コンピュータグラフィックス

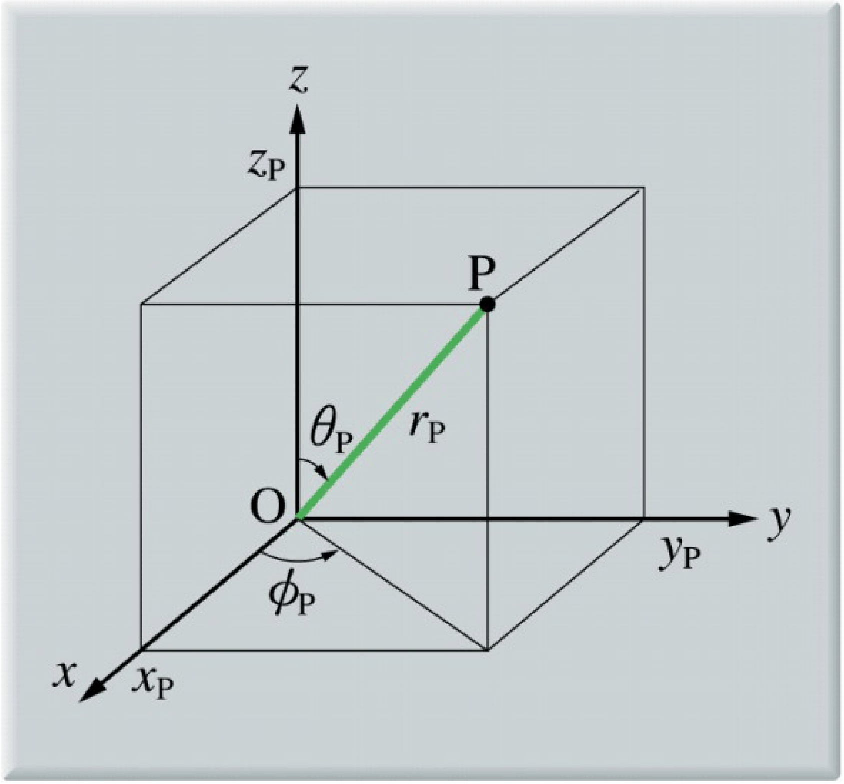
3. 3次元変換と投影

佐藤証 西9-613
akashi.satoh@uec.ac.jp

教科書P.26-40

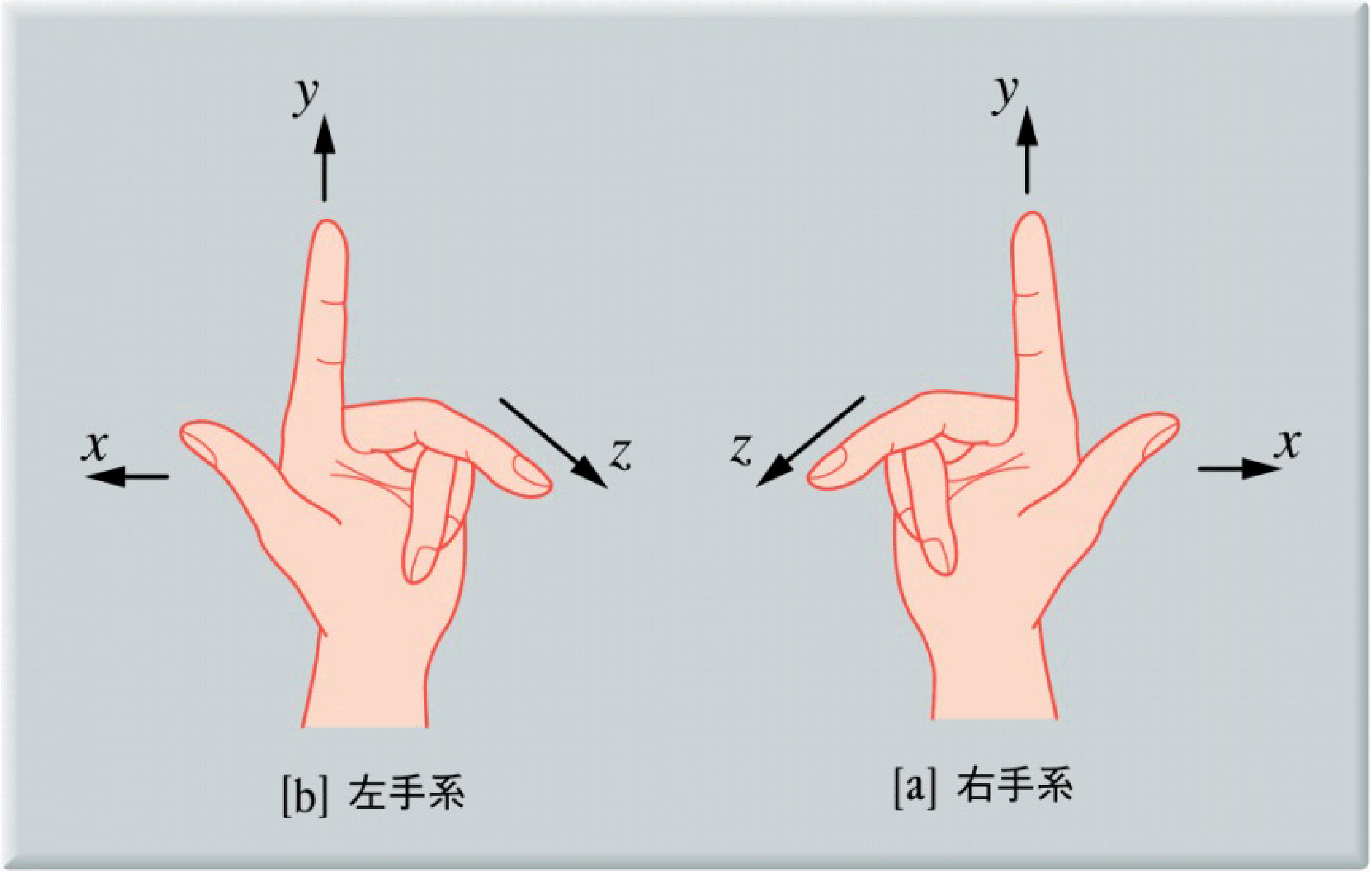
3次元直行座標系と極座標系

- 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$(r_P, \theta_P, \varphi_P)$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



円柱座標系

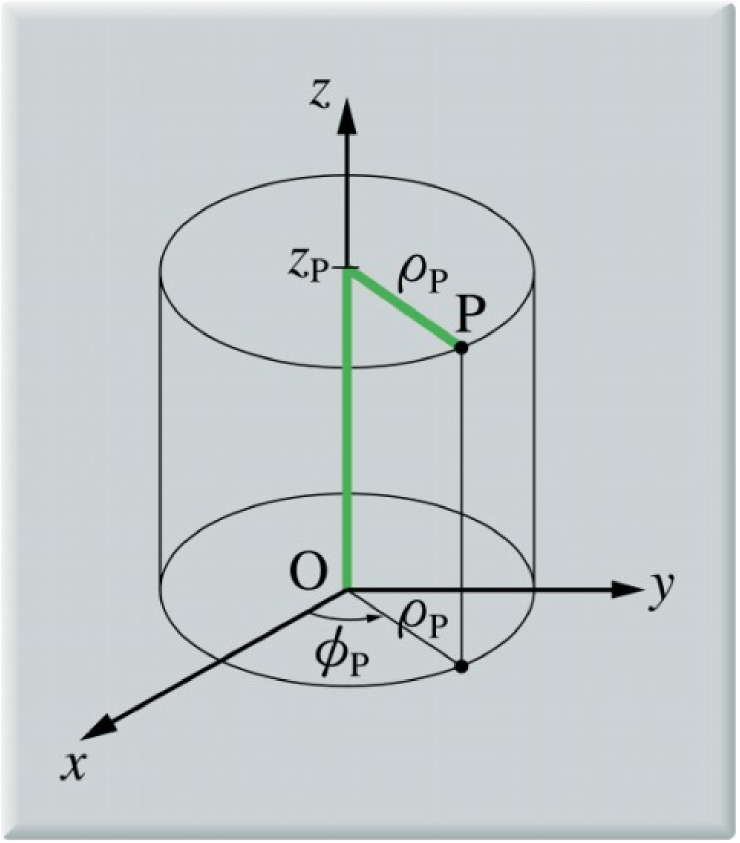
- 円柱座標系は座標を、z値、z軸からの距離ρ、z軸周りの角度φの組(z, ρ, φ)によって表現
  - 円柱座標系は2つの長さz, ρと1つの角度φで表す
  - 極座標系は１つの長さrと２つの角度θ, φで表す

円柱座標系

$$
\begin{cases}
x_P = \rho_P \cos \varphi_P \\
y_P = \rho_P \sin \varphi_P \\
z_P = z_P
\end{cases}
$$

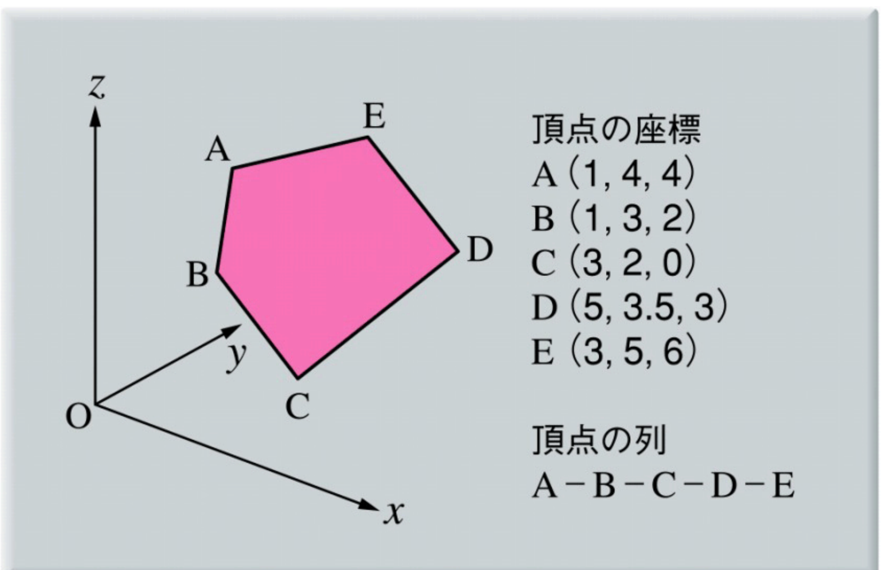
極座標系

$$
\begin{cases}
x_P = r_P \sin \theta_P \cos \varphi_P \\
y_P = r_P \sin \theta_P \sin \varphi_P \\
z_P = r_P \cos \theta_P
\end{cases}
$$

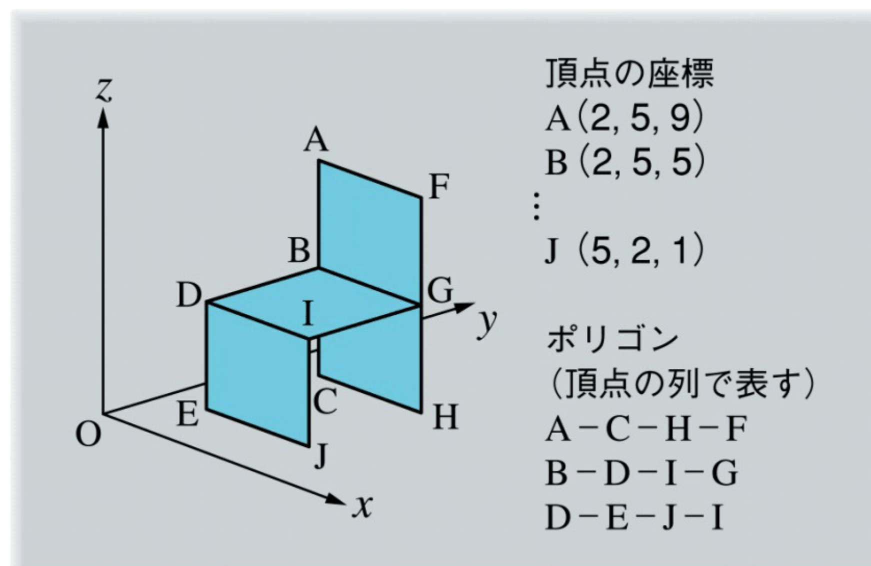


簡単なモデリング

- ３次元図形の形状を数値的な記述をモデリング、形状データを形状モデルと呼ぶ
- ポリゴンが形状モデルに広く用いられている
  - 頂点座標とその順序で表される
  - 頂点が4個以上の場合は平面上にあると限らない
  - 頂点が3個の三角形がよく用いられる



頂点５個のポリゴンの例



ポリゴンによる形状モデル

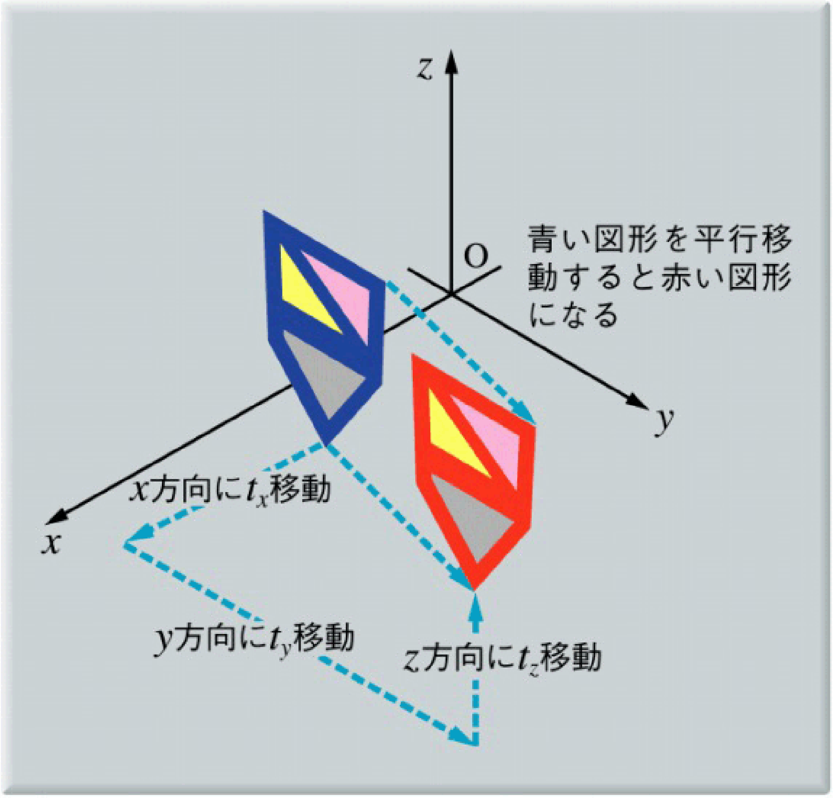
同次座標と3次元の基本変換

- 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

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

$$= T(t_x, t_y, t_z) \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

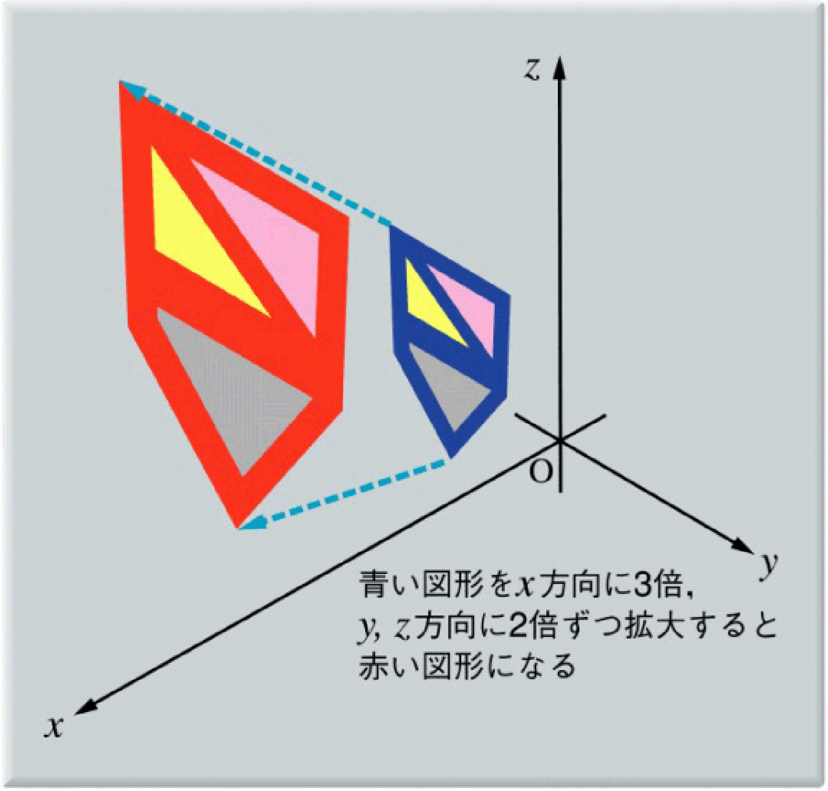


拡大・縮小

- 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

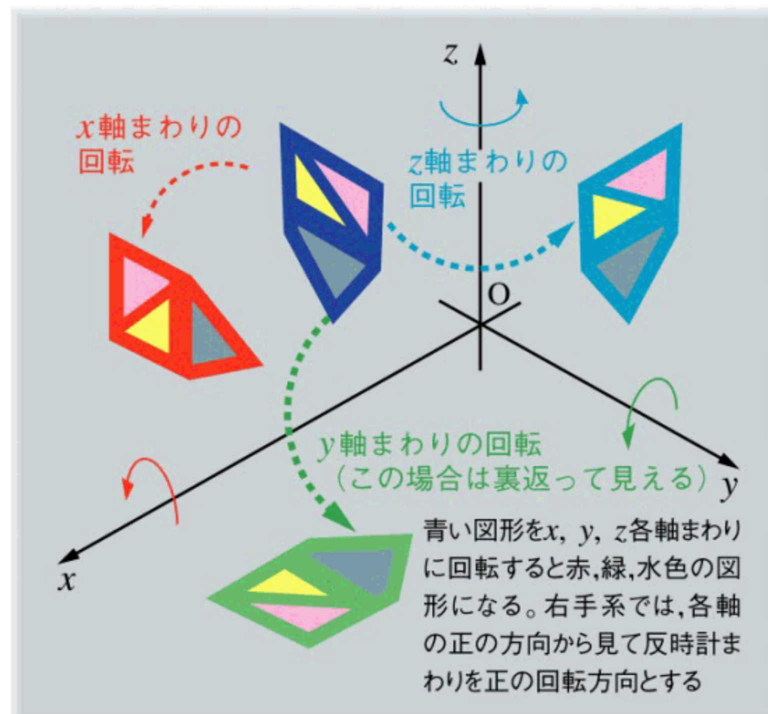
$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = R_x(\theta) \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

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

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

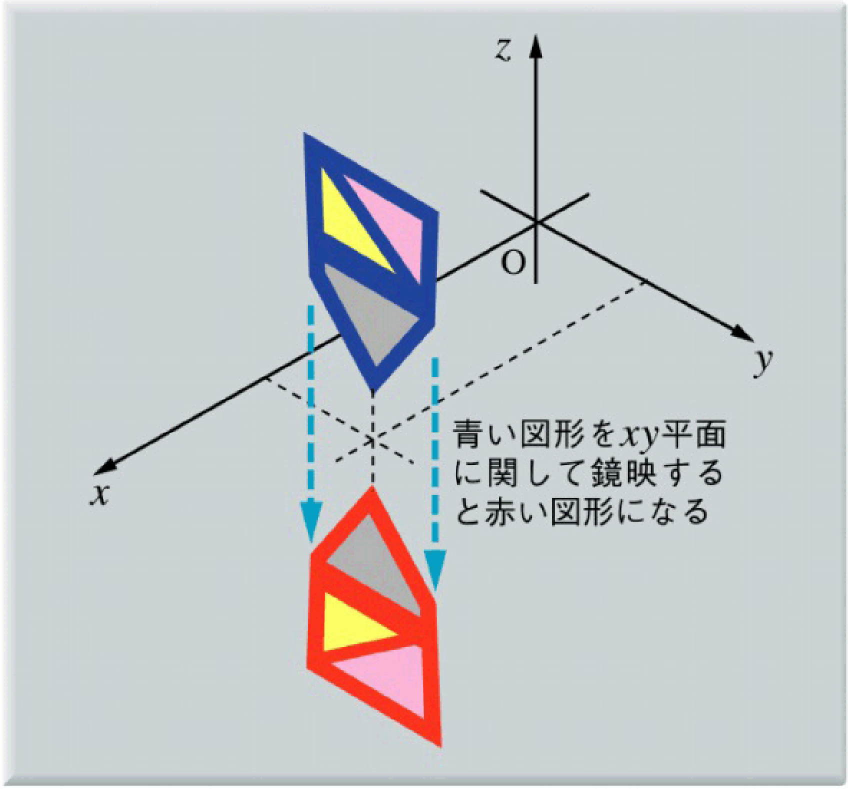


鏡映

- 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

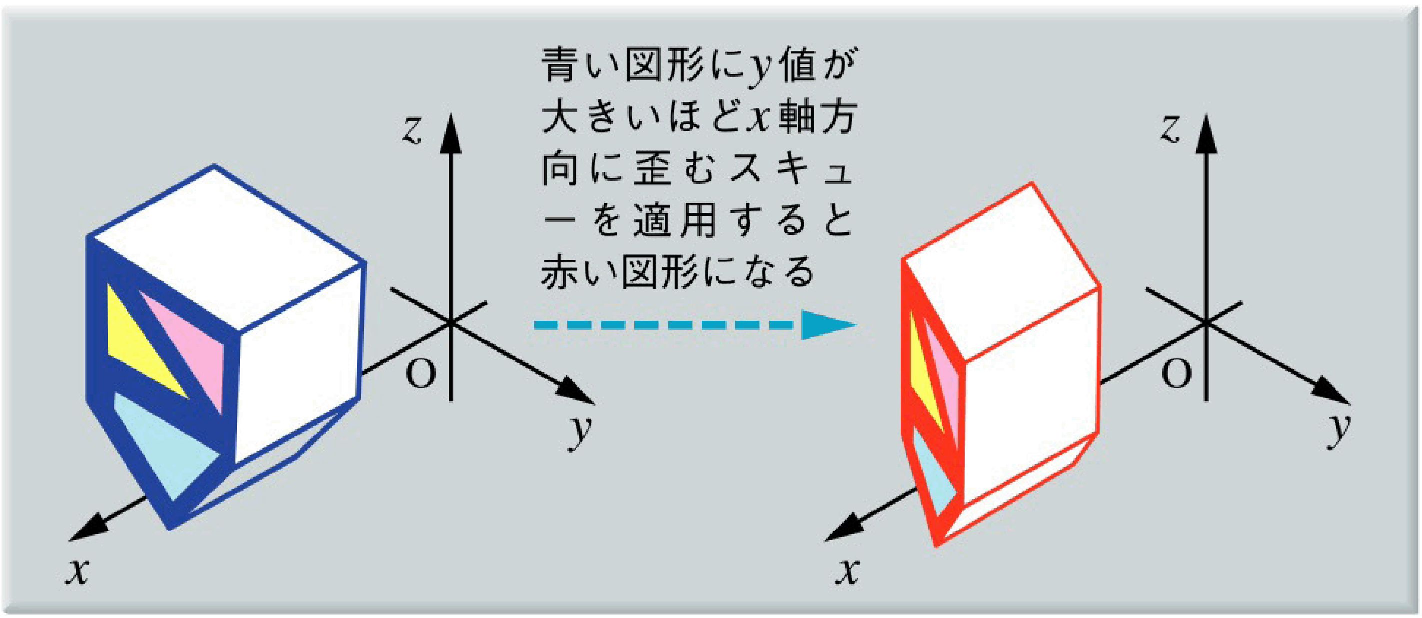
$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$



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（座標値の変化量が大きい）

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



3次元アフィン変換

- ２次元と同様に直線は直線に変換され、直線状の距離の比は保存される

(解釈1) 1つの座標系の中での点(x, y, z)から点(x', y', z')への移動

(解釈2) xyz座標系を$x'y'z'$座標系に動かしたときの同じ点のxyz座標から$x'y'z'$座標系への変換

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

3次元座標系における合成変換

- 2次元座標系と同様に変換$A_1, A_2, \cdots, A_n$を順番に行う合成変換$A=A_n\cdots A_2A_1$が定義できる

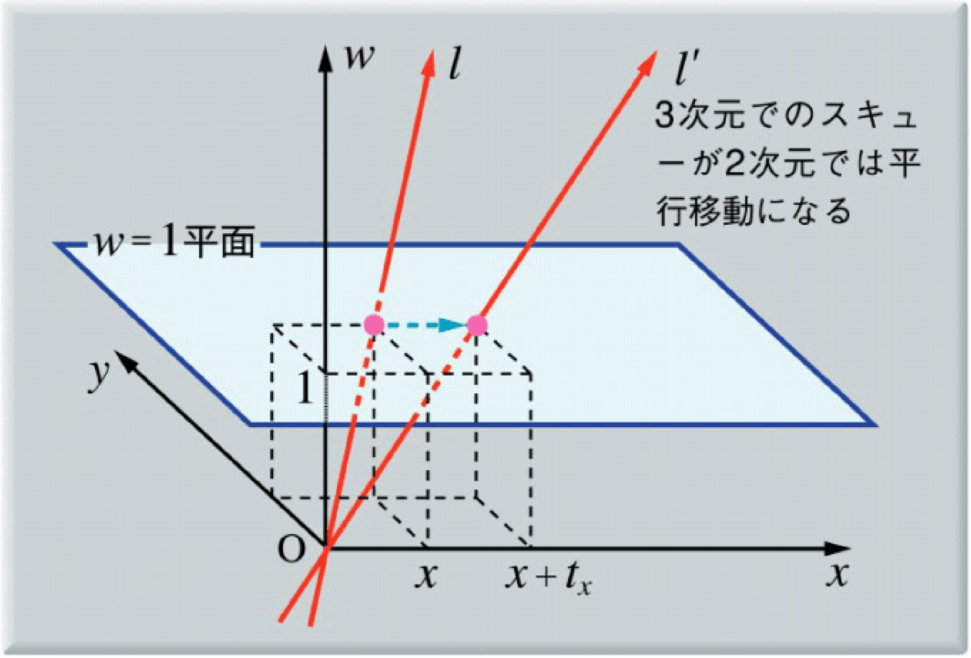
点(x_0, y_0, z_0)を中心とする拡大・縮小

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = T(x_0, y_0, z_0)S(s_x, s_y, s_z)T(-x_0, -y_0, -z_0)\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

同次座標による平行移動の行列表現

- ２次元同次座標系のwを第３軸と見なした３次元通常座標系を考える
- 原点Oと$(x, y, 1)$を結ぶ直線 l と、$(x+t_x, y, 1)$を結ぶ直線 l'への変換はwが同じ点同志で考えると、 w値が大きくなるほどx値を大きくするスキューとなる

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



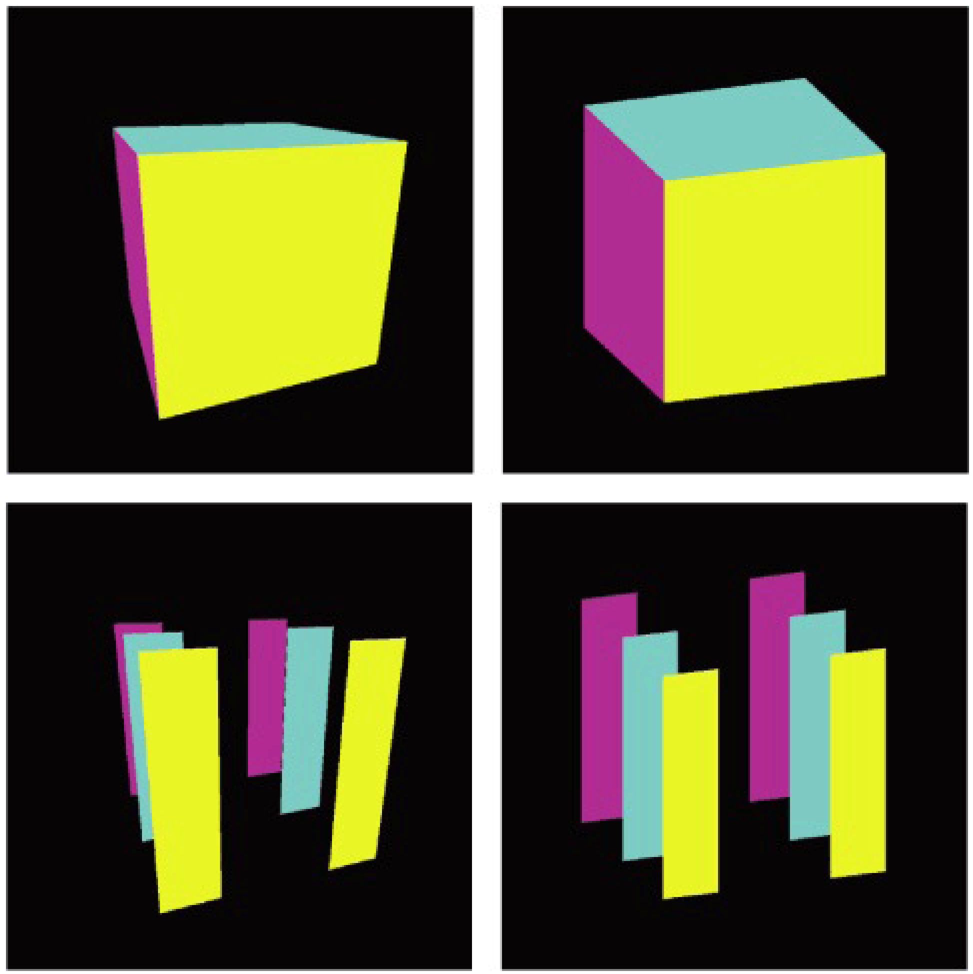
投影

- 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

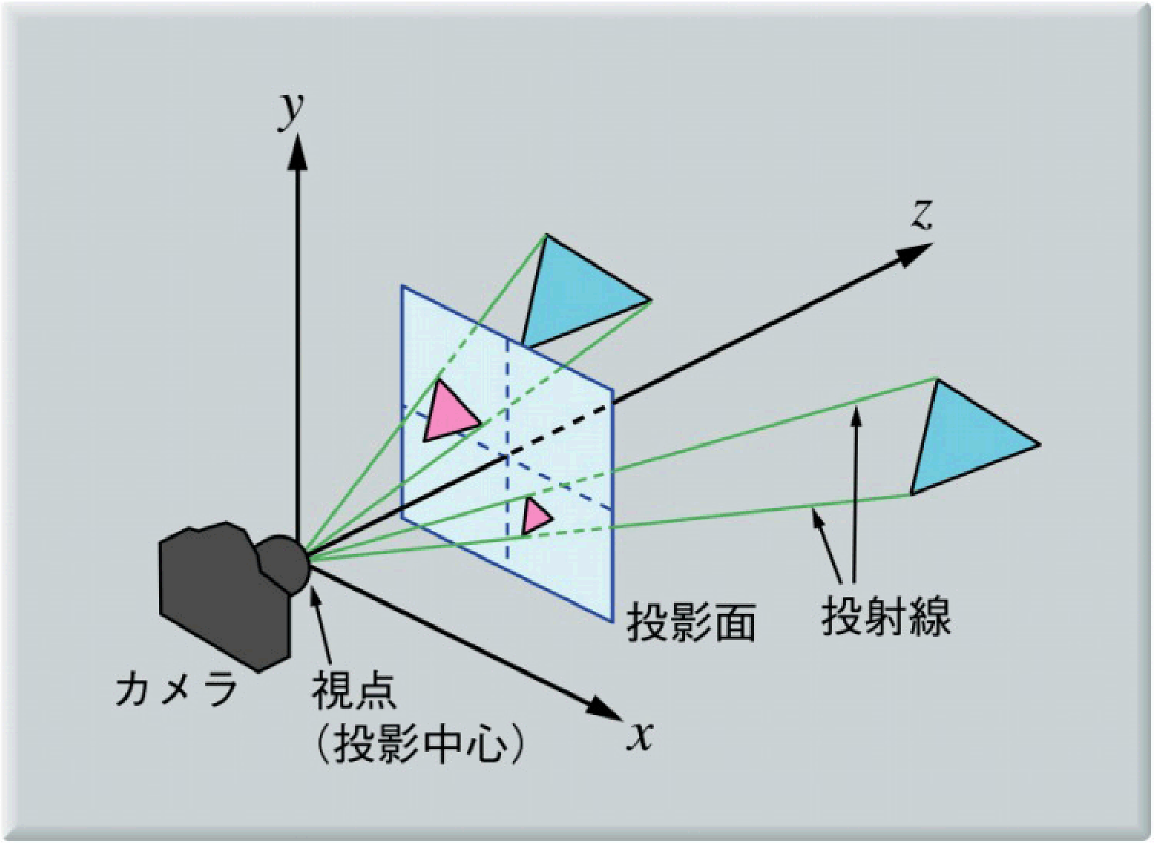
立方体

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次元図形の各店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

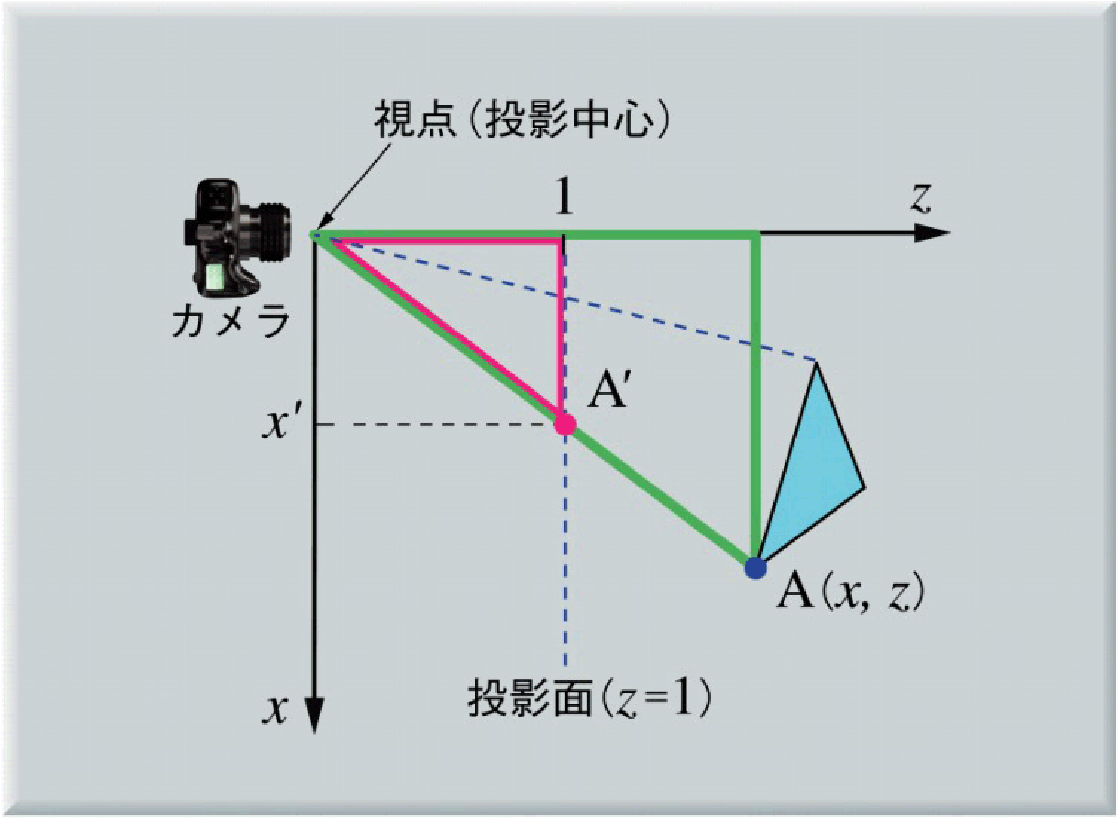


透視投影

- 遠くのもの(z値が大きいもの)が近くのもの(z値が小さいもの)より小さく描かれ遠近感がある
- 写実的な画像の作成や映画・ゲーム等に利用
- 平行線が歪むなど、ものの形の正確な把握には不向き

投影面 (z=1)への変換

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

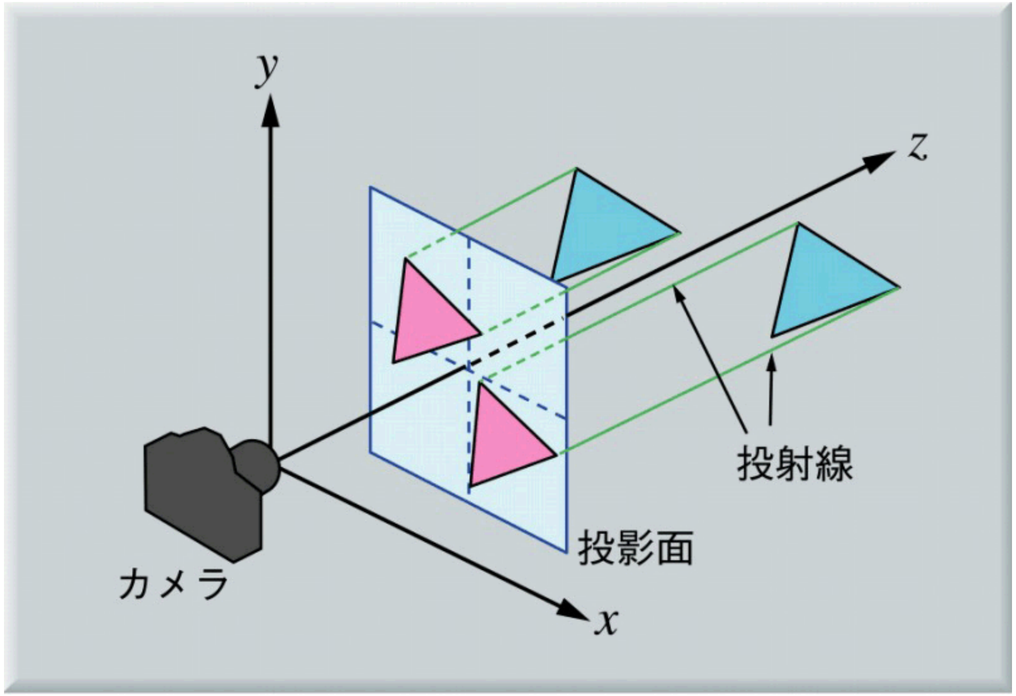


平行投影

- 3次元図形の各店からの当社線を平行に投射面におろす
- 視点を無限遠点に置いた投影
- 遠くのものと近くのものを同じ大きさで描画
- 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

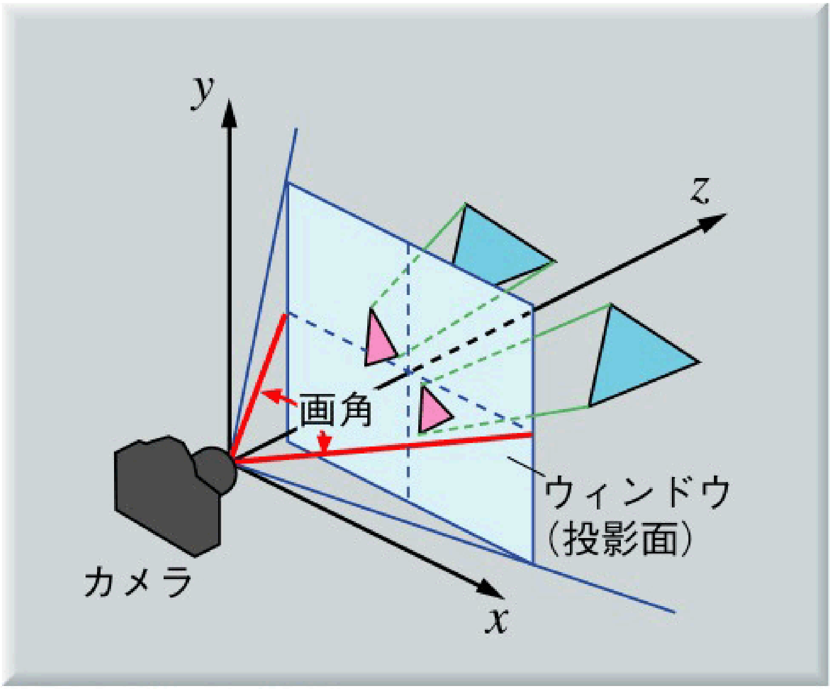
投影面 (z=k：任意) への変換

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

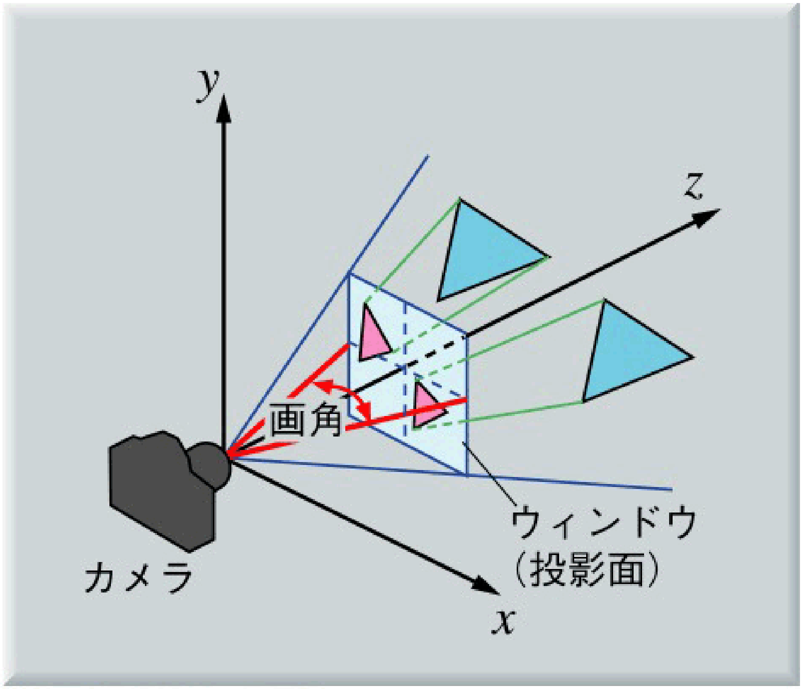


ビューボリュームと投影

- 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



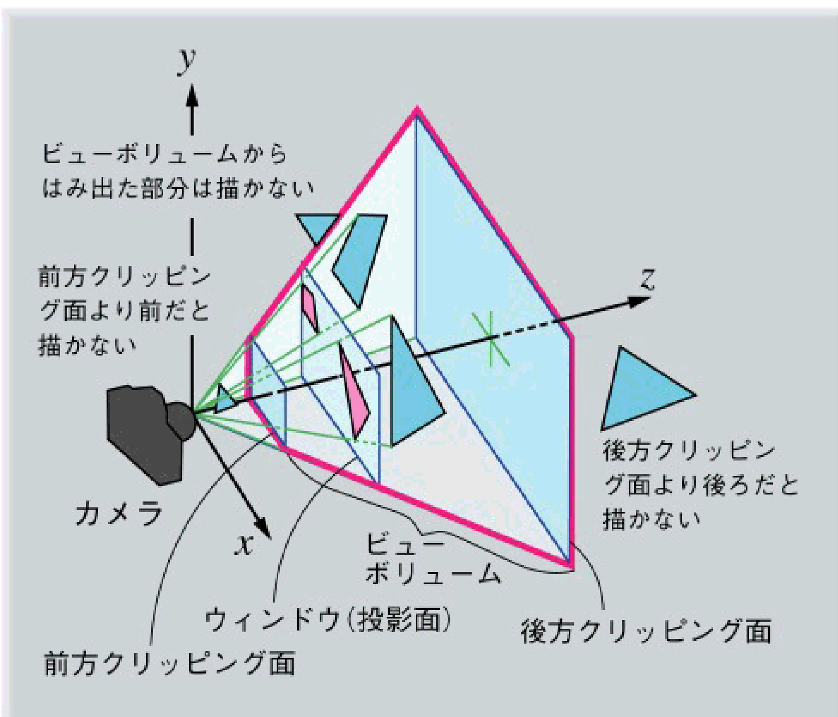
[a] 画角が大きい場合



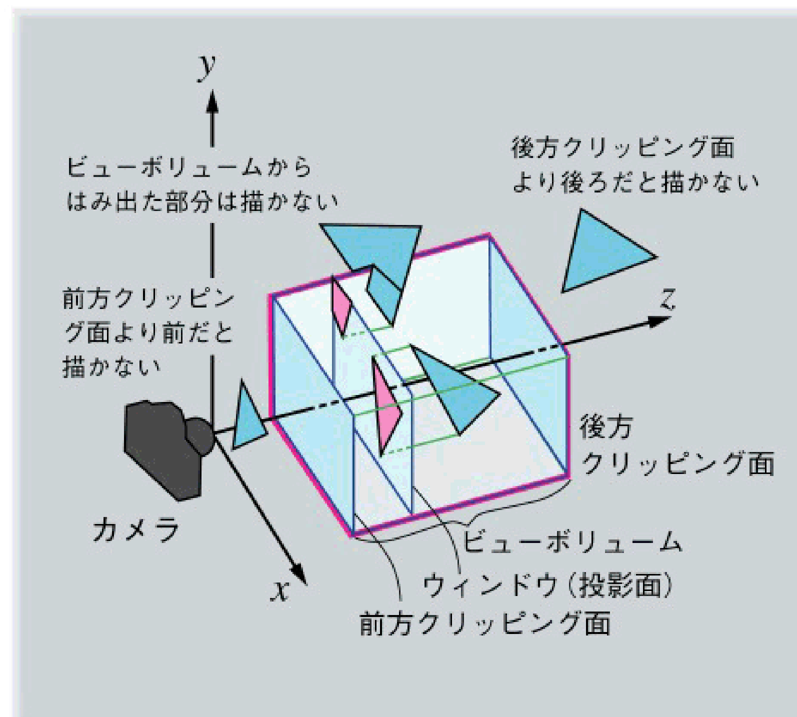
[b] 画角が小さい場合

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

ビューボリュームの投影の手順

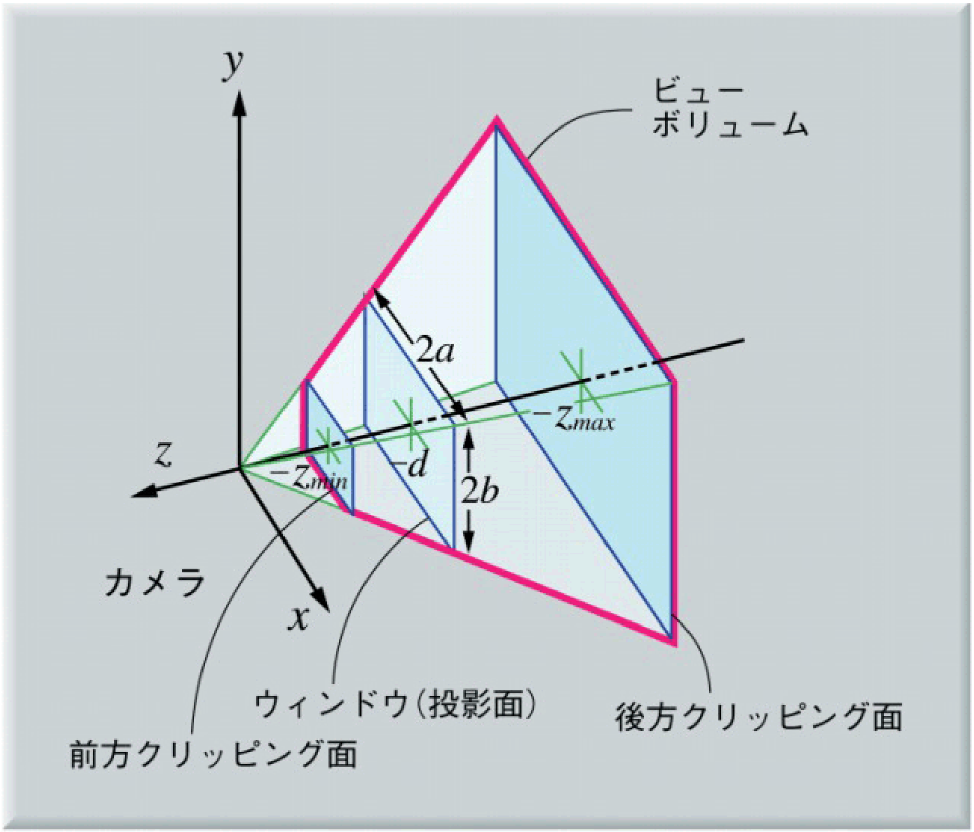
① 視点から遠いほど奥行の値を大きくするため、もとの座標系(カメラ座標系)を右手系から左手系に変換

② ビューボリュームを正規化ボリュームに変換

③ 正規化ビューボリュームを透視投影

透視投影の計算法

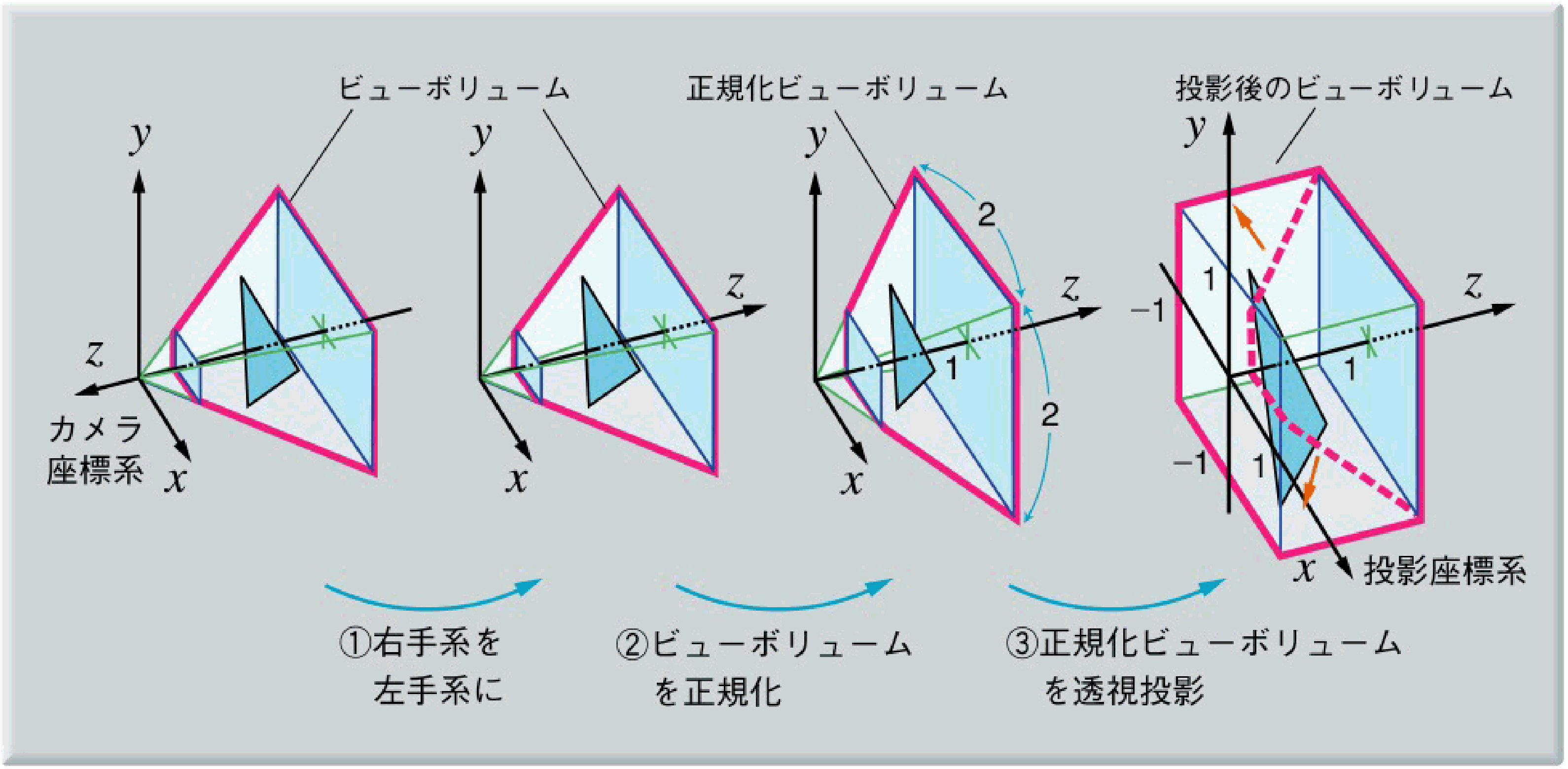
- 定義
  - 視点からウィンドウの距離:d
  - ウィンドウの大きさ横(x)方向：2a、縦(y)方向：2b
  - 視点から前方/後方クリッピング面までの距離：z_{min}/z_{max}



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)

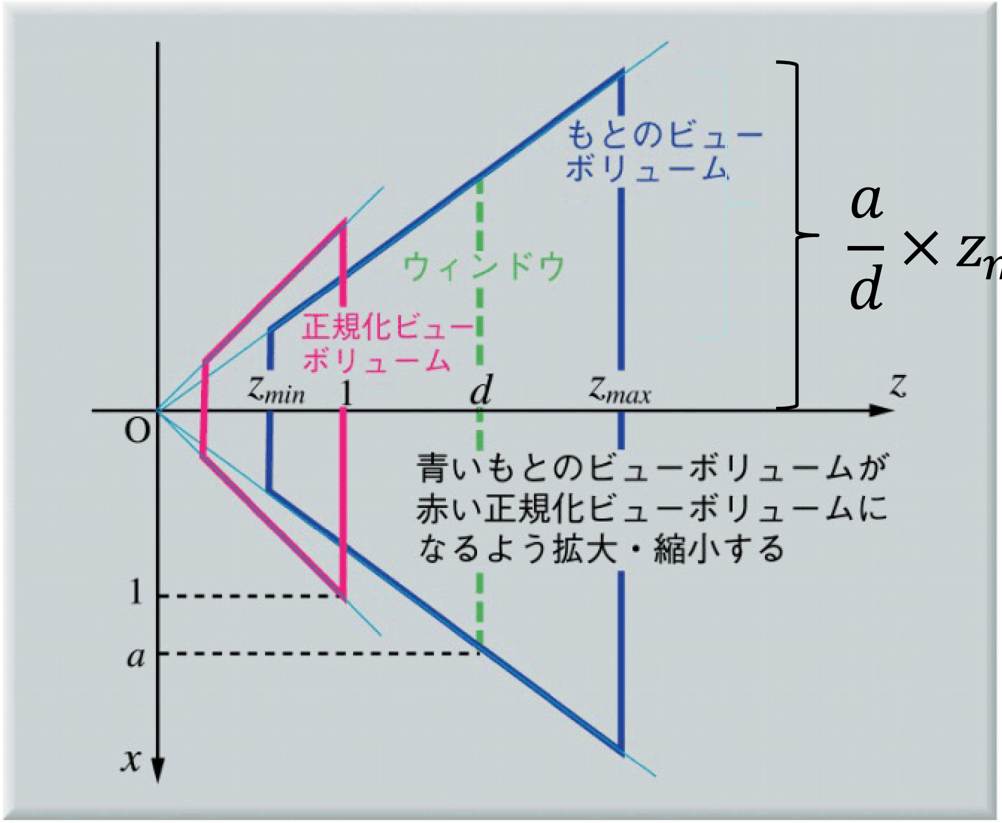
② ビューボリュームを正規化ビューボリュームに変換
- 後方リッピング面がz=1に、z=1でのビューボリュームの範囲が -1≦x≦-1、-1≦y≦-1の正方形になるよう拡大・縮小



透視投影の計算法

② ビューボリュームを正規化ビューボリュームに変換

$$S\left(\frac{d}{az_{max}}, \frac{d}{bz_{max}}, \frac{1}{z_{max}}\right)$$



$$\frac{a}{d} \times z_{max}$$

透視投影の計算法

③ 正規化ビューボリュームを透視投影

- 四角錐台を直方体に変形

$\tilde{z}_{min} = \dfrac{z_{min}}{z_{max}}$ とするとき

$$\begin{pmatrix} X' \\ Y' \\ Z' \\ W' \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \dfrac{1}{1-\tilde{z}_{min}} & -\dfrac{\tilde{z}_{min}}{1-\tilde{z}_{min}} \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = P(\tilde{z}_{min}) \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

①~③をまとめた透視投影変換

$$\begin{pmatrix} X' \\ Y' \\ Z' \\ W' \end{pmatrix} = P(\tilde{z}_{min}) S\left(\frac{d}{az_{max}}, \frac{d}{bz_{max}}, \frac{1}{z_{max}}\right) s(1,1,-1) \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

平行投影の計算法

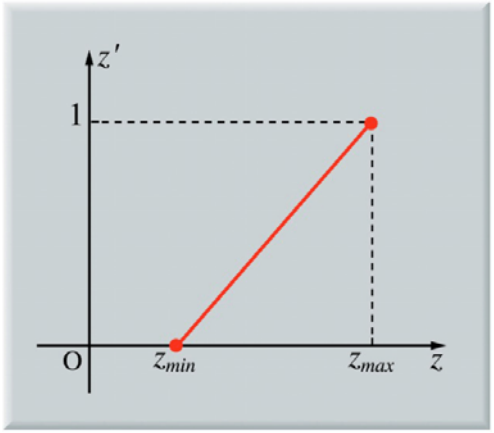
① カメラ座標系を右手系から左手系に変換

- 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

- 奥行zは$z_{min} \leqq z \leqq z_{max}$を$0 \leqq z' \leqq 1$の範囲に変換

$$\begin{pmatrix} x' \\ y' \\ z' \\ w' \end{pmatrix} = \begin{pmatrix} \frac{2}{x_{max} - x_{min}} & 0 & 0 & -\frac{x_{max} + x_{min}}{x_{max} - x_{min}} \\ 0 & \frac{2}{y_{max} - y_{min}} & 0 & -\frac{y_{max} + y_{min}}{y_{max} - y_{min}} \\ 0 & 0 & \frac{1}{z_{max} - z_{min}} & -\frac{z_{min}}{z_{max} - z_{min}} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$



平行投影の計算法

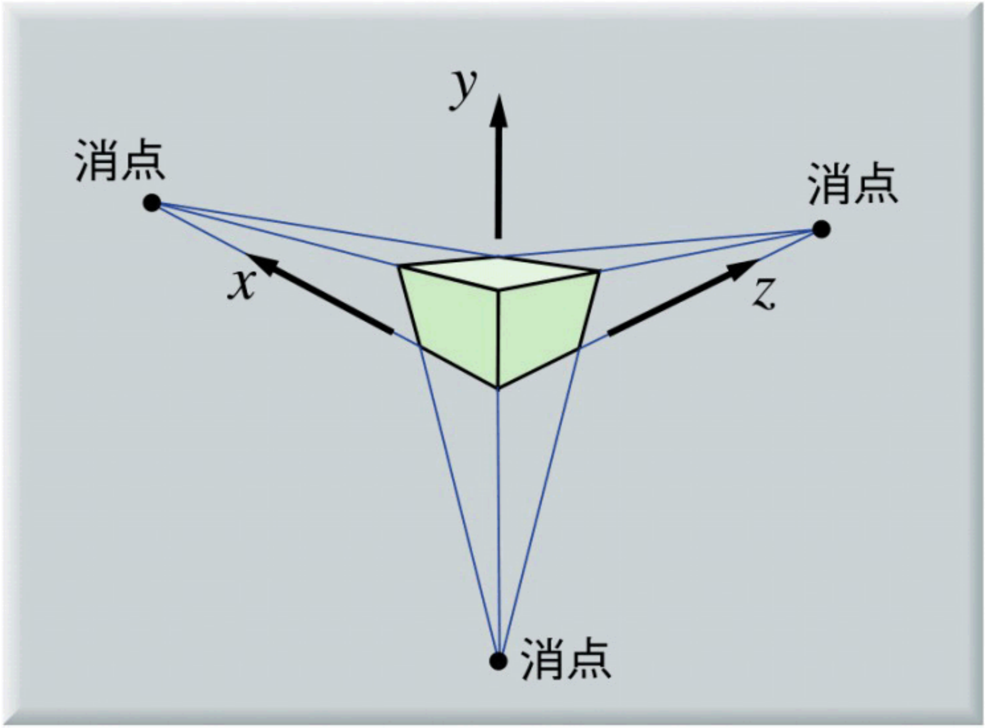
- ①～③をまとめた透視投影変換

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{2}{x_{max}-x_{min}} & 0 & 0 & -\frac{x_{max}+x_{min}}{x_{max}-x_{min}} \\ 0 & \frac{2}{y_{max}-y_{min}} & 0 & -\frac{y_{max}+y_{min}}{y_{max}-y_{min}} \\ 0 & 0 & \frac{1}{z_{max}-z_{min}} & -\frac{z_{min}}{z_{max}-z_{min}} \\ 0 & 0 & 0 & 1 \end{pmatrix} S(1,1,-1) \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

$$= \begin{pmatrix} \frac{2}{x_{max}-x_{min}} & 0 & 0 & -\frac{x_{max}+x_{min}}{x_{max}-x_{min}} \\ 0 & \frac{2}{y_{max}-y_{min}} & 0 & -\frac{y_{max}+y_{min}}{y_{max}-y_{min}} \\ 0 & 0 & \frac{-1}{z_{max}-z_{min}} & -\frac{z_{min}}{z_{max}-z_{min}} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

消失点とn点透視

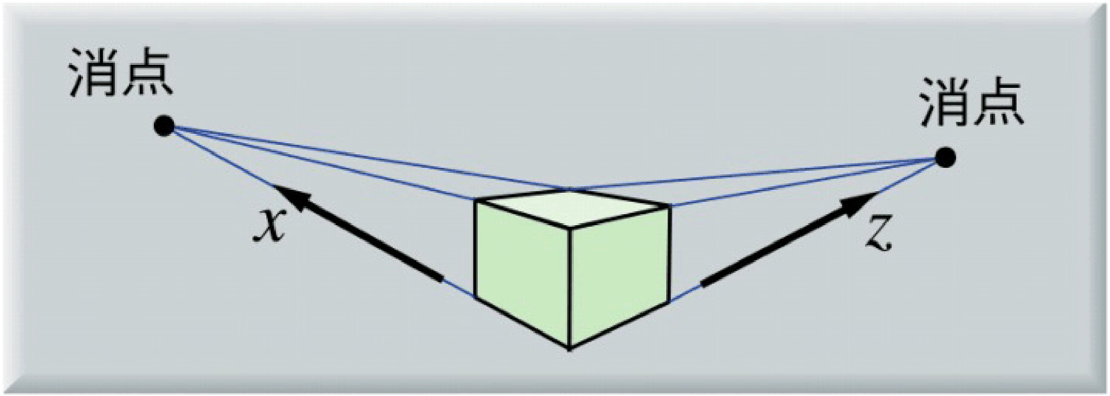
- 透視投影は遠近法とも呼ばれ、平行な直線群は投影図上では消失点に収束する
- 投影面に平行な直線群は消失点を生じない
- 投影面がx,y,zのどの軸にも平行でないとき各軸に平行な直線群はそれぞれ消失点を持ち、3点透視と呼ばれる



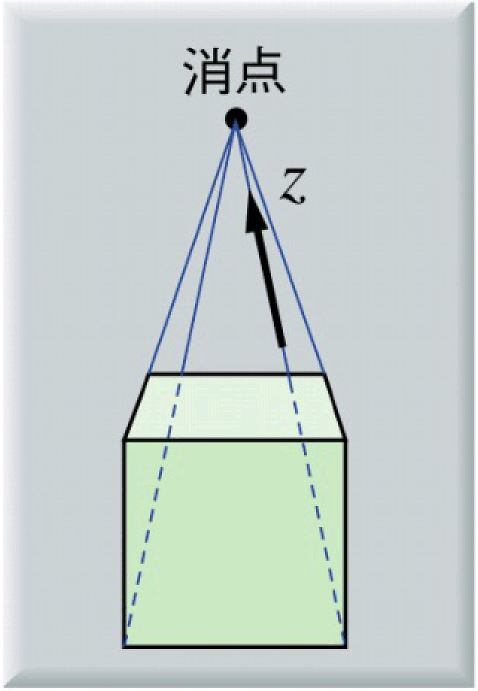
3点透視

消失点とn点透視

- 投影面が1つの軸と並行のとき、残りの2軸に平行な直線群が消失点を持ち、2点透視と呼ばれる
- 投影面が2軸と並行のとき、１軸に平行な直線群が消失点を持ち、1点透視と呼ばれる



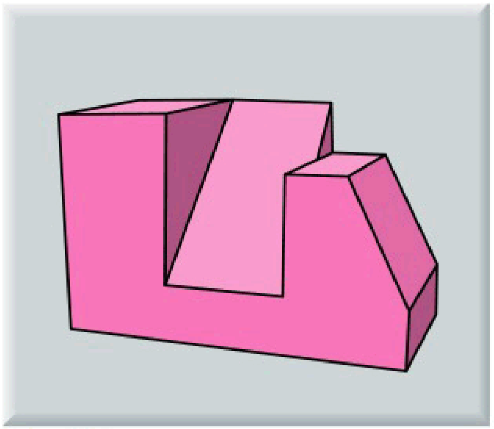
2点透視



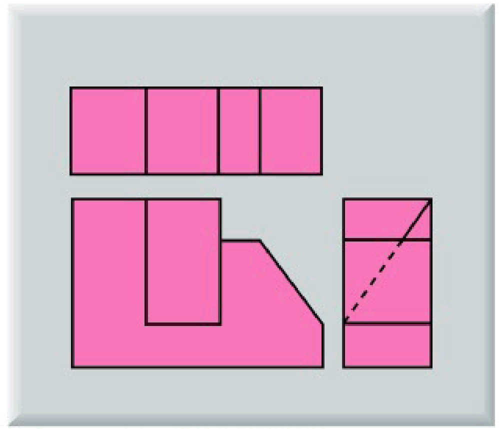
1点透視

平行投影の直投影

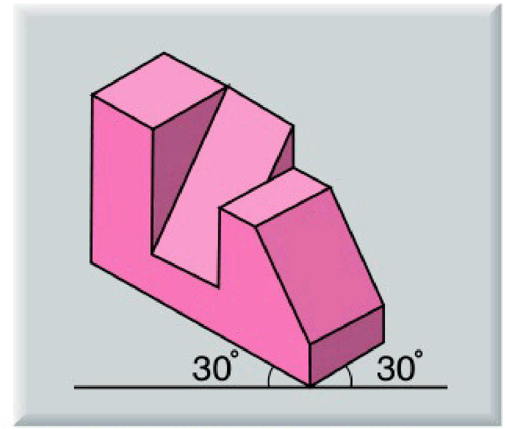
- 平行投影で投射線が投影面に垂直な場合を直投影、垂直でない場合を斜投影と呼ぶ
  - 三面図とアイソメ図は直投影の代表



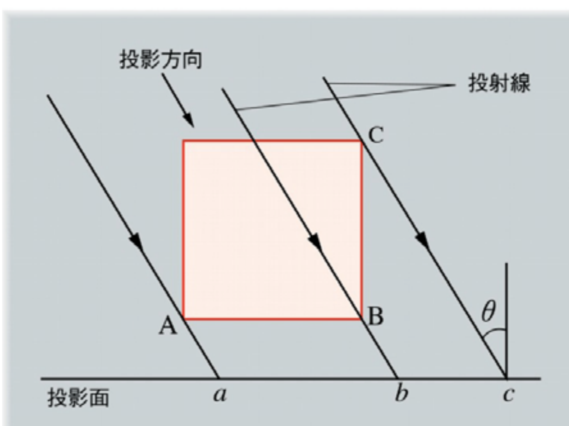
立体



三面図(第三角法)



アイソメ図(等測投影図)



斜投影