

# コンピュータグラフィックス

---

## 第4回

CGのための数学的基礎 1

～ 2次元, 3次元座標系～

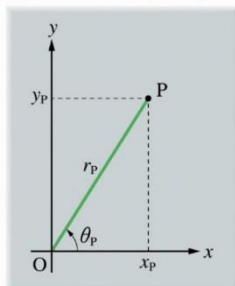
理工学部 兼任講師  
藤堂 英樹

# 本日の講義内容

## ■CGのための数学的基礎 1

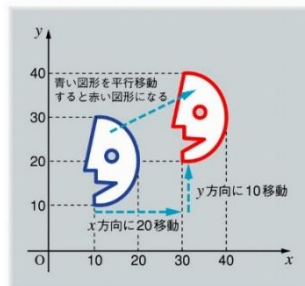
- 2次元座標系
- 3次元座標系

■図2.1—2次元直交座標系と極座標系



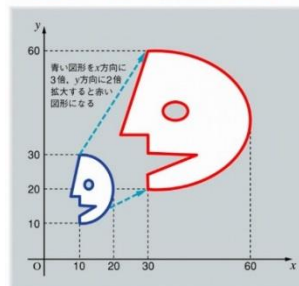
「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会 (CG-ARTS協会)

■図2.3—平行移動の例 ( $t_x = 20, t_y = 10$ )



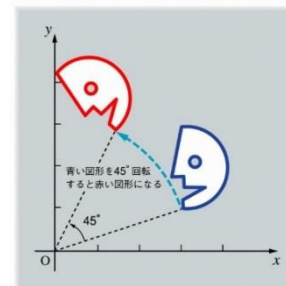
「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会 (CG-ARTS協会)

■図2.4—拡大・縮小の例 ( $s_x = 3, s_y = 2$ )



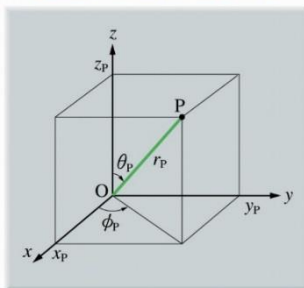
「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会 (CG-ARTS協会)

■図2.5—回転の例 ( $\theta = 45^\circ$ )



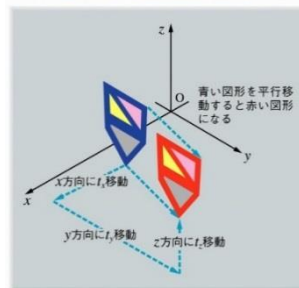
「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会 (CG-ARTS協会)

■図2.15—3次元直交座標系と極座標系



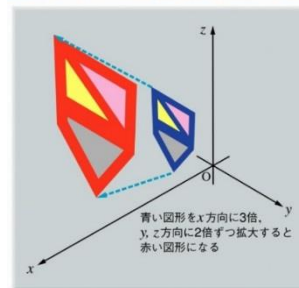
「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会 (CG-ARTS協会)

■図2.20—3次元での平行移動



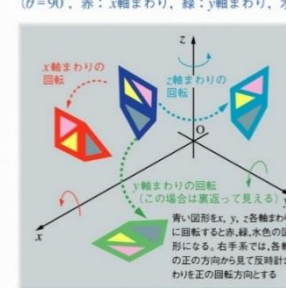
「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会 (CG-ARTS協会)

■図2.21—3次元での拡大・縮小 ( $s_x = 3, s_y = s_z = 2$ )



「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会 (CG-ARTS協会)

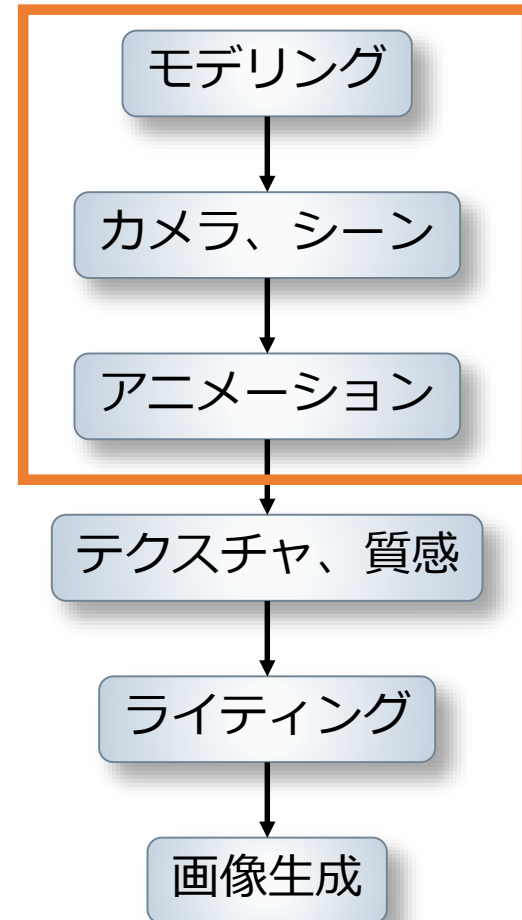
■図2.22—3次元での各軸まわりの回転 ( $\theta = 90^\circ$ , 赤: x軸まわり, 緑: y軸まわり, 水色: z軸まわり)



「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会 (CG-ARTS協会)

# CG制作の主なワークフロー

## ■3DCGソフトウェアの場合



# 座標変換の用途

- キャラクターの配置
  - 位置, 大きさの調整
- キャラクターのポーシング
  - 各関節角度の設定
- カメラの設定
  - 視点変更, ズームアップ

少しやってみましょう!

# 今回使用するソフトウェア

## ■Unity: Game Engine

- 無料バージョンがある
- アセット(データ等のリソース)が豊富

## ■Unityちゃん: Unity用データ

- 無料で使える
- 2次創作が可能



このコンテンツは、  
『[ユニティちゃんライセンス](#)』で  
提供されています。

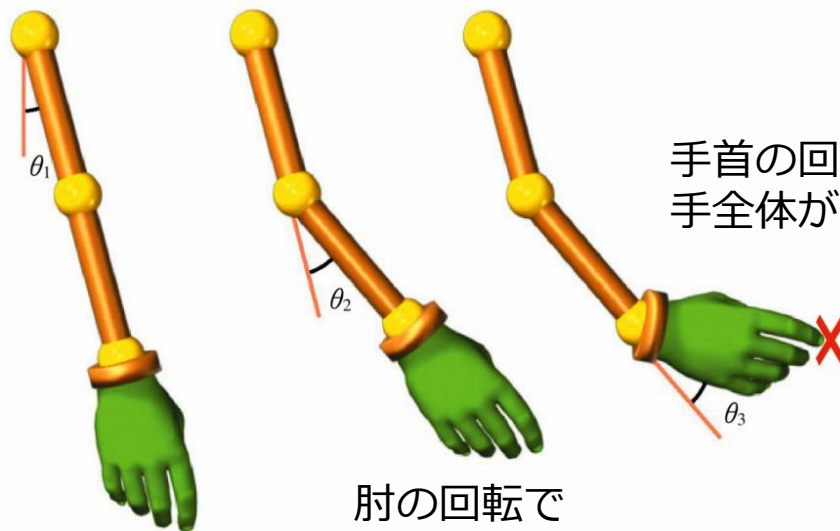
# ポージングの基本

## ■ フォワードキネマティクス (FK)

- 階層構造で多関節を表現



■ 図5.30 — フォワードキネマティクス



手首の回転で  
手全体が動く

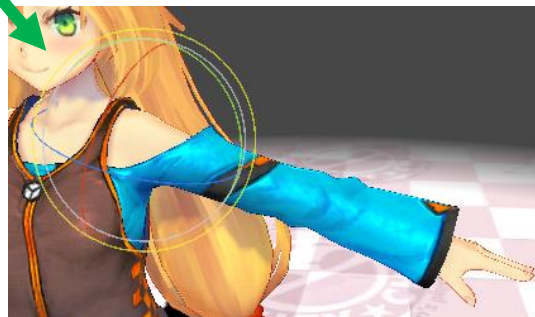
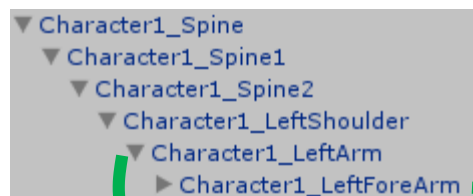
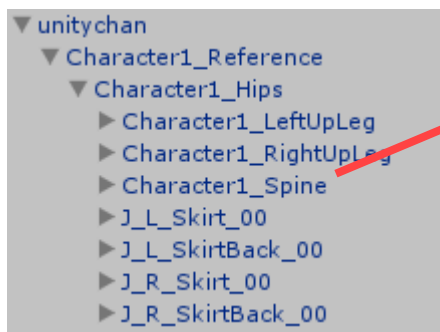
肩の回転で  
上腕から先も動く

肘の回転で  
前腕から先も動く

# キャラクターアニメーション

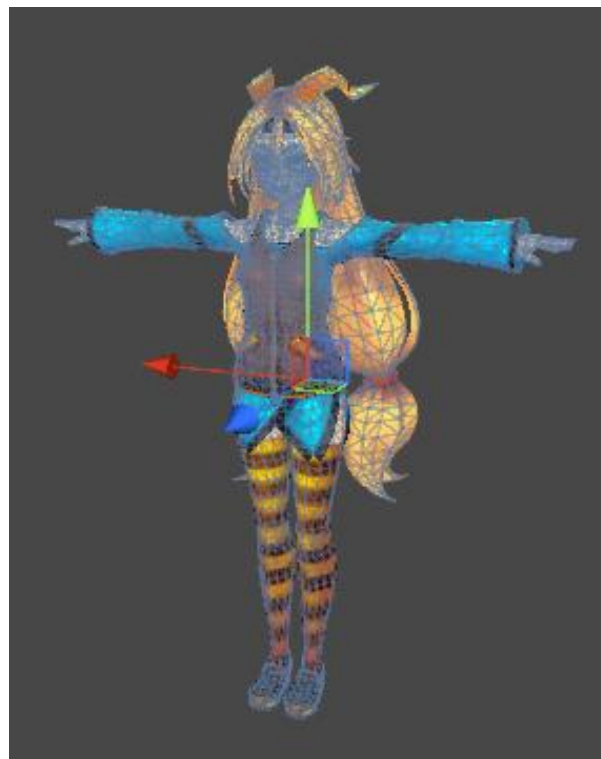
## ■Unityちゃん

- 同様の階層構造で多関節構造を表現
- FKでキャラクターのポーズ付けが可能



# 基本的な操作

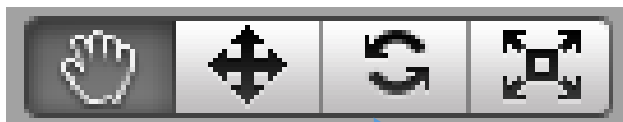
## ■動かしたい物を選択する





# 基本的な操作

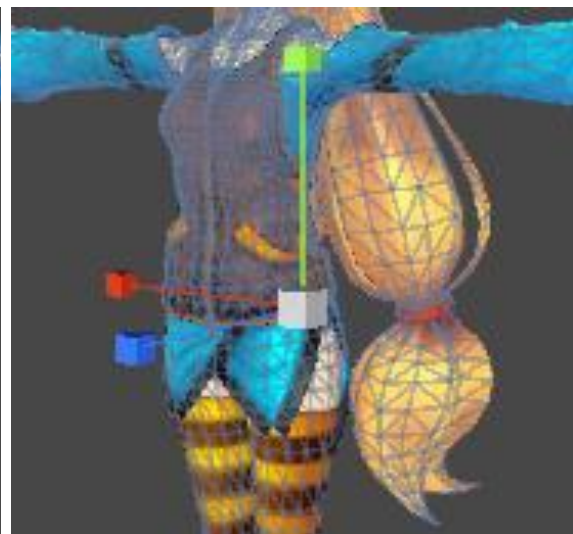
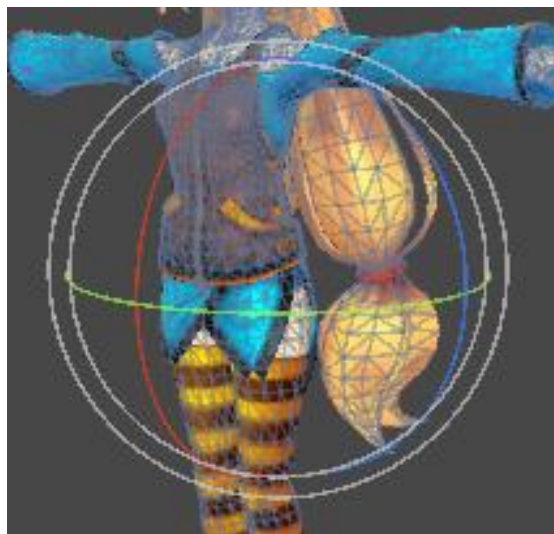
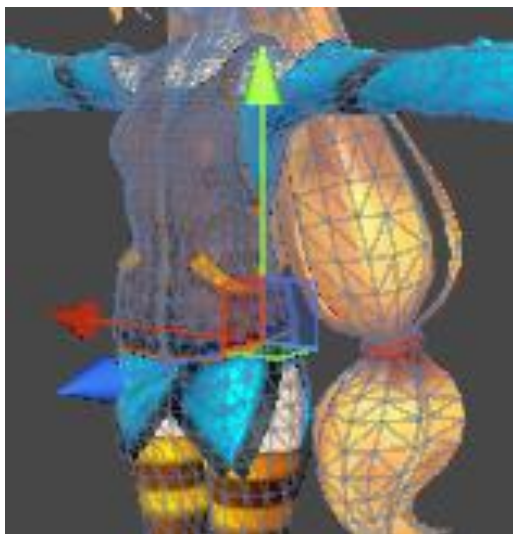
## ■編集モードの切り替え



移動  
(wキー)

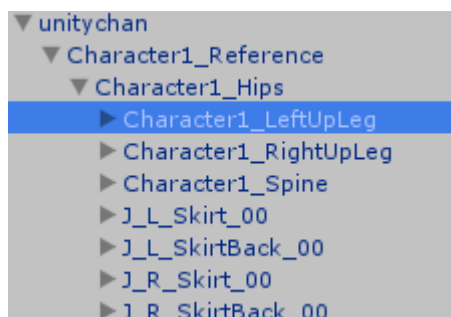
回転  
(eキー)

拡大・縮小  
(rキー)



# より細かいポーズの調整

- 動かしたい場所を選択する  
⇒回転操作でポーズを調整する



ヒエラルキーによる選択

- ・モデル同士の親子関係
- ・各関節を選択可能



# 2次元直交座標系と極座標系

## ■ 2次元直交座標系

- x軸とy軸による表現
- $P = (x_P, y_P)$

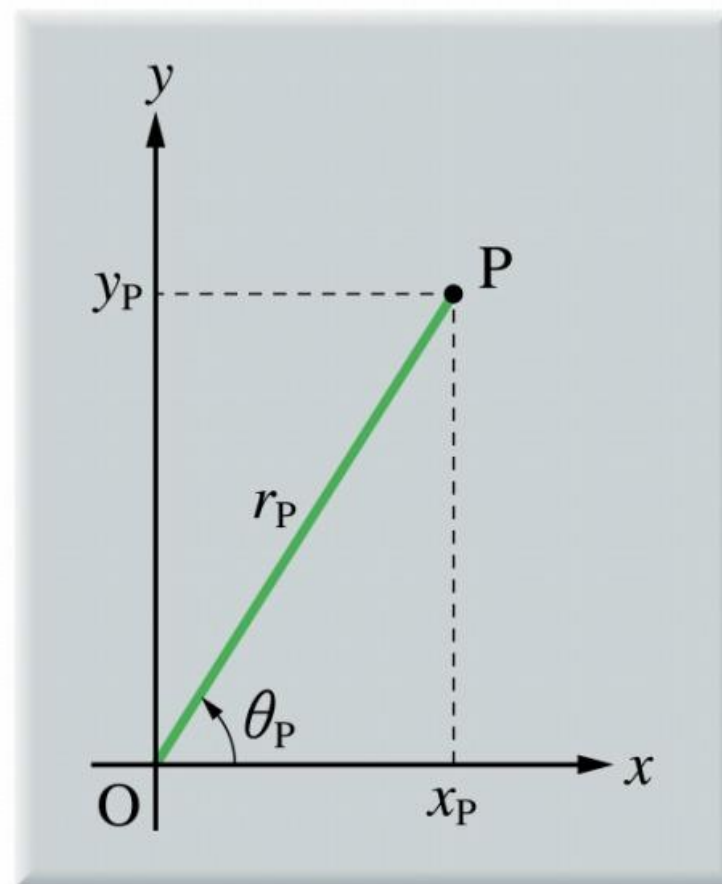
## ■ 極座標系

- 距離と角度による表現
- $P = (r_P, \theta_P)$

## ■ 対応関係

- $x_P = r_P \cos \theta_P$
- $y_P = r_P \sin \theta_P$

■ 図2.1——2次元直交座標系と極座標系



「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会 (CG-ARTS協会)

# 2次元図形の基本変換

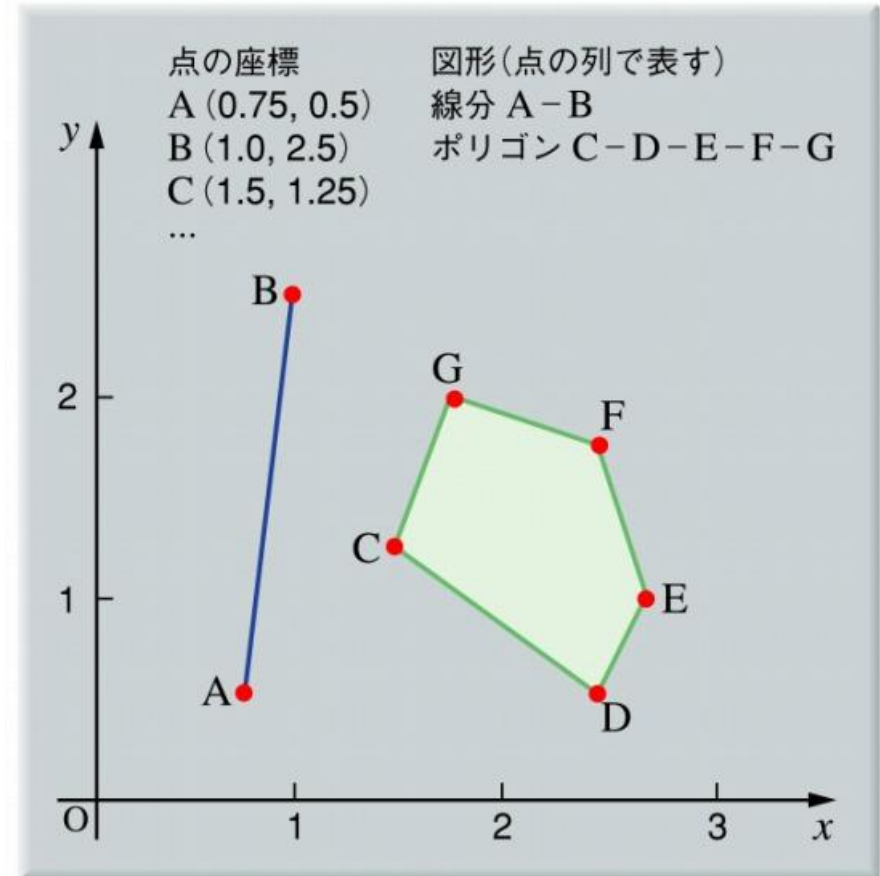
## ■ 図形の表現

- 座標と順序
- 例：線分, ポリゴン

## ■ 幾何学的変換

- 平行移動
- 拡大・縮小
- 回転

■ 図2.2——2次元図形の例



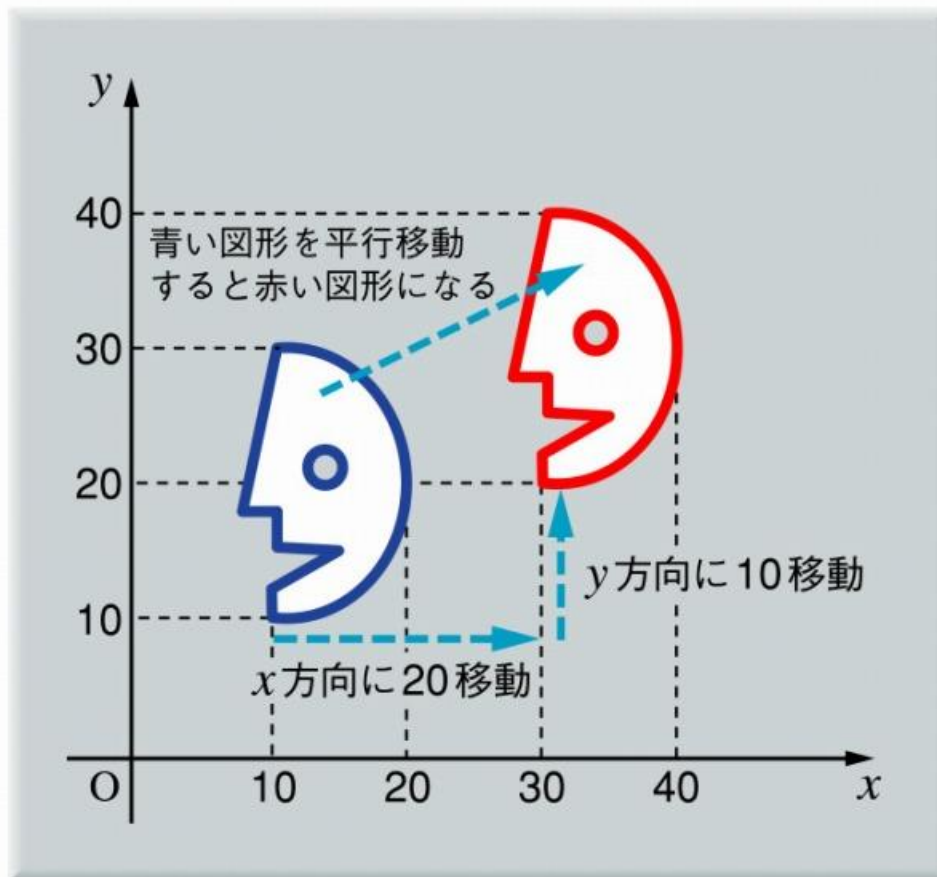
「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会 (CG-ARTS協会)

# 平行移動

## ■ $x, y$ 軸方向に $t_x, t_y$ 移動

- $x' = x + t_x$
- $y' = y + t_y$

■ 図2.3——平行移動の例 ( $t_x = 20, t_y = 10$ )



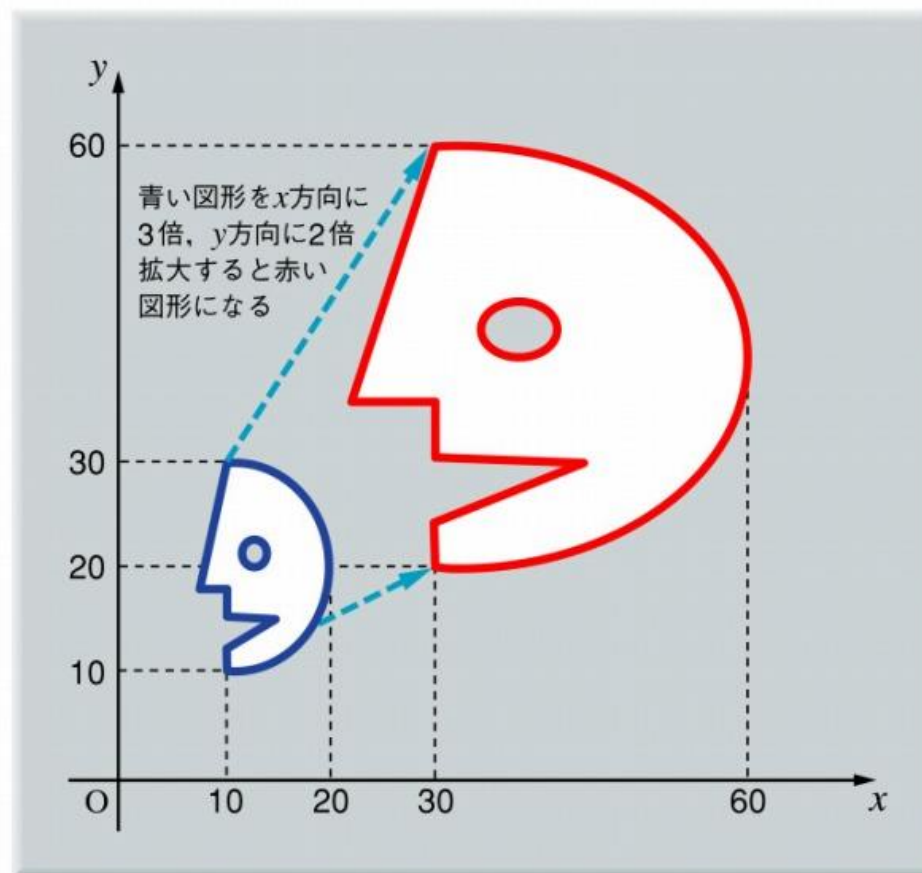
「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会 (CG-ARTS協会)

# 拡大・縮小

## ■ $x, y$ 軸方向に $s_x, s_y$ 倍

- $x' = s_x x$
- $y' = s_y y$

■ 図2.4——拡大・縮小の例 ( $s_x=3, s_y=2$ )



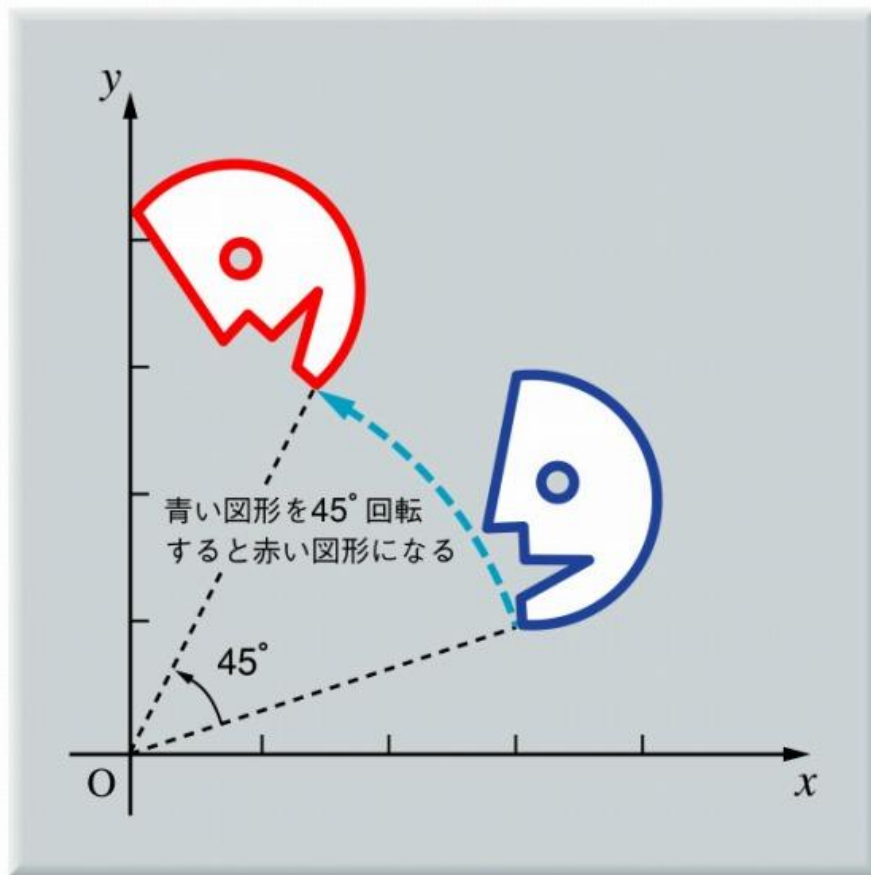
「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会 (CG-ARTS協会)

# 回転

## ■ 原点を中心に角度 $\theta$ 回転

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

■ 図2.5——回転の例 ( $\theta = 45^\circ$ )



「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会 (CG-ARTS協会)

# 同次座標

## ■ベクトルで座標を表した場合

- 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：
$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

平行移動だけベクトルの足し算

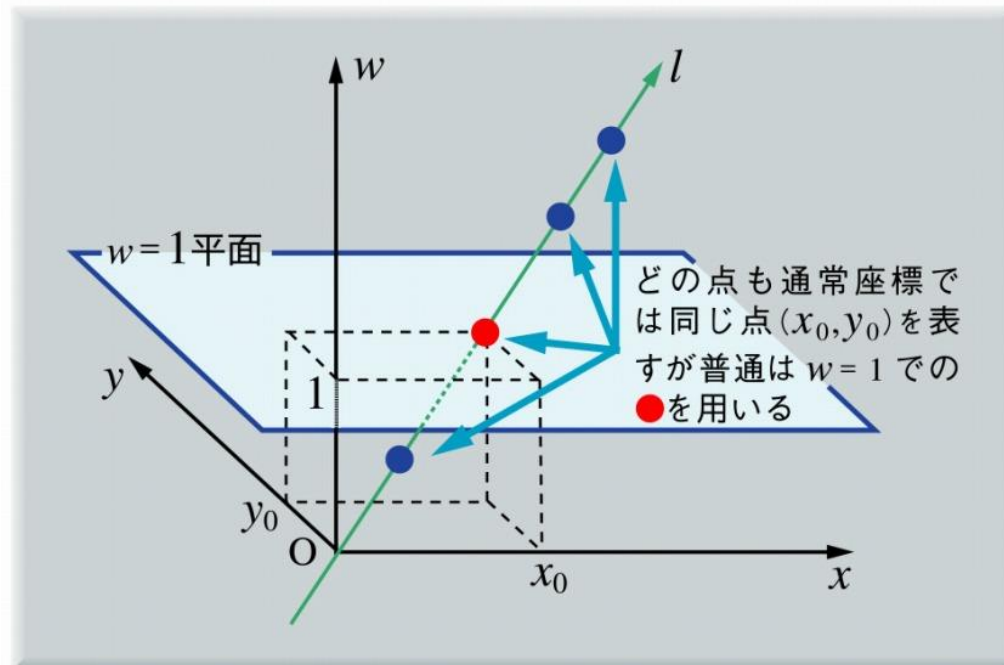


# 同次座標

## ■ 同次座標

- 実数 $w$ で座標を拡張
- $(x, y) \Rightarrow (wx, wy, w)$
- 例：
  - $(2, 3, 1) = (4, 6, 2)$
- 通常は  $(x, y, 1)$  の形

■ 図 2.6——直線  $l$  上のすべての点が通常座標の同じ点  $(x_0, y_0)$  を表す



「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会 (CG-ARTS協会)

# 同次座標

## ■ 同次座標で表した場合

- 拡大・縮小：
$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \underline{\mathbf{S}(s_x, s_y)} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

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

- 平行移動：
$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \underline{\mathbf{T}(t_x, t_y)} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

全て行列の積で表すことができる

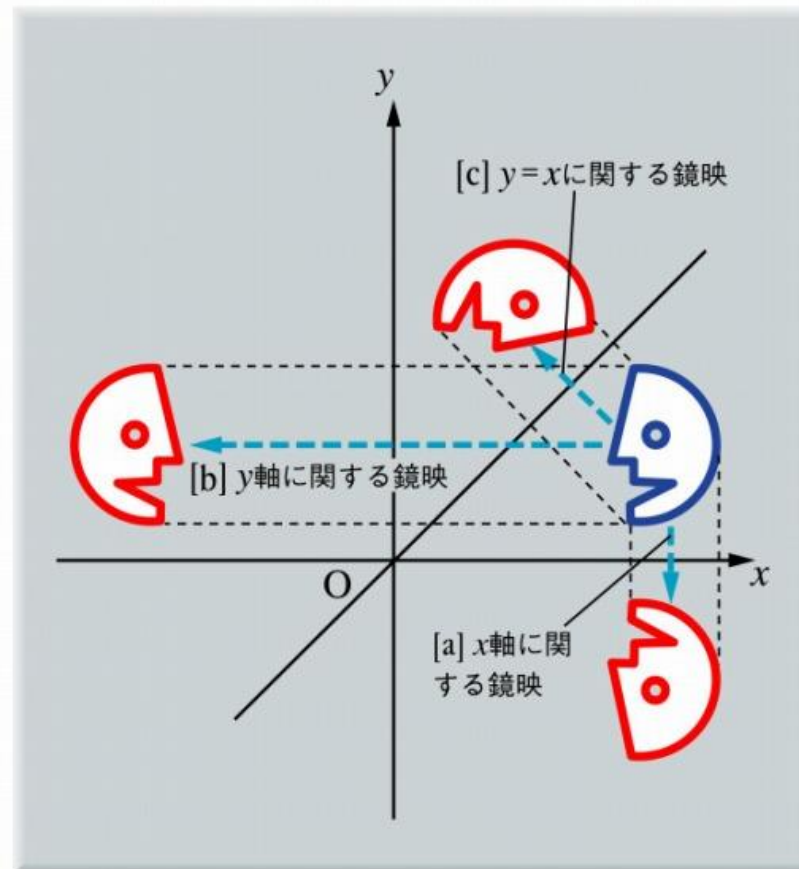
# 鏡映変換

■表2.1—鏡映における変換式

鏡映の種類	(同次座標による)変換式	図
$x$ 軸に関する鏡映	$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$	図2.8 [a]
$y$ 軸に関する鏡映	$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$	図2.8 [b]
直線 $y=x$ に関する鏡映	$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$	図2.8 [c]

「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会 (CG-ARTS協会)

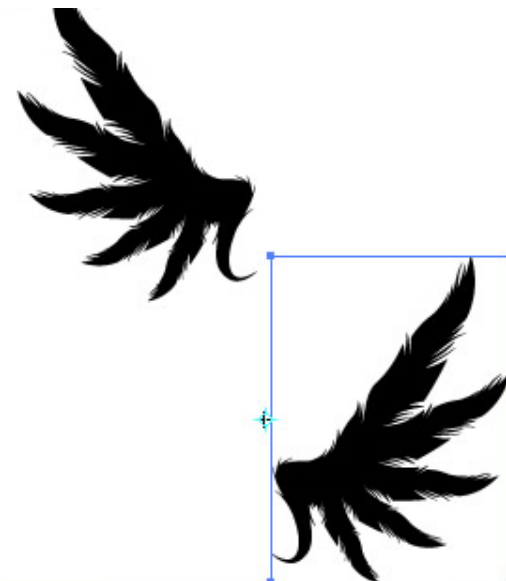
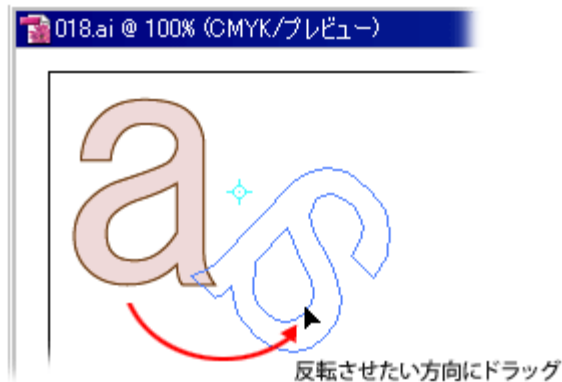
■図2.8—鏡映の例



「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会 (CG-ARTS協会)

# CGにおける鏡映変換の用途

- Adobe Illustratorのリフレクトツール、Power Pointの左右反転
  - 左右対称な物をコピーするのに使用

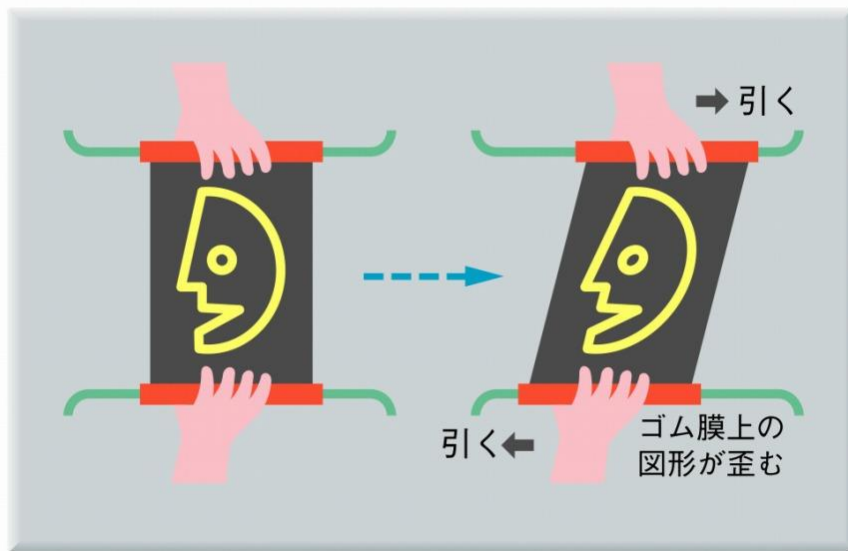


©Cafi Net

# スキュー（せん断）

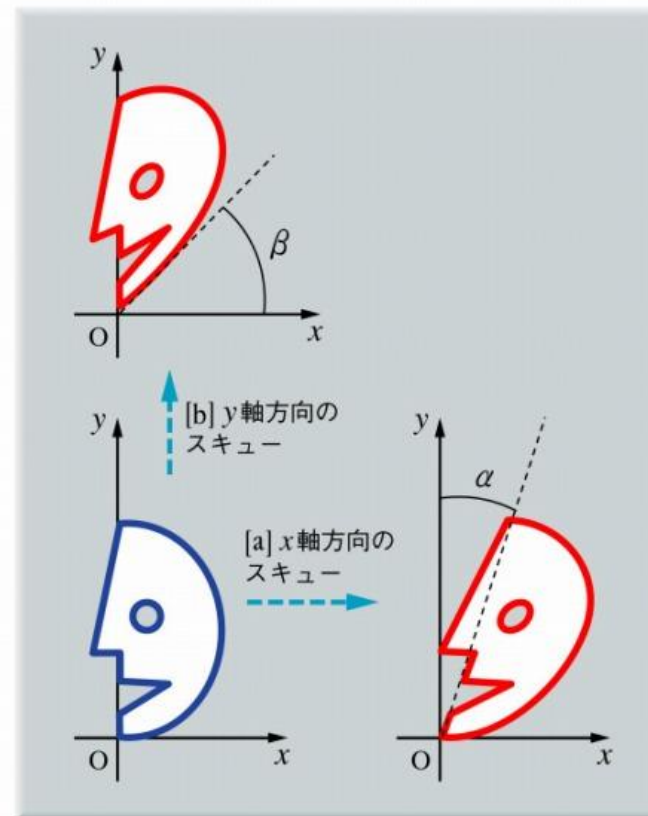
## ■長方形⇒平行四辺形の歪み

■図2.9—ゴム膜でのスキューの原理



「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会（CG-ARTS協会）

■図2.10—スキューの例



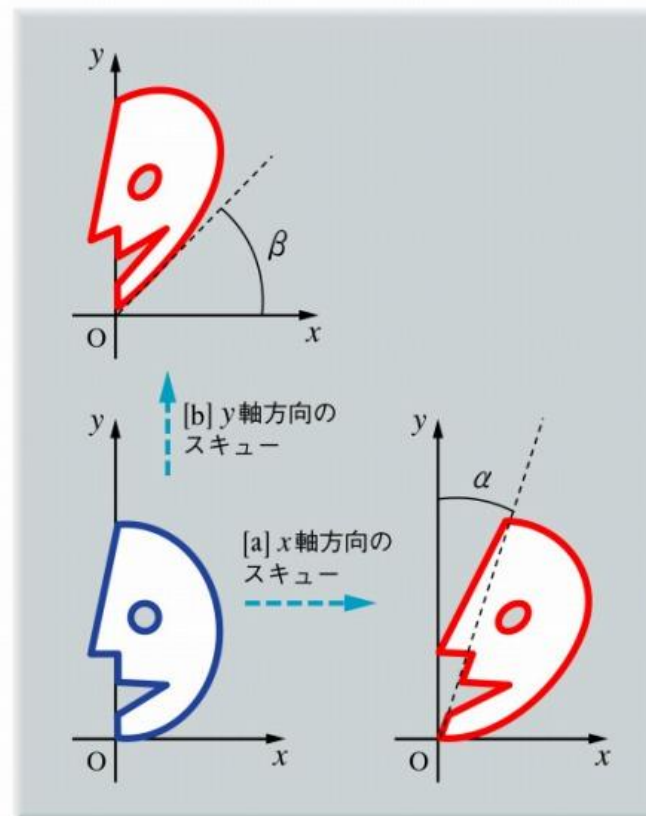
「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会（CG-ARTS協会）

# スキュー（せん断）

## ■長方形⇒平行四辺形の歪み

スキューの種類	(同次座標による)変換式
x軸方向のスキュー	$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & a & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$
y軸方向のスキュー	$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ b & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$

■図2.10——スキューの例

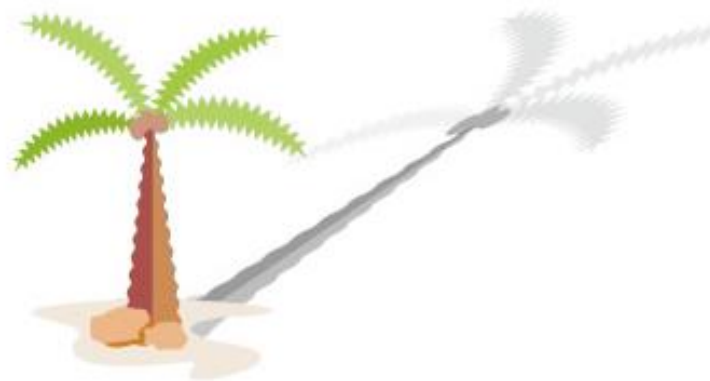
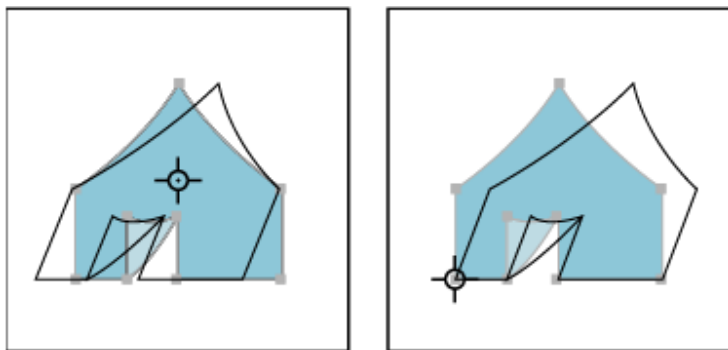


「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会 (CG-ARTS協会)

# CGにおけるスキューの用途

## ■ Adobe Illustratorのシアーツール

- オブジェクトを傾けたり, 歪めたりする
- 影をつけるのに便利



©Cafi Net

# 合成変換

- 点 :  $\mathbf{p} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$
- 変換の列 :  $A_1, A_2, A_3$
- 合成変換 :  $\mathbf{p}' = (A_3(A_2(A_1\mathbf{p})))$
- $A = A_3A_2A_1$  とすると  $\mathbf{p}' = A\mathbf{p}$

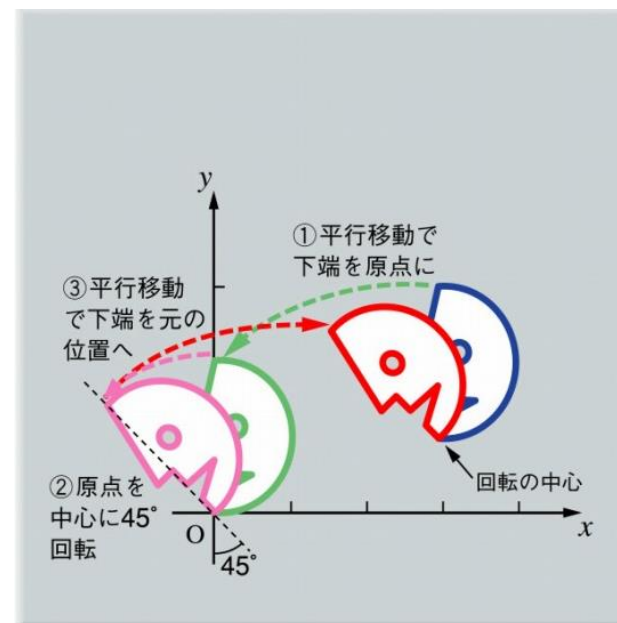


# 合成変換

## ■ 具体例

- 平行移動 $T(-x_0, -y_0) \Rightarrow$ 回転 $R(\theta) \Rightarrow$ 平行移動 $T(x_0, y_0)$

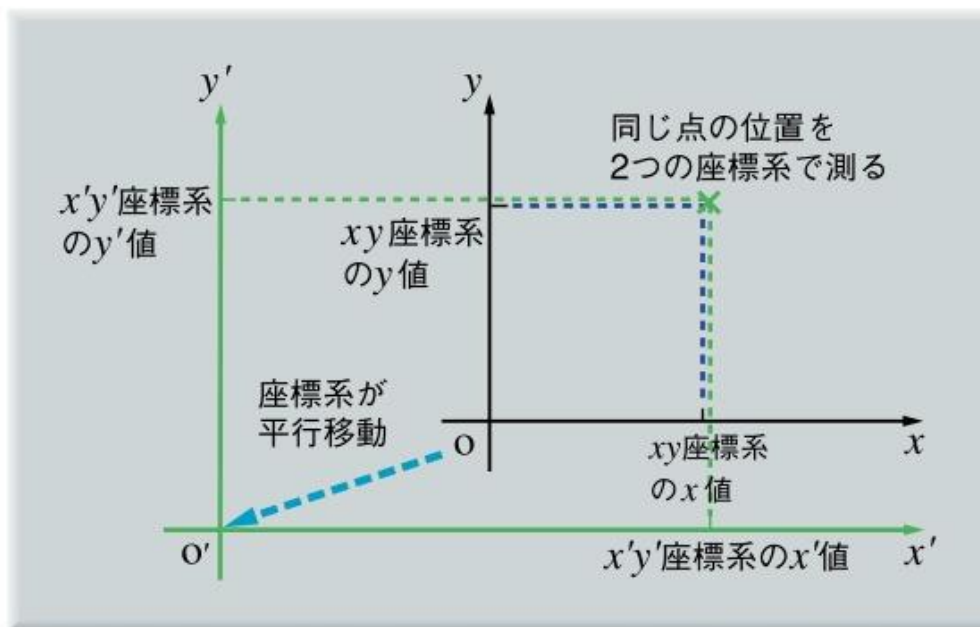
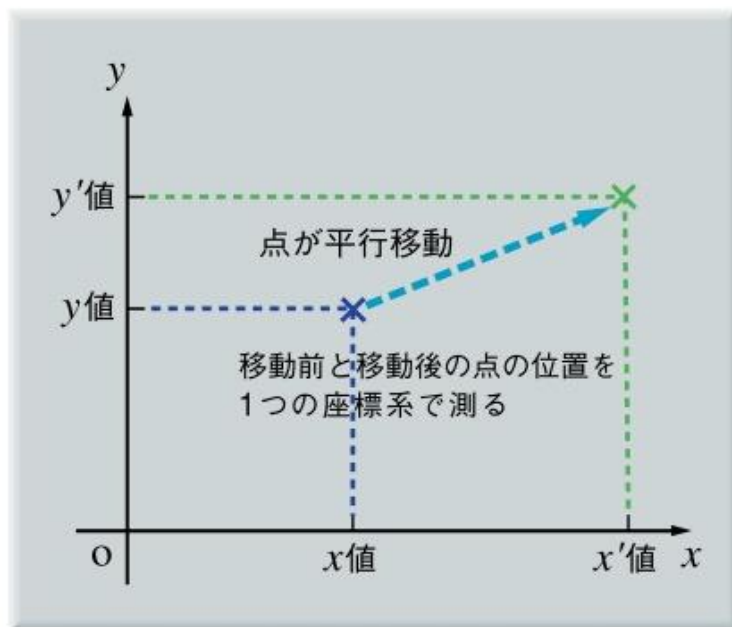
- $$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = T(x_0, y_0)R(\theta)T(-x_0, -y_0) \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$



# 2次元アフィン変換

## ■ 平行移動 $T(x, y)$ の解釈

■ 図2.13——平行移動の2つの解釈



[a] 1つの座標系における移動前の点の位置と移動後の位置との関係

[b] 1つの点に対する $xy$ 座標系での位置と $x'y'$ 座標系での位置の関係

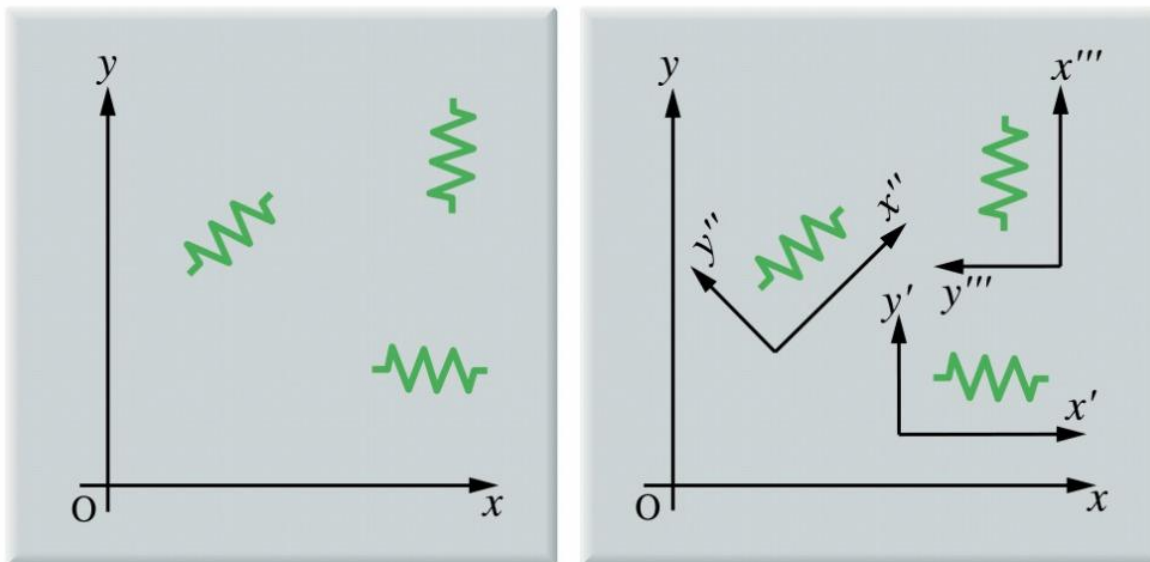
「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会 (CG-ARTS協会)

# 2次元アフィン変換

## ■座標系を動かす解釈

- 同じ図形をたくさん描く場合に有利

■図2.14——同じ図形を異なる位置や向きに描く場合



[a] 別々に座標値を指定しなければならない場合  
[b] 別々の座標系を用いて、同じ座標値で表せる場合

「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会 (CG-ARTS協会)

# 2次元アフィン変換

## ■ 幾何学変換の一般的な行列表現

$$\bullet \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

- : 拡大・縮小, 回転, スキュー
- : 平行移動

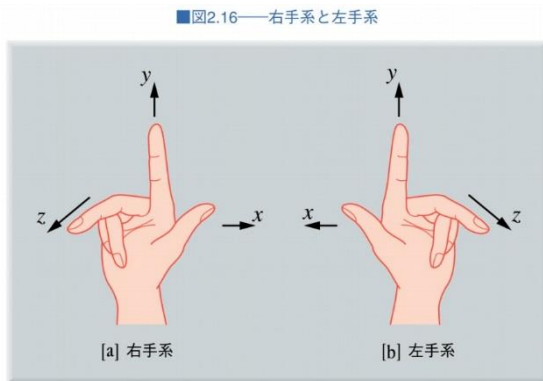
## ■ 種類

- 剛体変換: 形が変わらないアフィン変換
  - 平行移動, 回転

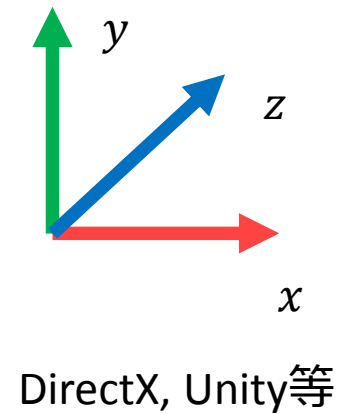
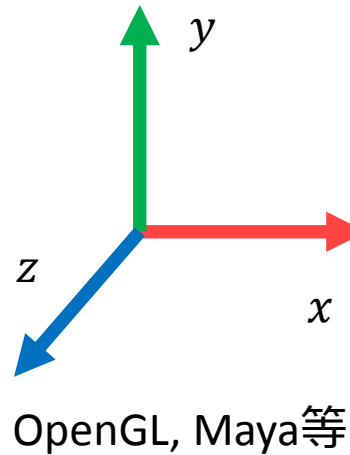
# 3次元座標系

## ■直交座標系

- 右手系
  - $z$ が手前を向いている
  - 映像制作系で一般的
- 左手系
  - $z$ が奥を向いている
  - ゲーム制作系で一般的



「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会 (CG-ARTS協会)



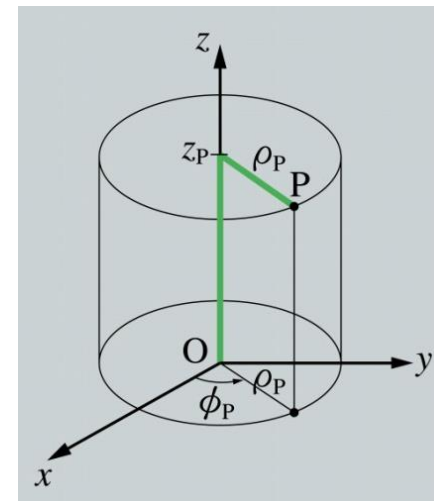
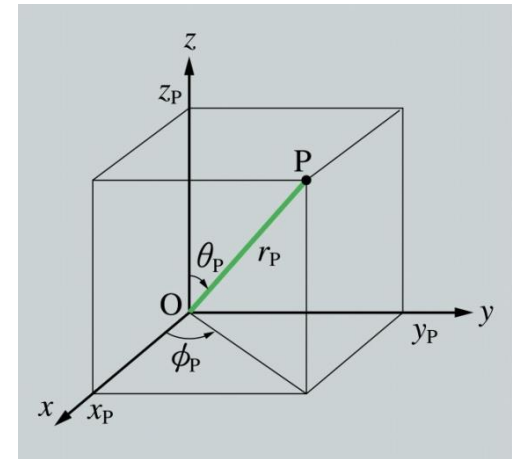
# 3次元座標系

## ■ 極座標

- $x_P = r_P \sin \theta_P \cos \phi_P$
- $y_P = r_P \sin \theta_P \sin \phi_P$
- $z_P = r_P \cos \theta_P$

## ■ 円柱座標

- $x_P = \rho_P \cos \phi_P$
- $y_P = \rho_P \sin \phi_P$
- $z_P = z_P$



# 簡単なモデリング

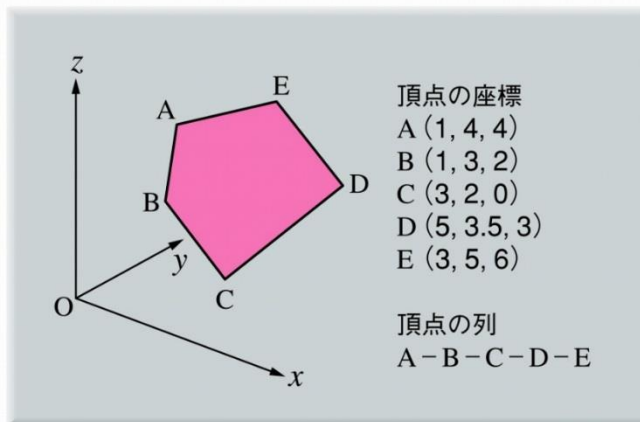
## ■モデリング

- 3次元図形の形状をデザイン
- 形状モデル：形状データ

## ■ポリゴン

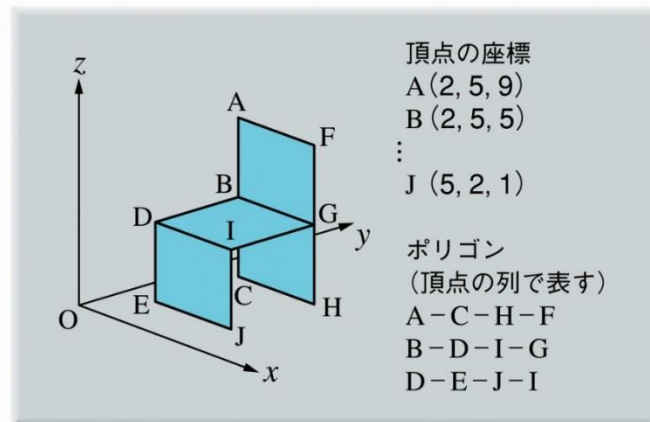
- 頂点座標
- 順序

■図2.18——頂点5個のポリゴンの例



[コンピュータグラフィックス] 2004年 / 財団法人画像情報教育振興協会 (CG-ARTS協会)

■図2.19——ポリゴンによる形状モデル



[コンピュータグラフィックス] 2004年 / 財団法人画像情報教育振興協会 (CG-ARTS協会)

# 3次元の同次座標

## ■ 2次元

- $(x, y) \Rightarrow (wx, wy, w) = (x, y, 1)$

## ■ 3次元

- $z$ の成分が増えるだけ
- $(x, y, z) \Rightarrow (wx, wy, wz, w) = (x, y, z, 1)$
- 2次元の場合と同様に平行移動等の変換を定義できる



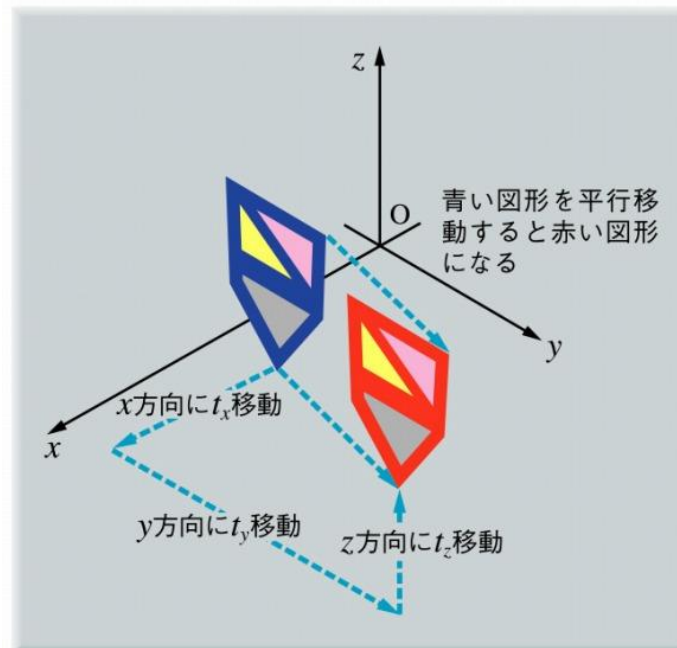
# 平行移動

■  $x, y, z$ 軸方向に  $t_x, t_y, t_z$  移動

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

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

■ 図2.20——3次元での平行移動



「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会 (CG-ARTS協会)

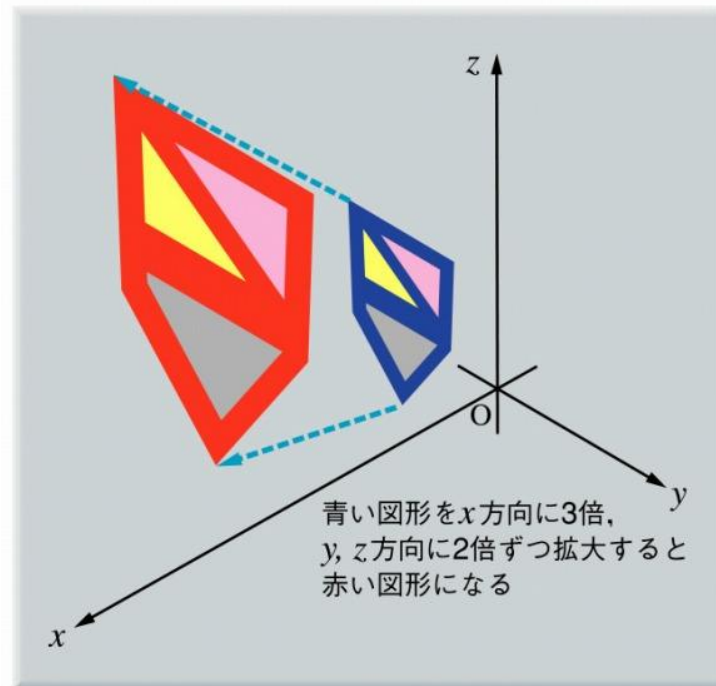
# 拡大・縮小

■  $x, y, z$ 軸方向に  $s_x, s_y, s_z$  倍

$$\bullet \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

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

■ 図2.21——3次元での拡大・縮小 ( $s_x=3, s_y=s_z=2$ )



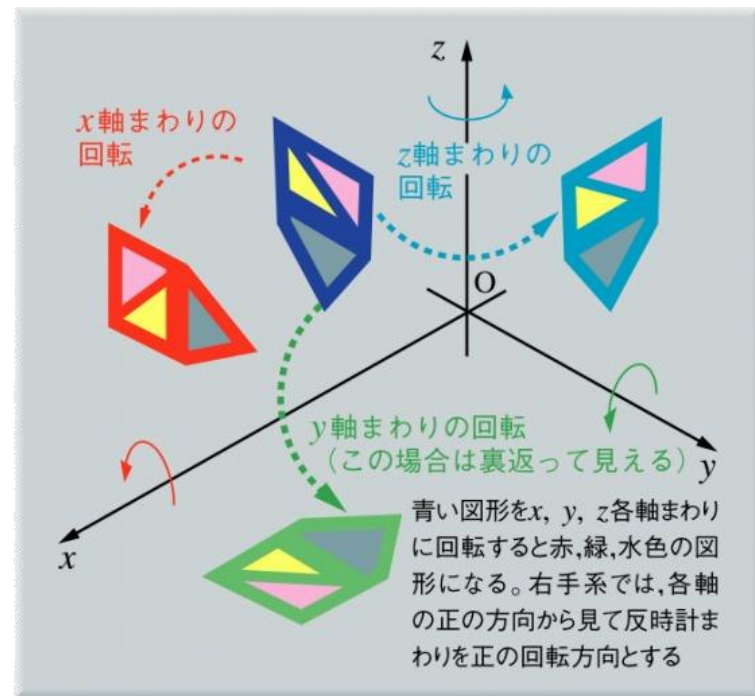
「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会 (CG-ARTS協会)

# 回転

## ■ 各軸周りの回転

- z 軸はまわりは 2次元の場合と同じ

■ 図2.22——3次元での各軸まわりの回転  
( $\theta=90^\circ$ , 赤: x軸まわり, 緑: y軸まわり, 水色: z軸まわり)



【コンピュータグラフィックス】2004年 / 財団法人画像情報教育振興協会 (CG-ARTS協会)

# 回転

## ■ x 軸まわりの回転

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

## ■ y 軸まわりの回転

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

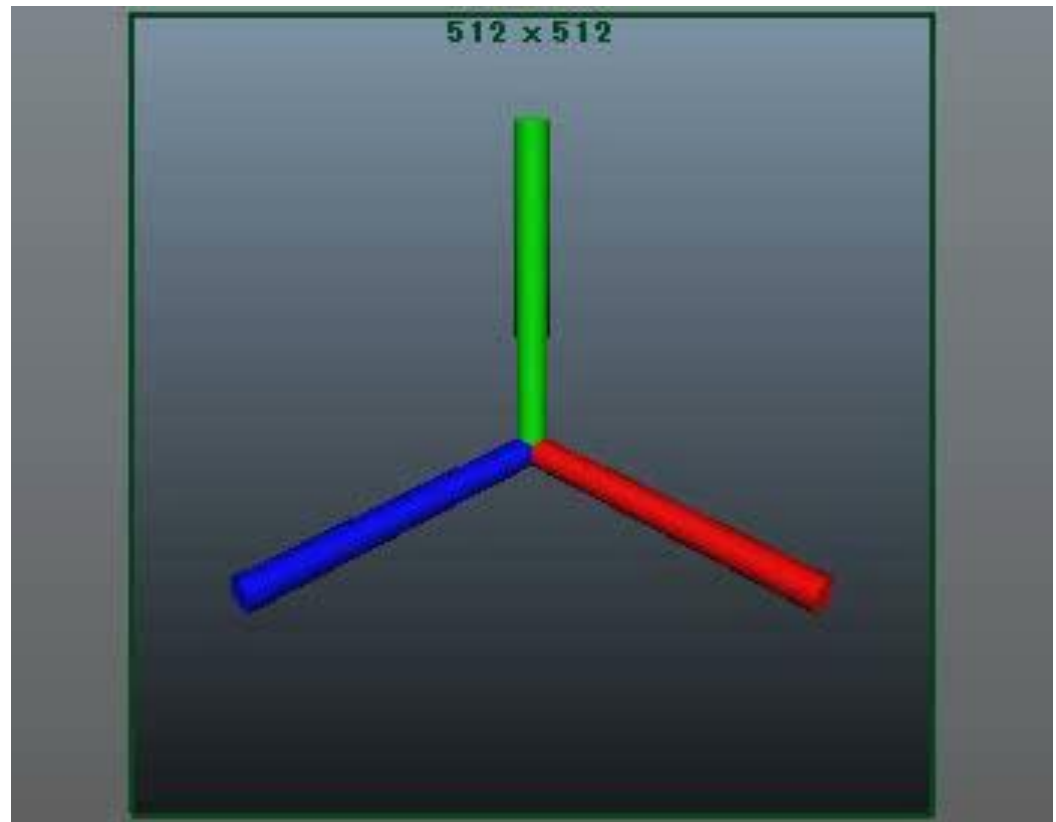
## ■ z 軸まわりの回転

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

# 回転

## ■オイラー角

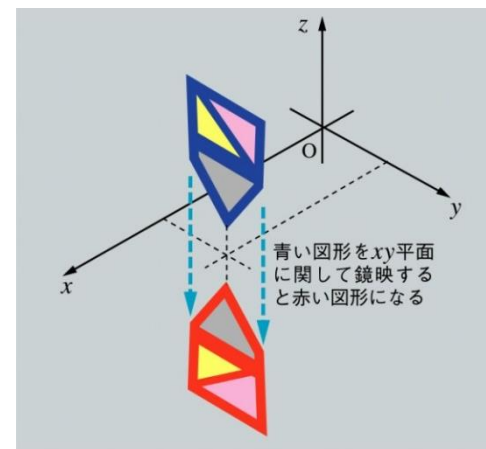
- CGソフトウェアで一般的な手法
- 1.  $x$ 軸周りに $\alpha$
- 2.  $y$ 軸周りに $\beta$
- 3.  $z$ 軸周りに $\gamma$
- 順番に回転する



# 鏡映とスキュー (せん断)

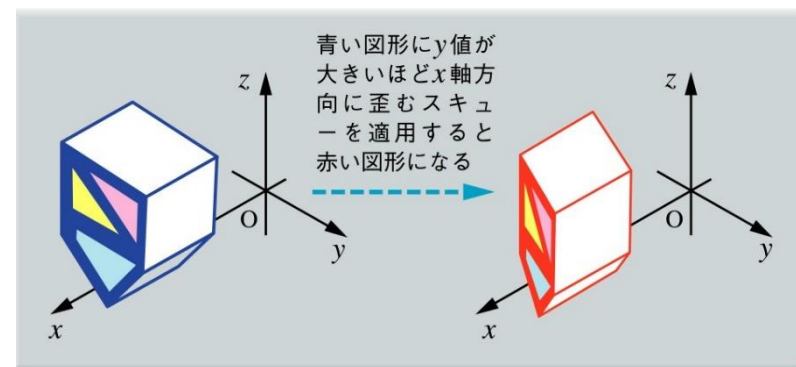
## ■ $xy$ 平面に関する鏡映

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



## ■ $y$ 値が大きいほど $x$ 軸方向に歪むスキュー

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



# 3次元アフィン変換

## ■ 幾何学変換の一般的な行列表現

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

- □: 拡大・縮小, 回転, スキュー
- □: 平行移動

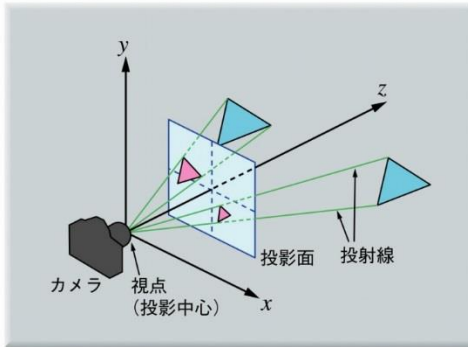
## ■ 種類

- 剛体変換: 形が変わらないアフィン変換
  - 平行移動, 回転

# 次回

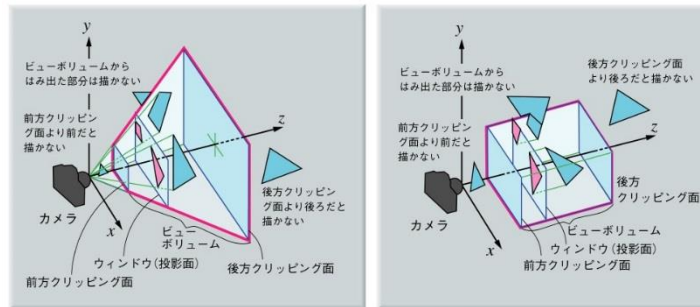
## ■ CG のための数学的基礎 2 ～投影変換～

■図2.27—透視投影の原理



「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会 (CG-ARTS協会)

■図2.31—ビューボリュームとクリッピング

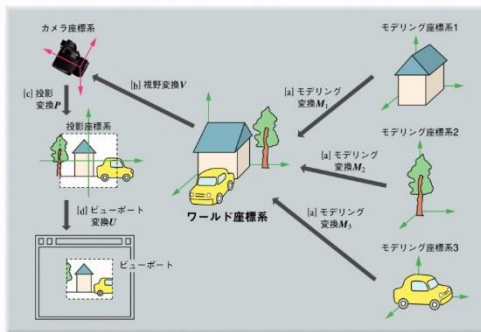


[a] 透視投影の場合

[b] 平行投影の場合

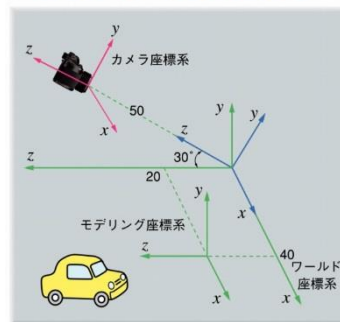
「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会 (CG-ARTS協会)

■図2.43—モデルから表示までの変換(ビューイングパイプライン)



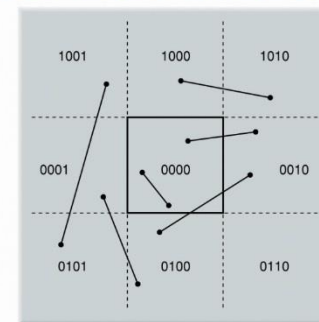
「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会 (CG-ARTS協会)

■図2.45—モデリング変換と視野変換の例



「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会 (CG-ARTS協会)

■図2.46—2次元クリッピングのための4ビットコード



「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会 (CG-ARTS協会)