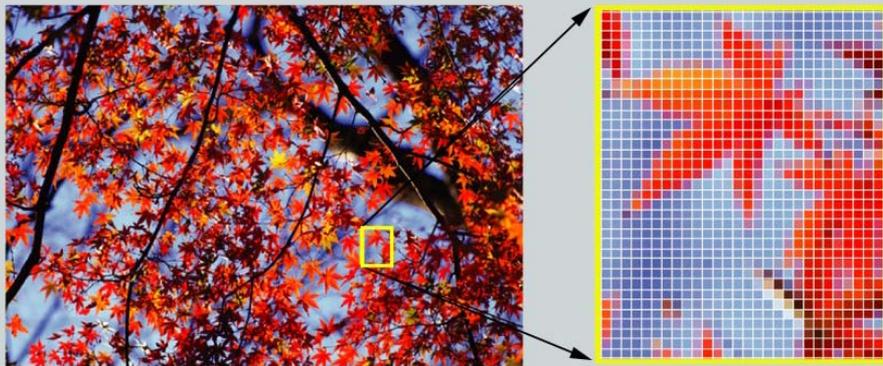
アナログ画像とデジタル画像

- どこまで拡大しても濃淡が連続的なアナログ画像は標本化と量子化によりデジタル画像に変換される
 - ① 標本化：離散的な位置にある濃淡信号を取り出す
 - ② 量子化：連続量の濃淡値を有限レベルの離散値に変換

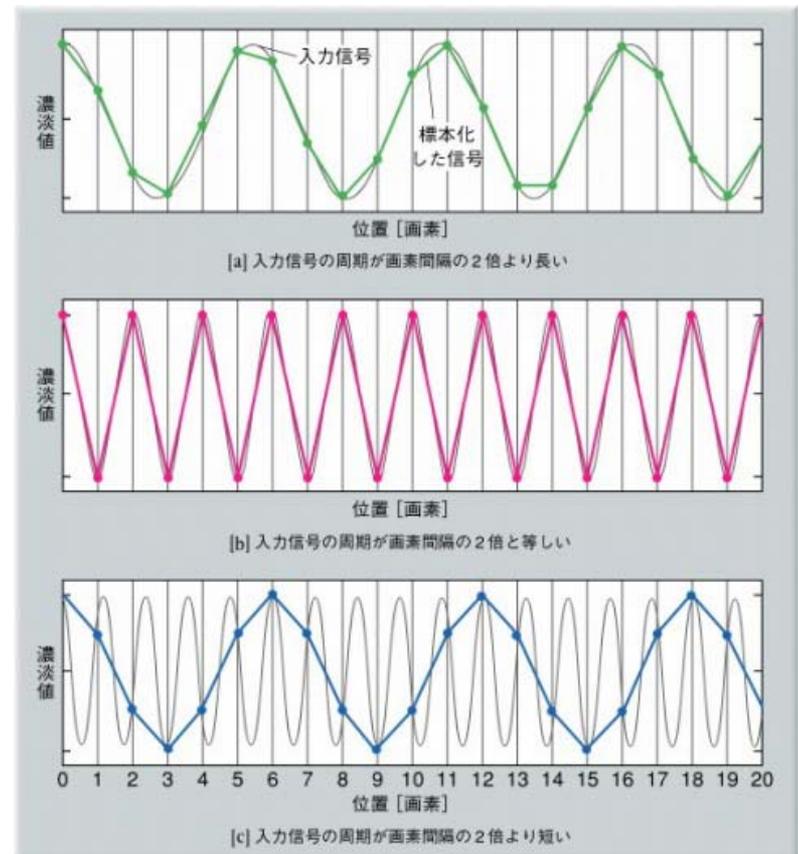


画像の標本化

- 2次元のアナログ画像に対して等間隔に配置した標本点での光強度を取り出す
- 画像上の位置に対して光強度が正弦波状に変化するアナログ画像は正弦波の周期の1/2よりも小さな感覚で標本化すれば元の画像が再現できる(標本化定理)
- 標本化間隔がこれより大きいと偽の信号が現れる(エイリアシング)



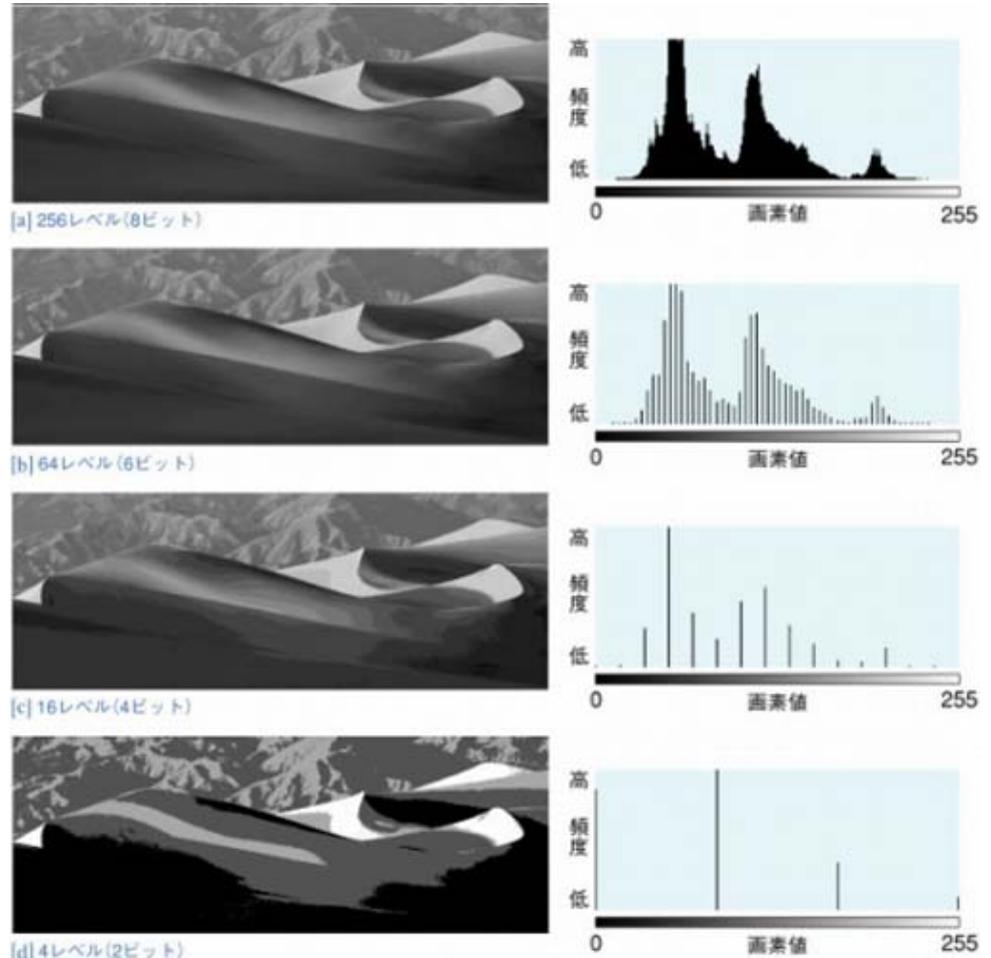
デジタル画像における標本化



標本化した濃淡値の信号

画像の量子化

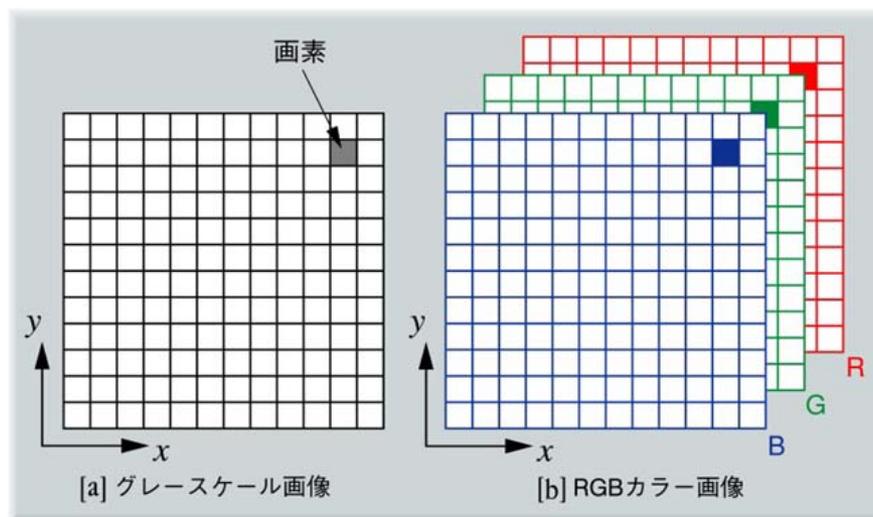
- 標本化したアナログ値を有限レベルの数値に変化する処理
- 量子化後の画素の値を画素地と呼ぶ
- 連続的な値を離散的な有限個の数値に置き換えるので丸め誤差(量子化誤差)が生じる
- 量子化が2レベルの画像を2値画像, 中間の灰色のレベルも表すときグレースケール画像と呼ぶ
- 量子化レベルが少ないと1段階変化するところに等高線のような境界の疑似輪郭(疑似エッジ)が現れる



量子化レベル数が異なる画像とヒストグラム

グレースケール画像とカラー画像

- グレースケール画像は明るさの情報だけを持つ
- カラー画像は三原色のそれぞれの画素値を持ち、三原色にR,G,Bを用いたものが一般的でRGBカラー画像と呼ばれる

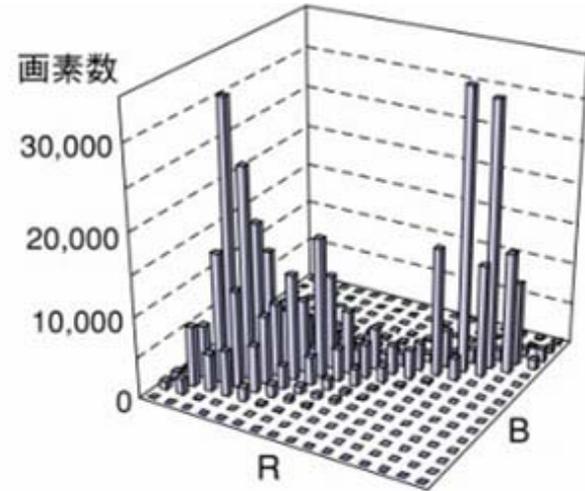


グレースケール画像とRGBカラー画像

| 画像サイズ | 量子化レベル数 | グレースケール/カラー | データサイズ |
|---------------|---------------|-------------|----------|
| 720 × 480 | 2 (1ビット) | (2値) | 約 42KB |
| 720 × 480 | 256 (8ビット) | グレースケール | 約 338KB |
| 720 × 480 | 256 (8ビット) | カラー | 約 0.99MB |
| 2,048 × 1,536 | 256 (8ビット) | グレースケール | 3MB |
| 2,048 × 1,536 | 256 (8ビット) | カラー | 9MB |
| 2,048 × 1,536 | 4,096 (12ビット) | カラー | 13.5MB |

限定色表示

- 入力画像が持つ色数を出力可能な色数に限定して出力することを限定色表示と呼ぶ
- 出力可能な色を全て並べたカラーマップから変換前後の値を表にしたルックアップテーブルを作成する
- 均等量子化法：色空間の量子化を均等に粗くする
- 頻度法：入力画像のカラーヒストグラムとのピークに近い色を優先的に表示色として用いる



[d] RGBヒストグラム空間(実際の色空間は3次元であるが、表示のため2次元空間で表現している)



[a] 原画像



[b] 均等量子化法



[c] 頻度法

ハーフトーニング

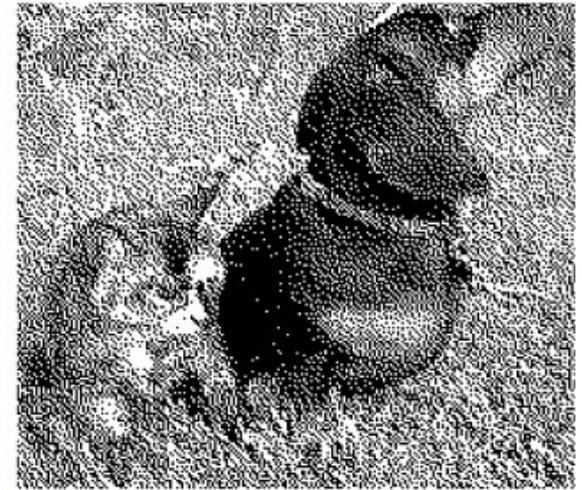
- 2値画像で空間的な白黒パターンの疎密を利用して濃淡の表行う手法
 - 濃度パターン法
 - ディザ法
 - 誤差拡散法



[a] 原画像 (濃淡画像)



[b] 2値画像



[c] ハーフトーニング

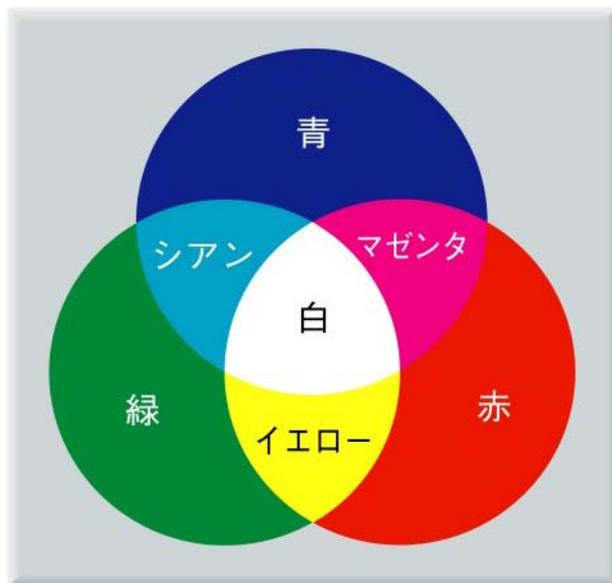
加法混色と減法混色

● 加法混色

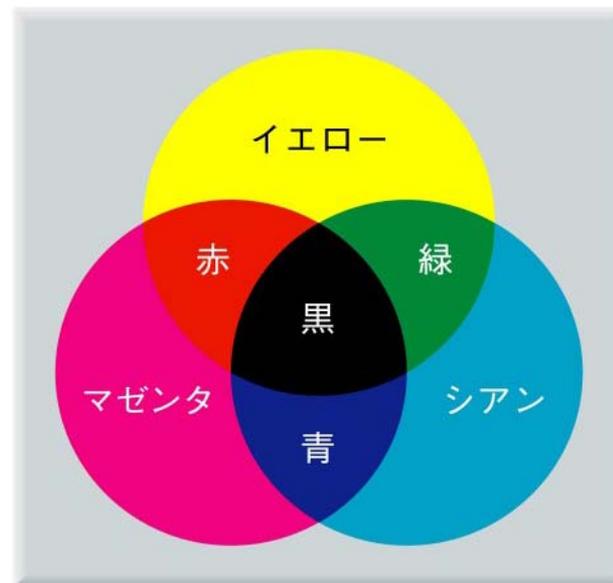
- 色を混ぜる毎に明るさが増加して白に近づく
- 光の三原色として知られるR,G,B
- カラーテレビ, CRT, 劇場のスポットライト等

● 減法混色

- 色を混ぜる毎に明るさが減少して黒に近づく
- 三原色はシアン(C),マゼンタ(M),イエロー(Y)
- 水彩絵の具やカラー印刷, 色セロファンの重ね合わせ等



加法混色



減法混色

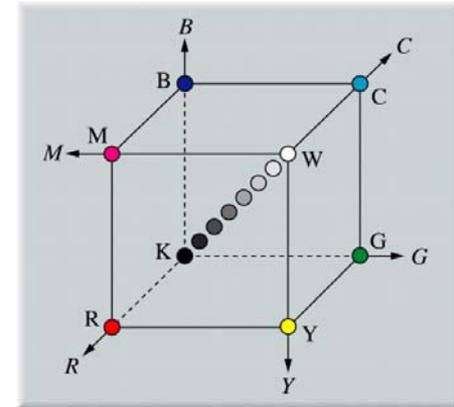
カラーモデル

- **RGBモデルとCMYモデル**：通常のCRTやプリンタの制御信号の強さは発光輝度や濃度と比例しないが，CGでは出力装置を単純な加法混合色装置や減法混合色装置にモデル化

$$\begin{pmatrix} r \\ g \\ b \end{pmatrix} + \begin{pmatrix} c \\ m \\ y \end{pmatrix} = 1$$

- **YIQモデル**：RGBモデルから次式で変換

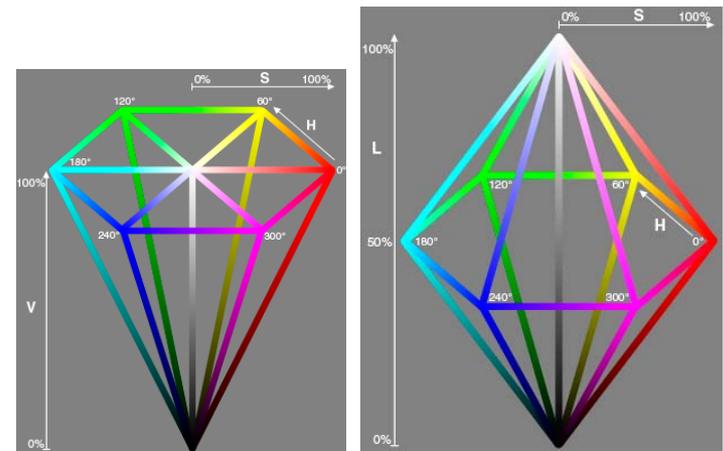
$$\begin{pmatrix} y \\ i \\ q \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.522 & 0.311 \end{pmatrix} \begin{pmatrix} r \\ g \\ b \end{pmatrix}$$



RGBモデルとCMYモデルの概念図

- **三属性に基づくモデル**：直感的にわかりやすい色相, 明度, 彩度を利用

- 六角錐モデルはRGBモデルを変形し，立方体をWの方向から平行投影
- 双六角錐モデルはRGBモデルの立方体の6点R, Y, G, C, B, Mを同一平面上の正六角形に配置したもの

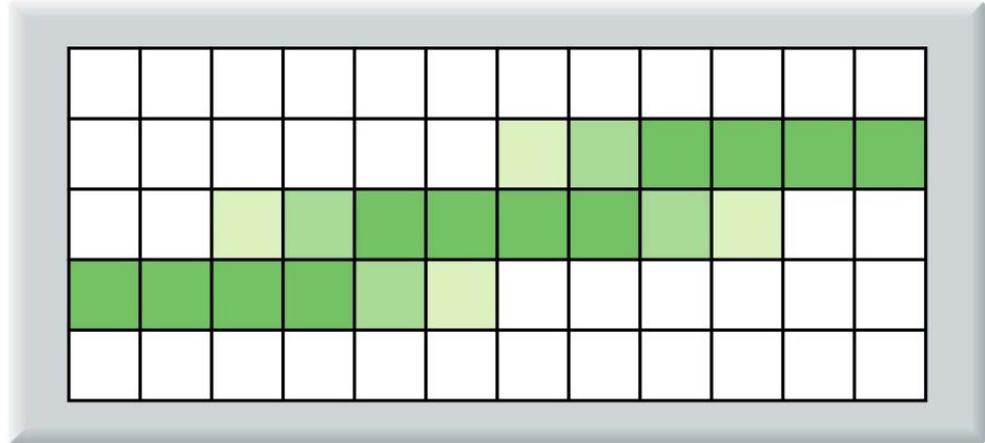


六角錐モデルと双六角錐モデル

アンチエイリアシング

- 斜め線や多角形の境界部分にジャギーが生じたり，細かい線や物体が寸断される現象をエイリアシングと呼ぶ

- デジタル変換時の標本化により，線や境界部で点は表示/非表示の一方しか選択できないことに起因



斜め線に対するアンチエイリアシング

- 階段状の画素部分に中間色の画素を配置して目立たなくすることをアンチエイリアシングと呼ぶ

- 境界部がなめらかになり，細かい領域の寸断や規則パターンのモアレが軽減される



[a] 原画像

[b] 処理結果

画像に対するアンチエイリアシング

画像処理2

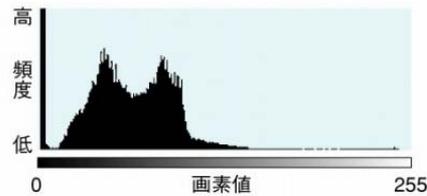


ヒストグラム

- 横軸に画素値を，縦軸に頻度を示すヒストグラムにより画像全体の性質が見える
- 明暗の分布は，明るい画像では明るいところに，暗い画像は暗いところに集まる
- コントラストの低い画像は分布が狭い範囲に塊．高い画像は画素地が全体に広がる



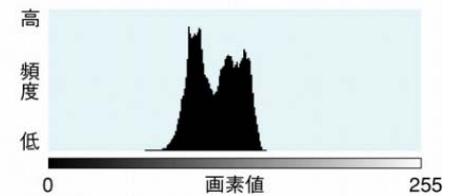
[a] 暗い画像



[b] 画像[a]のヒストグラム



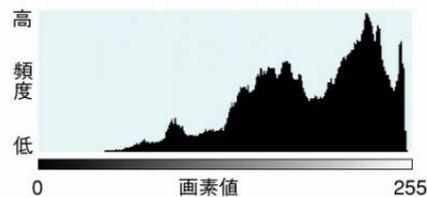
[a] コントラストの低い画像



[b] 画像[a]のヒストグラム



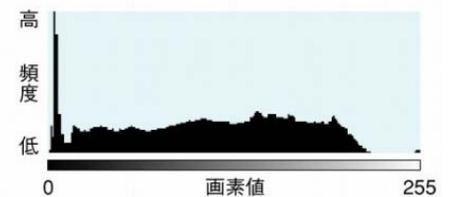
[c] 明るい画像



[d] 画像[c]のヒストグラム



[c] コントラストの高い画像



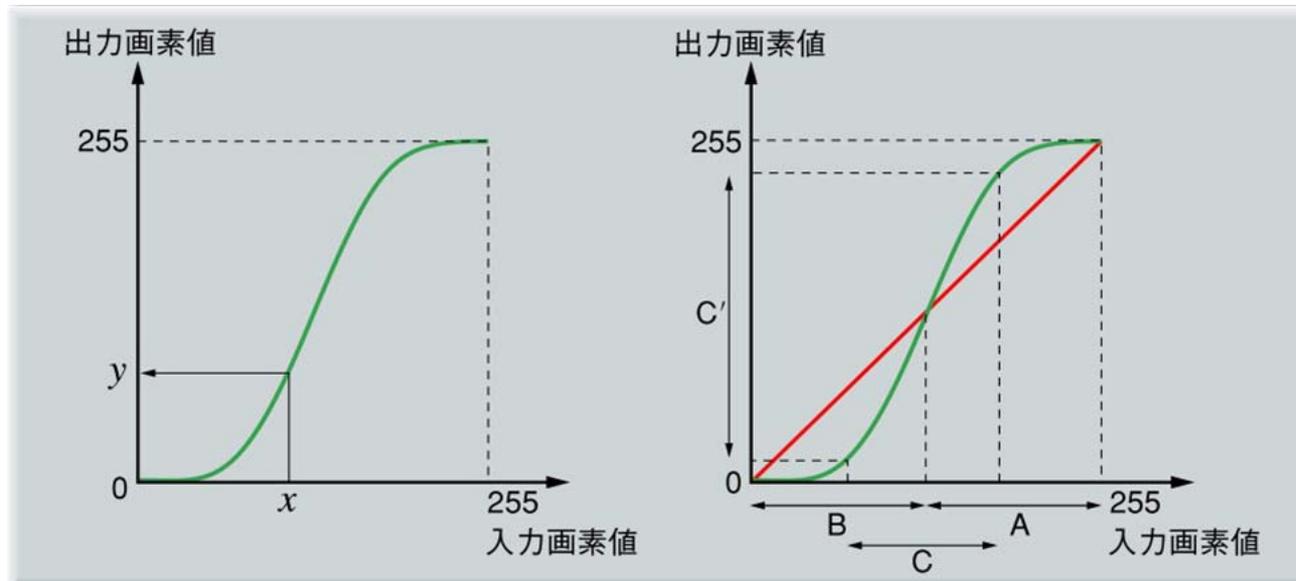
[d] 画像[c]のヒストグラム

画像の明暗とヒストグラム

画像のコントラストとヒストグラム

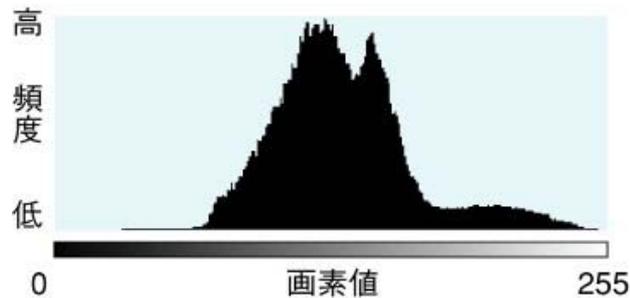
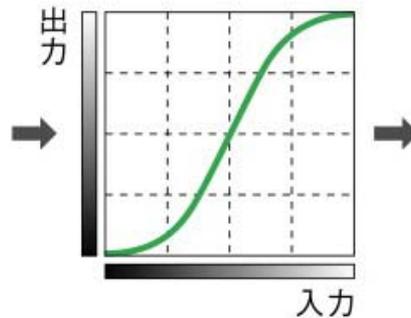
トーンカーブ

- 画像の濃淡を変化させるには，入力画像と出力画像の画素値の対応付け(階調変換関数)を与える
- 階調変換関数をグラフ表示したものをトーンカーブと呼ぶ
- カーブが直線よりも上にあれば明るく，下にあれば暗く変換される
- 傾きが直線よりも大きいとコントラストが高くなる

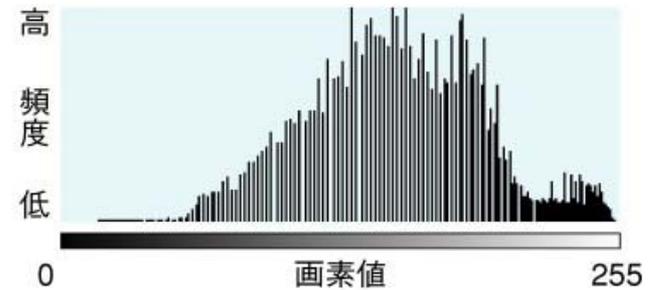


S字トーンカーブによる変換

- ヒストグラムの分布が中央に偏ったコントラストの低い画像は，中間調が引き伸ばされコントラストが上がる



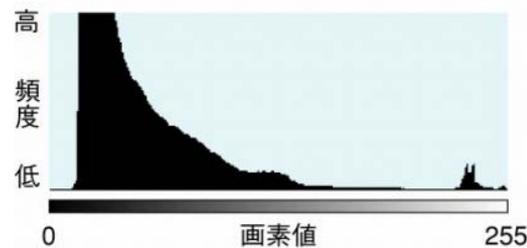
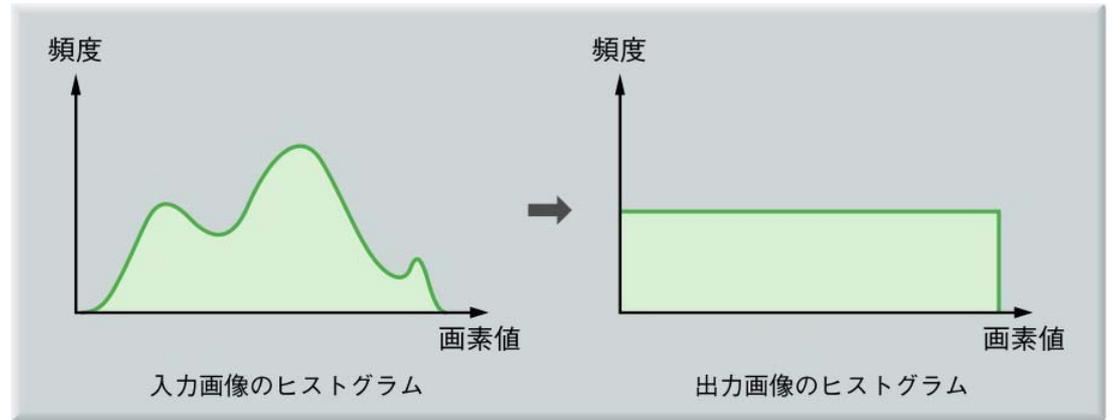
[a] 入力画像



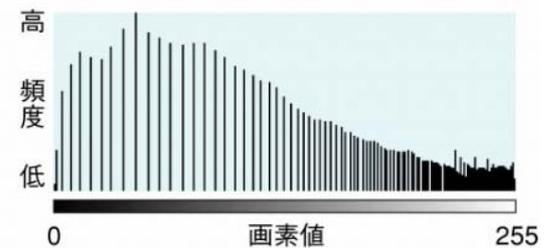
[b] 出力画像

ヒストグラム平坦化

- ヒストグラムの分布が集中している部分を引き伸ばす
ヒストグラム平坦化によってコントラストを上げることができる



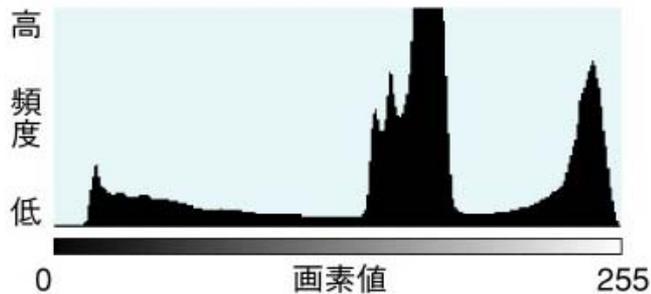
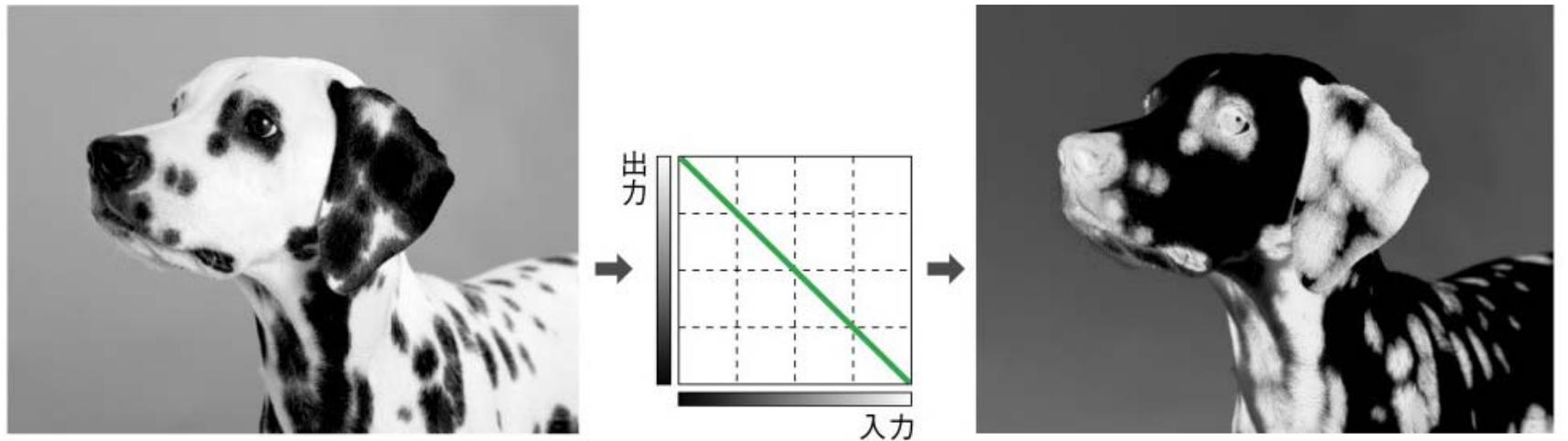
[a] 入力画像



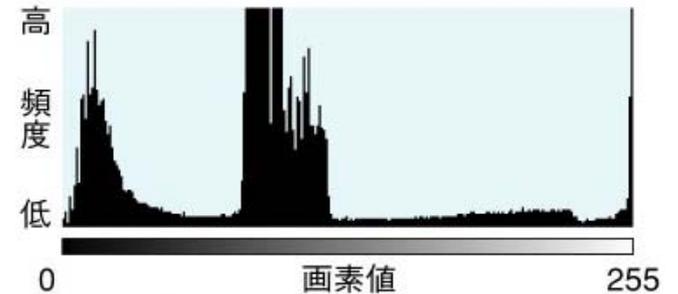
[b] 出力画像

濃淡の反転

- ヒストグラムの分布を折り返す処理はネガ・ポジ反転と呼ばれる



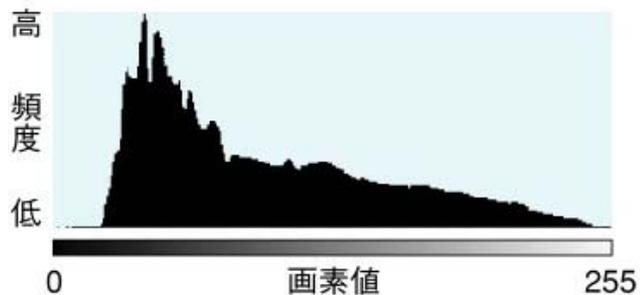
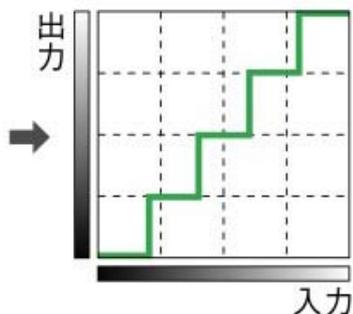
[a] 入力画像



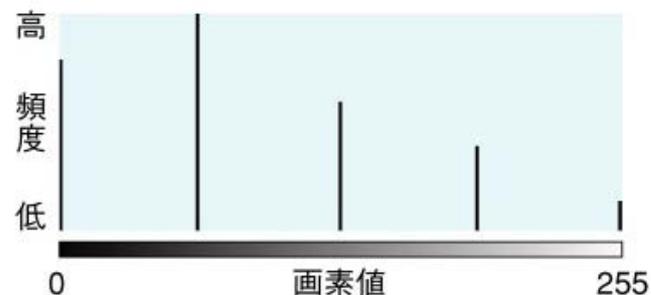
[b] 出力画像

ポスタリゼーション

- 階段状のトーンカーブにより出力画像の値が飛び飛びとなり色の等高線が現れる変換



[a] 入力画像



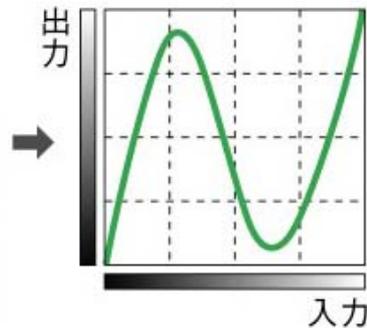
[b] 出力画像

ソラリゼーション

- 画像の濃淡の一部を反転させることで、ネガ画像とポジ画像が混ざったような効果が得られる



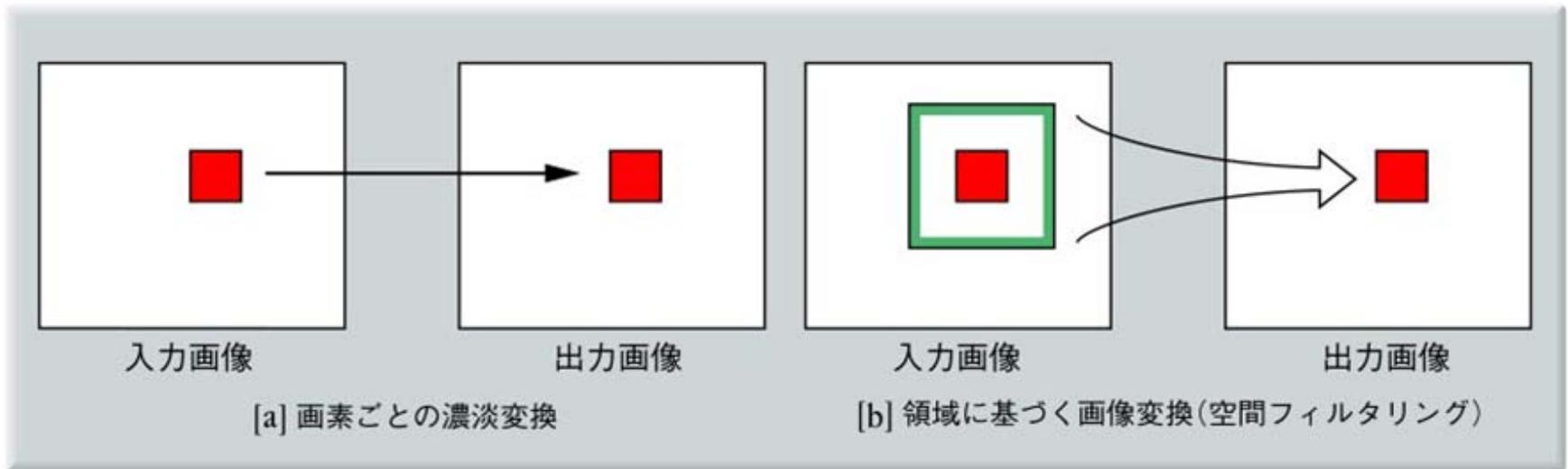
[a] 入力画像



[b] 出力画像

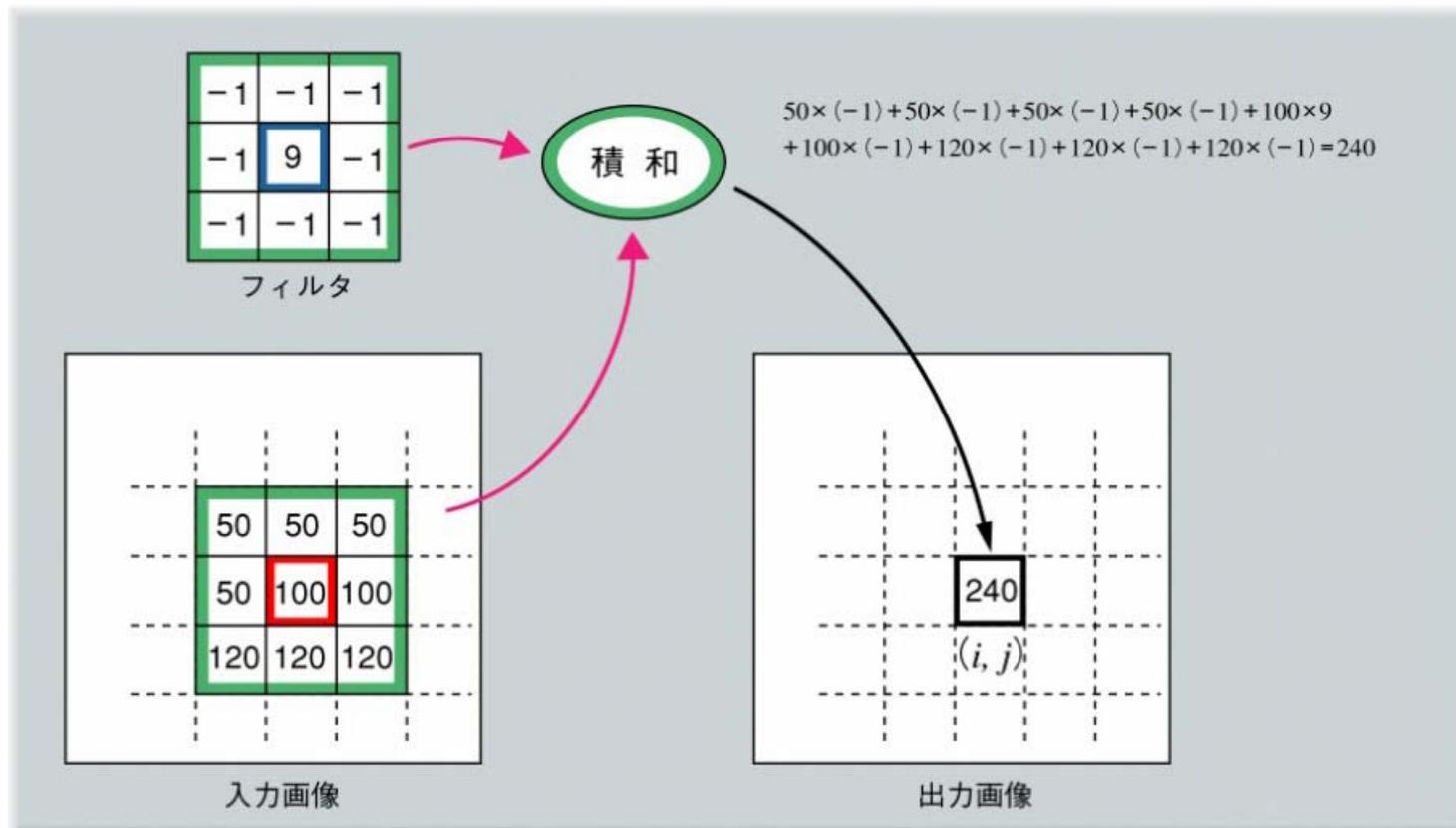
空間フィルタリング

- 出力画像の各画素値を入力画像の対応画素だけから求めるのではなく、対応画素の近傍領域も含めた処理を空間フィルタリングと呼ぶ



線形フィルタ

- フィルタの係数が図の様に3×3の画素に対して与えられているとき、中央を出力画素位置に合わせて3×3の入力画素値とフィルタ係数の積和演算結果を出力画素値とする
- 積和演算以外のフィルタを非線形フィルタと呼ぶ



平滑化

- ピンボケのようになめらかな濃淡変化を与える処理を平滑化と呼ぶ
- フィルタに覆われる領域内の画素値の平均値を求めるため平均化フィルタとも呼ばれる
 - 画素数が大きいフィルタほど効果が強い
- フィルタの中央ほど大きな重みを付ける加重平均は単純平均よりも自然に見える

| | | | | | | | | |
|---|---|---|----|----|----|----|----|----|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 9 | 9 | 9 | 25 | 25 | 25 | 25 | 25 | 25 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 9 | 9 | 9 | 25 | 25 | 25 | 25 | 25 | 25 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 9 | 9 | 9 | 25 | 25 | 25 | 25 | 25 | 25 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 9 | 9 | 9 | 25 | 25 | 25 | 25 | 25 | 25 |

[a] 3×3画素 [b] 5×5画素

平均化フィルタ

| | | | | | | | |
|----|----|----|-----|-----|-----|-----|-----|
| 1 | 2 | 1 | 1 | 4 | 6 | 4 | 1 |
| 16 | 16 | 16 | 256 | 256 | 256 | 256 | 256 |
| 2 | 4 | 2 | 4 | 16 | 24 | 16 | 4 |
| 16 | 16 | 16 | 256 | 256 | 256 | 256 | 256 |
| 1 | 2 | 1 | 6 | 24 | 36 | 24 | 6 |
| 16 | 16 | 16 | 256 | 256 | 256 | 256 | 256 |
| 2 | 4 | 2 | 4 | 16 | 24 | 16 | 4 |
| 16 | 16 | 16 | 256 | 256 | 256 | 256 | 256 |
| 1 | 2 | 1 | 1 | 4 | 6 | 4 | 1 |
| 16 | 16 | 16 | 256 | 256 | 256 | 256 | 256 |

[a] 3×3画素 [b] 5×5画素

加重平均による平均化フィルタ



[a] 入力画像



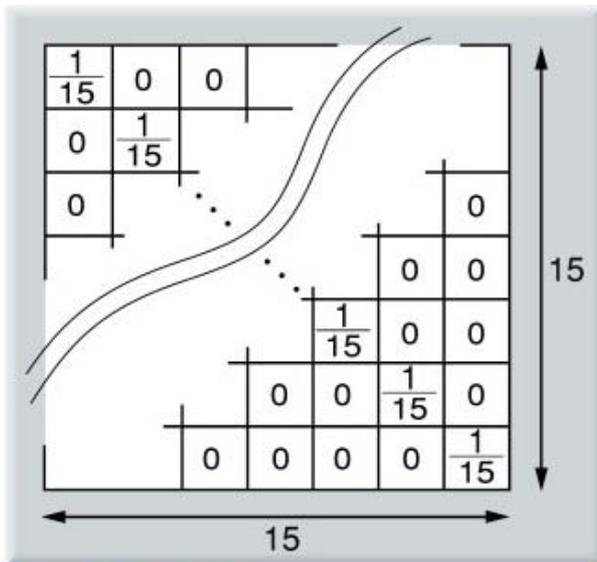
[b] 平均化フィルタ(3×3画素)の適用結果



[c] 平均化フィルタ(5×5画素)の適用結果

一方向の平滑化

- 平滑化を特定の方向に限って行うフィルタでは、一方向に流れるような効果が得られる



[a] フィルタの例



[b] 入力画像



[c] [a]の適用結果

メディアンフィルタ

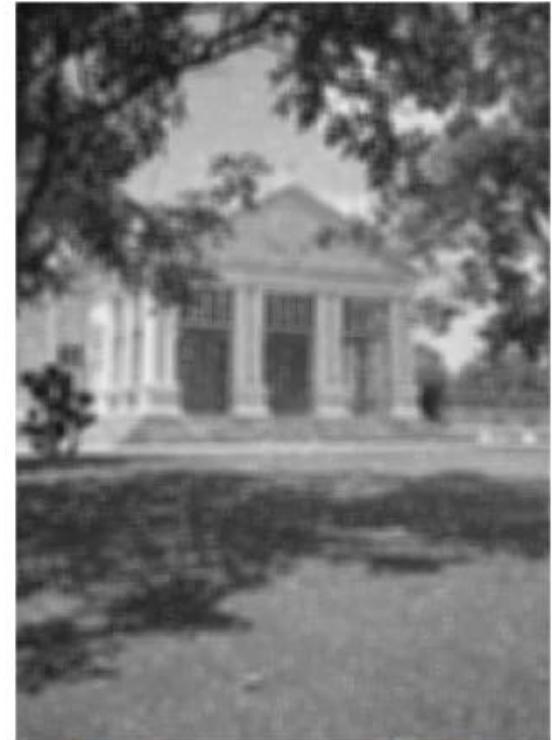
- 平均の代わりに、ある領域内の中央値を出力するものをメディアンフィルタと呼ぶ
- 平均化のようにエッジがなまることなくスパイク上のノイズが軽減される



[a] 入力画像



[b] メディアンフィルタの適用結果



[c] 平均化フィルタの適用結果

エッジ検出

- 注目画素と隣の画素の差を出力する処理は、連続関数の場合の微分に相当するため微分フィルタと呼ばれ、画像のエッジ検出に用いられる
- 横方向の差分を $\Delta_x f$ 、縦方向の差分を $\Delta_y f$ とするとき次の式で方向に依存しないエッジが得られる

$$\sqrt{(\Delta_x f)^2 + (\Delta_y f)^2}$$

- エッジ検出にはソーベルフィルタもよく用いられる

| | | |
|---|----|---|
| 0 | 0 | 0 |
| 0 | -1 | 1 |
| 0 | 0 | 0 |

[a] 微分フィルタ横方向



[b] 入力画像



[c] [a]の適用結果

| | | | | | |
|----|---|---|----|---|----|
| -1 | 0 | 1 | 1 | 2 | 1 |
| -2 | 0 | 2 | 0 | 0 | 0 |
| -1 | 0 | 1 | -1 | 2 | -1 |

[a] 横方向

[b] 縦方向

| | | |
|---|----|---|
| 0 | 1 | 0 |
| 0 | -1 | 0 |
| 0 | 0 | 0 |

[d] 微分フィルタ縦方向



[e] [d]の適用結果



[f] 微分の大きさ

ソーベルフィルタ

微分フィルタの適用例

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は鮮鋭化フィルタと呼ばれる

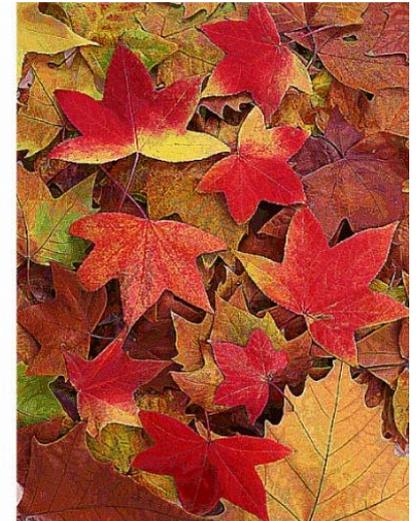
| | | | | | |
|----|----|----|----|----|----|
| 0 | -1 | 0 | -1 | -1 | -1 |
| -1 | 5 | -1 | -1 | 9 | -1 |
| 0 | -1 | 0 | -1 | -1 | -1 |

[a] [b]

鮮鋭化フィルタの例



[a] 入力画像



[b] 鮮鋭化フィルタの適用結果



[a] 入力画像



[b] 鮮鋭化フィルタの適用結果

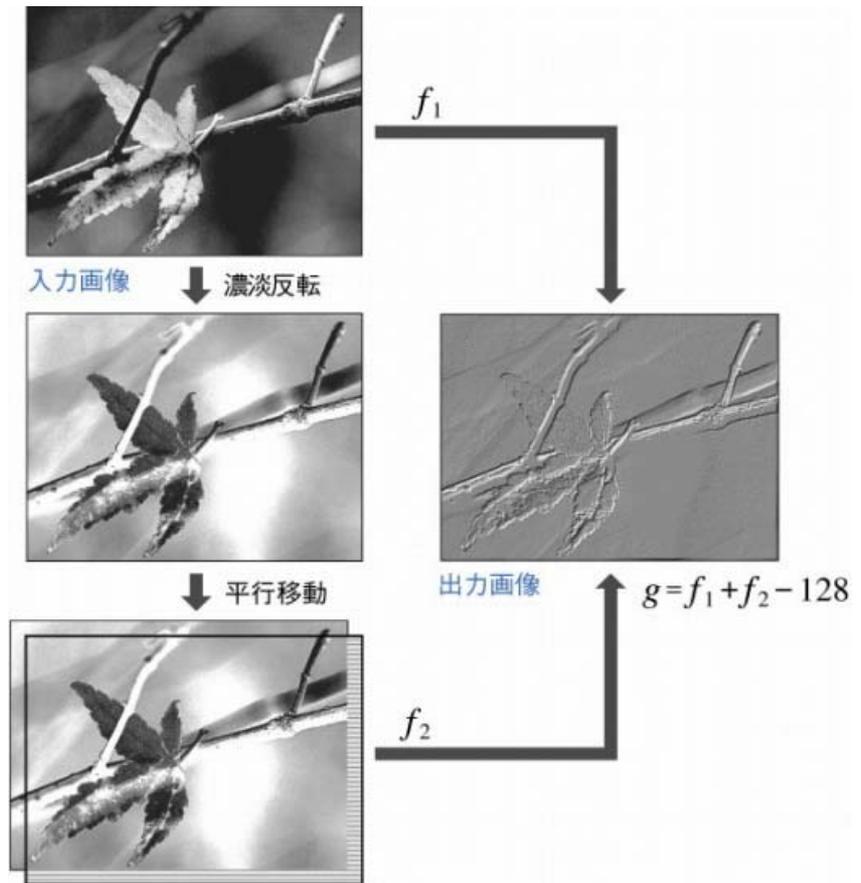
エンボス処理

- 画像の濃度差を利用して画像の一部を浮き出たせる処理
- 画素値を0~255とするとき, 入力画像 f_1 の濃淡を反転して数画素ずらした画像 f_2 から次式でエンボス画像を求める

$$g = f_1 + f_2 - 128$$

| | | |
|---|---|----|
| 1 | 0 | 0 |
| 0 | 0 | 0 |
| 0 | 0 | -1 |

エンボス処理のフィルタ



エンボス画像の生成