

# 画像工学

Rev. 2018.08.01

# 講義ホームページ

---

[cvwww.ee.ous.ac.jp/lect/ip/](http://cvwww.ee.ous.ac.jp/lect/ip/)



# 講義内容

---

## ✧ デジタル画像処理

画像変換、特徴量抽出、パターン認識、  
画像符号化、動画画像処理

## ✧ 画像処理プログラミング

Processingを使ったプログラム作成

→ レポート課題

## ✧ コンピュータグラフィックスの基礎

# 知覚と画像処理

---

人は見たものをどのようにして、認識・理解しているのか？

光 → 眼球 → 視神経 → 脳(情報処理)

コンピュータはどのような処理を行えば、  
「もの」を画像から認識・理解できるのか？

# 視細胞

---

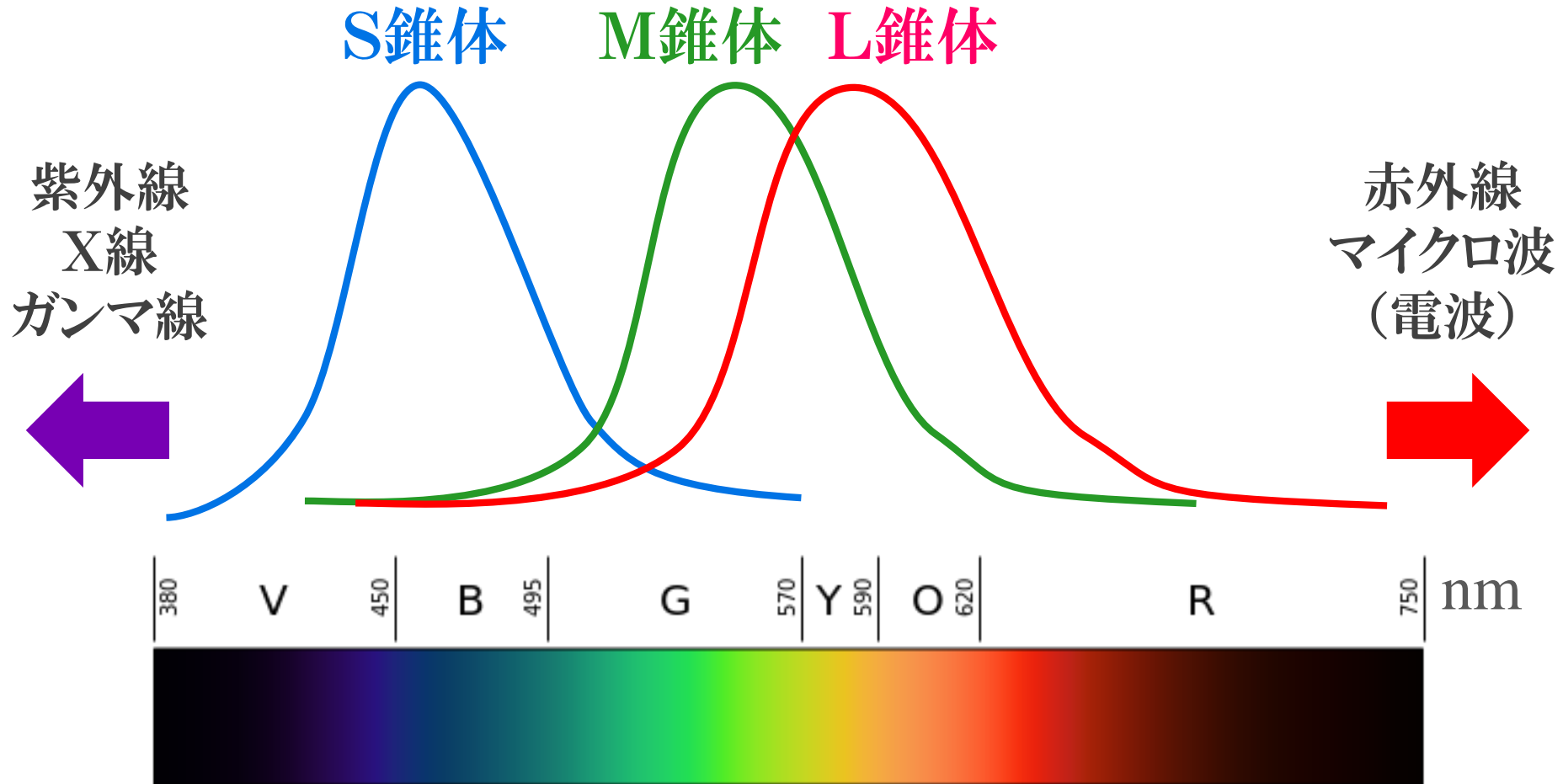
## 錐体細胞

明るい場所で反応する。3種類あり、色を区別できる。

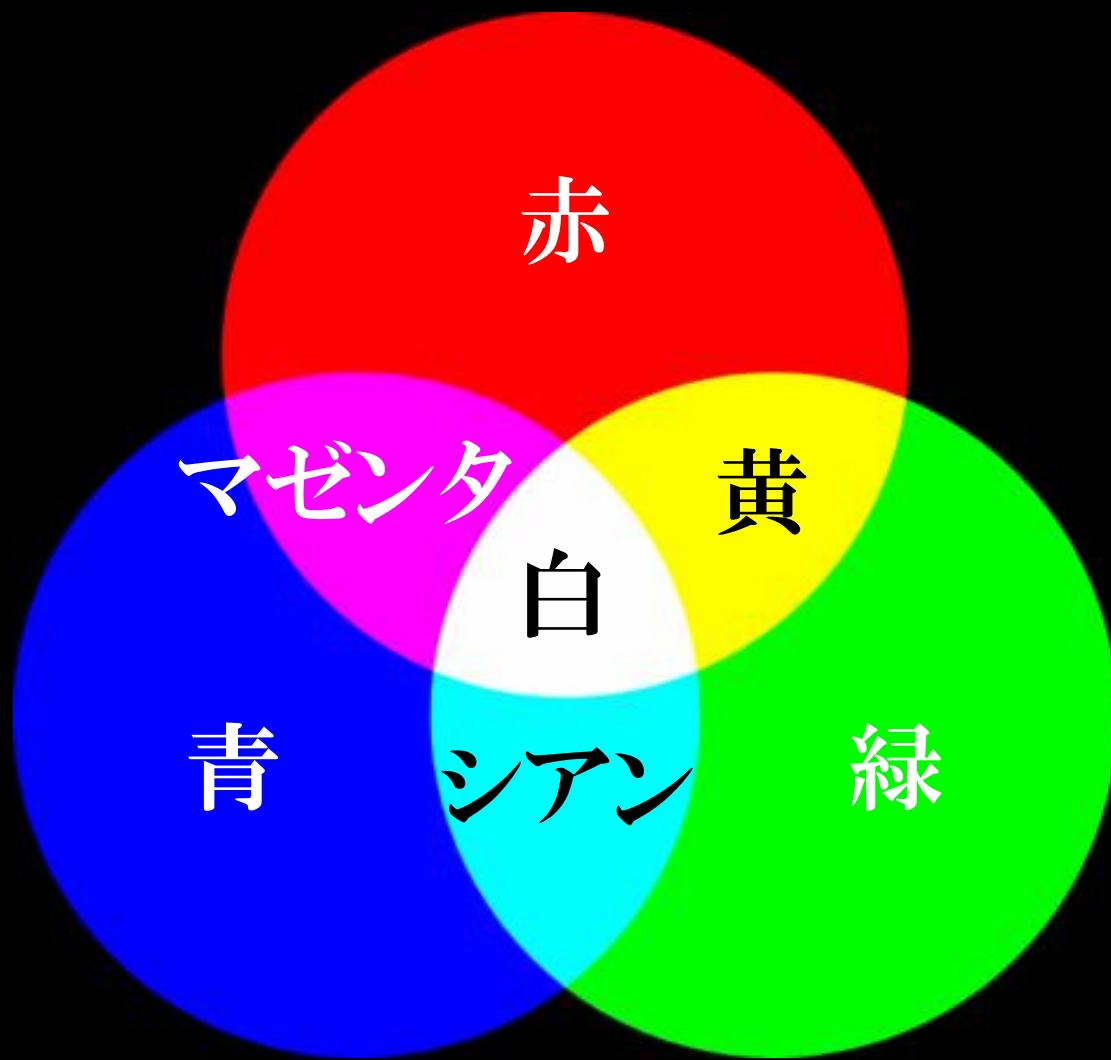
## 桿体細胞

暗い場所で反応する。色を区別できない。

# 可視光波長と錐体細胞の反応



# 光の3原色



# 人間の知覚

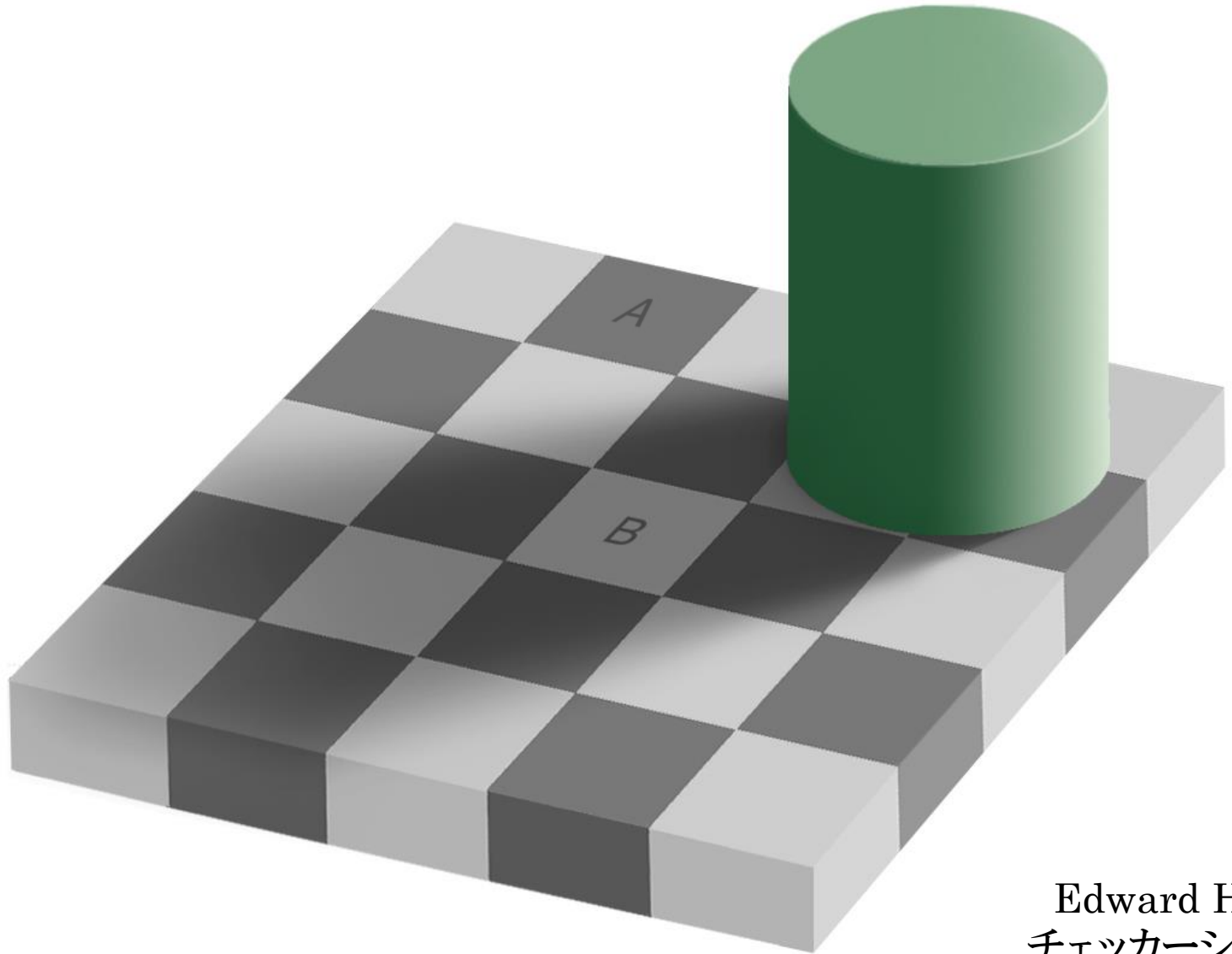
---

人間は、正確にものが見えているのか？



# AとBはどちらが明るい？

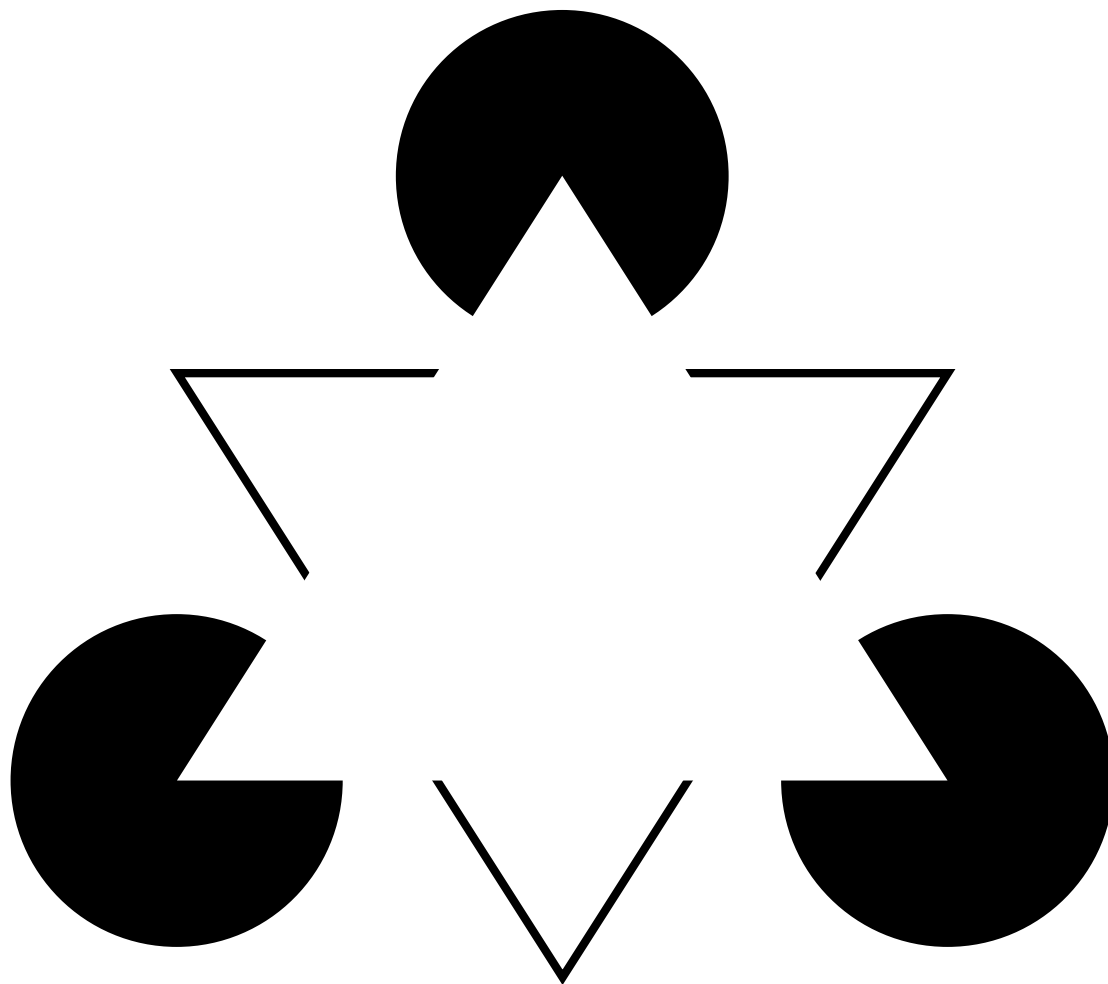
---



Edward H. Adelson  
チェッカーシャドウ錯視

# 主觀的輪郭

---



# 顔細胞（顔ニューロン）

---

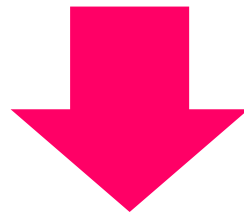
顔の認識に特化した神経細胞

# コンピュータによる知覚

---

コンピュータは、点の色や座標を正確に検出できる。(ミクロの視点)

しかし、顔認識などのように、画像を全体的に捉えることは簡単ではない。  
(マクロの視点)



研究の対象となる

# 画像処理技術の応用

---

- ✧ 見やすい画像への変換・補正
- ✧ 文字認識、指紋照合、顔識別、年齢推定
- ✧ 製品検査、交通量計測
- ✧ ジェスチャー認識、スポーツ映像解析
- ✧ 高画質での映像記録・映像伝送

# 画像処理の種類

---

## ❖ 画像処理 (image processing)

画像を加工して新たな画像を作る。

## ❖ 画像認識 (image recognition)

画像中の対象物が何であるか同定する。

## ❖ 画像計測 (image measurement)

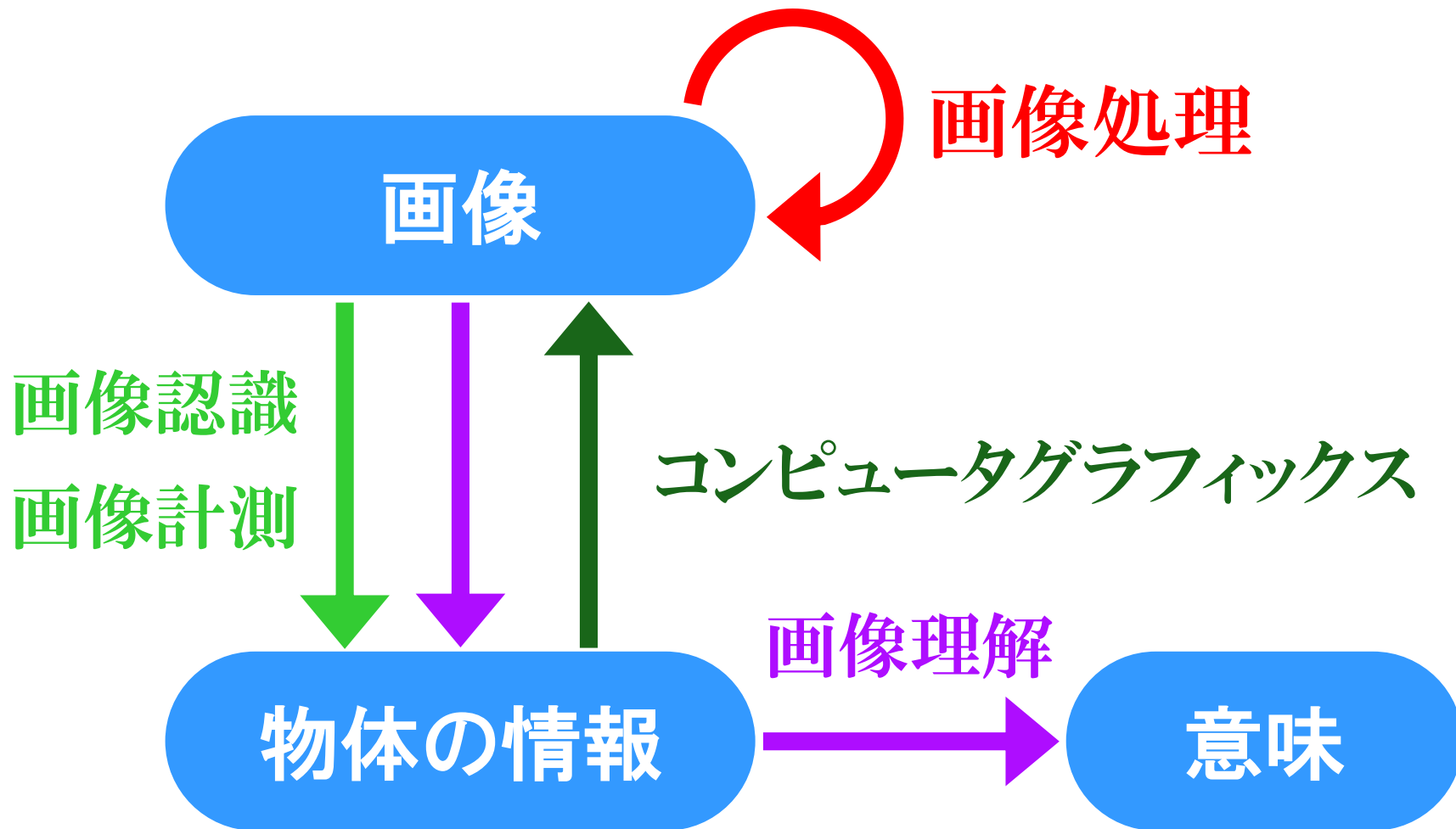
画像から対象物の形状や個数、動きの情報を得る。

## ❖ 画像理解 (image understanding)

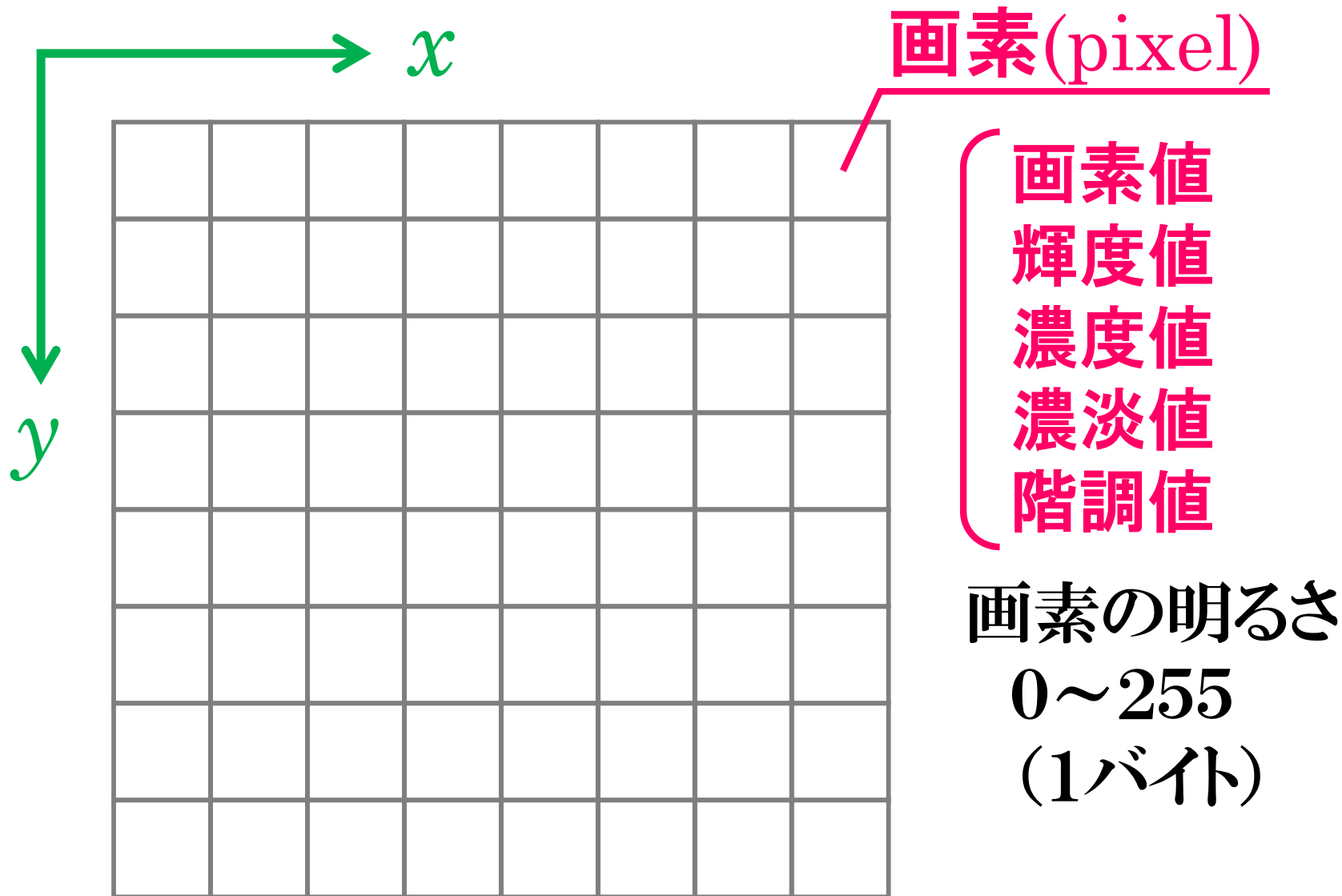
画像から得られる情報を統合し、画像がもつ意味を構造的に理解する。

# 画像処理とCG

---



# デジタル画像





# デジタル画像の取得

---

## ✦ AD変換

重要

### ① 標本化 (sampling)

空間のデジタル化

標本化間隔 (sampling interval)

解像度 (resolution)

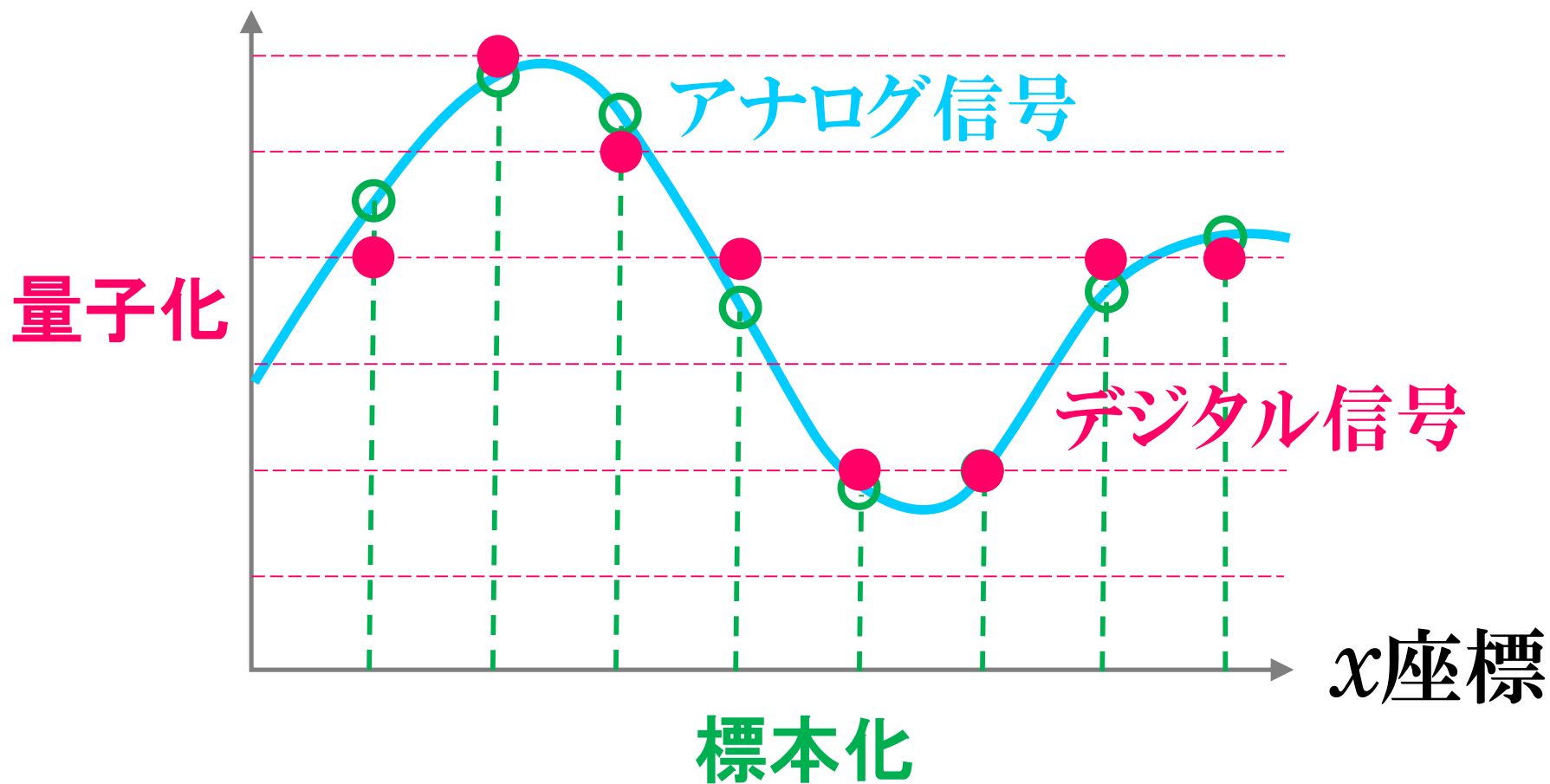
### ② 量子化 (quantization)

色のデジタル化

階調 (gradation)

# AD変換

画素値 (明るさ)



# 階調数による画像の種類

---

✧ 2値画像 (binary image)

2階調(黒・白)

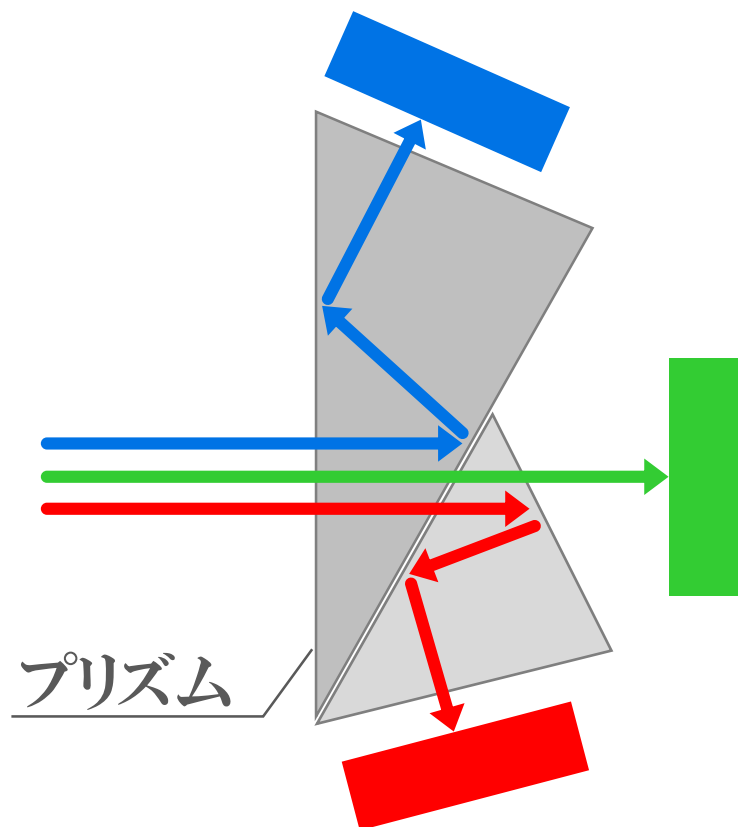
✧ 白黒濃淡画像 (grayscale image)

256階調

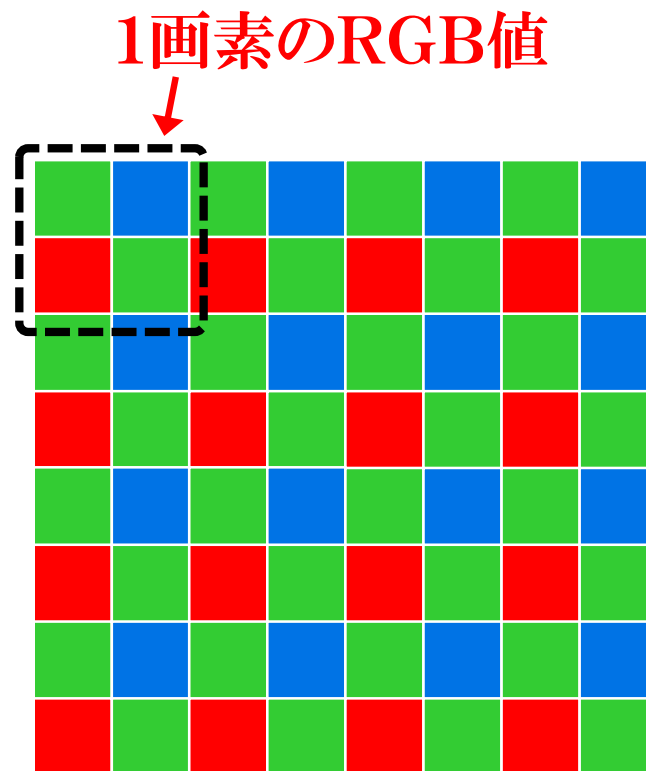
✧ カラー画像 (color image)

RGB(赤・緑・青) 各256階調

# RGBの取得



**三板式(3CCD方式)**  
プリズムでRGBを分光  
して取得する



**単板式(ベイヤー方式)**  
単板に並んだRGBのセル  
で取得する

# 画像の性質

---



明るい



鮮やか(コントラストが高い)



暗い

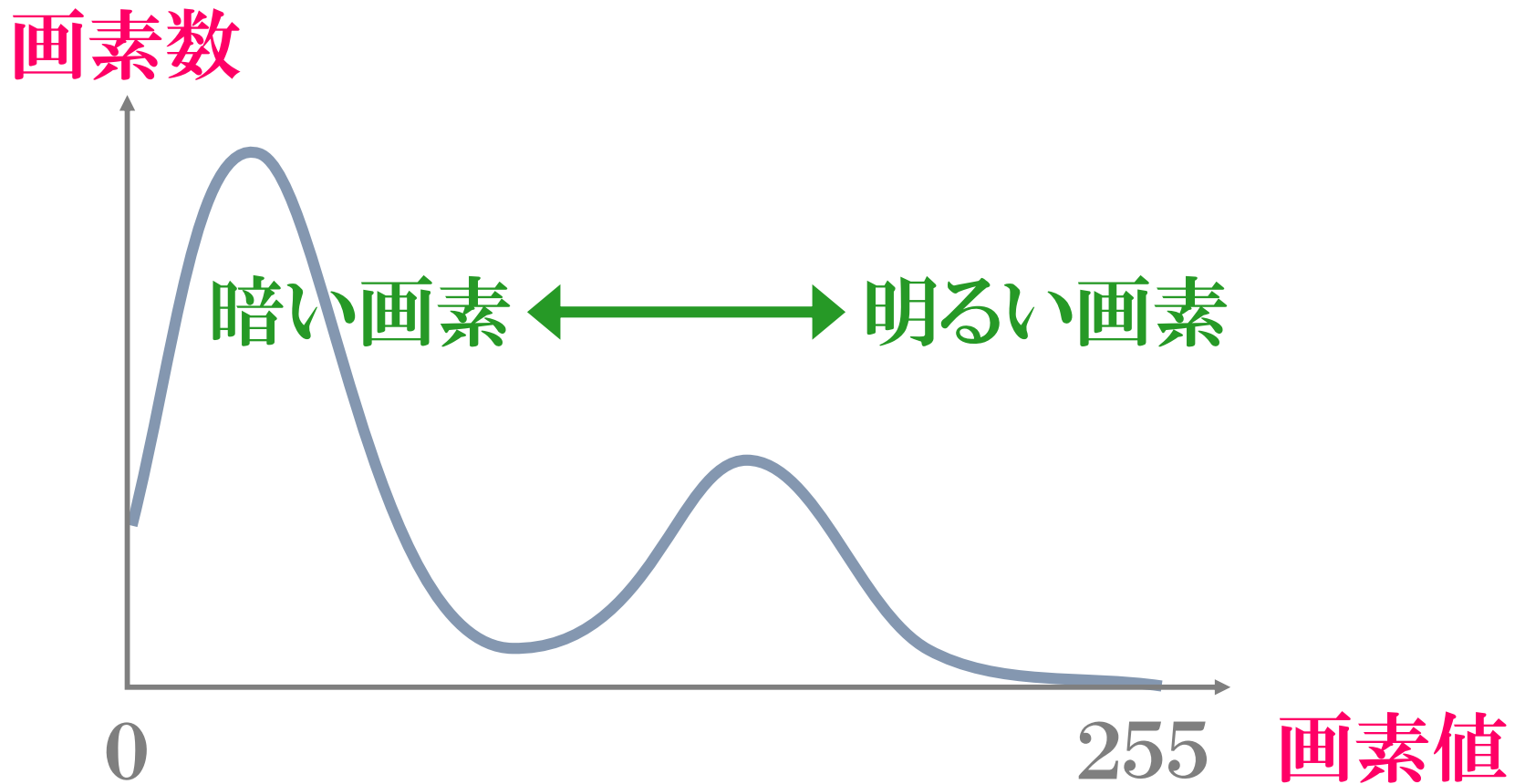


淡い(コントラストが低い)

# ヒストグラム (histogram)

画素値の頻度分布を表したグラフ

重要



# 画素値の解析

---

## ✧ 画素値の平均

小さい ➡ 暗い画像

大きい ➡ 明るい画像

## ✧ 画素値の分散

小さい ➡ コントラストが低い画像

大きい ➡ コントラストが高い画像

# 画像処理

---

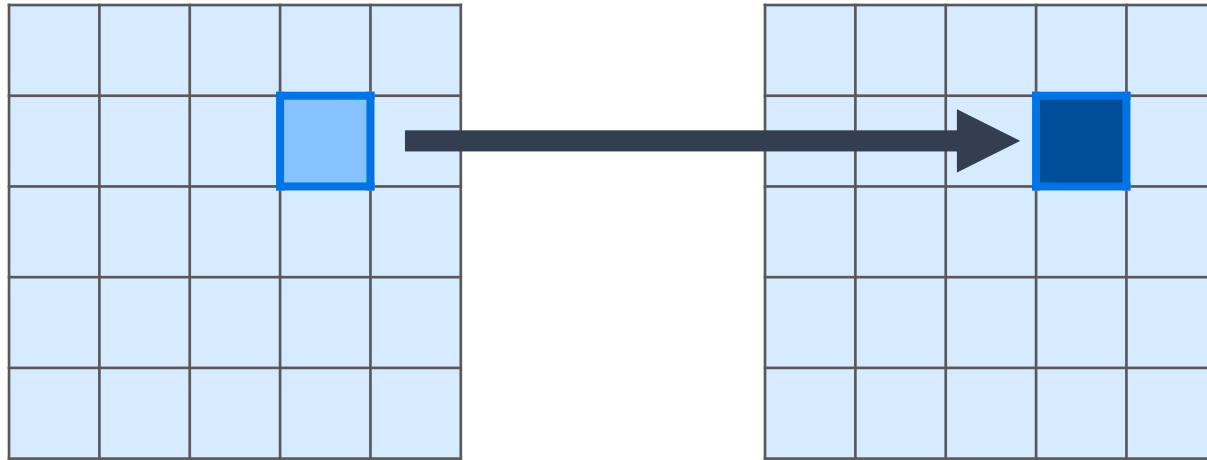
## 画像を加工して新たな画像を作る

- ✦ 画素ベースの画素値変換
- ✦ 領域ベースの画素値変換
- ✦ 幾何学変換
- ✦ 画像間演算



# 画素ベースの画素値変換

---



原画像

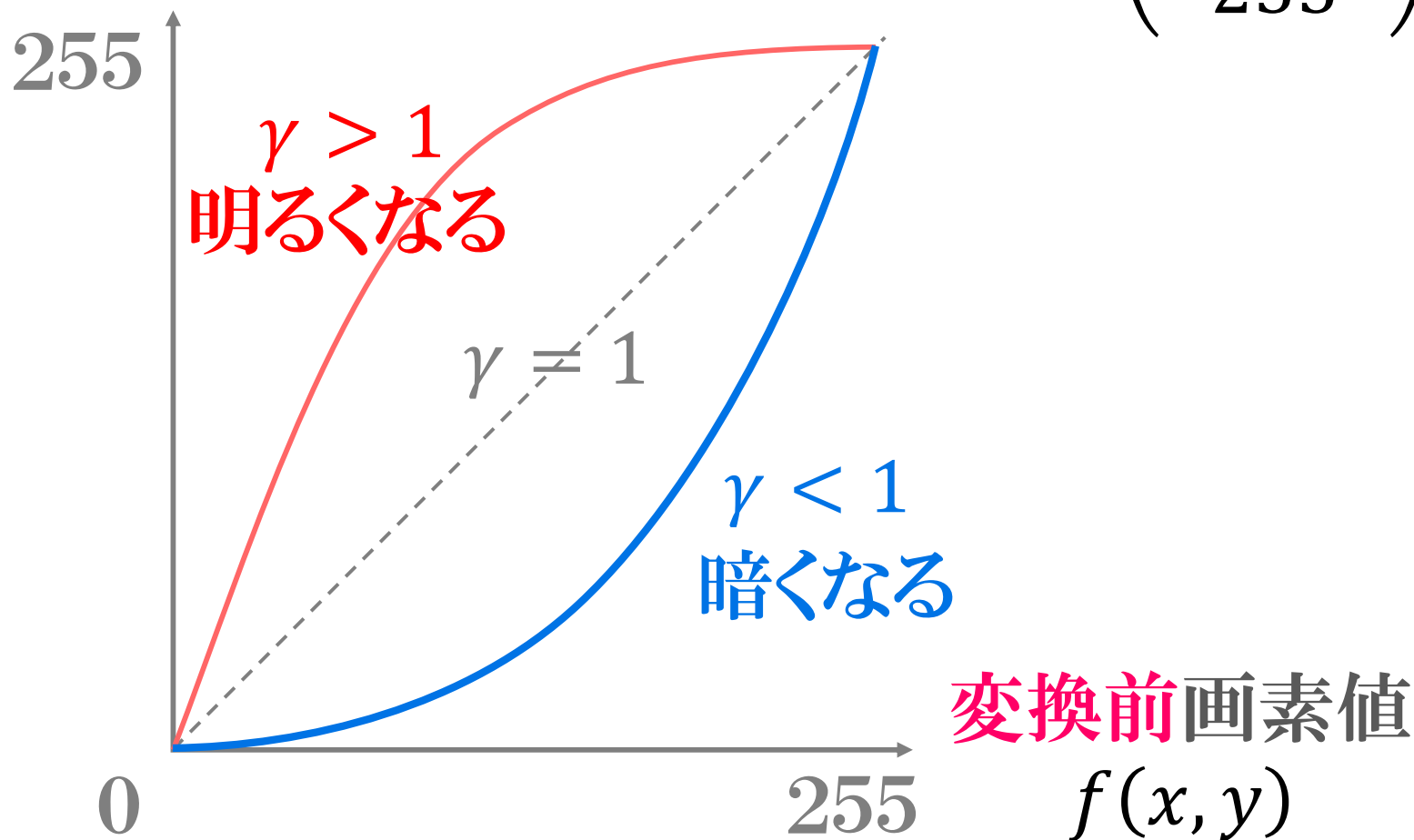
変換後画像

- 階調変換 (gray-level transformation)
- コントラスト強調 (contrast enhancement)

# ガンマ補正

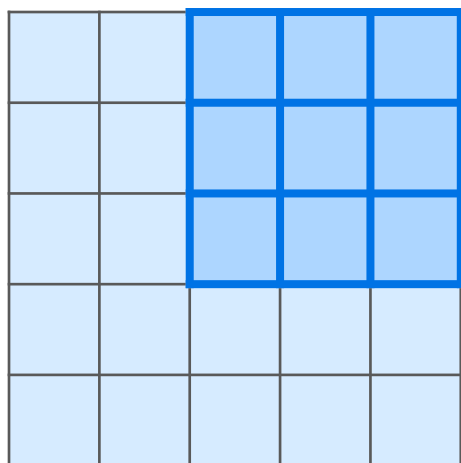
変換後画素値  
 $g(x, y)$

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



# 領域ベースの画素値変換

重要

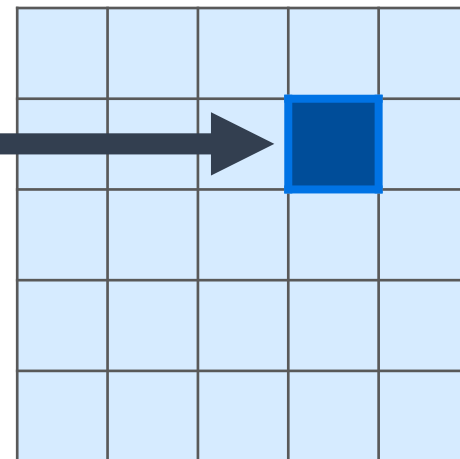


原画像

$\otimes$

$a_0$	$a_1$	$a_2$
$a_3$	$a_4$	$a_5$
$a_6$	$a_7$	$a_8$

空間フィルタ  
(spatial filter)



変換後画像

- 平滑化 (smoothing)
- 輪郭抽出 (edge extraction)
- 鮮鋭化 (sharpening)

# 空間フィルタリング

## 畳み込み演算

$$g(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b f(x + s, y + t) \cdot w(s, t)$$

原画像

$a = 1, b = 1$  の場合

$w(-1, -1)$	$w(0, -1)$	$w(1, -1)$
$w(-1, 0)$	$w(0, 0)$	$w(1, 0)$
$w(-1, 1)$	$w(0, 1)$	$w(1, 1)$

空間フィルタ

# 平滑化

---

画像をぼかす。ノイズを除去する。

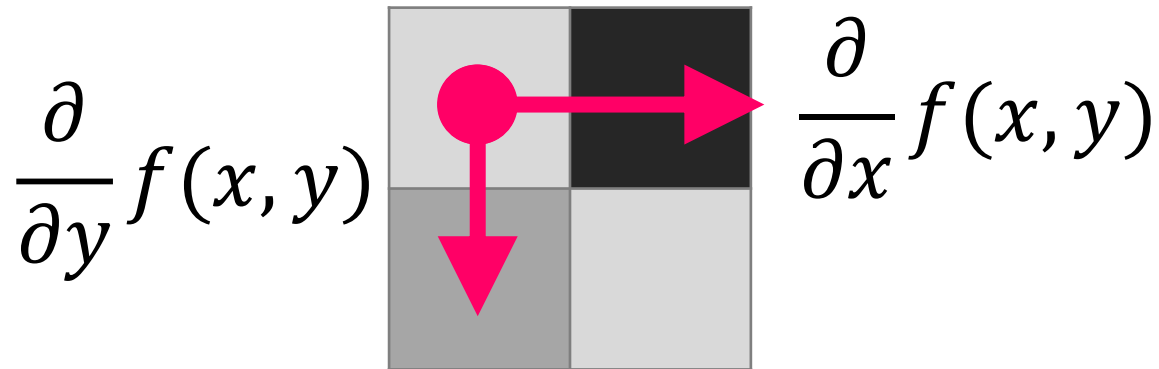
- ✿ 平均値フィルタ (average filter)
- ✿ 重み付き平均値フィルタ  
(weighted average filter)
- ✿ ガウシアンフィルタ (Gaussian filter)
- ✿ メディアンフィルタ (median filter)

# 画像の微分

重要

輝度の変化点を検出する。輪郭を検出する。

輝度勾配 (intensity gradient)



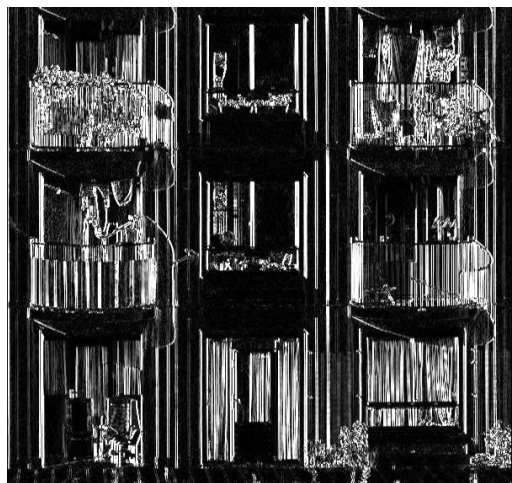
ラプラシアン (Laplacian)

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

# 1次微分の例

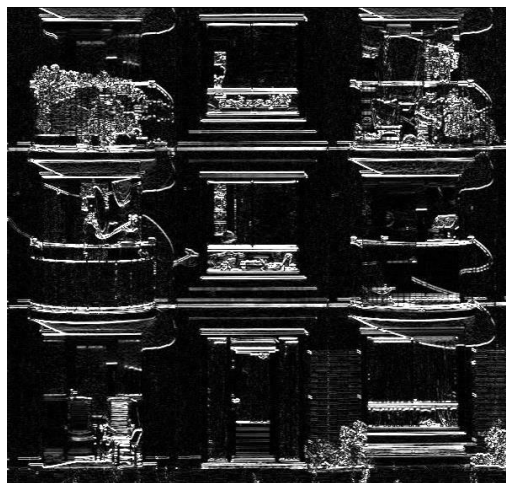
## 微分画像

微分値を画素値にした画像



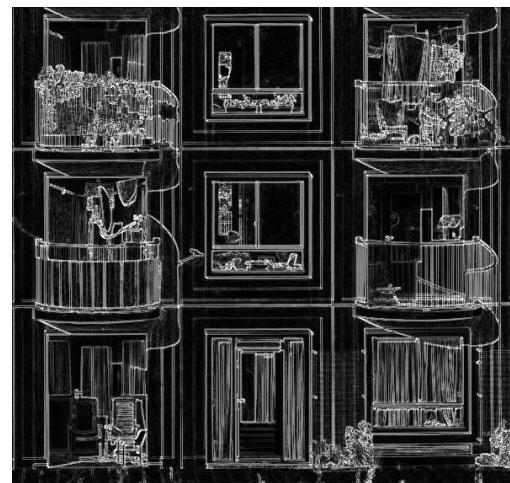
$x$ 軸方向の微分

縦向きのエッジを検出



$y$ 軸方向の微分

横向きのエッジを検出



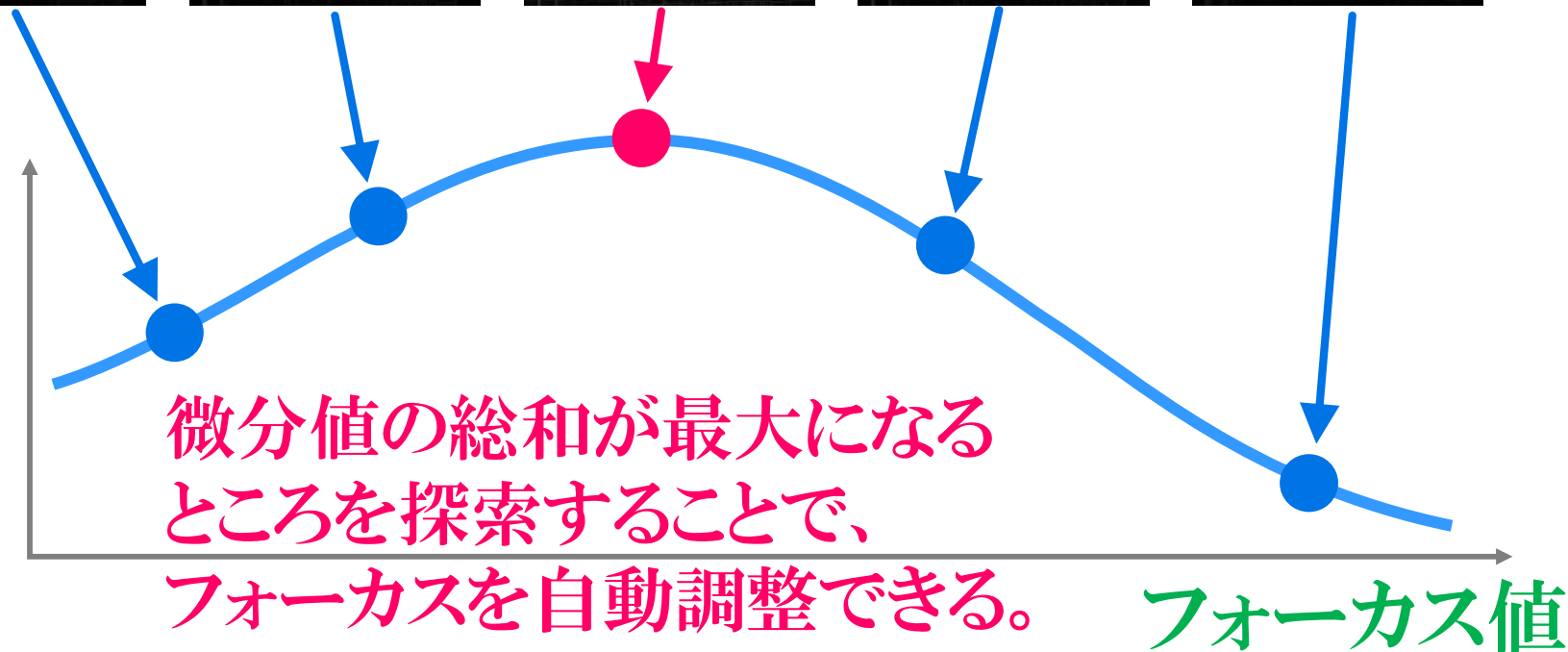
合成結果

# オートフォーカス

応用



全面素の  
微分値の総和





# 鮮鋭化フィルタ

---

原画像の濃淡を残したまま、エッジを強調する。

$$g(x, y) = f(x, y) - \nabla^2 f(x, y)$$

0	0	0
0	1	0
0	0	0

 $-$ 

0	1	0
1	-4	1
0	1	0

 $=$ 

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

# 鮮鋭化の例

---



原画像



鮮鋭化画像

# 幾何学変換

重要



- 線形変換 (linear transformation)
- アフィン変換 (affine transformation)
- 射影変換 (projective transformation)

# アフィン変換

重要

$$\begin{cases} x = aX + bY + c \\ y = dX + eY + f \end{cases} \rightarrow \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}$$

拡大縮小

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}$$

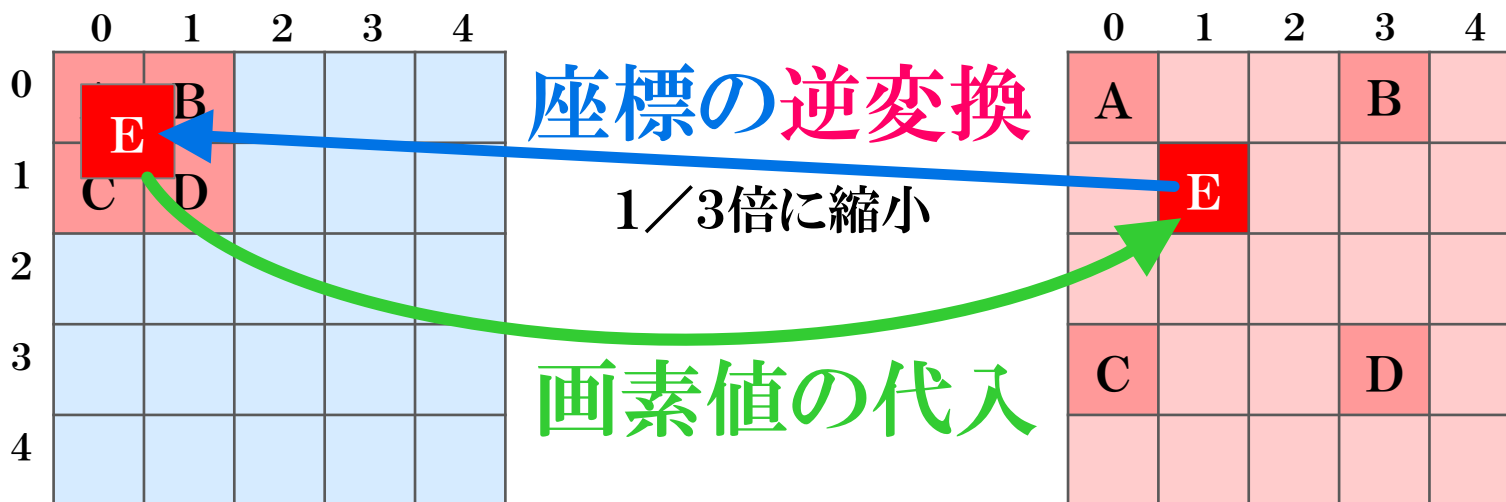
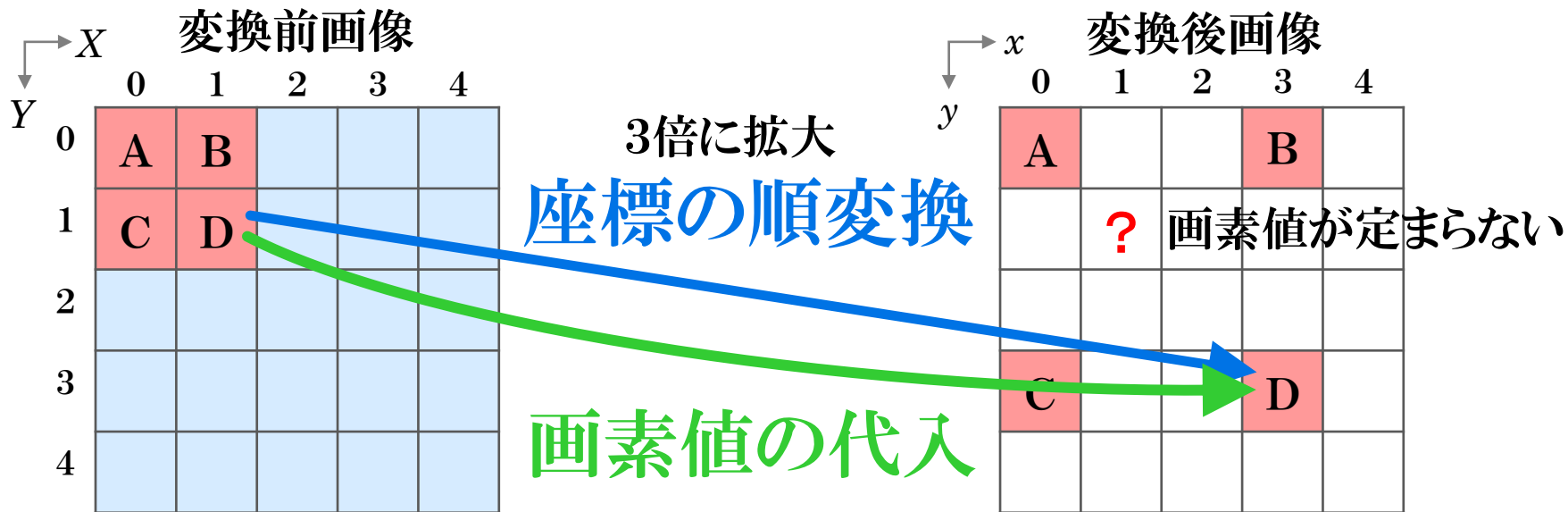
平行移動

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}$$

回転

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}$$

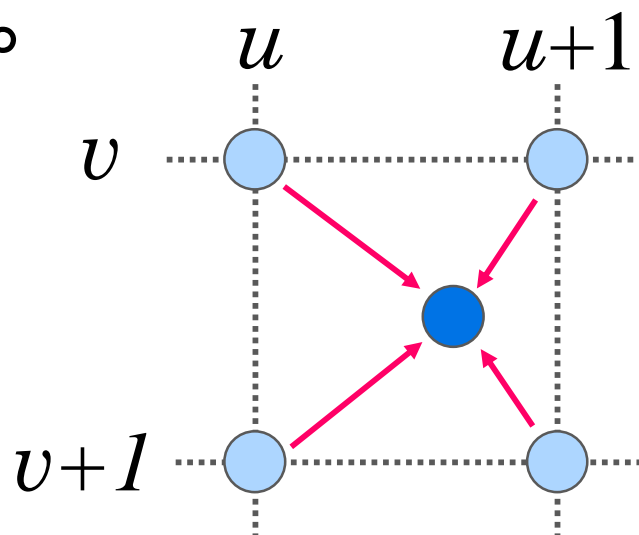
# 逆変換の利用



# 再標本化 (resampling)

重要

標本化された画像から、実数座標の画素値を求める。

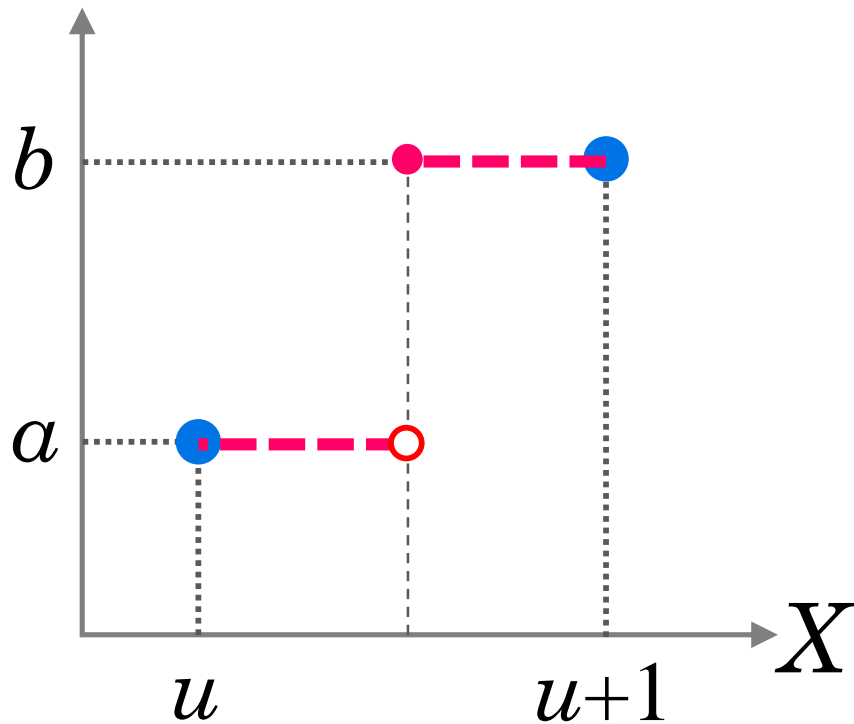


$u, v$  : 整数値

- 最近隣内挿法  
(nearest neighbor interpolation)
- バイリニア法 (bi-linear interpolation)

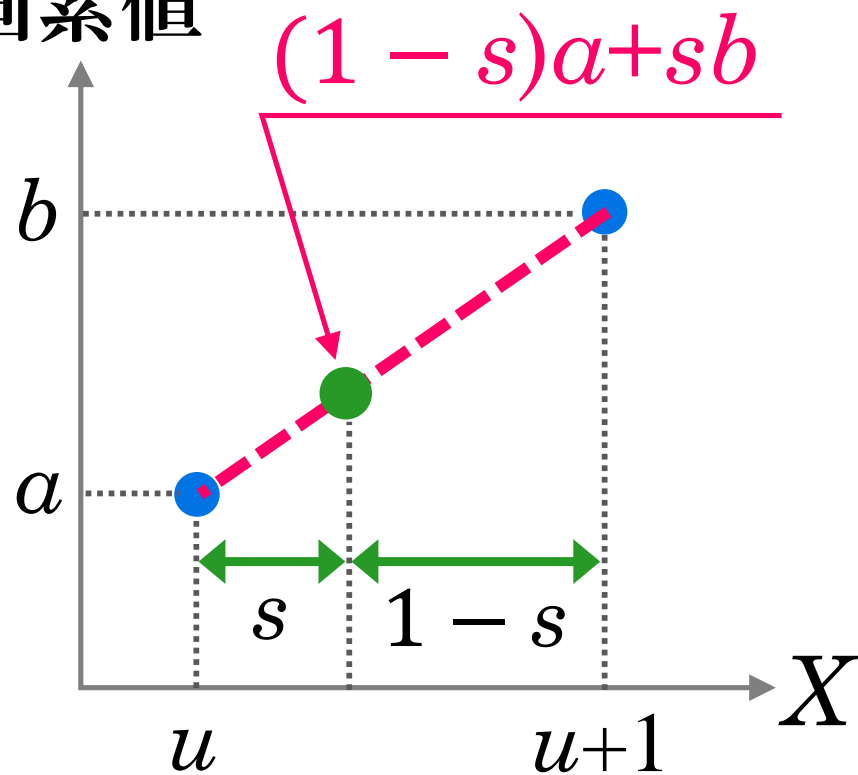
# 内挿

画素値



最近隣内挿

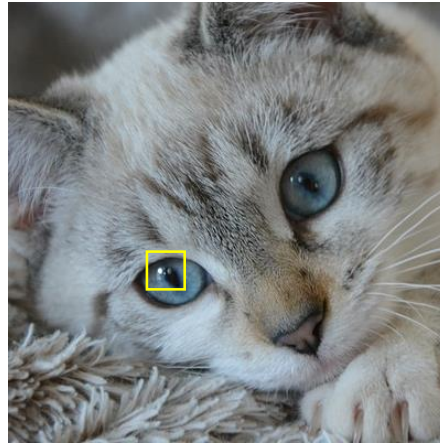
画素値



バイリニア

# 内挿法の比較

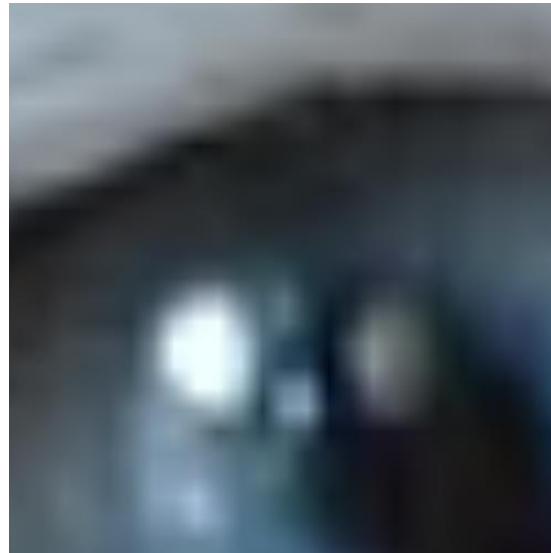
---



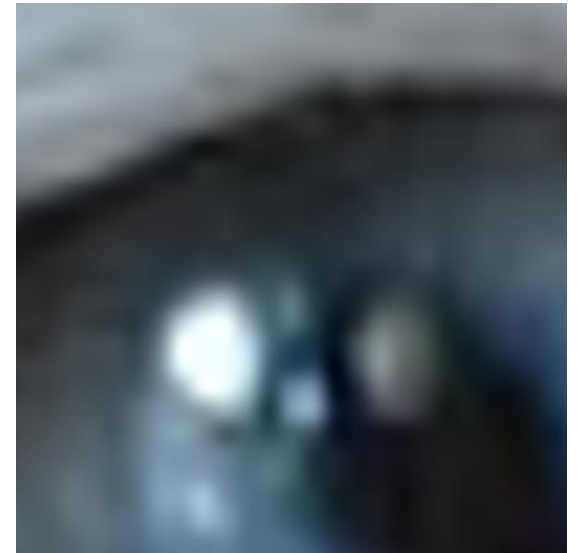
20倍に拡大



最近隣内挿



バイリニア

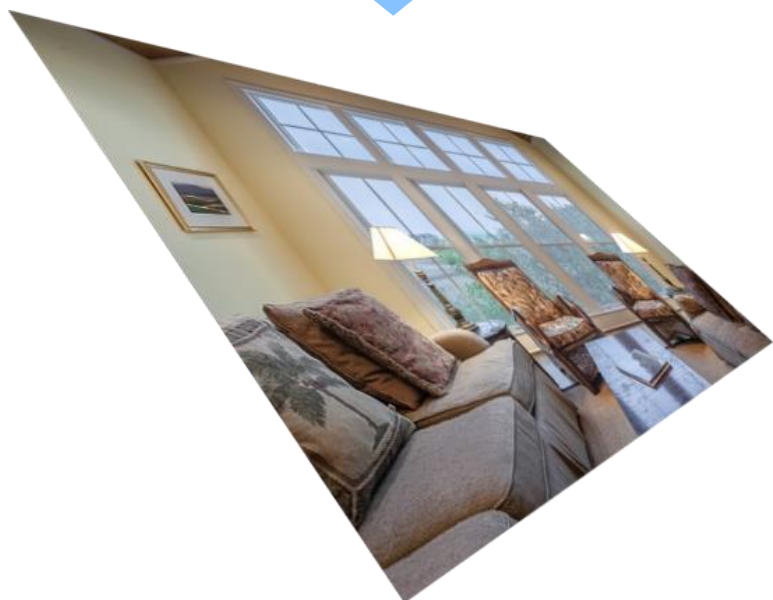


バイキュービック  
(bi-cubic)



# その他の変換：射影変換

遠近感のある変換を行う。



$$\begin{pmatrix} X' \\ Y' \\ Z \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

$$X = \frac{X'}{Z}$$

$$Y = \frac{Y'}{Z}$$

$x, y$  変換後座標  
 $X, Y$  変換前座標

# その他の変換：球面変換

球面への貼り付け行う。

$$\theta = \tan^{-1} \frac{y - y_0}{x - x_0}$$

$$r = \frac{2R}{\pi} \cdot \sin^{-1} \frac{\sqrt{(x - x_0)^2 + (y - y_0)^2}}{R}$$

$$X = x_0 + r \cos \theta$$

$$Y = y_0 + r \sin \theta$$

$x, y$  変換後座標

$X, Y$  変換前座標

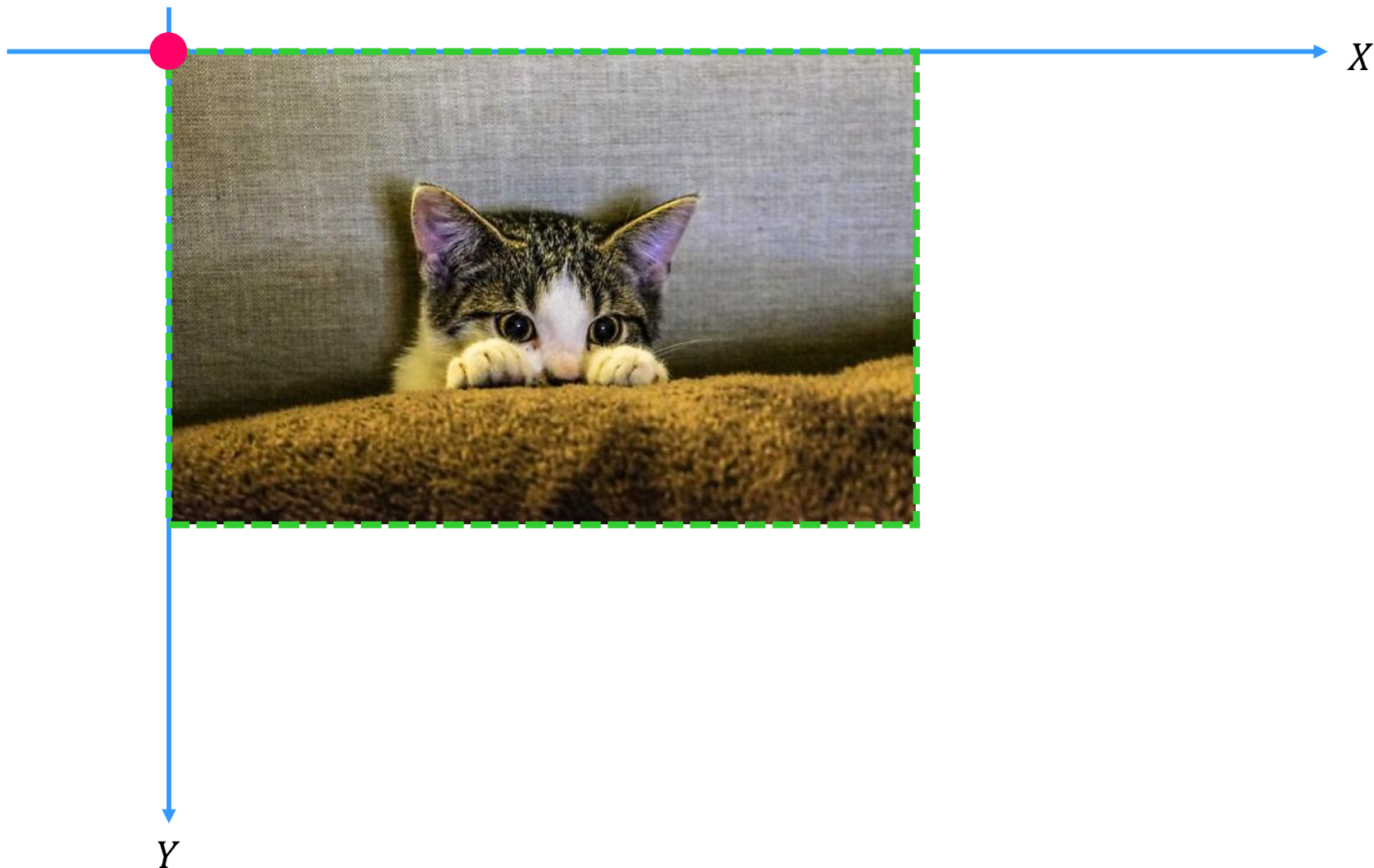
$x_0, y_0$  球の中心座標

$R$  球の半径

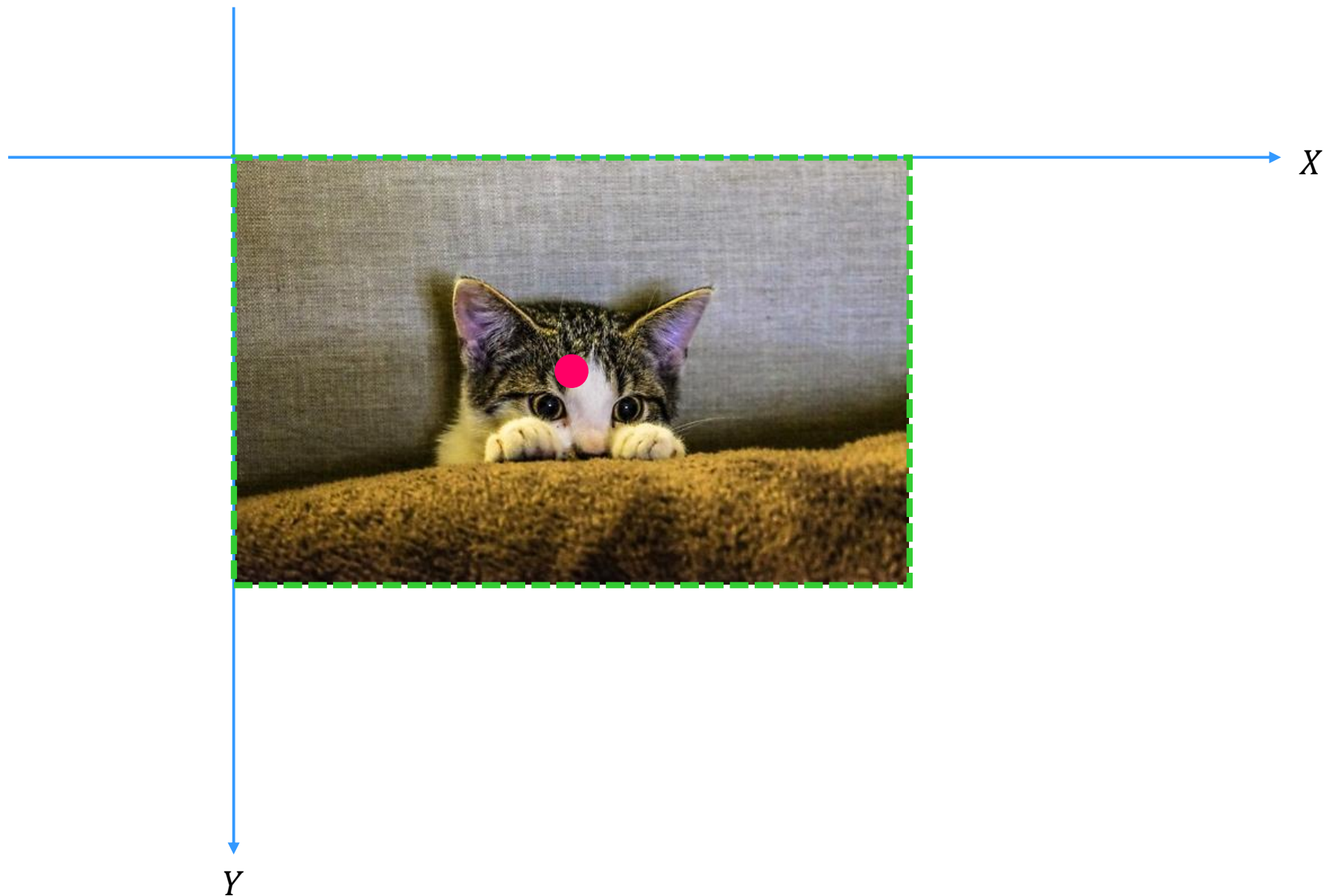


# 拡大縮小、回転における制限

原点を中心にして拡大縮小、回転をする。

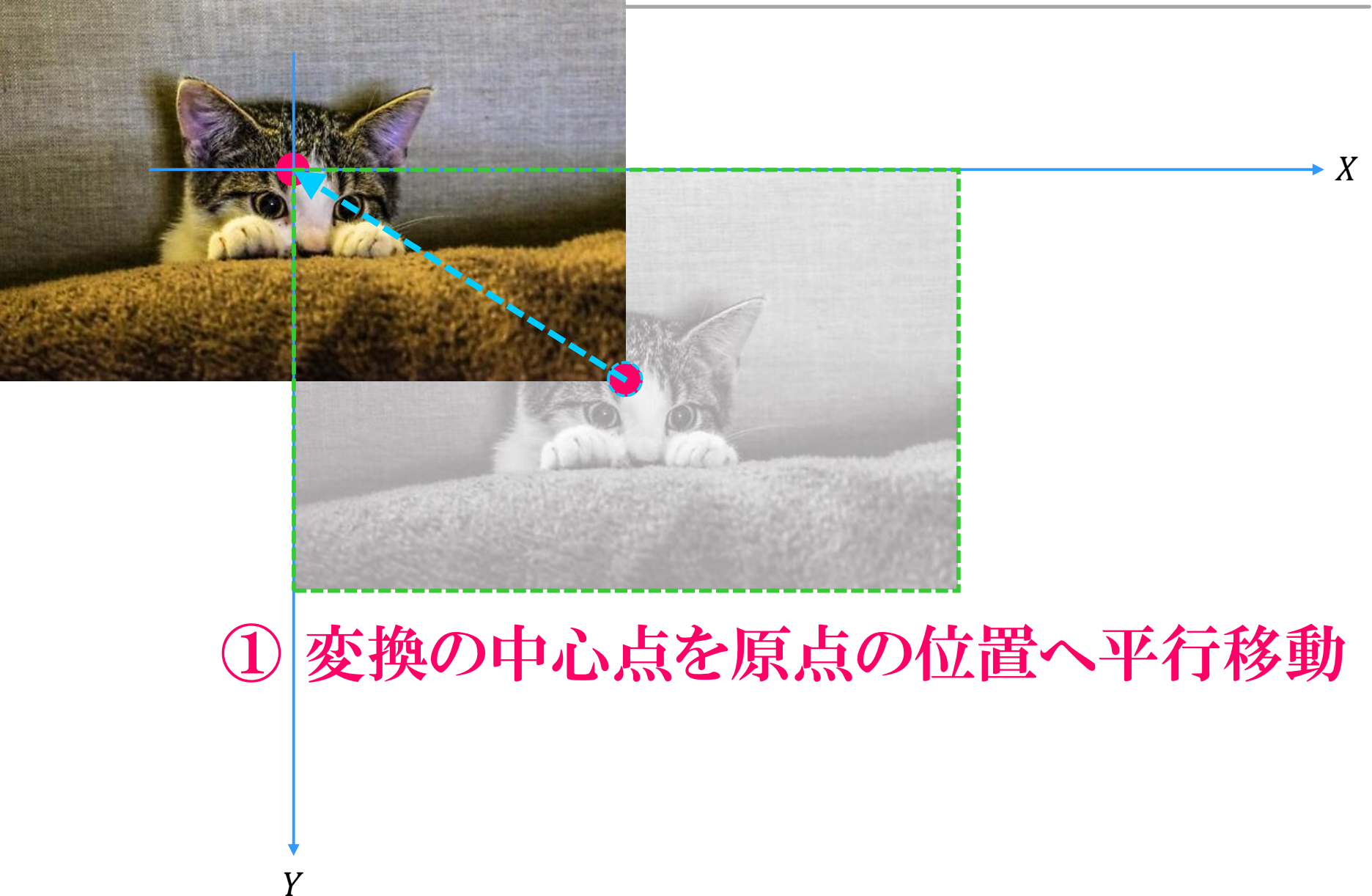


# 任意の点を中心にして拡大したい

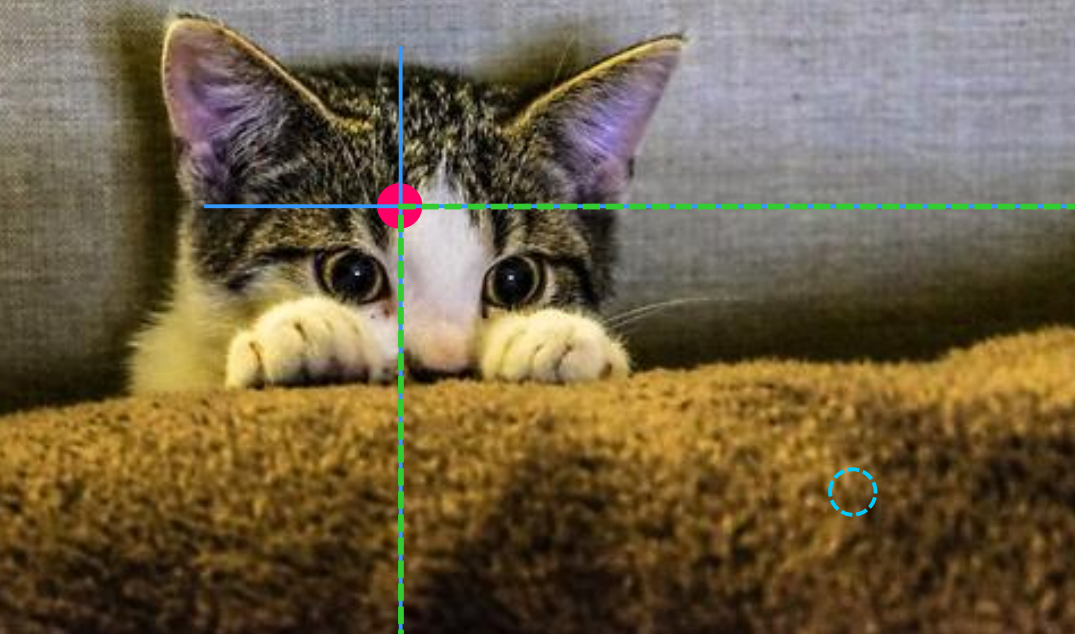




# 合成変換の考え方



# 合成変換の考え方

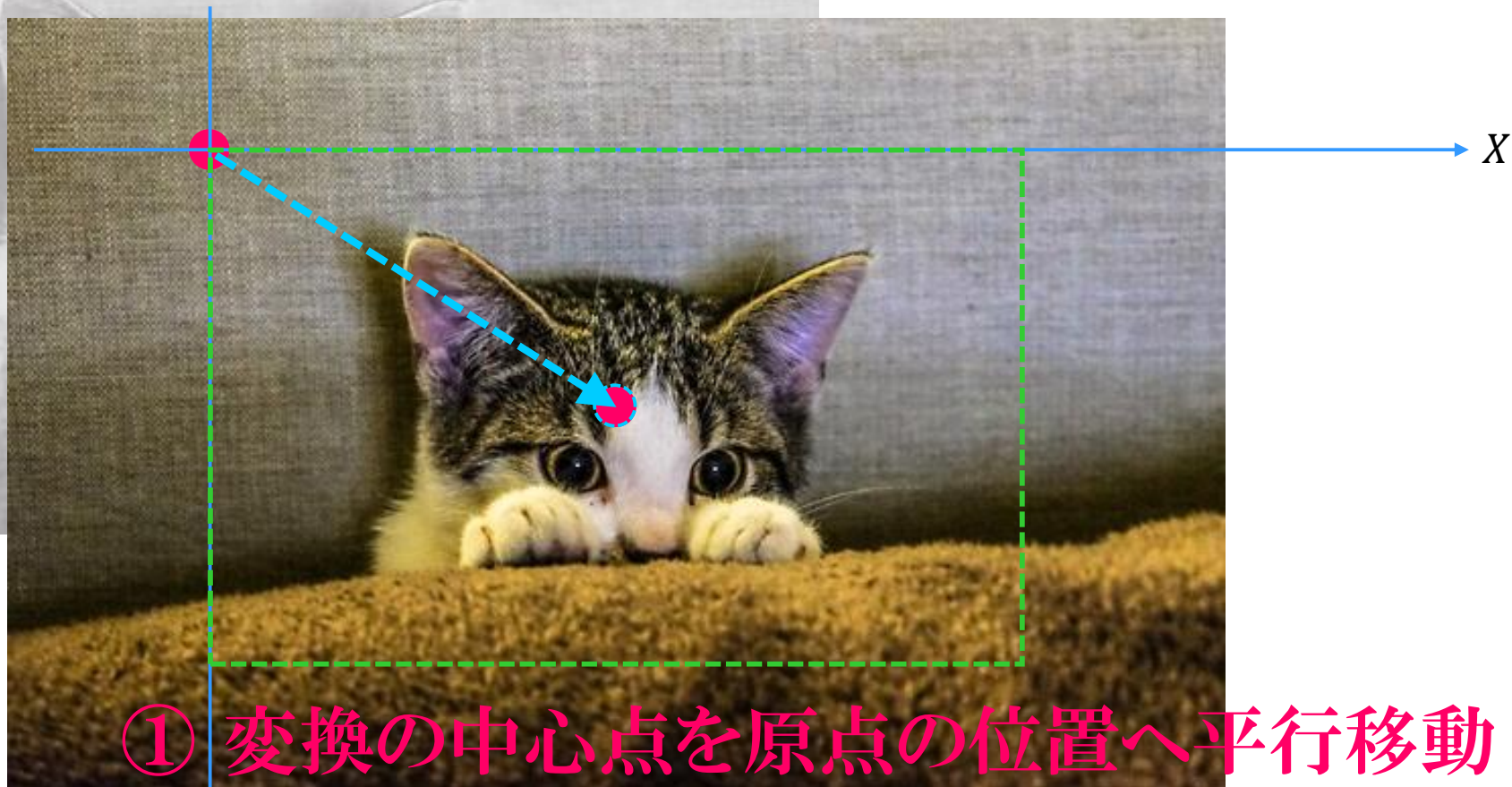


$X$

- ① 変換の中心点を原点の位置へ平行移動
- ② 拡大縮小、回転など

$Y$

# 合成変換の考え方

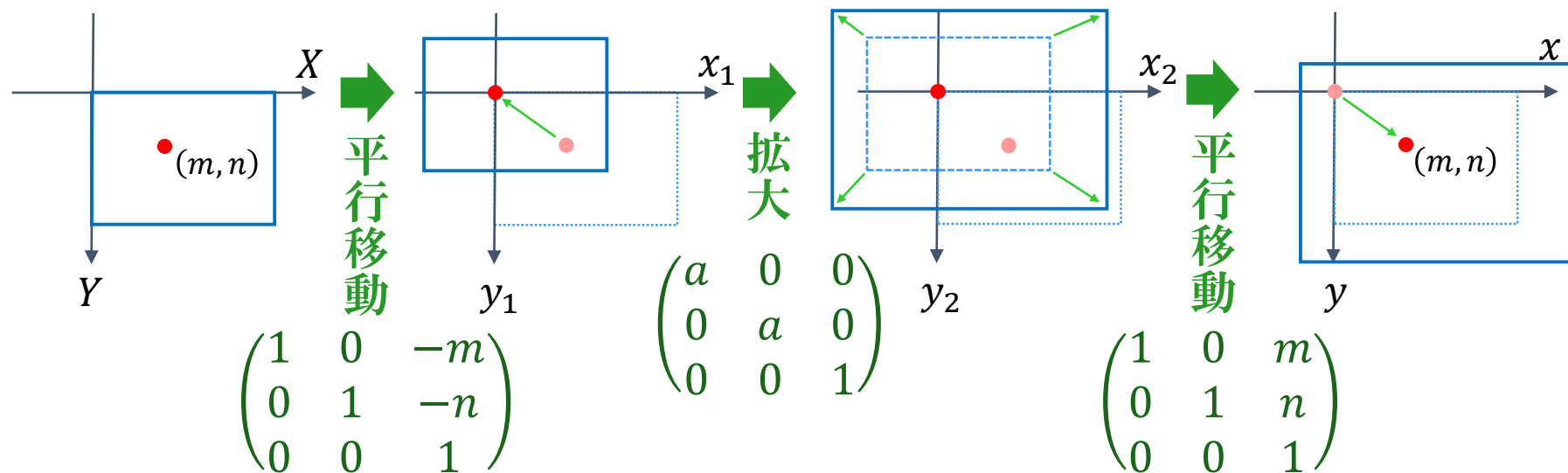


- ① 変換の中心点を原点の位置へ平行移動
- ② 拡大縮小、回転など
- ③ 変換の中心点を元の位置へ平行移動

Y

# 合成変換

例) 画像の座標 $(m, n)$ を中心にして $a$ 倍に拡大



順変換 
$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & m \\ 0 & 1 & n \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & -m \\ 0 & 1 & -n \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}$$

逆変換 
$$\begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & -m \\ 0 & 1 & -n \\ 0 & 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 1 & 0 & m \\ 0 & 1 & n \\ 0 & 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$



# スマホの写真表示の操作

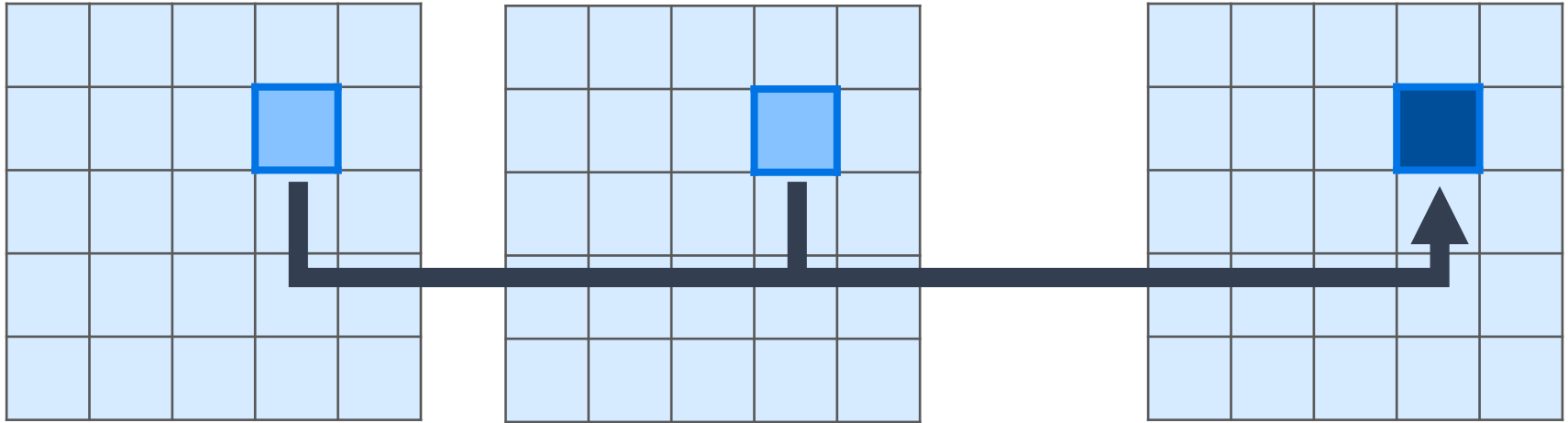
応用



- ① 2本の指の移動前と移動後の座標を検出する。
- ② 幾何学変換式に座標値を入れ、式を解いてパラメータ値(平行移動量、拡大縮小率、回転角度)を求める。
- ③ 得られたパラメータ値を基にして画像全体を幾何学変換する。

# 画像間演算

---



原画像A

原画像B

変換後画像

- ブレンディング (blending)
- 差分画像 (subtraction image)

# ブレンディング



画像A

+



画像B



加算



アルファブレンディング

# マスク処理

---



原画像

×



マスク画像





# 画像間平均(入力画像)



# 画像間平均(演算結果)



画像間平均値



画像間中央値



# 画像間差分(移動物体検出)



画像A



画像B(背景)



差分画像  $|A - B|$

# 2値画像処理

---

画像から対象物を抽出して、その形状特徴などから対象物の認識を行う。

- ① 各画素を**対象物**と**背景**とに分類する。
- ② 対象物の画素の連結を調べ、対象物の領域を抽出する。
- ③ 対象物の固有の特徴を検出する。



# 2値化 (binarization)

---

画素を対象物と背景とに分ける。

ある画素値をしきい値にして、白と黒の2階調に画像を変換する。

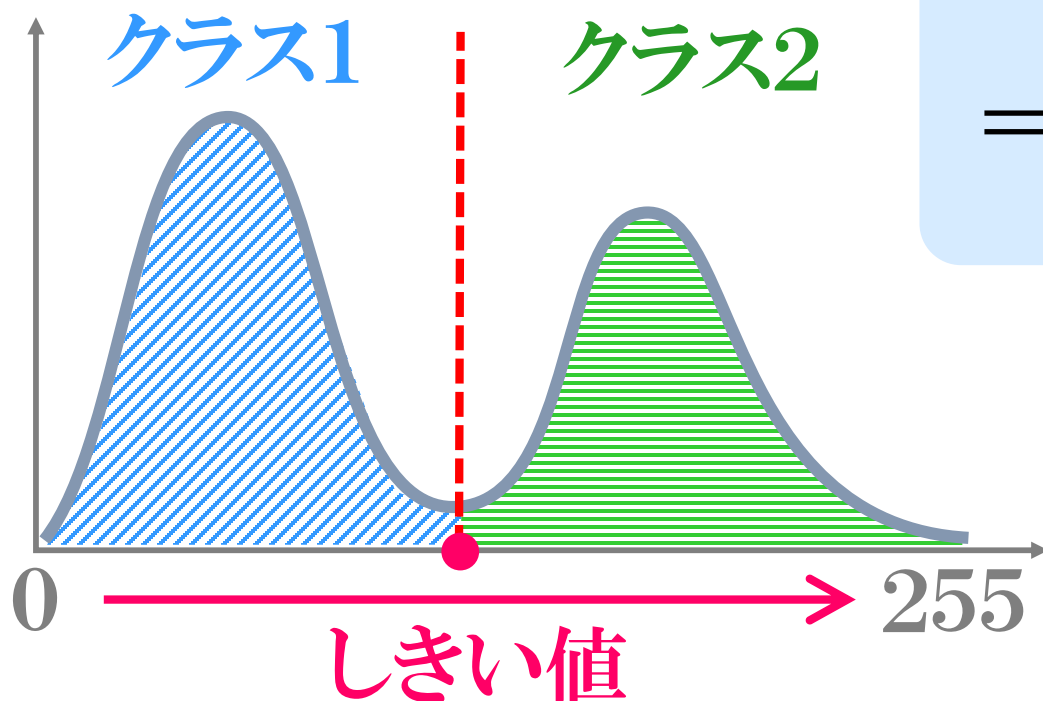
## しきい値の決定手法

- モード法 (mode method)
- P-タイル法 (p-tile method)
- 判別分析法 (discriminant analysis method)

# 判別分析法

ヒストグラムをしきい値で二つに分けるととき、  
その**分離度が最大**になるしきい値を求める。

ヒストグラム



**分離度**

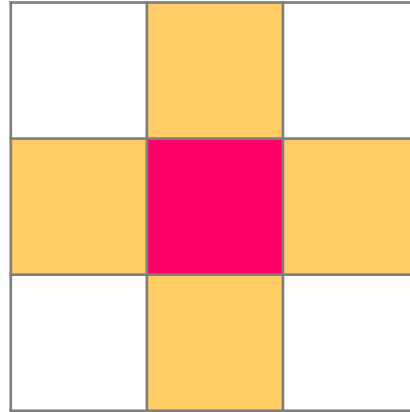
$$= \frac{\text{クラス間分散} \sigma_B^2}{\text{クラス内分散} \sigma_W^2}$$

しきい値を0から255  
まで変化させながら、  
分離度が最大になる  
ところを探す。

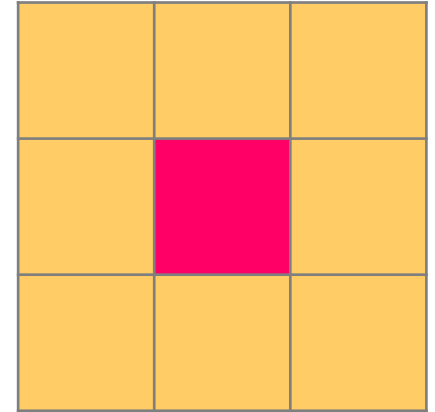
# 連結性 (connectivity)

隣り合う画素同士  
のつながり方

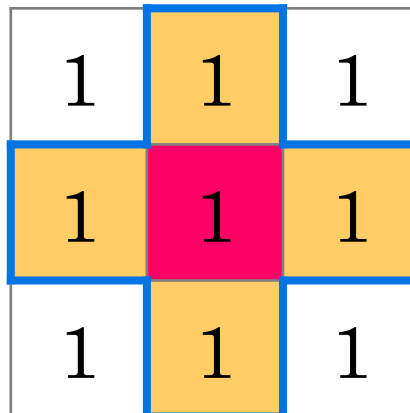
画素同士のつなが  
りを調べることで、  
領域が検出できる。



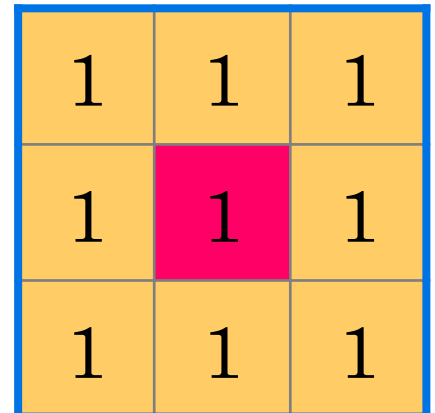
4近傍



8近傍



4連結



8連結

# ラベリング (labeling)

連結成分の個別番号(ラベル番号)を  
各画素に与える。

- 同一連結成分の画素は同じラベル番号になる。
  - ラベル番号は1から始める。
  - 背景の画素は0とする。
- |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|   |   |   |   |   |   |   |   |   |

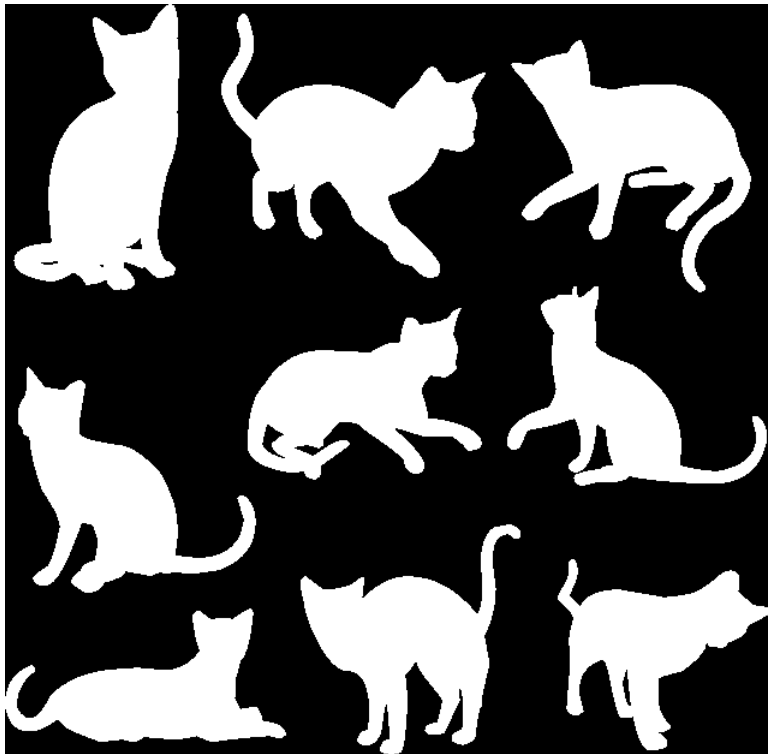
[illegible]

## 背景

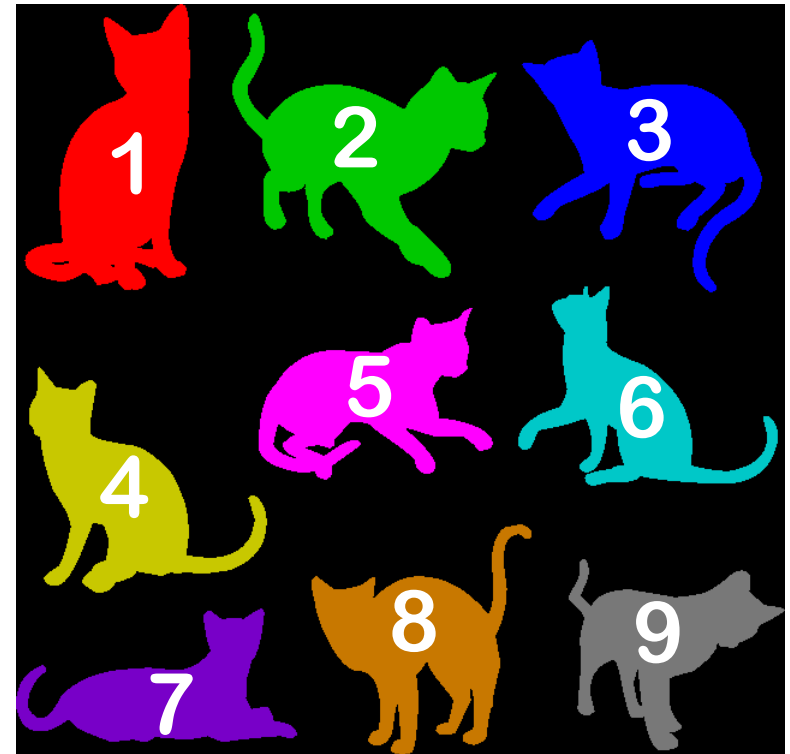
## 連結成分

# ラベリングの例

ラベリングによって、物体（個体）の数、個々の物体の占有領域が分かる。



2値画像



ラベリングの結果  
(ラベルごとに色付け)

# QRコードの読み取り

応用



2値化



連結成分検出  
マーカー検出

画像抽出  
幾何学変換



# モルフォロジー演算

---

✧ 膨張 (expansion)

✧ 収縮 (contraction)

✧ オープニング (opening)

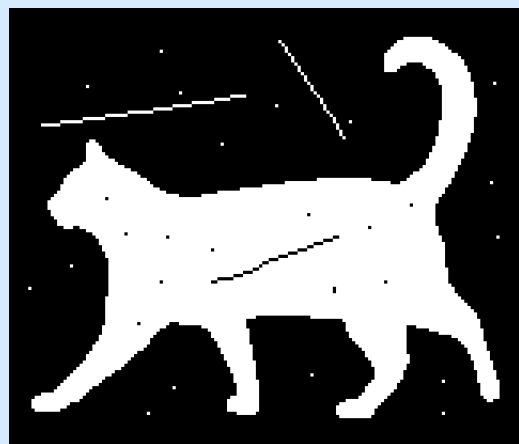
収縮n回 → 膨張n回

✧ クロージング (closing)

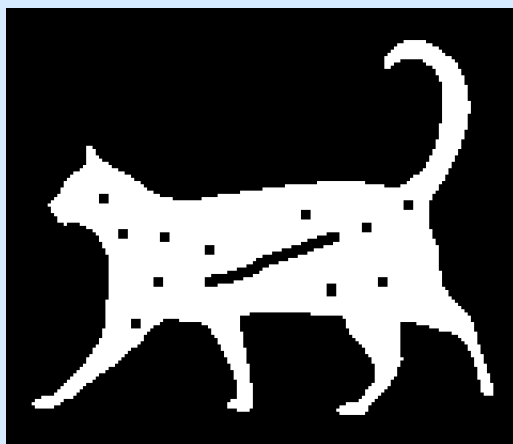
膨張n回 → 収縮n回

2値画像のノイズ除去に有効

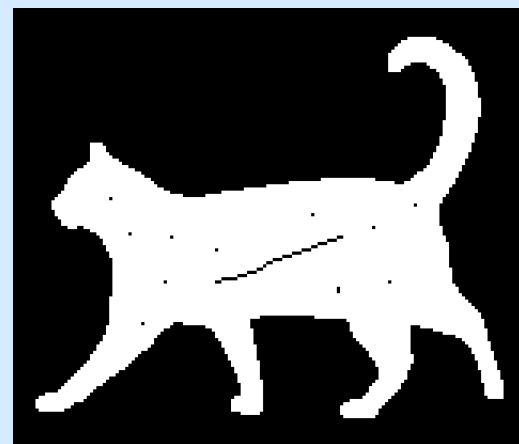
# オープニングとクロージングの例



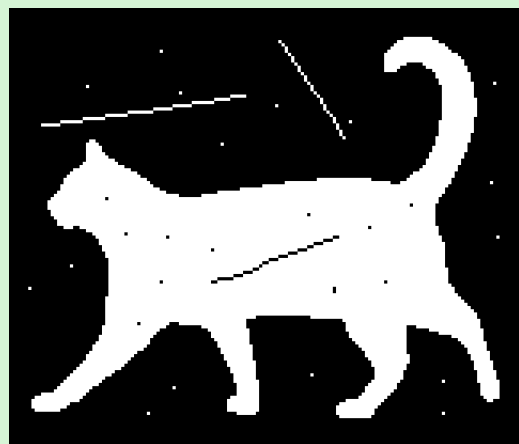
収縮



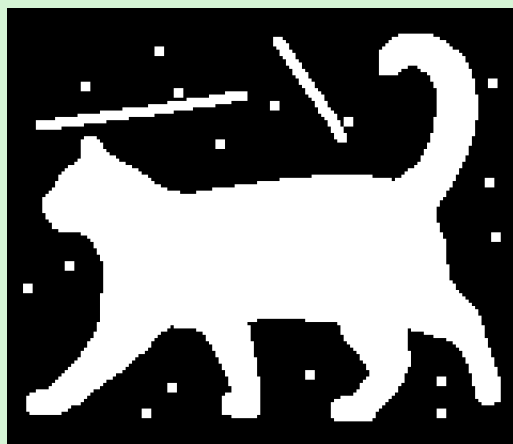
膨張



オープニング



膨張



収縮



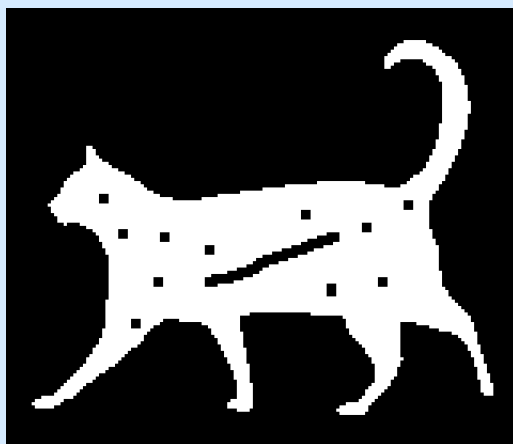
クロージング



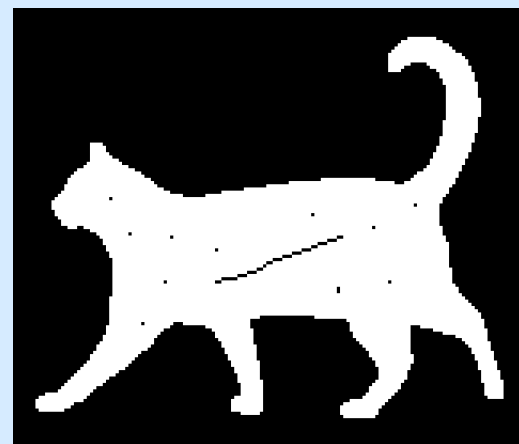
# オープニングとクロージングの例



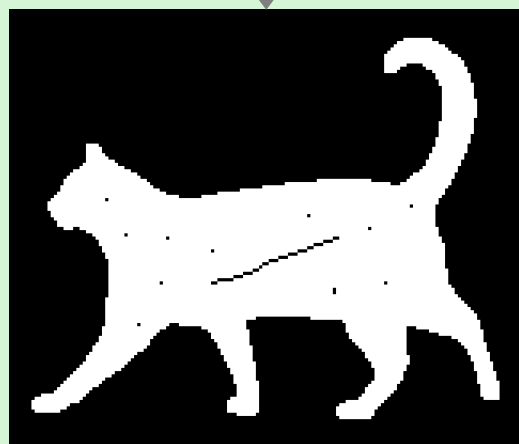
収縮



膨張



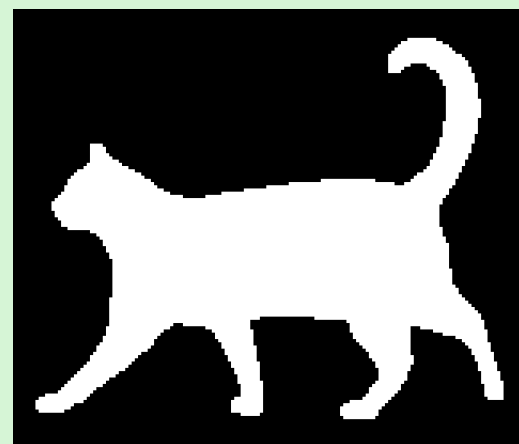
オープニング



膨張



収縮



クロージング

# 特徴抽出

---

連結成分ごとに特徴を抽出する。

- ✦ 形状特徴 (geometric feature)
- ✦ 骨格化 (skeletonization)
- ✦ 細線化 (thinning)

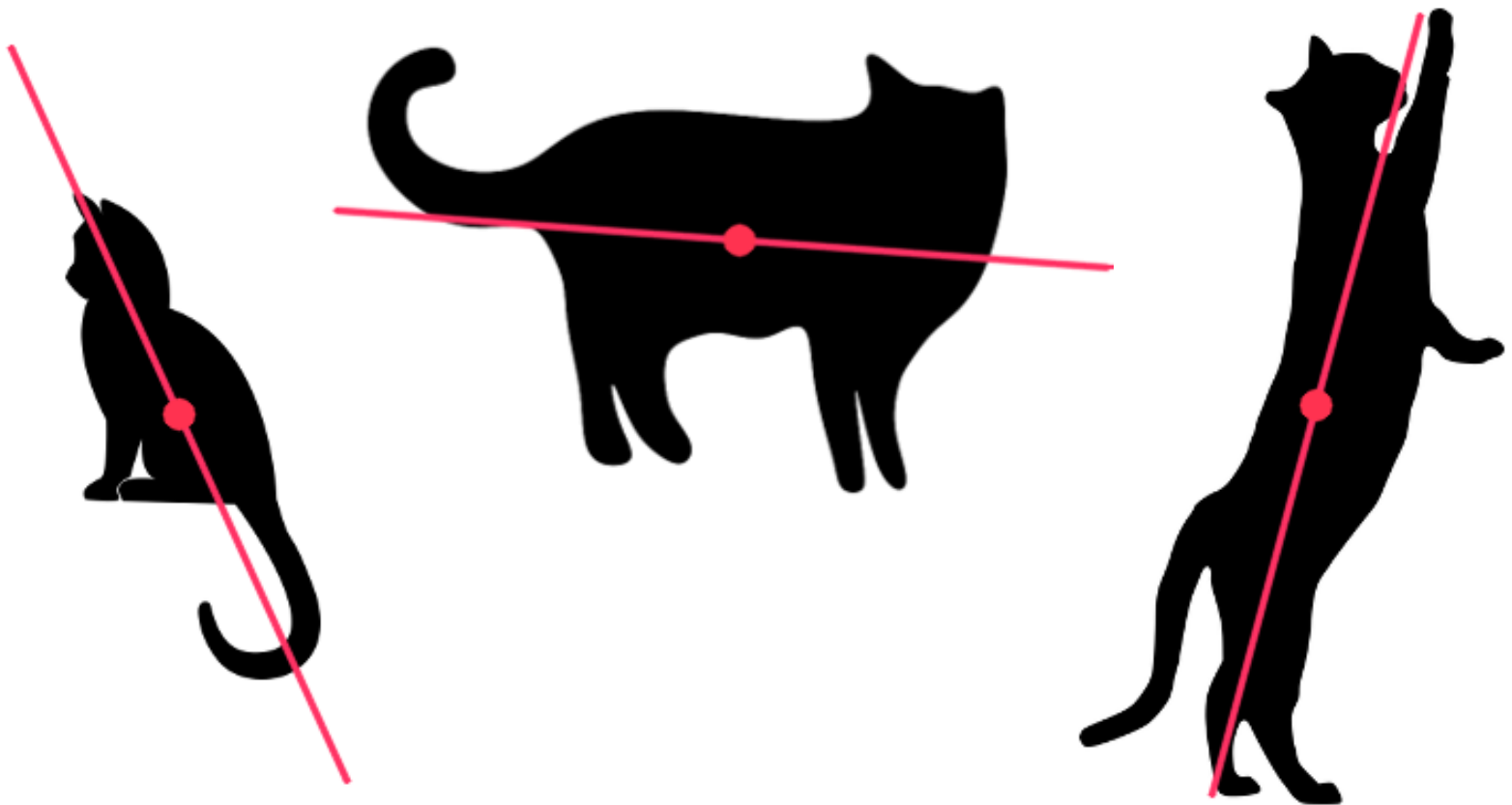
# 形状特徴

---

- ✿ 重心 (center of gravity)
- ✿ 外接長方形 (bounding box)
- ✿ 面積 (area)
- ✿ 周囲長 (perimeter)
- ✿ 円形度 (roundness)
- ✿ 主軸 (principal axis)

# 重心と主軸

---



(対象物を黒、背景を白としている)

# モーメント特徴

$$M_{ij} = \sum_x \sum_y x^i y^j f(x, y) \quad f(x, y) = \begin{cases} 0 & \text{背景} \\ 1 & \text{対象物} \end{cases}$$

面積  $M_{00}$

対象物上の全画素の $x^i y^j$ の総和

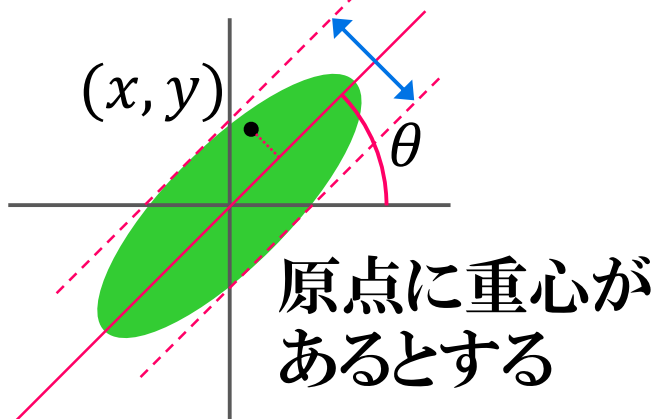
重心座標  $\left( \frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}} \right)$

主軸の角度  $\theta = \frac{1}{2} \tan^{-1} \frac{2m_{11}}{m_{20} - m_{02}}$

重心周りの  
モーメント  $m_{ij} = \sum_x \sum_y (x - x_g)^i (y - y_g)^j f(x, y)$

# 主軸の角度を求める式の導出

分散が最少になる



$$p(\theta) = \sum_x \sum_y (x \sin \theta - y \cos \theta)^2 f(x, y)$$

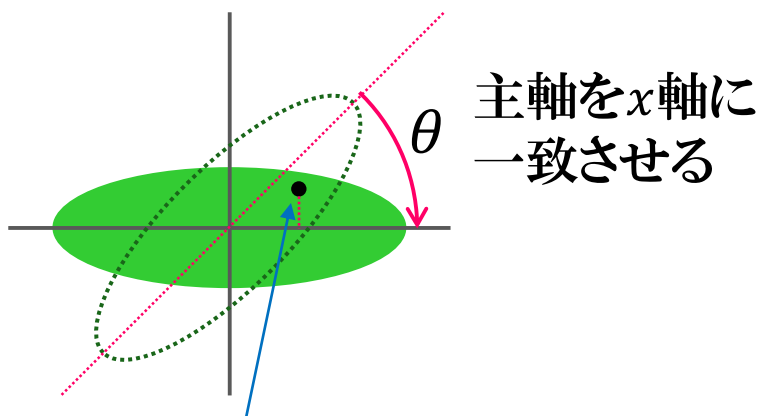
$p(\theta)$ が最少(極小)になる $\theta$ を求める。



$$\frac{d}{d\theta} p(\theta) = 0 \text{ を解く}$$

$$\tan 2\theta = \frac{2M_{11}}{M_{20} - M_{02}}$$

$$\theta = \frac{1}{2} \tan^{-1} \frac{2M_{11}}{M_{20} - M_{02}}$$



y座標

主軸との距離<sup>2</sup>

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

$$(x \sin \theta - y \cos \theta)^2$$

# 形状特徴認識の応用(ゲーム)

2007年7月21日 岡山リサーチパーク一般公開 おもしろ体験で一 出展参加

画像処理で脳トレ  
岡山理科大学工学部電気電子システム学科

## 同じ絵をさがせ!

残り時間 

得点 **27**

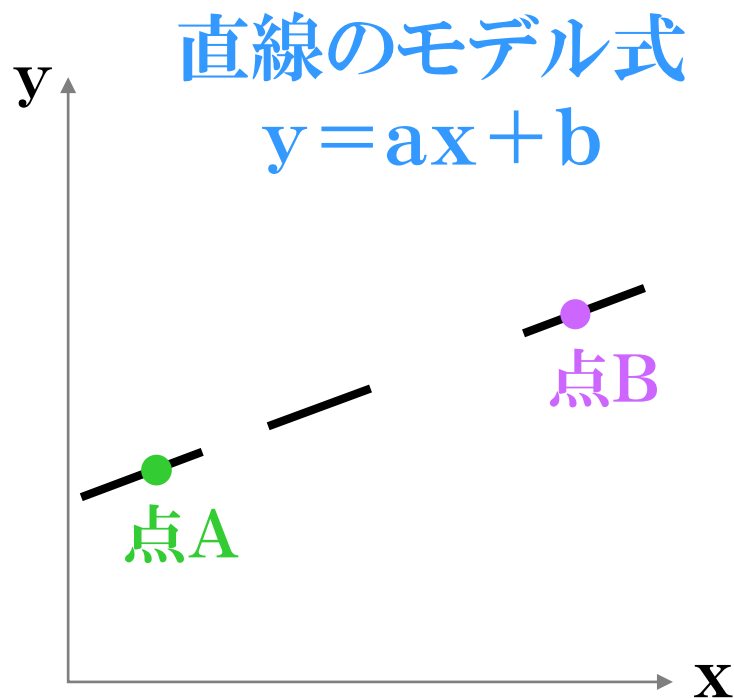




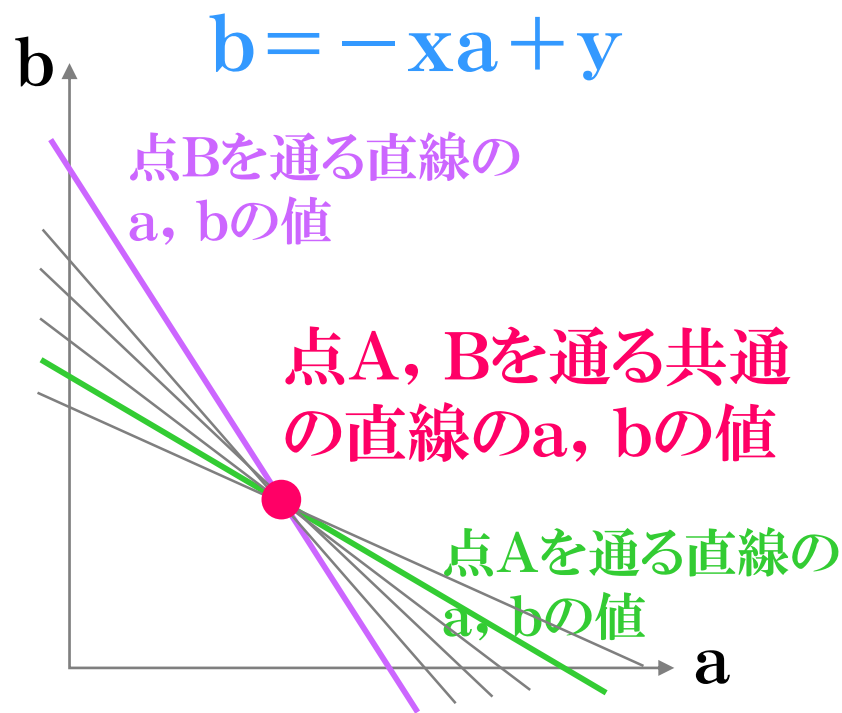
やさしい ふつう むずかしい 超ムズイ

# ハフ変換 (Hough transform)

不連続な輪郭線やノイズが含まれる画像から、モデル式で表された図形を抽出する。



x-y画像平面



パラメータ空間



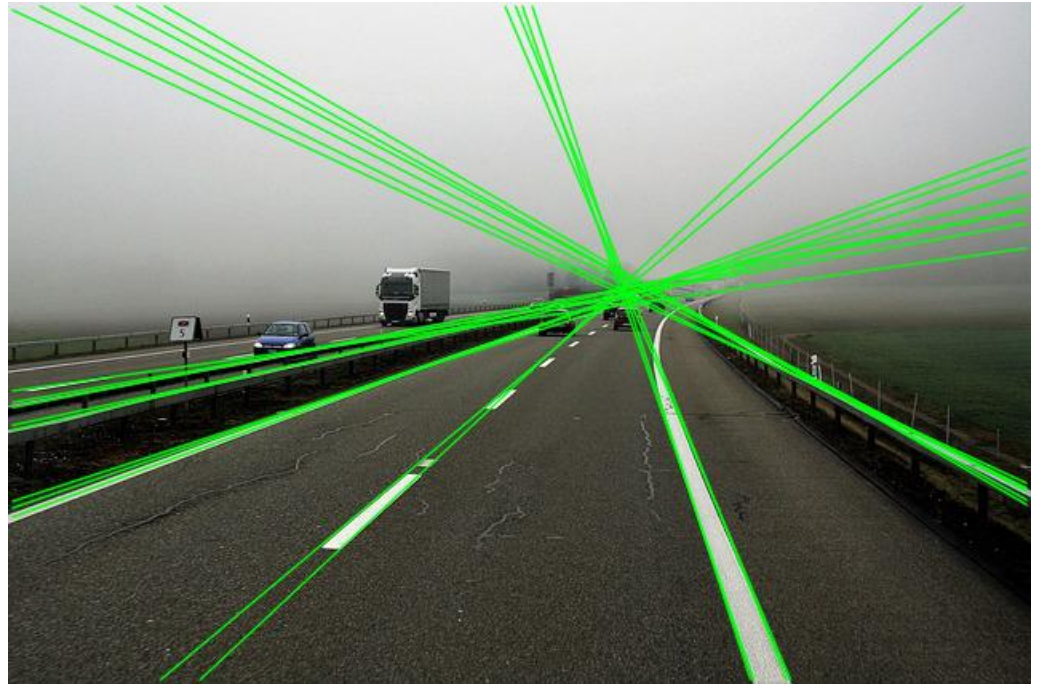
# 直線検出の例



原画像



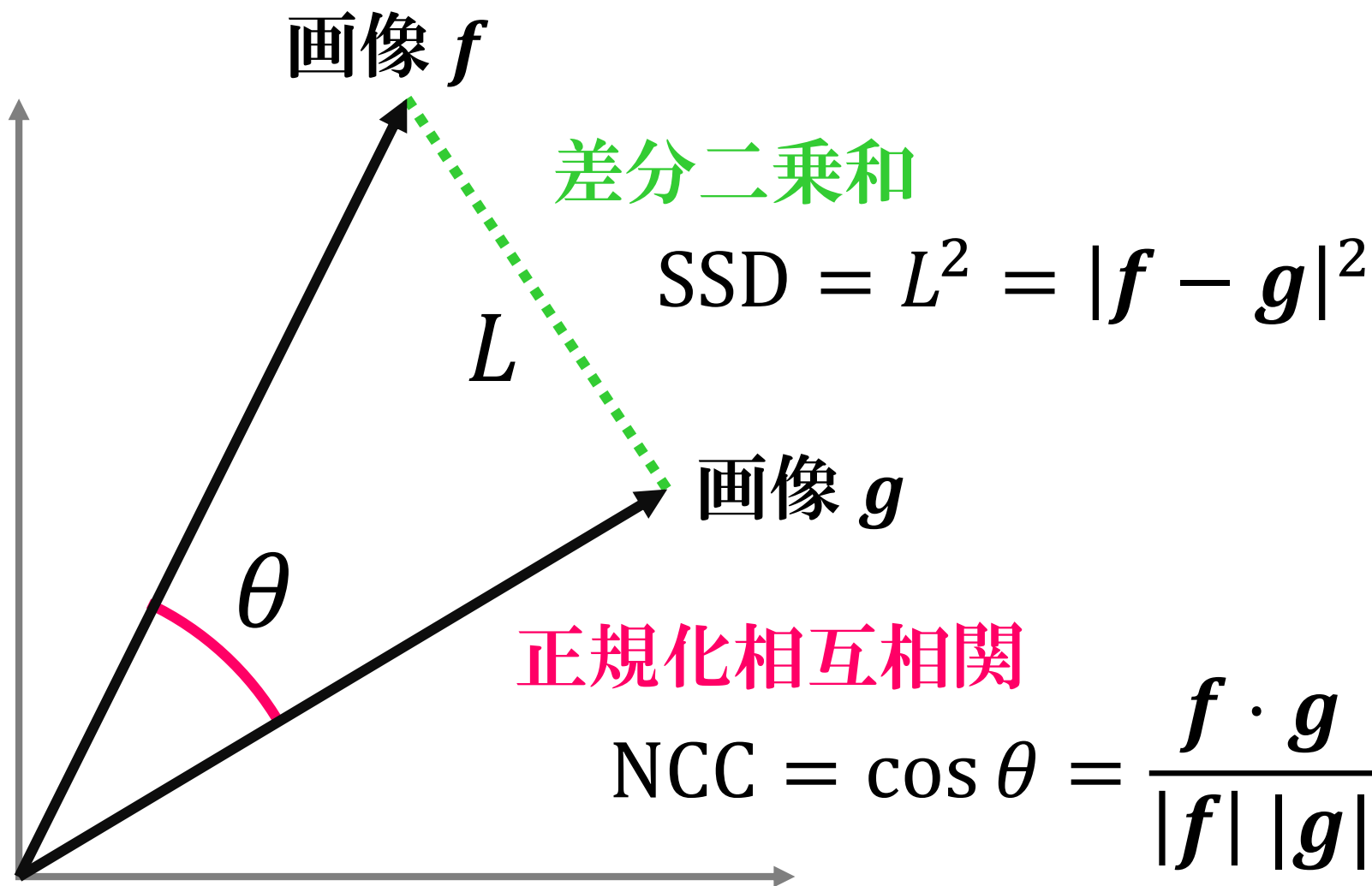
微分画像



ハフ変換による直線検出

# 2つの画像の類似度

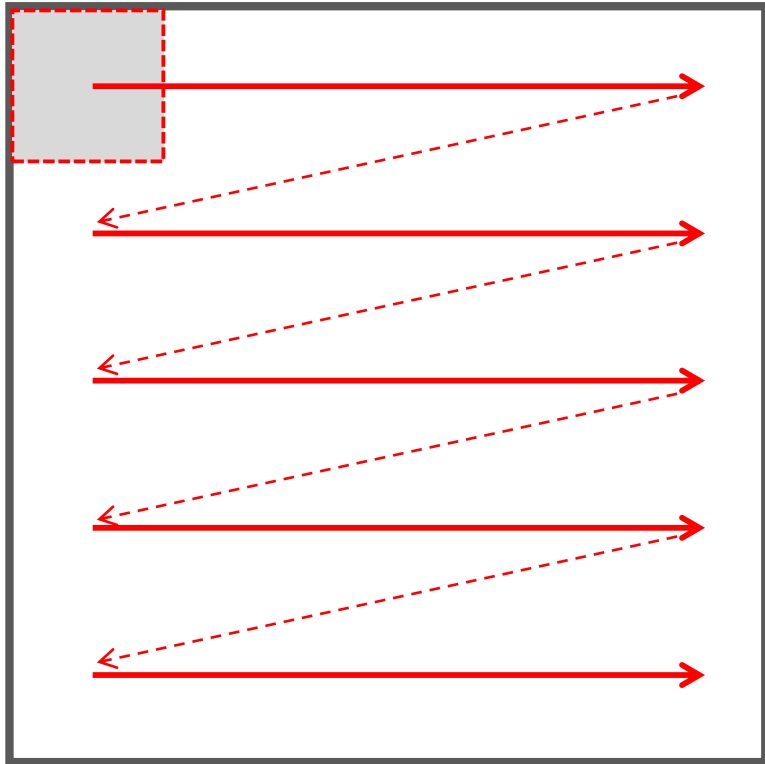
重要



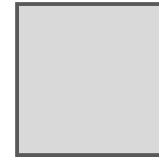
# テンプレートマッチング

重要

1画素ずつずらしながら類似度を計算



対象画像

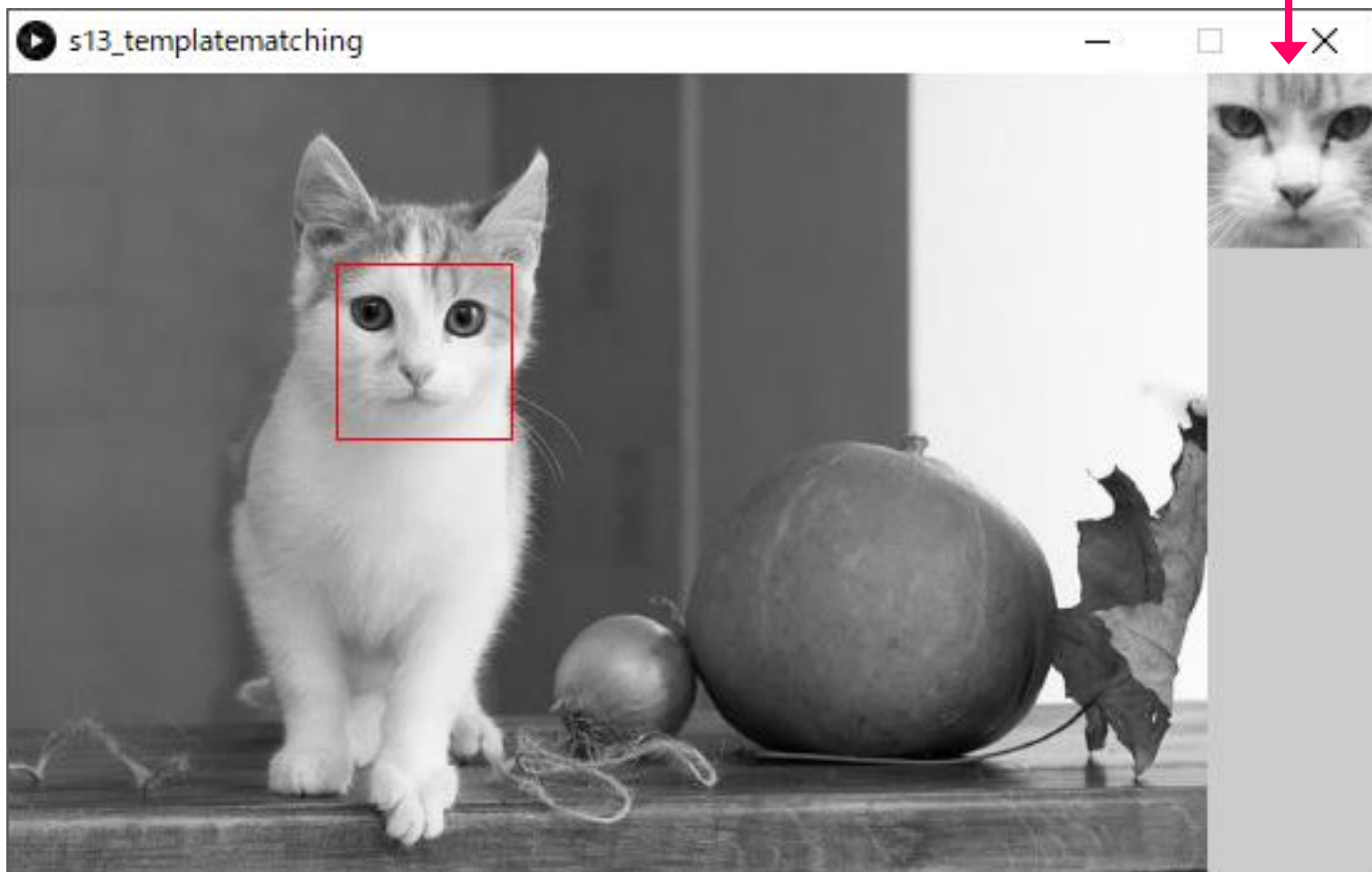


テンプレート画像  
(探索物の画像)

対象画像全体の中で、  
類似度が大きい座標を  
検出する。

# テンプレートマッチングの例

テンプレート画像



対象画像

# パターン認識

重要

画像の特徴を用いて、あらかじめ定めたクラスに画像を分類する処理

## ✧ 学習 (learning)

学習用画像群の特徴抽出を行い、各クラスのプロトタイプを定めて識別辞書を作る。

## ✧ 識別 (discrimination)

未知画像の特徴抽出を行い、識別辞書と比較して未知画像のクラスを決定する。

# パターン認識の流れ

学習

学習用画像群

特徴量

特徴抽出

特徴ベクトル

特徴空間

教師信号

クラス

プロトタイプ

識別

未知画像

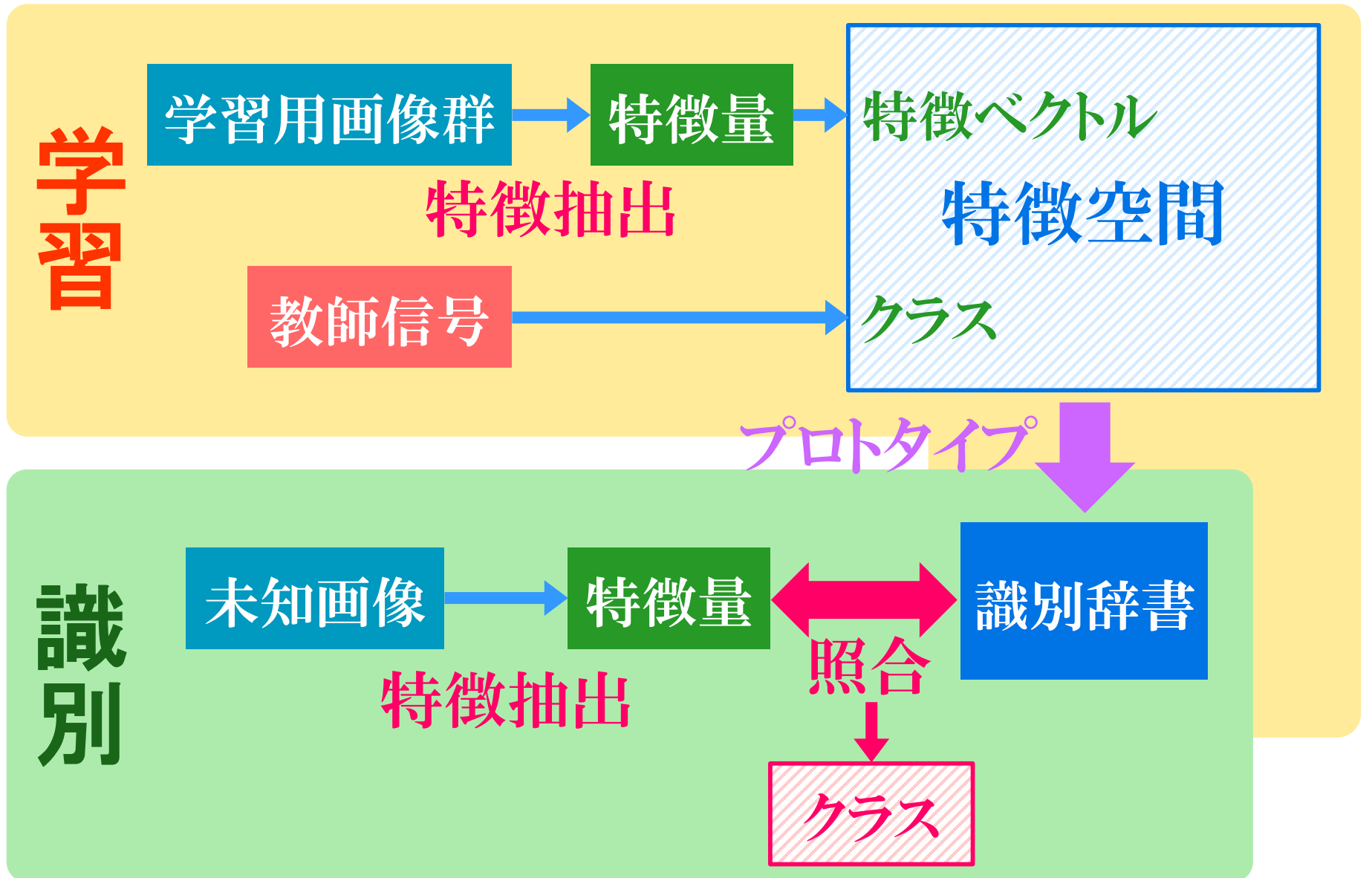
特徴量

特徴抽出

識別辞書

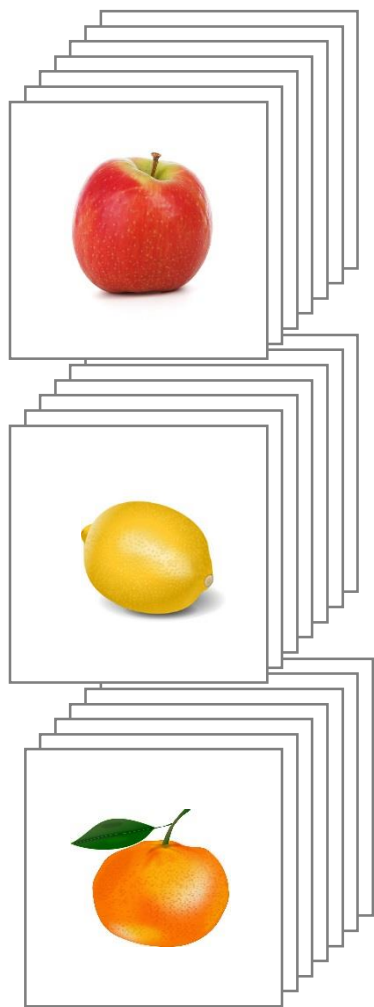
照合

クラス



# 学習

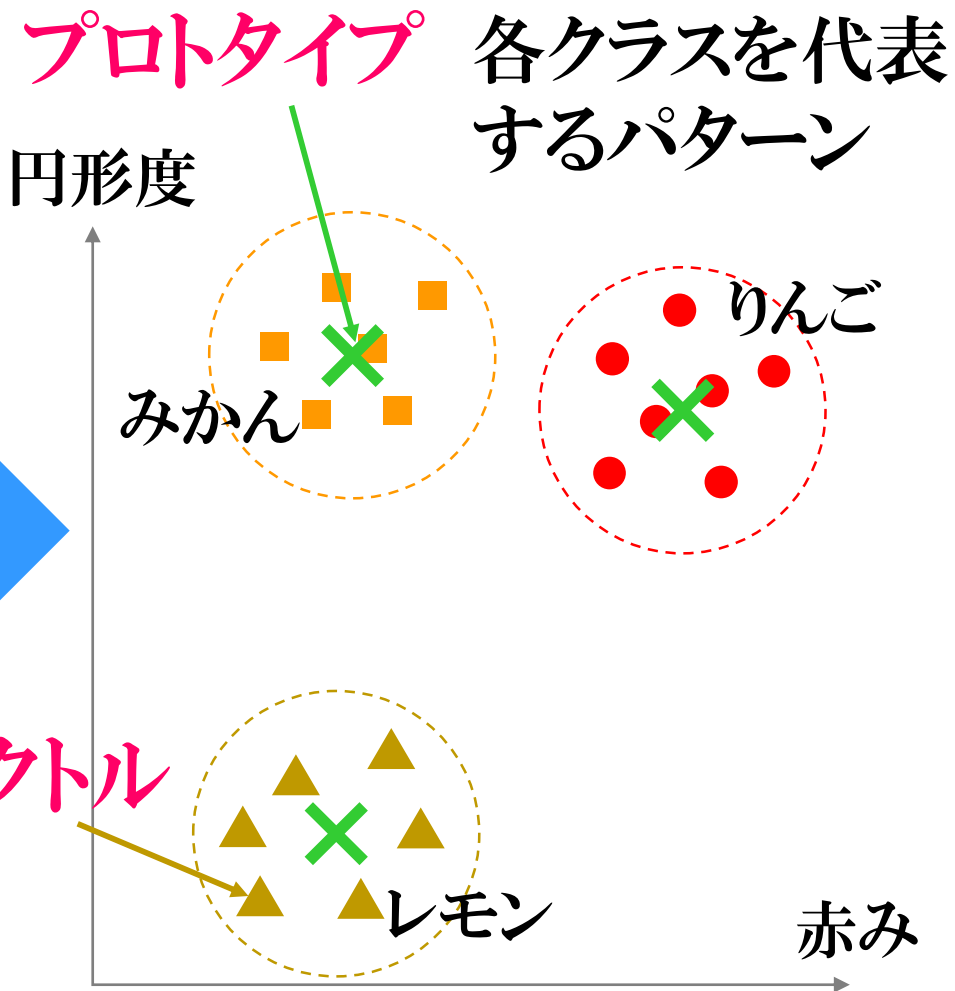
重要



学習用画像

特徴抽出

特徴ベクトル



特徴空間

# 識別

重要



未知画像

特徴抽出

円形度

プロトタイプ

みかん

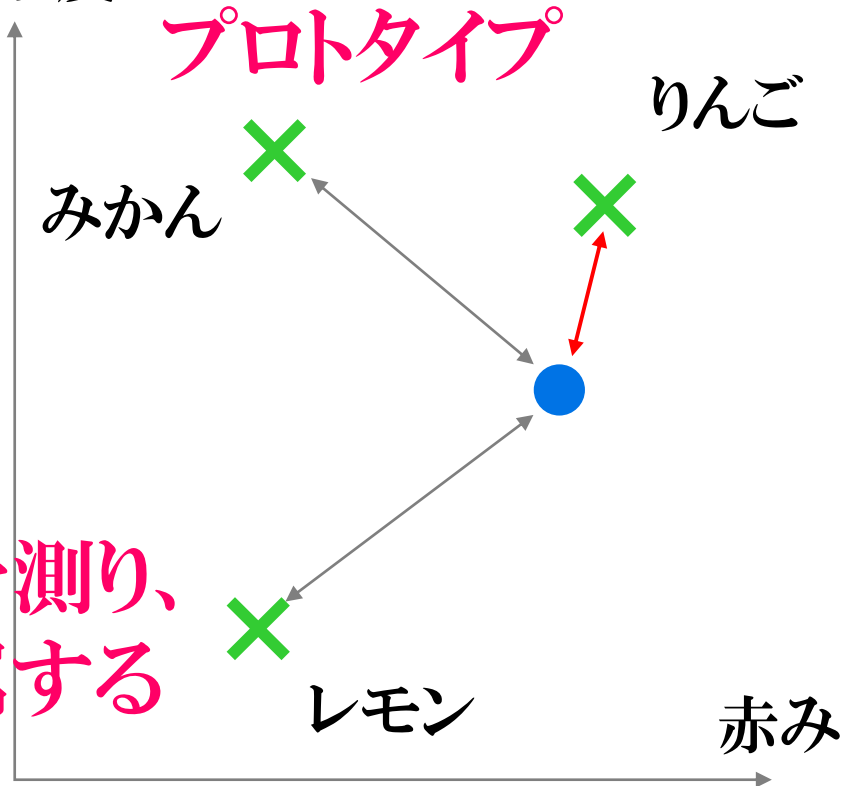
りんご

各プロトタイプとの距離を測り、  
最も近いプロトタイプが属する  
クラスに識別する。

レモン

赤み

特徴空間





# 画像認識の手法

---

## ❖ テンプレートマッチング

テンプレート画像に類似している部分を抽出する。  
特徴空間の軸は、各画素の画素値

## ❖ プロトタイプによるパターン認識

プロトタイプとの距離を測り、未知画像を識別する。  
特徴空間の軸は、形状特徴量など

## ❖ モデル式マッチング

対象物の数式(モデル式)で表し、モデル式のパラメータを画像から決めることで、対象物を検出する。  
特徴空間の軸は、モデル式のパラメータ

# 動画像処理

---

## オプティカルフロー (optical flow)

時間連続した2フレーム間の見かけの動きの分布

✧ ブロックマッチング法

✧ 勾配法 (gradient method)

仮定:  $f(x, y, t) = f(x + \Delta x, y + \Delta y, t + \Delta t)$

$$\frac{\partial f(x, y, t)}{\partial x} \frac{dx}{dt} + \frac{\partial f(x, y, t)}{\partial y} \frac{dy}{dt} + \frac{\partial f(x, y, t)}{\partial t} = 0$$

を解いて、 $\left(\frac{dx}{dt}, \frac{dy}{dt}\right)$ を求める。

# 移動物体検出

---

## ✧ 背景差分法

移動物体が存在していない背景だけの画像との差分により、移動物体を検出する。

## ✧ フレーム間差分法

3枚のフレーム画像の差分の論理積から、移動物体を検出する。

# 3次元画像処理

---

## 投影変換

- ✧ 平行投影 (orthogonal projection)
- ✧ 透視投影 (perspective projection)

3次元空間座標  $(X, Y, Z)$

2次元平面座標  $(x, y)$

焦点距離  $f$

$$x = f \cdot \frac{X}{Z}, \quad y = f \cdot \frac{Y}{Z}$$

# 3次元再構成

---

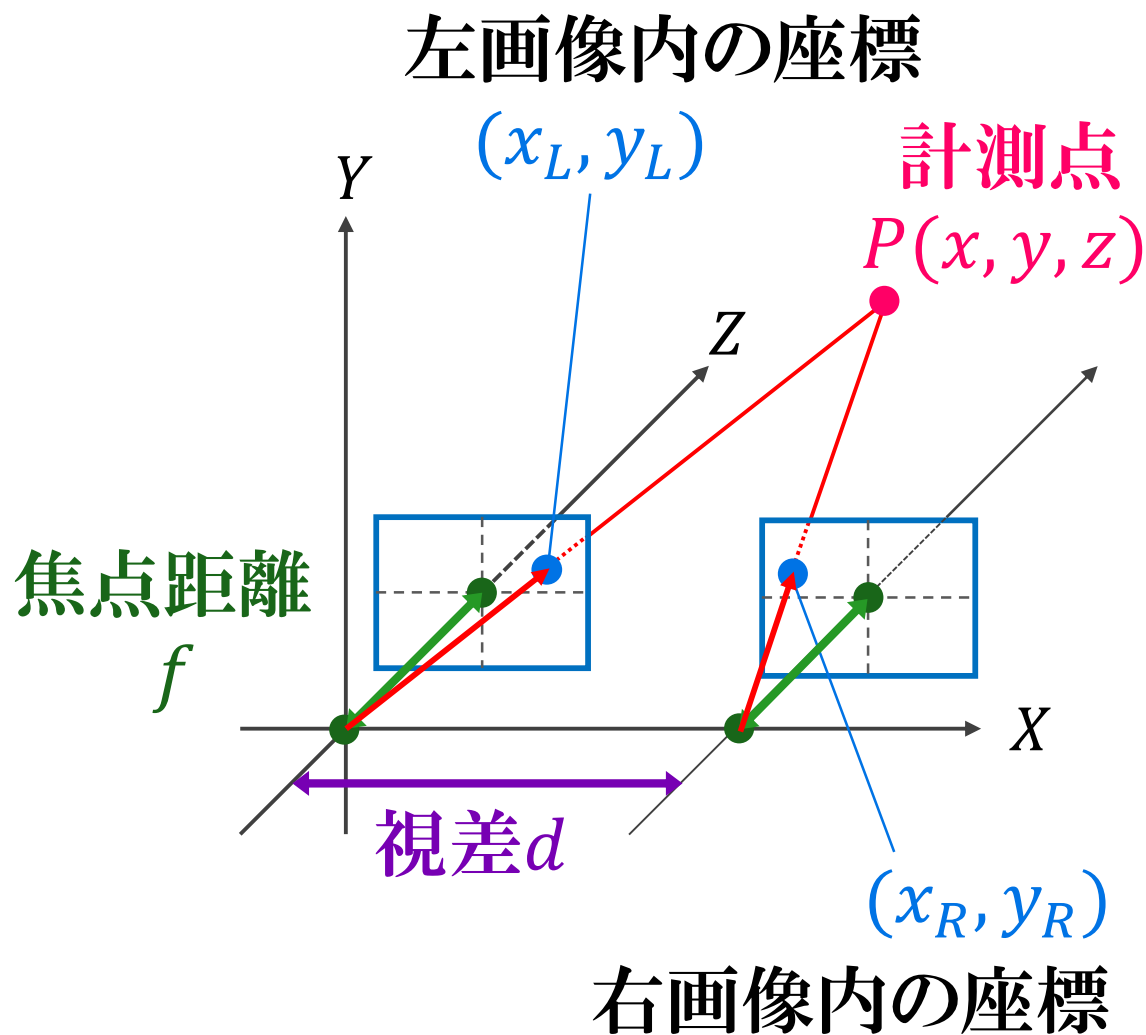
## ✧ ステレオ視 (stereo vision)

視点の異なる2枚の画像の対応点を見つけ  
(ステレオマッチング)、三角測量を用いて3次元座標を計測する方法。

## ✧ 能動的ステレオ視 (active stereo vision)

ステレオ視において、片方のカメラをレーザ光源などに変えて、ステレオマッチングの精度を向上させた方法。

# ステレオ視



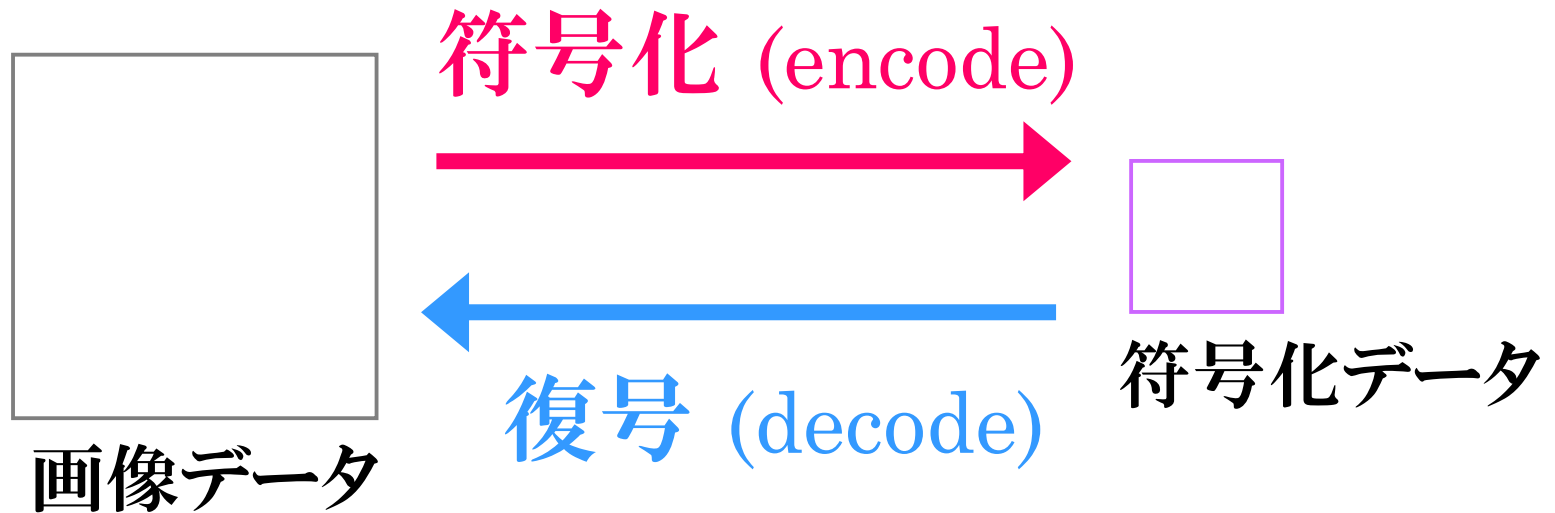
$$x = \frac{x_L d}{x_L - x_R}$$

$$y = \frac{y_L d}{x_L - x_R}$$

$$z = \frac{f d}{x_L - x_R}$$

# 画像符号化

重要



- **可逆符号化 (lossless coding)**  
復号すると、元のデータと同じになる。
- **非可逆符号化 (lossy coding)**  
復号すると、元のデータと厳密には同じにならない。

# 可変長符号

重要

画素値	固定長符号		可変長符号		
	符号語	符号長	出現確率	符号語	符号長
0	00	2 bit	1/2	0	1 bit
1	01	2 bit	1/4	10	2 bit
2	10	2 bit	1/8	110	3 bit
3	11	2 bit	1/8	111	3 bit

平均符号長

2 bit

1.75 bit



# 可逆符号化

---

## ❖ エントロピー符号化 (entropy coding)

出現確率が大きい画素ほど、短い符号語を与える。

### ● ハフマン符号 (Huffman coding)

## ❖ 予測符号化 (predictive coding)

隣接する画素値の差分をとることで、データのエントロピーを小さくしてからエントロピー符号化をする。

## ❖ ランレングス符号化 (run-length coding)

2値画像において、同じ画素値が連続する長さを符号化する。

# フーリエ変換

---

## ✧ フーリエ変換

$$F(u) = \int_{-\infty}^{\infty} f(x) \exp(-j2\pi ux) dx$$

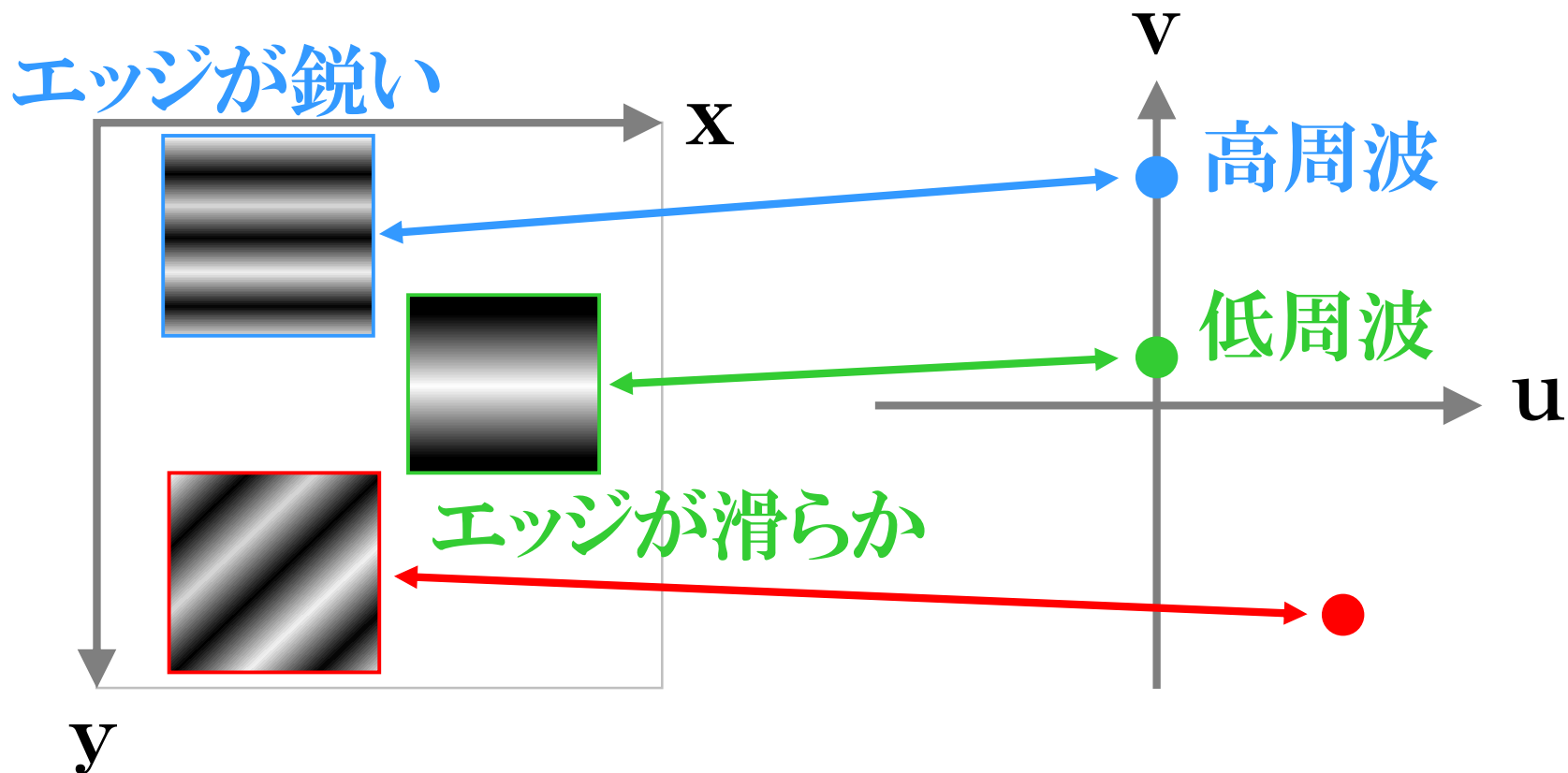
## ✧ 2次元フーリエ変換

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \exp(-j2\pi(ux + vy)) dx dy$$

## ✧ 2次元離散フーリエ変換

$$F(u, v) = \sum_{y=0}^{H-1} \sum_{x=0}^{W-1} f(x, y) \exp\left(-j2\pi\left(\frac{xu}{W} + \frac{yv}{H}\right)\right)$$

# 画像の周波数領域



空間領域

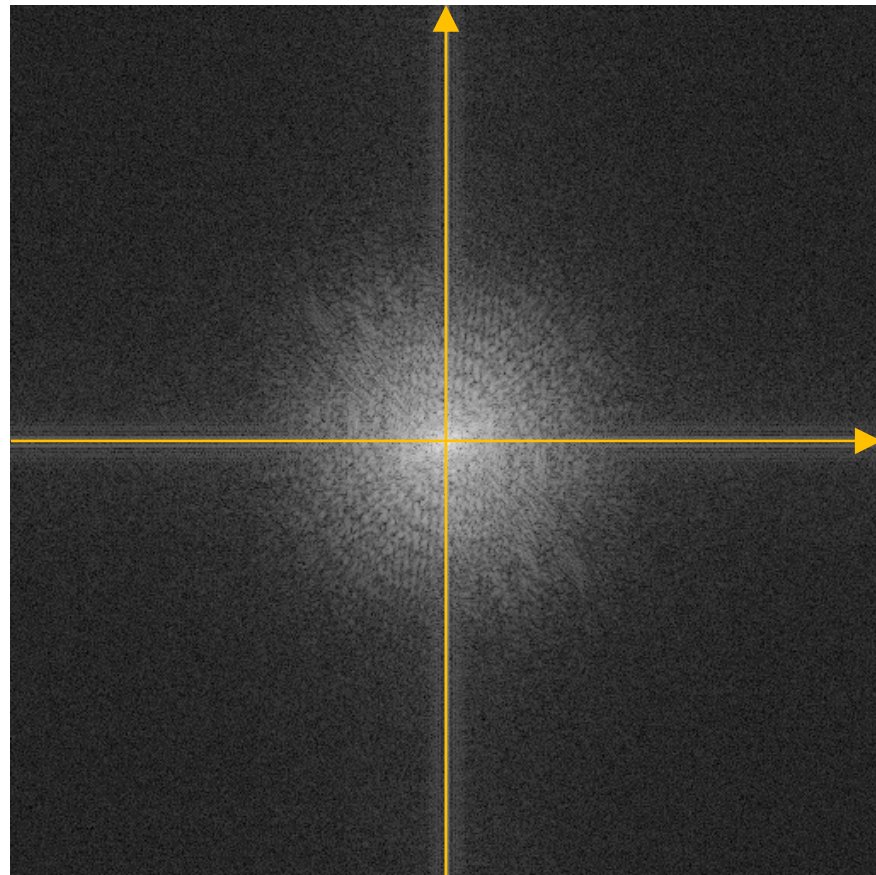
周波数領域

# 高周波成分が少ない画像

---



原画像

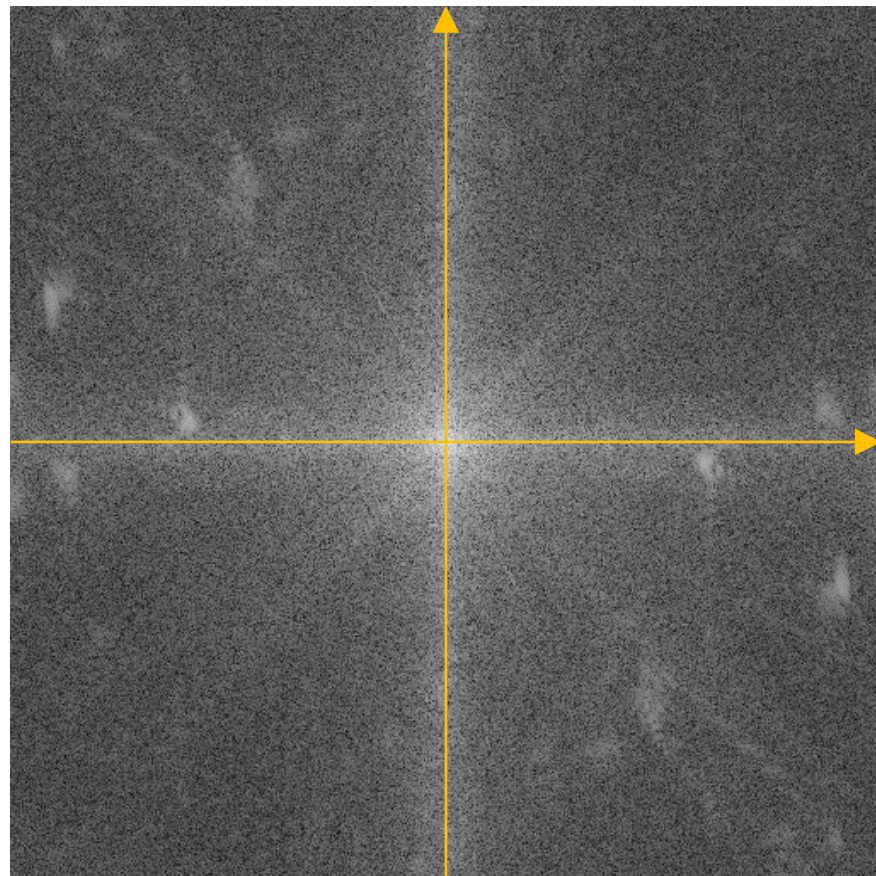


周波数領域画像

# 高周波成分が多い画像



原画像



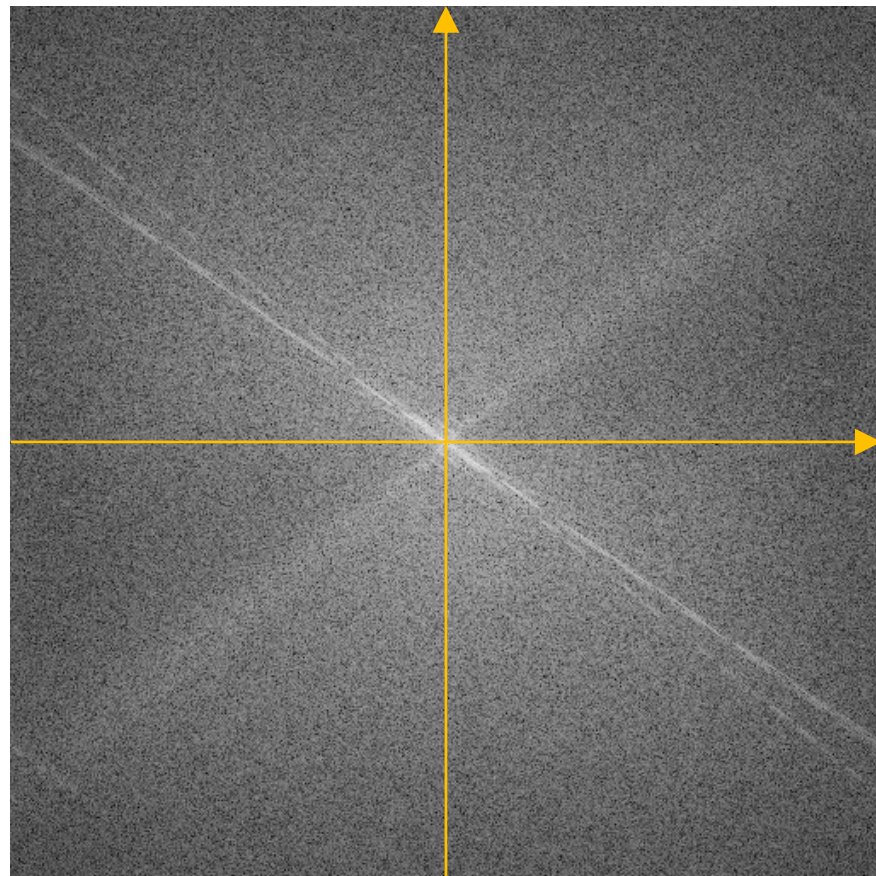
周波数領域画像



# 特定の高周波成分が多い画像



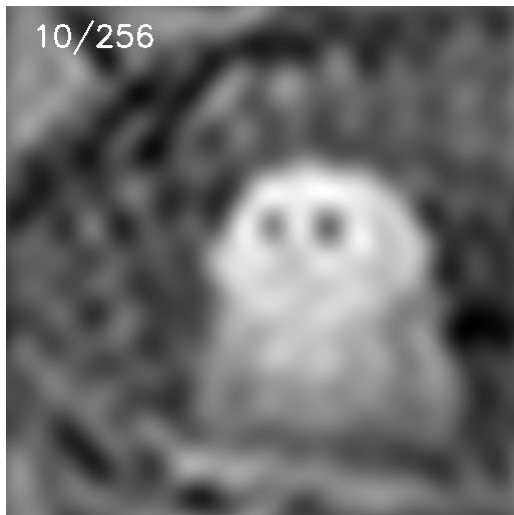
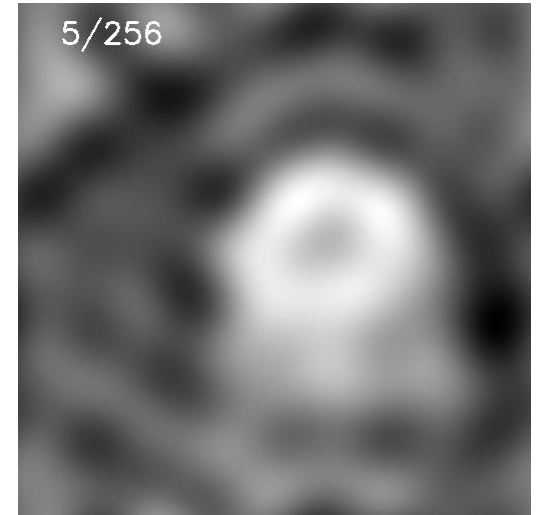
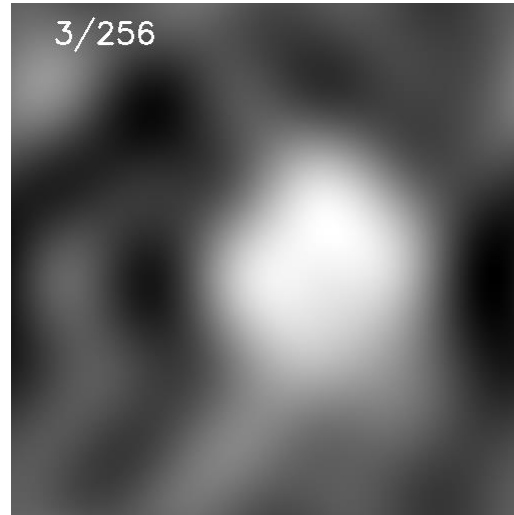
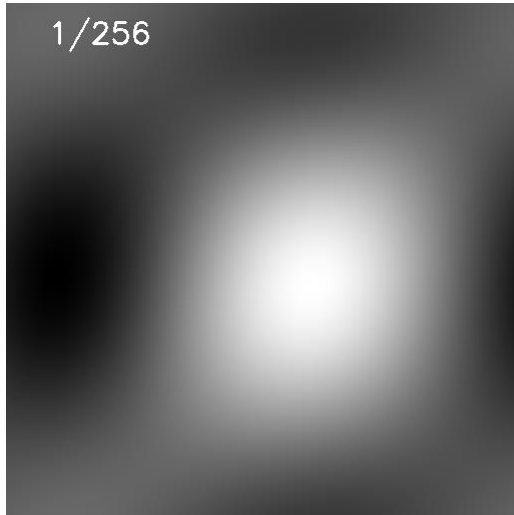
原画像



周波数領域画像

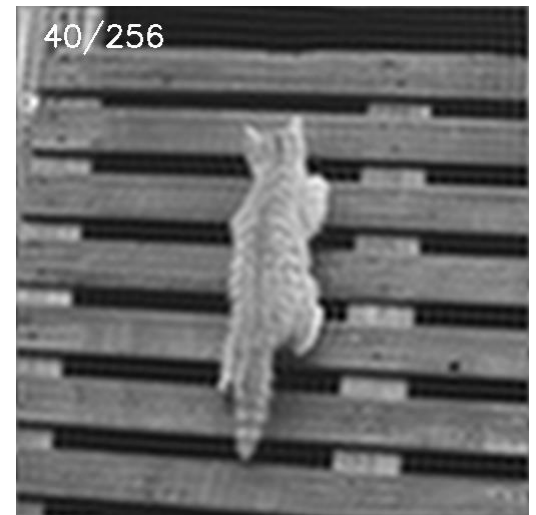
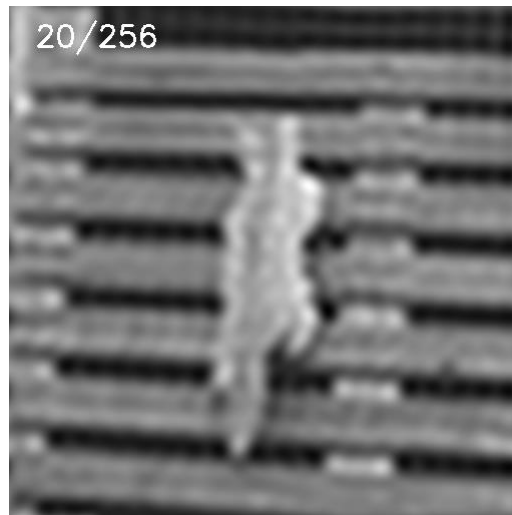
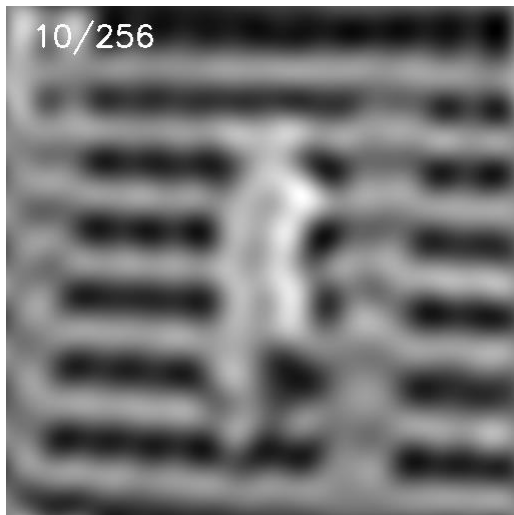
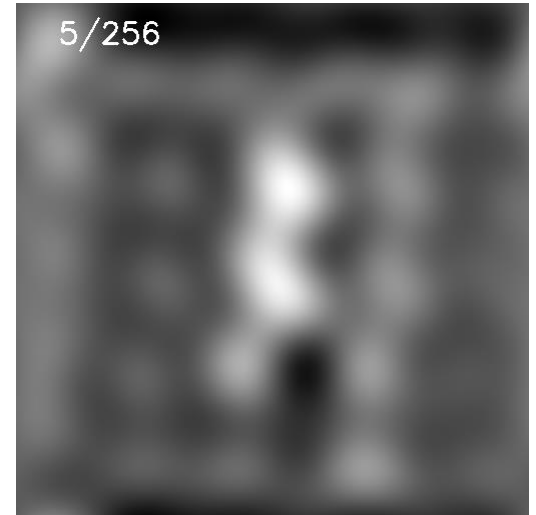
# 波の合成による画像生成

低周波数の波から順に積算した結果



# 波の合成による画像生成

低周波数の波から順に積算した結果

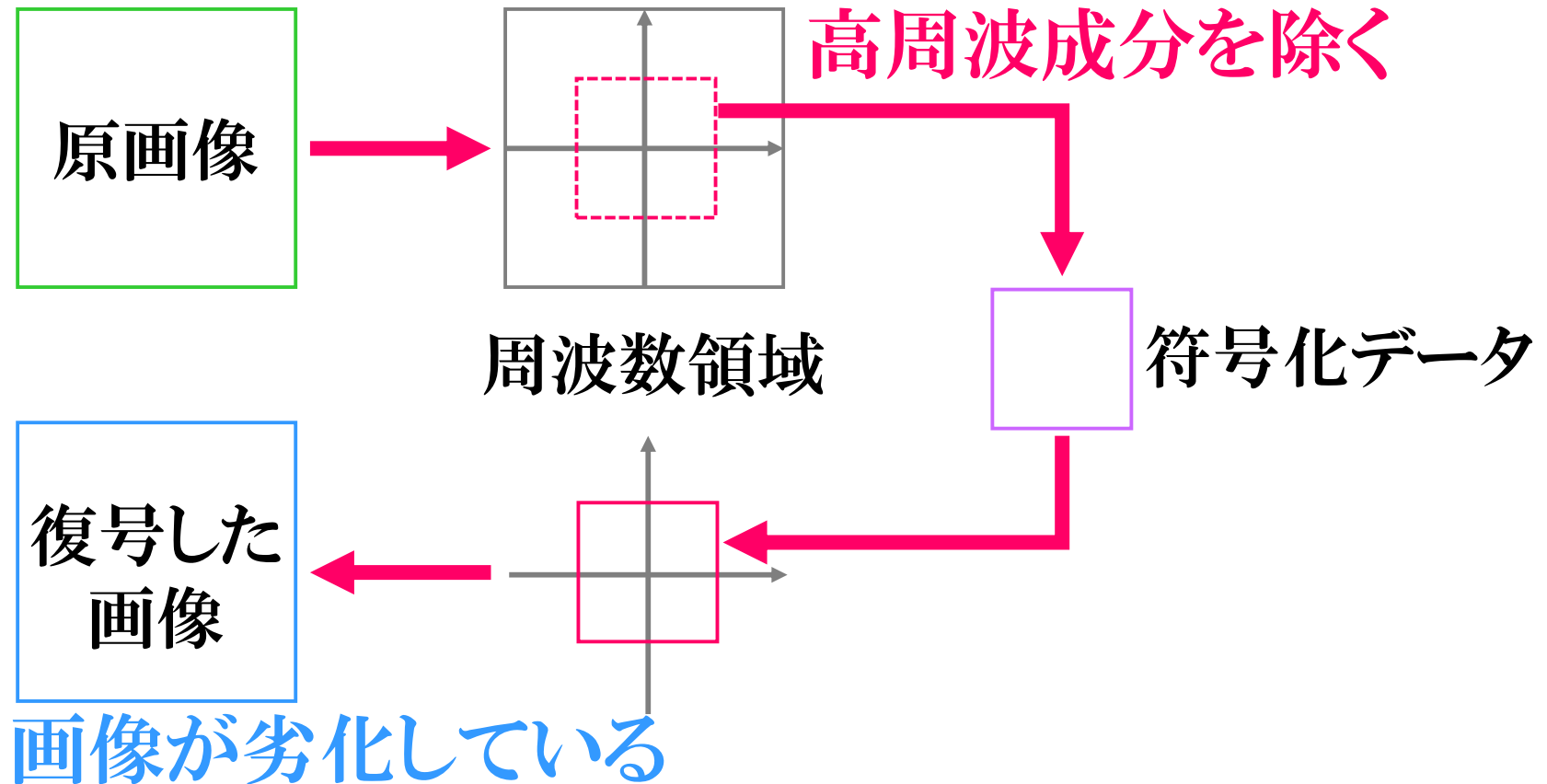




# 変換符号化

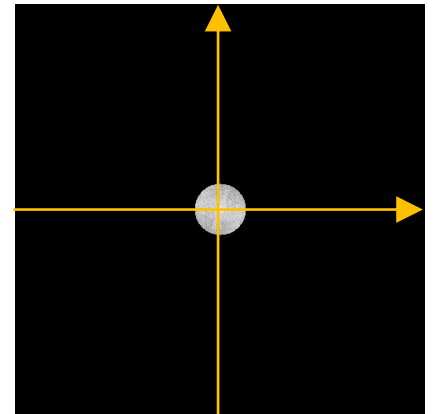
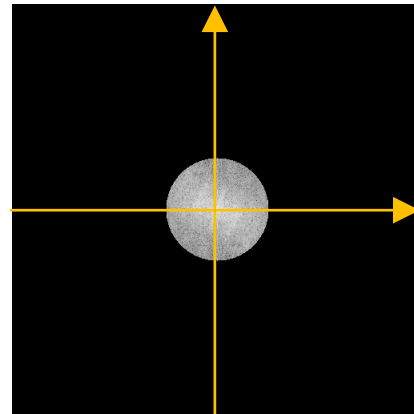
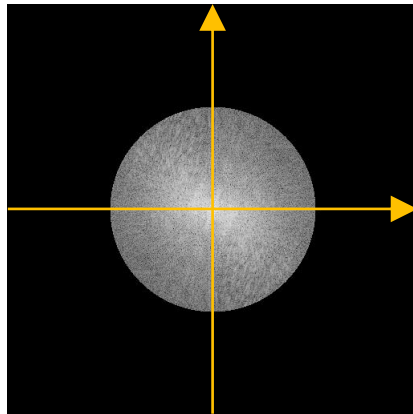
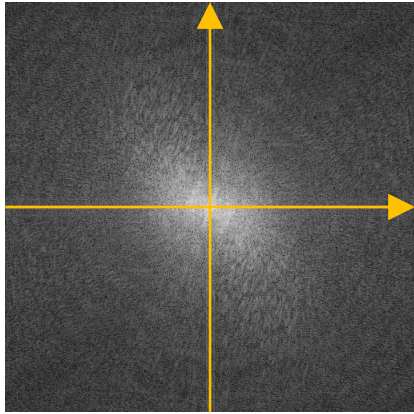
重要

画像の周波数成分から高周波成分を削除して、画像のデータ量を減らす。



# 高周波成分の除去と画像の劣化

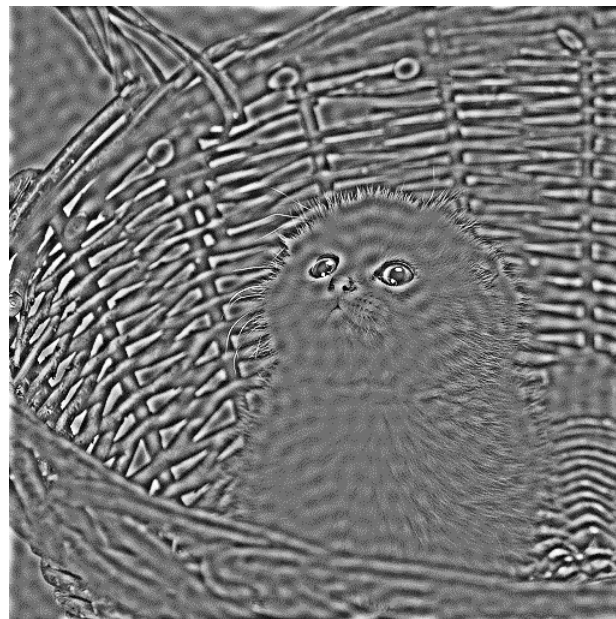
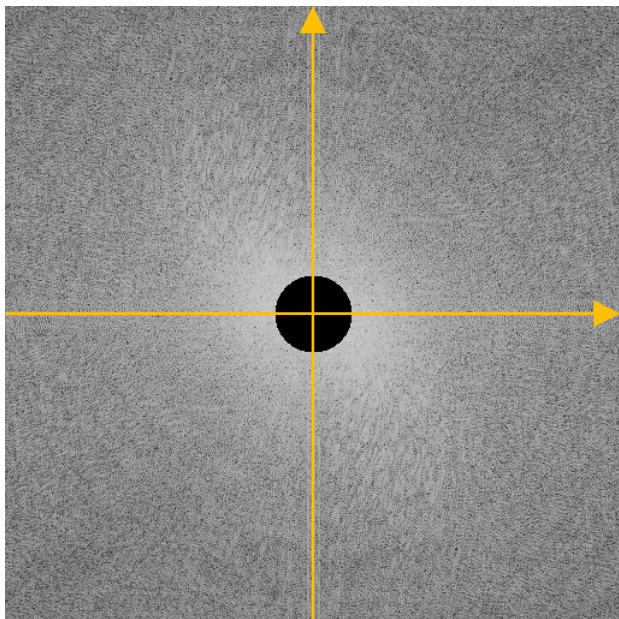
データ量 少ない



画像劣化 大きい

# 低周波成分の除去

---



高周波成分を除去 ➡ 画像のエッジがぼやける

低周波成分を除去 ➡ 画像のエッジが抽出される

# コンピュータグラフィックス

---

## ✧ モデリング

3次元空間における物体の形状や材質などを数値的に記述する。

## ✧ レンダリング

3次元物体のモデルを、光源や視点に基づいて、2次元画像に描画する。

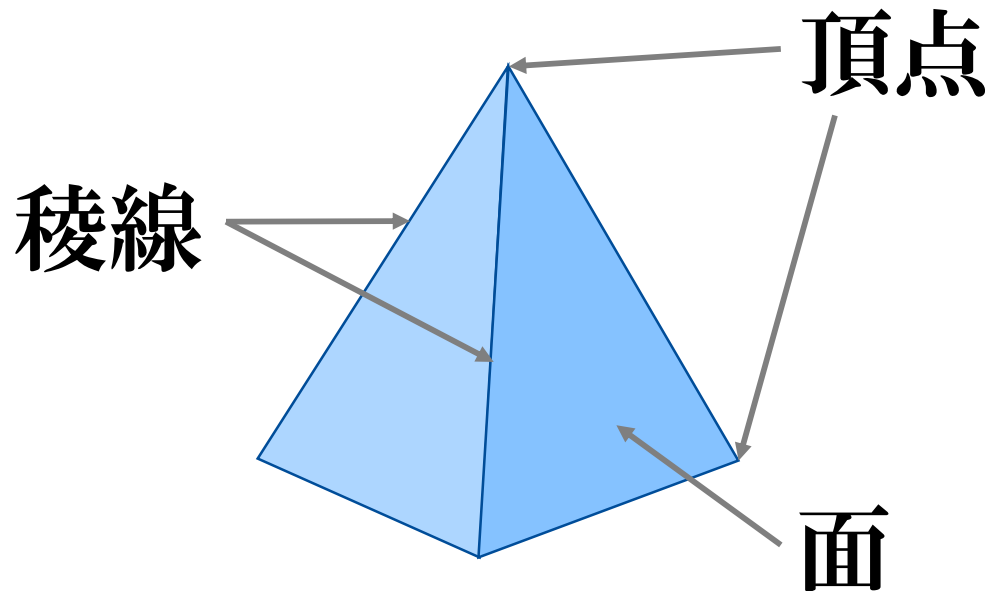
# 形状モデル

---

✧ ワイヤーフレームモデル

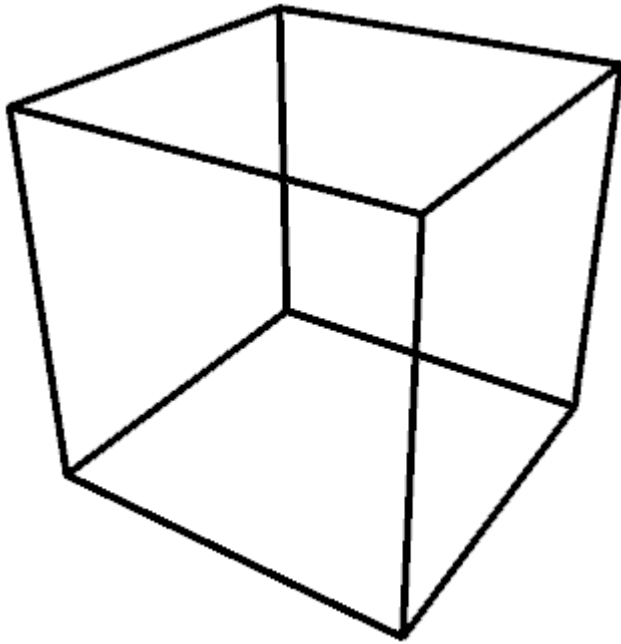
✧ サーフェスモデル

✧ ソリッドモデル

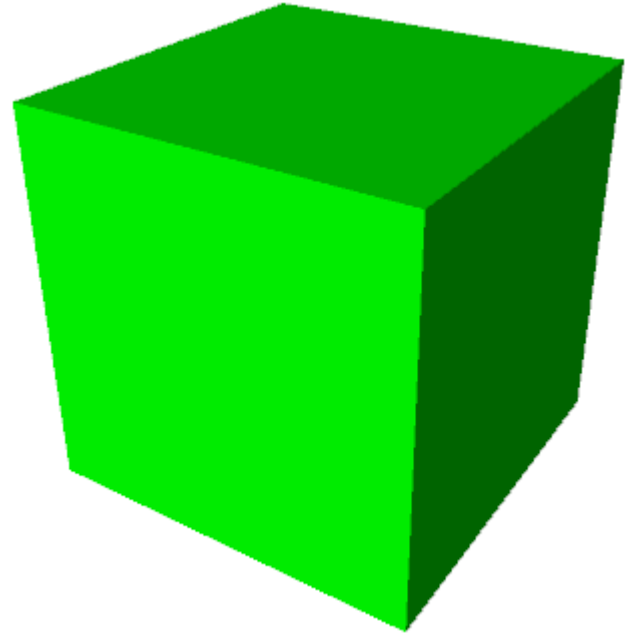


# 立方体

---



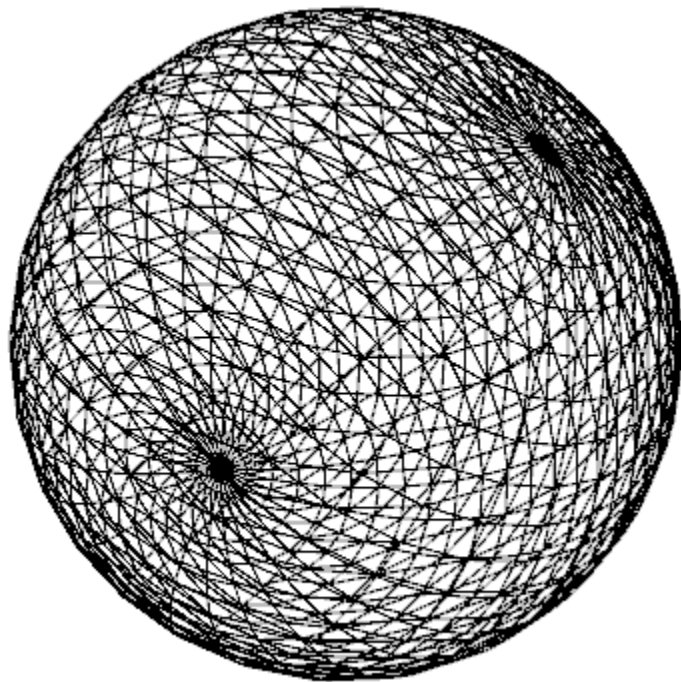
ワイヤーフレームモデル



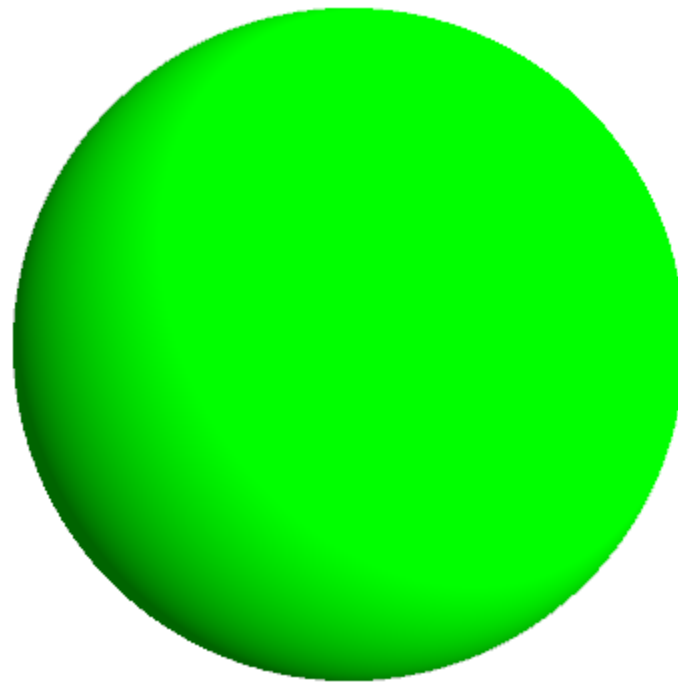
サーフェスモデル

# 球体

---



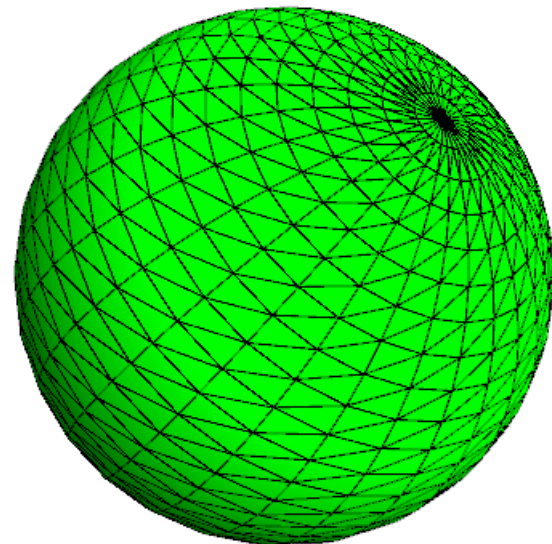
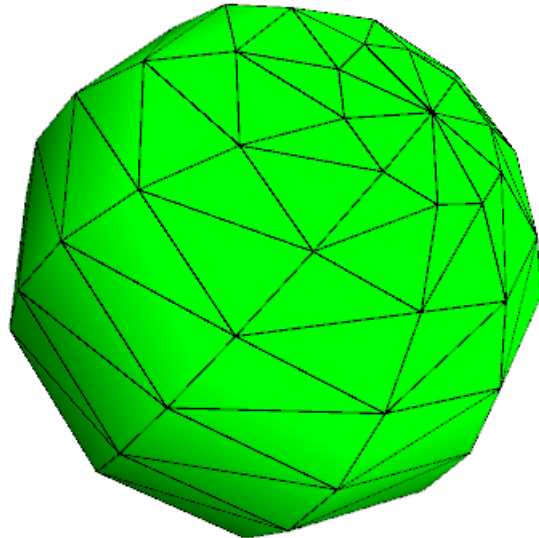
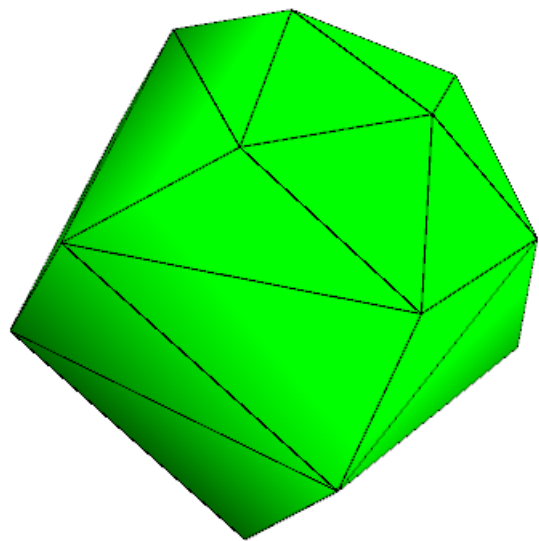
ワイヤーフレームモデル



サーフェスモデル



# ポリゴンによる曲面の再現



ポリゴン数  
少ない



ポリゴン数  
多い

計算量  
少ない



計算量  
多い

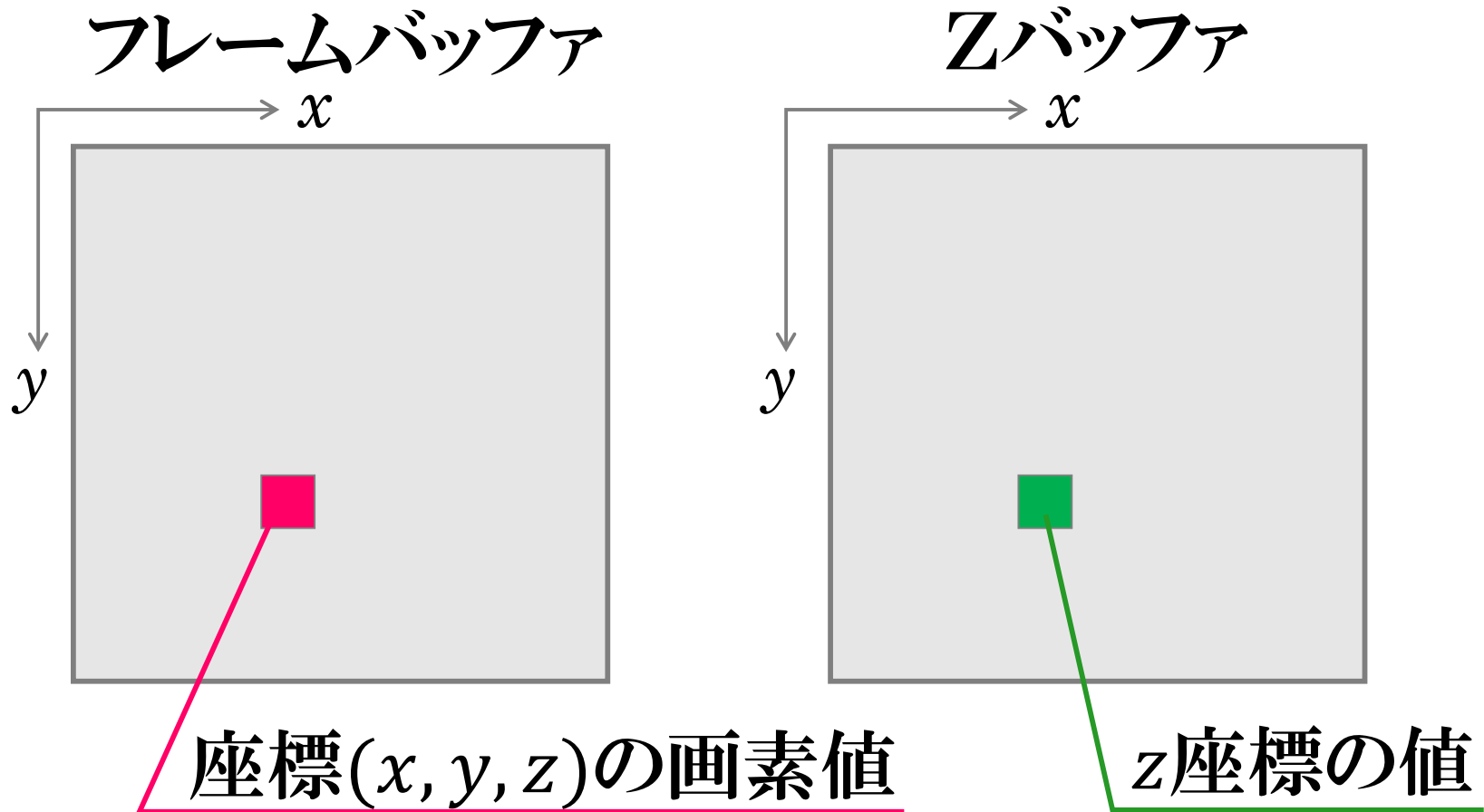


# レンダリング

---

- ① 透視投影
- ② 陰面消去
- ③ シェーディング
- ④ テクスチャマッピング

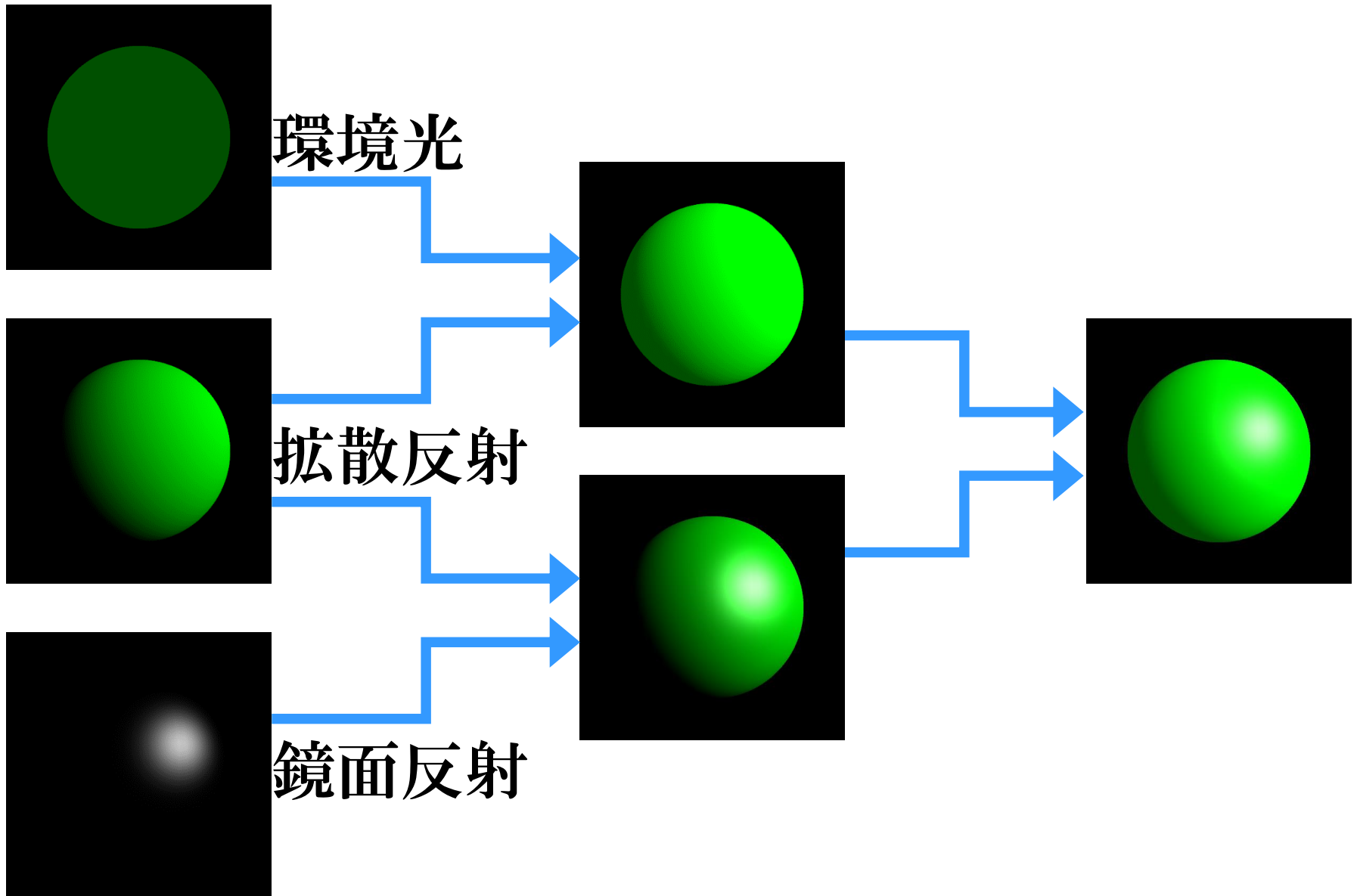
# Zバッファ法



ポリゴンを描画するとき、 $z$ 座標を比較して小さければ(手前に位置すれば)上書きする。

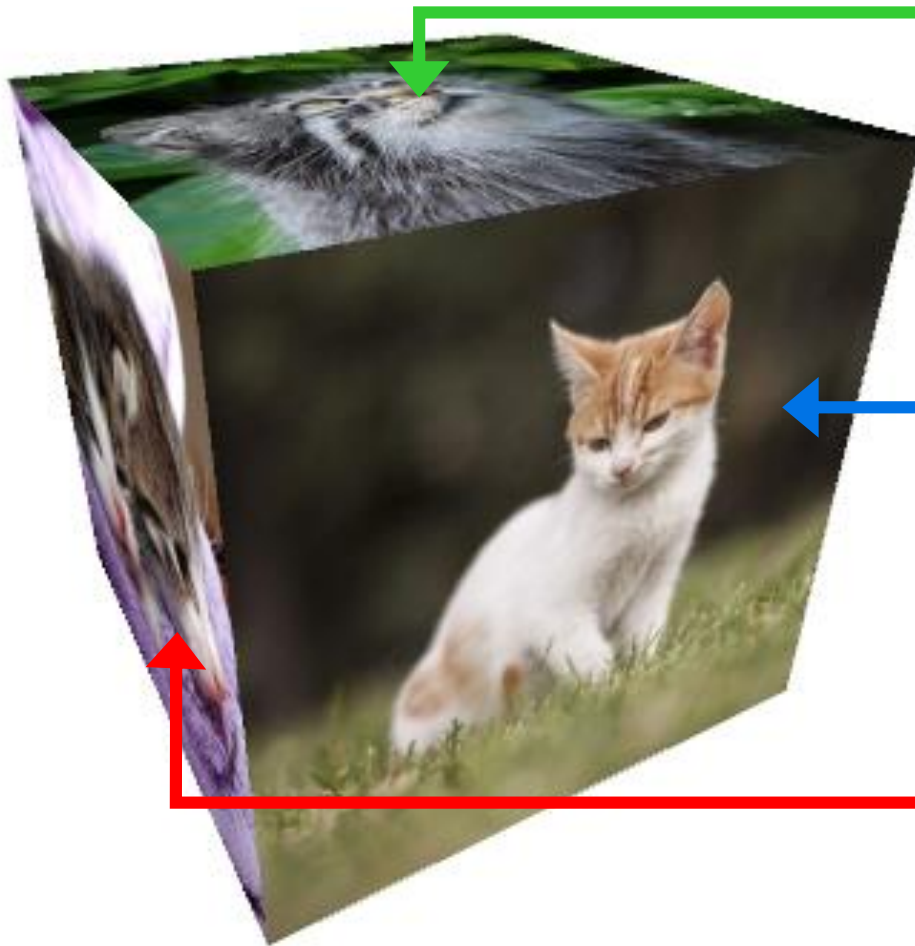
# シェーディング

---



# テクスチャマッピング

射影変換



# 画像処理とCGの応用

---

## ✧ バーチャルリアリティ (VR: Virtual Reality)

コンピュータ上に作られた仮想的な空間をCGなどを用いて疑似体験させる技術

## ✧ 拡張現実感 (AR: Augmented Reality)

風景の画像上にCGを合成することで、情報を視覚的に補う技術

## ✧ モーションキャプチャ (motion capture)

人の動作をリアルタイムに計測する技術

# VR、AR、MR

---

仮想空間

現実空間



バーチャル  
リアリティ  
(VR)



拡張現実  
(AR)

**複合現実 (MR: Mixed Reality)**

仮想空間の物体を、現実空間の物体と同じように  
操作できる。

同じ仮想空間を複数の人が共有して体験できる。

# ディープラーニング(深層学習)

---

多層ニューラルネットワークを用いて、大量の学習用画像から有効な特徴量を自動的に選び出して学習を行う手法

## ニューラルネットワーク

脳の神経回路をモデル化したコンピュータアルゴリズム

## 近年注目されている理由

- コンピュータ性能の進歩により、処理時間が実用レベルになった。
- 大量の学習画像さえ集められれば、画像処理のプロフェッショナルでなくても学習させることができる。

# 定期試験

---

試験日: 8月6日(月)

出題形式: 論述式    教科書・ノート持ち込み可

範囲:

- 画像変換（AD変換、ヒストグラム、空間フィルタ、幾何学変換、再標本化など）
- 2値画像処理（2値化、連結性、特徴パラメータなど）
- 画像認識（テンプレートマッチング、パターン認識、学習と識別など）
- 画像符号化（ハフマン符号化、変換符号化など）