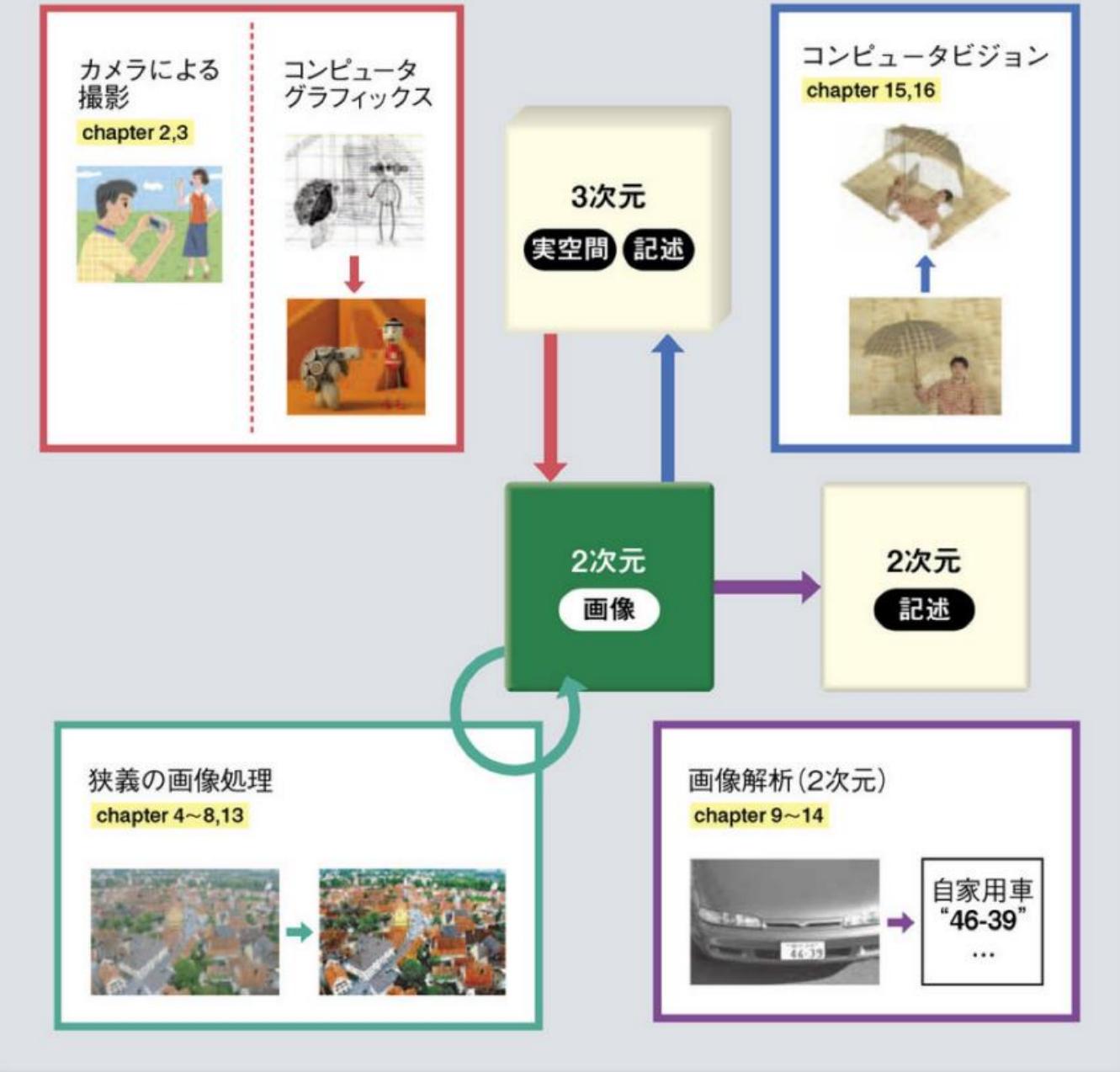


1. イントロダクション

入力	出力
三次元世界	画像



濃淡変換、空間フィルタリング、周波数領域フィルタリング、劣化画像の復元、幾何学変換

入力	出力
画像	三次元世界

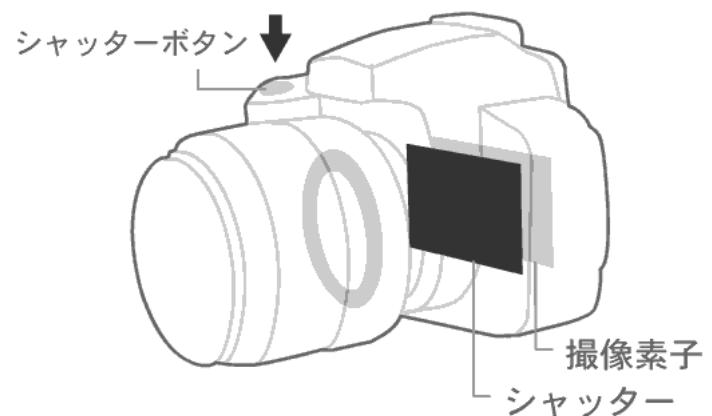
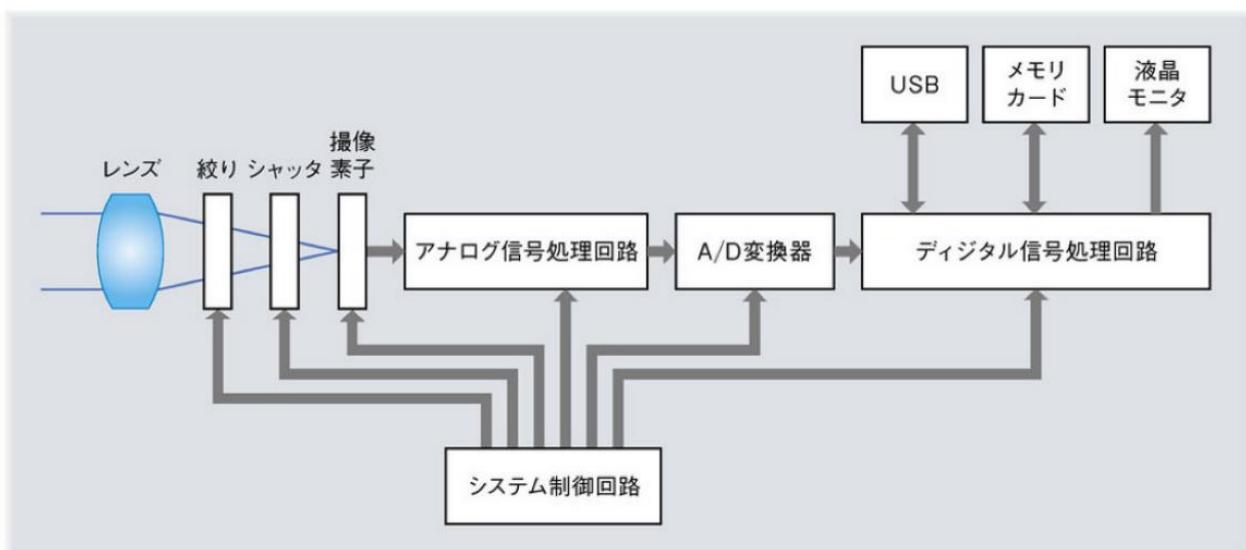
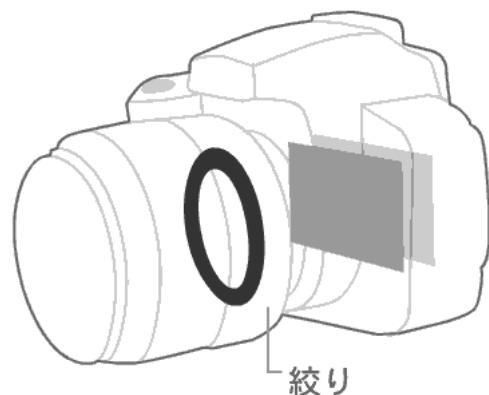
入力	出力
画像	映っている対象物情報

二値画像処理、領域処理、パターン検出・認識、画像認識、動画像処理

2. ディジタル画像の撮影

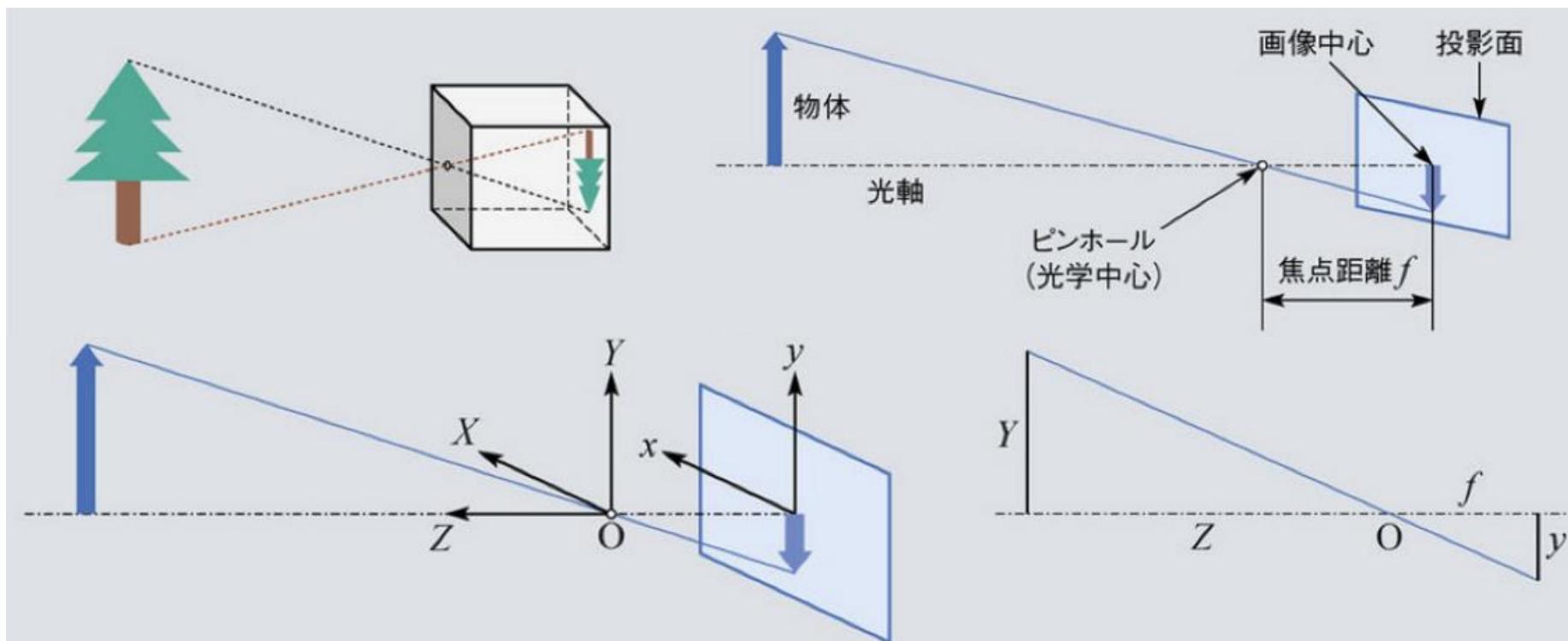
デジタルカメラの構成

- レンズ…対象からの光を撮像素子上に集光。
- 絞り…開き具合が可変で、光の通過量を調整。
- シャッタ…通常閉じていて、撮像素子に光が当たらないよう遮光。
撮影時ボタンを押して開く。
- 撮像素子…入射した光の量に応じた電圧を出力。
フィルムカメラのフィルムに対応する。
- A/D変換器…撮像素子からの電圧信号をデジタルデータに変換。



ピンホールカメラモデル

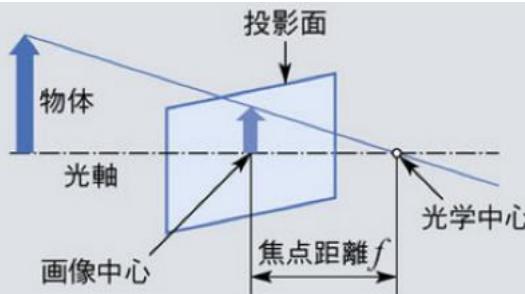
- 小さな穴を開けた暗箱の内側に上下左右反転した像を映す初期のカメラ
- 光学中心=ピンホール位置
- 焦点距離 f =光学中心($Z=0$)から投影面($Z=-f$)までの距離
- メリット:ピント合わせが不要。穴が小さいのでボケることがない。
- デメリット:穴が小さいのでレンズカメラに比べてかなり暗い(感度が低い)



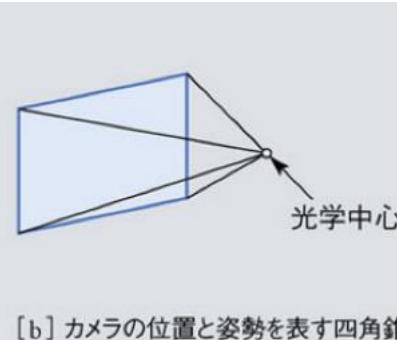
$$x = -f \frac{X}{Z}, \quad y = -f \frac{Y}{Z}$$

透視投影モデル

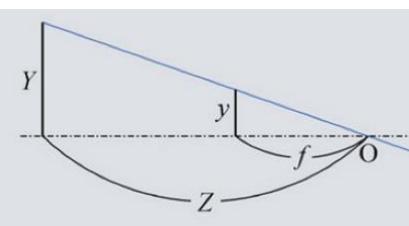
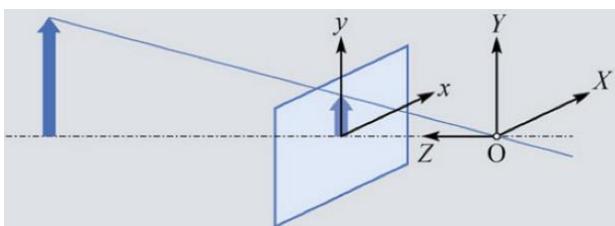
- 三次元の物体を平面上で表現するための図法
- ピンホールカメラモデルでは対象が上下左右反転してわざらわしいため、コンピュータビジョンなどでは透視投影モデルが用いられる。
- 三次元空間における平行な直線は無限遠では一点(消失点)に収束する。



[a] 透視投影モデル

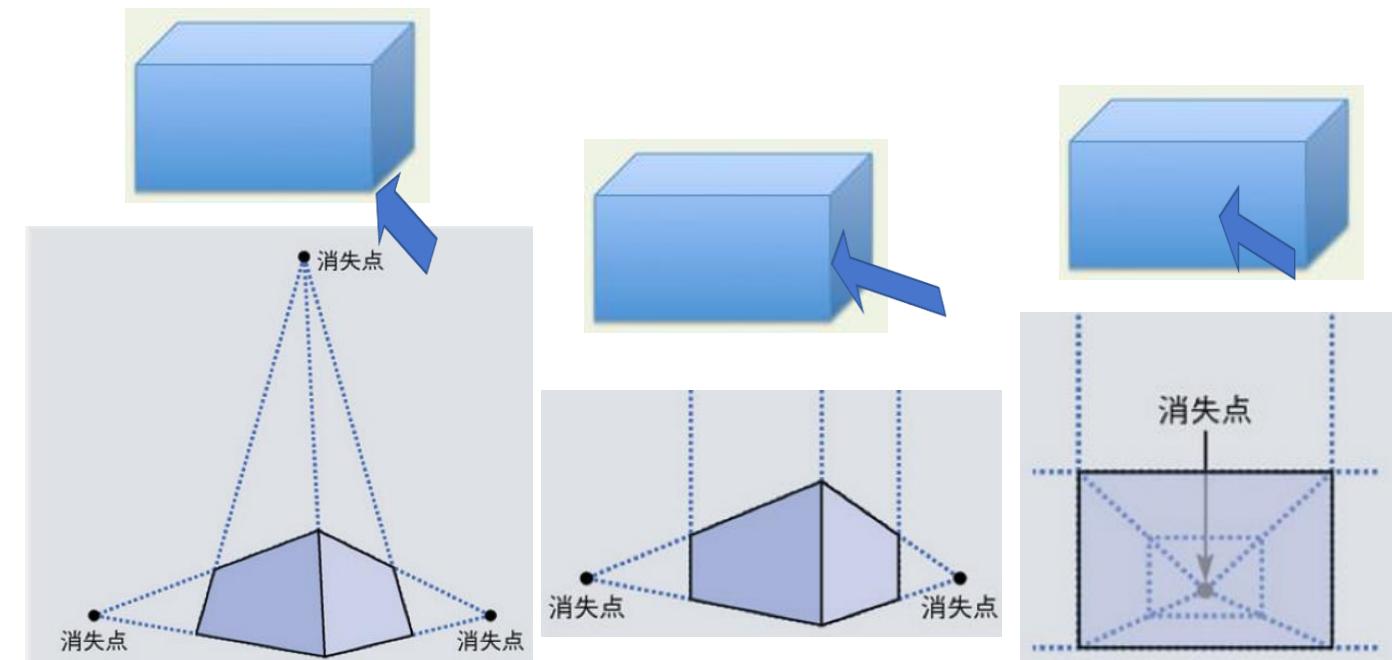


[b] カメラの位置と姿勢を表す四角錐



直線 $t\mathbf{u} + \mathbf{v}$ 上の点(X,Y,Z)の画像上の位置(x,y)

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} tu_x + v_x \\ tu_y + v_y \\ tu_z + v_z \end{pmatrix} \left\{ \begin{array}{l} x = f \frac{tu_x + v_x}{tu_z + v_z} = f \frac{u_x + v_x/t}{u_z + v_z/t} \\ y = f \frac{tu_y + v_y}{tu_z + v_z} = f \frac{u_y + v_y/t}{u_z + v_z/t} \end{array} \right.$$



薄肉レンズモデル

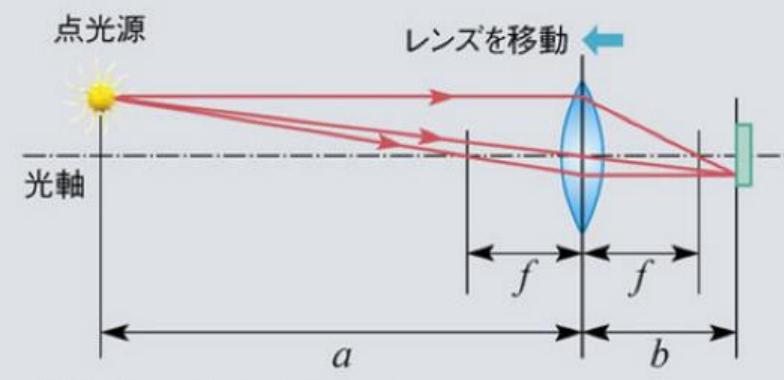
- 焦点距離 f と開口径で特性が表現できる。
- 光学中心=レンズの中心(主点)

ガウスのレンズ公式

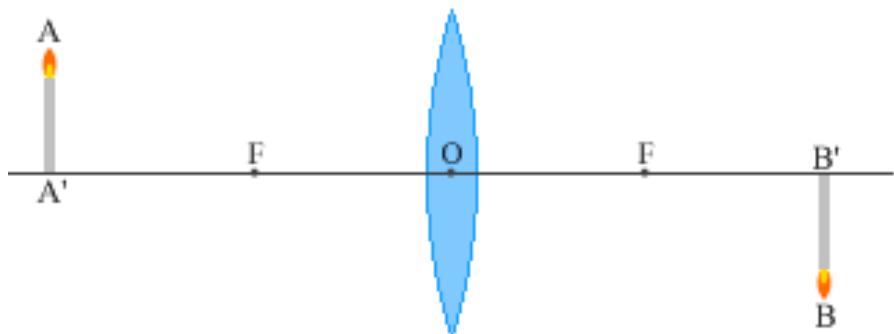
$$\frac{1}{a} + \frac{1}{b} = \frac{1}{f}$$

- 焦点距離 f =無限遠にある点光源からくる光(全て光軸に平行)が結像する距離

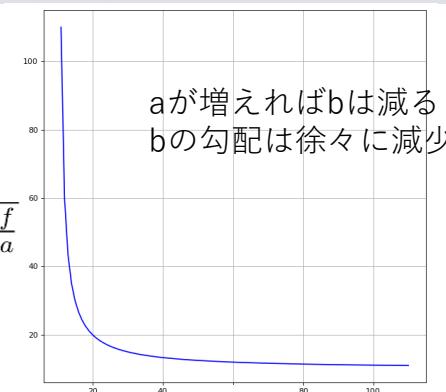
ガウスのレンズの公式を変形すると $b = \frac{f}{1 - \frac{f}{a}}$ → a が小→ f/a が大→分母が小→ b が大となり、 a が大きいと b は小さい。
従って、被写体までの距離 a が近い時は、レンズを手前に移動してピントを合わせる。



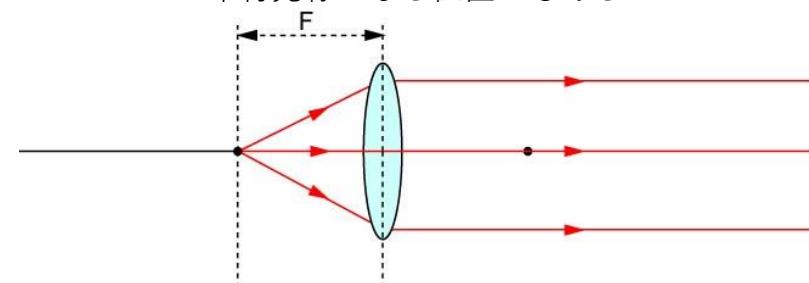
一点からレンズに向かって発せられた光線はすべて、レンズ通過後一点で交わる



$$b = \frac{f}{1 - \frac{f}{a}}$$

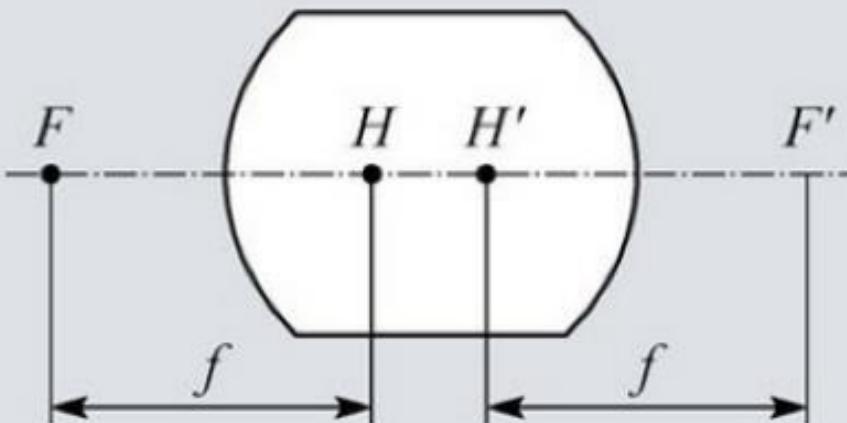


焦点距離は、その位置から発せられた光がレンズ通過後、平行光線になる位置もある

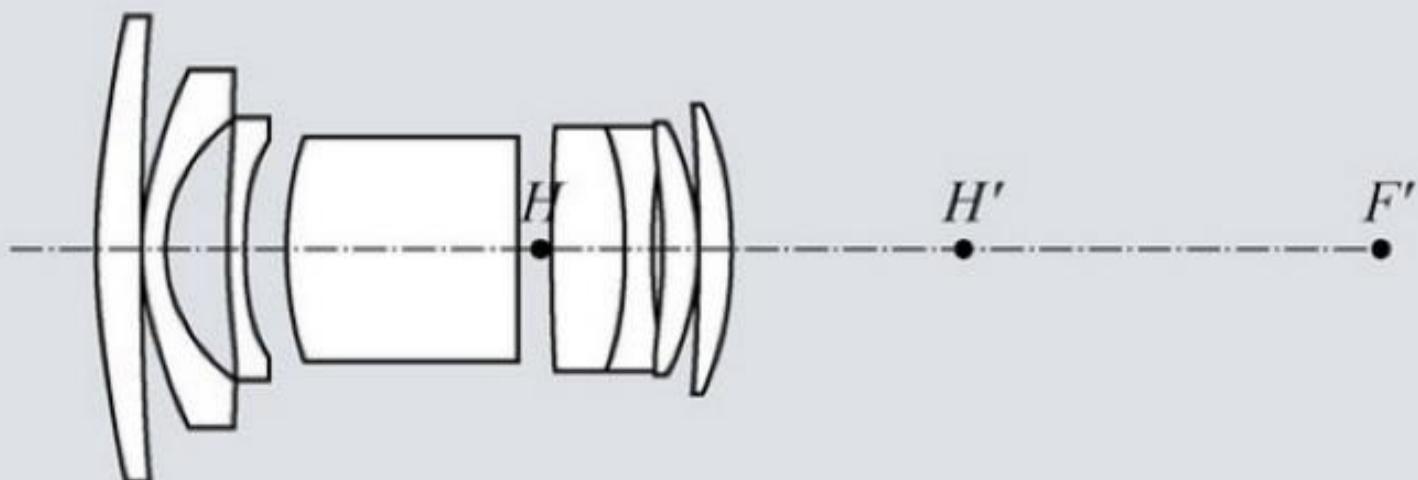


厚肉レンズモデル

- 焦点距離に対して、レンズの厚みを無視できないレンズ
- 実際のレンズは、複数枚のレンズが組み合わされているため、厚肉レンズモデルに近似できる。
- 主点を中心に回転させれば、カメラ位置を固定して回転させたことになる。
- 焦点距離=後ろ側主点 H' から後ろ側焦点 F' までの距離



[a] 厚肉レンズの例



[b] 写真用レンズの構成例

(『光機器の光学I』提供: (社) 日本オプトメカトロニクス協会)

歪曲収差

- ・ 収差…望遠鏡や写真機等のレンズ類による光学系において、被写体から像への変換の際、幾何的に理想的には変換されずに発生する色づきやボケやゆがみ。色収差と単色収差がある。
- ・ ザイデル収差…単色収差のうち、発生が著しい5種
- ・ 歪曲収差…ザイデル収差の一つ。
- ・ 半径方向…レンズの歪みにより、主点への入射角と出射角が異なるために生じる。
- ・ 接線方向…複数レンズ間の中心位置ずれや傾きによって生じる。

(x_d, y_d) =歪曲収差のあるレンズでの座標 (x_u, y_u, r) =理想的なレンズでの座標

κ はレンズ歪み係数でコンピュータビジョンのキャリブレーションで求められる。 \rightarrow この係数を用いれば歪曲収差を補正できる。

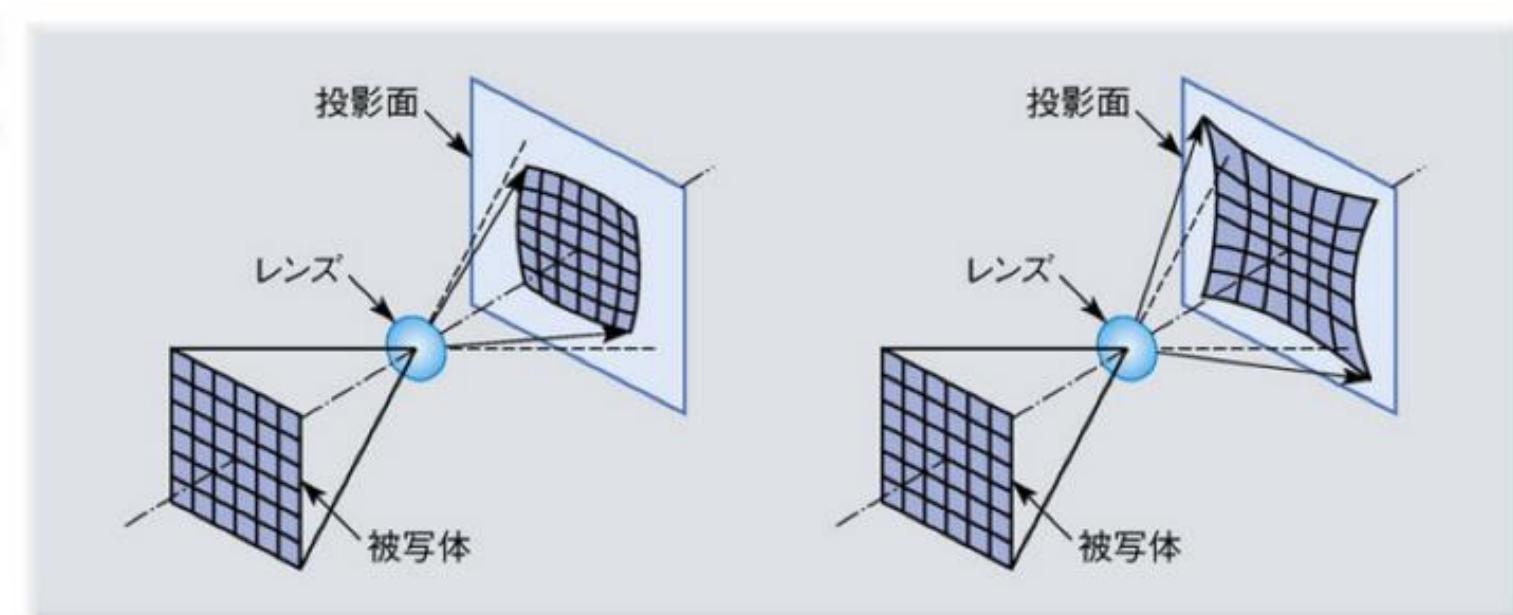
$\kappa_{3,4,5}$ は実使用上無視されることも多い

$$x_d = (1 + \kappa_1 r^2 + \kappa_2 r^4 + \kappa_5 r^6) x_u + 2\kappa_3 x_u y_u + \kappa_4 (r^2 + 2x_u^2)$$

$$y_d = (1 + \kappa_1 r^2 + \kappa_2 r^4 + \kappa_5 r^6) y_u + \kappa_3 (r^2 + 2y_u^2) + 2\kappa_4 x_u y_u$$

$$x_d = (1 + \kappa_1 r^2 + \kappa_2 r^4) x_u$$

$$y_d = (1 + \kappa_1 r^2 + \kappa_2 r^4) y_u$$



周辺光量の低下

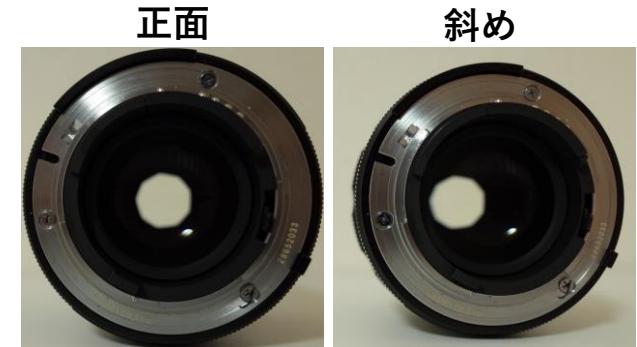


F1.4



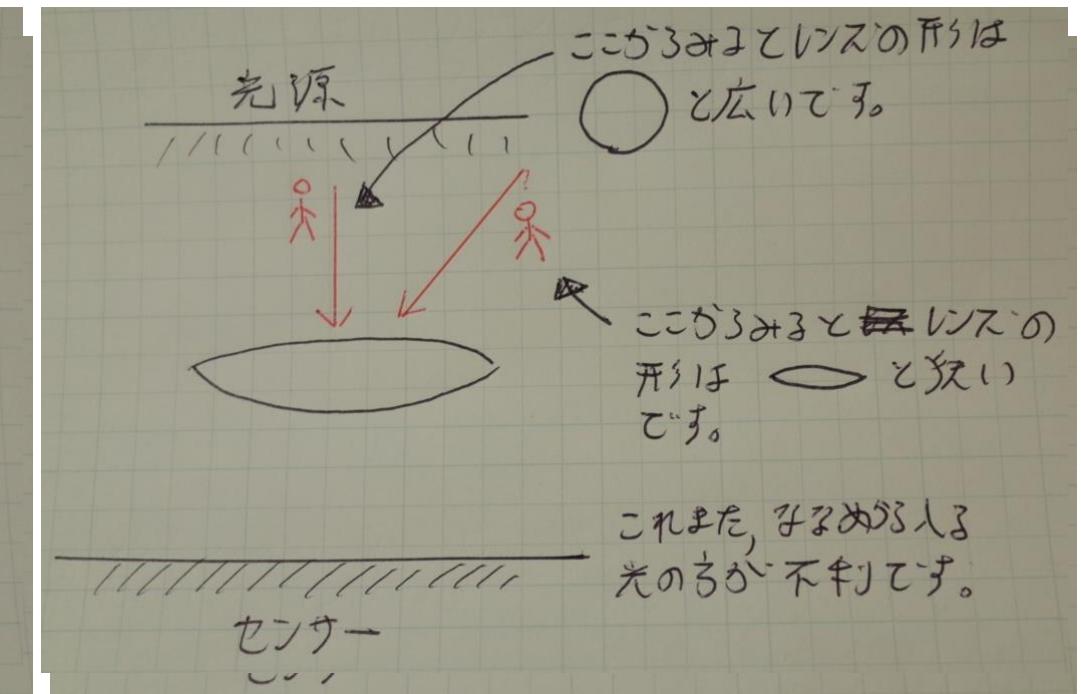
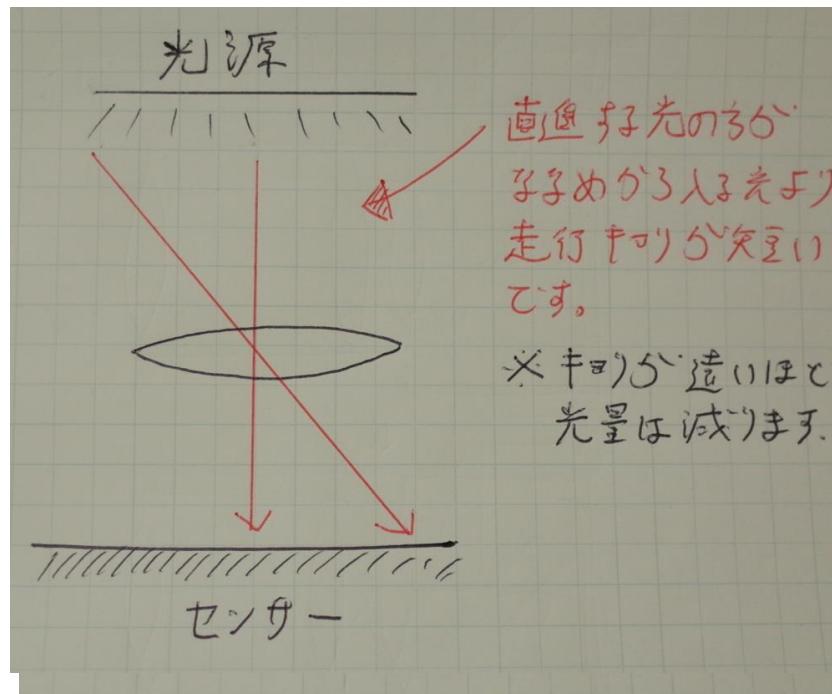
原因1=口径食(ヴィネット)
斜めから入った光が鏡筒に遮断されることで生じる。
絞り値を上げることで改善される。

F5.6



原因2=コサイン四乗則

照度は入射角のコサイン四乗に比例して小さくなる。絞っても効果はない。
入射角度が大きい部分はより多くの光線を入れるなどの工夫によって改善される。



撮影パラメータ

1. 撮影画角

1. 撮像素子サイズ
2. レンズ焦点距離
3. 1.2の組み合わせ

2. 画像の明るさ

1. 絞り
2. シャッタースピード(露光時間)
3. ISO感度

3. 被写界深度(ピント)

4. フレームレート(動画)

撮影画角

- デジカメは、焦点を無限遠の物体に合わせている。すなわち焦点距離に撮像素子を置いている。
- 近くの物体を撮影する際は、レンズを移動させて撮像素子までの距離を変えたり、レンズの一部を動かして焦点距離自体を変えることでピント合わせを行う。
- 画角は、撮像素子の端点とレンズ中心を結んだ線のなす角で表現される(ある一点から発せられて撮像素子に入る光線のうち、レンズの中心を通る光線は最短距離で最も光度が高いため)。撮像素子サイズSと焦点距離fに依存する。

$$\theta = 2 \tan^{-1} \frac{S/2}{f}$$

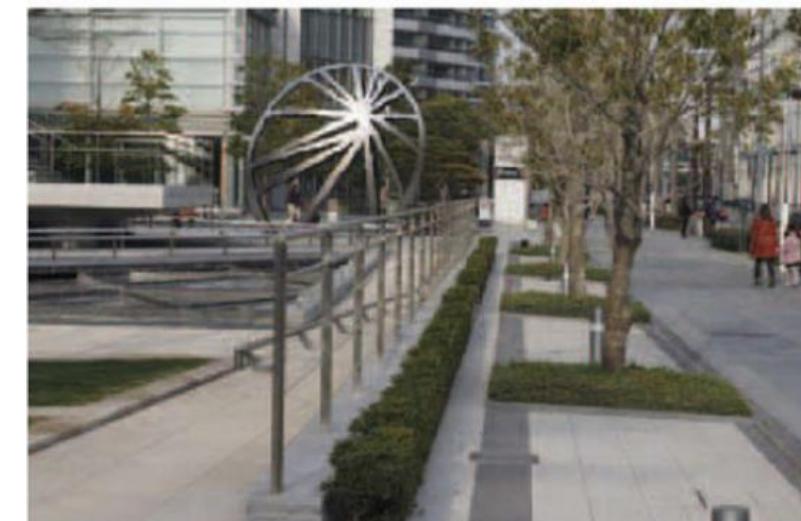
- 焦点距離が大きいほど、画角は小さくなり、遠くのものが拡大される。

フィルム時代はサイズが35mmで固定であったこの名残で
撮像素子サイズを35mmに変換した画角

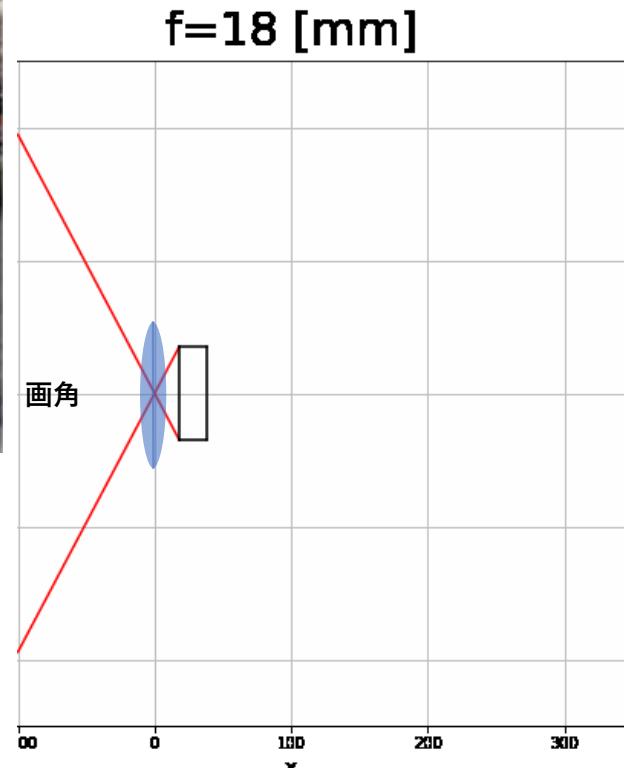
35mmフィルム換算 焦点距離	対角線画角	水平画角		垂直画角
		90.0°	67.4°	
18 mm	100.5°	90.0°	67.4°	
24 mm	広角レンズ	3.7°	53.1°	
28 mm	75.4°	65.5°	46.4°	
35 mm	63.4°	54.4°	37.8°	
50 mm	標準レンズ	9.6°	27.0°	
85 mm	28.6°	23.9°	16.1°	
100 mm	24.4°	20.4°	13.7°	
135 mm	望遠レンズ	5.2°	10.2°	
200 mm	12.3°	10.3°	6.87°	
300 mm	8.25°	6.87°	4.58°	



f=34 [mm]



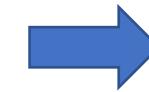
f=80 [mm]



□ = 撮像素子

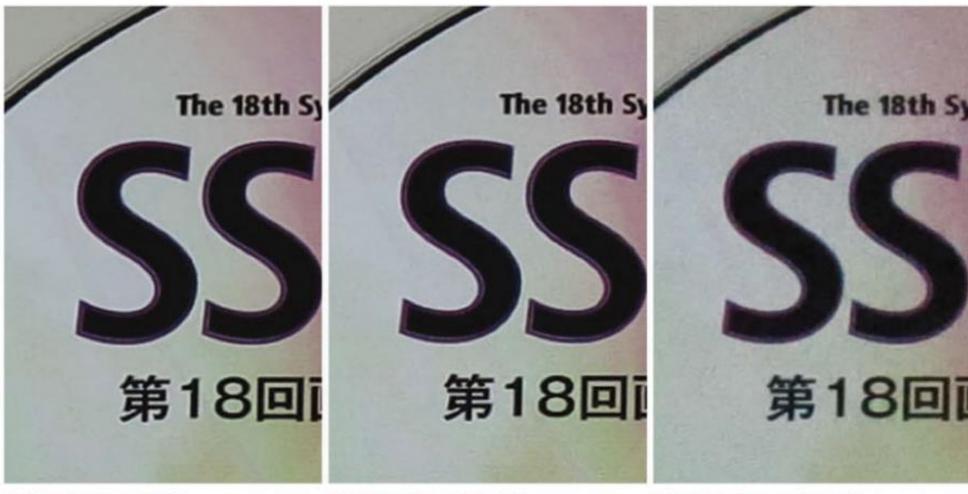
画像の明るさ

- ・(有効)レンズ径D2倍=(有効)レンズ面積4倍=明るさ4倍
- ・焦点距離f2倍=像の高さ2倍=像面積4倍=明るさ1/4倍

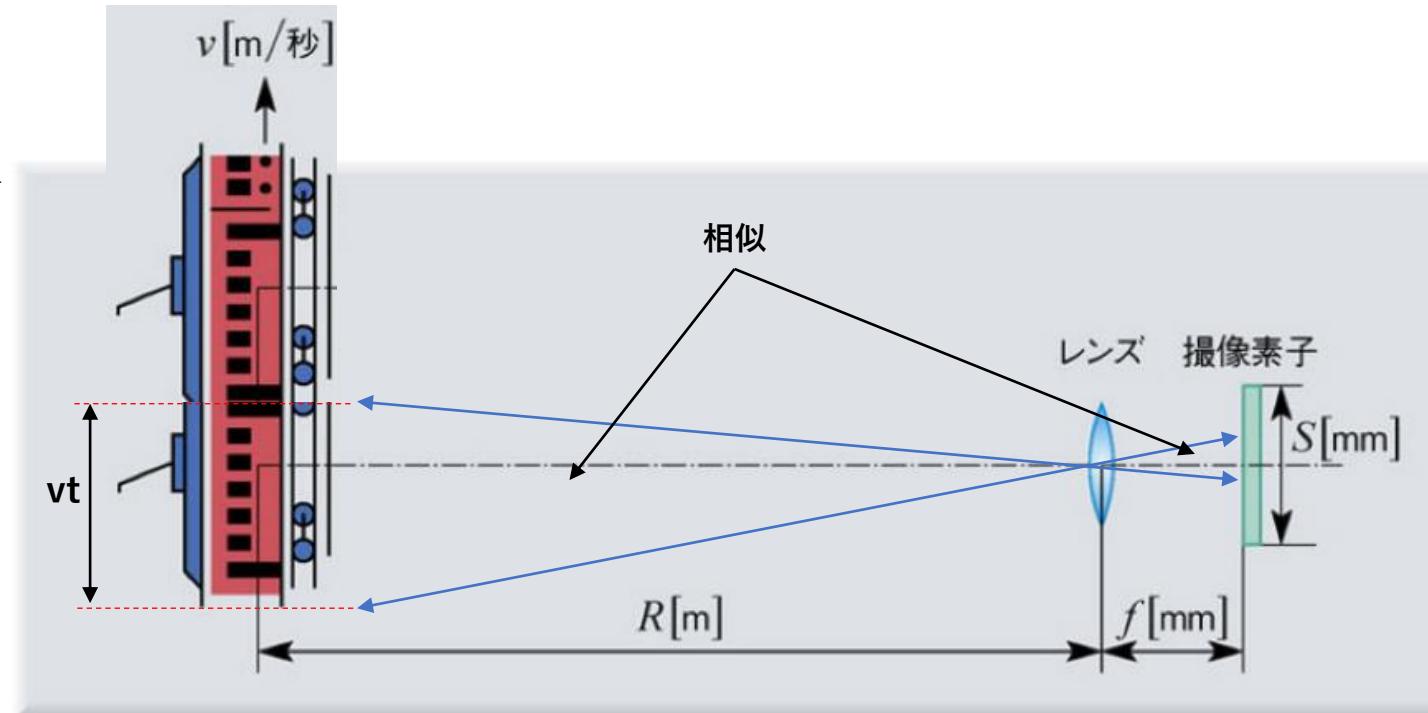


$$I \propto \left(\frac{D}{f}\right)^2 \quad F\text{ナンバ} = \frac{f}{D} \quad I \propto \frac{1}{F\text{ナンバ}^2}$$

- ・シャッタースピード(露光時間)= $1/2^n, n=10, 9, 8, \dots$ 秒
- ・露光時間を1段階上げる=光エネルギー(=光量×時間=明るさ)2倍
- ・電子シャッタ(露光時間を電子的に制御)は極めて短い露光時間も可
- ・ISO感度(光に対するフィルム感度)≒ゲイン→アナログ信号処理回路で増幅可



■図2.19—ISO感度の違いによるノイズの変化(絞り値4)

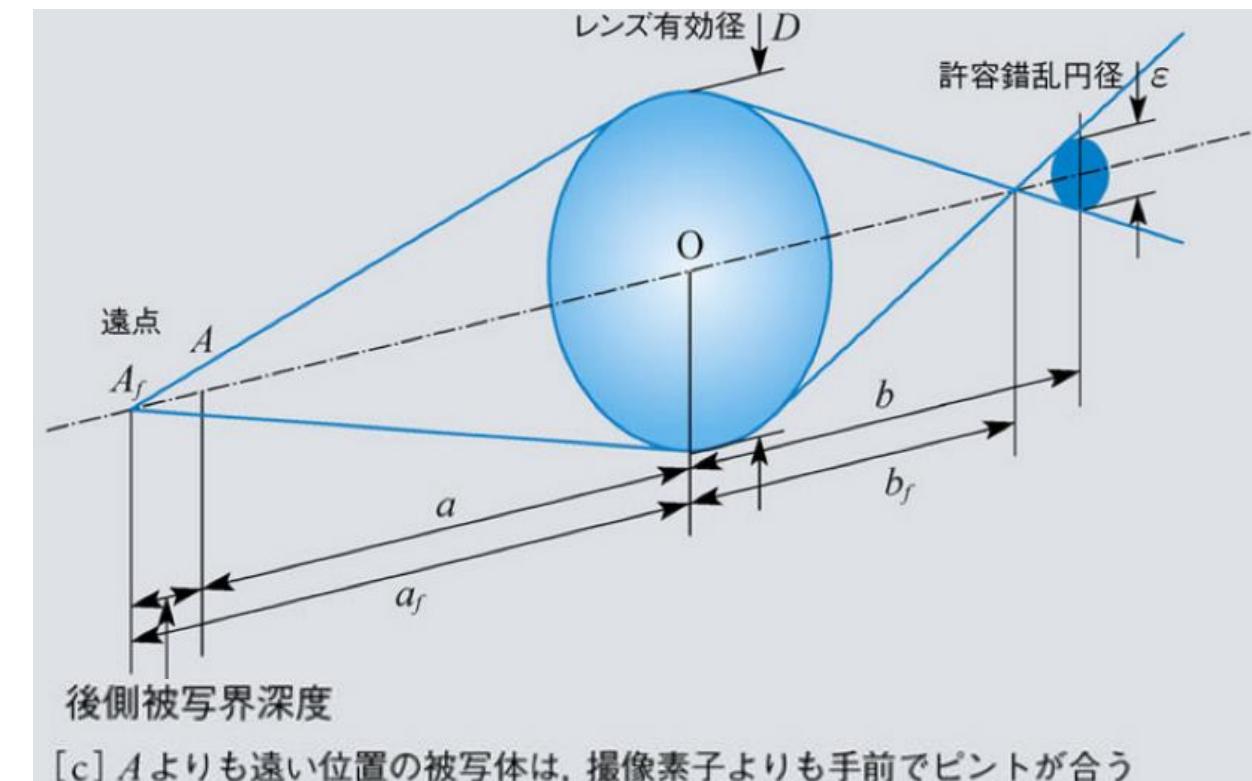
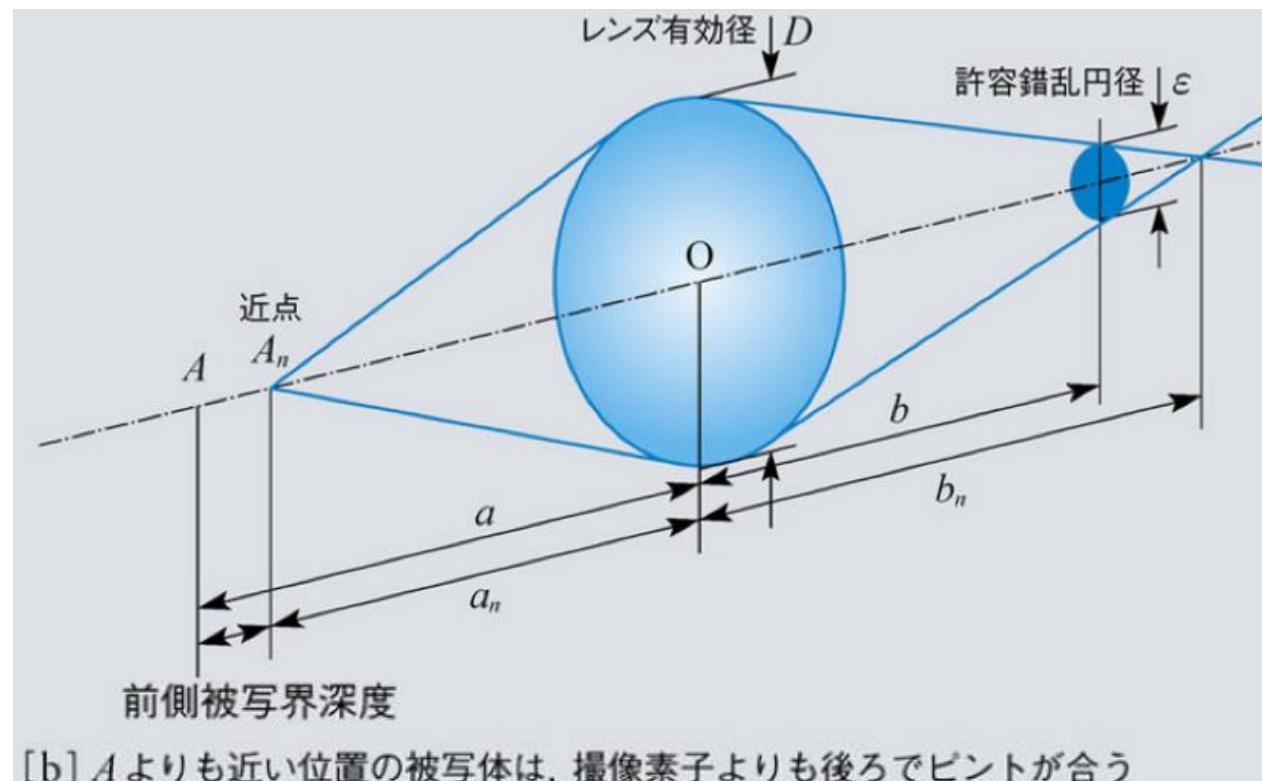
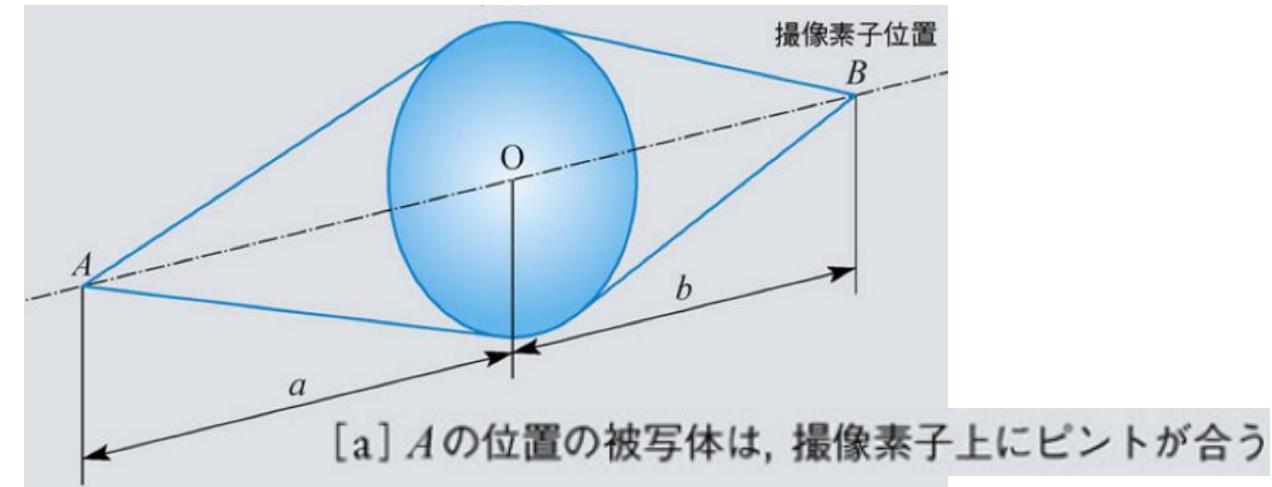


■図2.20—移動する被写体をぶれなく撮影できるシャッタスピードを求める

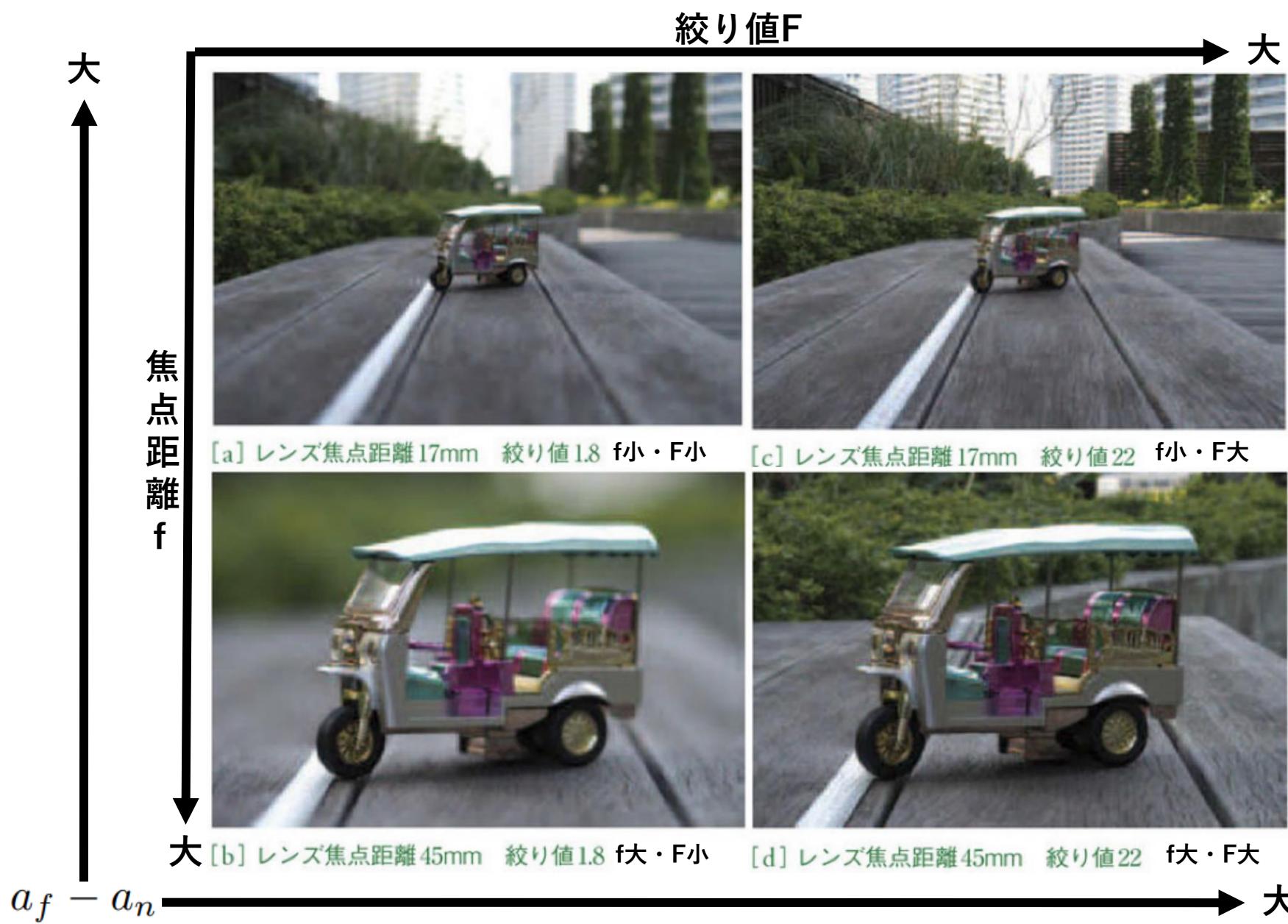
$$vt_s : R = \frac{S}{N} : f \quad \text{移動距離を1pix(S/N mm)以内に抑えるような、}$$
$$t_s = \frac{S}{N} \frac{R}{vf} \quad \text{露光時間} t_s \text{に設定すればブレのない撮影が可能}$$

被写界深度

- 錯乱円=集光位置と撮像素子位置にずれがありピントが合わないことでできる円形の像
- 許容錯乱円=点とみなせる最大の錯乱円(円径 ε)
- 被写界深度=遠点 A_f から近点 A_n までのピントが合っているとみなせる距離範囲
→深い=近くも遠くもピントが合う



被写界深度の性質



$$\text{後側被写界深度 } a_f - a = \frac{\epsilon Fa^2}{f^2 - \epsilon Fa}$$

$$\text{前側被写界深度 } a - a_n = \frac{\epsilon Fa^2}{f^2 + \epsilon Fa}$$

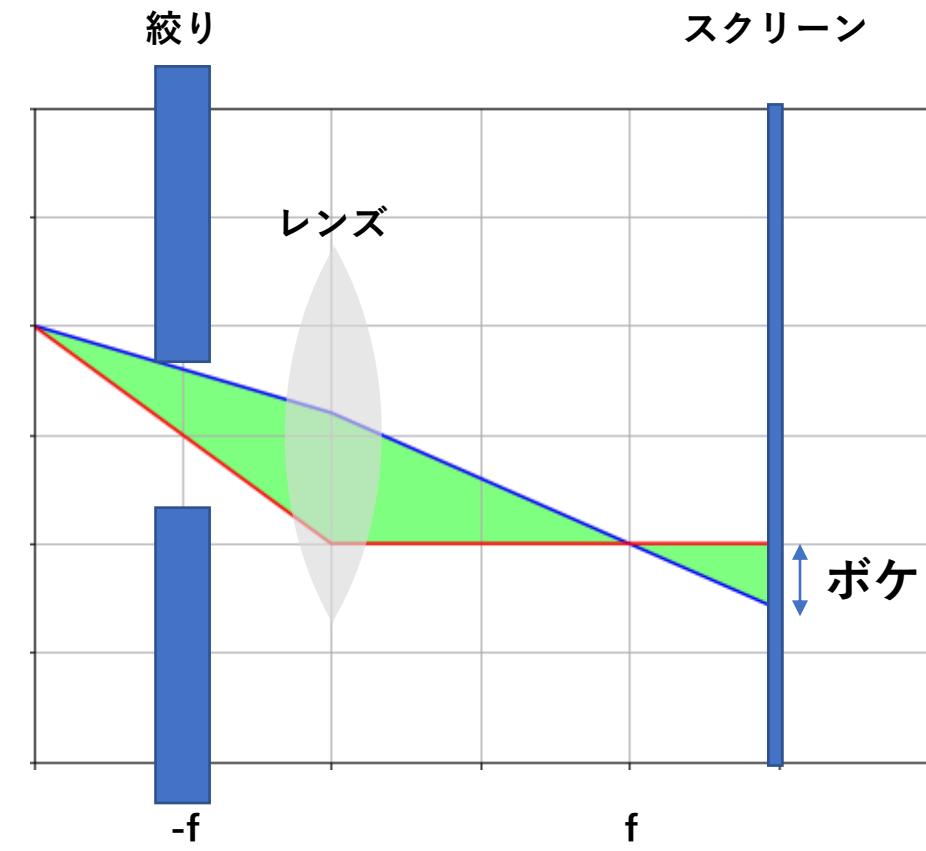
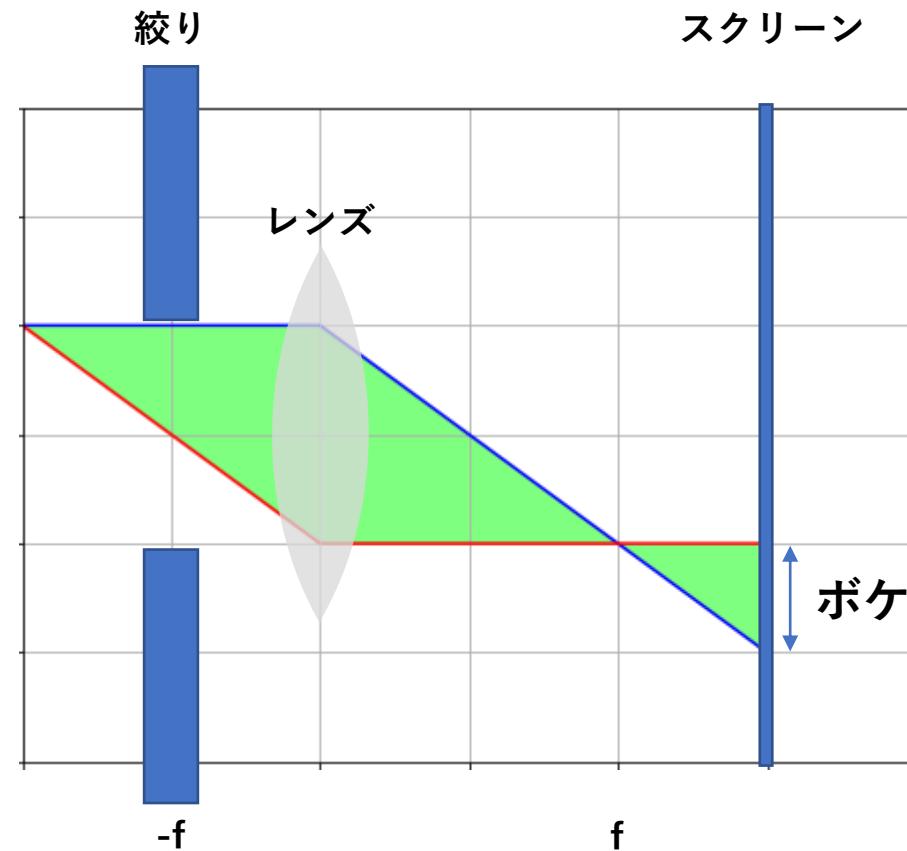
$$\text{被写界深度 } a_f - a_n = \frac{2\epsilon Fa^2 f^2}{f^4 - \epsilon^2 F^2 a^2}$$

ただし、 $a \gg f$ の近似を用いた。

$$F \text{ ナンバ} = \frac{f}{D}$$

- 後側と前側で異なるのは分母のみで、その値は後側の方が小さいから、**後側の方が深い**
- 分子にF、分母に-F²があるため**Fが大きくなるほど分母の方が相対的に小さくなり、深くなる**
- 分子にf²、分母にf⁴があるため、**fが大きくなるほど分母の方が相対的に大きくなり、浅くなる**
- 分子にa²、分母に-a²があるため、**aが大きくなるほど分母は分母の方が相対的に小さくなり、深くなる**

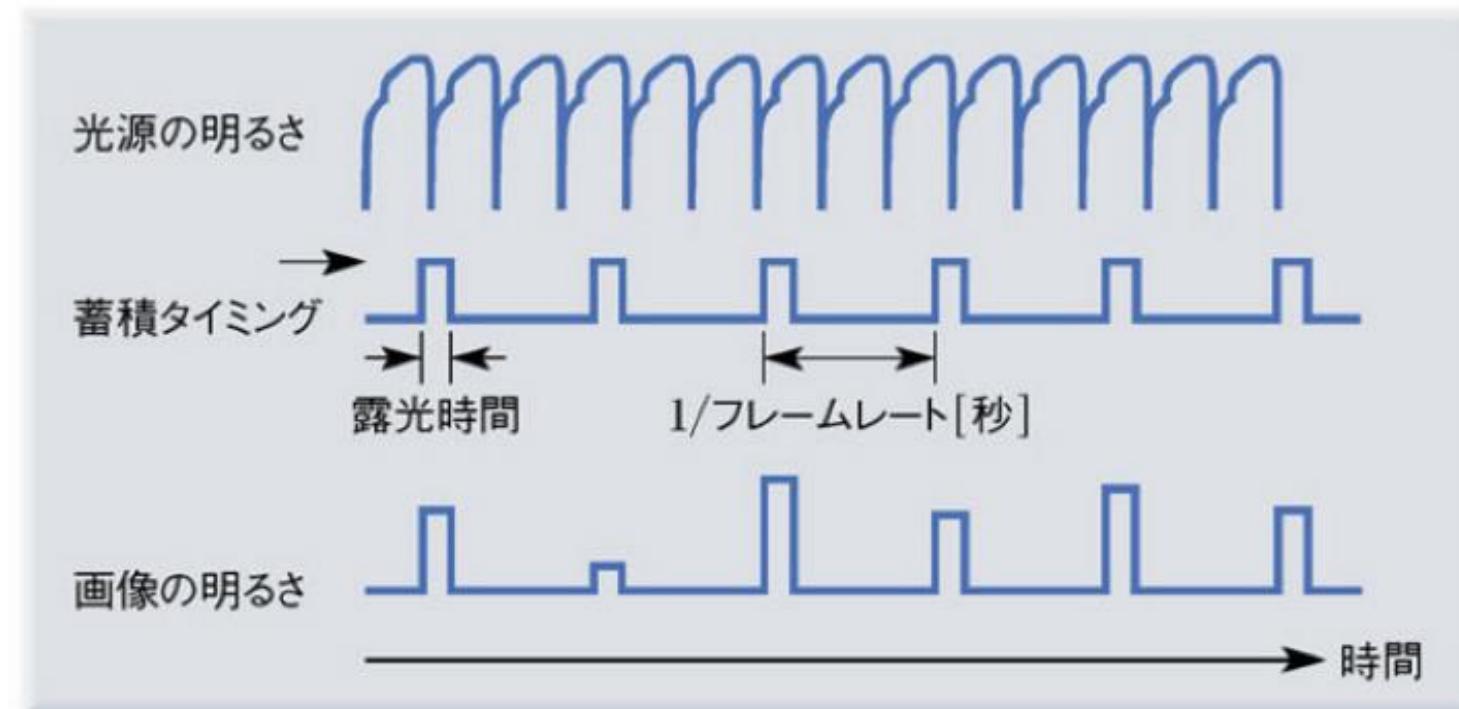
絞りとボケの関係



絞り値を大きくすると、ボケは小さくなる

フレームレート

- ・ フレームレート=1秒あたりの画像枚数[fps]
- ・ 露光時間<1/フレームレート
- ・ フリッカ=蛍光灯や水銀灯の明滅によりフレームごとに明るさが異なってしまう現象
- ・ フリッカ対策
 1. 光源を高周波点灯蛍光灯や、白熱灯に変える。
 2. シャッタースピードを光源の明滅の周期の整数倍に設定。



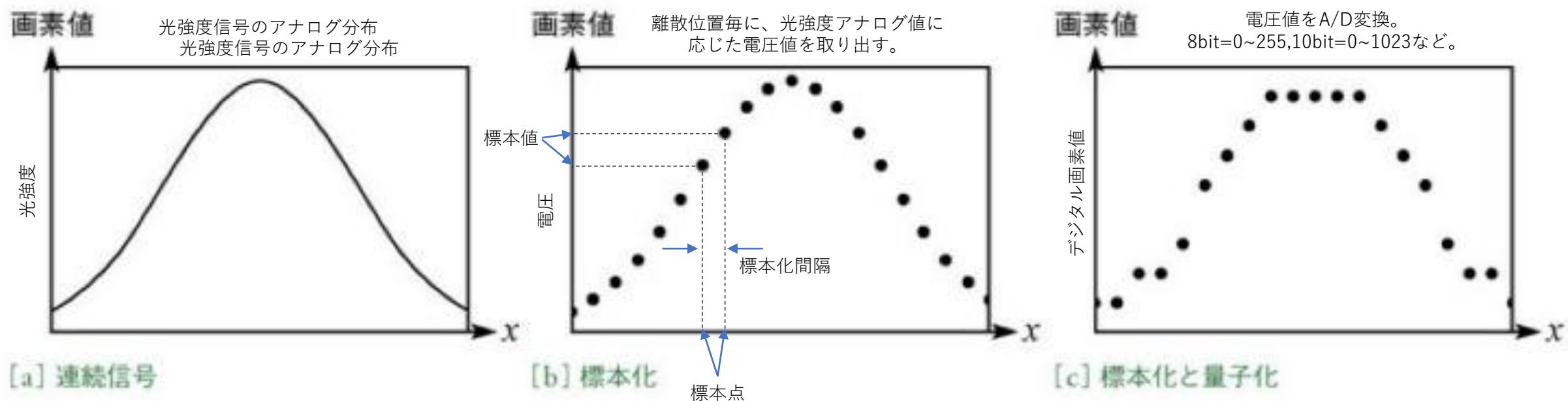
■図2.26——フリッカの発生過程

画像のデジタル化

- 標本化と量子化
- カメラ応答関数
- 時系列画像
- カラー画像

標本化と量子化

- グレースケール画像…撮像素子上に結像した像に標本化と量子化を施したもの
- 量子化レベル数…デジタル画素値の段階数。8bitなら256、10bitなら1024
- 量子化誤差…A/D変換時の丸め誤差



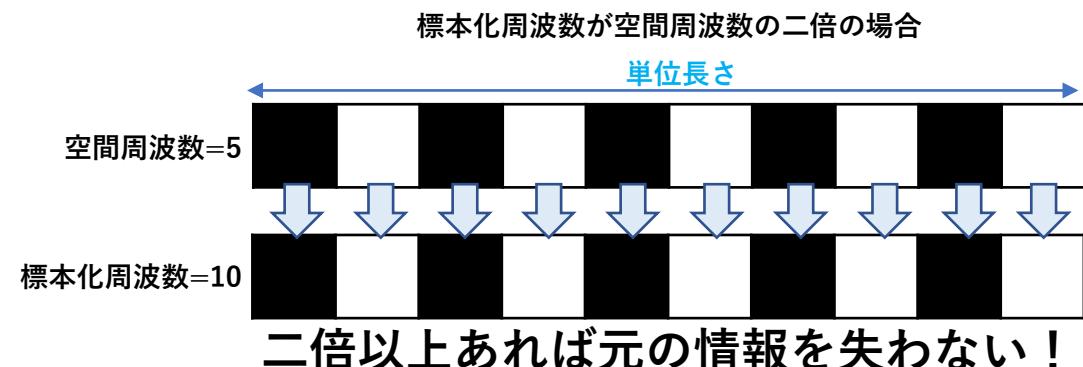
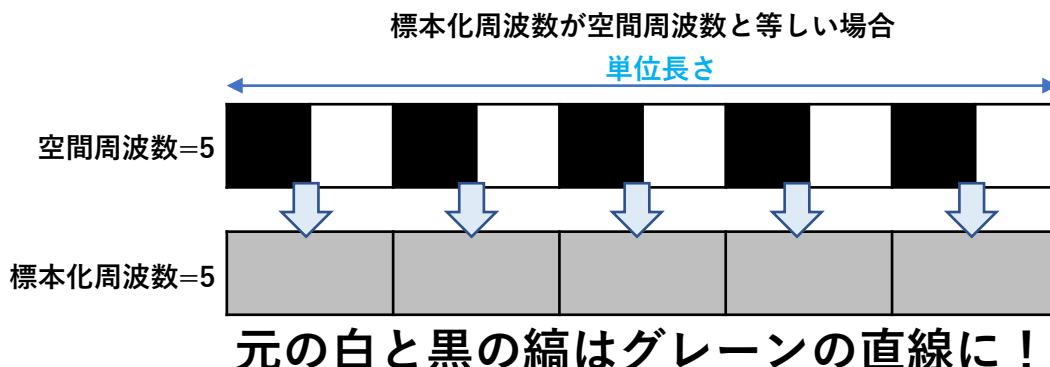
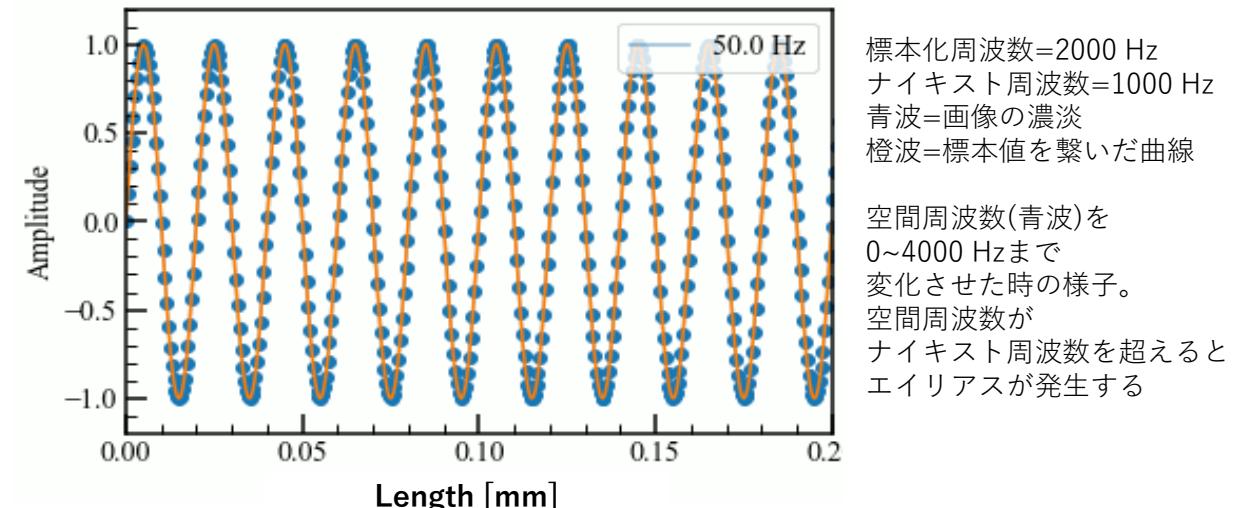
■図2.30——アナログ画像の標本化と量子化

標本化定理・エイリアシング

- ・ 空間周波数…単位長さあたりの濃淡の繰り返し数
- ・ 標本化周波数…単位長さあたりの標本点数=標本化間隔の逆数
- ・ ナイキスト周波数…標本化周波数の1/2
- ・ 標本化定理…アナログ信号の2f_{max}以上の標本化周波数で標本化すれば、信号の情報を失わない。
- ・ エイリアシング…標本化定理を満たしていないときに生じる偽信号。動画像では、車輪が逆回転しているように見えるなど、対象の移動周波数とフレームレートとの間に生じる。
- ・ 補間…標本化定理を満たしている場合に、元のアナログ信号を求ること。

$$f(t) = \sum_{k=-\infty}^{\infty} f_k \operatorname{sinc} \frac{t-t_k}{\tau} \quad \operatorname{sinc} x = \frac{\sin \pi x}{\pi x}$$

τ は標本化間隔



ダウンサンプリング・疑似輪郭

- ・ ダウンサンプリング…カメラの画素数>ディスプレイの画素数の時に画像サイズを縮小すること
- ・ $1/N$ に縮小する前、 $1/2N$ 以上の周波数を遮断しないと、エイリアシングが発生



遮断せずに縮小

遮断後に縮小



疑似輪郭(疑似エッジ)

量子化レベル数が小さいことによって、
画素値の境界が等高線のように画像に現れること。

量子化雑音

量子化誤差によって生じる歪み。疑似輪郭もその一つ。



[a] 量子化レベル数256, 8ビット量化



[d] 量子化レベル数8, 3ビット量化

カメラ応答関数

- ・ カメラ応答関数…入射光強度を入力として画素値を出力とする関数
- ・ ディスプレイには各機器固有の発色特性(クセ)があり、正規化された入力と出力の関係は $y_{disp} = x_{disp}^\gamma$ で表せる。
- ・ ガンマ補正…そのクセの分を補正するために、撮影時に $y_{cam} = x_{cam}^{\frac{1}{\gamma}}$ で変換しておくことで、カメラへの入力とディスプレイへの出力の関係を線形にすること

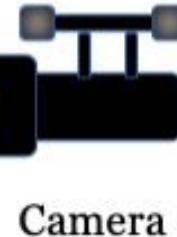
$$y_{disp} = x_{disp}^\gamma = (y_{cam})^\gamma = (x_{cam}^{\frac{1}{\gamma}})^\gamma = x_{cam} \quad \text{各値は正規化しているとする}$$



Windows,多くのディスプレイ,ブラウン管 $\rightarrow \gamma = 2.2$

Mac OS $\rightarrow \gamma = 1.8$

産業用カメラ $\rightarrow \gamma = 1.0$



$$x_{cam}$$



$$y_{cam} = \underbrace{x_{cam}^{\frac{1}{\gamma}}}_{x_{disp}} = x_{disp}$$



$$x_{disp}$$



$$y_{disp} = x_{disp}^\gamma = (y_{cam})^\gamma$$

カラー画像

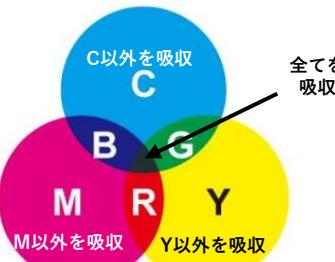
• 混色

「加法混色の三原色」であるRGB
モニター、スポットライトなど光の色



混色すると色が明るくなる

「減法混色の三原色」であるCMY
絵具、カラー印刷など物体の色



混色すると色が暗くなる

バイリニア補間

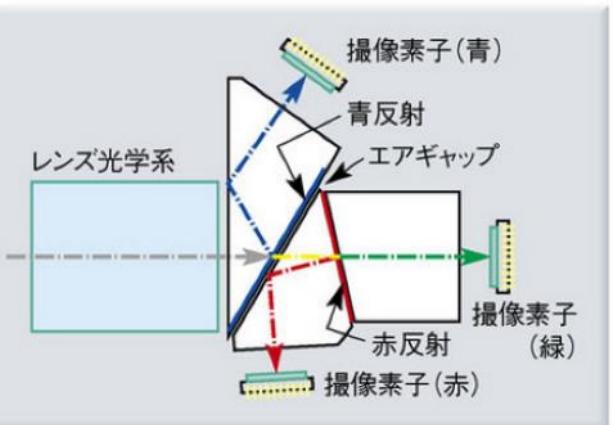
R1	G2	R3	G4	R5	G6
G7	B8	G9	B10	G11	B12
R13	G14	R15	G16	R17	G18
G19	B20	G21	B22	G23	B24
R25	G26	R27	G28	R29	G30
G31	B32	G33	B34	G35	B36

$$G15 = (G9 + G14 + G16 + G21)/4$$

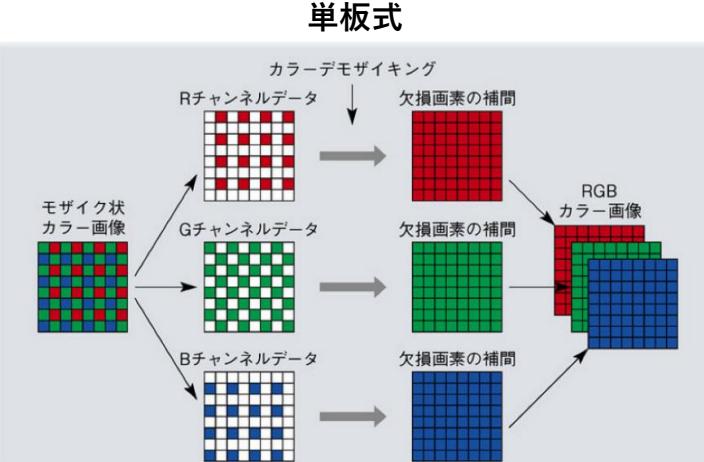
$$R16 = (R15 + R17)/2$$

• カラー画像撮影方法

3板式



単板式



エッジセンシング補間

バイリニアではある単色エッジが点線になり、それにより偽色が表れることがある。エッジセンシング補間では、その影響を省く。同一エッジ上の画素値は差が少ないと利用して、注目画素が上下方向エッジか左右方向エッジ化を特定し、上下方向なら上下の画素の平均、左右方向なら左右の画素の平均で補間

$$\alpha = |(R13+R17)/2 - R15|$$

$$\beta = |(R3+R27)/2 - R15|$$

$$G15 = \begin{cases} (G14+G16)/2 & (\alpha < \beta), \\ (G9+G21)/2 & (\alpha > \beta), \\ (G14+G16+G9+G21)/4 & (\alpha = \beta). \end{cases}$$

(左右の平均-対象)

(上下の平均-対象)

左右方向エッジ

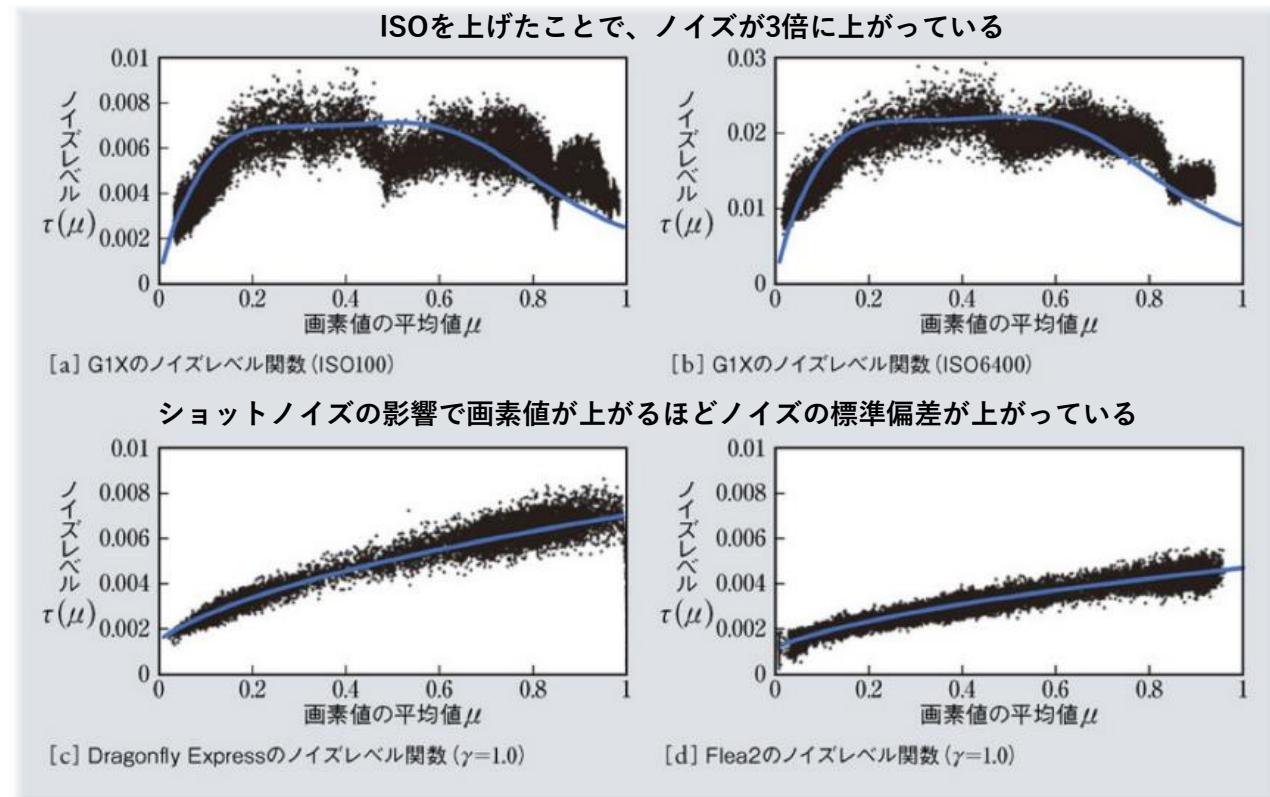
上下方向エッジ

上下と左右に差がない

3. 画像の性質と色空間

画像のノイズ

- 各受光素子に到達するフォトン数はポアソン分布に従う。
- 光ショットノイズ…ある受光素子についてのフォトン数の揺らぎ。複数回の測定によって得られる平均フォトン数n、分散nで揺らぐ。
- 暗時ノイズ…フォトン数によらないノイズ(暗電流、読み出し、固定パターンなど)の総称。平均0のガウシアンで近似できる。
- ノイズレベル関数 τ …説明変数=各画素の画素値の平均値(正規化)、目的変数=各画素値の標準偏差とした関数
→ 安定した照明の下、固定カメラで、静止シーンを複数回撮影して求めた各画素位置の平均値、標準偏差からプロットすれば得られる



■図3.4——キヤノンG1XとPoint Grey Research社の2種類のカメラのノイズレベル関数の測定結果例(黒点)
画素値の平均値 μ は、0から1に正規化されている。ノイズレベルは、画素値の最大値が1のときの大きさである。それぞれ、100枚の画像から求めている。

光強度と露光時間の積(露光量) I カメラ応答関数 g 画素値 M
 n_{ps} は露光量 I に依存する光ショットノイズ、 n_c はガウスノイズ

$$M = g(I + n_{ps} + n_c)$$

$$M \approx g(I) + g'(I)(n_{ps} + n_c) \quad g \text{に関して } I \text{ のまわりで1次泰イラー展開}$$

$$\mu = E[M] \approx g(I) \quad E[n_{ps}] = E[n_c] = 0 \text{ であるため}$$

$$\tau^2(\mu) = (g'(I))^2 (I\sigma_{ps}^2 + \sigma_c^2) = (g'(I)\sigma_c)^2 (I\rho^2 + 1)$$

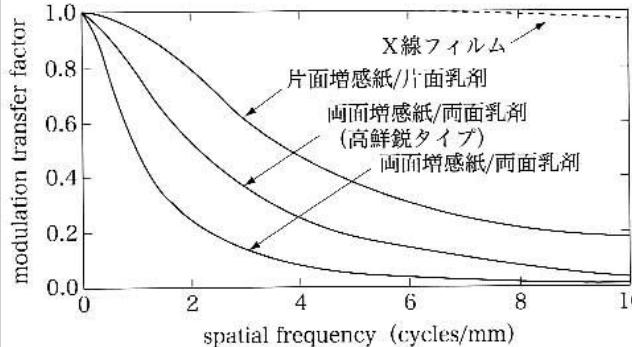
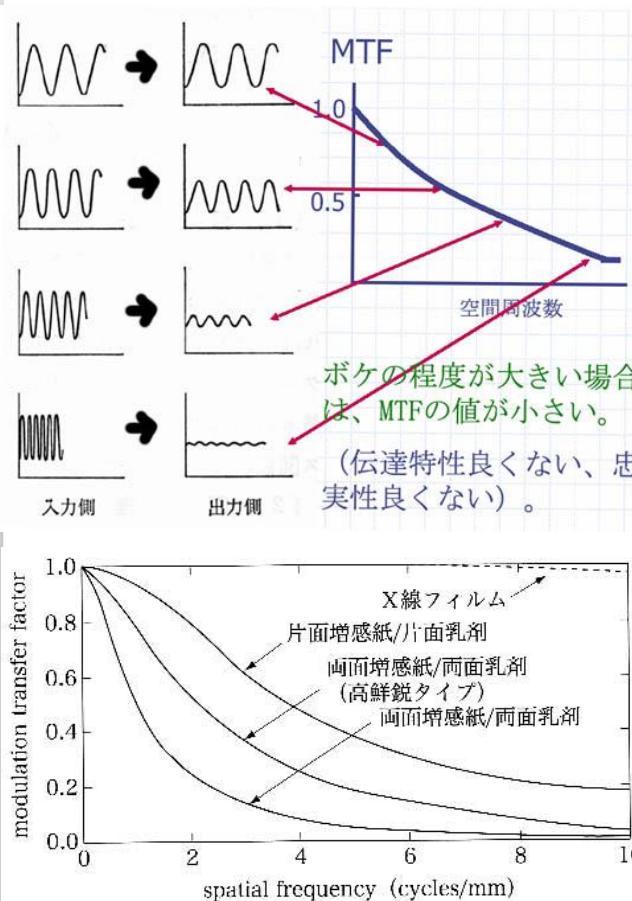
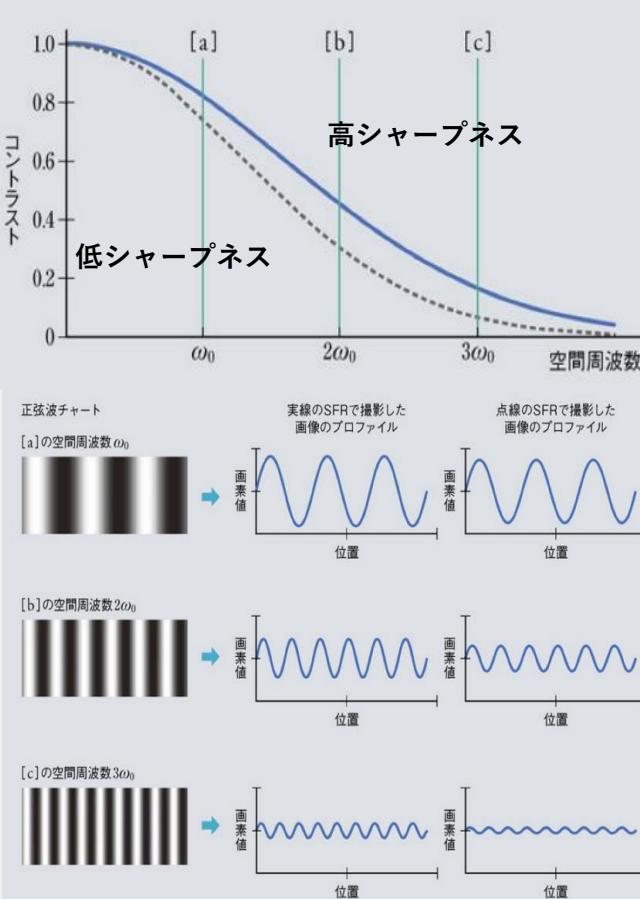
$$\tau(\mu) = g'(g^{-1}(\mu))\sigma_c \sqrt{g^{-1}(\mu)\rho^2 + 1}$$

コントラストとシャープネス

- コントラスト…ヒストグラムの広がりの大きさ
- シャープネス(鮮鋭度)…エッジの急激さ。ボケ具合。
- シャープネスの測定方法…空間周波数に対するコントラスト(**空間周波数応答:SFR**)を計測
空間周波数が高くなるほど、コントラストは低下

SFR(≒OTF、MTF)

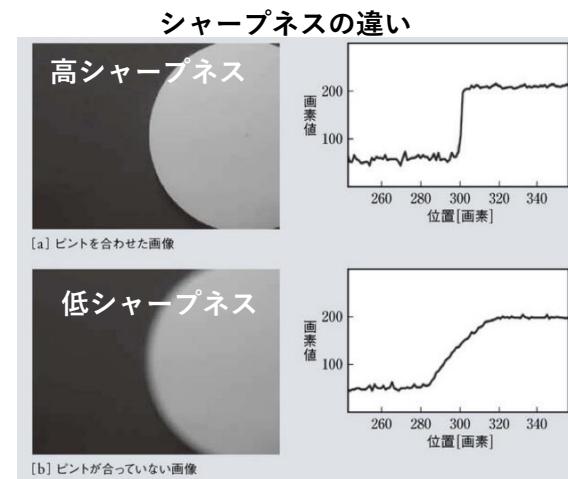
画像は様々な要因でボケる。細かいものほどボケる。カメラは通常空間周波数が高くなるほど明暗の差が再現しづらくなる。人間の目でも白黒の縞模様が徐々に細かくなつていけばやがて全体がグレーに見えることに対応する。細かいもののコントラストをどこまで再現できるかがSFR。



$$C1 = I_{\max} - I_{\min}$$

$$C2 = (I_{\max} - I_{\min}) / (I_{\max} + I_{\min})$$

$$C3 = I_{\max} / I_{\min}$$



解像力

どれだけ細かなものまで再現されるか。
解像力テストチャートを撮影し、識別できる限界線密度で定義されている

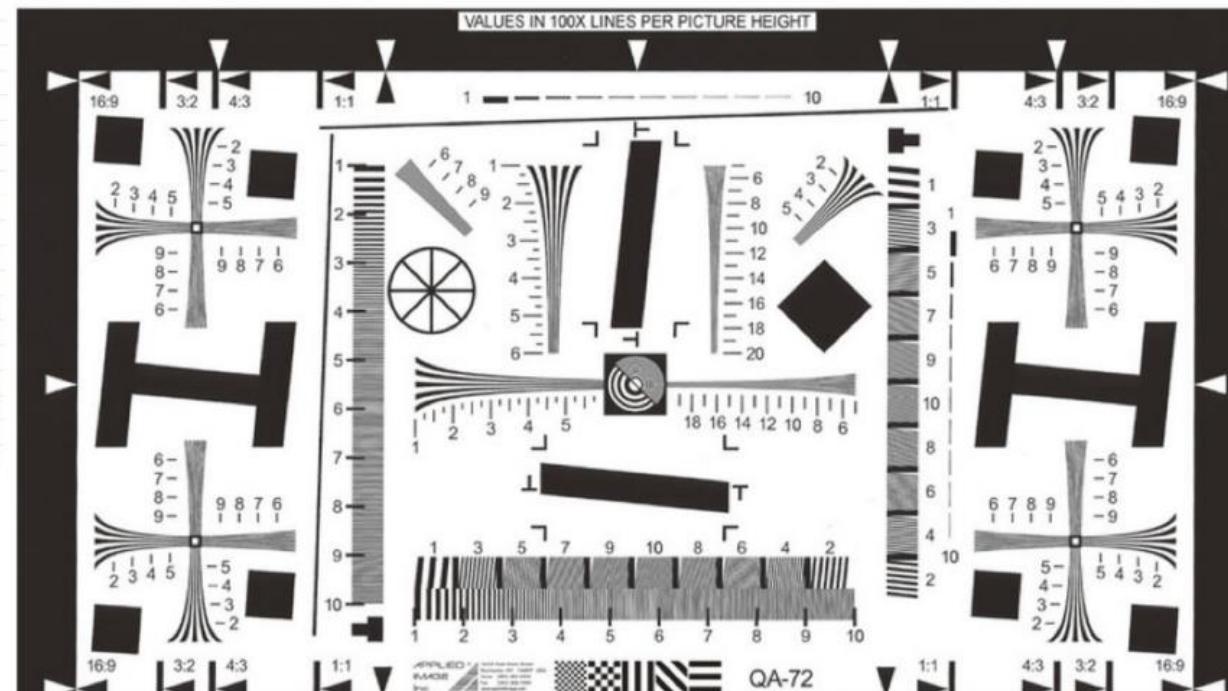
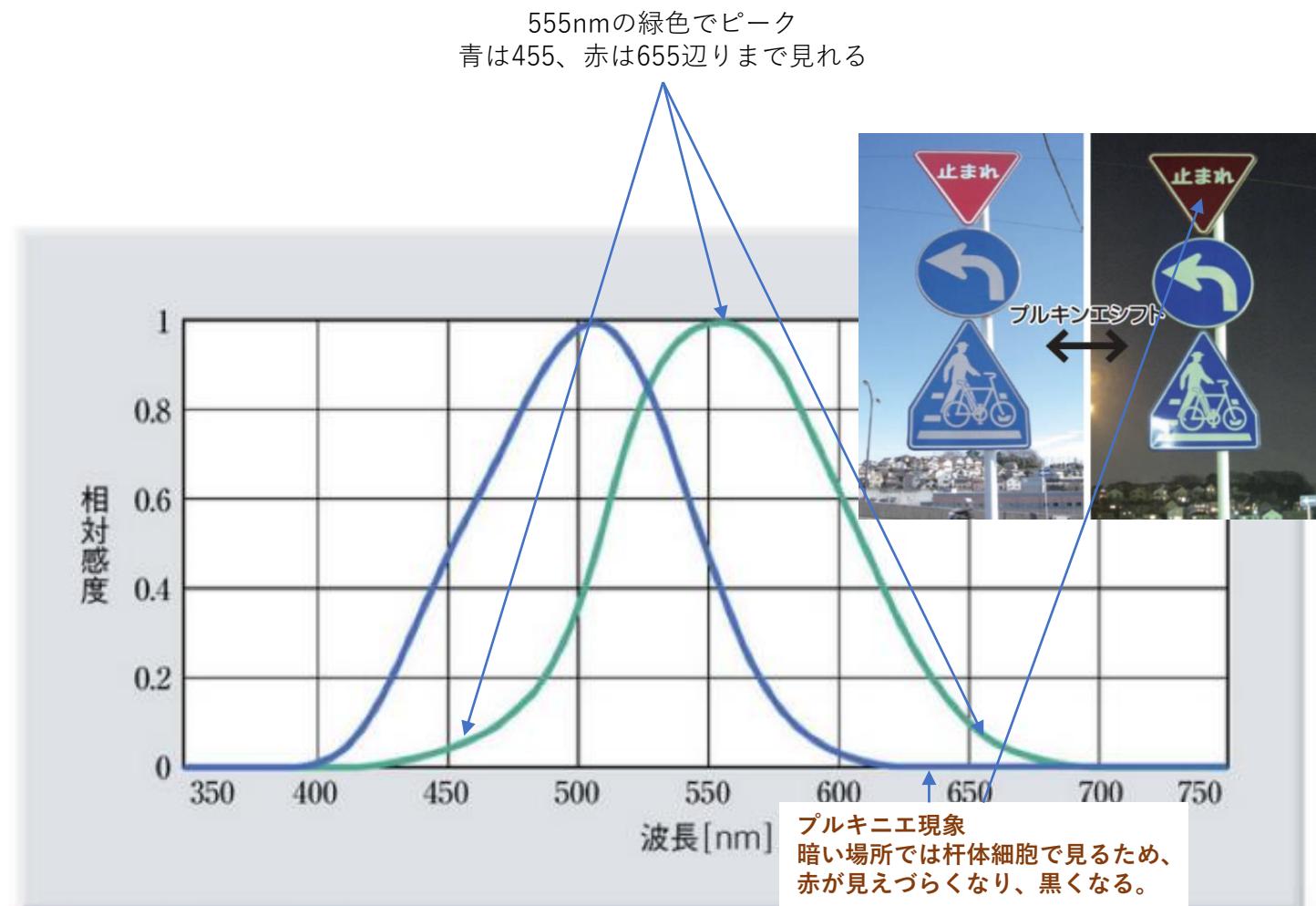
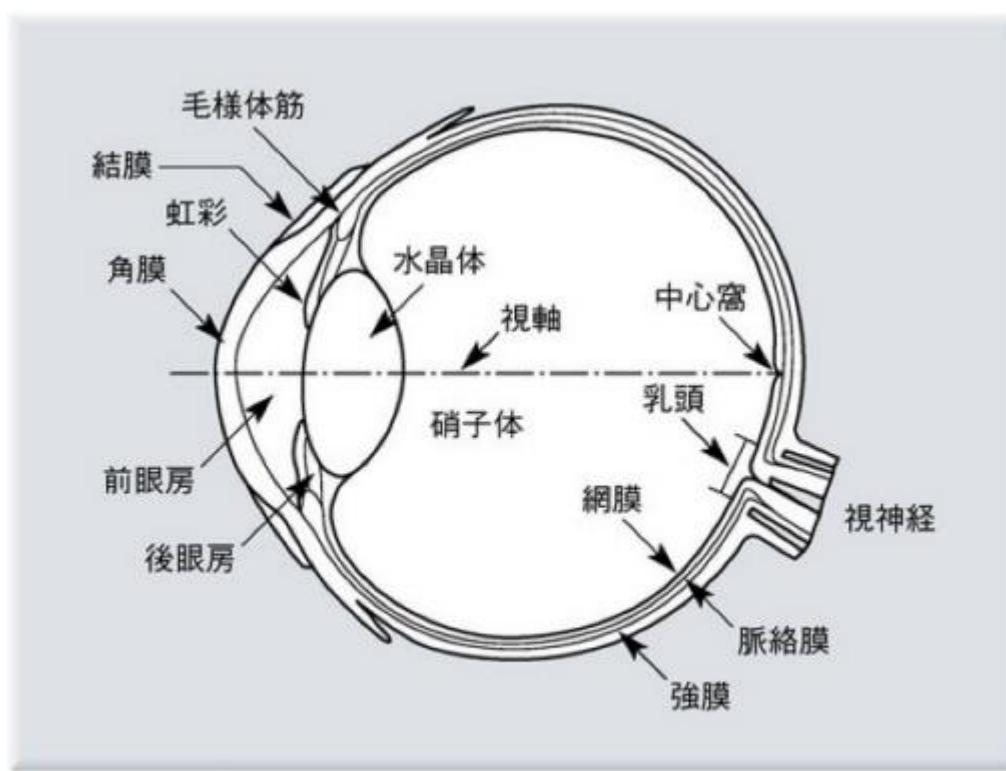


図3.7—解像力テストチャートの例

(Applied Image QA-72 Resolution Chart from Applied Image Inc. Reproduced with permission of Applied Image Inc.)

人間の視覚

- 明所視…錐体細胞(L =赤, M =緑, S =青)で明るい場所を見る視覚。レッド、ミドリ、セイで覚える
- 暗所視…杆体細胞で暗い場所を見る視覚
- 薄明視…中間の明るさで推移、杆体両方が働く視覚



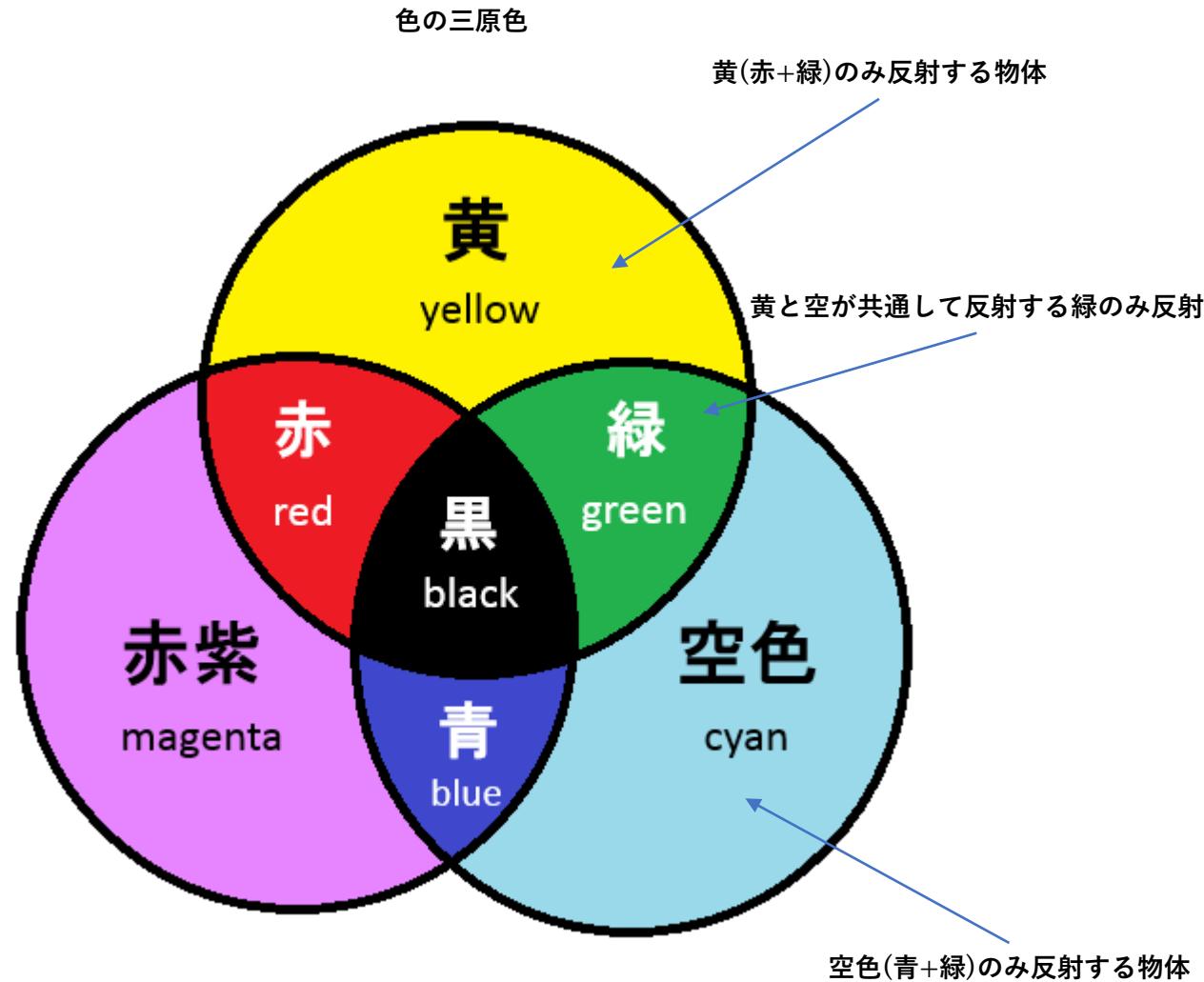
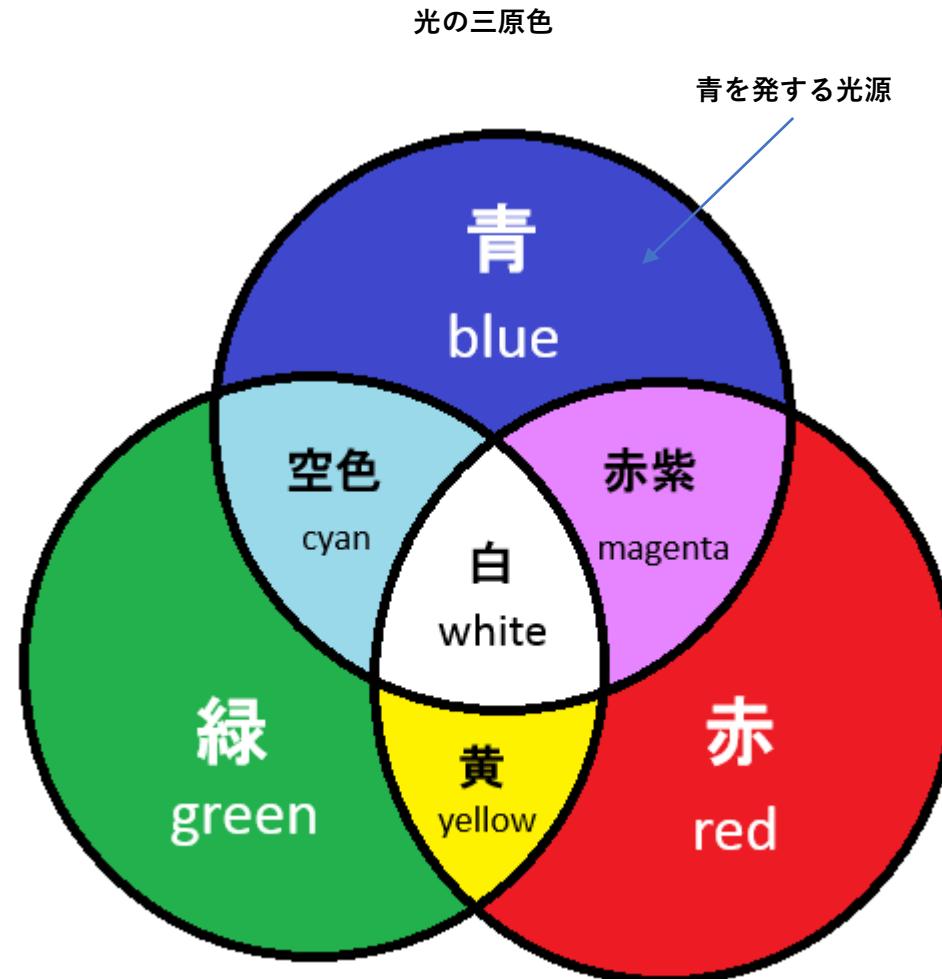
■図3.9——視感度特性

緑線は標準分光視感効率 $V(\lambda)$ (CIE1924, standard photopic luminosity function) を表し、錐体による明所視の視感度特性を表す。

青線は杆体による暗所視の視感度特性 $V'(\lambda)$ (CIE1951, standard scotopic luminosity function) を表す。

■図3.8——眼球構造 (右眼を上方から見た図)

光と色の三原色



表色系と色空間

表色系

色を数字で表すことを可能にする体系。顔色系と混色系に二分される。

混色系

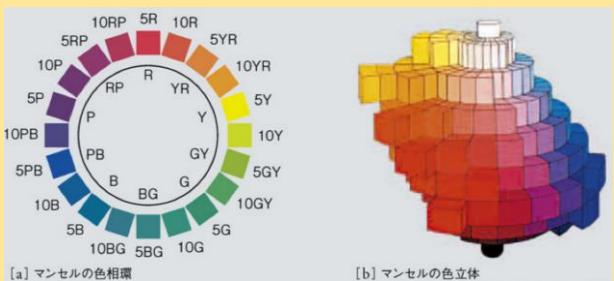
実験に基づいて色を定量的に扱う表色系

顔色系

記号や色票などで定性的に扱う表色系

マンセル表色系

例えば、10R7/8なら色相10R、明度7、彩度8



CIE-L*a*b*色空間

色の類似度について色度座標上の距離よりも、人間の知覚に近い値を計算するためのCIE均等色空間の一つ。

$$L^* = 116 f(Y/Y_n) - 16$$

$$a^* = 500(f(X/X_n) - f(Y/Y_n))$$

$$b^* = 200(f(Y/Y_n) - f(Z/Z_n))$$

f は引数が小さくても勾配が無限大にならないようにするための関数。

$$f(\alpha) = \begin{cases} \alpha^{\frac{1}{3}}, & \alpha > (24/116)^3 \\ (841/108)\alpha + 16/116, & \alpha \leq (24/116)^3 \end{cases}$$

類似度はL,a,b座標上でのユークリッド距離 Δ で求める。

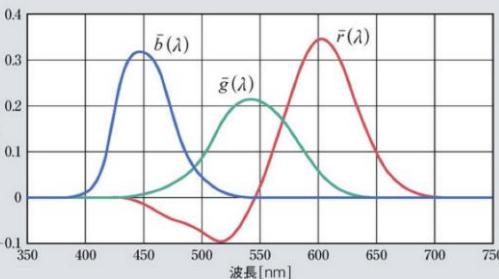
$$\Delta = [(L_1^* - L_2^*)^2 + (a_1^* - a_2^*)^2 + (b_1^* - b_2^*)^2]^{\frac{1}{2}}$$

CIE-RGB表色系

原刺激RGBを波長700nm、546.1nm、435.8nmとして等色関数を用いて、反射光Cを求める体系。しかし負の値があるため手計算しづらい、全ての色は表せないという欠点がある。原刺激の係数を三刺激値と呼ぶ。

$$C = RR + GG + BB$$

$$\begin{cases} R = \int_V L(\lambda) \bar{r}(\lambda) d\lambda \\ G = \int_V L(\lambda) \bar{g}(\lambda) d\lambda \\ B = \int_V L(\lambda) \bar{b}(\lambda) d\lambda \end{cases}$$



sRGB色空間(Standard RGB)

様々なデバイス、OS間に共通に利用されている色空間。
測定したり、再現したりする色が装置に依存する色情報をデバイス依存色、CIE-XYZ、sRGBなどの装置に依存しないものはデバイス独立色と呼ぶ。
カラー画像を撮影したり、表示したりするときは、デバイス依存色とデバイス独立色間の変換を行い、装置間で伝えている。そのための標準規格がsRGB。
sRGBであればどのデバイスもサポートしているため、sRGBに変換して伝送される。

sRGB → CIE-XYZ

γ変換

$$E'_{sRGB} = E'_{8bit}/255.0$$

$$E_{sRGB} = \begin{cases} E'_{sRGB}/12.92, & E'_{sRGB} \leq 0.04045 \\ ((E'_{sRGB} + 0.055)/1.055)^{2.4}, & E'_{sRGB} > 0.04045 \end{cases}$$

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.4124 & 0.3576 & 0.1805 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9505 \end{pmatrix} \begin{pmatrix} R_{sRGB} \\ G_{sRGB} \\ B_{sRGB} \end{pmatrix}$$

CIE-XYZ→sRGB

$$\begin{pmatrix} R_{sRGB} \\ G_{sRGB} \\ B_{sRGB} \end{pmatrix} = \begin{pmatrix} 3.2406 & -1.5372 & -0.4989 \\ -0.9689 & 1.8758 & 0.0415 \\ 0.0557 & -0.2040 & 1.0570 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

γ変換

$$E'_{sRGB} = \begin{cases} 12.92E_{sRGB}, & E_{sRGB} \leq 0.0031308 \\ 1.055E_{sRGB}^{(1/2.4)} - 0.055, & E_{sRGB} > 0.0031308 \end{cases}$$

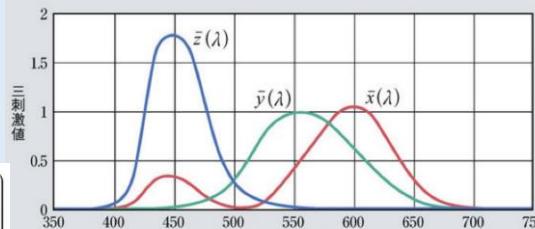
$$E'_{8bit} = 255.0E'_{sRGB}$$

CIE-XYZ表色系

CIE-RGB表色系において等色関数を負の値を持たない値XYZに線形変換した表色系。

$$\begin{cases} X = \int_V L(\lambda) \bar{x}(\lambda) d\lambda \\ Y = \int_V L(\lambda) \bar{y}(\lambda) d\lambda \\ Z = \int_V L(\lambda) \bar{z}(\lambda) d\lambda \end{cases}$$

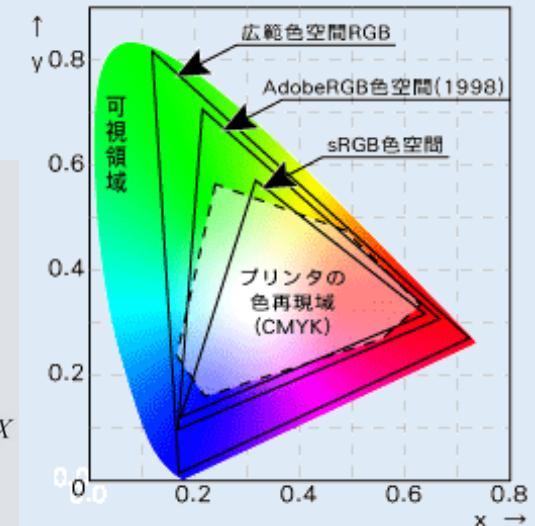
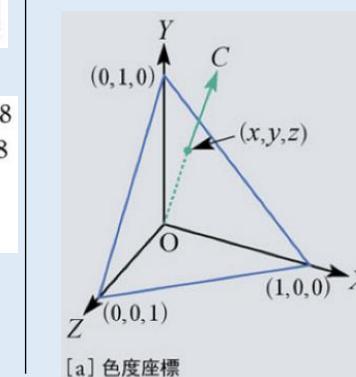
$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 2.7689 & 1.7517 & 1.1302 \\ 1.0000 & 4.5907 & 0.0601 \\ 0.0000 & 0.0565 & 5.5943 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$



色度座標・xy色度図

x,y,zの明るさの和を1とした場合の色を表すベクトルを色度座標と呼ぶ。x,yの明るさが決まればzも決まり色は一意に決まるため、xy平面上その色をプロットしたものはxy色度図と呼ばれ、人間が知覚できる全ての色を表現している。釣りがね状の曲線部分をスペクトル軌跡、底辺の直線を純紫軌跡と呼ぶ。ディスプレイが表示できる色の範囲は三原色の点を頂点に持つ三角形の内部になる。

$$\begin{cases} x = \frac{X}{X+Y+Z} \\ y = \frac{Y}{X+Y+Z} \\ z = \frac{Z}{X+Y+Z} \end{cases}$$



YUV変換・HSI変換

- YUV変換…アナログカラーテレビ放送の設計で、電波の占有帯域を小さくしつつ高品質な動画像を放送するため、RGB値を輝度と二種の色差に変換すること

$$\begin{cases} Y = 0.299R' + 0.587G' + 0.114B' \\ U = 0.492(B' - Y) = -0.147R' - 0.289G' + 0.436B' \\ V = 0.877(R' - Y) = 0.615R' - 0.515G' - 0.100B' \end{cases}$$

$$\begin{cases} Y = 0.299R' + 0.587G' + 0.114B' \\ C_b = 0.564(B' - Y) = -0.169R' - 0.331G' + 0.500B' \\ C_r = 0.713(R' - Y) = 0.500R' - 0.419G' - 0.081B' \\ Y = 0.2126R' + 0.7152G' + 0.0722B' \\ C_b = 0.5389(B' - Y) = -0.1146R' - 0.3854G' + 0.5000B' \\ C_r = 0.6350(R' - Y) = 0.5000R' - 0.4542G' - 0.0458B' \end{cases}$$

- 明るさ一定で、彩度を変えたい場合、RGBを色相(hue)、彩度(saturation)、明度(intensity)に変換(HSI変換)し、これらの値を調整し、再びRGBに戻す(HSI逆変換)。

- YCbCr
- HSV(六角錐モデル)

$$I_{\max} = \max\{R, G, B\}, I_{\min} = \min\{R, G, B\}$$

$$I = I_{\max}$$

$$S = (I_{\max} - I_{\min}) / I_{\max}$$

$$H = \begin{cases} \frac{G-B}{I_{\max}-I_{\min}} \frac{\pi}{3} & (I_{\max}=R) \\ \frac{B-R}{I_{\max}-I_{\min}} \frac{\pi}{3} + \frac{2}{3}\pi & (I_{\max}=G) \\ \frac{R-G}{I_{\max}-I_{\min}} \frac{\pi}{3} + \frac{4}{3}\pi & (I_{\max}=B) \end{cases}$$

$$(R, G, B) = \begin{cases} (I, T, P) & (h=0) & h = \text{floor}\left(\frac{3}{\pi}H\right) \\ (Q, I, P) & (h=1) & P = I(1-S) \\ (P, I, T) & (h=2) & Q = I\left(1-S\left(\frac{3}{\pi}H-h\right)\right) \\ (P, Q, I) & (h=3) & T = I\left(1-S\left(1-\frac{3}{\pi}H+h\right)\right) \\ (T, P, I) & (h=4) \\ (I, P, Q) & (h=5) \end{cases}$$

- HSL(双六角錐モデル)

$$I_{\max} = \max\{R, G, B\}, I_{\min} = \min\{R, G, B\}$$

$$I = (I_{\max} + I_{\min})/2$$

$$S = \begin{cases} \frac{I_{\max} - I_{\min}}{I_{\max} + I_{\min}} & (I \leq 0.5) \\ \frac{I_{\max} - I_{\min}}{2 - (I_{\max} + I_{\min})} & (I > 0.5) \end{cases}$$

$$H = \begin{cases} \frac{G-B}{I_{\max}-I_{\min}} \frac{\pi}{3} & (I_{\max}=R) \\ \frac{B-R}{I_{\max}-I_{\min}} \frac{\pi}{3} + \frac{2}{3}\pi & (I_{\max}=G) \\ \frac{R-G}{I_{\max}-I_{\min}} \frac{\pi}{3} + \frac{4}{3}\pi & (I_{\max}=B) \end{cases}$$

$$R = X \quad \left(h = H + \frac{2}{3}\pi \right)$$

$$G = X \quad (h = H)$$

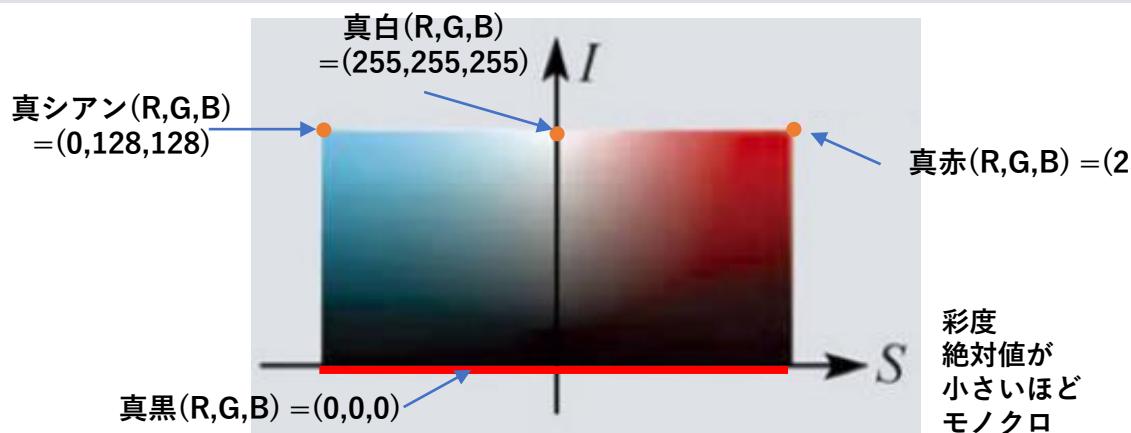
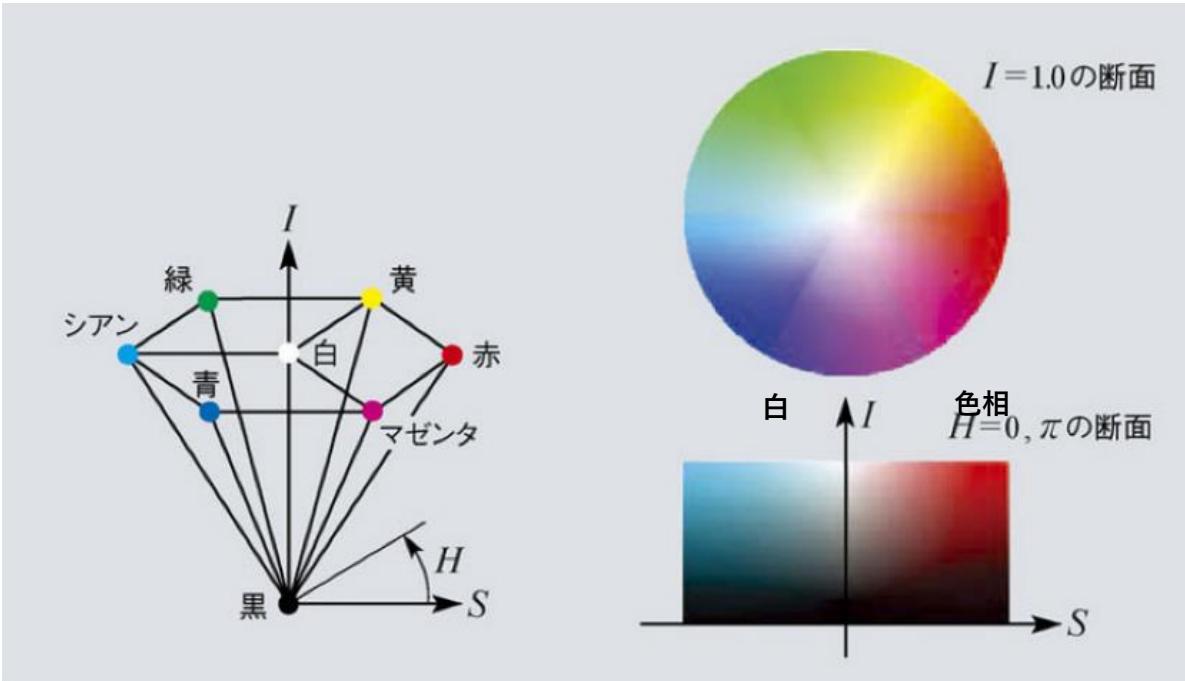
$$B = X \quad \left(h = H - \frac{2}{3}\pi \right)$$

$$\begin{aligned} h' &= \begin{cases} h & (h < 0) \\ h + 2\pi & (h > 2\pi) \end{cases} & M_2 &= \begin{cases} I(1+S) & (I \leq 0.5) \\ I + S - IS & (I > 0.5) \end{cases} \\ X &= \begin{cases} M_1 + (M_2 - M_1) \frac{3}{\pi} h' & \left(h' < \frac{\pi}{3}\right) \\ M_2 & \left(\frac{\pi}{3} \leq h' < \pi\right) \\ M_1 + (M_2 - M_1) \left(4 - \frac{3}{\pi} h'\right) & \left(\pi \leq h' < \frac{4}{3}\pi\right) \\ M_1 & \left(\frac{4}{3}\pi \leq h' \leq 2\pi\right) \end{cases} \\ M_1 &= 2I - M_2 \end{aligned}$$

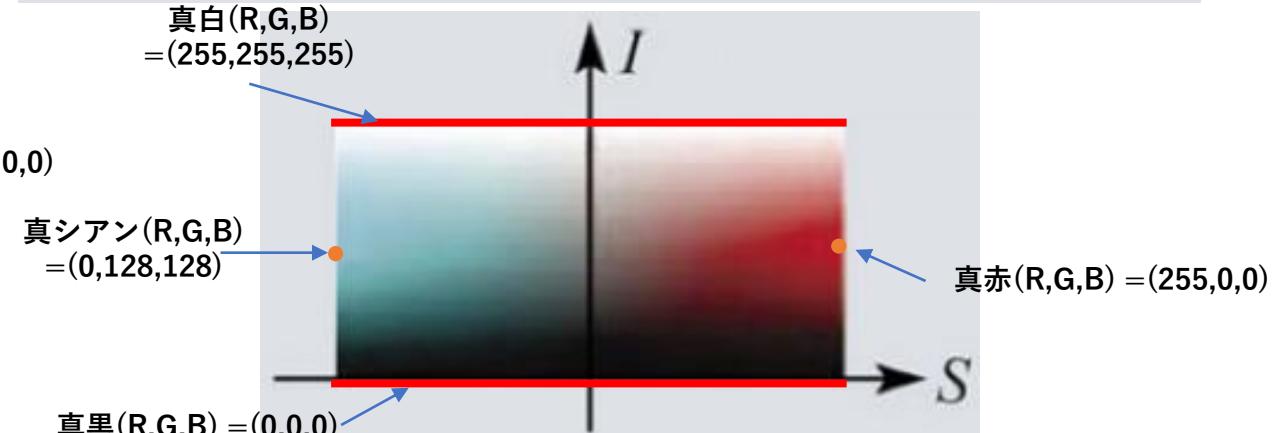
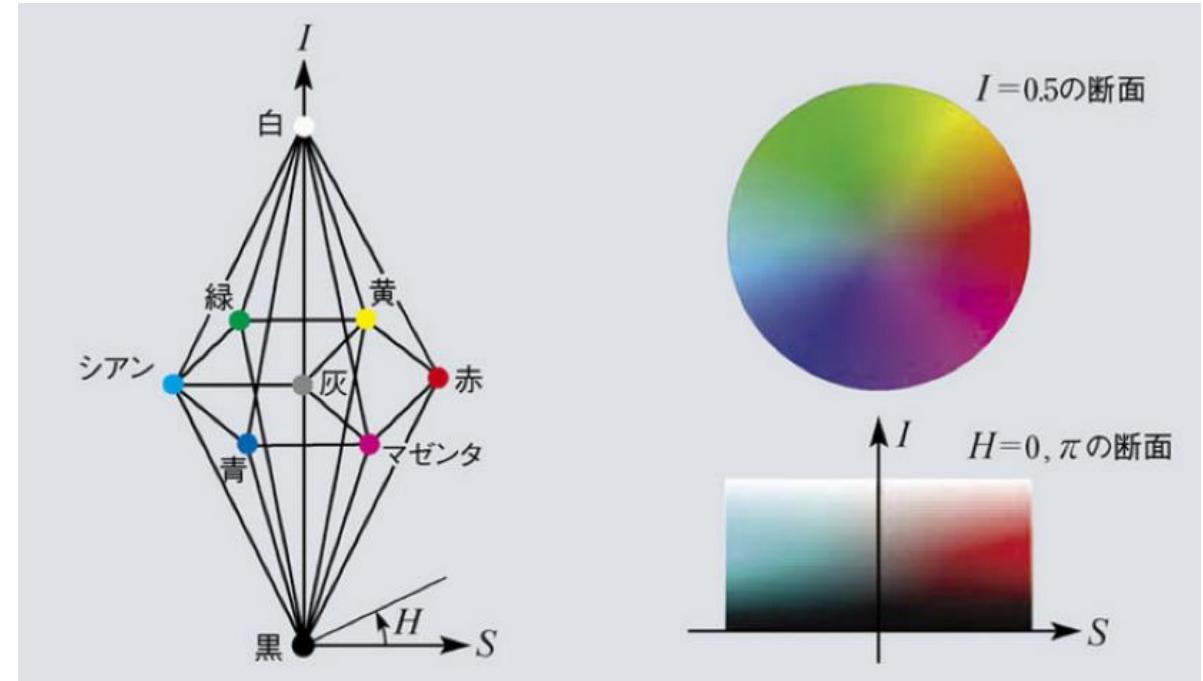
HSVとHSLの概念図

色相Hの変換は角度変換で表される。角度方向Rを基準としてRGBの順。
すなわち赤が角度0となり、黄、緑、シアン、青、マゼンタの順に 60° ずつ推移する

六角錐モデル



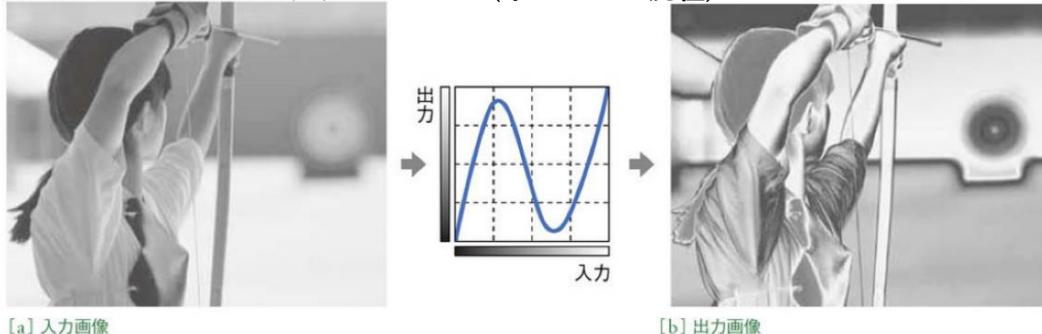
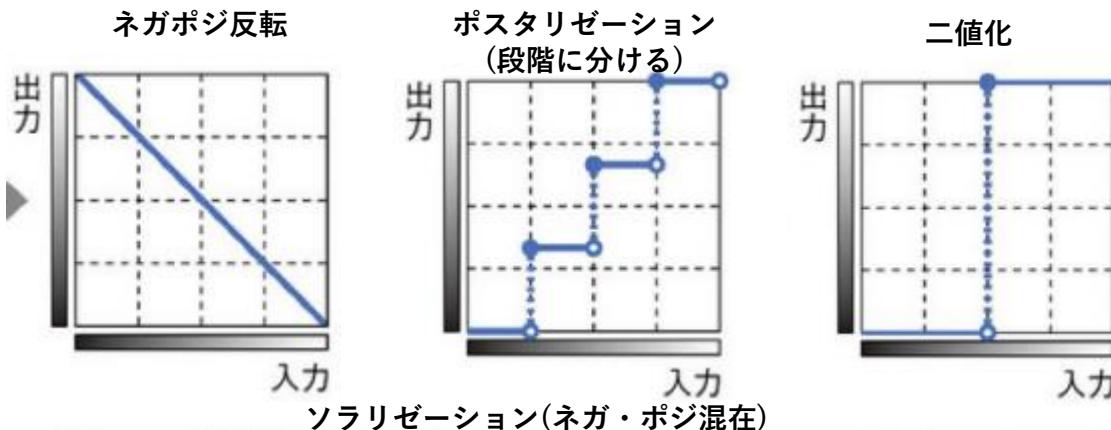
双六角錐モデル



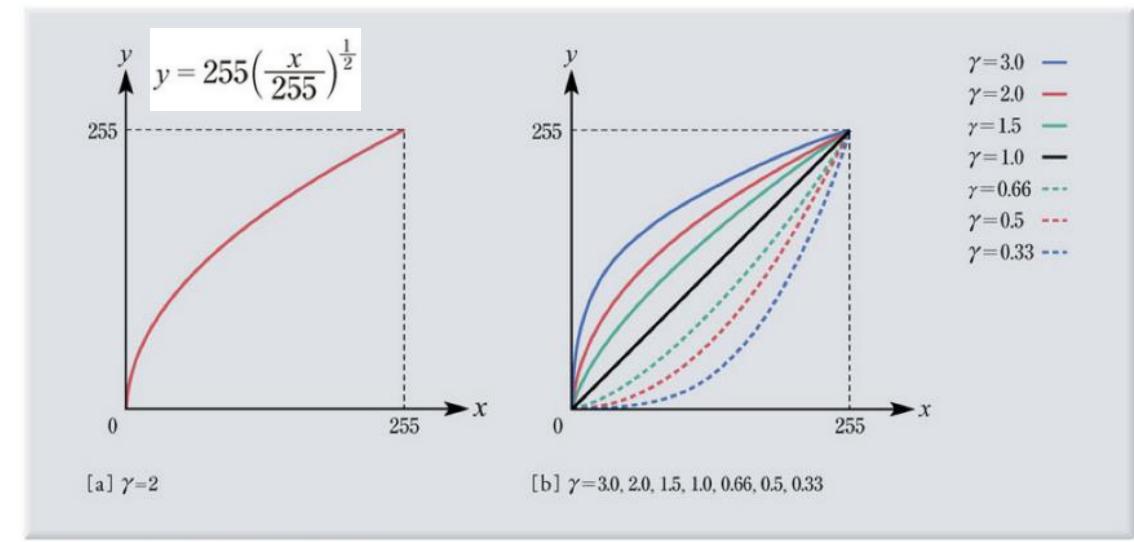
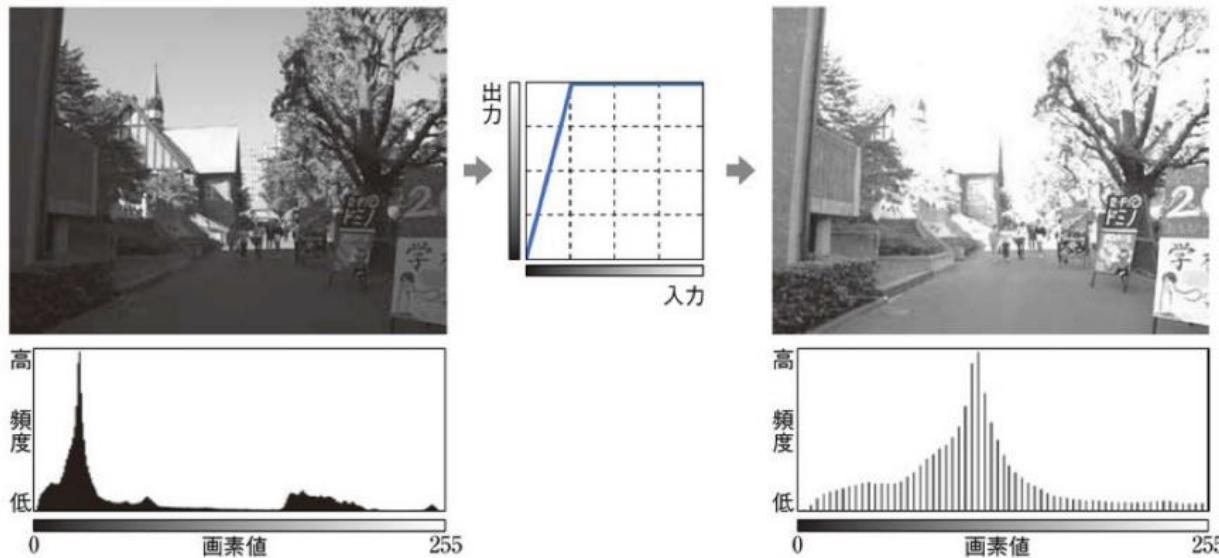
4.画素ごとの濃淡変換

トーンカーブ

- 階調変換関数…濃淡変換の入力と出力の関係
- トーンカーブ…階調変換関数をグラフ化したもの
 - S字…低い画素はより低く、高い画素はより高くしてコントラスト向上
 - 折れ線型…ヒストグラムの幅を広げたり、狭めたりして、コントラストを変える
 - ガンマ変換…折れ線よりも入力の情報を保ったまま、コントラストを変える



- ヒストグラム平坦化
→ ヒストグラムを見て適切な階調変換関数を探すのは手間なので、
0~255の各画素値に対して $M \times N / 256$ に近づくように画素値を変換して
コントラストを上げる。

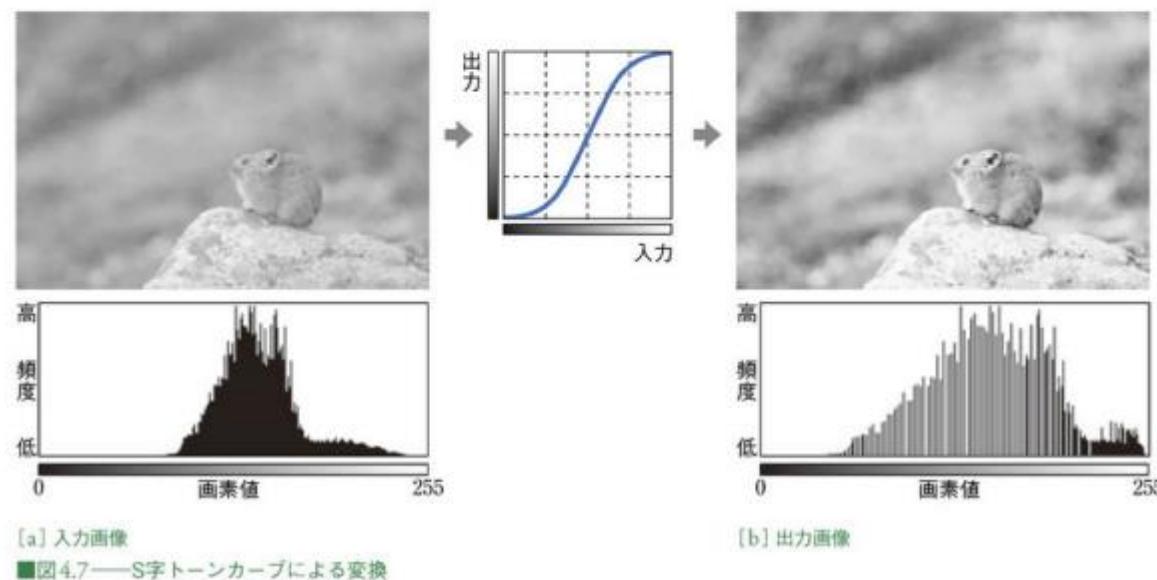


■図4.5——ガンマ変換のトーンカーブ

S字トーンカーブ

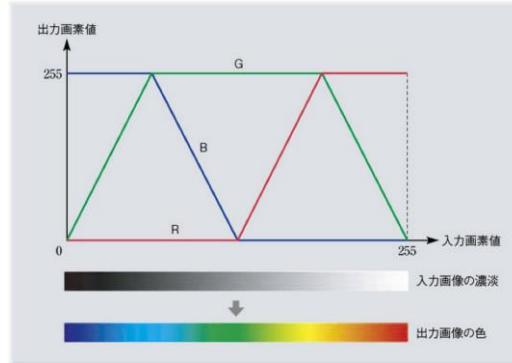
4-1-4 S字トーンカーブによる変換

図4.7は、S字型のトーンカーブを利用した変換を示している。入力画像では、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となっているが、出力画像では、変換により中間調が引き延ばされ、画像のコントラストが上がっていることがわかる。

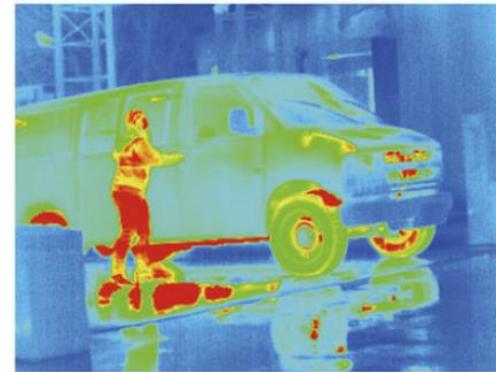


カラー画像

- 疑似カラー…グレースケールの画素値を三つのトーンカーブで変換し、区別がつきにくい微妙な違いを際立たせる。
- HSIの変換…RGBよりHSIの方が理解しやすいため、HSIに変換後それを調整することも多い



[a] 遠赤外線画像



[b] 擬似カラー画像

- 色補正…カメラや光源による色特性を補正し、CIE-XYZやsRGBの色を再現すること
→ 既知のsRGB値に近づくような変換行列を最小二乗法で求める

$$\begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} = \mathbf{M} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad E(\mathbf{M}) = \| \mathbf{Y} - \mathbf{MX} \|_F$$
$$\mathbf{M} = \mathbf{YX}^+ = \mathbf{YX}^\top (\mathbf{XX}^\top)^{-1}$$

\mathbf{M} が求めたい行列
 $E(\mathbf{M})$ を最小にする



X (3*24 行列)



Y (3*24 行列)



カラーチェッカー(24色)

[e] sRGB画像

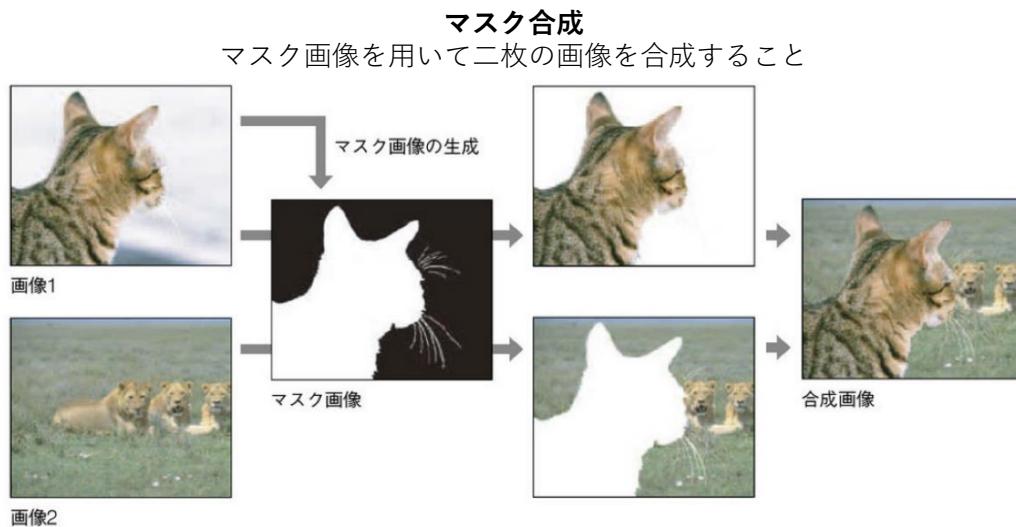
行列の差の最小二乗解

n 個の未知数に対する m 個の関係式があり、この全ての関係式を満たす未知数を求めたい。 $n \leq m$ であれば解は $\hat{\mathbf{x}} = \mathbf{A}^{-1}\mathbf{b}$ に一意に決まるが、 $n > m$ の時は最小化問題を解く。

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m \end{cases} \quad \mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix} \quad \mathbf{Ax} = \mathbf{b} \quad J(\mathbf{x}) = \|\mathbf{Ax} - \mathbf{b}\|^2$$

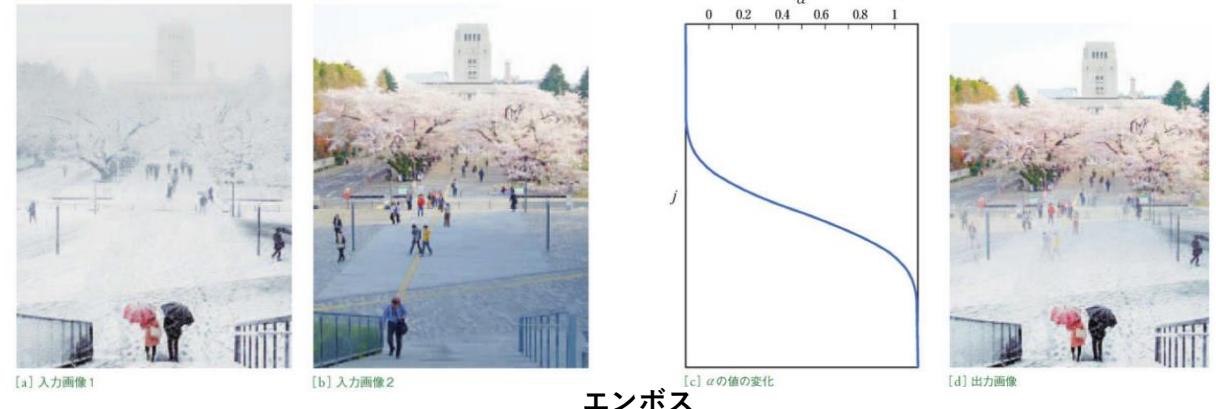
この時、最小二乗解は、 $\hat{\mathbf{x}} = \mathbf{A}^+ \mathbf{b}$ となり $\mathbf{A}^+ = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top$ は疑似逆行列と呼ばれる。

複数画像

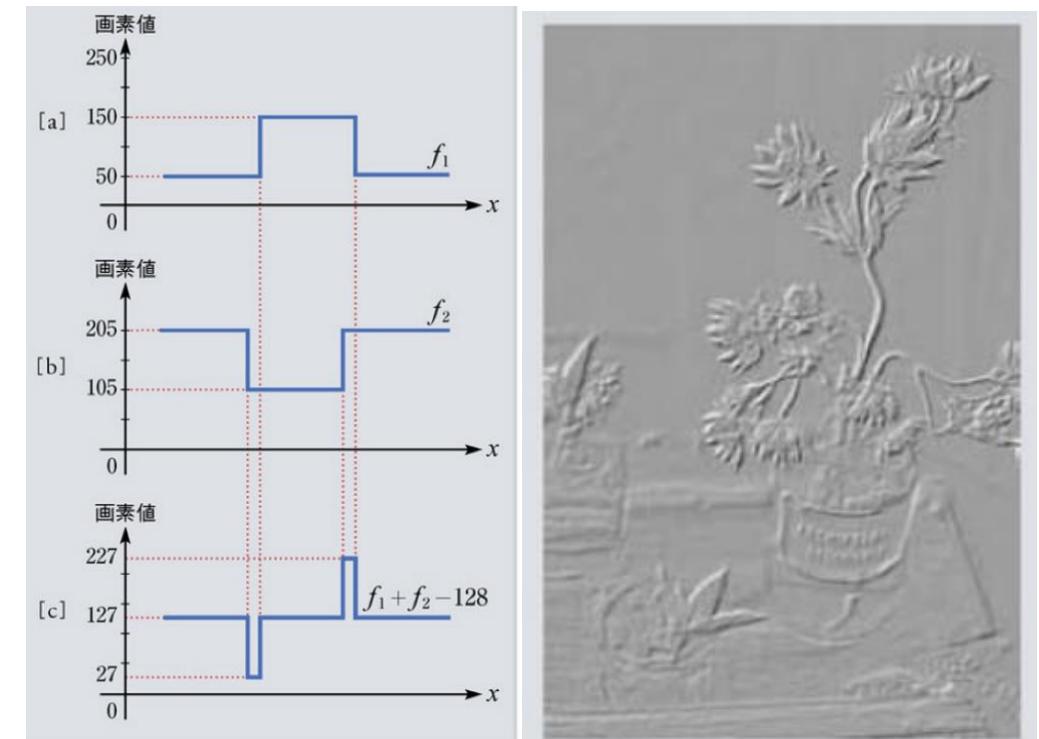


アルファブレンディング
二枚の画像の重み付き平均をとり二枚の画像を混ぜること。
動画像における α ブレンディング(アハ体験)はディジルブ又はオーバーラッピングと呼ばれる。

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



エンボス
ネガポジ反転×平行移動画像と元画像を合算し-128することによって
127を中心に浮彫調の画像を生成すること



5. 領域に基づく濃淡変換 (空間フィルタリング)

平滑化

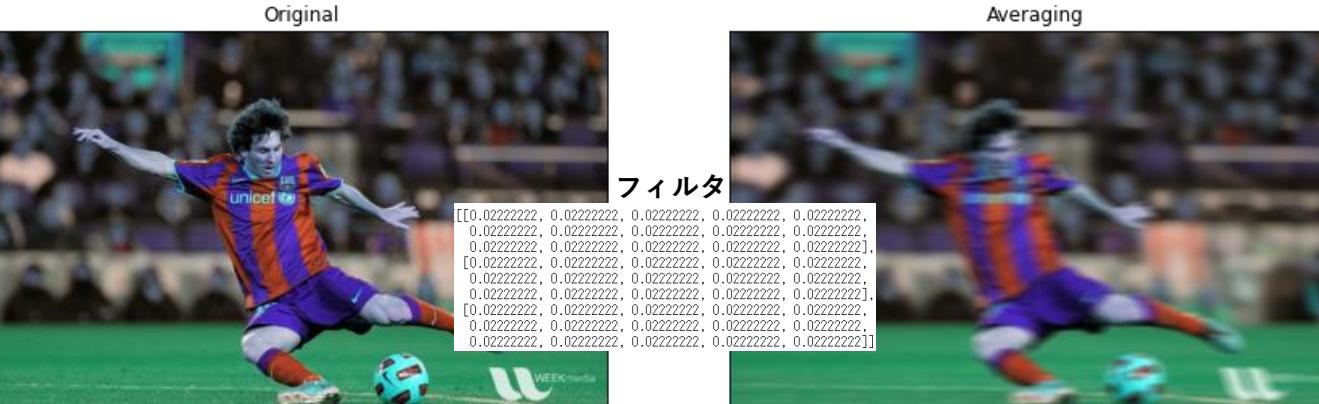
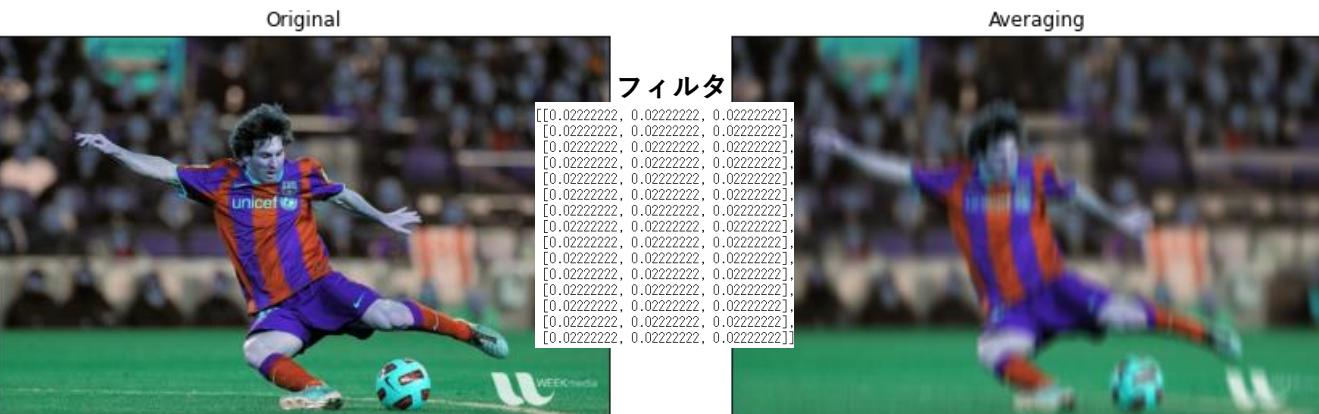
平滑化

ボケのような滑らかな濃淡変化を与える処理。前処理やノイズ軽減に用いられる。

平均化フィルタ

サイズが大きいほど平滑化の効果が大きい
フィルタが縦長の場合は縦方向、
横長の場合は横方向にボケる

$\frac{1}{25}$	$\frac{1}{25}$	$\frac{1}{25}$	$\frac{1}{25}$	$\frac{1}{25}$
$\frac{1}{25}$	$\frac{1}{25}$	$\frac{1}{25}$	$\frac{1}{25}$	$\frac{1}{25}$
$\frac{1}{25}$	$\frac{1}{25}$	$\frac{1}{25}$	$\frac{1}{25}$	$\frac{1}{25}$
$\frac{1}{25}$	$\frac{1}{25}$	$\frac{1}{25}$	$\frac{1}{25}$	$\frac{1}{25}$
$\frac{1}{25}$	$\frac{1}{25}$	$\frac{1}{25}$	$\frac{1}{25}$	$\frac{1}{25}$



重み付き平均化

加重平均化フィルタやガウシアンフィルタ。

ガウシアンフィルタは σ が大きいほど

平滑化の効果が大きくなる。

ガウシアンフィルタのように中央に重みをつけた
フィルタでは、元の値の付近を重視するので

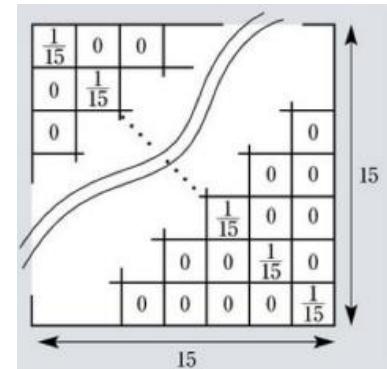
より滑らかで自然な平滑化になる。

$\frac{1}{256}$	$\frac{4}{256}$	$\frac{6}{256}$	$\frac{4}{256}$	$\frac{1}{256}$
$\frac{4}{256}$	$\frac{16}{256}$	$\frac{24}{256}$	$\frac{16}{256}$	$\frac{4}{256}$
$\frac{1}{16}$	$\frac{2}{16}$	$\frac{1}{16}$	$\frac{6}{256}$	$\frac{24}{256}$
$\frac{2}{16}$	$\frac{4}{16}$	$\frac{2}{16}$	$\frac{4}{256}$	$\frac{16}{256}$
$\frac{1}{16}$	$\frac{2}{16}$	$\frac{1}{16}$	$\frac{1}{256}$	$\frac{4}{256}$



[a] 入力画像

特定方向の平滑化



[b] フィルタリング結果

エッジ抽出(エッジ付近以外をゼロにする)

微分フィルタ

注目画素とその隣の画素との差を出力するフィルタ。x方向差分とy方向差分を要素を持つベクトルを**勾配**と呼び、勾配の大きさを画素値とすれば、あらゆる方向のエッジ抽出ができる。
しかし、同時にノイズも抽出してしまう。

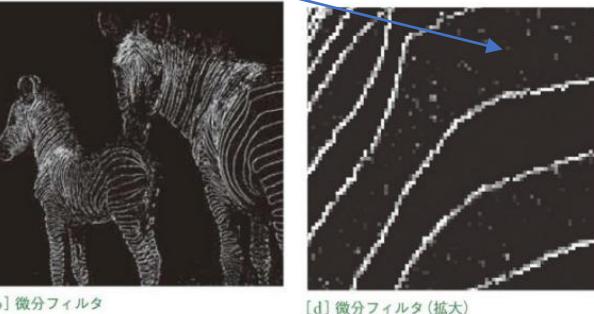
勾配 ($\Delta_x f(i, j), \Delta_y f(i, j)$)

大きさ $\sqrt{(\Delta_x f(i, j))^2 + (\Delta_y f(i, j))^2}$

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 \\ -\frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 0 \end{bmatrix}$$



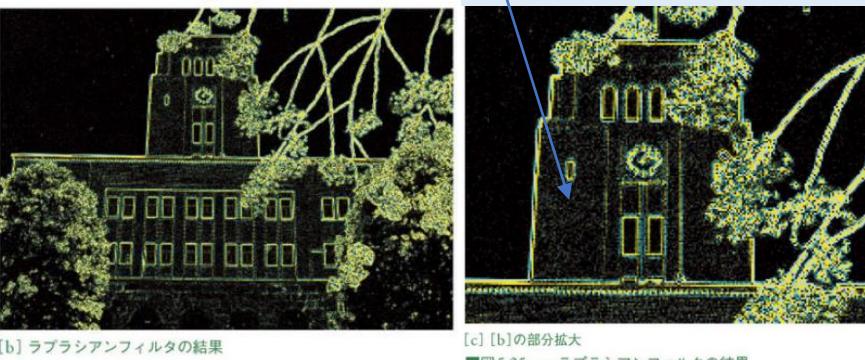
[a] 入力画像

二次微分

微分を二回繰り返す処理。一次微分によって得た画素位置は注目画素から半画素ずれているのに対し、二次微分は注目画素の位置に相当する。

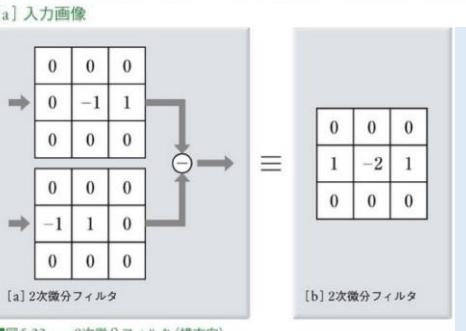
ラプラシアンフィルタ

縦横両方向の二次微分の和。エッジを強く強調し、方向にも依存しないが、ノイズも強調してしまう

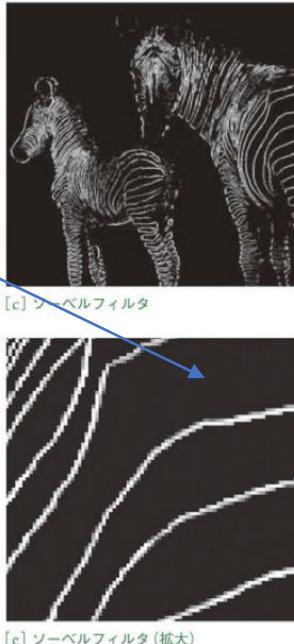
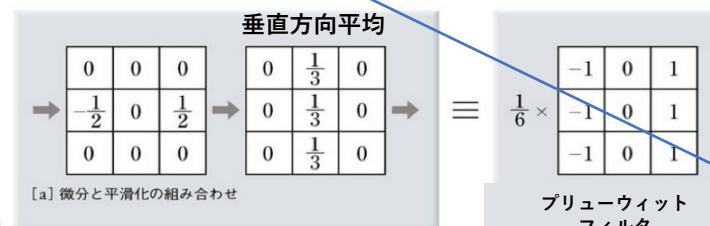


DoGフィルタ(Difference of Gaussian)
LoGを二つのガウシアンの差で近似

$$h_g(x, y; \sigma_1) - h_g(x, y; \sigma_2)$$



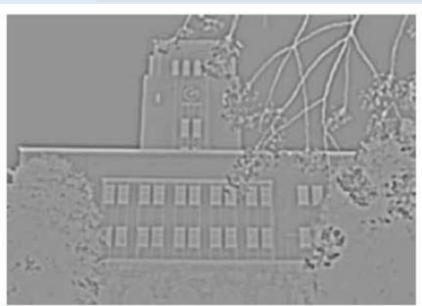
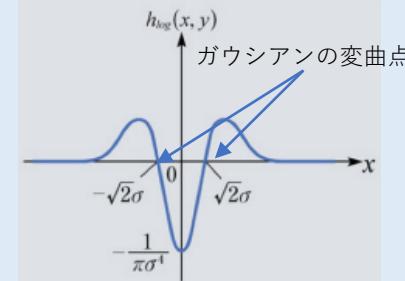
微分と平滑化の組み合わせ
ノイズを低減しながらエッジ抽出



LoGフィルタ(Laplacian of Gaussian)

$$h_{log}(x, y) = \frac{x^2 + y^2 - 2\sigma^2}{2\pi\sigma^6} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

二次元ガウシアンとラプラシアンを
ひとまとめにしたもの。
エッジ上で大きい値、それ以外ではほぼ0になる



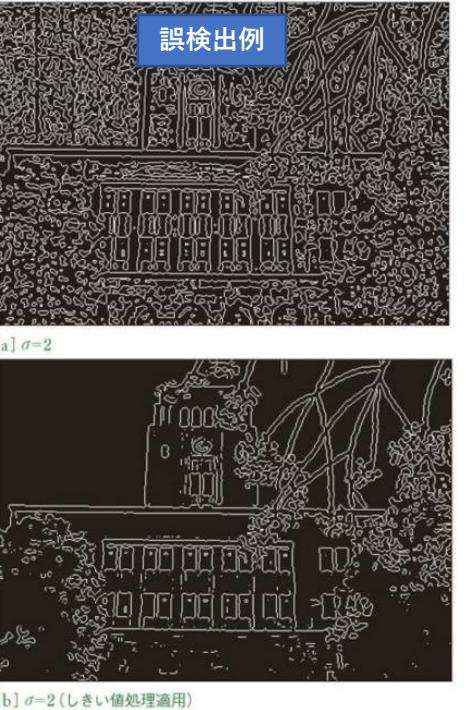
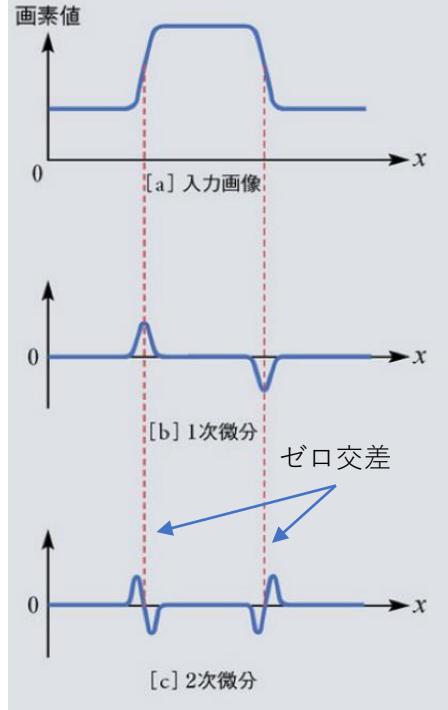
[d] $\sigma=8$

[b] $\sigma=2$

エッジ検出(エッジ位置を1ピクセル単位で特定)

LoGフィルタのゼロ交差法

小さな濃淡変化をエッジとして誤検出しないように、勾配が閾値 σ 以下のものを除去し、ゼロ交差となる位置をエッジとする



ケニーのエッジ検出法

ノイズ低減(ガウシアンフィルタ)

↓
エッジ抽出(ソーベルフィルタ)

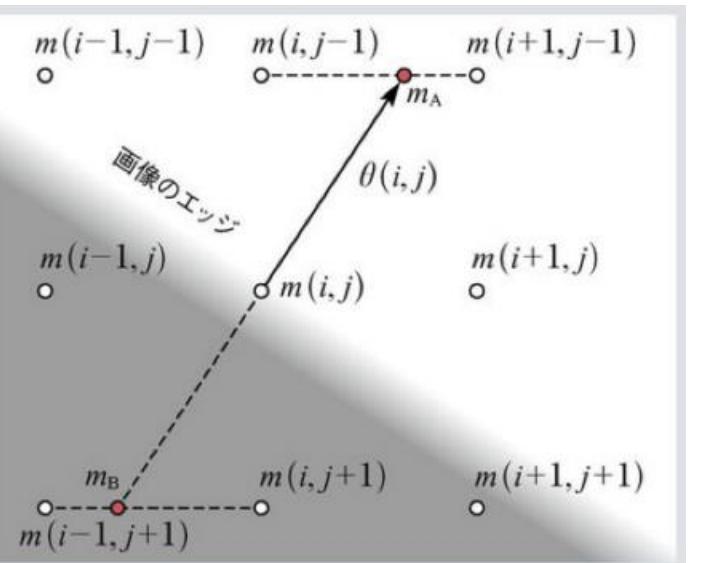
↓
|勾配|最大位置をエッジ候補とする

↓
エッジ候補のうち、
勾配が Th_{high} よりも高い画素はエッジ
 Th_{low} よりも低い画素は非エッジ

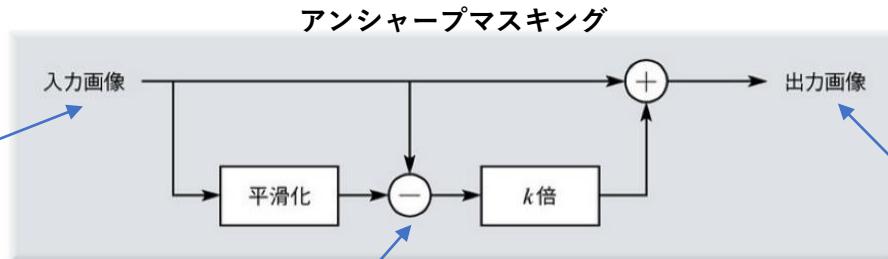
その間の画素はエッジと接していればエッジ
閾値を二段階にするとエッジがとぎれとぎれになりづらい

エッジ特定方法

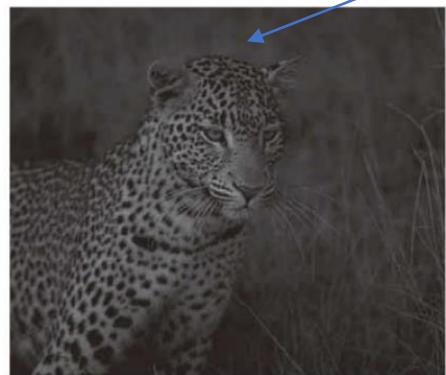
|勾配|が大きいとエッジの可能性も大きいが、
単に|勾配|が閾値以上の画素をエッジとすると、
濃淡の変化が幅を持つ場合に、エッジも幅を持ってしまう。従って、勾配方向 θ に沿って
|勾配|が最大の位置をエッジ候補とする。



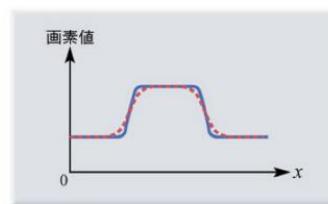
鮮鋭化(元の画像の濃淡を残したままエッジを強調)



■図5.32——アンシャープマスキングの流れ



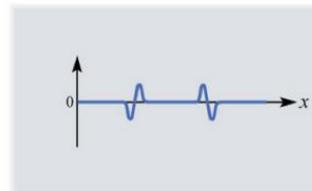
[a] 入力画像



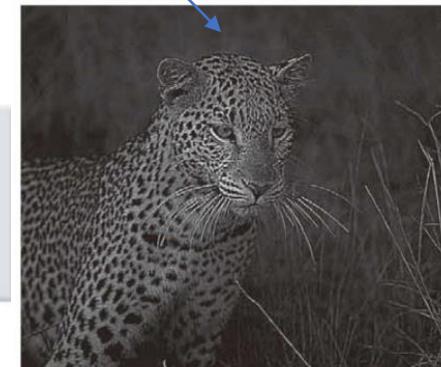
[a] 入力画像(実線)とその平滑化(点線)



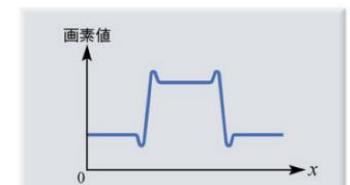
[b] 入力画像から平滑化した画像を引き算した画像



[b] [a]の実線から点線を引いたもの



[c] 鮮鋭化された画像



[c] [a]の実線に対し、[b]を定数倍して足したもの

■図5.33——アンシャープマスキングによる画像の鮮鋭化
マスキングのようす

鮮鋭化フィルタ

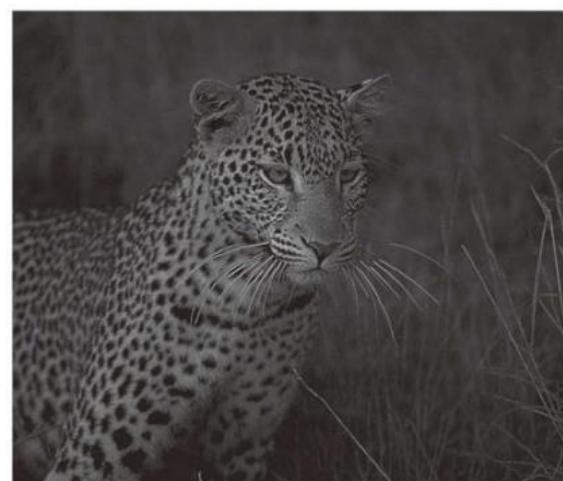
アンシャープマスキングを一つのフィルタで表現

$$I_5 + k(I_5 - \frac{1}{9}(I_1 + I_2 + I_3 + I_4 + I_5 + I_6 + I_7 + I_8 + I_9))$$

$$(1+k)I_5 - \frac{k}{9}(I_1 + I_2 + I_3 + I_4 + I_5 + I_6 + I_7 + I_8 + I_9)$$

$$(1 + \frac{8}{9}k)I_5 - \frac{k}{9}(I_1 + I_2 + I_3 + I_4 + I_6 + I_7 + I_8 + I_9)$$

$\frac{k}{9}$	$-\frac{k}{9}$	$-\frac{k}{9}$
$-\frac{k}{9}$	$1 + \frac{8}{9}k$	$-\frac{k}{9}$
$\frac{k}{9}$	$-\frac{1}{9}$	$\frac{k}{9}$



[b] $k=4$



[c] $k=9$



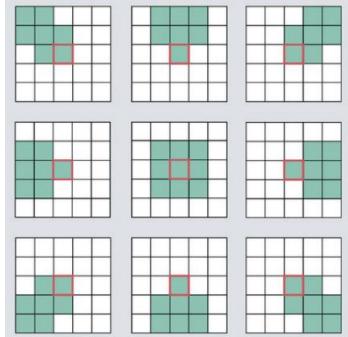
[d] $k=18$

エッジを保存した平滑化(全て非線形フィルタ)

平均化によりエッジがなまる=エッジの角の画素値を小さくしてしまっている→エッジを保存して平滑化するためにはエッジの角もなるべくエッジに近い値にする

最小分散法(仮)

9通りの局所領域の画素値の分散が
最小となる領域で平均をとる



K最近隣平均化フィルタ

近隣のk個の画素値が近い画素だけで平均をとる

バイラテラルフィルタ

注目画素からの距離と画素値の差の両方に応じて
ガウス分布に従う重みをつけた平均化
差が大きいほど小さくなる。

$$g(i, j) = \frac{\sum_{n=-W}^W \sum_{m=-W}^W w(i, j, m, n) f(i+m, j+n)}{\sum_{n=-W}^W \sum_{m=-W}^W w(i, j, m, n)}$$

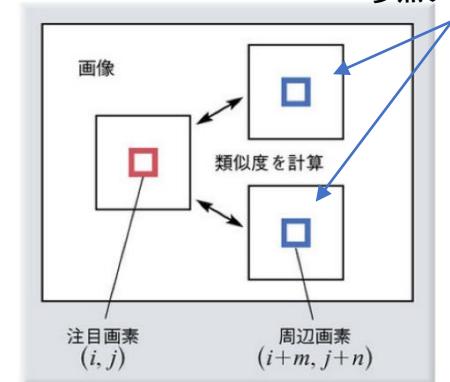
$$w(i, j, m, n) = \exp\left(-\frac{m^2 + n^2}{2\sigma_1^2}\right) \exp\left(-\frac{(f(i, j) - f(i+m, j+n))^2}{2\sigma_2^2}\right)$$

ノンローカルミーンフィルタ

注目画素(赤□)を中心としたブロックと
周辺の同サイズの参照ブロックとの間の
類似度を求める。その類似度に応じて、
参照ブロックの中心画素(青□)を加重平均したもの。

これにより、注目画素がエッジ上の場合は、
少し離れた同一エッジの値に近似できるため、高精
度でエッジを保存した平滑化が可能。

$$w(i, j, m, n) = \exp\left(-\frac{\sum_{l=-W}^W \sum_{s=-W}^W (f(i+s, j+l) - f(i+m+s, j+n+l))^2}{2\sigma_3^2}\right)$$



メディアンフィルタ
スパイク状ノイズに強い。



[a] 入力画像



[c] バイラテラルフィルタの結果



[b] 平均化フィルタの結果



[d] ノンローカルミーンフィルタの結果



[b] 平均化フィルタの結果



[c] メディアンフィルタの結果



6.周波数領域における フィルタリング

周波数フィルタリング

ローパスフィルタによる平滑化

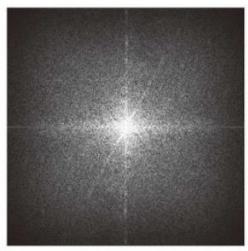
フィルタの値がある特定の周波数を境に急激に変化するため不自然な濃淡が表れる。



[a] 入力画像



[e] 出力画像



[b] [a]の振幅スペクトル



[c] ローパスフィルタ



[d] フィルタリング後の振幅スペクトル



[a] 入力画像



[b] 出力画像1



[c] 出力画像2



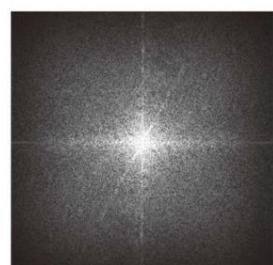
ハイパスフィルタによるエッジ抽出



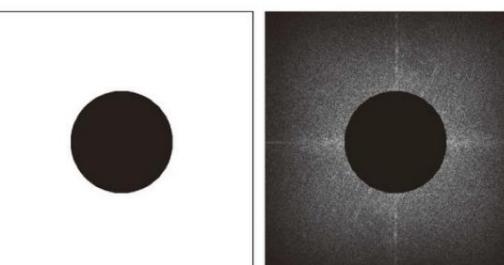
[a] 入力画像



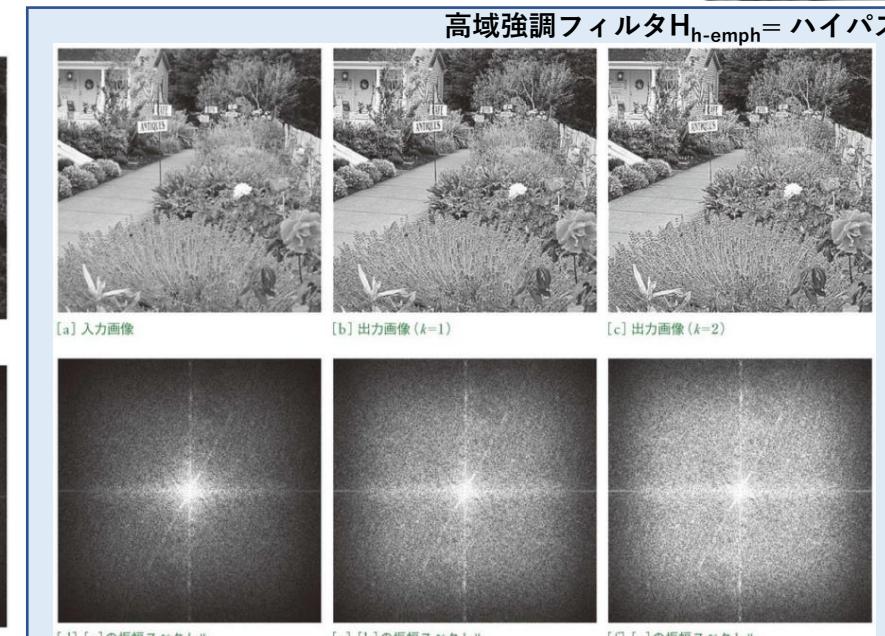
[c] 出力画像



[b] [a]の振幅スペクトル



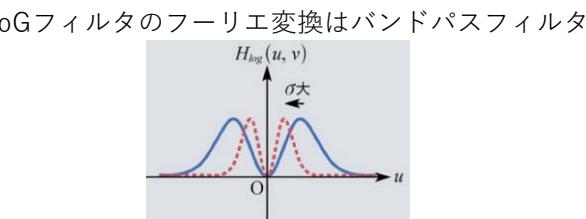
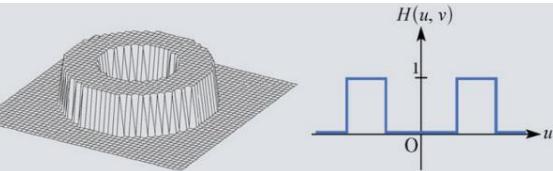
[c] ハイパスフィルタ



空間フィルタリングと周波数フィルタリングの関係
畠み込み積分のフーリエ変換=フーリエ変換の積



バンドパスフィルタ(u, v)
中間的な周波数のみ残すフィルタ

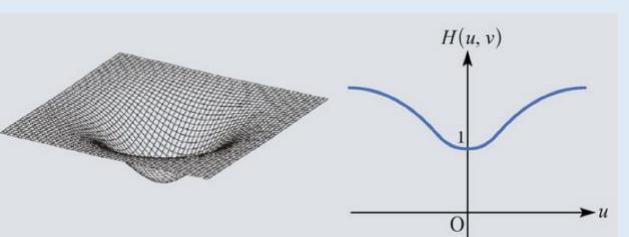


$$H_{high}(u, v) = 1 - H_{low}(u, v)$$

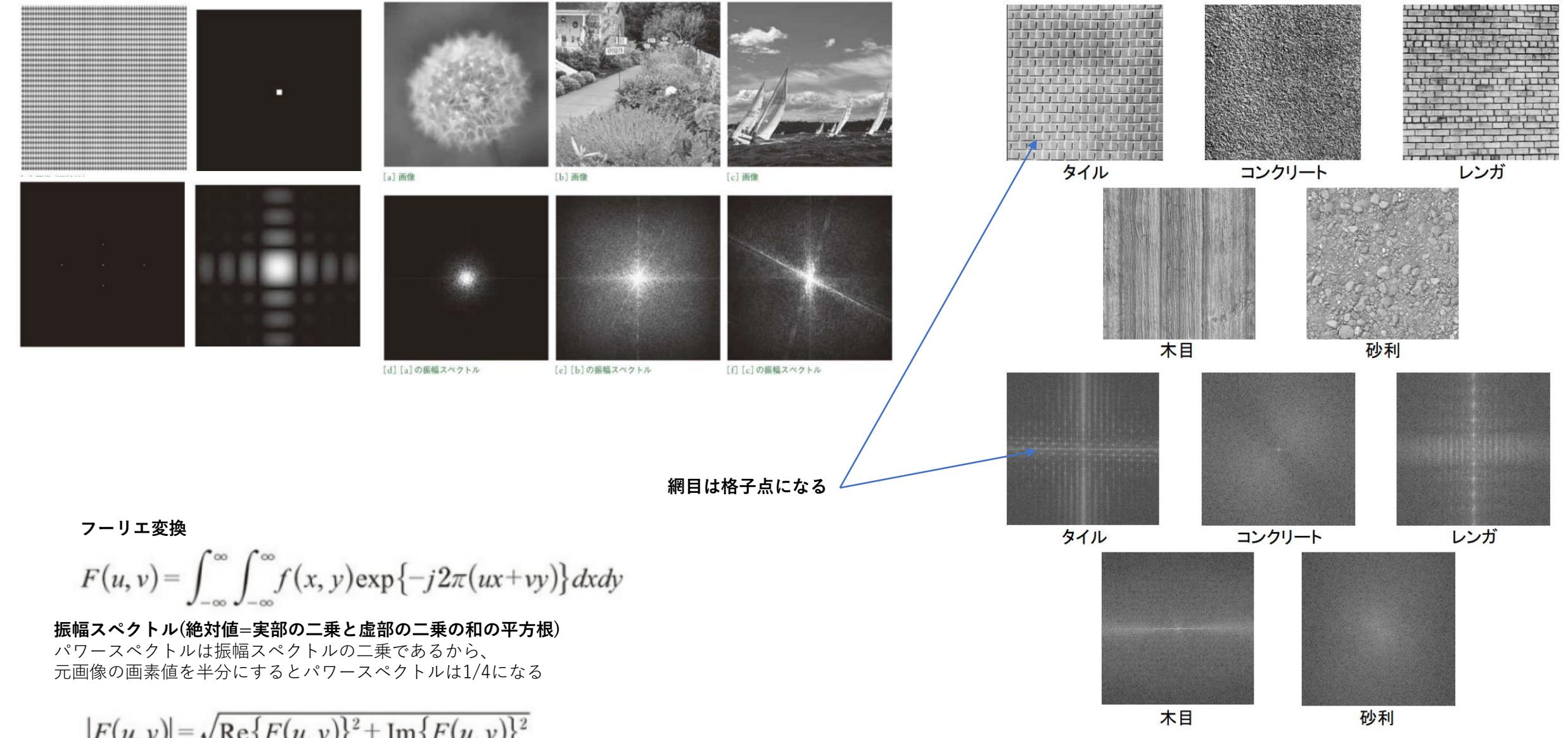
$$H_{h\text{-emph}}(u, v) = 1 + kH_{high}(u, v)$$

$$H_{h\text{-emph}}(u, v) = k + 1 - kH_{low}(u, v)$$

ハイパスフィルタは直流成分($u=v=0$ の成分)を含む低周波成分を除去するため、暗くなる。高域強調フィルタは、低周波成分を保ちつつ、高周波を強調できる。



パワースペクトルからわかること



7.画像の復元と生成

ボケ・ブレ画像の復元

ボケの点拡がり関数 $h(x,y)$ のモデル化

ボケは方向に依らず一定であるから、
ボケのPSFは原点からの距離のみに
依存した関数になるはずという
観点でモデル化。代表的なものは
二次元ガウス分布とローパスフィルタ。
デルタ関数は画像が劣化しない、
理想的な点拡がり関数

二次元ガウス分布

σ はボケ量の大きさを表す

$$h(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{r^2}{2\sigma^2}\right)$$

ローパスフィルタ

$$h(x, y) = \begin{cases} \frac{1}{\pi R^2} & (x^2+y^2 \leq R^2) \\ 0 & (x^2+y^2 > R^2) \end{cases}$$

逆フィルタ

畳み込み結果のフーリエ変換はフーリエ変換の積となる。

劣化画像 g のフーリエ変換 G に **フィルタ h のフーリエ変換 H の逆数**(逆フィルタ)をかけば、
原画像 f のフーリエ変換 F が得られる。

$$K_{inv}(u, v) = \frac{1}{H(u, v)}$$

$$K_{inv}(u, v)G(u, v) = \frac{1}{H(u, v)}F(u, v)H(u, v) = F(u, v)$$

しかし、実際には劣化画像にはノイズも含まれる。

$$g(x, y) = f(x, y) * h(x, y) + n(x, y)$$

$$G(u, v) = F(u, v)H(u, v) + N(u, v)$$

H が 0 や 0 に近い場合にそのノイズ成分が増幅されてしまうという欠点がある。

$$K_{inv}(u, v)G(u, v) = \frac{1}{H(u, v)}F(u, v)H(u, v) + \frac{1}{H(u, v)}N(u, v)$$

ウェーナー フィルタ

ノイズを含んだ劣化画像を復元するため、原画像との誤差が最小になるよう求められたフィルタ。

$$K_w(u, v) = \frac{1}{H(u, v)} \frac{|H(u, v)|^2}{|H(u, v)|^2 + |N(u, v)|^2 / |F(u, v)|^2}$$

一般に f や n は未知であるため、分母の第二項を適当な定数 Γ で代用することが多い。

Γ の小ささは、復元度とノイズ増幅度に影響する。

$$K_w(u, v) = \frac{1}{H(u, v)} \frac{|H(u, v)|^2}{|H(u, v)|^2 + \Gamma}$$

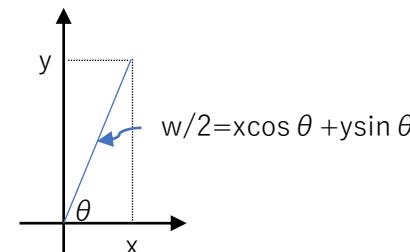
$$g(x, y) = f(x, y) * h(x, y) \quad G(u, v) = F(u, v)H(u, v)$$

ブレの点拡がり関数 $h(x, y)$ のモデル化

カメラが θ の方向に等速直線運動していく、
幅 w に 1 次元で伸びていると仮定したモデル。

$$h_\theta = \begin{cases} \frac{1}{w} & (|x \cos \theta + y \sin \theta| \leq \frac{w}{2} \text{かつ}, x \sin \theta - y \cos \theta = 0) \\ 0 & (\text{それ以外}) \end{cases}$$

$w=5, \theta=45^\circ$ の場合



0	0	0	0	1/5
0	0	0	1/5	0
0	0	1/5	0	0
0	1/5	0	0	0
1/5	0	0	0	0

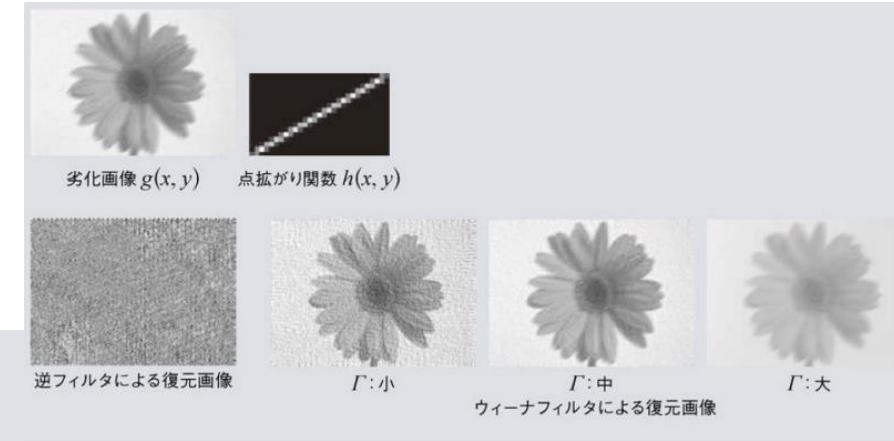


図 7.5——カメラぶれによる劣化画像からの画像復元

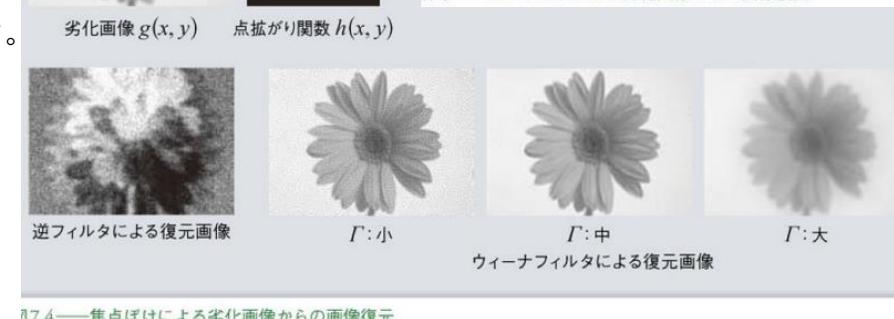


図 7.4——焦点ぼけによる劣化画像からの画像復元

モデルパラメータ σ 、 θ 、 w の推定
これらのモデルのフーリエ変換は、
周期的に 0 になる。

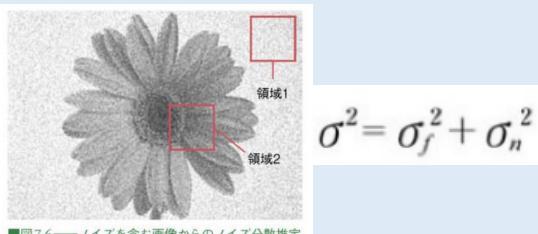
従って劣化画像のフーリエ変換 $G(u, v)$ も
同様であることから推定。

ノイズ除去・超解像・ダイナミックレンジ

ノイズ除去手法

分散推定

テクスチャがない領域の分散=ノイズの分散
より、ノイズの大きさを推定



$$\sigma^2 = \sigma_f^2 + \sigma_n^2$$

■図7.6——ノイズを含む画像からのノイズ分散推定

フィルタ

ガウシアン、バイラテラル

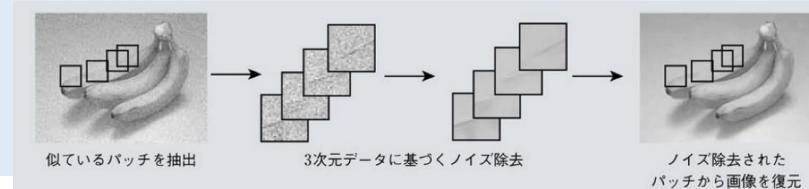
滑らかさ拘束付き最小二乗法

ノイズは滑らかではない、すなわち高周波成分であるから、高周波成分が多いほど大きくなる関数をコスト関数としてそれを最小化する原画像を推定。

$$E = \frac{1}{\sigma^2} \iint (g(x, y) - f(x, y))^2 dx dy + \lambda_f \rho_f(f(x, y))$$

$$\rho_f(f(x, y)) = \iint |d(x, y) * f(x, y)|^2 dx dy \quad d(x, y) = \text{ハイパスフィルタ}$$

BM3D(Block Matching and 3D filtering)



画像超解像

画素数の少ない(高周波成分を含まない)画像から
画素数の多い(高周波数成分を含む)画像を復元すること。

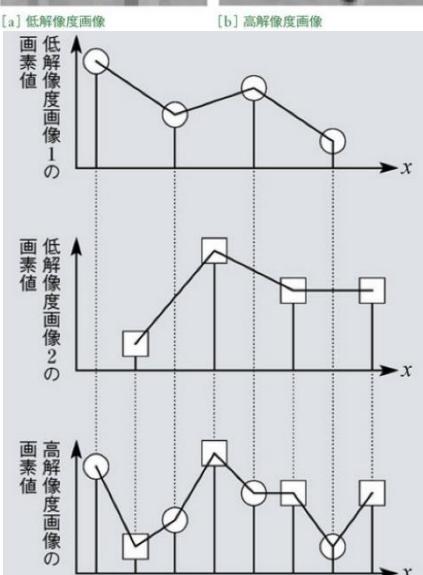
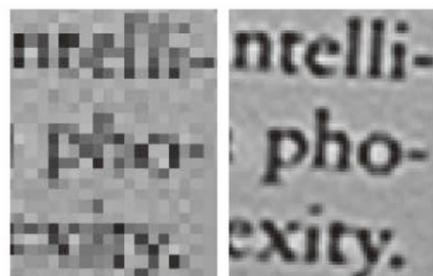
事例ベース

あらかじめ学習しておいた低解像度画像と
高解像度画像の対応関係に基づいて推定。

複数フレーム

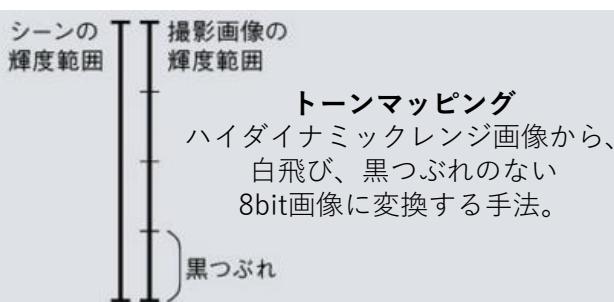
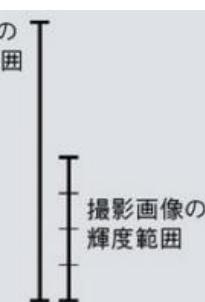
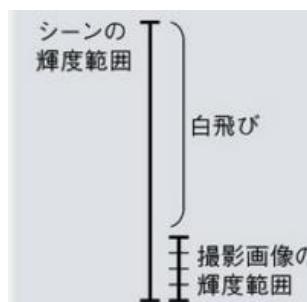
少しずつ位置をずらした複数の低解像度画像から
高解像度画像を復元する方法。
k番目の画素値の行列 \mathbf{g}_k と高解像度画像から
 \mathbf{g}_k と同じ位置の成分を抜き出した行列 $\mathbf{H}\mathbf{W}_k\mathbf{f}$
の差の最小化によって求める。
第二項は拘束を表す。

$$E = \sum_{k=1}^N \frac{1}{\sigma_k^2} \|\mathbf{g}_k - \mathbf{H}\mathbf{W}_k\mathbf{f}\|^2 + \lambda_f \rho_f(\mathbf{f})$$



ダイナミックレンジ

観測可能な明るさの範囲。明るい領域と暗い領域が含まれる画像では
ダイナミックレンジが不足し、飽和したり疑似輪郭が生じたりする。
それを避けるため、シーンの輝度範囲が異なるダイナミックレンジの狭い複数の画像からダ
イナミックレンジの広い画像、すなわちハイダイナミックレンジ画像を復元する。



露出設定 大



露出設定 中



露出設定 小



トーンマッピングされた
ハイダイナミックレンジ
合成画像

ガイド画像・勾配を利用した復元

フラッシュなしの画像a=ノイズが大きいが色表現は正しい→入力画像
フラッシュありの画像b=ノイズが少ないが色表現が正しくない→ガイド画像
ガイド画像bを利用して入力画像aからノイズを除去することを考える。

ジョイントバイラテラルフィルタ

バイラテラルフィルタで、重み計算時にガイド画像の画素値を利用しノイズ除去。
ノイズが少ないなど条件のよいガイド画像を利用することでより精度が高まる。

ガイデットフィルタ

入力画像fを目的変数、ガイド画像hを説明変数とした線形回帰。一度に画像全体に対して回帰を行うのは困難のため、小領域ごとに行つたあと平均をとる。ノイズはランダムだから回帰の出力はノイズを含まない値に近づくはず。

$$f(i,j) \approx a \cdot h(i,j) + b$$

$$g(i,j) = \bar{a}_{i,j} \cdot h(i,j) + \bar{b}_{i,j}$$

$$E_{i,j}(a_{i,j}, b_{i,j})$$

$$= \sum^W \sum^W (f(i+n, j+m) - a_{i,j}h(i+n, j+m) - b_{i,j})^2 + \epsilon a_{i,j}^2$$

$$\bar{a}_{i,j} = \frac{1}{(2W+1)^2} \sum_{n=-W}^W \sum_{m=-W}^W a_{i+n, j+m}$$

$$\bar{b}_{i,j} = \frac{1}{(2W+1)^2} \sum_{n=-W}^W \sum_{m=-W}^W b_{i+n, j+m}$$



暗部補正

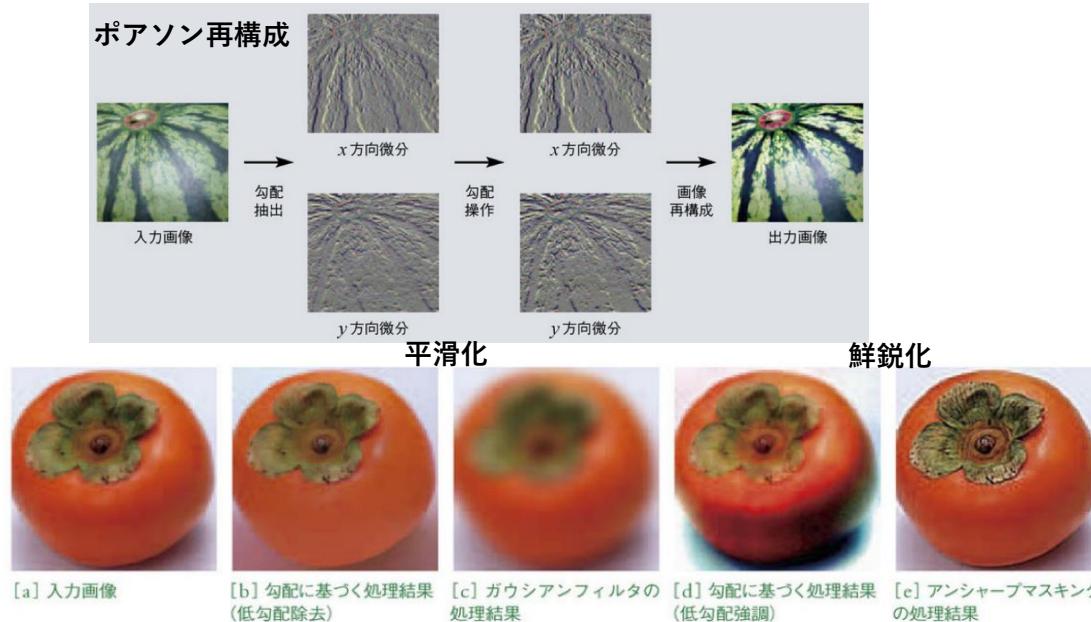


暗部の勾配を強調することで、画像のコントラストを上げられる。
ヒストグラム平坦化よりも元の雰囲気を維持したまま補正できる。

ポアソンブレンディング

処理前後の勾配の差の最小化問題(ポアソン方程式)を解くことで得られる

$$E = \int \int \phi(\nabla f(x, y), \mathbf{G}(x, y)) dx dy \quad \phi(\nabla f(x, y), \mathbf{G}(x, y)) = \frac{1}{2} \|\nabla f(x, y) - \mathbf{G}(x, y)\|^2$$



低勾配除去

小さい勾配のみ0にすることでエッジ(高勾配)を保存した平滑化が可能

低勾配強調

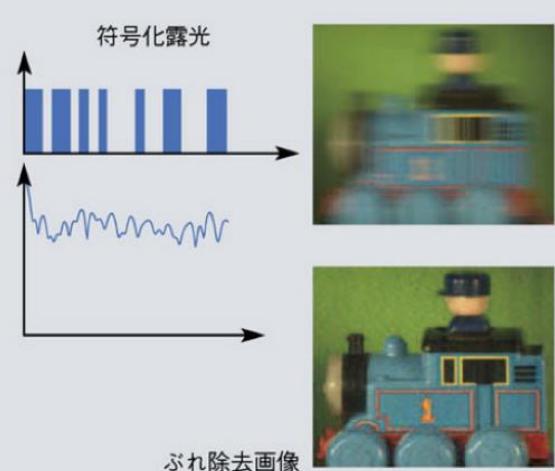
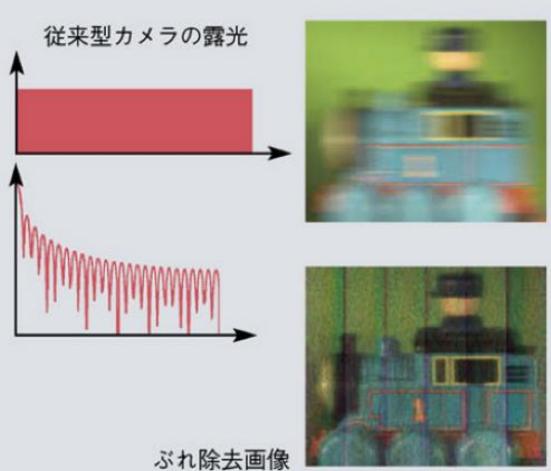
エッジなどの高勾配部分は変わらず、陰影などの低勾配のみが強調されている。



[c] 勾配に基づく処理結果
(ポアソンブレンディング)

コンピューターショナルフォトグラフィ(CP)

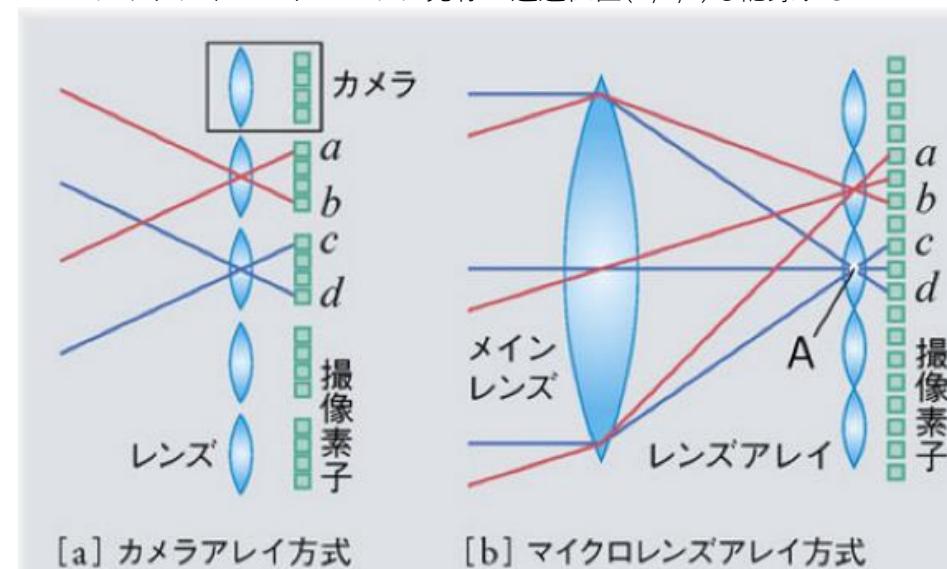
- 従来の画像復元…光が撮像素子に入った後の情報を工夫する
- CP…撮像素子に入る前まで含めて工夫する
- リフォーカス…ライトフィールドカメラで撮影した情報から、3次元情報を復元し、ピントの合った画像を生成
- 符号化露光…露光時間中複数回シャッタを開閉することで、より時間に対する光の分布情報を失わない撮影方法
- 符号化開口…X,Y,Zに対する符号化 → 絞り付近に部分的なマスクを設置し、積分への重みづけができる。



$$\text{プレノプティック関数 } I = P(X, Y, Z, \theta, \phi, \lambda, t)$$

ライトフィールドを満たす光の分布関数

ライトフィールドカメラ
従来のカメラは平行光線を一点に集めてしまうため、レンズのどこを通過した光かわからない
ライトフィールドカメラは光線の通過位置(X,Y,Z)も記録する



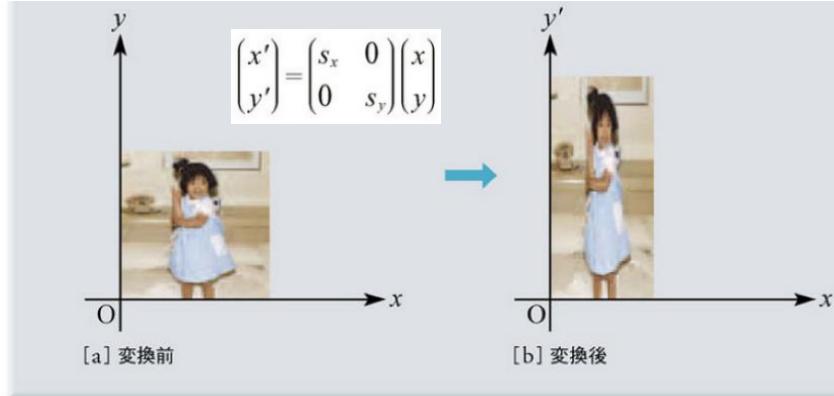
■図7.24——符号化開口の例¹²⁾ (提供: 兵庫県立大学 日浦慎作)

絞り部分に光を部分的に遮断するマスクを挿入することで、ピント外れによるぼけの除去処理を安定化することができる。

8.幾何学的変換

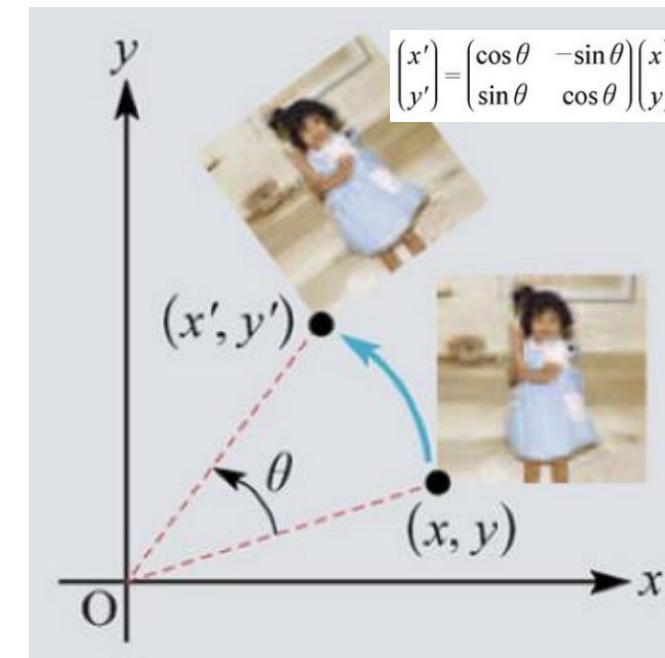
線形変換

拡大・縮小



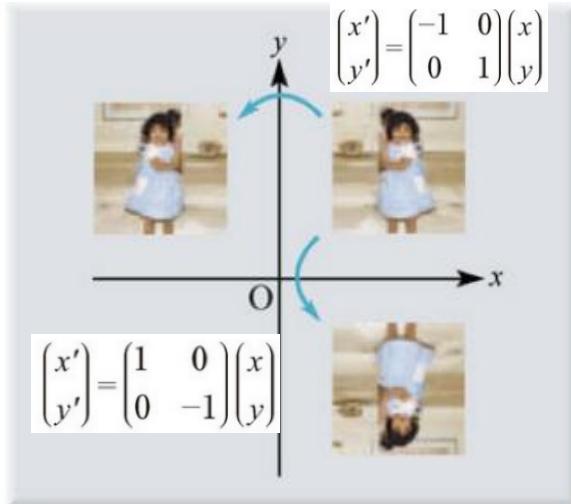
■図8.3——拡大・縮小 ($s_x=0.75, s_y=1.5$)

回転

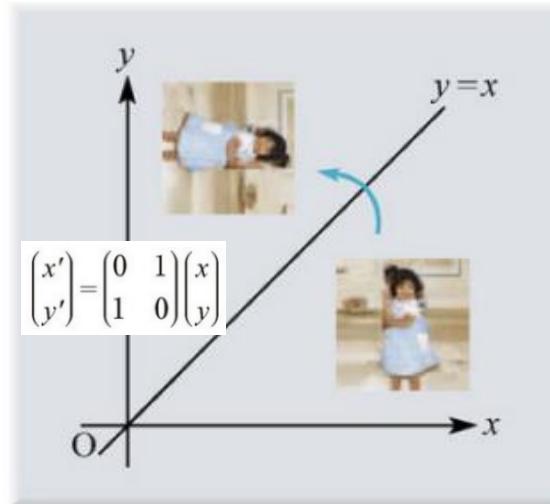


スキュー

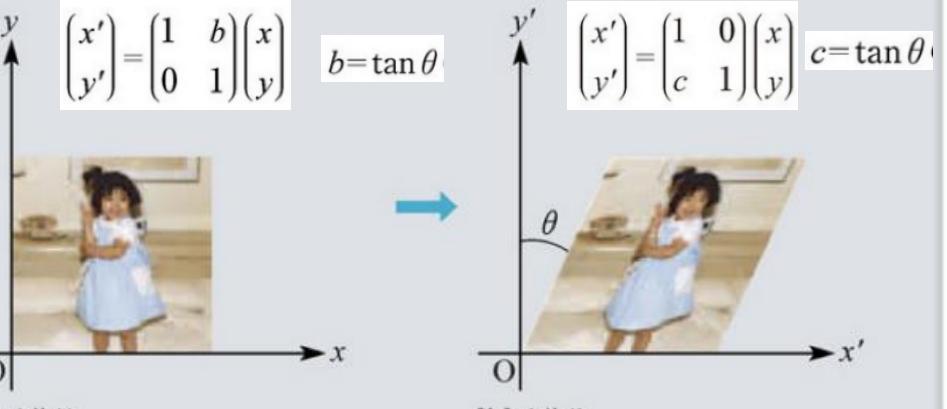
長方形を傾けて平行四辺形にする



■図8.5——鏡映 (x 軸, y 軸)



■図8.6——鏡映 ($y=x$)



■図8.7——スキュー (x 軸方向)

非線形変換

同次座標(x,y,1)

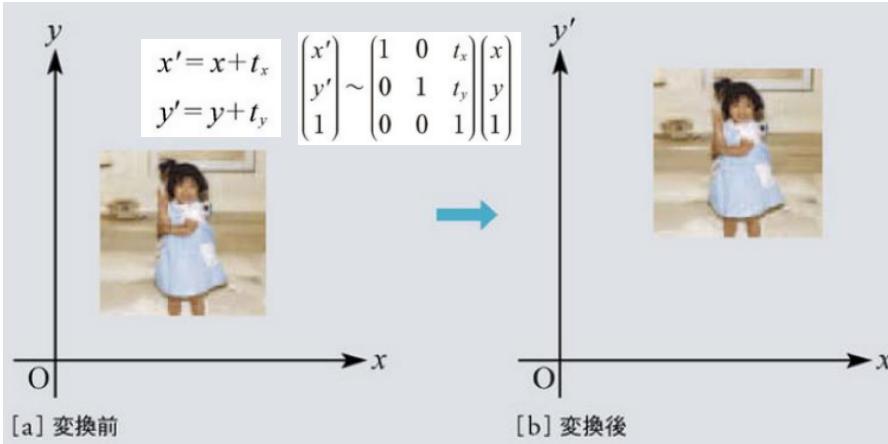
平行移動を線形変換行列で表現するために変換された座標
同次座標(ξ_1, ξ_2, ξ_3)は以下を満たす。

$$x = \frac{\xi_1}{\xi_3} \quad y = \frac{\xi_2}{\xi_3}$$

変換のパラメータはこれと複数の対応点座標より求める。
同次座標は、定数倍しても上の式を満たす
従って、変換行列も定数倍しても同じ変換を意味する

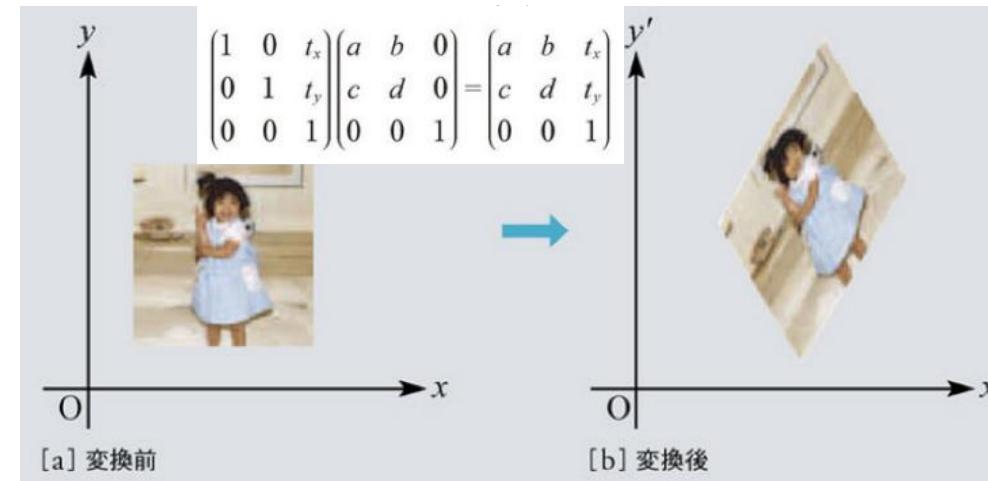
平行移動

線形変換では表せないため、同次座標で表現



アフィン変換

任意の線形変換と平行移動を組み合わせた変換
任意の平行四辺形を別の任意の平行四辺形に移す



ユークリッド変換

回転×平行移動

形や大きさ一定のまま、任意の位置に移動

$$\begin{pmatrix} \cos \theta & -\sin \theta & t_x \\ \sin \theta & \cos \theta & t_y \\ 0 & 0 & 1 \end{pmatrix}$$

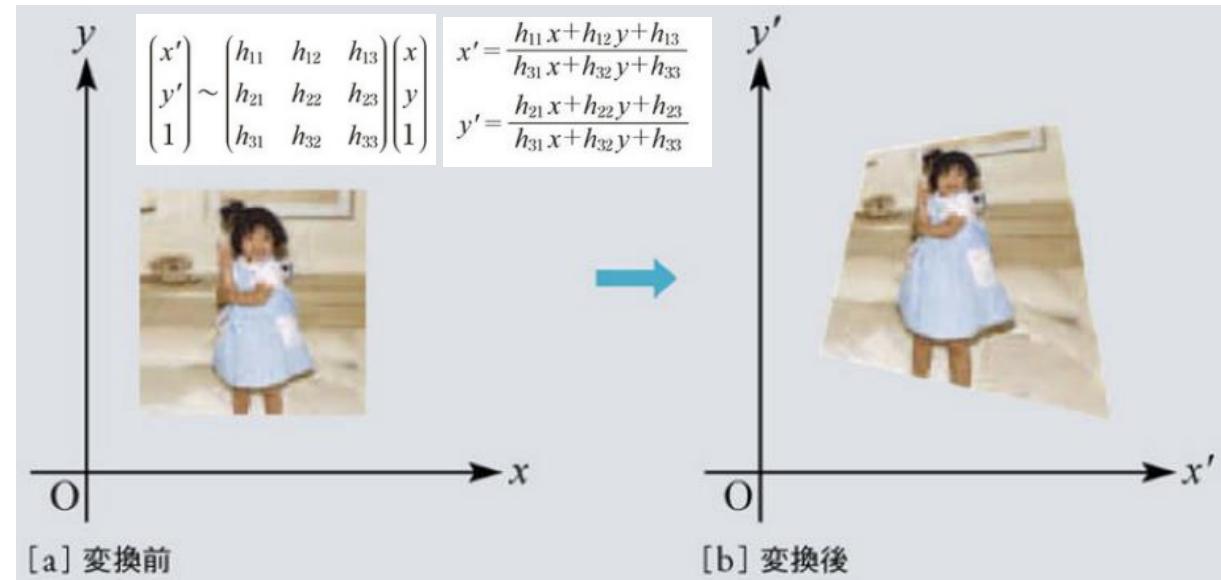
相似変換

拡大・縮小×回転×平行移動

↓
拡大・縮小×ユークリッド変換

射影変換

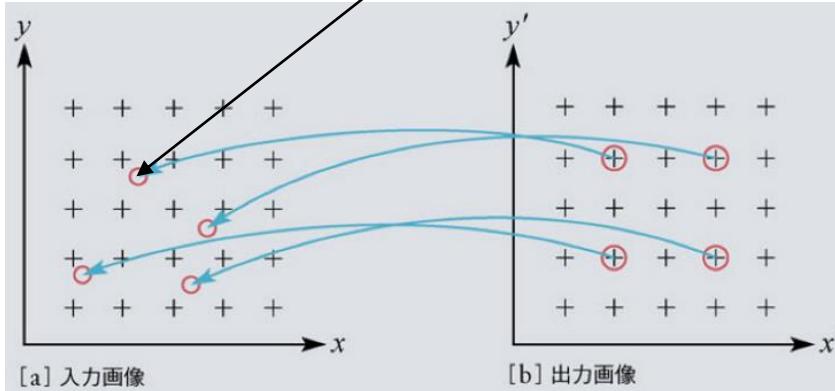
一般的な変換。任意の四角形を任意の四角形に移す変換
変換行列を h_{33} で割れば1になるので、未知パラメータは9つではなく8つ



画像の再標本化と補間

画像の再標本化

幾何学的変換をすると標本化位置からずれるため、画素位置を補正すること。具体的には、出力画像に逆変換を施した場合の入力画像上の画素値を補間によって求める。



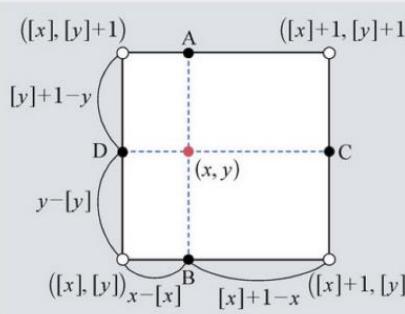
ニアレストネイバー

最も近い画素値を利用(四捨五入)

$$I(x, y) = f([x+0.5], [y+0.5])$$

バイリニア補間

最近4点の距離重み付き平均

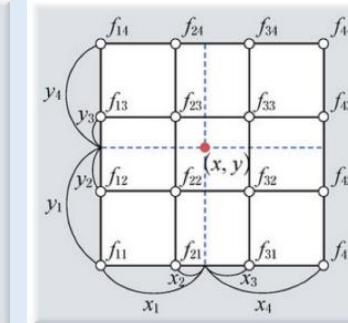


補間方法

バイキュービック補間

周辺16点と三次多項式から導出

$$I(x, y) = (h(x_1) \ h(x_2) \ h(x_3) \ h(x_4)) \begin{pmatrix} f_{11} & f_{12} & f_{13} & f_{14} \\ f_{21} & f_{22} & f_{23} & f_{24} \\ f_{31} & f_{32} & f_{33} & f_{34} \\ f_{41} & f_{42} & f_{43} & f_{44} \end{pmatrix} \begin{pmatrix} h(y_1) \\ h(y_2) \\ h(y_3) \\ h(y_4) \end{pmatrix}$$



$$h(t) = \begin{cases} |t|^3 - 2|t|^2 + 1 & (|t| \leq 1) \\ -|t|^3 + 5|t|^2 - 8|t| + 4 & (1 < |t| \leq 2) \\ 0 & (2 < |t|) \end{cases}$$

ただし、

$$x_1 = 1+x-[x]$$

$$x_2 = x-[x]$$

$$x_3 = [x]+1-x$$

$$x_4 = [x]+2-x$$

$$y_1 = 1+y-[y]$$

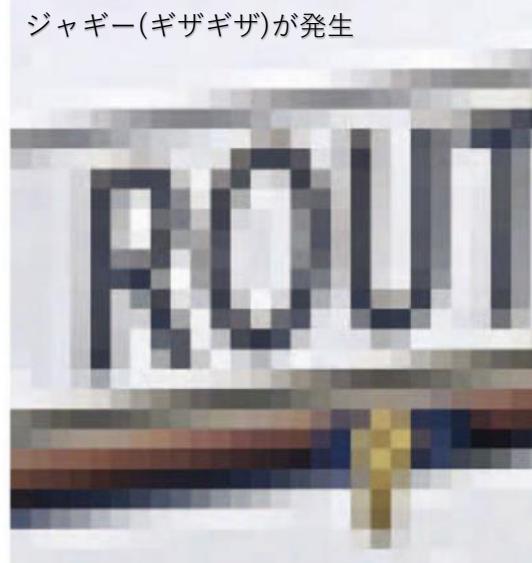
$$y_2 = y-[y]$$

$$y_3 = [y]+1-y$$

$$y_4 = [y]+2-y$$

演算量 バイキュービック>バイリニア>ニアレストネイバー

ジャギー(ギザギザ)が発生



エッジがなまる



最も自然



イメージモザイキング

イメージモザイキング

カメラを動かし撮影した複数の画像を繋ぎ合わせ、
広角かつ高解像度の画像を得ること

- 特徴点の検出とマッチング(画像間で対応させる、SIFT)
- つなぎあわせるための幾何学的変換(射影変換行列)の推定

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} \sim \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}} \\ y' = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}}$$

→ 射影変換行列は定数倍の不定性があるため h_{33} を1とする

$$xh_{11} + yh_{12} + h_{13} - xx'h_{31} - x'yh_{32} = x'$$

$$xh_{21} + yh_{22} + h_{23} - xy'h_{31} - yy'h_{32} = y'$$

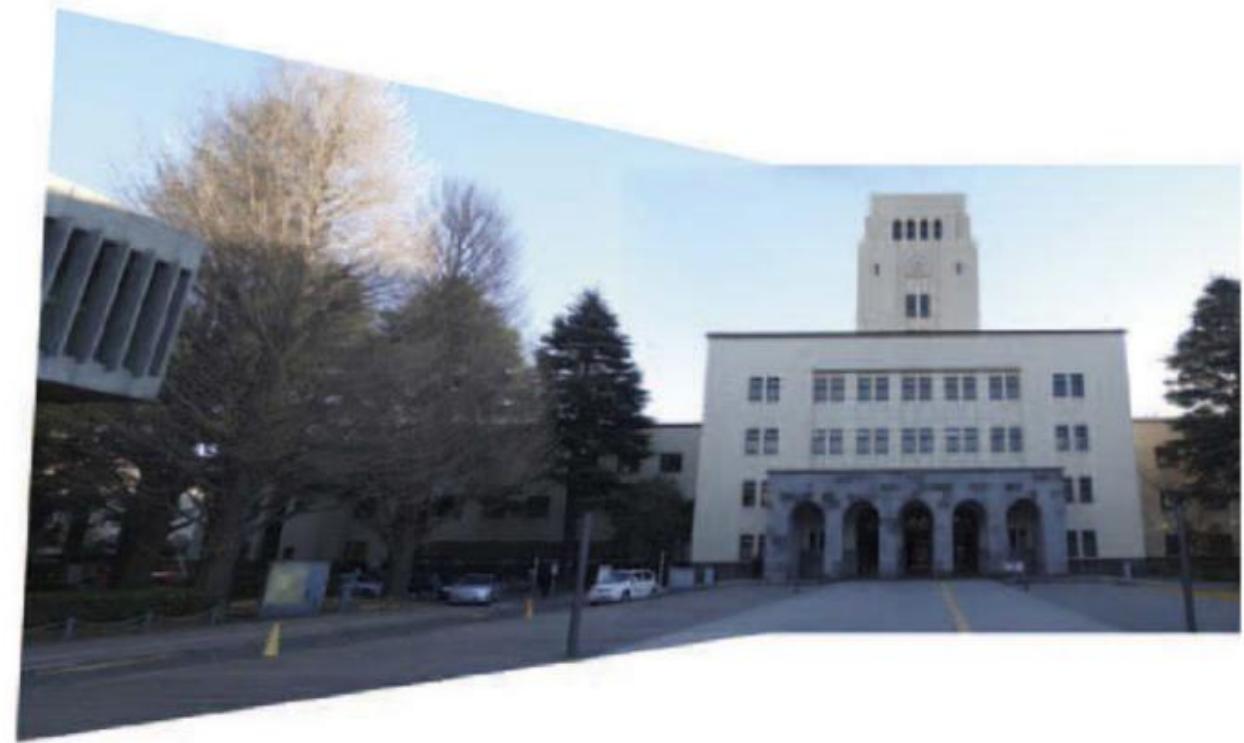
→ 線形変換の未知パラメータは8個なので、特徴点の対応の組みが4組以上あれば、計8つの方程式が得られ、解は一意に決まる。ない場合は疑似逆行列を用いて最小二乗解。

$$A\mathbf{h} = \mathbf{b}$$

$$\mathbf{h} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}$$

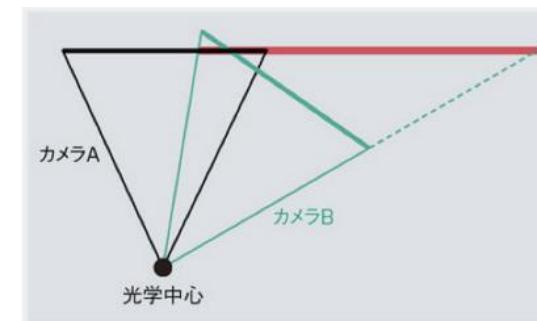
$$A = \begin{pmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1x'_1 & -x'_1y_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1y'_1 & -y_1y'_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2x'_2 & -x'_2y_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -x_2y'_2 & -y_2y'_2 \\ \vdots & & & & & & & \end{pmatrix} \quad \mathbf{h} = \begin{pmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} x'_1 \\ y'_1 \\ x'_2 \\ y'_2 \\ \vdots \end{pmatrix}$$

- 求めた射影変換を用いて変換し、 α ブレンディングでつなぎめを目立たなくさせて合成

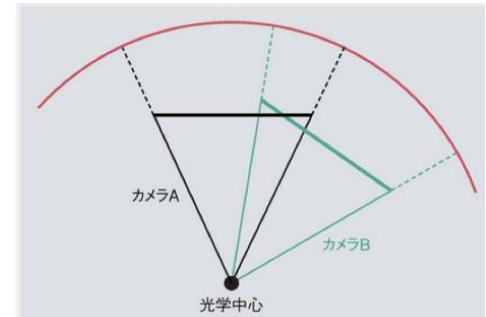


平面パノラマ

幾何学的変換によって平面に投影された
対象物の画像を再現すること。



円筒面パノラマ (球面パノラマ)

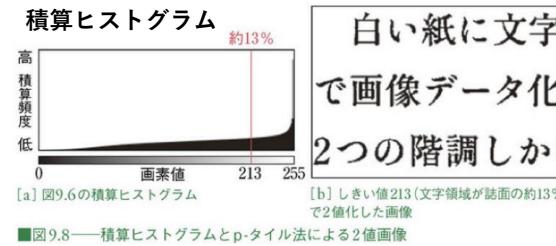
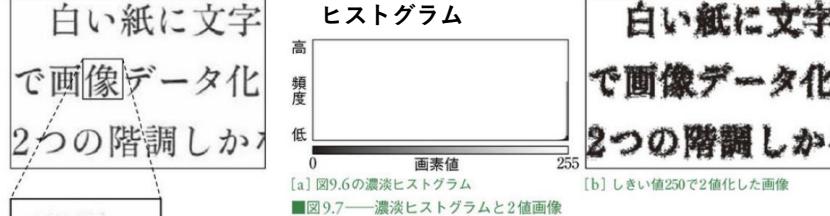


9.二值画像处理

閾値の決め方

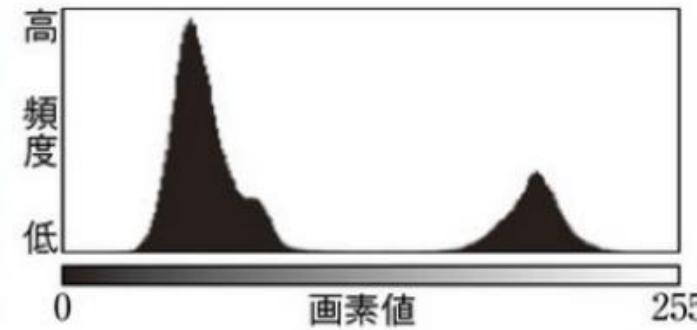
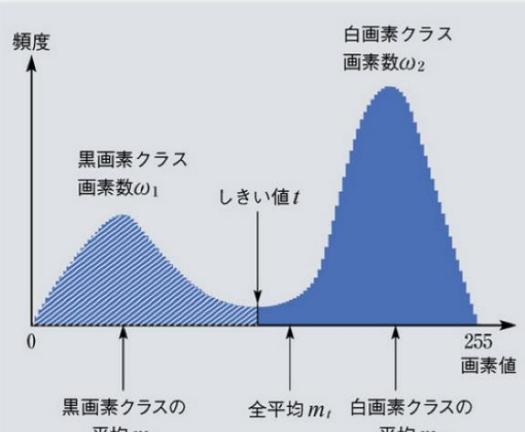
P-タイル法

画素値の低いところから度数を積算し、事前に調べた画像中の文字部分の割合を超えるところを閾値とする方法



モード法

ヒストグラムが二つの山を持つ場合に、その谷を閾値とする。十分な画素数がないとノイズによって谷がはっきりしないためノイズを除去する必要がある。



判別分析法

分離度=(クラス間分散/クラス内分散)を最大にする閾値を求める。
白黒の平均値の差を大きくしつつ、白黒それぞれの分散を小さくすることで分離する。

分離度

$$\frac{\sigma_b^2}{\sigma_w^2} = \frac{\sigma_b^2}{\sigma_t^2 - \sigma_b^2} \quad \sigma_t^2 = \sigma_b^2 + \sigma_w^2$$

クラス間(between)分散=白黒それぞれの平均同士の差

$$\sigma_b^2 = \frac{\omega_1(m_1 - m_t)^2 + \omega_2(m_2 - m_t)^2}{\omega_1 + \omega_2} = \frac{\omega_1 \omega_2 (m_1 - m_2)^2}{(\omega_1 + \omega_2)^2}$$

1=黒クラス, 2=白クラス

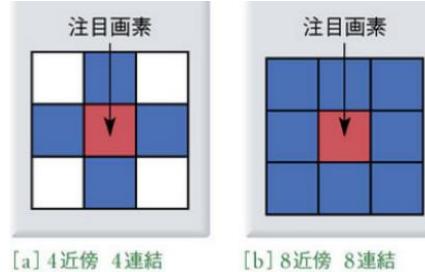
クラス内(within)分散=黒の分散と白の分散の平均

$$\sigma_w^2 = \frac{\omega_1 \sigma_1^2 + \omega_2 \sigma_2^2}{\omega_1 + \omega_2}$$

輪郭追跡

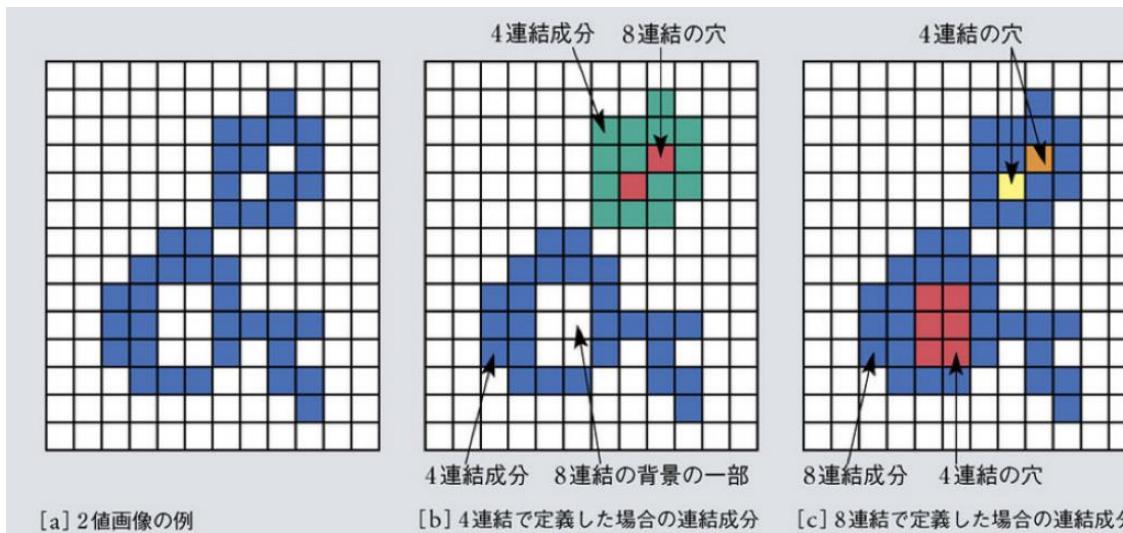
輪郭追跡

連結成分の境界を求める。ラスタスキャンしていく、黒画素を見つけたら、侵入ピクセルから右回りに走査し、黒画素が見つかったらその位置に移動するという処理を繰り返し、追跡済みの画素に同じ方向で侵入したら終了。



[a] 4近傍 4連結

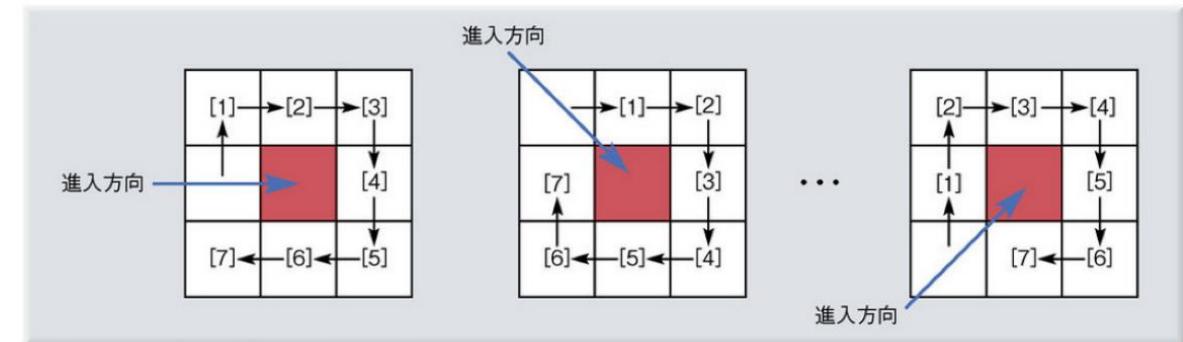
[b] 8近傍 8連結



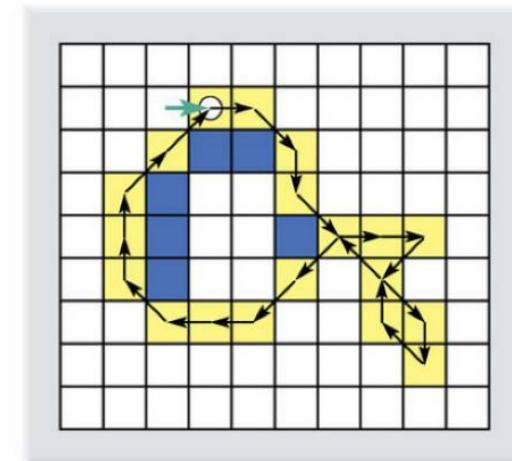
[a] 2値画像の例

[b] 4連結で定義した場合の連結成分

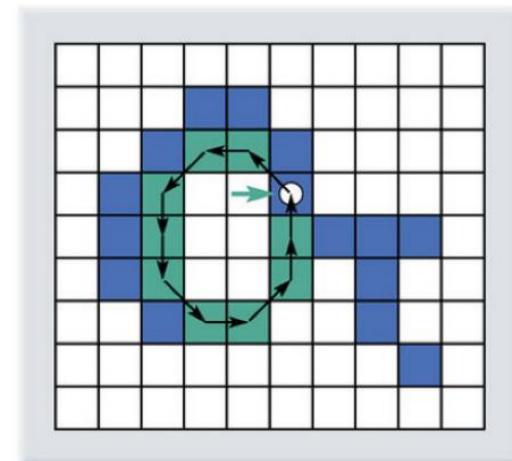
[c] 8連結で定義した場合の連結成分



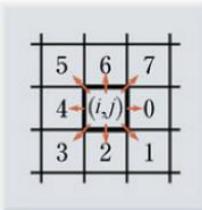
■図9.13—進入方向と黒画素探索の関係



■図9.14—外輪郭の追跡



■図9.15—内輪郭の追跡



■図9.16—
方向コード

チェインコード

輪郭追跡の方向コードの記録。
この情報をNNなどに入力して文字認識を行うこともある。

膨張・収縮

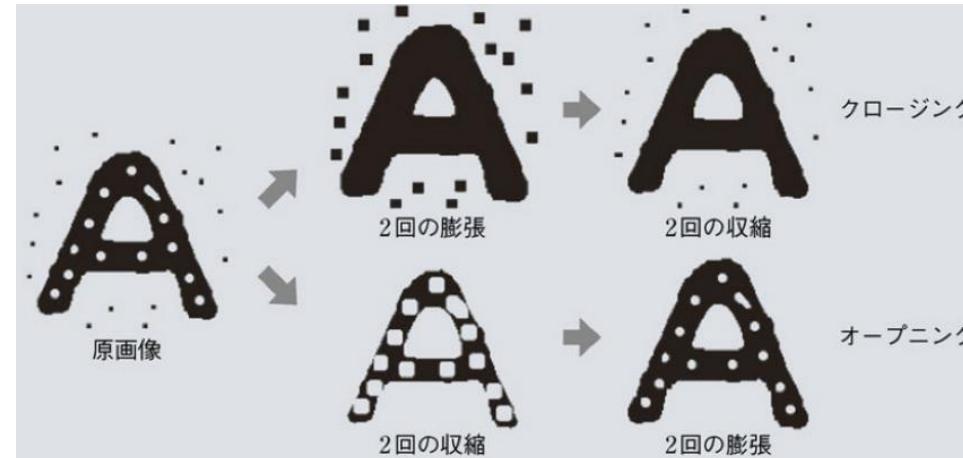
収縮

穴・背景に接する画素を一回りはぎとる

膨張

穴・背景に接する画素にを一回り加える

輪郭追跡の応用→クロージング(膨張→収縮)・オープニング(収縮→膨張)



クロージング・オープニングを組み合わせたノイズ除去



ラベリング…連結成分ごとに異なる番号(ラベル)をつける



ラスタスキヤンして、注目画素の左、左上、上、右上の画素を参照しラベルを持っていればそのラベルを、持っていないければ新たなラベルを付与。参照した画素のうち複数の画素がラベルを持っていた場合は、最小のラベルを付与。その際、ルックアップテーブルの番号も最小の番号に置き換える。

Src	Dst
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
⋮	⋮
⋮	⋮
⋮	⋮



Src	Dst
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
⋮	⋮



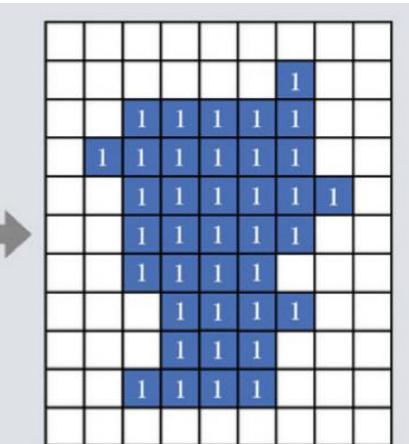
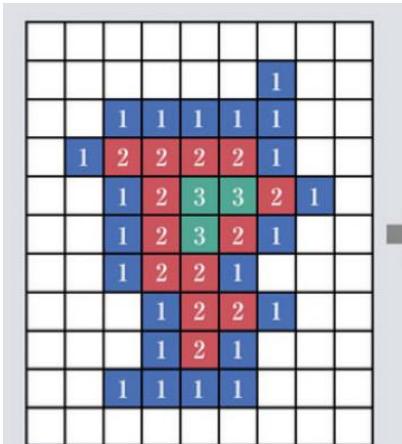
Src	Dst
1	1
2	2
3	1
4	4
5	5
6	6
7	7
8	8
9	9
10	10
⋮	⋮
⋮	⋮
⋮	⋮



Src	Dst
1	1
2	1
3	1
4	4
5	5
6	5
7	7
8	8
9	9
10	10
⋮	⋮
⋮	⋮
⋮	⋮

輪郭追跡を用いた方法

輪郭追跡した外輪郭にラベル1を付与。ラベル1は白画素として連結成分に対して輪郭追跡を行い、ラベル2を付与。この処理を繰り返す。
同じ連結成分についていたラベルを統合して同じラベル番号を振り直す。



形状特徴パラメータ・距離

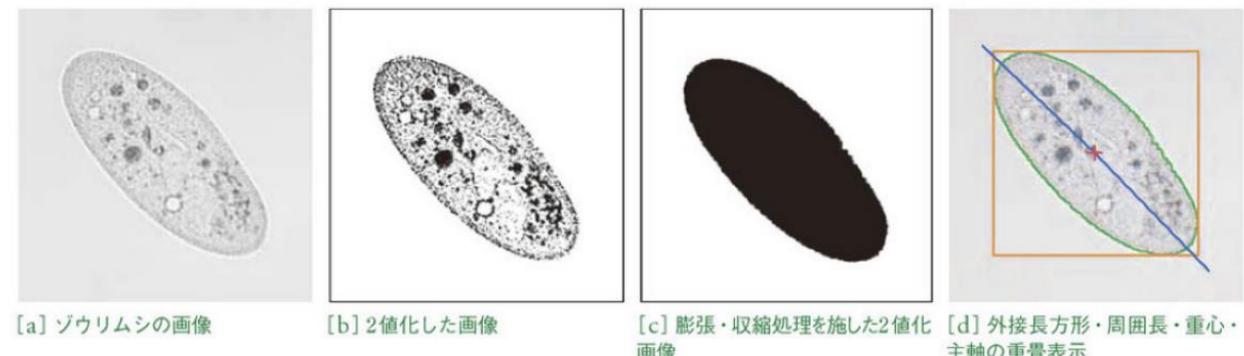
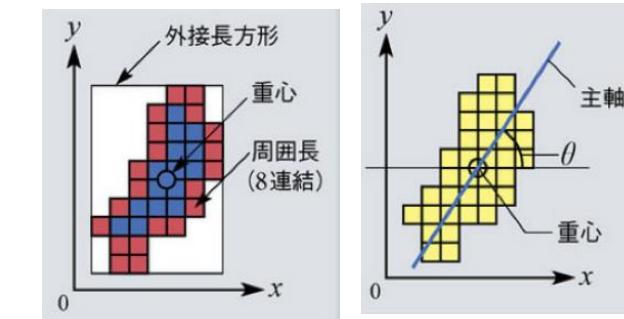
形状特徴パラメータ

二値画像中の連結成分の形状の特徴を数値化したもの

1. 重心
2. 外接長方形…対象に接する最小長方形。おおまかな大きさを知る
3. 面積S、周囲長L(上下左右移動回数+斜め移動回数× $\sqrt{2}$)
4. 円形度… $4\pi S/L^2$ 。円の時1、複雑な形状の時ほど小さい値をとる。
5. オイラー数…連結成分数-穴数
6. モーメント特徴… $f(i,j)$ は座標(i,j)が黒画素なら1、白なら0となる値
 $M(0,0)$ =面積、 $M(1,0)/M(0,0)$ =重心x座標、 $M(0,1)/M(0,0)$ =重心y座標を表す。
7. 主軸方向 θ …モーメント特徴から求められる。

$$M(p, q) = \sum f(i, j) i^p j^q$$

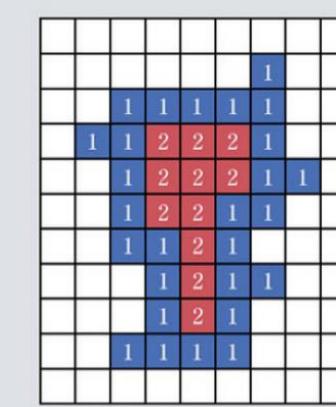
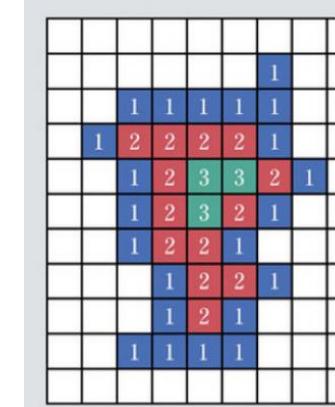
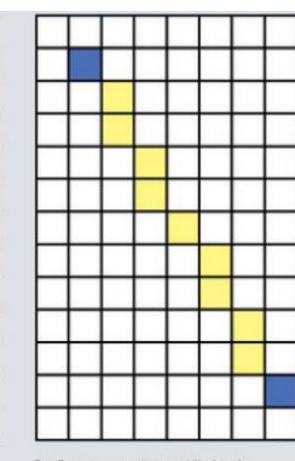
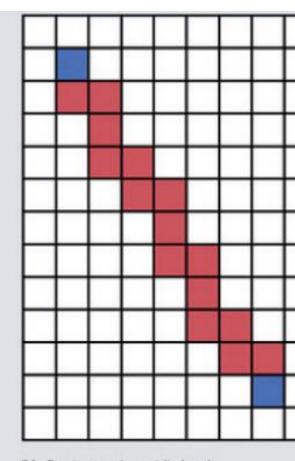
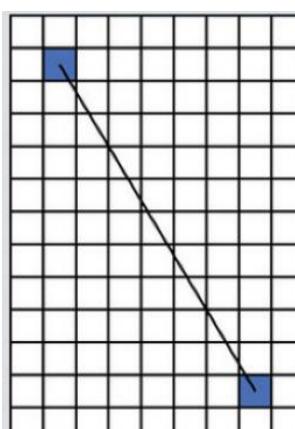
$$\theta = \frac{1}{2} \tan^{-1} \left(\frac{2 \times M(1, 1)}{M(2, 0) - M(0, 2)} \right)$$



距離変換画像

輪郭追跡によるラベリングから得る

距離



細線化・特徴点抽出・ベクトル化

細線化

連結性を保ったまま画素幅まで削る処理。

①～③を満たしたら削る。画素値の更新方法は逐次法と並列法の二種類ある。③端点でないこと。

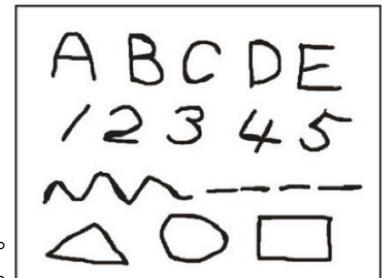
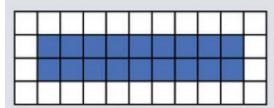
逐次法

更新した画像で次の処理も行う。

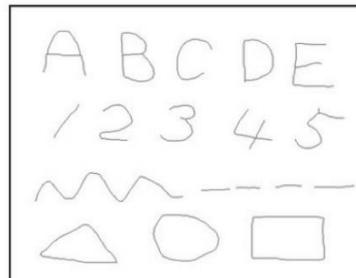
線画像がずれてしまう傾向あり。

並列法

更新する画像とは別に、削っていいか判定するための検証用画像を用意。全画素に対し、更新が終わったら更新用画像を次の検証用画像に用いる。逐次法に比べ位置ずれしにくい。しかし二画素幅の連結成分は消滅してしまう。



[a] 文字や図形の2値画像



[b] 細線化画像

①境界上にある黒画素であること。

②白画素に変更しても連結性が保存されること。

③端点でないこと。

細線の特徴点抽出

論理フィルタを用いて細線を端点、分岐点、孤立点、通過点に分類

0 0 0	* 0 0	* 1 *	0 0 *	0 0 0	1 0 0	0 0 1	0 0 0
0 1 0	1 1 0	0 1 0	0 1 1	0 1 0	0 1 0	0 1 0	0 1 0
* 1 *	* 0 0	0 0 0	0 0 *	1 0 0	0 0 0	0 0 0	0 0 1

[a] 端点

* 1 *	0 1 *	0 0 0	* 1 0	1 0 0	1 0 1	1 0 1	0 0 1
1 1 1	0 1 1	1 1 1	1 1 0	0 1 0	0 1 0	0 1 0	0 1 0
0 0 0	0 1 *	* 1 *	* 1 0	1 0 1	1 0 0	0 0 1	1 0 1

[b] 分岐点(3分岐)

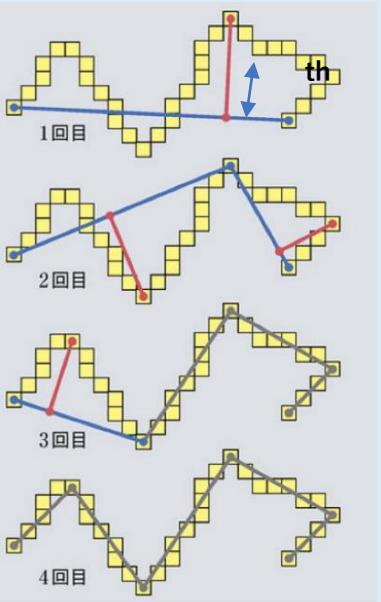
* 1 *	1 0 1	0 0 0					
1 1 1	0 1 0	0 1 0					
* 1 *	1 0 1	0 0 0					

[c] 分岐点(4分岐)

[d] 孤立点

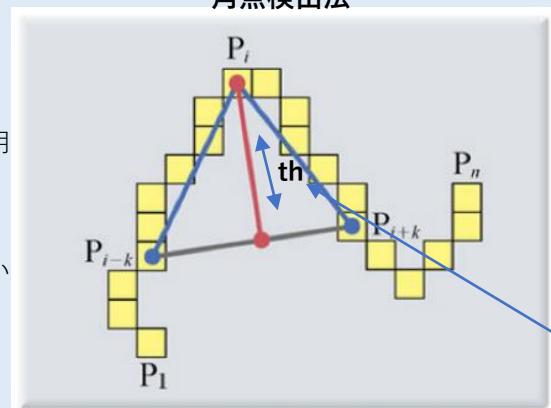
*は0でも1でもよい。理想的な細線化を行った場合は、*はすべて0となる。図の論理フィルタによって、端点、分岐点、孤立点を検出できる。それ以外は通過点となる。

二分割法



画素と線分の最大距離が閾値以上ならその点を分割点として採用
しかし、角点に分割点があるとは限らない

角点検出法



任意の点 P_i から k 画素離れた二点を結ぶ直線に引いた垂線について考える

垂線=長
 P_i から二点への二直線のなす角は大きい
→ P_i は分割点

よって、垂線に対して閾値を設ける

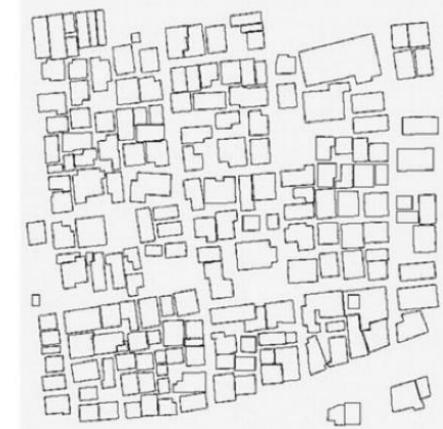
二分割法より
自然なベクトルが得られる

ベクトル化の応用例

地図の画像データをベクトル化し、閉ループを形成する図形のみ家屋として抽出



[a] 紙の地図をスキャナで画像データとして取り込んだ画像



[b] 線分をベクトル化し、家屋らしき图形を抽出した結果

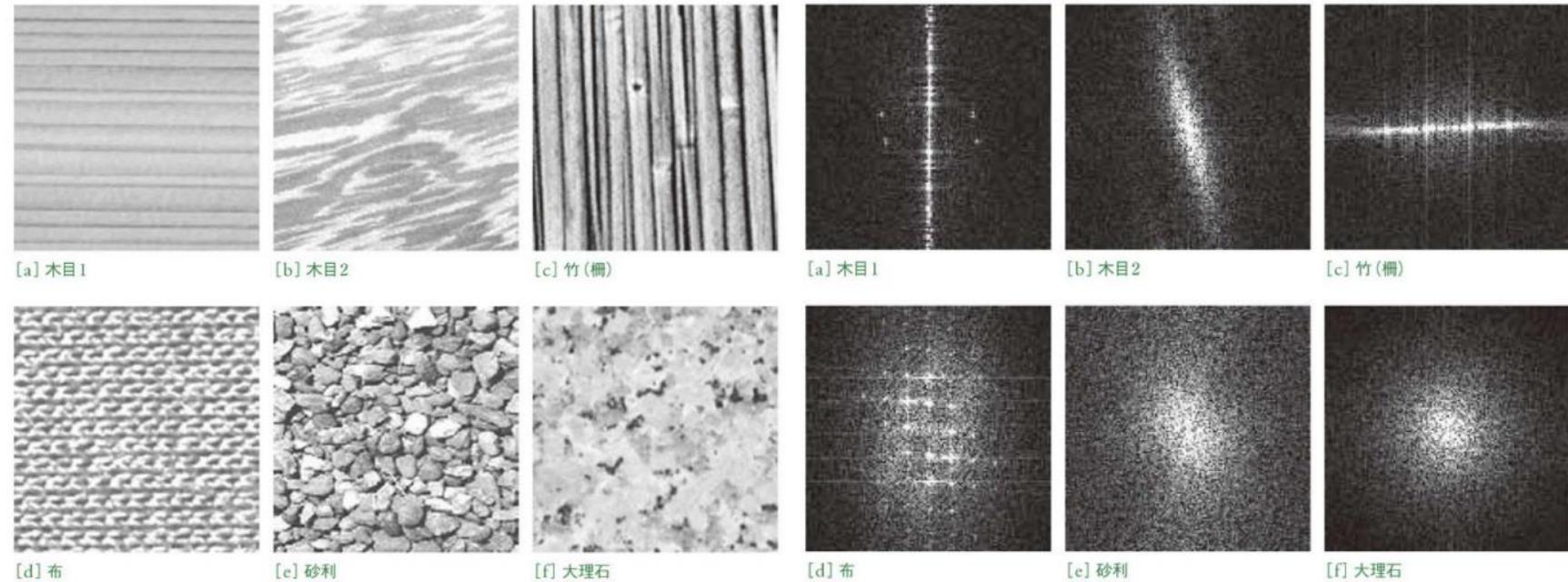
10.領域処理

テクスチャの特徴量1 パワースペクトル

テクスチャ

特定の素材の表面を表す繰り返しの輝度パターン。このパターンに対する特徴量として二次元フーリエ変換・ガボールフィルタ・同時生起行列がある。これらの特徴量は、布を識別したり、撮影した風景を森林・海岸などに領域分割したりするのに使われる。

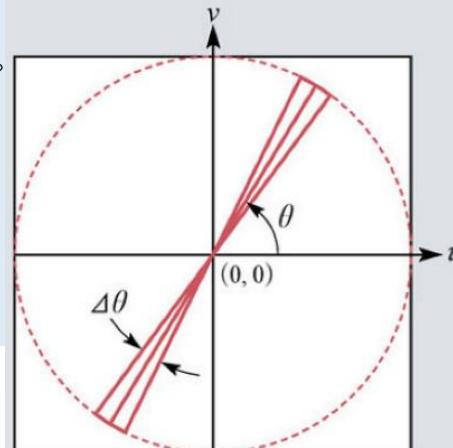
パワースペクトル



パワースペクトルを極座標で表し、幅 $\Delta\theta$ を持つ扇状成分 $p(\theta)$ 、幅 Δr をもつ同心円成分 $p(r)$ に分解

扇状成分 $p(\theta)$

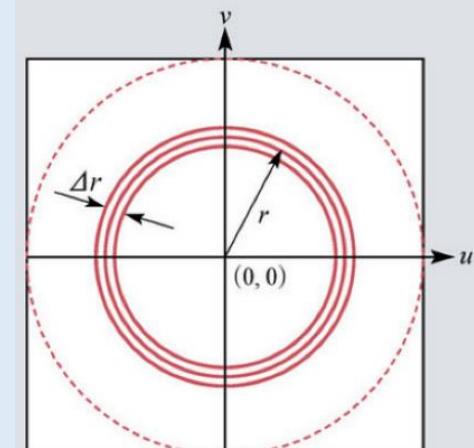
θ の方向の輝度の変化の大きさを示す。



$$p(\theta) = \sum_{\theta'=\theta-\frac{\Delta\theta}{2}}^{\theta+\frac{\Delta\theta}{2}} \sum_{r=0}^{\infty} p(\theta', r)$$

同心円成分 $p(r)$

r の位置に相当する周波数成分の多さを示す。



$$p(r) = \sum_{r'=r-\frac{\Delta r}{2}}^{r+\frac{\Delta r}{2}} \sum_{\theta=0}^{2\pi} p(\theta, r')$$

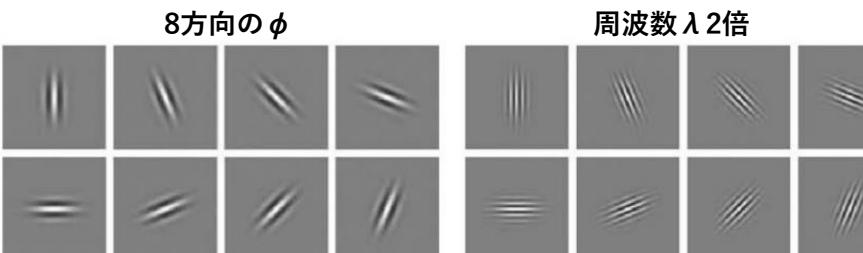
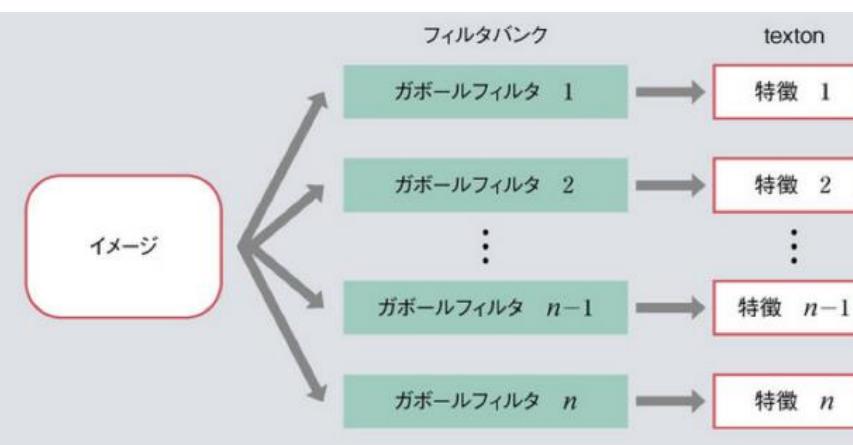
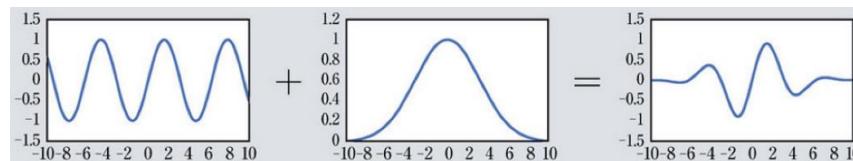
テクスチャの特徴量2 ガボールフィルタ

ガボールフィルタ

パワースペクトルとは異なり、領域のどの位置にどの方向の周波数がどれくらいの強度であるか知ることができる。
sin波(cos波)とガウス関数のかけ合わせ。パラメータを操作した複数のガボールフィルタ(フィルタバンク)で複数の特徴量(texton)を抽出

$$g(x, y, \lambda, \varphi) = \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \exp(2\pi\lambda i(x \cos \varphi + y \sin \varphi))$$

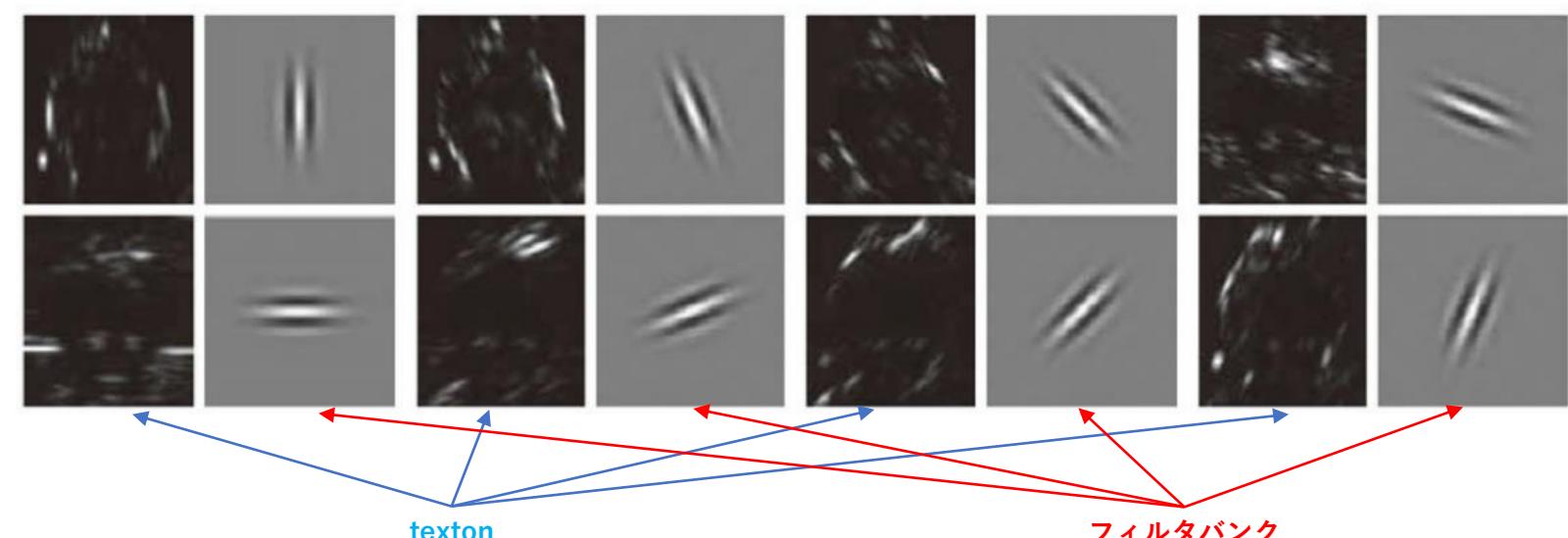
入力画像に畳み込むと、周波数 λ 、角度 φ の成分のみが抽出される。



■図10.6——8方向に向きを変えたガボールフィルタ例



■図10.8——ガボールフィルタの出力結果の例
上段：入力画像（左図）に図10.6のガボールフィルタの出力結果を重ねて出力した図（右図）。入力図全体のすべての方向のエッジが抽出されたことが確認できる。
下段：個々のフィルタの出力結果。



■図10.7——周波数を2倍にした8方向のガボールフィルタ例

texton

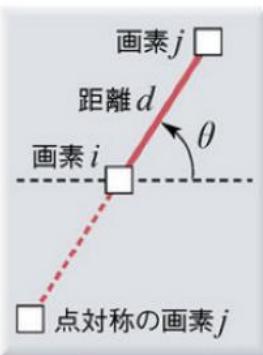
フィルタバンク

テクスチャの特徴量3 同時生起行列

同時生起行列

離れた二つの画素対の出現確率からテクスチャの統計的特徴量を求める行列。

特定の方向、特定の距離の明るさの組み合わせの頻度がわかる。



[a] 2値画像			
1	0	0	1
0	1	0	0
0	1	0	1
1	0	1	1

画素 <i>i</i> の画素値		画素 <i>i</i> の画素値		画素 <i>i</i> の画素値		画素 <i>i</i> の画素値			
画素 <i>j</i> の画素値	0	1	画素 <i>j</i> の画素値	0	1	画素 <i>j</i> の画素値	0	1	
画素 <i>j</i> の画素値	0	4	9	0	6	4	0	6	7
画素 <i>j</i> の画素値	1	9	2	1	4	4	1	7	4

$H_{\langle 1, 0^\circ \rangle}$		$H_{\langle 1, 45^\circ \rangle}$		$H_{\langle 1, 90^\circ \rangle}$		$H_{\langle 1, 135^\circ \rangle}$	
----------------------------------	--	-----------------------------------	--	-----------------------------------	--	------------------------------------	--

■図10.10——2値画像とその同時生起行列の例

0	0	1	1
0	0	1	1
0	2	3	2
2	2	3	3

部分画像

始点の輝度		終点の輝度					
0	1	2	3	0	1	2	3
4	2	1	0	4	2	1	0
2	4	0	0	2	4	0	0
1	0	6	1	1	0	6	1
0	0	2	0	0	0	2	0

同時生起行列

図30 部分画像と算出した同時生起行列(角度0°)の模式図

$$ENR = \sum_{L_i=0}^{L-1} \sum_{L_j=0}^{L-1} \{P_\delta(L_i, L_j)\}^2 \quad (\text{エネルギー : energy})$$

$$CNT = \sum_{k=0}^{L-1} k^2 P_{x-y}(k) \quad (\text{コントラスト : contrast})$$

$$CRR = \frac{\sum_{L_i=0}^{L-1} \sum_{L_j=0}^{L-1} L_i L_j P_\delta(L_i, L_j) - \mu_x \mu_y}{\delta_x \delta_y} \quad (\text{相関 : correlation})$$

$$VAR = \sum_{L_i=0}^{L-1} \sum_{L_j=0}^{L-1} (L_i - \mu_x)^2 P_\delta(L_i, L_j) \quad (\text{分散 : variance})$$

$$EPY = - \sum_{L_i=0}^{L-1} \sum_{L_j=0}^{L-1} P_\delta(L_i, L_j) \log \{P_\delta(L_i, L_j)\} \quad (\text{エントロピー : entropy})$$

$$SEP = - \sum_{k=0}^{2L-2} P_{x+y}(k) \log \{P_{x+y}(k)\} \quad (\text{サムエントロピー : sum entropy})$$

$$IDM = \sum_{L_i=0}^{L-1} \sum_{L_j=0}^{L-1} \frac{1}{1 + (L_i - L_j)^2} P_\delta(L_i, L_j) \quad (\text{逆差分モーメント : inverse difference moment})$$

領域分割処理1 領域統合法

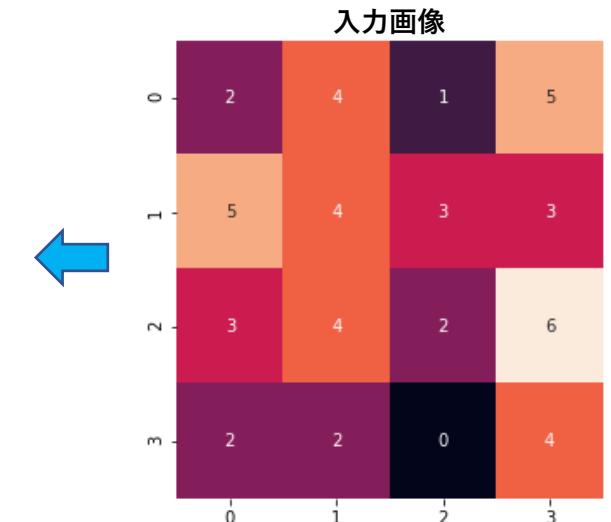
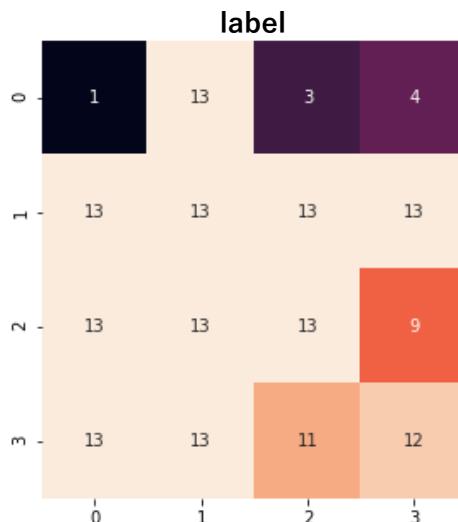
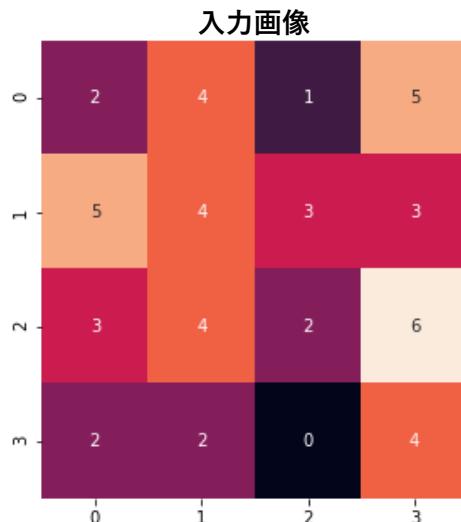
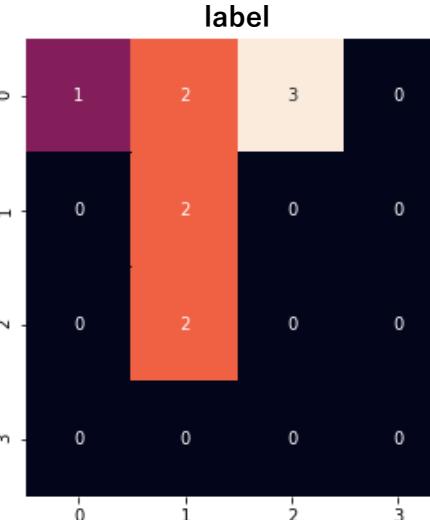
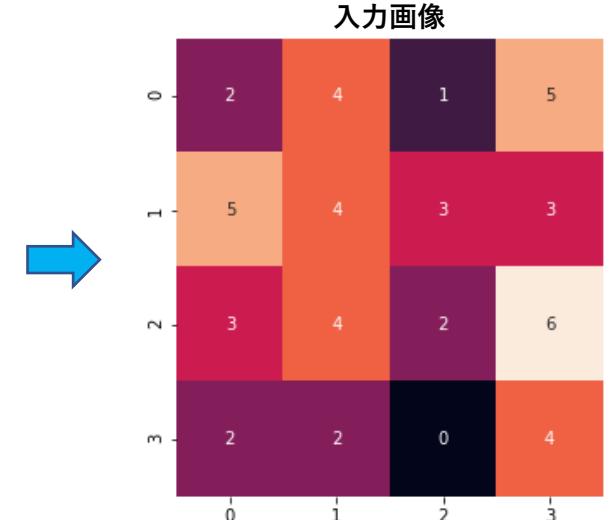
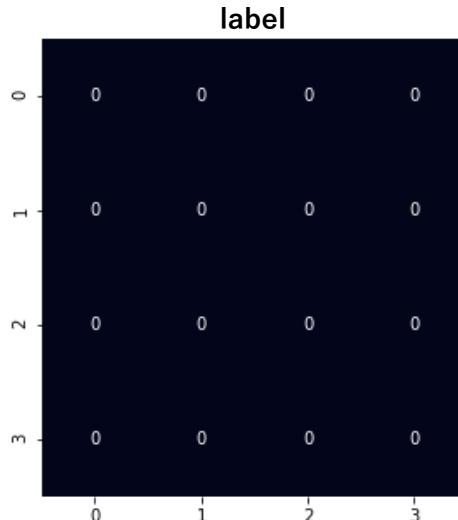
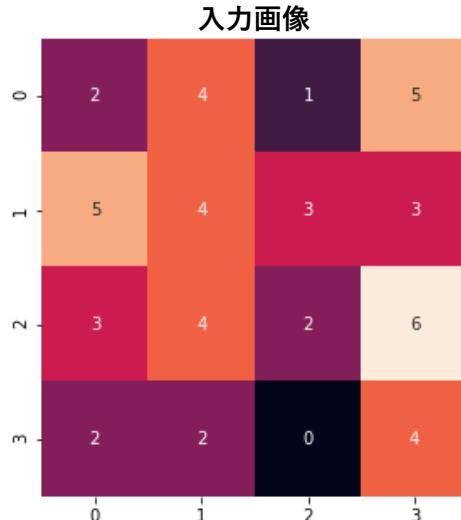
領域分割

画像中の性質が同じ物体の領域ごとに分割する。二値画像のラベリングが離れた場所にある複数の連結成分を分割するのに対し、領域分割は隣接している別の物体(顔と髪など)をも分割する。

領域統合法

ラスタスキャンして隣接する同一画素値の画素は同一のラベル付け。隣接する集合の平均値の差を求め上限値未満なら統合して新たなラベルで統合。

統合できなくなるまで繰り返す。下の例は上限値が1の場合。



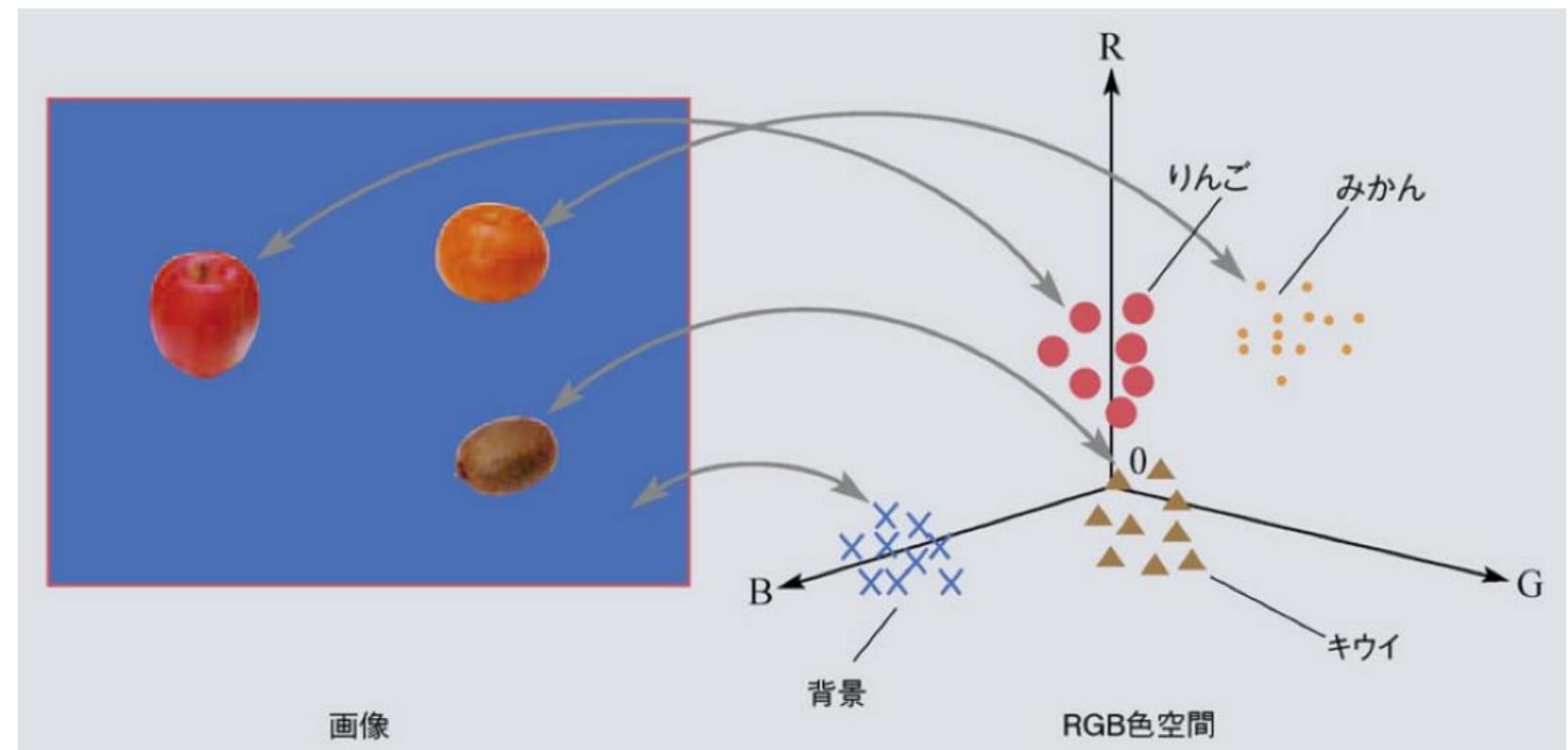
領域分割処理2 画素特徴量クラスタリング

画素特徴量クラスタリング

特徴が近い画素は特徴量空間で近くにあり、まとめているという前提に基づいた手法。

画素ごとの特徴量算出
↓
特徴空間に写像
↓
特徴空間でクラス分け

色情報から画素をRGB空間でクラスタリングする様子

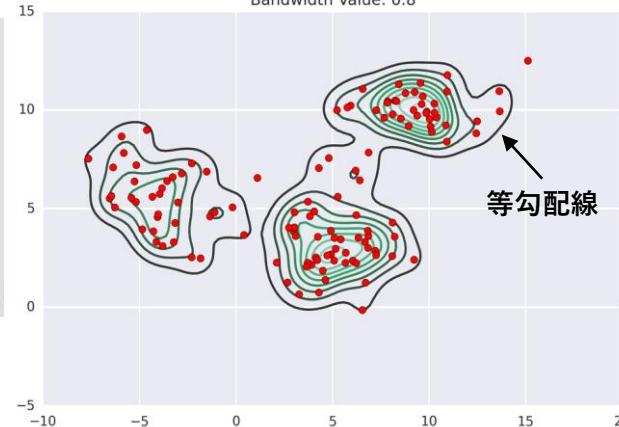
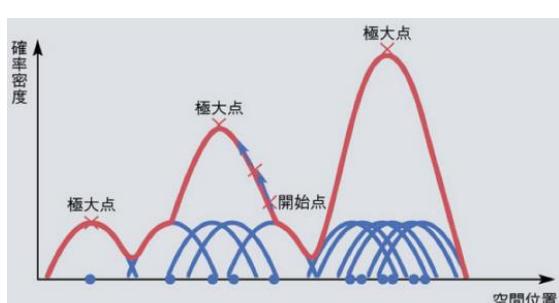


領域分割処理2 画素特徴量クラスタリングミーンシフト

ミーンシフト

特徴量空間において、画素の確率密度関数を推定し、勾配を上るように
極大点に移すことによってクラス分けを行う。クラス数=極大点数となる

Bandwidth Value: 0.8



勾配法を用いた極大点探索

初期空間位置 y_0 を与える。



近傍の重心との差だけ位置を更新

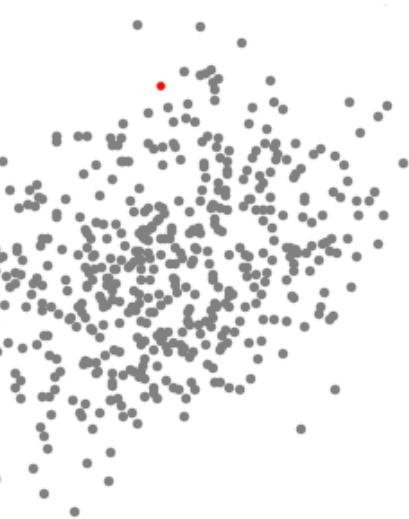
$$y_{j+1} \leftarrow y_j + \left(\frac{\sum_{i=1}^N g_i x_i}{\sum_{i=1}^N g_i} - y_j \right)$$

↔
重心



$|y_{j+1} - y_j| < \delta$ (しきい値) と

なるまで繰り返す



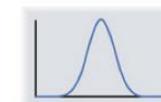
確率密度関数の推定

確率密度関数は以下の式で推定できる。

h はカーネル関数の中心からの半径(ガウシアンの σ に相当)。 c は正規化係数。

ガウシアン (Gaussian) カーネル

$$K(t) = \exp\left(-\frac{|t|^2}{2}\right)$$

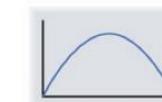


プロファイル

$$k(t) = \exp\left(-\frac{t}{2}\right)$$

エパネックニコフ (Epanechnikov) カーネル

$$K(t) = \begin{cases} 1-|t|^2 & (|t| \leq 1) \\ 0 & (\text{それ以外}) \end{cases}$$



プロファイル

$$k(t) = \begin{cases} 1-t & (0 \leq t \leq 1) \\ 0 & (t > 1) \end{cases}$$



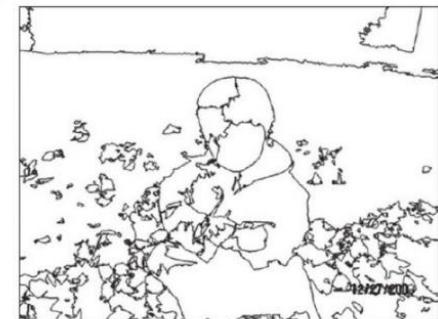
[a] 原画像



[b] $(h_s, h_r)=(7, 6.5)$



[d] $(h_s, h_r)=(21, 19.5)$



領域分割処理3 スネーク

スネーク

対象物体を囲む閉曲線を初期値として与え、閉曲線をそのエネルギーを小さくする方向に移動していく、
対象物のエッジに沿って貼りつくことによって領域を抽出する。

$$\varepsilon(v) = \alpha \varepsilon_{len} + \beta \varepsilon_{curv} + \gamma \varepsilon_{img}$$

$$\varepsilon_{len} = \int_0^1 \left\| \frac{dv(s)}{ds} \right\|^2 ds$$

:弧長に対応する項
線分を短くするエネルギー

$$\varepsilon_{curv} = \int_0^1 \left\| \frac{d^2v(s)}{ds^2} \right\|^2 ds$$

:曲率に対応する項
凹凸を抑えるエネルギー

$$\varepsilon_{img} = - \int_0^1 \left\| \nabla (G \otimes I(v(s))) \right\| ds$$

:勾配強度に対応する項
エッジを超えないための
エネルギー

Step1: 曲線は指定

Step2: 変数を設定・初期化

d : 頂点の総移動量

n : 繰り返し回数

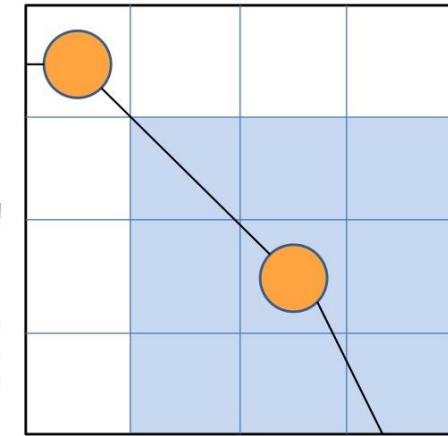
Step3: n=0にする

頂点の8近傍の $\varepsilon(v)$ を
それぞれ計算

Step4: $\varepsilon(v)$ 最小の画素へ移動
し、移動量をdに加算

Step5: すべての頂点において
計算をし、Step3に戻る

Step6: しきい値以下のdになる
か、n回処理したら終了



閉曲線と媒介変数

閉曲線は一つのxに対し、複数のyが存在するため、
単純に $y=f(x)$ では表せない。

シンプルに表すため、媒介変数を使う。

例えば円の場合、

$$x^2 + y^2 = r^2$$

$$x = r \cos \theta$$

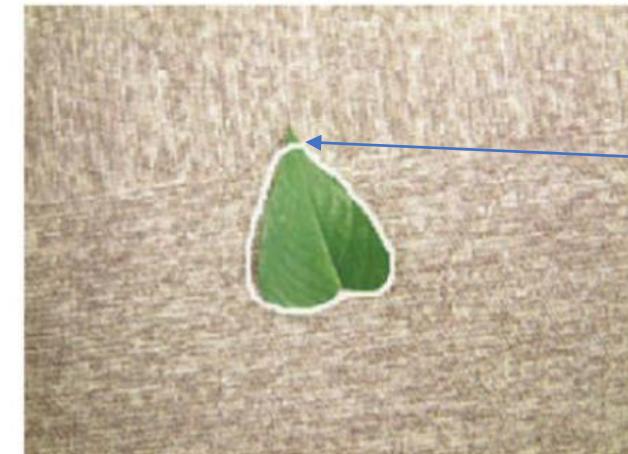
$$y = r \sin \theta \quad (0 \leq \theta \leq 2\pi)$$

$$x = r \cos 2\pi s$$

$$y = r \sin 2\pi s \quad (0 \leq s \leq 1)$$



[a] 葉の画像と閉曲線の初期値

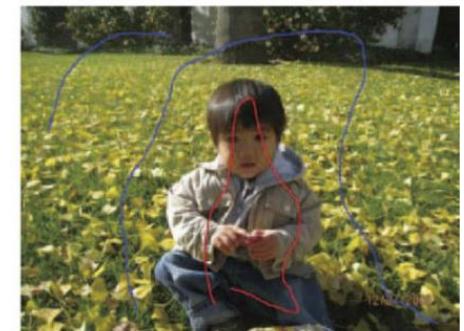
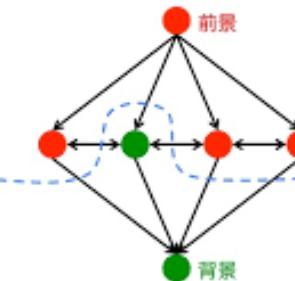


[b] 抽出結果

より短く、より凹凸が
ないように移動するため、
鋭い角度を持ったエッジは
抽出しづらいという欠点がある。

領域分割処理4 グラフカット

グラフカット ~segmentation~



[a] 赤線、青線でそれぞれ対象と背景の一部を指定



[b] 子どもの領域(対象)を取り出した画像



[c] 風景の背景画像



[d] [b]の人物画像に[c]を合成した画像

グラフカット

各画素を対象と背景に分割する。画素値の対象物体らしさ(背景らしさ)と、同一クラスの画素の類似度との和をコスト関数として、最小となる分割線を求める手法。

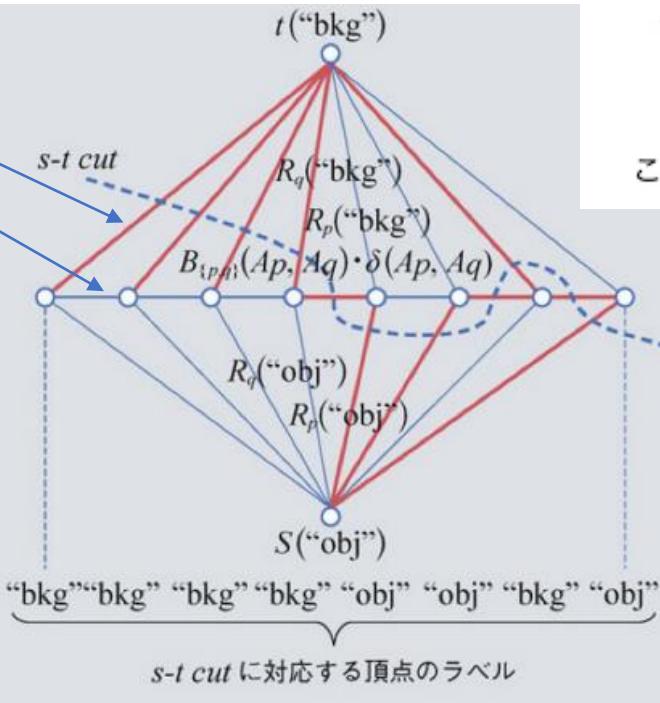
縦のつながり=対象物体らしさ、背景らしさ $R(L)$
横のつながり=隣接する画素との類似度 $B(L)$

↓
 $R(L)$ と $B(L)$ の総和が最小になるような $s-t$ cut を求める。

$$E(L) = R(L) + \lambda \cdot B(L)$$

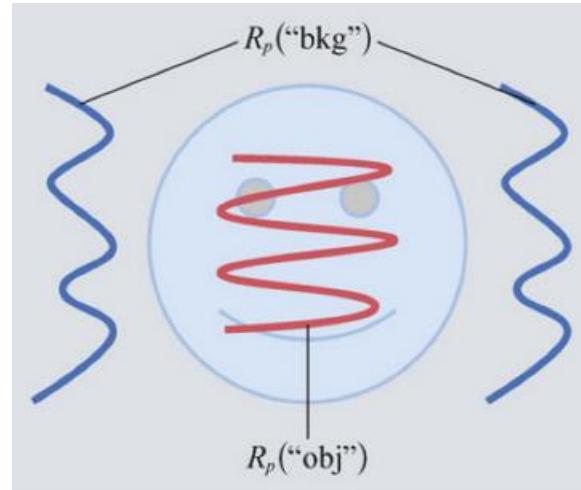
$$R(L) = \sum_{p \in P} R_p(L_p)$$

$$B(L) = \sum_{\{p, q\} \in N} B_{\{p, q\}} \cdot \delta(L_p, L_q), \quad \delta(L_p, L_q) = \begin{cases} 1 & (L_p \neq L_q) \\ 0 & (\text{それ以外}) \end{cases}$$



このグラフを $s-t$ -mincut するだけ

指定値
あらかじめ対象と背景の一部を与えておく



コスト関数の一例

$$R_p("obj") = -\ln \Pr(I_p | "obj") \quad \Pr(I_p | "obj") \text{ は指定値から求めた obj に属する画素の内、画素値 } I_p \text{ である割合}$$

$$R_p("bkg") = -\ln \Pr(I_p | "bkg")$$

$$B_{\{p, q\}} = \exp\left(-\frac{(I_p - I_q)^2}{2\sigma^2}\right) \cdot \frac{1}{\text{dist}(p, q)} \quad \text{隣接する画素 } (p, q) \text{ が同じ画素値の場合は } 0 \text{ になる}$$

$$\delta\{L_p, L_q\} = \begin{cases} 1 & (L_p \neq L_q) \\ 0 & (\text{それ以外}) \end{cases}$$

11. パターン・図形・特徴 の抽出とマッチング

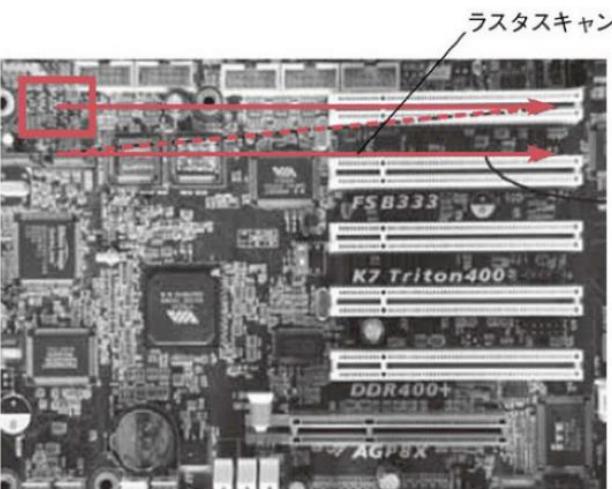
テンプレートマッチング1 類似度・サブピクセル位置推定

パターン

特定の画素値の並び。

テンプレートマッチング

あらかじめ標準パターンをテンプレートとして用意しておき、
それと対象画像内の同サイズの領域との間の類似度を
計算することで、位置を特定すること



類似度・相違度

SSD(偏差平方和)

= ベクトル先端のユークリッド距離の二乗

$$R_{SSD} = \sum_{j=0}^{N-1} \sum_{i=0}^{M-1} (I(i, j) - T(i, j))^2$$

SAD(偏差絶対値和)

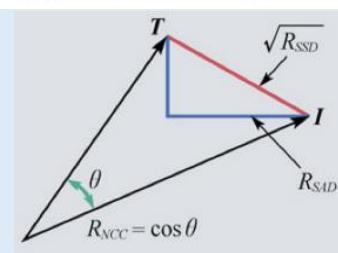
= ベクトル先端の市街地距離

$$R_{SAD} = \sum_{j=0}^{N-1} \sum_{i=0}^{M-1} |I(i, j) - T(i, j)|$$

NCC(ベクトルのなす角の余弦)

= 1に近いほど類似している

$$R_{NCC} = \frac{\sum_{j=0}^{N-1} \sum_{i=0}^{M-1} I(i, j)T(i, j)}{\sqrt{\sum_{j=0}^{N-1} \sum_{i=0}^{M-1} I(i, j)^2} \times \sqrt{\sum_{j=0}^{N-1} \sum_{i=0}^{M-1} T(i, j)^2}}$$



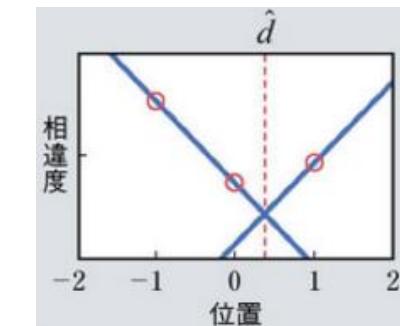
ZNCC(相互相関係数)

= 1に近いほど類似

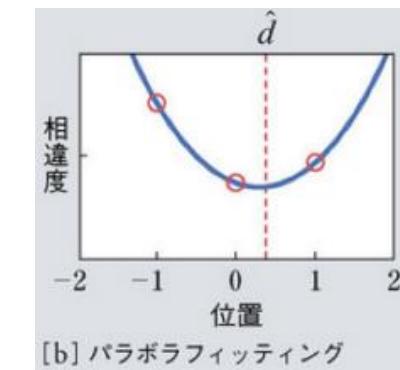
$$R_{ZNCC} = \frac{\sum_{j=0}^{N-1} \sum_{i=0}^{M-1} ((I(i, j) - \bar{I})(T(i, j) - \bar{T}))}{\sqrt{\sum_{j=0}^{N-1} \sum_{i=0}^{M-1} (I(i, j) - \bar{I})^2} \times \sqrt{\sum_{j=0}^{N-1} \sum_{i=0}^{M-1} (T(i, j) - \bar{T})^2}}$$

サブピクセル位置推定

相違度を画素ごとではなく、連続関数でフィッティングし、
相違度が最小となる位置を求ること。



[a] 等角直線フィッティング



[b] パラボラフィッティング

SADに合う

$$\hat{d} = \begin{cases} \frac{1}{2} \frac{R(1)-R(-1)}{R(0)-R(-1)} & (R(1) < R(-1)) \\ \frac{1}{2} \frac{R(1)-R(-1)}{R(0)-R(1)} & (\text{それ以外}) \end{cases}$$

R(0)=最小相違度
R(-1),R(1)=R(0)の隣接相違度

SSD、相互相関係数に合う

$$\hat{d} = \frac{R(-1)-R(1)}{2R(-1)-4R(0)+2R(1)}$$

テンプレートマッチング2 高速探索法

高速探索法

限られた範囲のみで効率的に類似度を計算するテンプレートマッチング

残差逐次検定法(SSDA)

SAD計算時、残差が閾値を超えたらその時点で
目的の位置ではないと判断し、加算を打ち切る。
閾値は過去の残差の最小値とする(閾値自動決定法)。

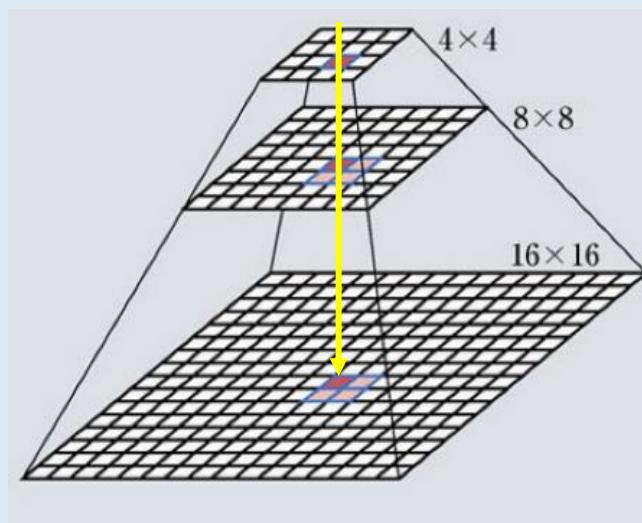
粗密探索法

テンプレートと入力画像とともに
数段階の平均化によって解像度を下げる。

↓
その状態でテンプレートマッチング

↓
解像度を1段階上げ、対応の範囲のみでテンプレートマッチング

↓
元の解像度まで繰り返す。

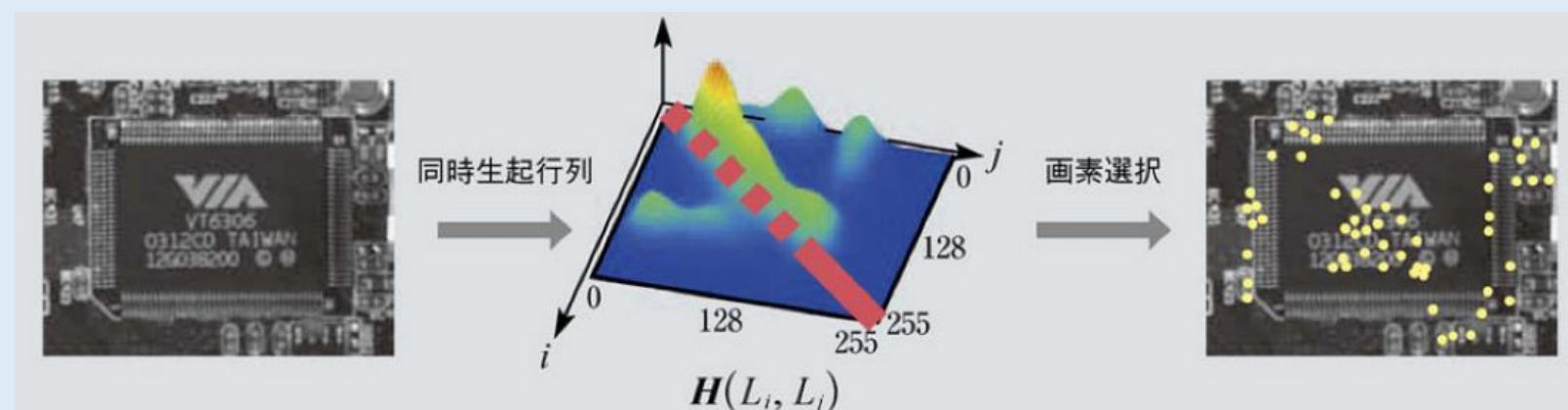


同時生起行列による参照画素の絞り込み

同時生起行列 $p(L_i, L_j)$ が低い

↓
 L_i と L_j の組み合わせが起こることは珍しい

↓
テンプレートの i, j の画素のみを参照して類似度を計算すれば効率がよい

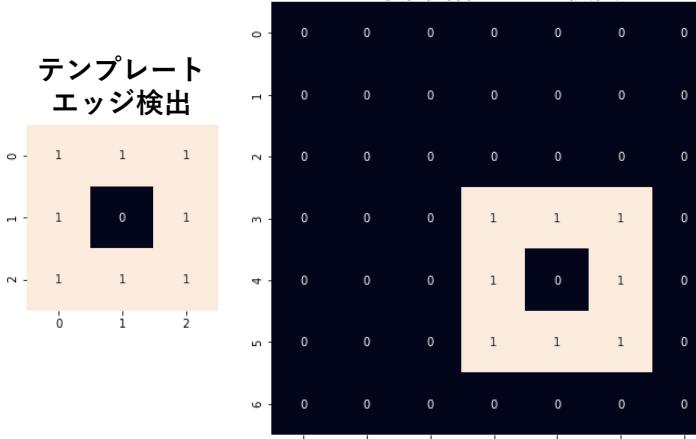


テンプレートマッチング3 チャンファーマッチング

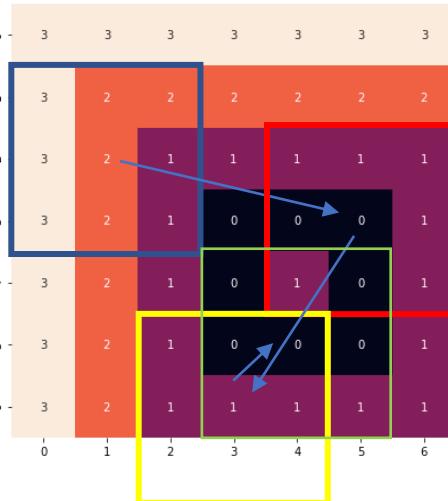
チャンファーマッチング

エッジ検出したテンプレートと、エッジ検出&距離変換&微分した対象画像の類似度計算によるマッチング

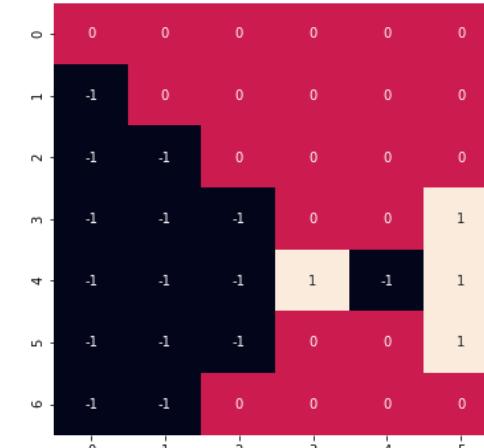
対象画像エッジ検出



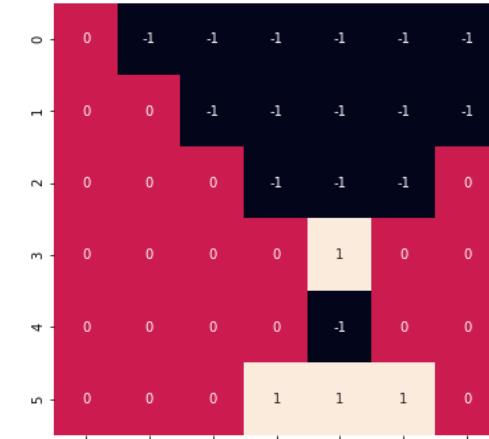
距離変換画像



距離画像x方向微分



距離画像y方向微分



位置更新方法

上のテンプレートを青枠(0,2)に置いた場合、

$$x' = x - (-5) = 0 + 5 = 5$$

$$y' = y - (-1) = 2 + 1 = 3$$

従って、赤枠(5,3)に移動。

$$x' = x - (2) = 5 - 2 = 3$$

$$y' = y - (-3) = 3 + 3 = 6$$

従って、黄枠(3,6)に移動。

$$x' = x - (-1) = 3 + 1 = 4$$

$$y' = y - (1) = 6 - 1 = 5$$

従って、緑枠(4,5)に移動。

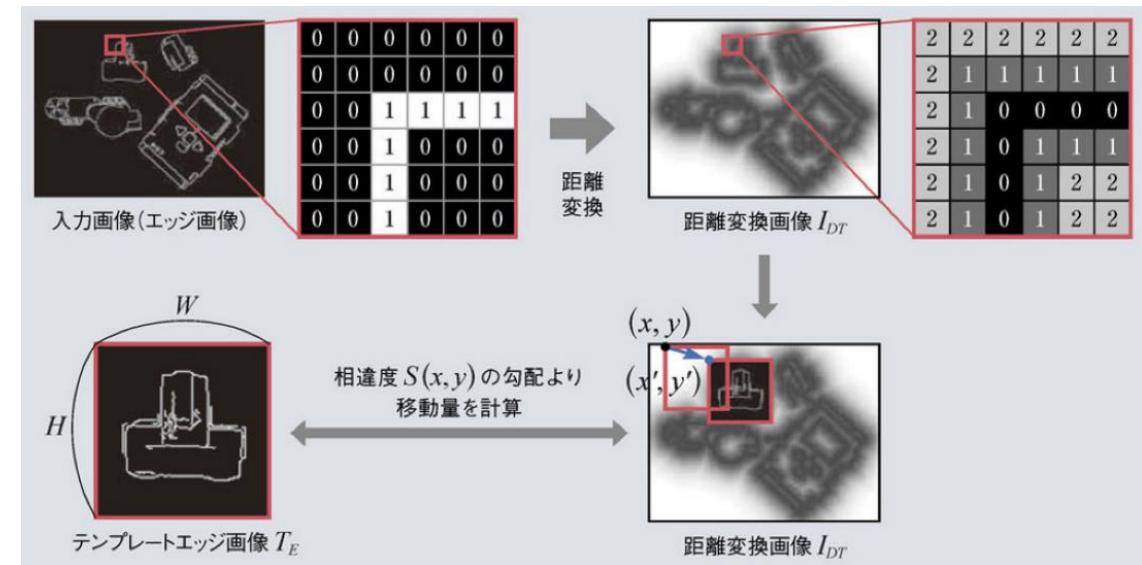
これを繰り返すことで、正しい位置(4,4)を探索

$$S(x, y) = \sum_{j=0}^H \sum_{i=0}^W T_E(i, j) I_{DT}(x+i, y+j)$$

$$\frac{\partial}{\partial x} S(x, y) = \sum_{j=0}^H \sum_{i=0}^W T_E(i, j) \frac{\partial}{\partial x} I_{DT}(x+i, y+j)$$

$$\frac{\partial}{\partial y} S(x, y) = \sum_{j=0}^H \sum_{i=0}^W T_E(i, j) \frac{\partial}{\partial y} I_{DT}(x+i, y+j)$$

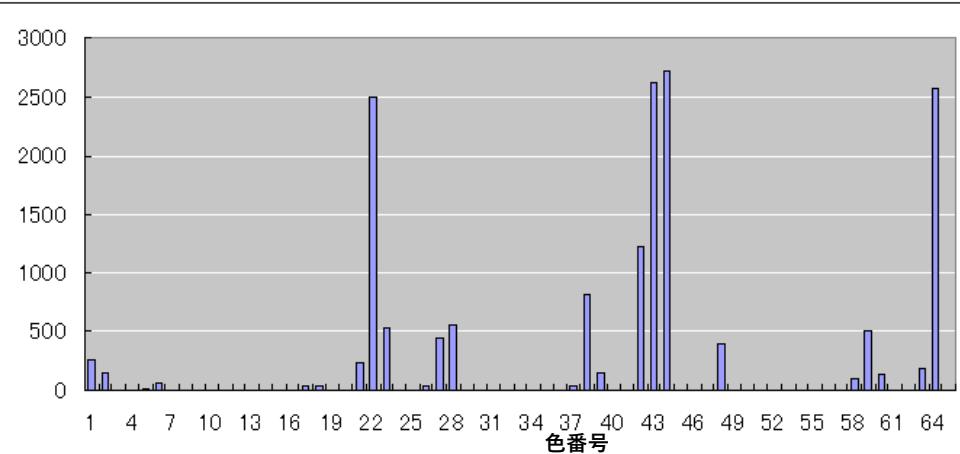
$$(x', y') = \left(x - \frac{\partial}{\partial x} S(x, y), y - \frac{\partial}{\partial y} S(x, y) \right)$$



テンプレートマッチング4 アクティブ探索法

カラーヒストグラム

$256^3 = 1600$ 万色を減色して、描画。下は64色に減色した例。



カラーヒストグラムの類似度

ヒストグラムインタセクション

各ヒストグラムを正規化する
↓
各色番号の頻度を比較する
↓
小さい方を積算する
↓
元のヒストグラムの積算は1であるから
1に近い = 類似度が高いと言える

$$\rho(p, q) = \sum_{i=1}^n \min(p_i, q_i)$$

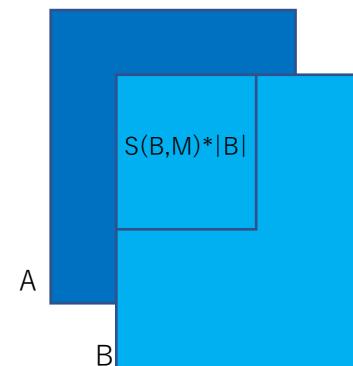
バタチャリヤ係数

各ヒストグラムを正規化する
↓
各色番号の頻度の幾何平均をとる
↓
元のヒストグラムの積算は1であるから
1に近い = 類似度が高いと言える

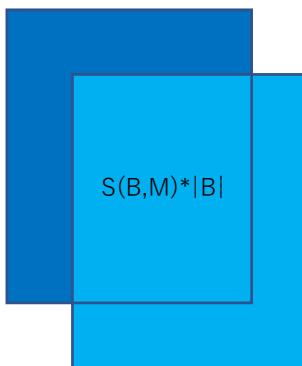
$$\rho(p, q) = \sqrt[n]{\prod_{i=1}^n p_i * q_i}$$

テンプレートとの間のカラーヒストグラム類似度を計算済みの矩形領域と、どれだけ重なっているかに応じて類似度を計算すべきかを判断する方法。探索するか否かを判断したい矩形領域をA、探索済みの矩形領域をB、テンプレートをM、 $S(B, M)$ はBとMの類似度(カラーヒストグラムの類似度)、 $|A|$ 、 $|B|$ はA、Bに含まれる画素数とする。カラーヒストグラムの類似度は二つの領域の重なっている領域の割合であるから、 $S(B, M)$ はBのうちMと等しい画素の割合を意味する。従って、 $S(B, M)*|B|$ はBのうちMと等しい画素の個数を意味する。このことから、 $|A \cap B|$ 、 $S(B, M)$ を用いて、 $S(A, M)$ の上限値が求まる。この上限値が低い場合は対象のパターンが存在しないので、計算を省く。このような類似度計算回数を大幅に削減した方法をアクティブ探索法と呼ぶ。

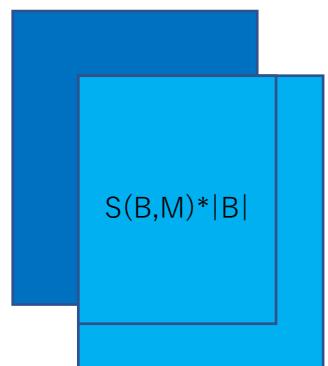
$S(B, M)*|B| < |A \cap B|$ の場合



$S(B, M)*|B| = |A \cap B|$ の場合



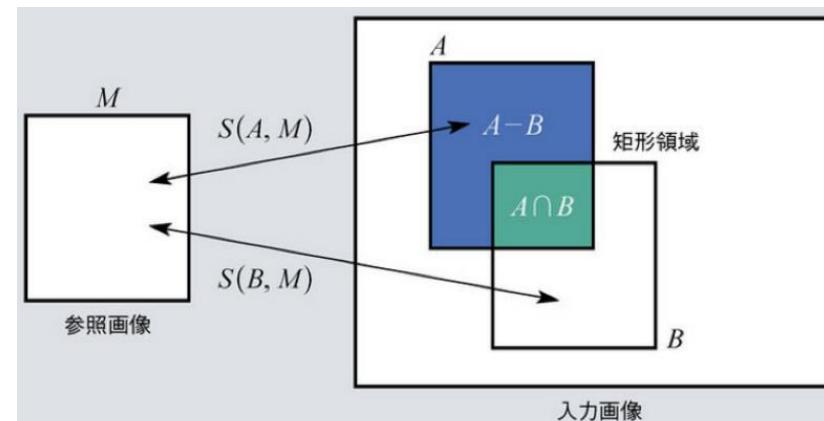
$S(B, M)*|B| > |A \cap B|$ の場合



$$S(A, M) = \frac{S(B, M)|B| + |A - B|}{|A|}$$

$$S(A, M) = \frac{|A \cap B| + |A - B|}{|A|} = 1$$

$$S(A, M) = \frac{|A \cap B| + |A - B|}{|A|} = 1$$



類似度 $S(B, M)$	共通領域 $A \cap B$ の重なり率		
	90%	50%	10%
0.8	0.9	1.0	1.0
0.5	0.6	1.0	1.0
0.1	0.2	0.6	1.0

特徴点検出1 ハリスのコーナー検出

特徴点を検出できれば、その点からマッチングを行うことで効率化できる。

ハリスのコーナー検出

コーナー領域はx方向、y方向どちらも勾配が大きいという性質を利用。

計算方法は以下。なぜこうなるのかは不明のため、
とりあえず手順だけ覚える。

$$\mathbf{M}(x, y) = \begin{pmatrix} S_{xx}(x, y) & S_{xy}(x, y) \\ S_{xy}(x, y) & S_{yy}(x, y) \end{pmatrix}$$

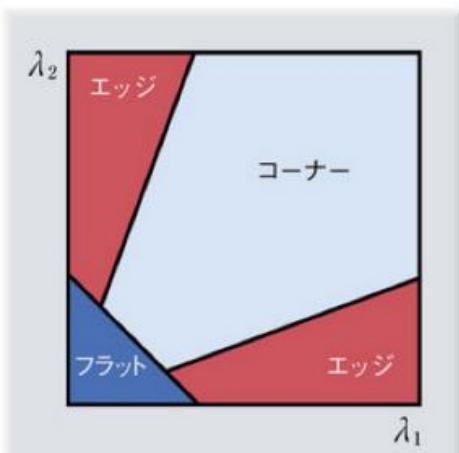
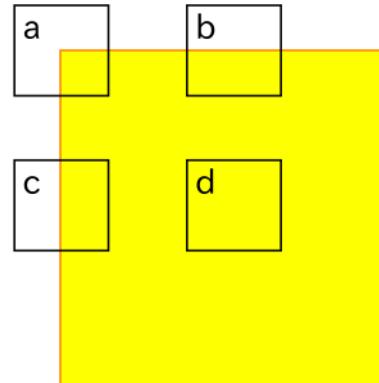
を求める。

S_{xx} = x方向の勾配の大きさの二乗のガウス畳み込み

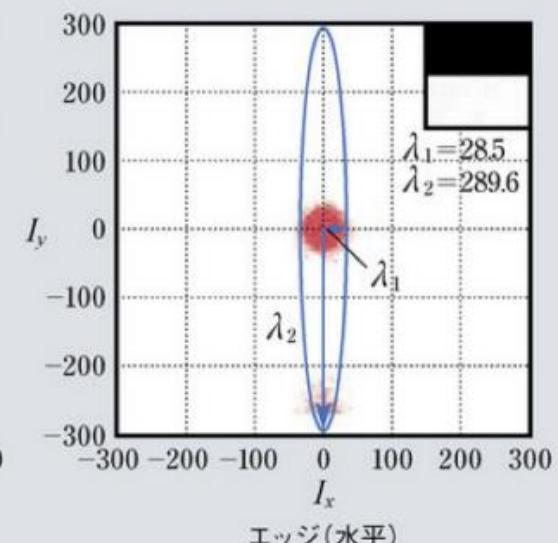
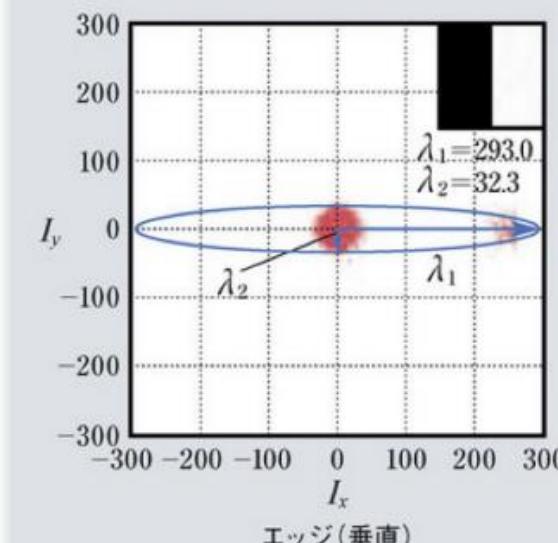
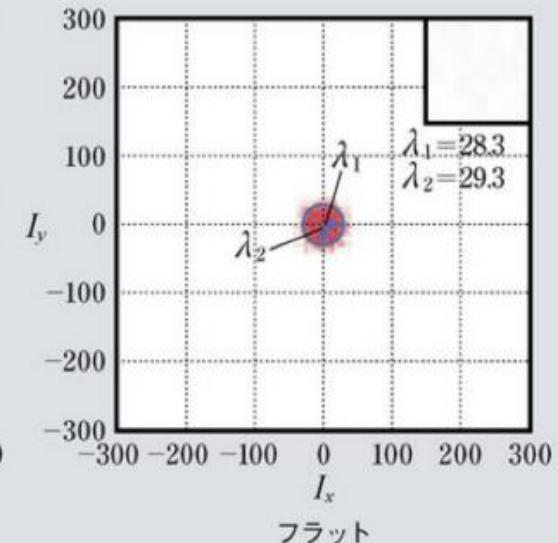
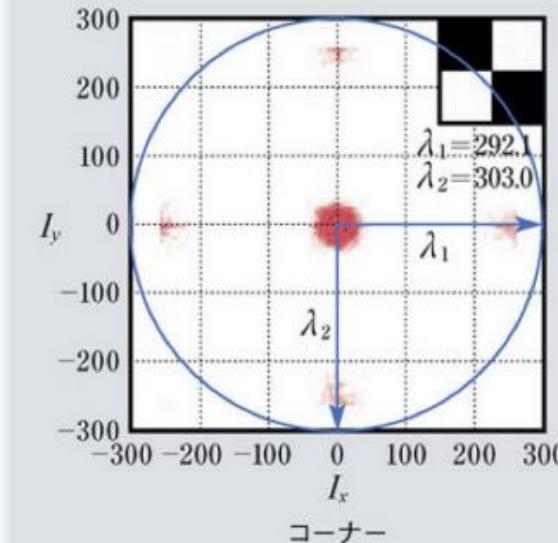
$$S_{xx} = G(\sigma') * I_{xx}$$

以下のRが閾値を超えたらコーナー

$$R = \det \mathbf{M} - k (\text{tr } \mathbf{M})^2$$



■図11.9—固有値による判定



特徴点検出2 FASTのコーナー検出

学習

大量の画像を用意

↓
各画像について、注目画素とその周囲16画素の値を比較し、明るい、暗い、類似の三値化

↓
16画素を並べて、明るいもしくは暗いが閾値以上続いたら、その注目画素のラベルをコーナー、閾値未満なら、非コーナーのラベル付け



↓
16画素の三値と注目画素のラベルを用いて、決定木を構築

推論

学習済みの決定木に注目画素の周囲16画素の特徴ベクトルの入力

↓
決定木をトラバーサルし、最終的なたどり着いた先のクラスに分類

右の例では、
1は注目画素より明るいから左へ。

↓
5は注目画素のより暗いから右へ。

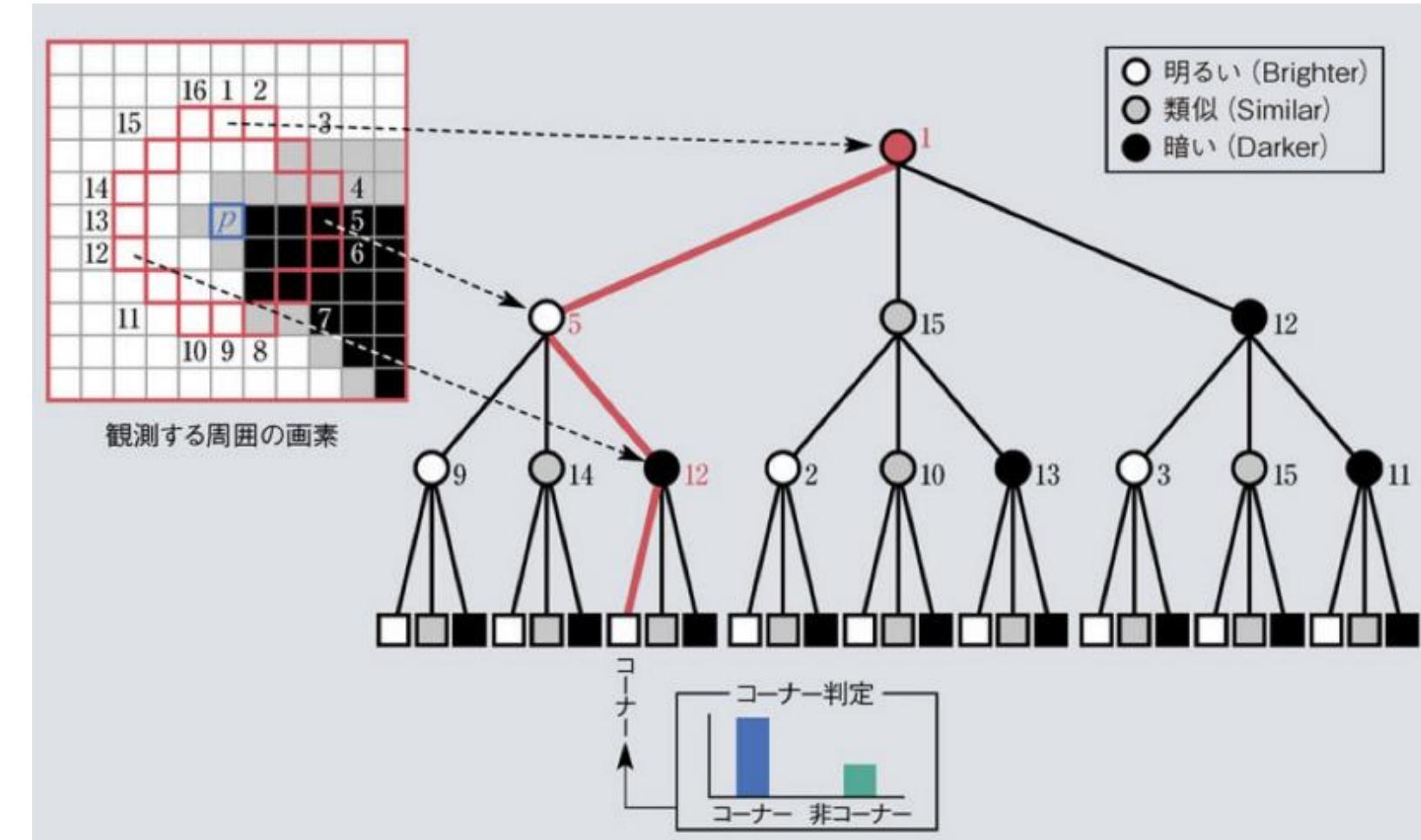
↓
12は注目画素より明るいから左へ。

↓
コーナーと推論

3画素のみを比較すればよいため高速。

FAST のコーナー検出

注目画素の周囲16画素が注目画素より明るいかどうかによって、決定木で高速分類



DoG画像を用いた特徴点とスケールの検出

スケール

ある特徴点を表す範囲。ボケの度合 σ で表される。
ある特徴点を最もよく表すスケールは、付近の物体の大きさに依存するので、
拡大・縮小によらず、同じ範囲となる。
言い換えると、スケールとはガウシアンフィルタのぼかし範囲 σ [pix]なので、
拡大率の違う二つの画像間で対応する特徴点のスケールは、
拡大した画像の方が拡大率の分だけ大きい

DoG画像

異なるスケールの画像の差分(ノイズを除去した勾配画像LoGの近似)

DoG画像を用いて、特徴点とその特徴点を最もよく表すスケールを同時に抽出

注目画素を、その前後のスケールの画像を含む26近傍と比較

↓
注目画素が最大or最小 = 最も特徴がある

↓
特徴点候補とする

↓
その時のスケールをその特徴点を最もよく表すスケールとする

↓
これを全スケール、全画素に対して行う

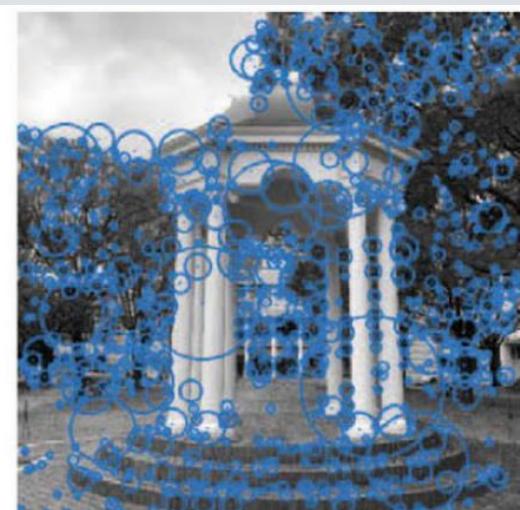
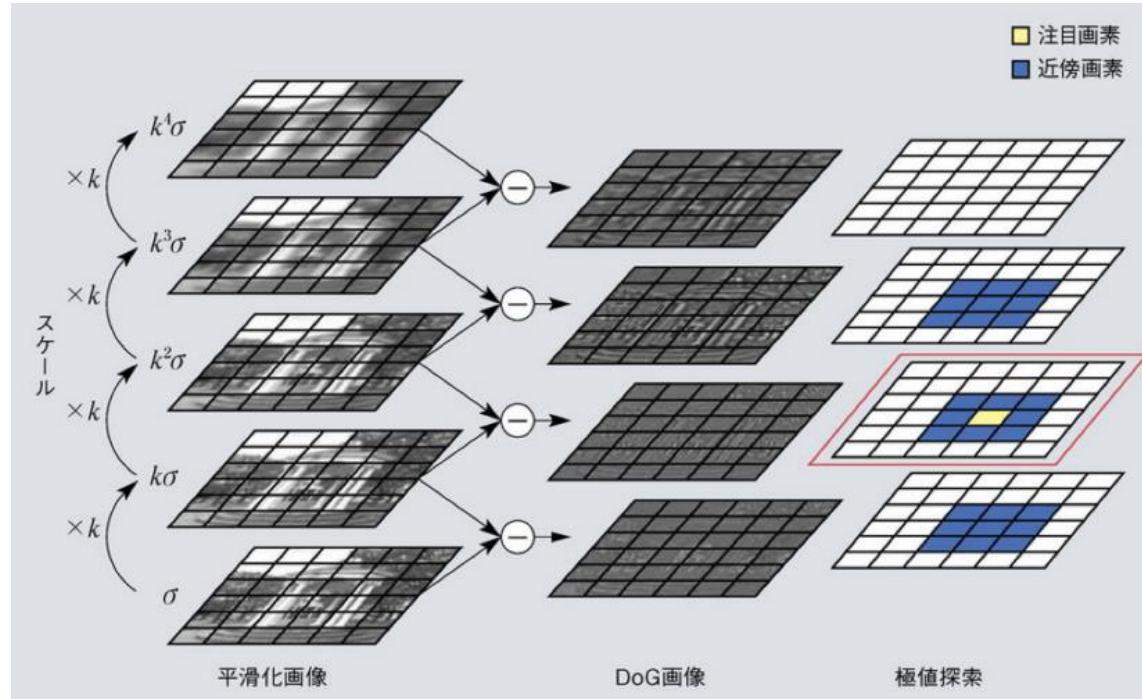
↓
エッジ上の点は一意にマッチングができないため除去

↓
サブピクセル位置推定

↓
その位置で新たにDoG画像の出力

↓
閾値未満なら大きな特徴ではないため除去

この手順でマッチングしやすい特徴点のみを抽出



SIFTとオリエンテーションの検出

SIFT

拡大・縮小・回転がある画像同士でも同一となる特徴量の一つ

オリエンテーション

特徴点領域の勾配方向の代表値。画像座標軸x,yではなく、物体の向きに依存する量のため、画像を回転させても同一の値となる。

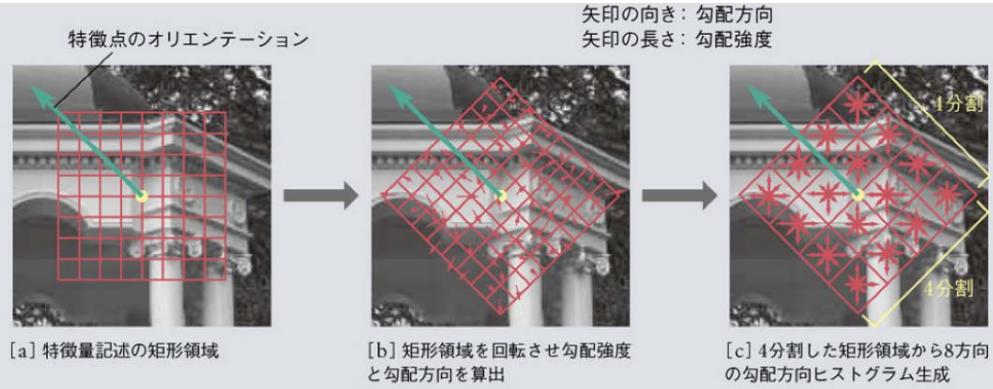
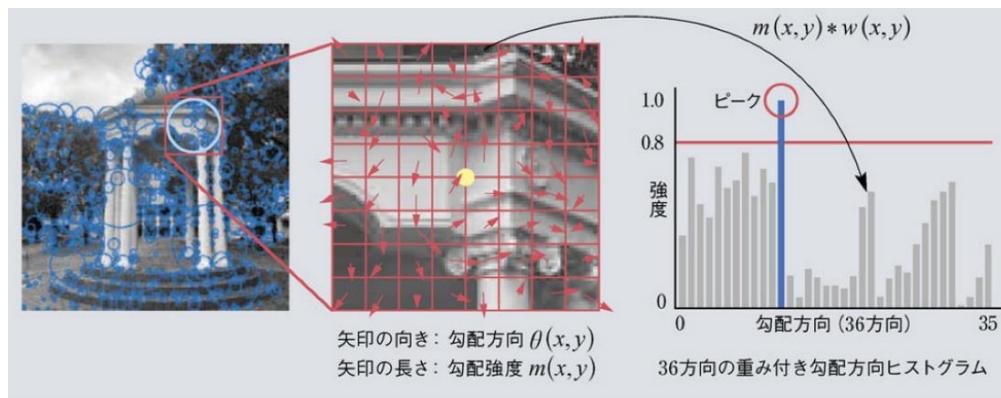
オリエンテーションの導出

特徴点を最もよく表すスケールに合わせた矩形領域の平滑化画像の各画素の勾配を求める。

その方向を36方向に量子化

勾配方向を横軸、その方向の勾配強度を特徴点からの距離の重みづけをして、積算したグラフ(ヒストグラム)を作成し、正規化

強度の積算が0.8を超える方向をオリエンテーションとする。



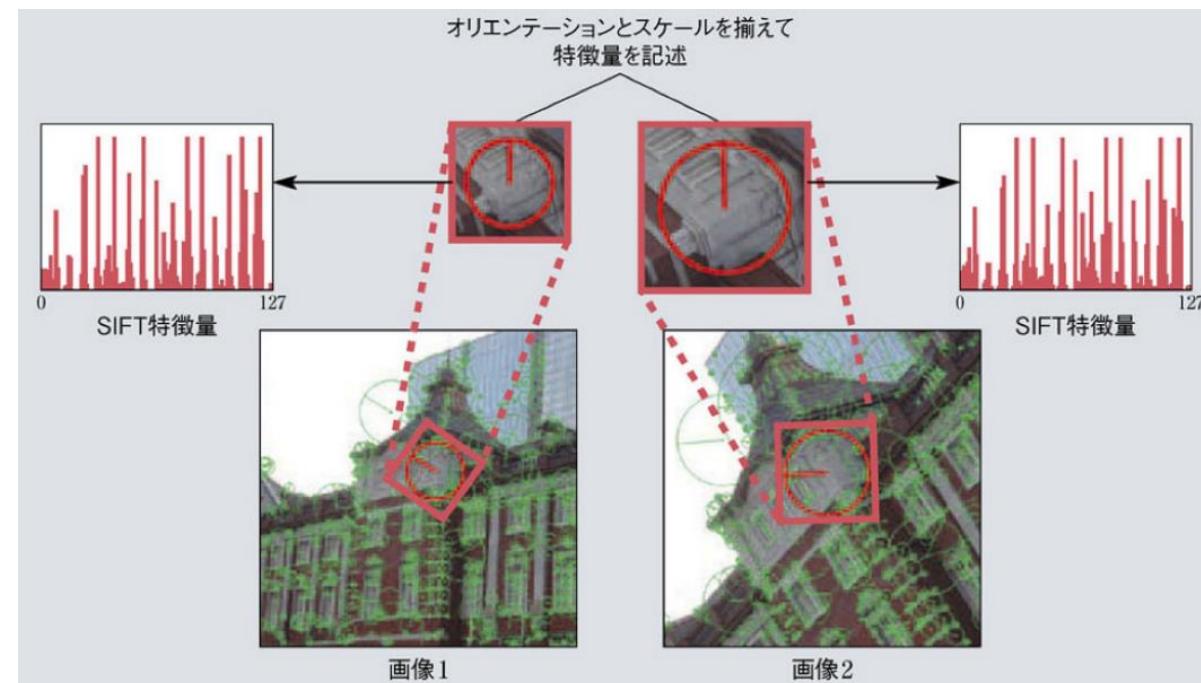
SIFTの導出

矩形領域をオリエンテーションに合うように回転

↓
4×4ブロックに分割

各ブロックx,yに平行な方向を含む8方向のヒストグラムを作成

↓
得られた $4 \times 4 \times 8 = 128$ 次元量がSIFT



二值特徴量

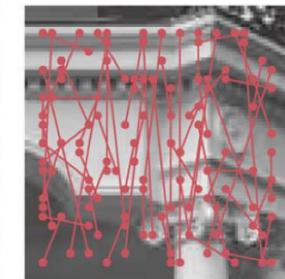
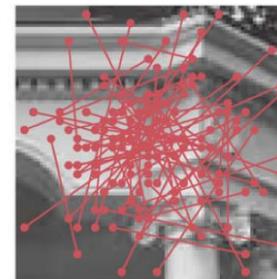
SIFTに比べ、低容量かつ高速なスケール不变特徴量
特徴量を2値ベクトルで表現する手法

BRIEF

ランダムに選ばれた
二点の画素値の
差の符号化

ORB

BRIEFにおいて、
サンプリングペアを
教師なし学習で決定



マッチング

マッチング

異なる画像間で検出された各特徴点の特徴量を比較し、
画像間の対応付けをすること

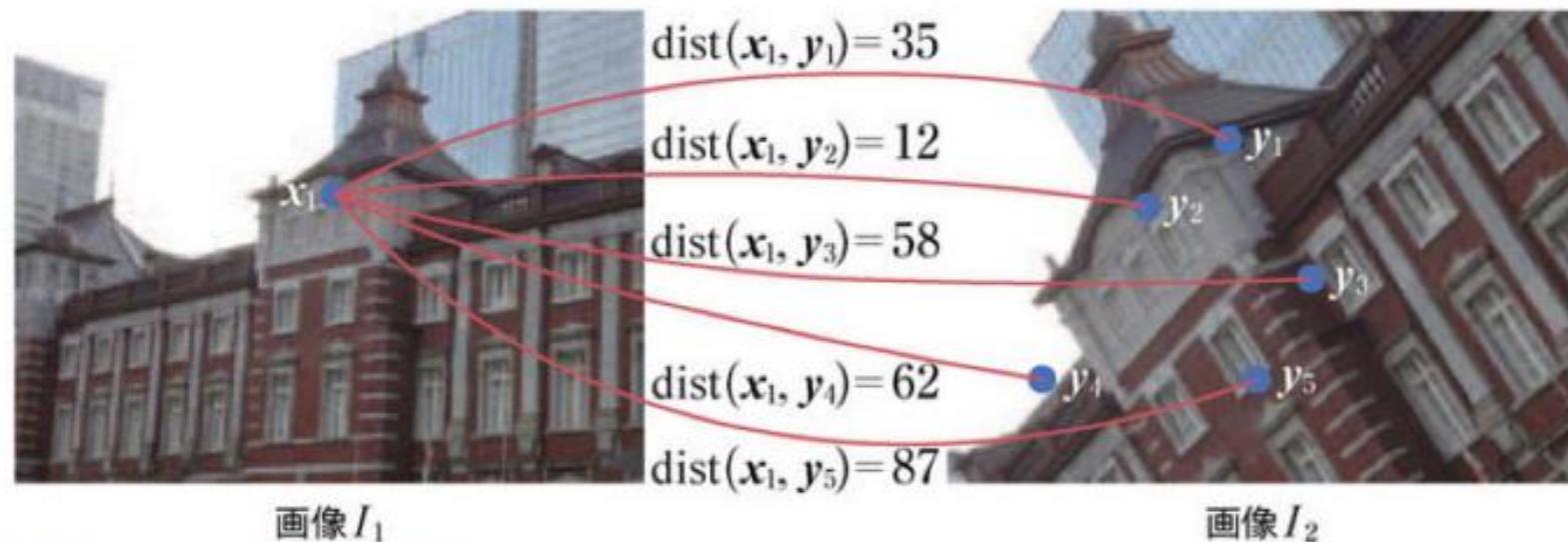
128次元のSIFT特徴量同士のユークリッド距離を
特徴点の全ての組み合わせに対して行う

$$\text{dist}(x, y) = \sum_{i=1}^{128} (x_i - y_i)^2$$

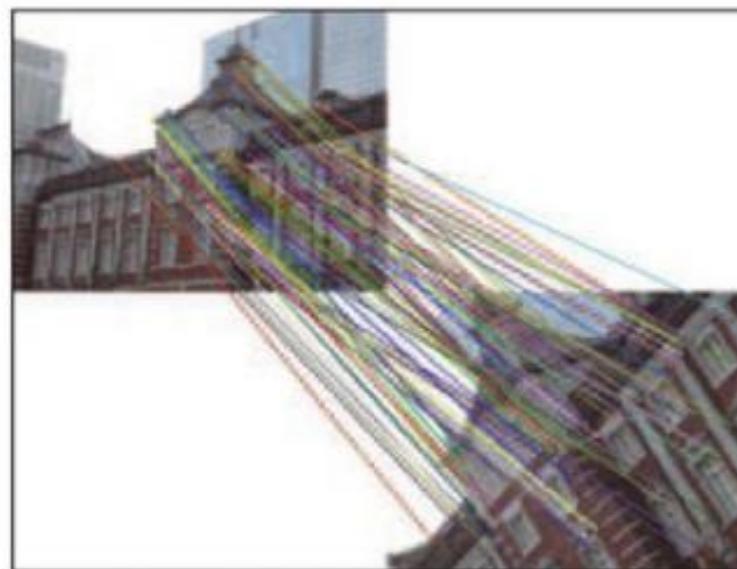
↓
最も近いものと二番目に近いものとの比を算出
比が小さい=1位がダントツで距離が近い

↓
比が閾値未満なら対応点とする

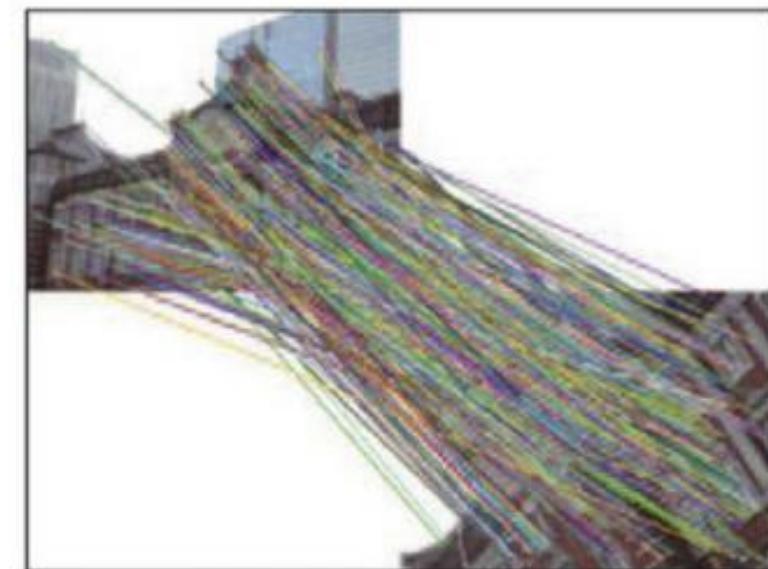
$$\frac{\text{dist}_1(x, y)}{\text{dist}_2(x, y)} < k$$



■図 11.19——対応点の探索



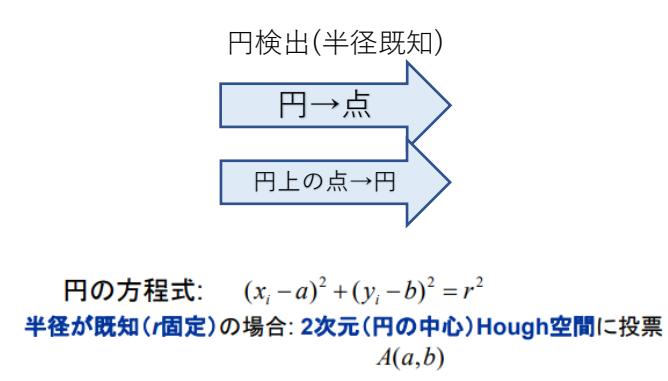
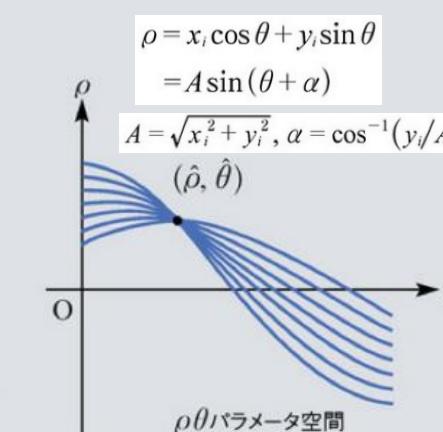
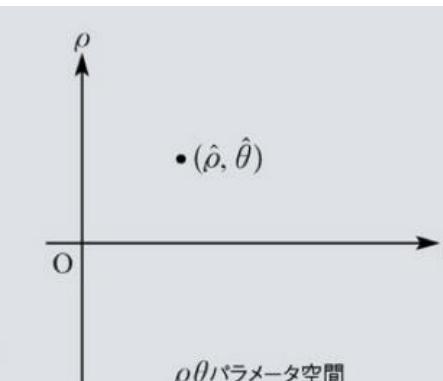
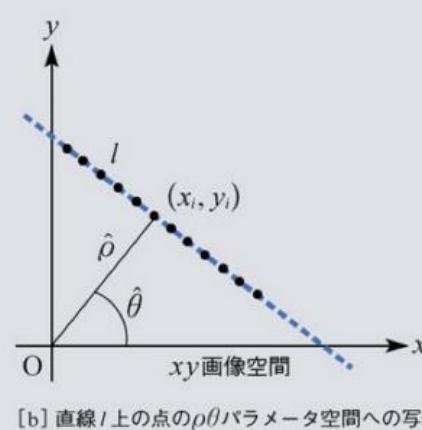
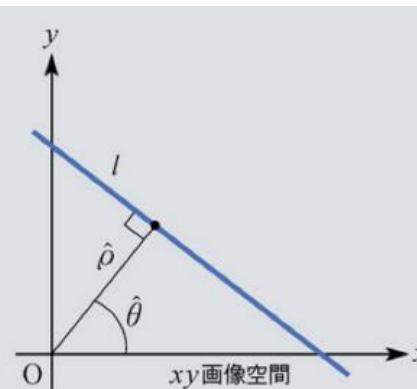
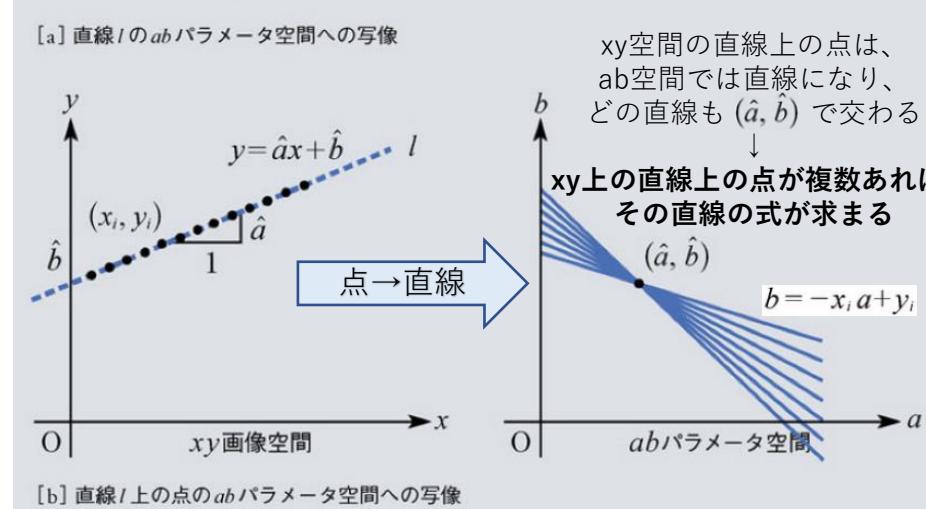
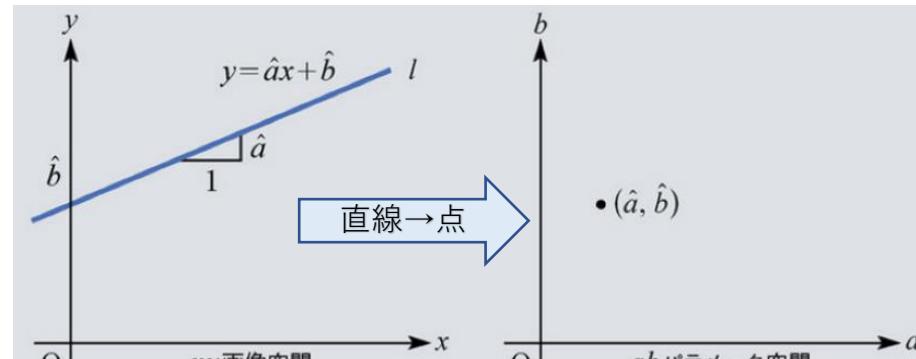
[a] $k=0.25$ の場合の対応づけ結果



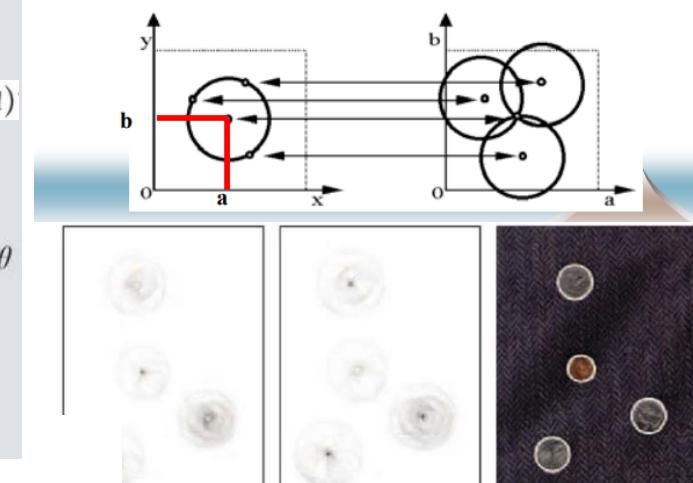
[b] $k=0.8$ の場合の対応づけ結果

■図 11.20——2画像間の対応点とマッチング

ハフ変換 不連続な直線・円の検出



パラメータ式: $a = x_i + r \cos \theta$; $b = y_i + r \sin \theta$



直線検出方法

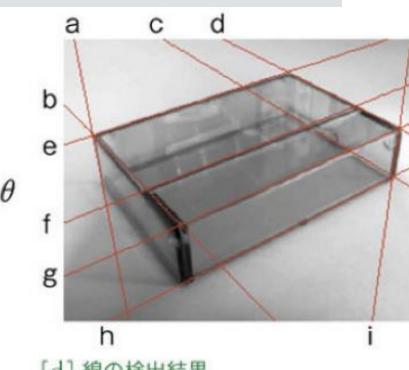
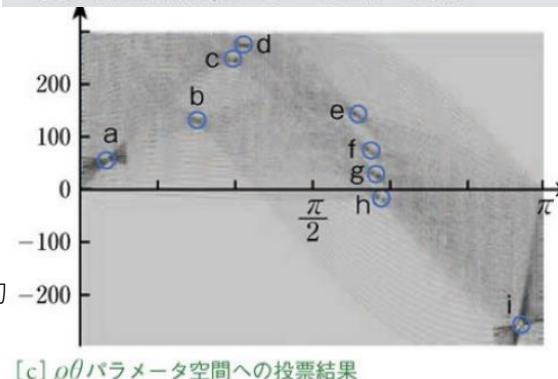
ab 空間を小さなセルに分割(セルが小さいほど高精度かつ要メモリ)

↓
二値画像上の線候補画素を ab 空間に写像したときに
通過するセルをインクリメント(投票)

↓
投票度数が大きなセルの座標が直線の傾きと切片

ab の取りうる範囲は無限大となるため、かわりに $\rho\theta$ を用いるのが一般的

$$\hat{\rho} = x \cos \hat{\theta} + y \sin \hat{\theta}$$



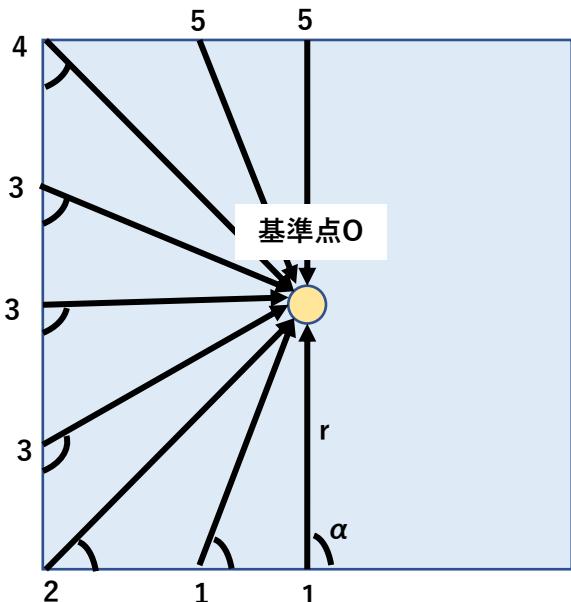
ントコイン(銅)に対する投票結果
[c] 25セントコイン(銀)に対する投票結果



一般化ハフ変換 パラメータで表せない不定形図形の検出

一般化ハフ変換

任意の幾何学図形は形状とポーズ(平行移動(tx,ty)、回転角度θ、スケールs)の組み合わせで表現できる。通常のハフ変換では形状とポーズを同時に表現している。例えば、円のパラメータは、(a,b,r)三つのうちどれか一つではなく、全てによって形状も回転もスケールも表している。一般化ハフ変換では形状とポーズを別々に考える。形状は事前に形状定義表で、ポーズは(tx,ty,r,θ)の4次元パラメータ空間の交点で求める。

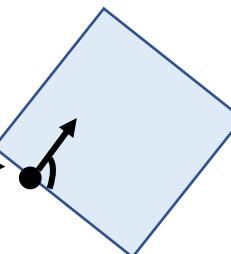


形状定義表

i	エッジ法線方向	基準点方向ベクトル(r, α)
1	90°	(r, 90°) (2/√3 r, 60°)
2	45°	(√2 r, 45°)
3	0°	(2/√3 r, 30°) (r, 0°) (2/√3 r, -30°)
4	-45°	(√2 r, -45°)
5	-90°	(2/√3 r, -60°) (r, -90°)

投票方法

対象画像上の特徴点のエッジ法線方向を求める。



$$t_x = x_j + sr_j \cos(\alpha_j + \theta) \quad t_y = y_j + sr_j \sin(\alpha_j + \theta)$$

すべての(θ, s)の組み合わせに対して(tx,ty,θ,s)に投票する

投票度数の高いポーズパラメータによって図形の検出が可能

ランダム化ハフ変換(直線の場合)

より高精度かつ高精度かつ効率的なハフ変換

特徴点をランダムに二点選択
↓
その二点を通る直線を二次元パラメータ空間に投票
↓
ノイズより、真の直線上の二点が
選ばれる可能性が高いため、高精度
↓
あらかじめランダム抽出の回数を
決めておけるので効率的

n個のパラメータの場合はn個の特徴点を選び、
n個の式を連立させてn個のパラメータを求め、
同様に投票

顕著性マップ

特徴統合理論

人が物体を認識するとき、形状、色、明るさなどの簡単な情報に分解し、その情報を組み合わせて認識している。

注目度

人は、視界から重要なものの、見るべきものを取捨選択している。
注目度は人が注視しやすいかの指標。

注目度の高い画素には高い画素値を割り当てた画像。特徴統合理論に基づき、注目度は画素値、色空間、勾配方向のそれぞれの特徴マップから、異なるスケールのガウシアンフィルタを用いて算出。

画素値の特徴マップ

$\sigma=2,3,4,5,6,7,8$ について
各単色画像ガウシアンフィルタを適用
↓
それらの平均をとる

$$I(\sigma) = (r(\sigma) + g(\sigma) + b(\sigma))/3$$

↓
差分を足し合わせる

$$I = \sum_{\sigma \in S} \sum_{\delta \in N} (I(\sigma) - I(\sigma + \delta))$$

勾配方向の特徴マップ

0~135°まで45°刻みの4方向に対して
ガボールフィルタを適用
↓
差分を足し合わせる

$$O = \sum_{\sigma \in S} \sum_{\delta \in N} \sum_{\theta \in \Theta} (O(\sigma, \theta) - O(\sigma + \delta, \theta))$$

色空間の特徴マップ

R,G,B,Yを算出

$$\begin{aligned} R(\sigma) &= r(\sigma) - (g(\sigma) + b(\sigma))/2 \\ G(\sigma) &= g(\sigma) - (r(\sigma) + b(\sigma))/2 \\ B(\sigma) &= b(\sigma) - (r(\sigma) + g(\sigma))/2 \\ Y(\sigma) &= (r(\sigma) + g(\sigma))/2 - |(r(\sigma) - g(\sigma))/2 - b(\sigma)| \end{aligned}$$

↓
人の色の識別に重要な色の差分情報を算出

$$RG(\sigma) = R(\sigma) - G(\sigma)$$

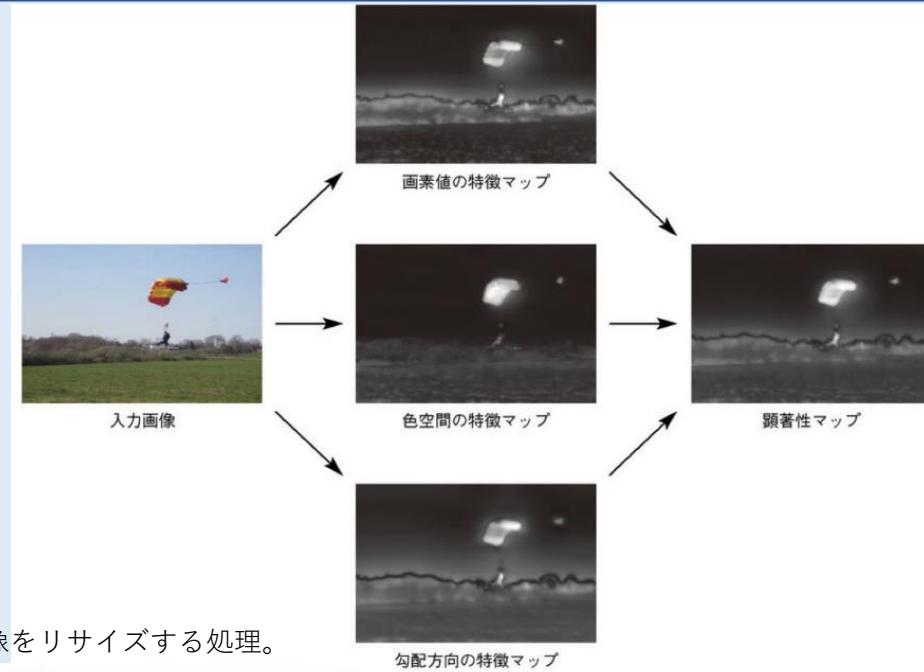
$$BY(\sigma) = B(\sigma) - Y(\sigma)$$

↓
差分を足し合わせる

$$C = \sum_{\sigma \in S} \sum_{\delta \in N} (RG(\sigma) - RG(\sigma + \delta)) + (BY(\sigma) - BY(\sigma + \delta))$$

顕著性マップ

入力画像から顕著性マップを算出するプロセス



シームカービング

画像中の重要な領域のアスペクト比を変更せずに、画像をリサイズする処理。



[a] 入力画像

[b] リサイズ後の画像

[c] シームカービング

■図11.28——顕著性マップを用いたシームカービング処理を行った結果

12. 画像特徴量を
入力としたパターン認識

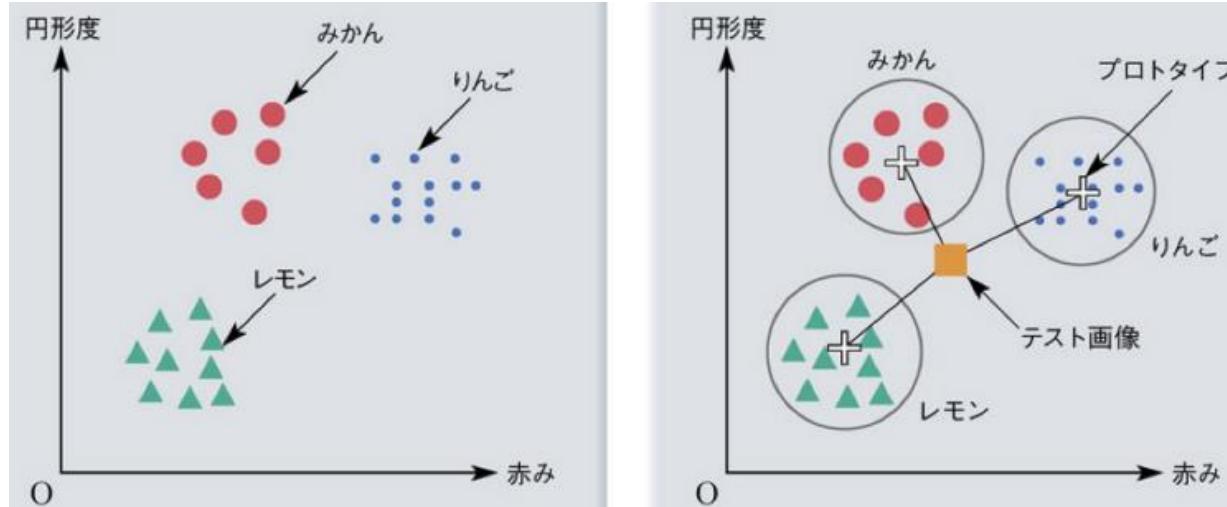
距離計算によるパターン認識1 プロトタイプ法

パターン認識

画像群を同一の特徴を持つクラスに分類すること

プロトタイプ法

プロトタイプ(学習用画像の代表的なパターン、平均値、中央値など)との
特徴空間での距離が最も近いクラスに分類



マハラビノス距離

プロトタイプまでの距離は一般的にユークリッド距離が用いられるが、二つのクラスのプロトタイプまでの
距離が等しい場合は、クラス内の分散が大きい方に分類した方が精度が高いと考えられる。

そういった背景から、クラスの分散を反映させた距離がマハラビノス距離
特徴量がK個ある場合のクラスcへのマハラビノス距離は以下になる。

$$d_m^{(c)}(\mathbf{x}) = (\mathbf{x} - \mathbf{M}^{(c)})^T (\mathbf{S}^{(c)})^{-1} (\mathbf{x} - \mathbf{M}^{(c)})$$

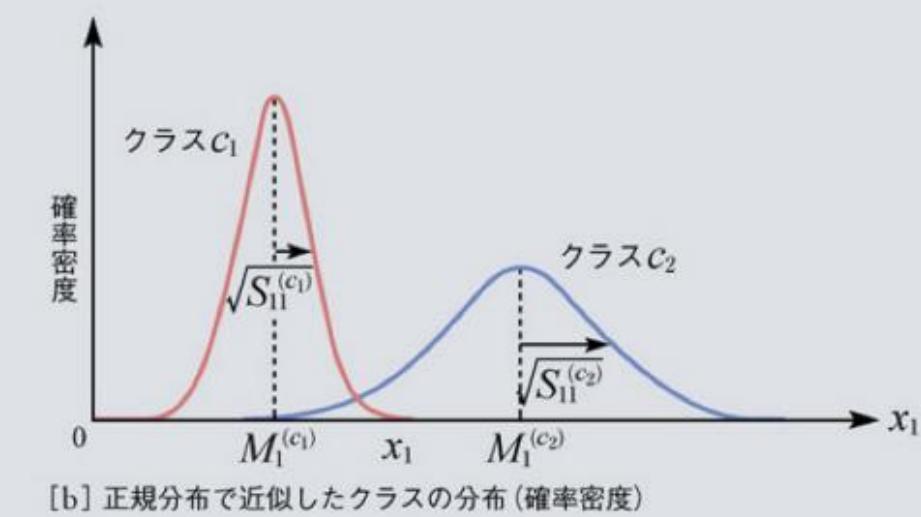
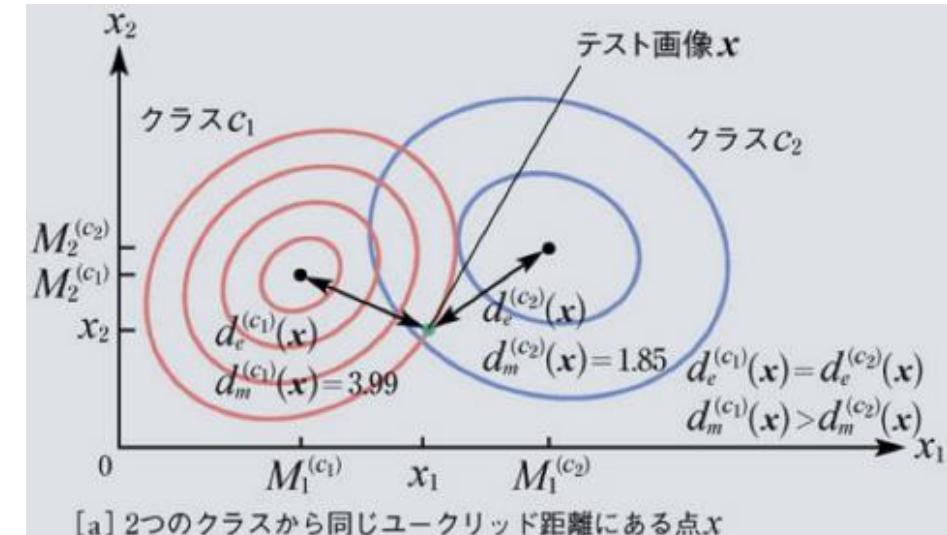
一次元マハラビノス距離は標準化統計量zの絶対値に等しい

特徴量空間での
プロトタイプへのベクトル

$$\mathbf{M}^{(c)} = (M_1^{(c)}, M_2^{(c)}, \dots, M_K^{(c)})^T$$

$$M_i^{(c)} = \frac{1}{N^{(c)}} \sum_{n=1}^{N^{(c)}} x_{ni}$$

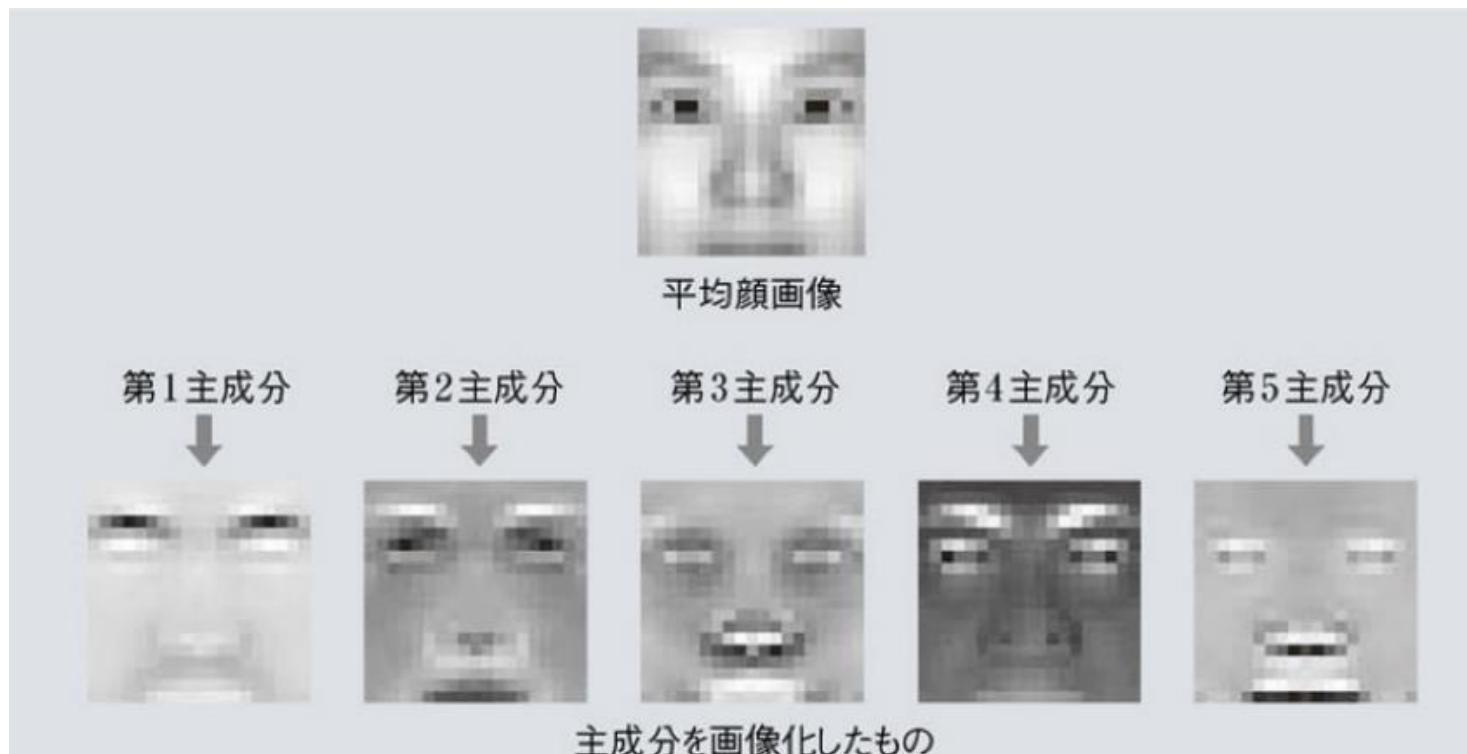
$$\mathbf{S}^{(c)} = \begin{pmatrix} S_{11}^{(c)} & S_{12}^{(c)} & \cdots & S_{1K}^{(c)} \\ S_{21}^{(c)} & S_{22}^{(c)} & \cdots & S_{2K}^{(c)} \\ \vdots & \vdots & & \vdots \\ S_{K1}^{(c)} & S_{K2}^{(c)} & \cdots & S_{KK}^{(c)} \end{pmatrix}$$



主成分分析

特徴空間上で平均値を通り、拡がりの最も大きい方向の直線(第一主成分)を求める
↓
第一主成分に直行し、平均を通り、二番目に拡がりの大きい直線(第二主成分)を求める
↓
以上を繰り返す

平均を通る分散の大きい方向 = 分散共分散行列の固有ベクトルであるから、
固有値が大きい順に第一主成分、第二主成分…(基底と呼ばれる)となる。
この中から大きいものから任意の個数選べば、その個数の次元まで削減できる。
元の特徴ベクトルはこれらの基底の線形結合で表せる。



距離計算によるパターン認識2 NN法・kNN法

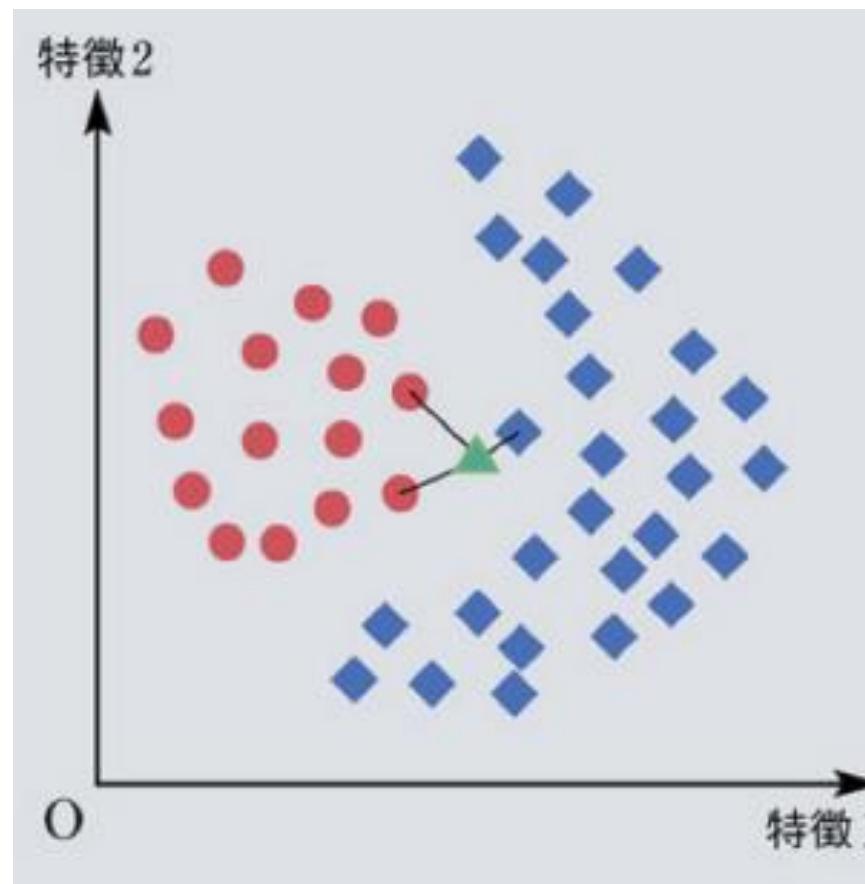
NN法(Nearest Neighbor Method)

特徴空間上で最も距離が近いデータが属するクラスに分類

全ての距離を計算しなくてはならないため、計算時間と要するという欠点がある
下図の▲はこの方法では■に分類

kNN法

特徴空間上で最も距離が近いk個のデータの内、属するクラスが最も多いクラスに分類
下図の▲はこの方法では●に分類



距離計算によるパターン認識3 kd-tree法

kd-tree

特徴量の中央値で全学習データを1クラス1データになるまで分割。NN法に比べて高速な識別が可能。

学習

分割する特徴次元を選択
(右図では特徴1)

↓
中央値となるデータで座標軸に直行する超平面で分割

↓
分割された各空間でも同様に分割を行う

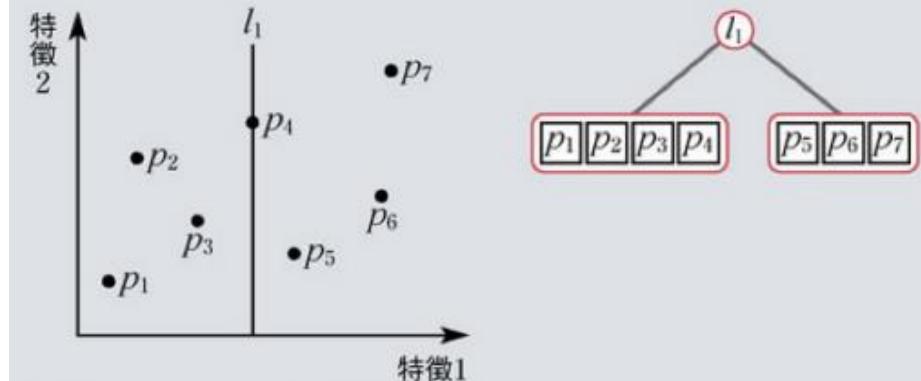
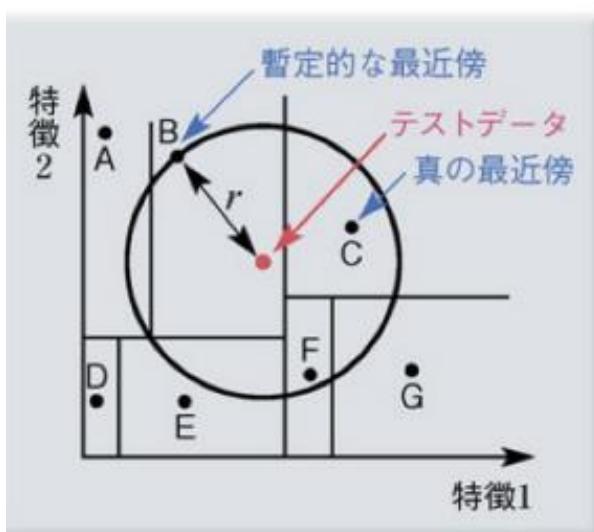
↓
全てのデータが分けられるまで繰り返す

推論

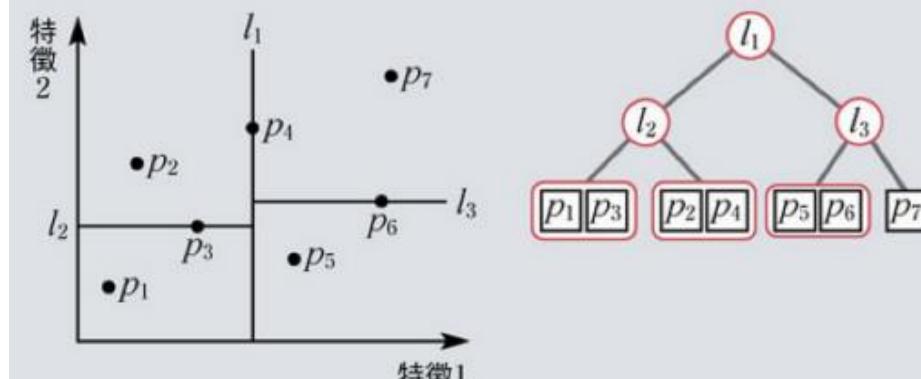
テストデータを構築した気に入力し、たどり着いたクラスの中で
最も近いデータを暫定的に最近傍とする

↓
暫定的な最近傍との距離rの半径内で、他のクラスも含んだ最近傍を探索し、それを真の最近傍とする。

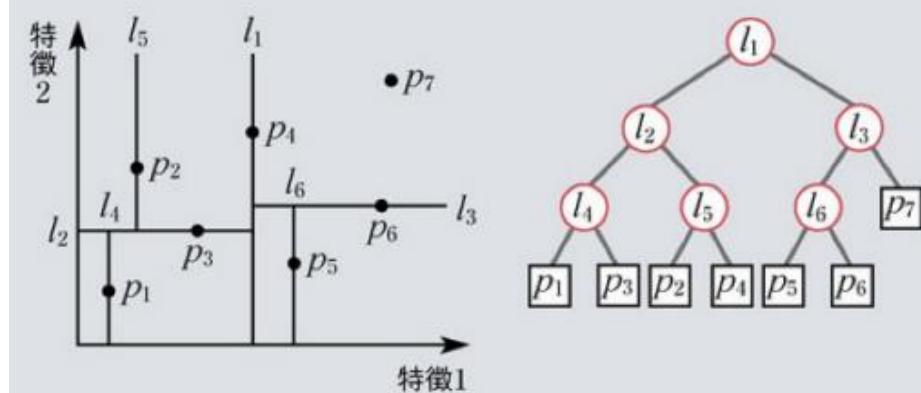
↓
真の最近傍が属するクラスに分類する



[a] 根ノードにおけるデータ集合の分割



[b] 子ノードにおけるデータ集合の分割



[c] 構築された木

距離計算によるパターン認識4 ハッシングによる近似最近傍探索

ハッシュ関数

類似したデータを入力した場合は同じハッシュを出力し、
類似していないデータを入力した場合は異なるハッシュ値を出力する関数。

ハッシングによる近似最近傍探索

登録(学習)データのd次元特徴量からハッシュ関数を用いてハッシュ表を作成。
推論時は、テストデータのハッシュ値が最も近いデータのクラスに分類

$$d \text{ 次元の特徴量 } x = \{x_1, x_2, \dots, x_d\}$$

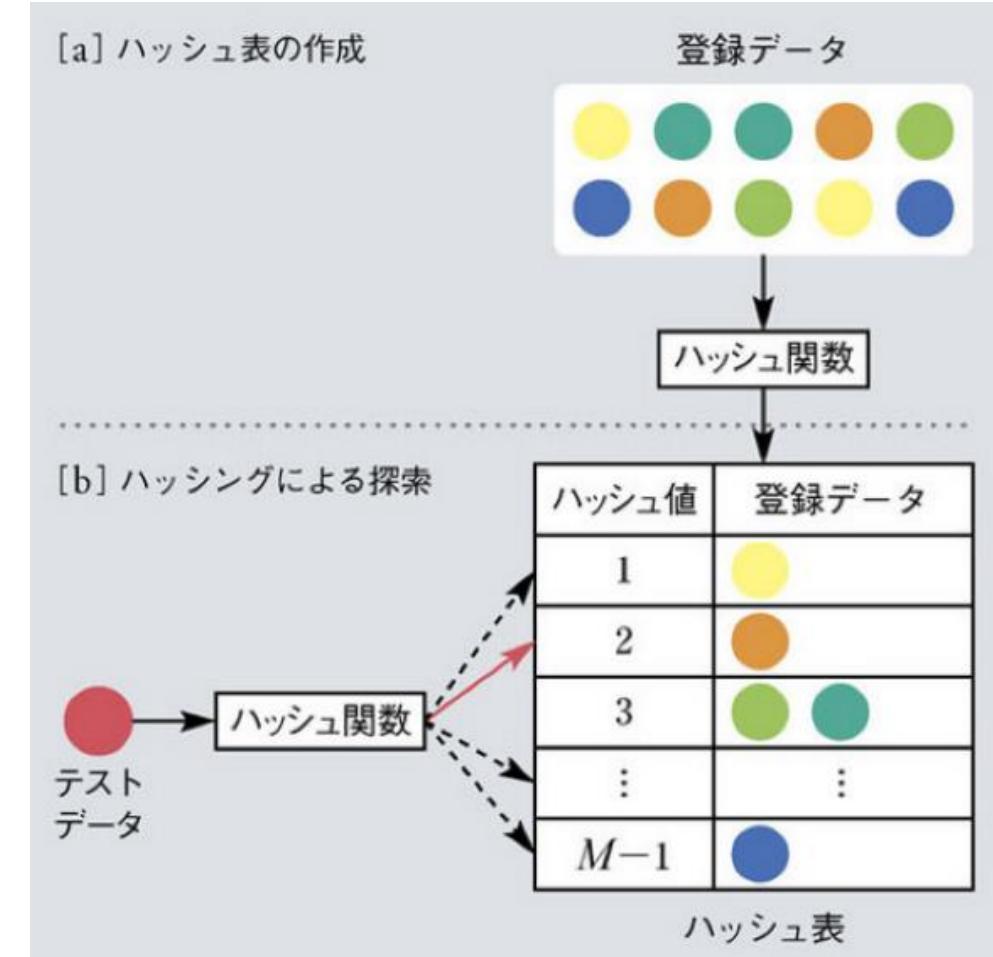
各自特徴量に対し、平均より大きければ1、それ以外は0とする

$$b_i = \begin{cases} 1 & (x_i - m_i < 0) \\ 0 & (\text{それ以外}) \end{cases}$$

$$2 \text{ 値ベクトル } b = \{b_1, b_2, \dots, b_d\}$$

ハッシュ関数の一例

$$h(b) = \left(\sum_{i=1}^d b_i 2^{(i-1)} \right) \text{ 剩余 } \mod M$$



距離計算によるパターン認識5 線形判別分析

線形判別分析

特徴空間上で、クラス間をよく分類できる新たな基底(直線)を選択する手法。
主成分分析と似ているが、主成分分析が全データの次元を減らす基底を選択するのに対し、
線形判別分析はクラスを考慮してよく離す基底を探す。
以下の $\mathbf{S}_b \mathbf{S}_w^{-1}$ の固有値が大きい順に固有ベクトルをクラス数-1個選ぶとこれが基底となる。

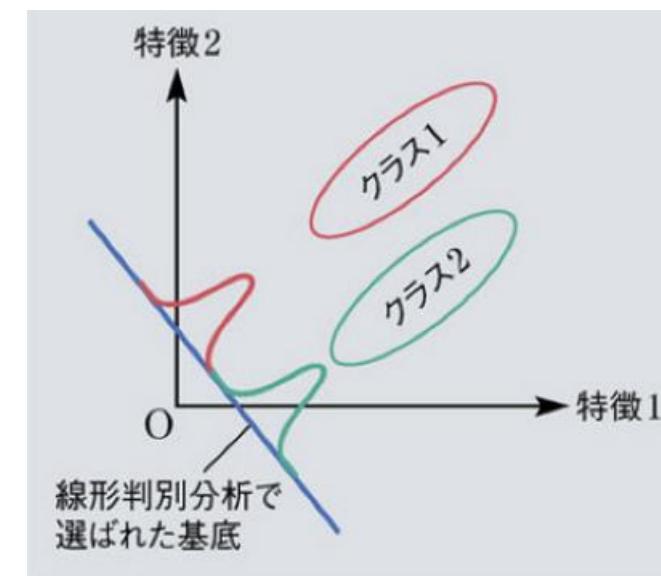
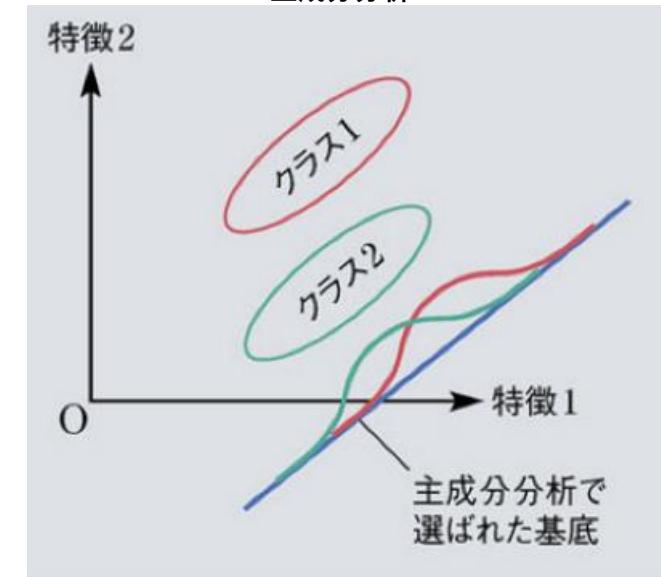
$$\mathbf{S}_b \mathbf{S}_w^{-1} \mathbf{u}_j = \lambda_j \mathbf{u}_j$$

クラス間分散共分散行列 $\mathbf{S}_b = \frac{1}{N} \sum_c N^{(c)} (\mathbf{M}^{(c)} - \mathbf{M})(\mathbf{M}^{(c)} - \mathbf{M})^T$

クラス内分散共分散行列 $\mathbf{S}_w = \frac{1}{N} \sum_c N^{(c)} \mathbf{S}^{(c)}$

クラスcの分散共分散行列 $\mathbf{S}^{(c)} = \frac{1}{N^{(c)}} \sum_{x \in c} (x - \mathbf{M}^{(c)})(x - \mathbf{M}^{(c)})^T$

主成分分析



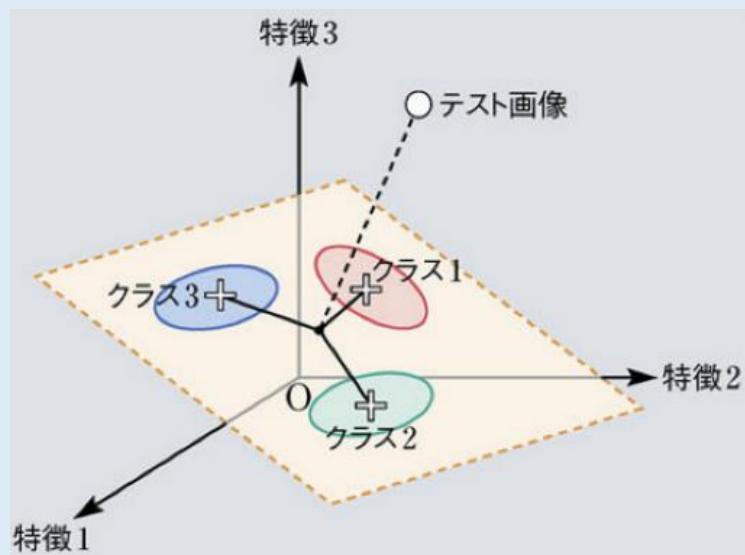
距離計算によるパターン認識6 部分空間法

部分空間法

特徴空間そのものを利用して識別する手法。

SELFIC法

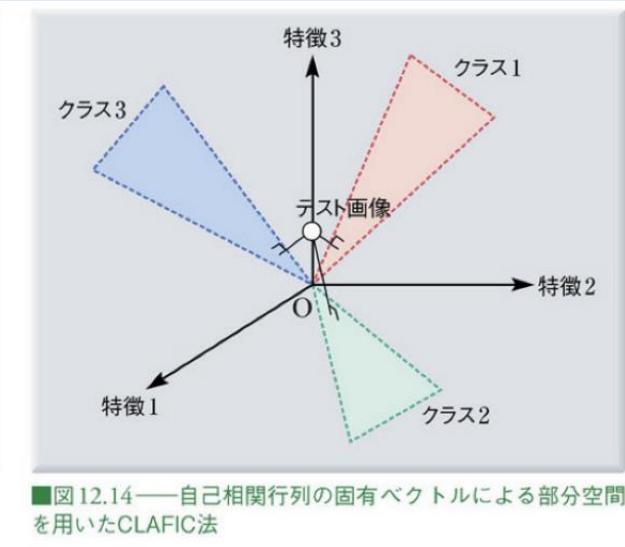
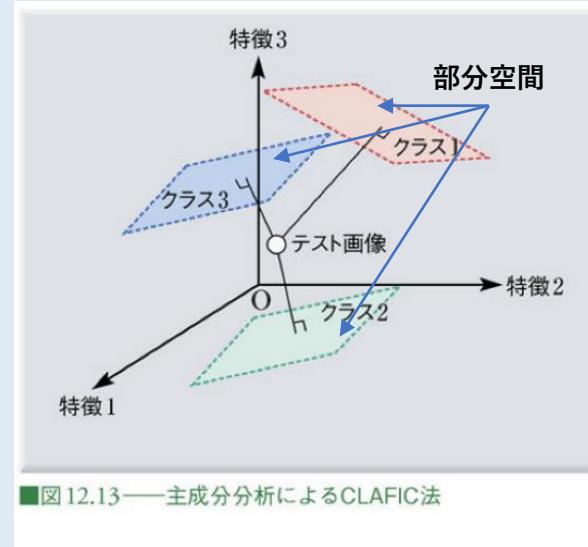
学習用画像に対して特微量次元が多い場合に、
主成分分析によって特微量の次元を削減したのち、
プロトタイプ法などを用いるパターン認識



CLAFIC法

学習用画像に対し、クラスごとに主成分分析や自己相関行列を用いた次元削減を行い、
低次元の部分空間を求める

↓
推論時は、テストデータと各部分空間との距離を求め、最も近い部分空間のクラスに分類



機械学習を用いたパターン認識1 k-meansクラスタリング

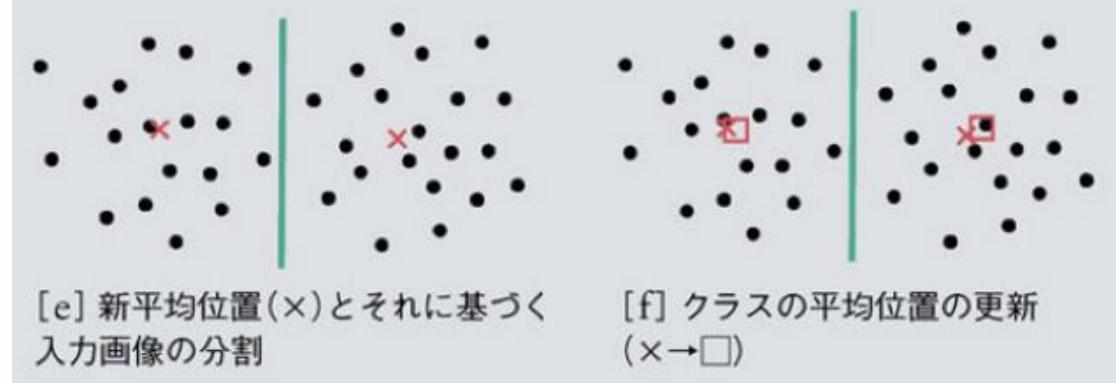
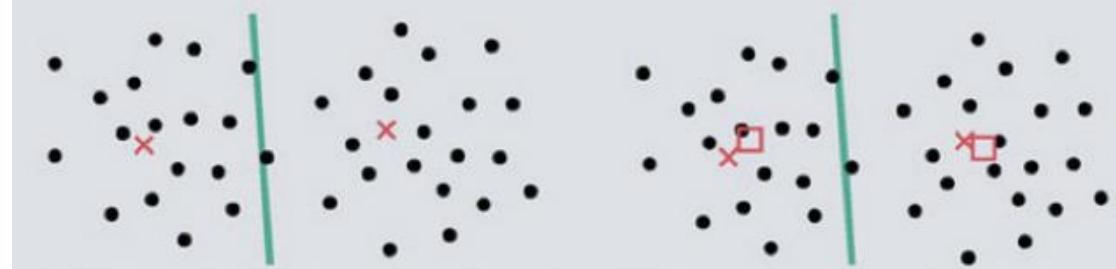
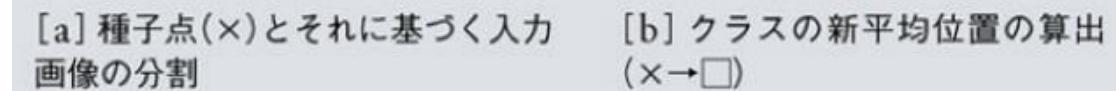
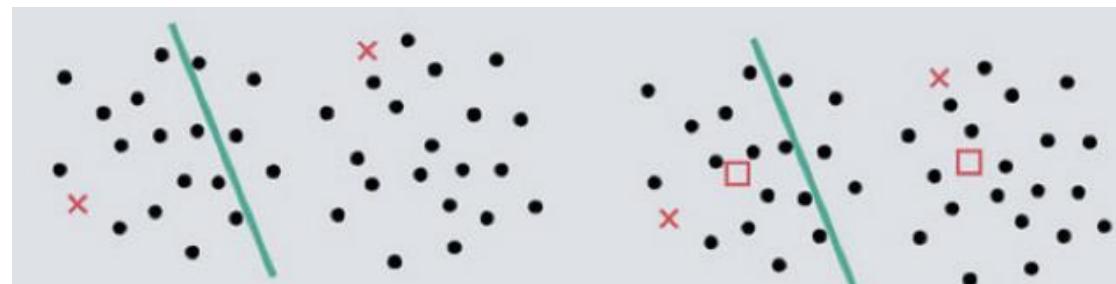
k-meansクラスタリング

kクラスに分類できると仮定して、特徴空間にk個の種子点をばらまき、各画像の特徴点を距離が近い種子点クラスに分類。

↓
分類後の平均点を新たな種子点として、再度距離を計算
↓
最も近い種子点に分類

クラスの移動がなくなるまで繰り返す。

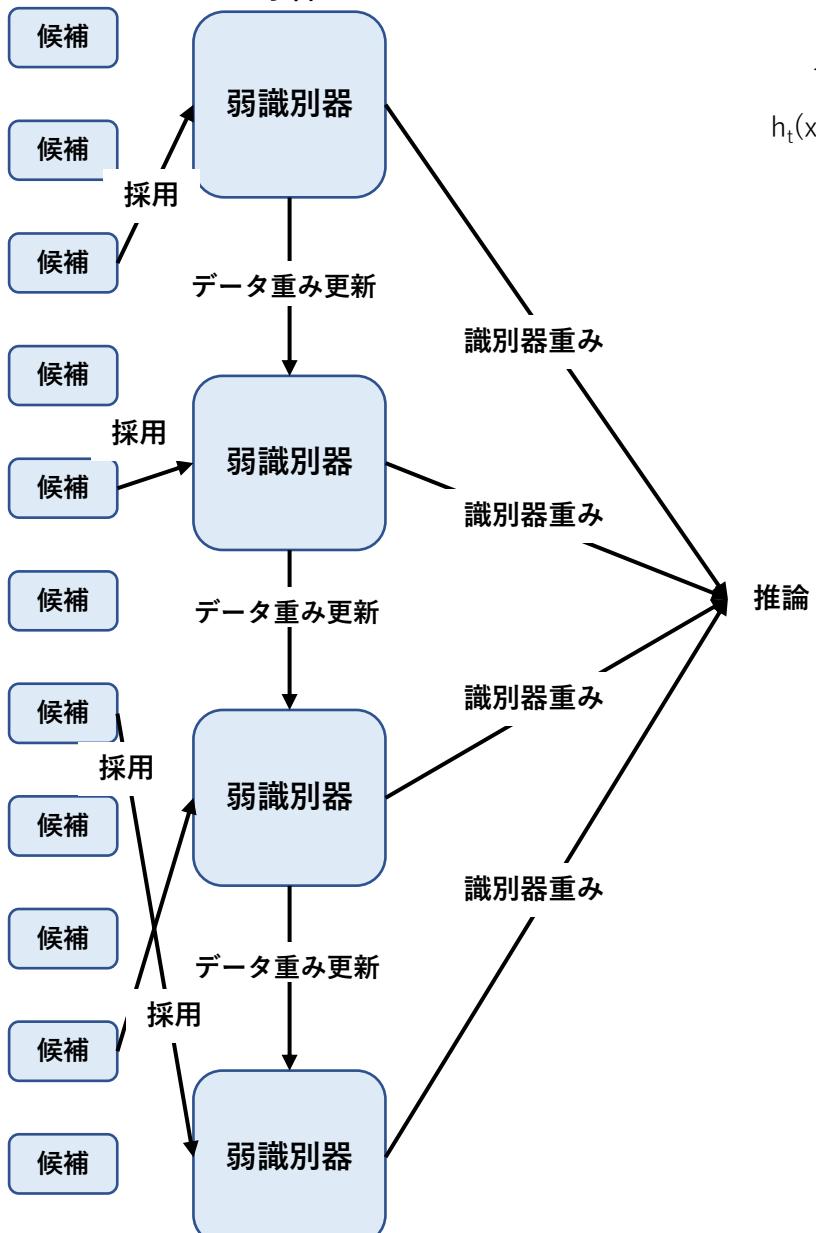
$k=2$



機械学習を用いたパターン認識2 アダブースト

アダブースト

学習 学習サンプルに重みをつけながら直列で複数の弱識別器をアンサンブル学習。得られた複数の弱識別器の出力にも重みをつけ二値分類を行う。



入力データ数N = 3、繰り返し回数(弱識別器数)T = 2の場合、
 $D_{t,n} = t$ 回目の学習における、n個目の入力データの重み
 $h_t(x_n) = t$ 個目の識別機による、n個目の入力データの予測値 = {1, -1}
 $\varepsilon_t = t$ 回目の学習における、エラー率
 $y_n = n$ 番目のデータの正解ラベル = {1, -1}

全弱識別器候補から最もエラー率が小さいモデルを採用
学習サンプルの重み $D_{1,n}$ を初期化

$$D_{1,n} = 1/N = 1/3$$

全データに対するエラー率 ε_1 を算出

$$\epsilon_1 = \sum_{n: h_n(x_{n-}) \neq u} D_{1,n}$$

認分類した入力データの重みの合計

上のエラー率から1回目の弱識別器に対する重み α_1 を算出
エラー率が高いほど大きくなる

$$\alpha_1 = \frac{1}{2} \ln\left(\frac{1 - \epsilon_1}{\epsilon_1}\right)$$

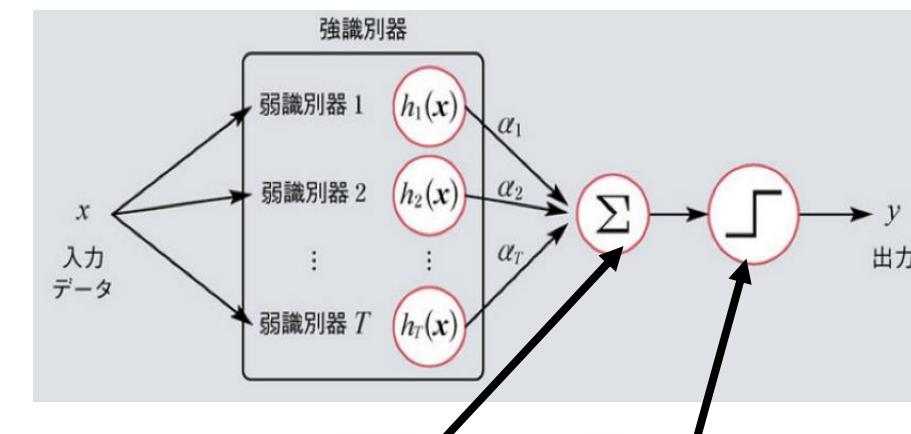
$y_n h(x_n)$ は正解のとき1、不正解の時-1となるから
 $- \alpha y_n h$ は正解の時ほど小さくなることを利用して

$$D_{3,\infty} \equiv D_{1,\infty} \exp(-\alpha_1 y_\infty h_1(x_\infty))$$

重みの総和が 1 になるように正規化

$$D_{2,n} = \frac{D_{2,n}}{D_{2,1} + D_{2,2} + D_{2,3}}$$

以上を繰り返し、弱識別器を2個得る



$$H(x) = \text{sgn}\left(\sum_{t=1}^T \alpha_t h_t(x)\right)$$

$$\text{sign } x = \text{sgn } x = \begin{cases} 1 & x > 0, \\ 0 & x = 0, \\ -1, & x < 0 \end{cases}$$

機械学習を用いたパターン認識3 サポートベクタマシン

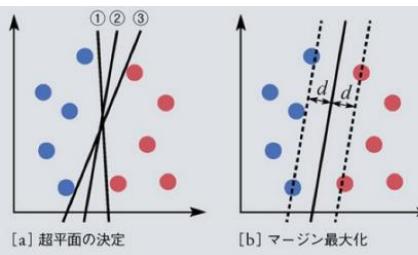
サポートベクタ

特徴空間上でデータを分離する
超平面に最も近いデータ

マージン

超平面とサポートベクタとの距離
学習サンプル x_i から超平面の距離 d_i

$$d_i = \frac{y_i(\mathbf{w}^\top \mathbf{x}_i + b)}{\|\mathbf{w}\|}$$



サポートベクタマシン(SVM)

特徴空間上でマージンを最大化することによって、データを二クラスに線形分類する超平面を導く手法
推論は識別関数 $f(\mathbf{x}) = \text{sgn}(\mathbf{w}^\top \mathbf{x} + b)$ によって超平面の上側と下側で分類する。

ハードマージンSVM

識別誤差を許さない超平面を学習する

学習方法

マージンを最小化するような \mathbf{w}, b を求めることを考える

$$\underset{\mathbf{w}, b}{\operatorname{argmax}} \min_i \frac{y_i(\mathbf{w}^\top \mathbf{x}_i + b)}{\|\mathbf{w}\|}$$

$$= \underset{\mathbf{w}, b}{\operatorname{argmax}} \left\{ \frac{1}{\|\mathbf{w}\|} \min_i \{y_i(\mathbf{w}^\top \mathbf{x}_i + b)\} \right\}$$

↓
超平面の方程式は定数倍の不定性を持つため、
 \mathbf{w}, b の組は一つに定まらないので、
マージン = $1/\|\mathbf{w}\|$ の制約を設ける

$$y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1$$

↓
 $1/\|\mathbf{w}\|$ を最大化するということは以下と等価

$$\underset{\mathbf{w}, b}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{w}\|^2$$

この式を目的関数と呼ぶ

↓
ラグランジュの未定乗数法を用いて最小化問題を解く

$$L(\mathbf{w}, b, a) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^N a_i \{y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1\}$$

↓
 \mathbf{w} と b について微分した後それを 0 と置くことで、
以下を得る

$$\mathbf{w} = \sum_{i=1}^N a_i y_i \mathbf{x}_i \quad 0 = \sum_{i=1}^N a_i y_i$$

↓
これをもとのラグランジアンに代入すると

$$L(a) = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N y_i y_j a_i a_j \mathbf{x}_i^\top \mathbf{x}_j - \sum_{i=1}^N \sum_{j=1}^N y_i y_j a_i a_j \mathbf{x}_i^\top \mathbf{x}_j + \sum_{i=1}^N a_i$$

$$= \sum_{i=1}^N a_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N y_i y_j a_i a_j \mathbf{x}_i^\top \mathbf{x}_j$$

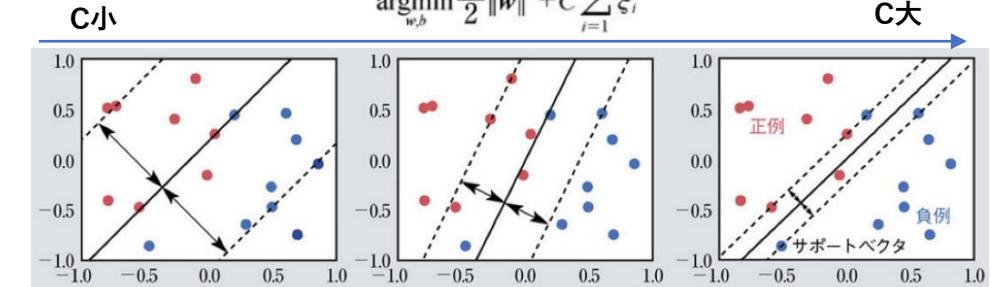
↓
 a を最大化するように SMO アルゴリズムなどを用いて
最適化することにより、 \mathbf{w}, b を得る。

ソフトマージンSVM

線形分離不可能な場合にある程度の誤差を許す超平面を学習
ハードマージンの目的関数にスラック関数 ξ を用いてペナルティ項を加える。
これは、誤分類したサンプルから超平面までの距離の総和である
 C はハードの度合を表すパラメータ。 C が大きいほど、第一項を重視し、
マージンが大きくなり、無限大でハードマージンSVMとなる。

$$\underset{\mathbf{w}, b}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i$$

C 大



線形分離不可能の場合

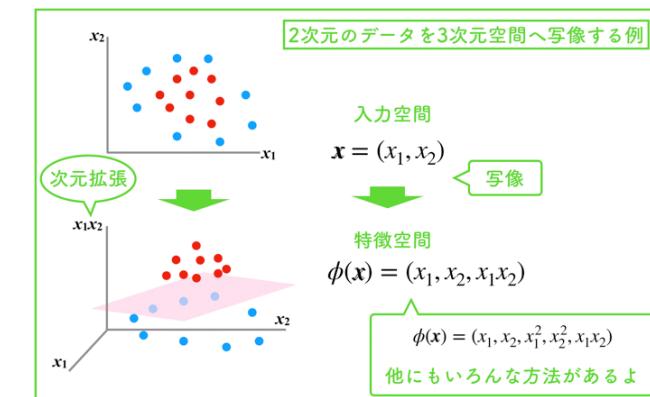
線形分離不可能な場合は非線形写像 Φ によって次元を上げて
分離可能な超平面を探す。しかし、それにより計算量は爆発的に上がる。
ハードマージンと同様ラグランジュ方程式を解いていくと、以下が得られる。

$$f(\mathbf{x}) = \text{sgn}(\mathbf{w}^\top \Phi(\mathbf{x}) + b) \quad f(\mathbf{x}) = \text{sgn} \left(\sum_{i=1}^m (\mathbf{w}_i \langle \Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}_i) \rangle) + b \right)$$

この式には非線形写像された特微量の内積のみで表される。
この内積をもとの特微量から計算できれば、 Φ を求める必要はない。すなわち、

$$f(\mathbf{x}) = \text{sgn} \left(\sum_{i=1}^m (\mathbf{w}_i K(\mathbf{x}, \mathbf{x}_i)) + b \right)$$

この手法をカーネルトリック、 $K(\mathbf{x}, \mathbf{x}_i)$ をカーネル関数と呼ぶ



$$K(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{x}_1^\top \mathbf{x}_2$$

$$K(\mathbf{x}_1, \mathbf{x}_2) = (1 + \mathbf{x}_1^\top \mathbf{x}_2)^p$$

$$K(\mathbf{x}_1, \mathbf{x}_2) = \exp \left(-\frac{\|\mathbf{x}_1 - \mathbf{x}_2\|^2}{2\sigma^2} \right)$$

他にもいろんな方法があるよ

超平面の方程式・点と超平面の距離

n 次元の超平面の法線ベクトルを $w = (w_1, w_2, w_3, \dots, w_n)$ 、

超平面上の二点を $x = (x_1, x_2, x_3, \dots, x_n)$ 、

$p = (p_1, p_2, p_3, \dots, p_n)$ とする、

\vec{PX} と w は直交するから、 $\vec{PX} \cdot w = 0$

$$(x - p) \cdot w = w \cdot x - w \cdot p = wx + w_0 = 0$$

ただし、 $w_0 = -w \cdot p$

点 P を通る超平面の垂線上の点 Q と超平面の距離 $d = |\vec{PQ}|$ を求める。

\vec{PQ} は w と平行であるから、実数 k を用いて $q - p = kw$ と表せる。

変形して、 $p = q - kw$

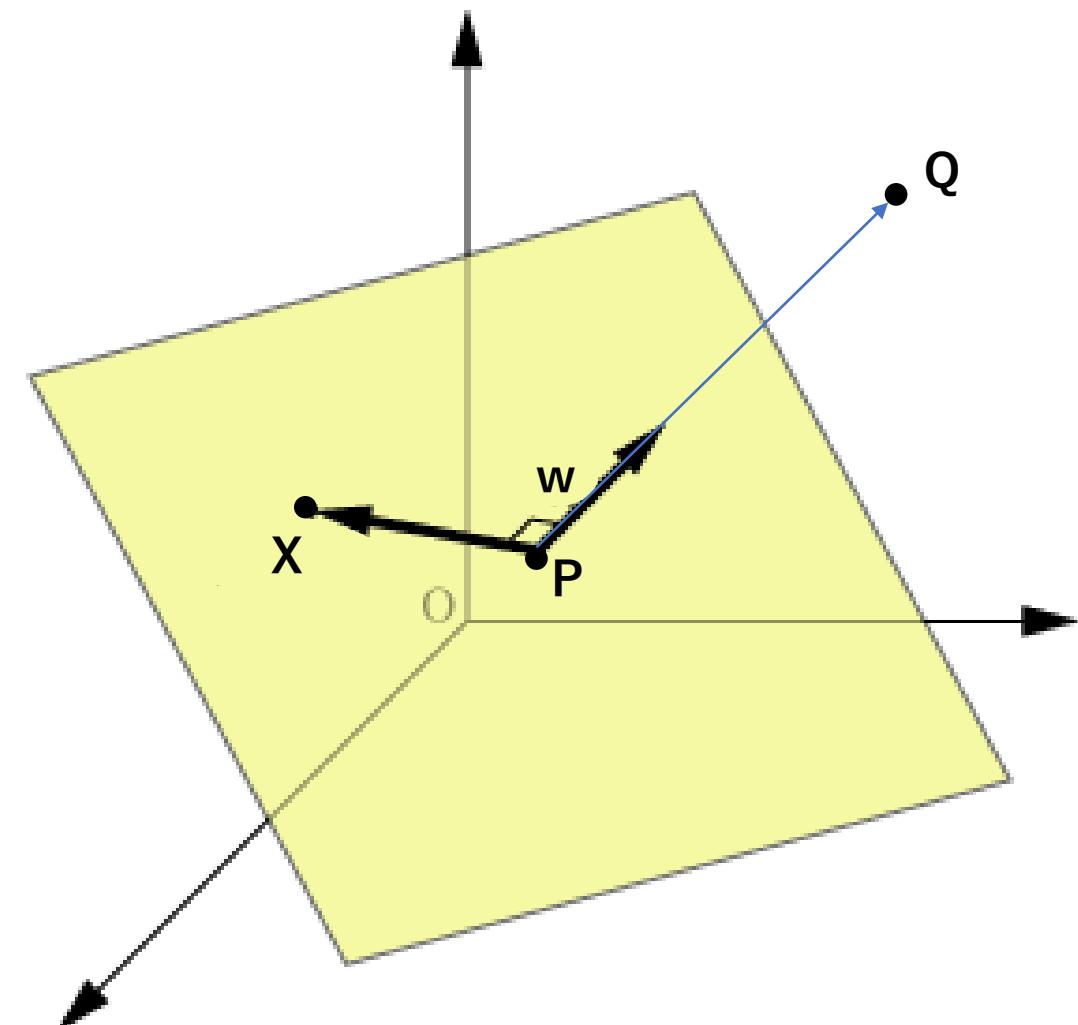
一方、 P は超平面上の点であるから、 $wp + w_0 = 0$

よって、 $w(q - kw) + w_0$

展開して、 $wq - k\|w\|^2 + w_0 = 0$

従って、 $k = \frac{wq + w_0}{\|w\|^2}$

であるから、 $d = |\vec{PQ}| = |k|\|w\| = \frac{|wq + w_0|}{\|w\|}$



ラグランジュの未定乗数法(制約条件が不等式の場合)

- ラグランジュ未定乗数法

- 目的関数の、極値の最大化や最小化を行う際に用いられる
数学的手法

KKT条件 (1/3)

- 制約条件

- q個の等式条件と、r個の不等式条件がある場合を考える
- 目的関数: $f(x)$
- 制約条件: $g_i(x) = 0 \quad (i = 1, 2, \dots, q)$
- $h_j(x) \leq 0 \quad (j = 1, 2, \dots, r)$

- ラグランジュ関数の定義

- 制約条件が2つなので、 λ と μ の2つのラグランジュ乗数を導入

- $L(x, \lambda, \mu) = f(x) + \sum_{i=1}^q \lambda_i g_i(x) + \sum_{j=1}^r \mu_j h_j(x)$
最小化をする場合

KKT条件 (2/3)

- 最適化問題ごとの対応

- 目的関数を最小化したい場合
- $L(x, \lambda, \mu) = f(x) + \sum_{i=1}^r \lambda_i g_i(x) + \sum_{j=1}^q \mu_j h_j(x)$
- 目的関数を最大化したい場合
- $L(x, \lambda, \mu) = f(x) + \sum_{i=1}^r \lambda_i g_i(x) - \sum_{j=1}^q \mu_j h_j(x)$
- この場合は $\mu > 0$ になる
- 制約条件の符号が変わると、 μ の符号も変化する

KKT条件 (3/3)

- 最適性条件

- 通常のラグランジュ未定乗数法と同様の方法で解く
- その時に得られた最適な値 \bar{x} は次のような条件を満たさなければならない
- $\nabla_x L(x, \lambda, \mu) = \nabla f(\bar{x}) + \sum_{i=1}^q \lambda_i \nabla g_i(\bar{x}) + \sum_{j=1}^r \mu_j \nabla h_j(\bar{x}) = 0$
 ∇ : 勾配
各項の微分結果の総和が0

- $h_k(\bar{x}) \leq 0, \mu_k \geq 0,$
- $\mu_k h_k(\bar{x}) = 0$

KKT条件

例題

- 例題

- 関数 $f(x, y) = x + y$ を、
制約条件 $g(x, y) = x^2 + y^2 - 1 = 0$ の下で最大化する
- ラグランジュ関数の定義
- $L(x, y, \lambda) = x + y + \lambda(x^2 + y^2 - 1)$
- 最適化条件より、各変数に対し偏微分をする
- $\frac{\partial}{\partial x} L = 1 + 2\lambda x = 0, \frac{\partial}{\partial y} L = 1 + 2\lambda y = 0, \frac{\partial}{\partial \lambda} L = x^2 + y^2 - 1 = 0$
- これにより、 $x = \pm \frac{1}{\sqrt{2}}, y = \pm \frac{1}{\sqrt{2}}$ が求まるため、
最適な値を求めることができる

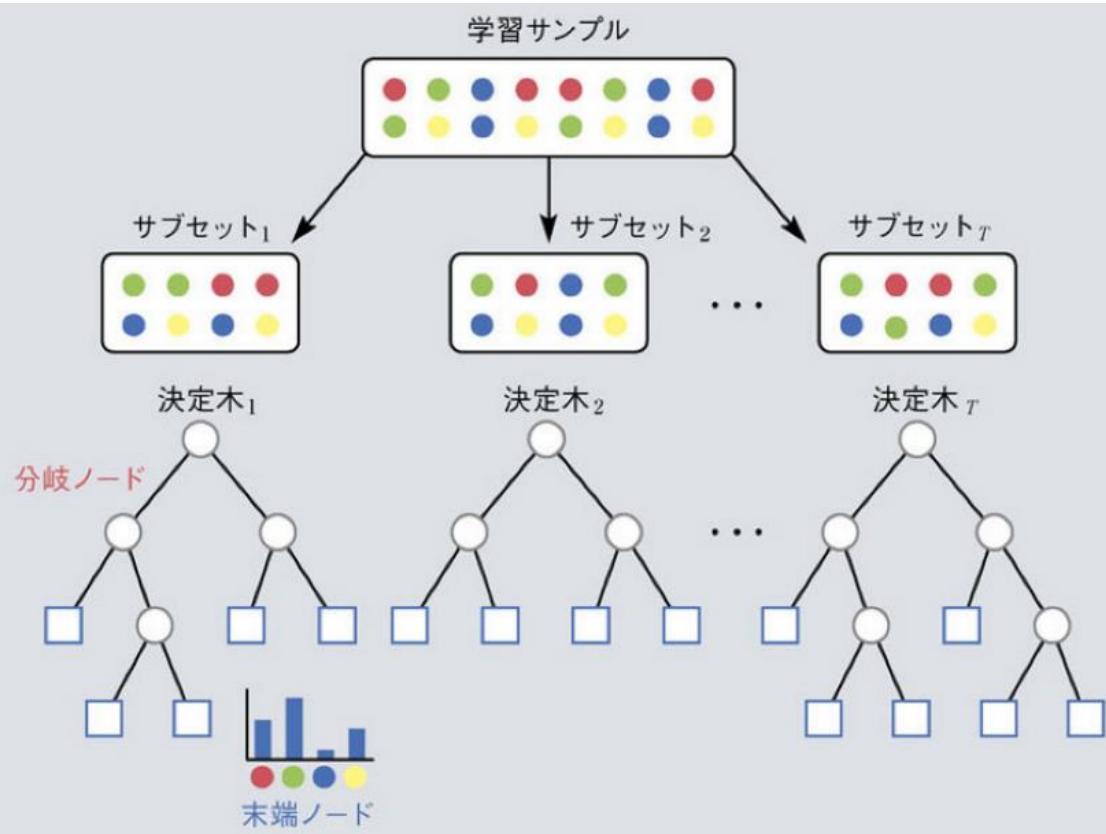
機械学習を用いたパターン認識4 ランダムフォレスト

ランダムフォレスト

データセットはブートストラップ

複数の決定木分析モデルのバギング

出力は各決定木の末端ノードにおけるヒストグラムの平均値

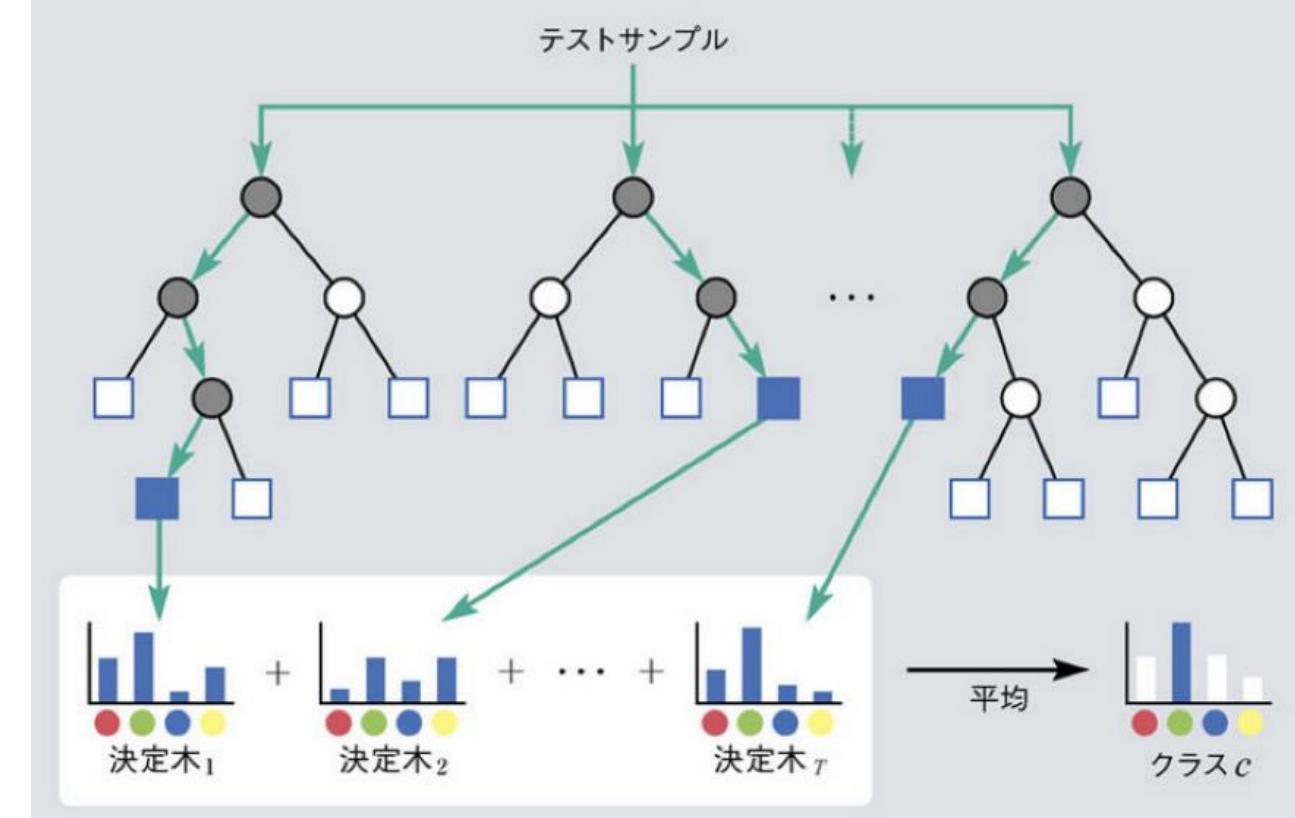


バギング

複数のモデルを並列で学習

ブートストラップ

復元抽出でサブデータセットを作ること



決定木分析

決定木分析

各ノードに対し、特徴量F回、閾値T回、乱数で決定。
その中で**情報利得**が最大になるような特徴と閾値の組み合わせを選ぶ。

情報利得
分岐によるエントロピー減少量

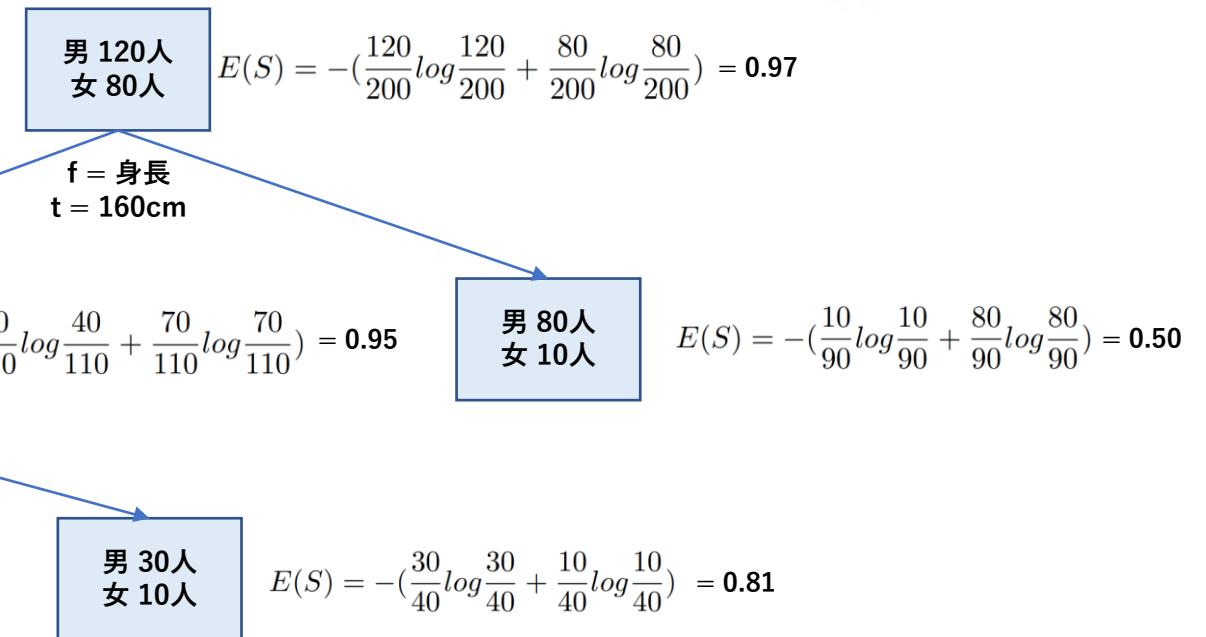
$$\Delta E = E(S) - \frac{|S_l|}{|S_n|}E(S_l) - \frac{|S_r|}{|S_n|}E(S_r)$$

$$\Delta E(S) = 0.97 - \left(\frac{110}{200}0.95 + \frac{90}{200}0.50 \right) = 0.28$$

$$\Delta E(S) = 0.95 - \left(\frac{70}{110}0.59 + \frac{40}{110}0.81 \right) = 0.22$$

エントロピー
異なるクラスのデータがどれだけ混ざっているか。データの乱雑さ。

$$E(S) = -\sum_{i=1}^n p(c_i)\log p(c_i)$$

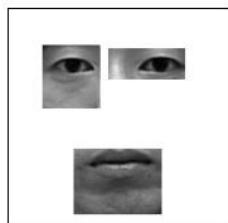


機械学習による画像認識の応用例1 顔検出と顔識別

Haar-like特徴量

顔を見つける着眼点

- ・ 全体を見なくても、特徴点な部分を見れば顔とわかる
- ・ 細かな特徴を見なくとも、大きな明暗差で顔とわかる



特徴的なところだけ着目



大きな明暗差

Haar-like特徴の算出

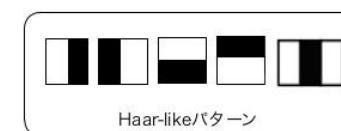
- ・ 2つの領域の輝度差に着目し、局所的なエッジ成分・線成分をとらえる

- ・ 特徴量 $H(r1,r2) = A \text{と } B \text{ の平均輝度の差を算出}$

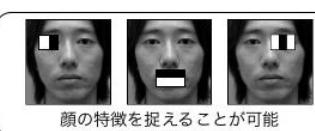


$$H(r1,r2) = \underline{S(r1)} - \underline{S(r2)}$$

領域Aの平均輝度 領域Bの平均輝度



Haar-like/パターン

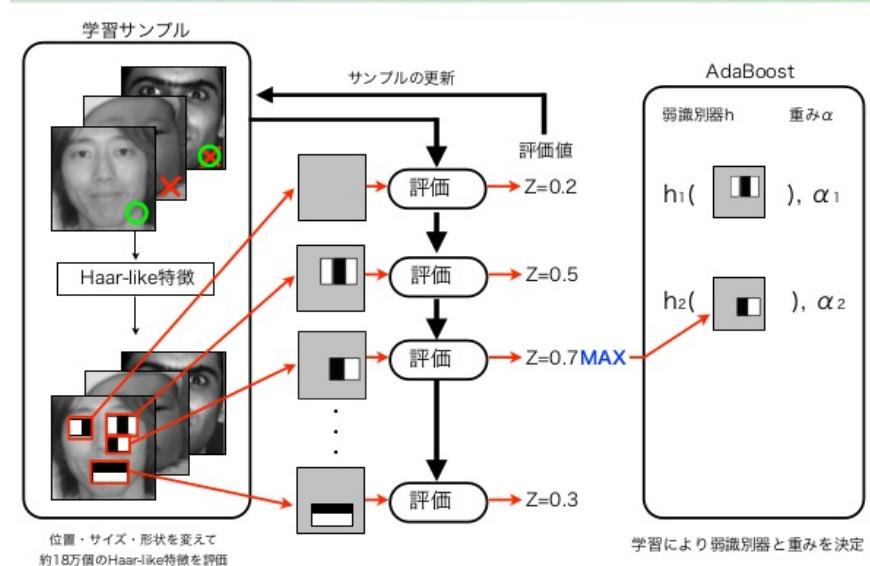


顔の特徴を捉えることが可能

顔検出

パターン、大きさ、縦横比、位置の異なる大量のHaar-like特徴量を用意
これらを弱識別器候補として、アダブーストで学習

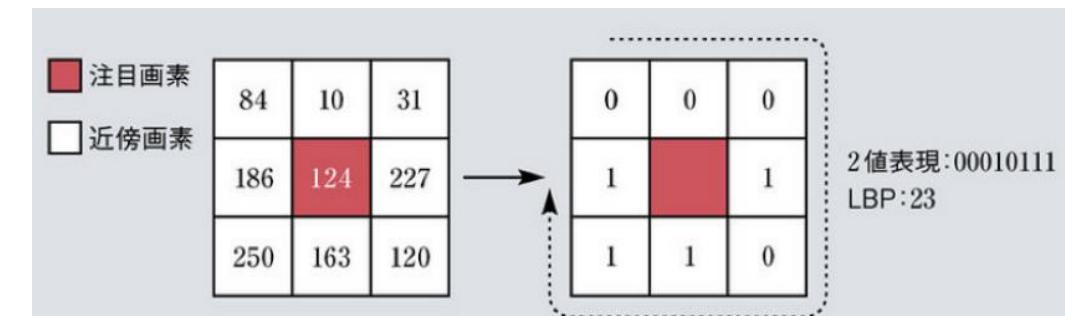
統計的学習手法AdaBoostの学習



LBP(Local Binary Pattern)

注目画素に対し、明るい周辺画素は1、暗い周辺画素は0として符号化。
これを10進数に変換したものをLBPと呼ぶ。

明るさが変化してもその大小関係は変わらないため、照明変動に頑健な局所特徴量。



顔識別

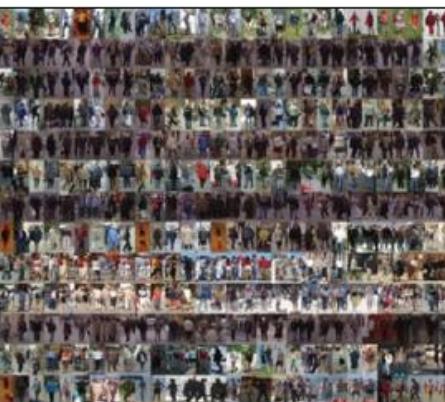
局所領域内における複数のLBPからヒストグラムを作成し、
これを特徴量とした特徴空間にてNN法によって、
最も類似している登録画像を求め、識別する。

機械学習による画像認識の応用例2 人検出

HOGの算出法1

人らしさとは？

- 全サンプルのエッジ画像の平均をとってみると

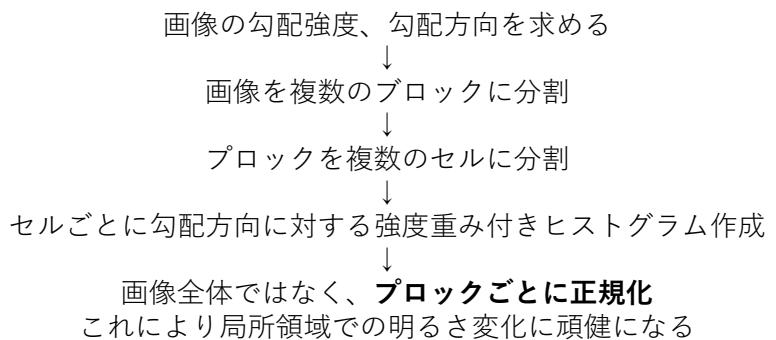


人画像 (約3,000枚)

平均勾配画像

HoG特徴量

人らしさはエッジ(勾配)で決まるという考えに基づいた人らしさの指標



人検出

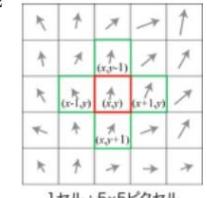
得られたセル数 × 勾配方向数の次元のHoG特徴量空間において、SVMにより人かどうかを識別

1. 輝度の勾配方向と強度の算出

$$\text{勾配強度} : m(x, y) = \sqrt{f_x(x, y)^2 + f_y(x, y)^2}$$

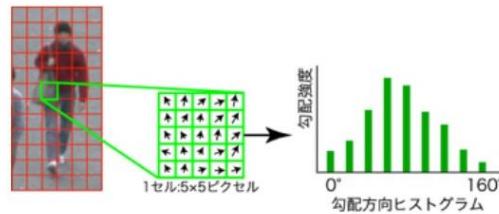
$$\text{勾配方向} : \theta(x, y) = \tan^{-1} \frac{f_y(x, y)}{f_x(x, y)}$$

$$\begin{cases} f_x(x, y) = L(x+1, y) - L(x-1, y) \\ f_y(x, y) = L(x, y+1) - L(x, y-1) \end{cases}$$



2. ヒストグラムの作成

- 0°から180°までを、20°ずつ9方向に分割



HOGの算出法2

3. ブロック領域での正規化

- あるn番目のHOG特徴量について正規化
- 分母は1ブロック($q \times q$ セル)含まれるHOG特徴量の総和
- 1つのセルに対し複数回の正規化

$$v(n) = \frac{v(n)}{\sqrt{\left(\sum_{k=1}^{q \times q \times N} v(k)^2 \right) + \epsilon}} \quad (\epsilon = 1) \quad N: \text{勾配方向数}$$

- 例：30×60の入力画像の場合

- 1セル：5×5ピクセル、1ブロック：3×3セル

$$(4 \times 10) \times (3 \times 3) \times 9 = 3,240\text{個(次元)} \text{のHOG特徴量}$$

正規化回数 1ブロックのセル数 勾配方向数

機械学習による画像認識の応用例3 類似画像検索

BoVW(Bag of Visual Words)

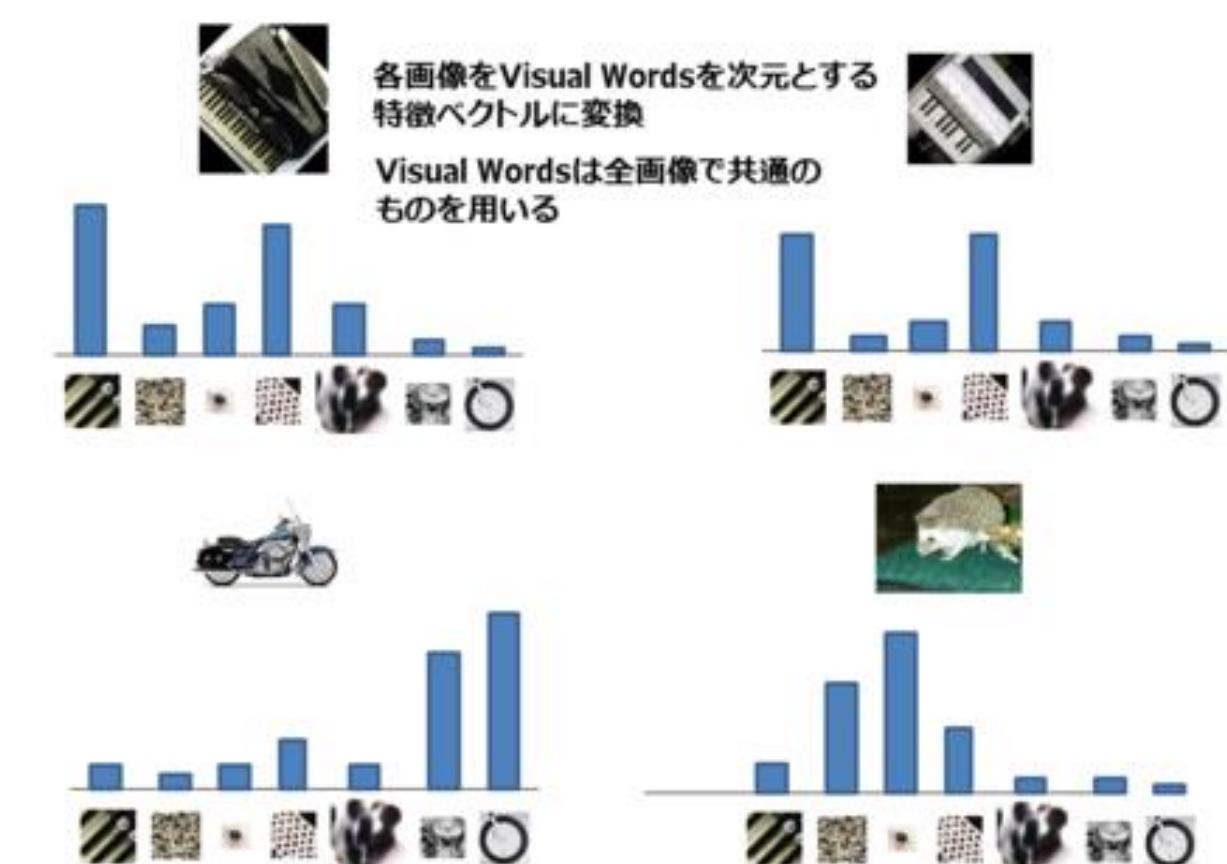
学習
大量の画像を用意
↓
各画像から局所特徴量SIFTを計算
↓
k-means法でSIFTをクラスタリング
↓
得られたk(数百から数千)個のクラスを**Visual Words**と呼ぶ

元の大量の画像それぞれの各SIFTについて、最も近いVisual Wordsに投票することで、各画像ごとのVisual Wordsヒストグラムを作成

↓
このk次元ヒストグラムを画像の特徴量として、この特徴空間で、kd-tree法により探索木を作成

推論

推論時はこの木に入力して最近傍を見つける



機械学習による画像認識の応用例4 人体姿勢推定

人体姿勢推定

距離画像上で、ある体のパーツ位置に対し周囲の特定の領域の輝度の変化が規則的なパターンを持つことを利用した識別

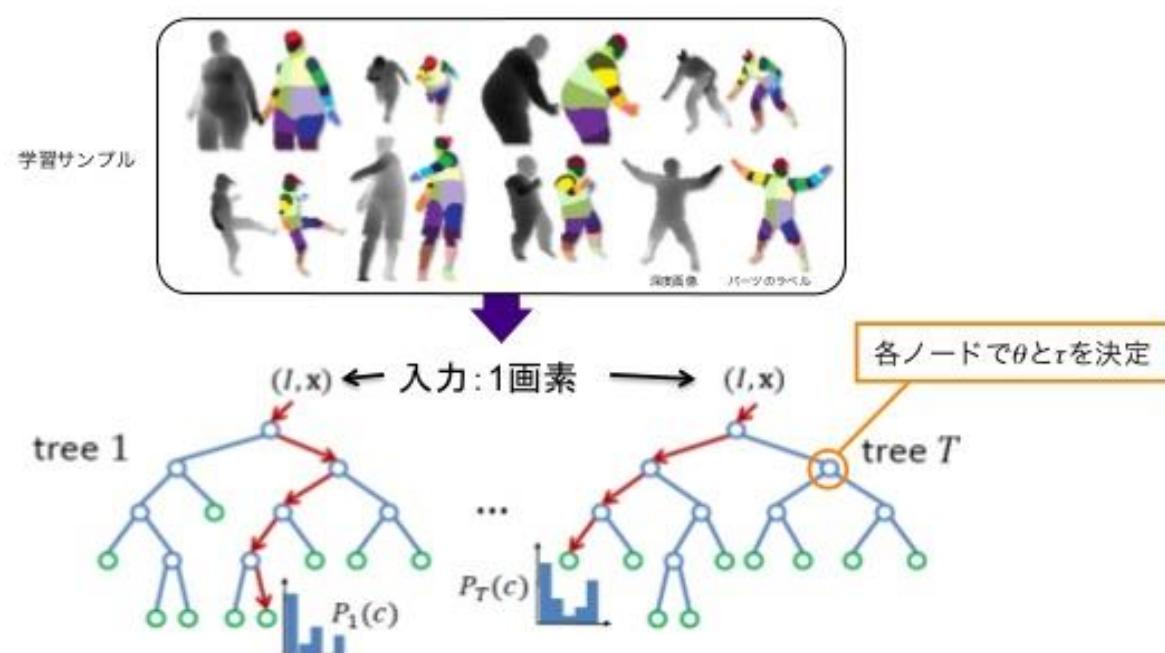
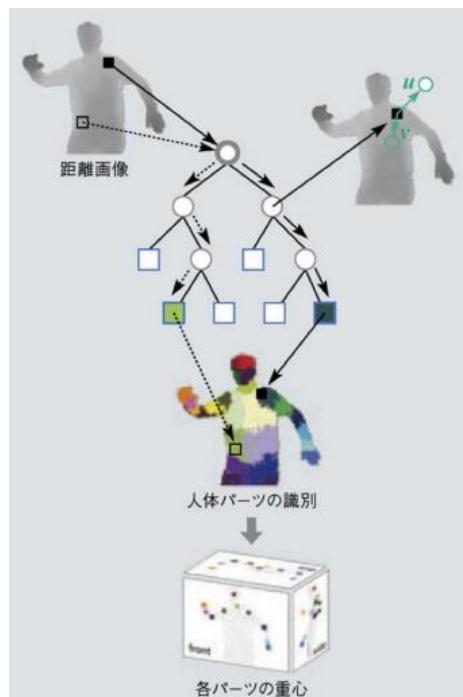
入力 = 距離画像から検出した人体画像の注目画素

モデル = ランダムフォレスト

出力 = 注目画素が属するパート

Random Forests による識別器の学習

- 生成した人体パーツ画像データを学習
 - 学習の入力：1画素毎単位の特徴量 + 正解人体パーツラベル



13.画像を入力とした
パターン認識(深層学習)

誤差逆伝搬法によるニューラルネットワークの学習

勾配降下法

ニューラルネットワークの重みの更新方法

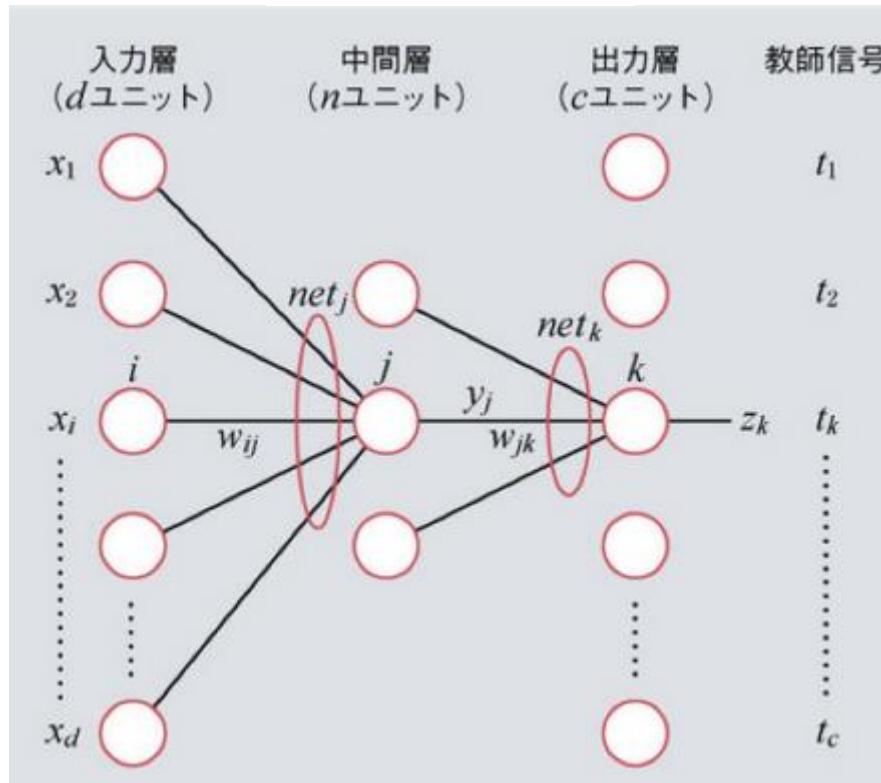
誤差の各重み方向の勾配を引くことで、誤差の極小値に近づける

w_{ij} を求めるのに後ろの層の重み w_{jk} を使うため、誤差逆伝搬法と呼ばれる

$$J = \frac{1}{2} \sum_{k=1}^c (z_k - t_k)^2$$

$$\begin{cases} w_{ij} \leftarrow w_{ij} - \rho \frac{\partial J}{\partial w_{ij}} \\ w_{jk} \leftarrow w_{jk} - \rho \frac{\partial J}{\partial w_{jk}} \end{cases}$$

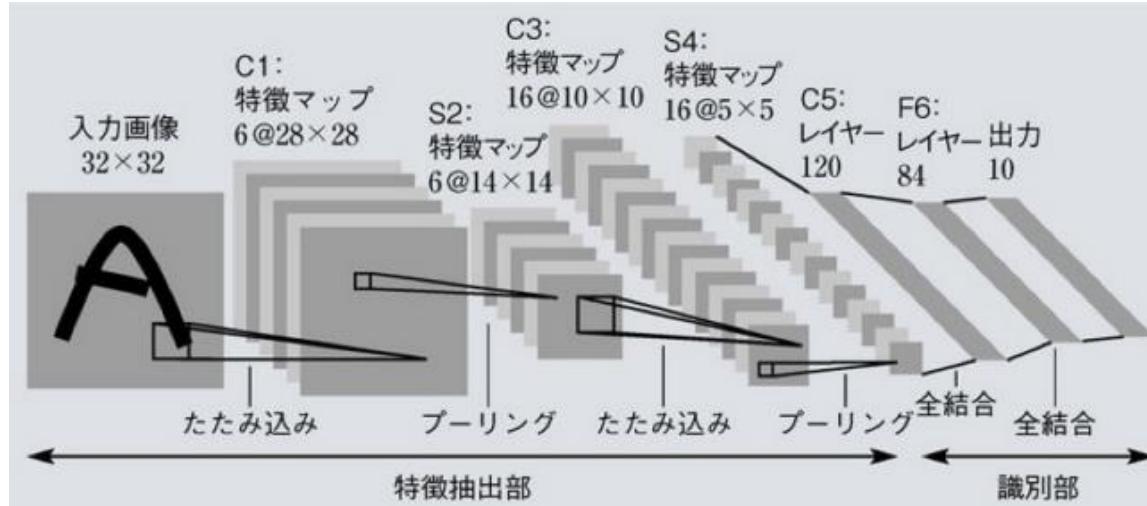
$$\frac{\partial J}{\partial w_{ij}} = \left(\sum_{k=1}^c (z_k - t_k) z_k (1 - z_k) w_{jk} \right) y_j (1 - y_j) x_i$$



畳み込みニューラルネットワーク

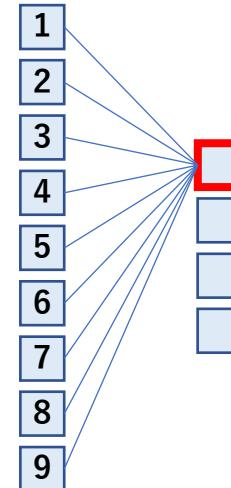
CNN

画像からの特徴量抽出と識別を同時に使うモデル。特徴量抽出は畳み込みとプーリング、識別はニューラルネットワークによって行う。ニューラルネットワークでは、中間層を増やすにつれて、過学習しやすいという問題と誤差伝播が進まなくなる勾配消失問題があったが、CNNは各層間が一部しか結合しておらず、結合が疎になっているため、過学習を防げる。

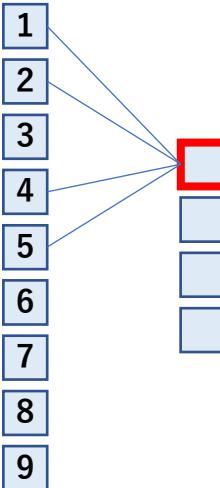


ニューラルネットワークの場合

1	2	3
4	5	6
7	8	9
1	2	3
4	5	6
7	8	9



CNNの場合 (2×2 のフィルタ)



汎化能力の向上

ドロップアウト

学習時、いくつか(50%くらい)のニューロンをランダムに無効化。
次の学習時にはまた別の組み合わせでランダムに無効化。これによりパラメータ数が下がり過学習が減る。

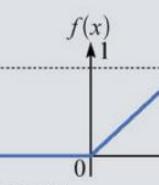
活性化関数の選別

シグモイドは多層になると勾配消失問題が生じやすいため、
CNNでは、ReLUやLeaky ReLUが用いられる。

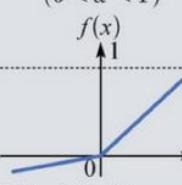
$$f(x) = \frac{1}{1+e^{-x}}$$



$$f(x) = \max(x, 0)$$

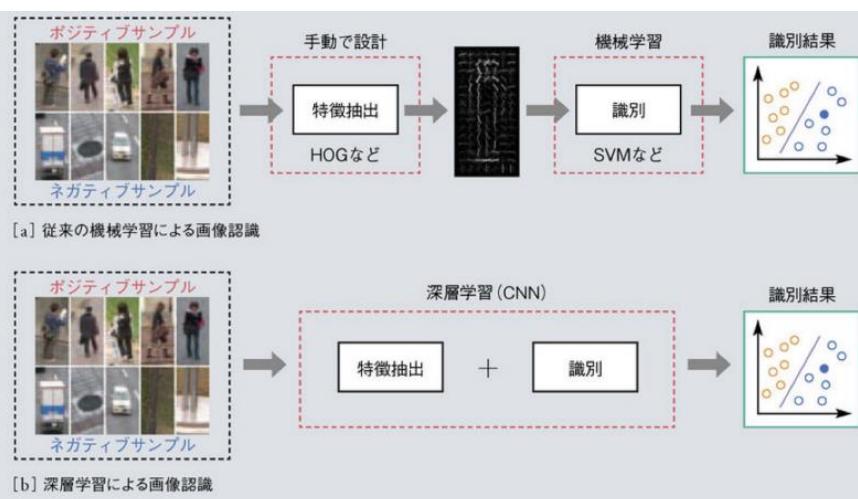
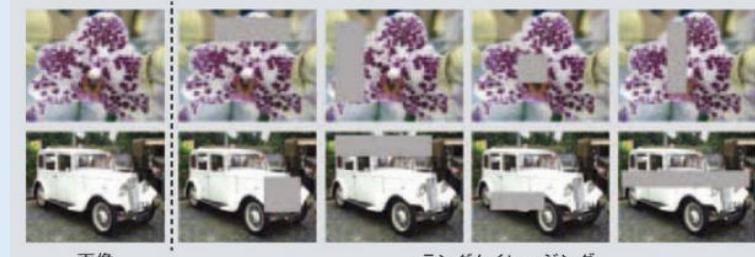


$$f(x) = \max(\alpha x, x) \quad (0 < \alpha < 1)$$



ランダムイレージング

学習画像の一部をランダムに削除して、データ数を負やす手法



CNNによる画像認識1 画像分類

■表13.1—IILSVRCにおける画像分類の性能

ネットワーク(チーム)	年度	エラー率
AlexNet(トロント大学)	2012	15.3%
GoogLeNet(Google)	2014	6.67%
ResNet(マイクロソフトリサーチアジア)	2015	3.56%
SENet ¹¹⁾	2017	2.25%
人間	—	5.1%



■図13.12—AlexNetによる1,000クラス画像識別

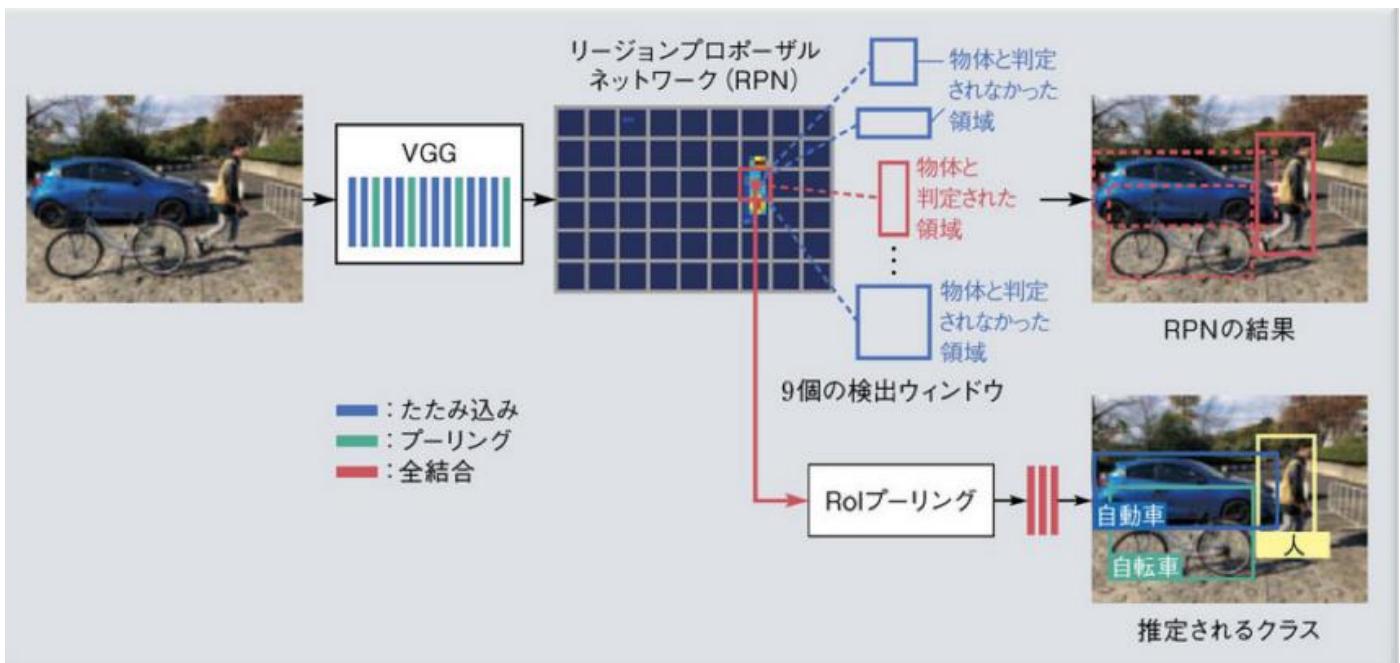
たとえば右端のヒョウでは、ほかの候補であるジャガーやチーターなどと比べ、ヒョウである確率が大幅に高い結果となっている。

CNNによる画像認識2 物体検出

機械学習による物体検出 = 縦横比一定領域のみの探索 → 2クラス分類
CNNによる物体検出 = 任意の縦横比の領域の探索 → 多クラス分類(様々な物体の検出)

Faster R-CNN(リージョンプロポーザル手法)

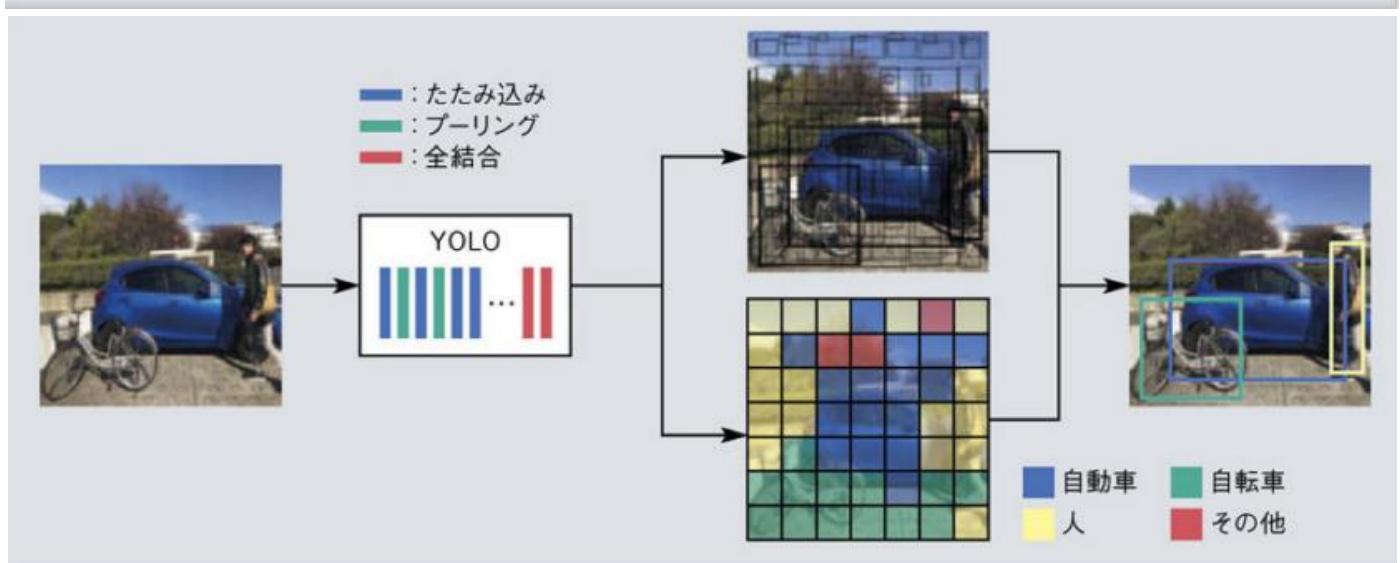
複数回の畳み込みとプーリングにより特徴マップ生成
↓
特徴マップをラスタスキャンし、注目画素を中心に縦横比と大きさの異なる9つの検出ウィンドウをあてはめる
↓
この領域をリージョンプロポーザルネットワークに入力し、物体らしさのスコアと矩形領域の座標を得る
↓
その領域に対しROIプーリングで一定のサイズに変換
↓
全結合層に入力し、多クラス分類



YOLO(You Only Look Once)(シングルショット手法)

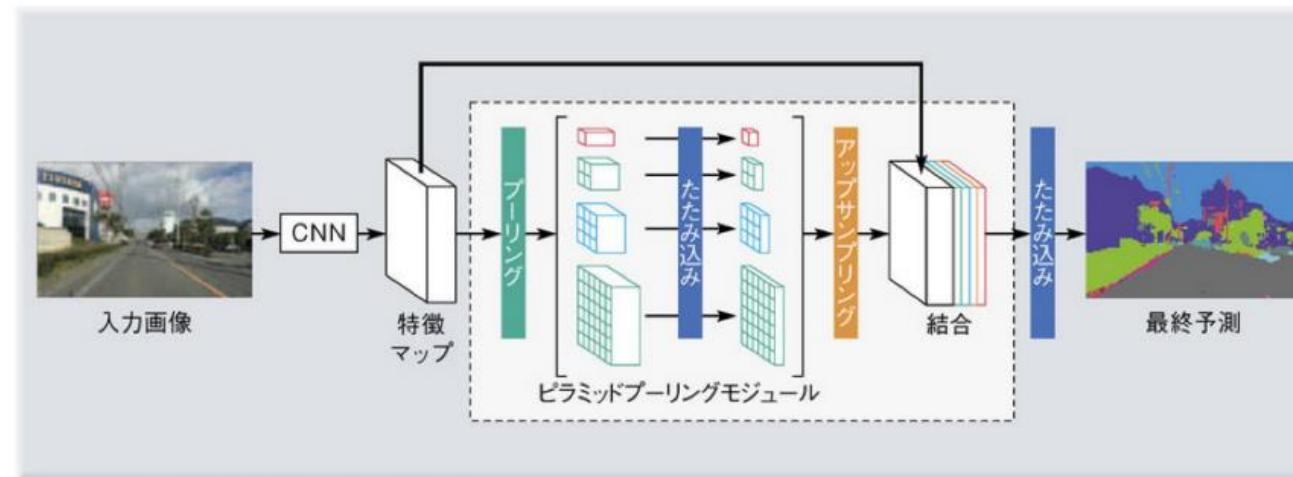
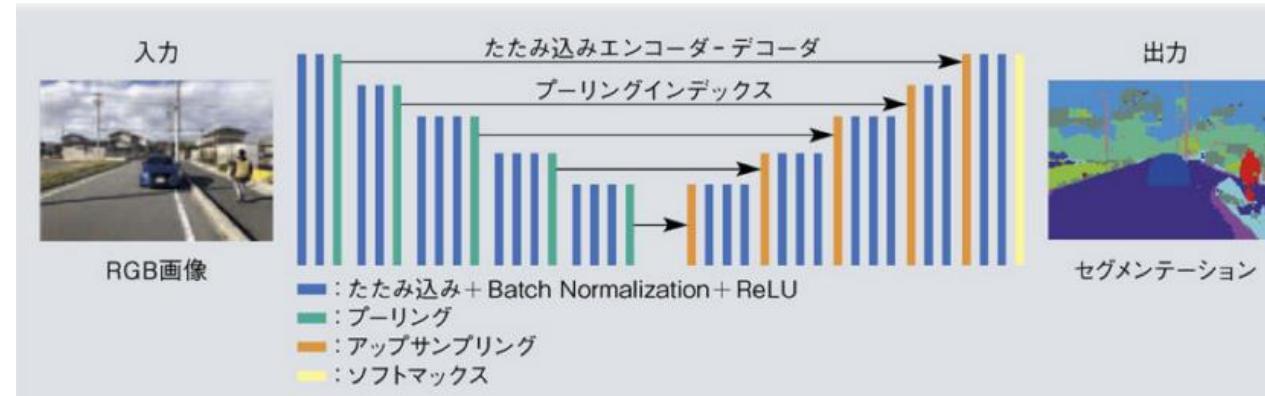
複数回の畳み込みとプーリングにより特徴マップ(7×7)生成
↓
特徴マップを全結合に入力
↓
出力は $7 \times 7 = 49$ 個で、
画像全体を 7×7 に分割した各領域が属するクラス(20種)の確率を意味する

Faster R-CNNよりもシンプルで、リアルタイム物体検出が可能



CNNによる画像認識3 セマンティックセグメンテーション

画像の画素単位での物体カテゴリを求めるモデル



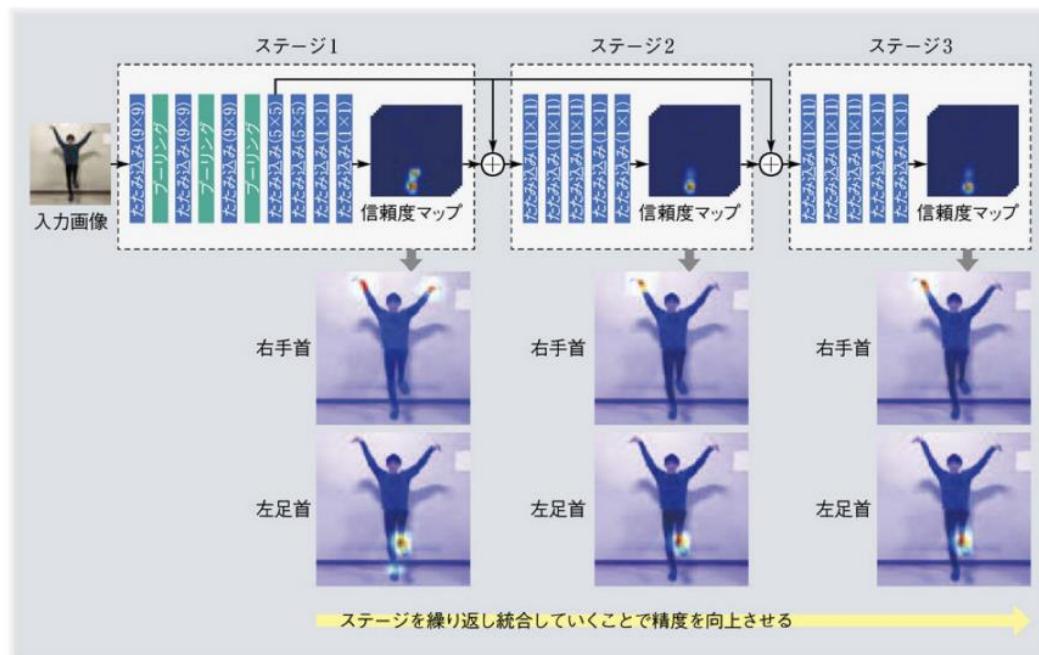
CNNによる画像認識4 姿勢推定

姿勢推定

画像中に含まれる人体の各部位の位置を特定すること

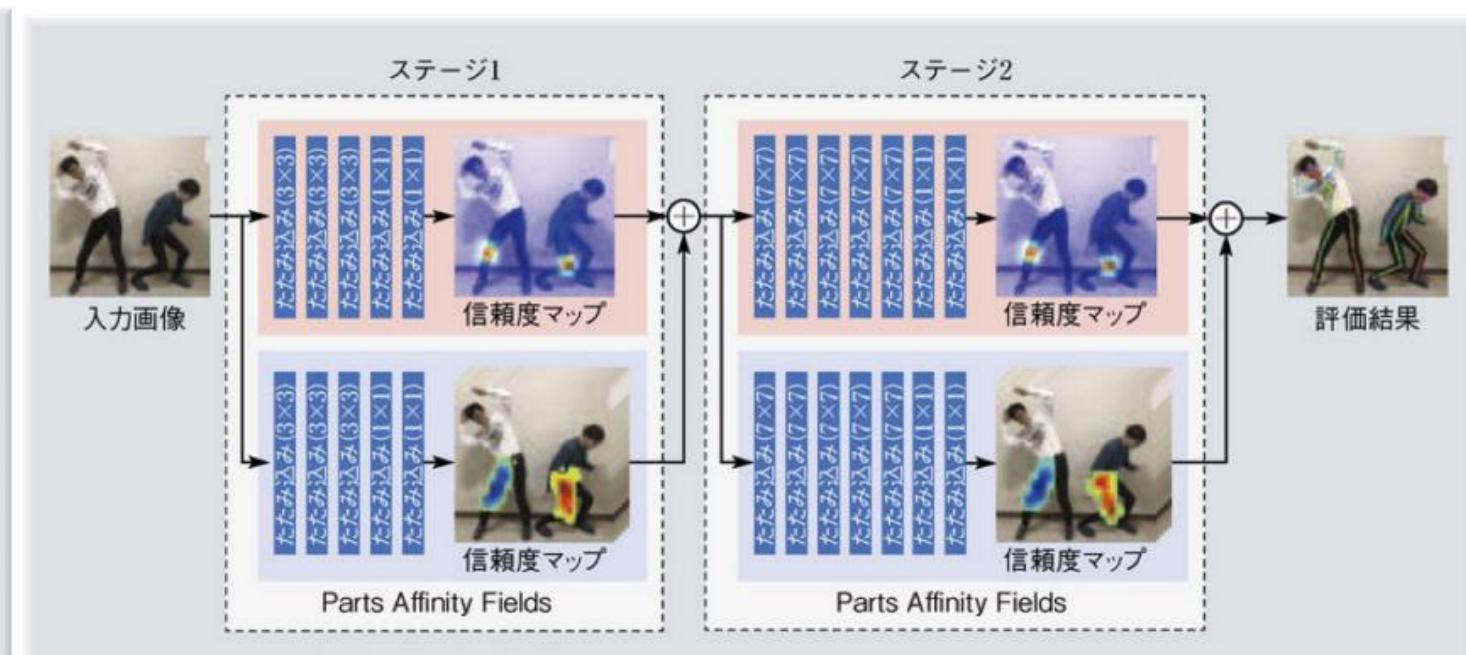
コンボリューションナルマシン

各部位の座標を信頼度マップとして段階的に出力。
畳み込み処理をステージ1から3にかけて繰り返し統合していくことで
大局的な骨格情報を考慮した人体部位の位置座標の検出が可能



オープンポーズ

複数の人体の各部位の位置座標を検出する手法。
通常の信頼度マップに加えて、Parts Affinity Fieldsとよばれる
部位間のつながりを表す信頼度マップも同時に output。
この情報を統合しながら精度を向上させる。



CNNによる画像認5 GAN

PGGAN(Progressive Growing GAN)

GANやDCGANでは生成できる
画像の解像度は 64×64 が限界

PGGANでは段階的に層数とラベル画像の解像度を
増やすことで 1024×1024 の画像を生成可能

GAN

生成器と識別器の二つのニューラルネットワークについて、
複数回の学習を交互に行うことで、
何もないところから(乱数)から自然な画像を生成するモデル

生成器

入力 = ランダムノイズ
ラベル = 画像

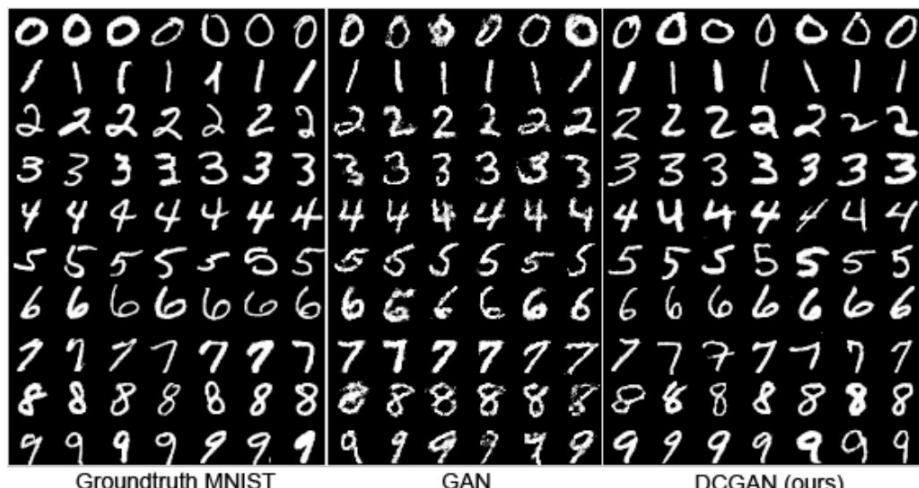
乱数から自然な画像を生成するように学習が進行

識別器

入力 = 生成器によって生成された画像 or 実画像
ラベル = Fake or Real
生成器によって生成された画像かどうかを見抜くように学習が進行

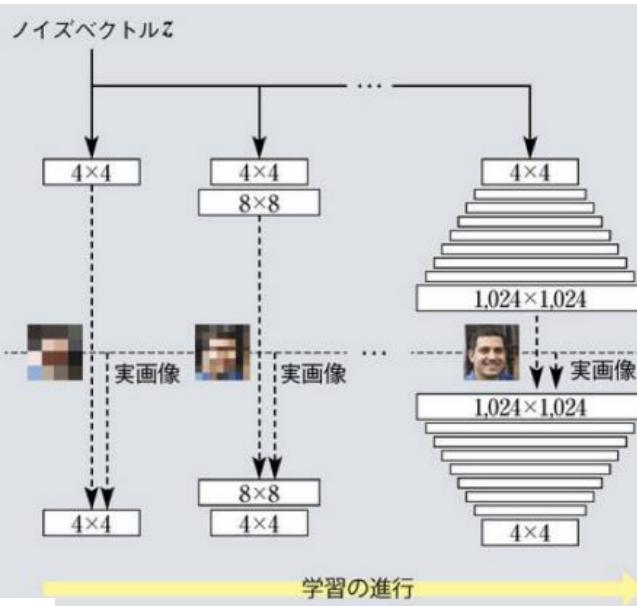
DCGAN(Deep Convolutional GAN)

GANにおいてニューラルネットワークの代わりに
CNNを用いたもの。GANよりも自然な画像が得られる。



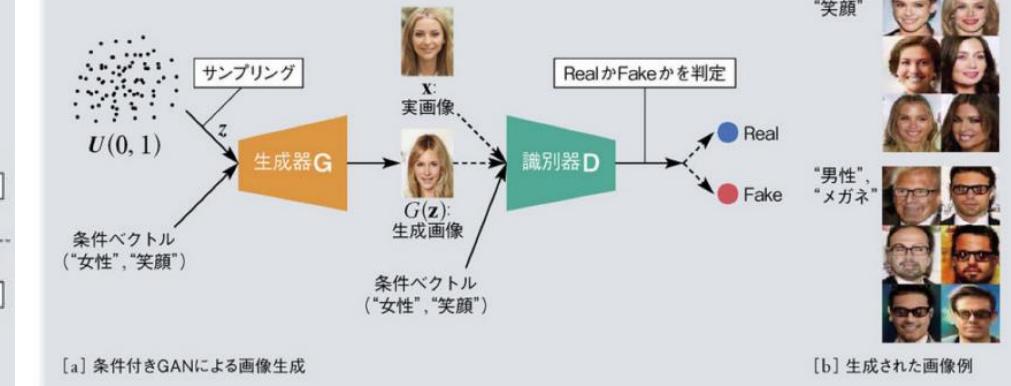
モード崩壊

GANにおいて類似した画像しか生成できなくなる問題



CGAN(Conditional GAN)

生成器にノイズベクトルに加えて条件ベクトルを入力。さらに、ラベルは条件を満たした画像とすることで条件を満たす画像の生成が可能



pix2pix(GANによる画像変換)

生成器は画像を入力とし、正しく画像変換をするよう学習
識別機は、変換前後の画像ペアを入力とし、それが生成器によって行われた画像変換かどうかを見抜くように学習
これにより、所望の画像変換を行うモデルが得られる

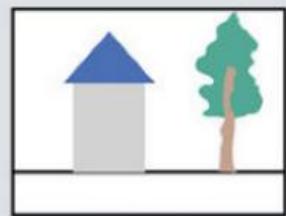


14. 動画像処理

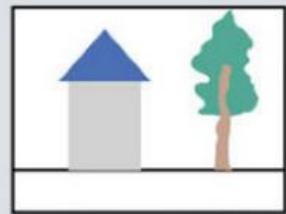
画像差分による移動体検出1 背景差分法

背景差分法

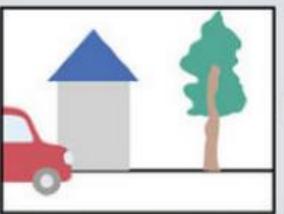
事前に撮影した画像と、同視野の移動物体が移った画像を、二値化&オープニング・クロージングを繰り返し、差分をとる。



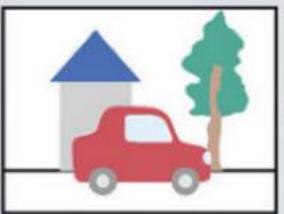
背景画像



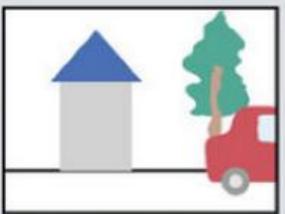
時刻 t_0 の画像



時刻 t_0+1 の画像



時刻 t_0+2 の画像



時刻 t_0+3 の画像



時刻 t_0
移動物体画像



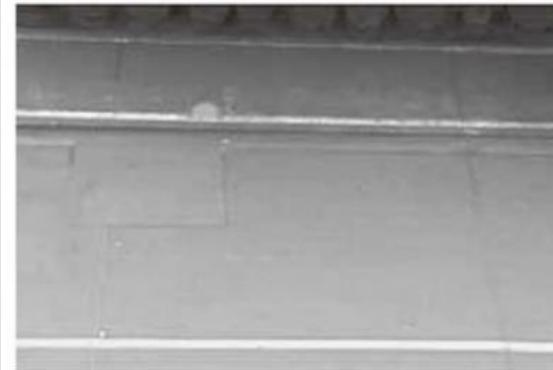
時刻 t_0+1
移動物体画像



時刻 t_0+2
移動物体画像



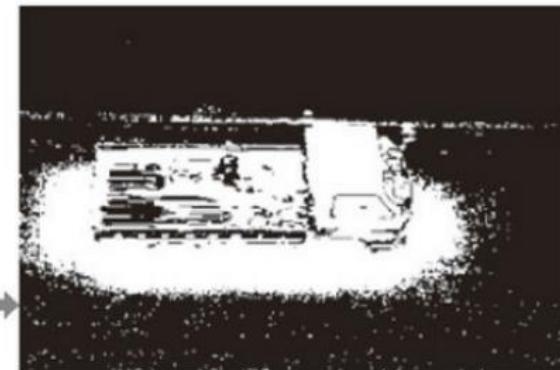
時刻 t_0+3
移動物体画像



[a] 背景画像



[b] 移動物体が写った画像



[c] 差分画像の2値画像



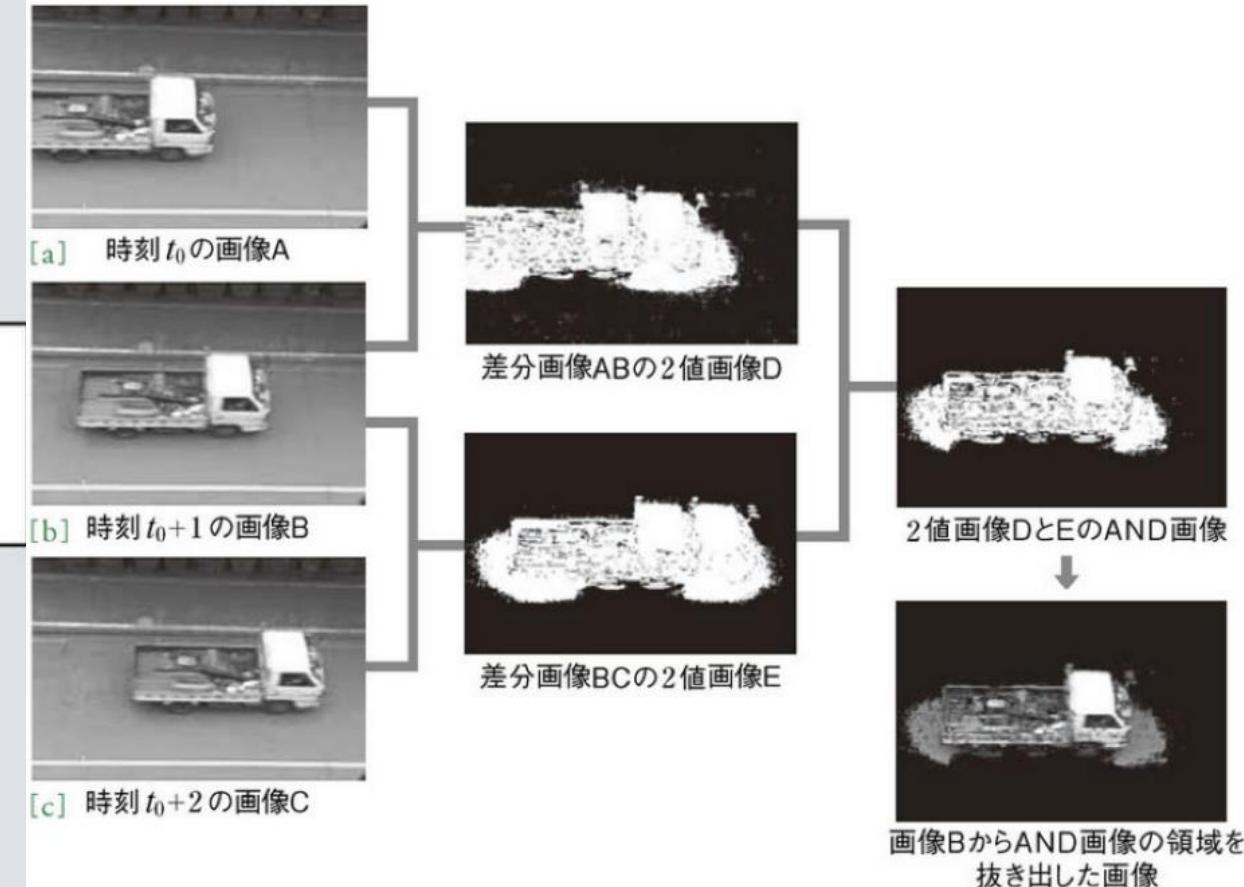
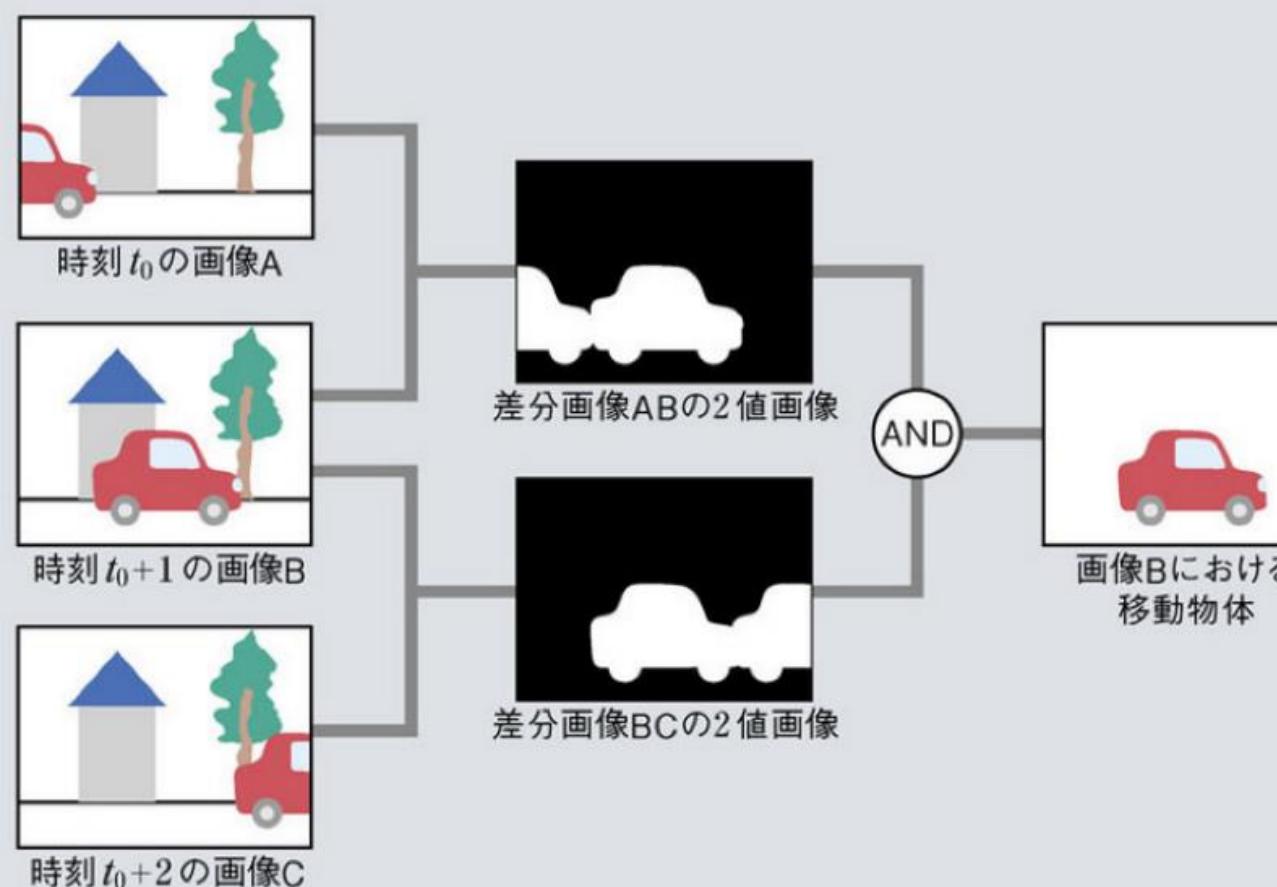
[d] 移動物体画像

画像差分による移動体検出2フレーム間差分法

フレーム間差分法

背景画像を事前に用意できない場合の代用法

連続する三フレームA,B,Cを用意
↓
差分画像A-B、B-Cを得る
↓
それぞれ二値化
↓
A-BとB-Cの論理積をBの移動体領域とする



画像差分による移動体検出3 統計的背景差分法

統計的背景差分法

風で揺れる葉っぱなど背景画像が変動している場合を考慮した背景差分法

事前に背景画像を複数枚用意し、画素値のヒストグラムを作成



そこからある画素値が背景に属する確率(相対頻度)を求める

$$p(I|\omega_0) = \frac{q(I)}{\sum_{I=0}^{255} q(I)}$$



一方、ある画素値が移動物体に属する確率は未知なので等確率とおく

$$p(I|\omega_1) = \frac{1}{256}$$



対象画像の各画素に画素値Iについて、
 $p(I|\omega_0)$ の方が大きければ、 ω_0 、
 $p(I|\omega_1)$ の方が大きければ、 ω_1 に分類



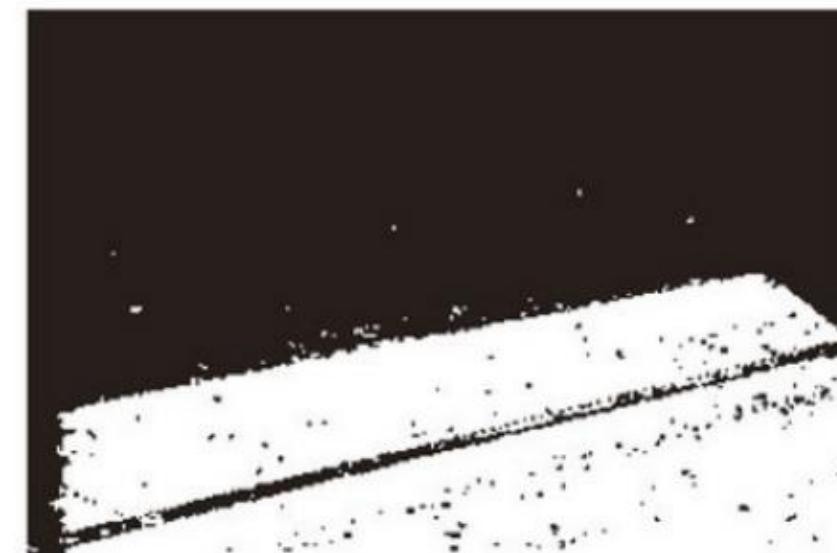
[a] 背景画像



[b] 白いトラックが通過したシーンの画像



[c] 背景画像と対象画像の差分画像



[d] 統計的背景差分法による判別結果

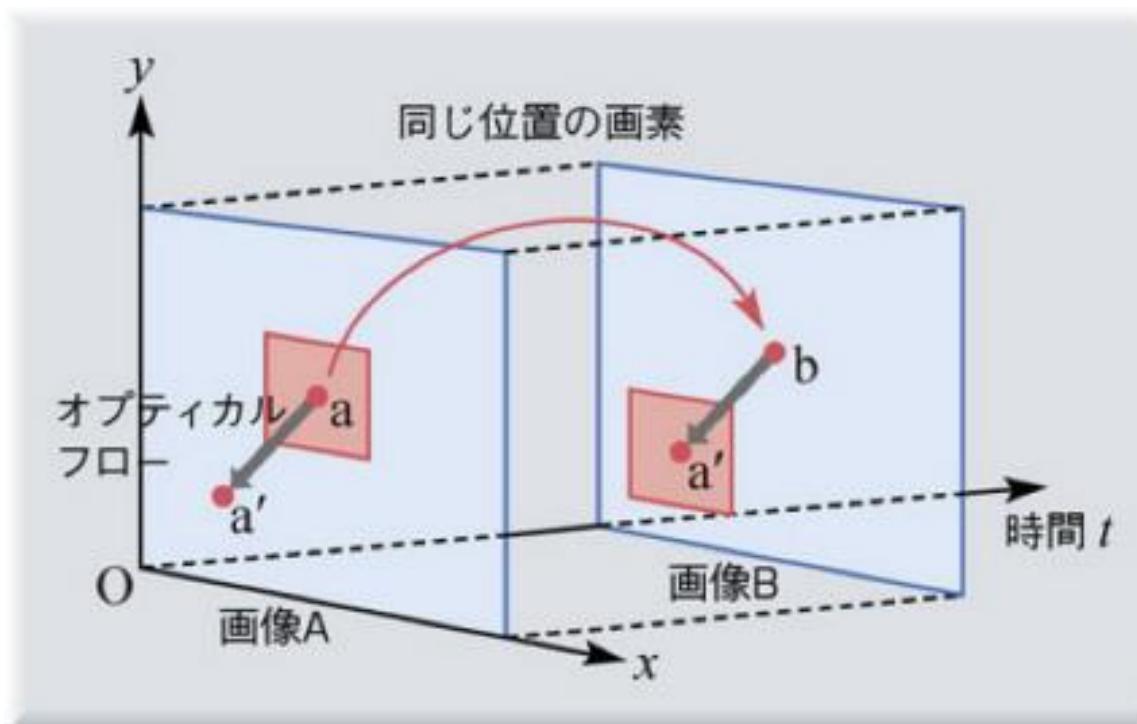
オプティカルフローの導出1 ブロックマッチング

オプティカルフロー

対象物体のフレーム間の画素の変位を表す二次元ベクトル
カメラに近い物体ほど大きいベクトルになる

ブロックマッチング法

時間的に前の画像の小領域をテンプレートとして、
時間的に後ろの画像に対してテンプレートマッチング(相違度を計算)
この処理を画像全体にやることになるため、計算量は膨大という欠点がある



オプティカルフローの導出2 勾配法

勾配法

対象物体の移動量が微小であると仮定した場合のオプティカルフローの導出法

オプティカルフローの拘束条件式

勾配法を解く上でのベースとなる条件式

移動前後で画素値は不变であるから、

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t)$$

↓
テイラー展開をして、移動量は微小であるから二次以降の項を削除して

$$I(x, y, t) \doteq I(x, y, t) + \frac{\partial I}{\partial x} \Delta x + \frac{\partial I}{\partial y} \Delta y + \frac{\partial I}{\partial t} \Delta t$$

↓
両辺を Δt で割り、

$$\frac{\partial I}{\partial x} \frac{\Delta x}{\Delta t} + \frac{\partial I}{\partial y} \frac{\Delta y}{\Delta t} + \frac{\partial I}{\partial t} = 0$$

$$\frac{\partial I}{\partial x} \frac{\partial x}{\partial t} + \frac{\partial I}{\partial y} \frac{\partial y}{\partial t} + \frac{\partial I}{\partial t} = 0$$

↓
オプティカルフローを

$$(u, v) = \left(\frac{\partial x}{\partial t}, \frac{\partial y}{\partial t} \right)$$

とすると、

$$I_x u + I_y v + I_t = 0$$

オプティカルフローの拘束条件式は、未知数2に対して、方程式1のため、LK法、拡張LK法、Horn Schunk法で(u,v)を絞り込む

オプティカルフローの導出2-1 LK法

LK法

周辺の画素はすべてオプティカルフローが同じとして、周辺m画素すべてをおなじ(u,v)で拘束条件式に当てはめて擬似逆行列により最小解を得る

$$\mathbf{G}\mathbf{f} + \mathbf{b} = 0$$

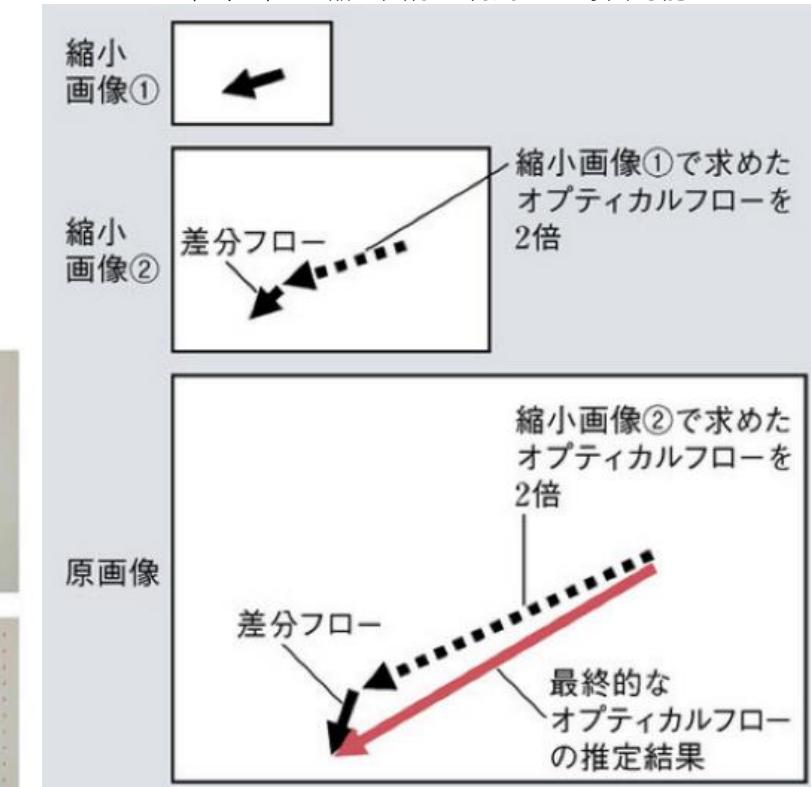
$$\mathbf{G} = \begin{pmatrix} I_{x1} & I_{y1} \\ \vdots & \vdots \\ I_{xm} & I_{ym} \end{pmatrix}, \quad \mathbf{f} = \begin{pmatrix} u \\ v \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} I_{t1} \\ \vdots \\ I_{tm} \end{pmatrix}$$

$$\mathbf{f} = -(\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T \mathbf{b}$$



移動量が大きい場合

1/2、1/4の縮小画像を利用して導出可能



オプティカルフローの導出2-2-3 拡張LK法・Horn-Shunck法

ワープ画像 $W(x; p)$

画素 x がオプティカルフロー p によって移動した画像
オプティカルフローが並進の場合

$$W(x; p) = \begin{pmatrix} x + p_1 \\ y + p_2 \end{pmatrix}$$

アフィン変形の場合

$$W(x; p) = \begin{pmatrix} (1 + p_1) \cdot x + p_3 \cdot y + p_5 \\ p_2 \cdot x + (1 + p_4) \cdot y + p_6 \end{pmatrix}$$

$$= \begin{pmatrix} 1 + p_1 & p_3 & p_5 \\ p_2 & 1 + p_4 & p_6 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

拡張LK法

LK法が領域内の全画素が並進移動していると仮定したのに対し、アフィン変形を仮定した手法
アフィン変形によるワープ画像と真の画像の差を最小化する
ただし、アフィン変換は非線形であるため擬似逆行列では最小解は求まらない
よって、 p について初期値を与え Δp を繰り返し求めながら p を更新する

$$\sum_x (I(W(x; p + \Delta p)) - T(x))^2$$

$$p \leftarrow p + \Delta p$$

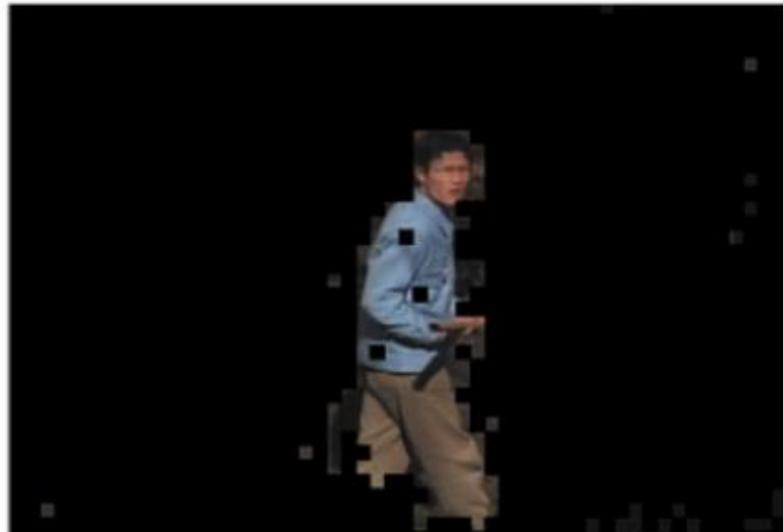
Horn-Shunck法

移動量に対するコスト関数を設け、それを最小化するオプティカルフローを求める

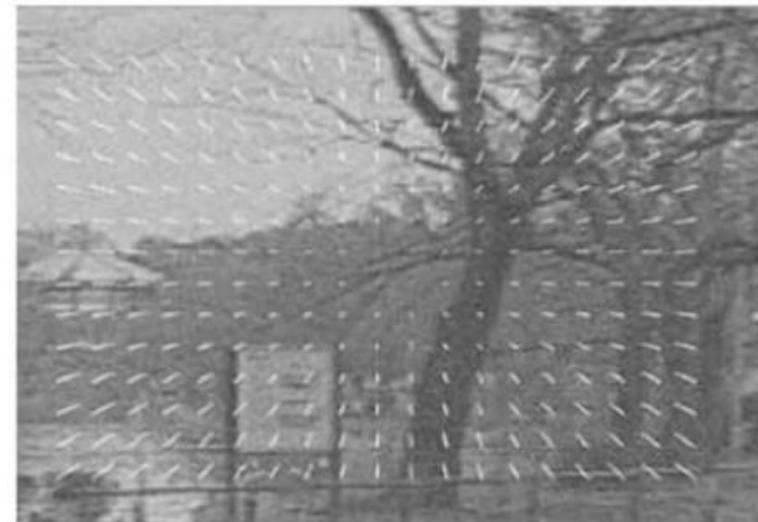
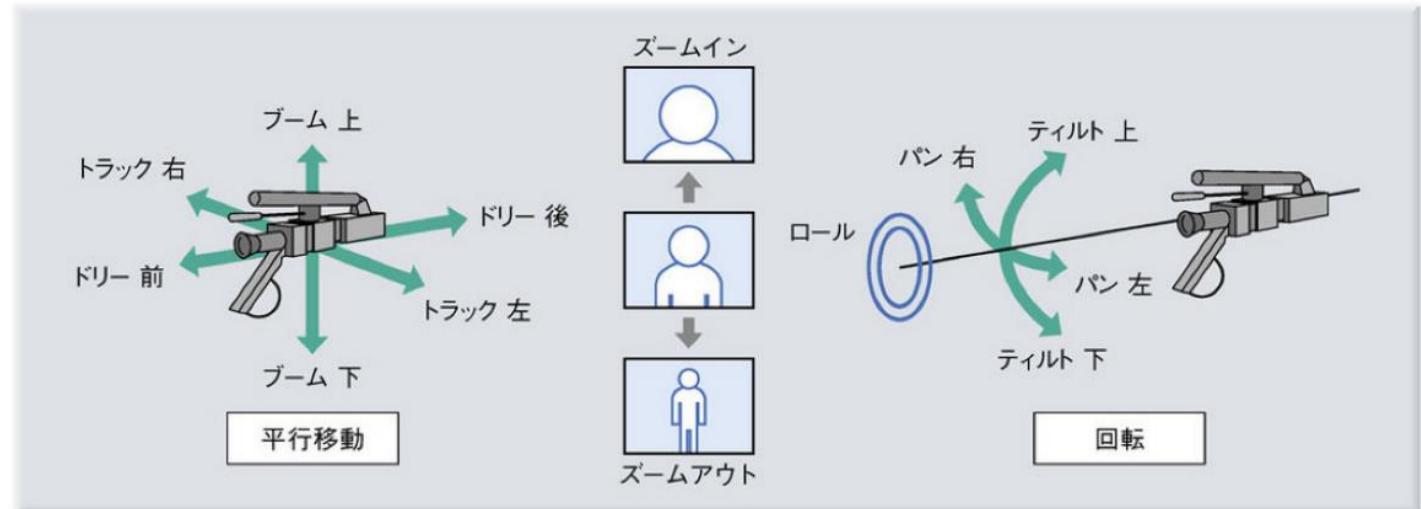
$$E = \sum_x \sum_y (I_x u + I_y v + I_t)^2 + \lambda \sum_x \sum_y \left(\left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial u}{\partial y} \right)^2 + \left(\frac{\partial v}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial y} \right)^2 \right)$$

オプティカルフローの利用例

移動体検出



カメラのモーション推定



[a] 前方へのドリー



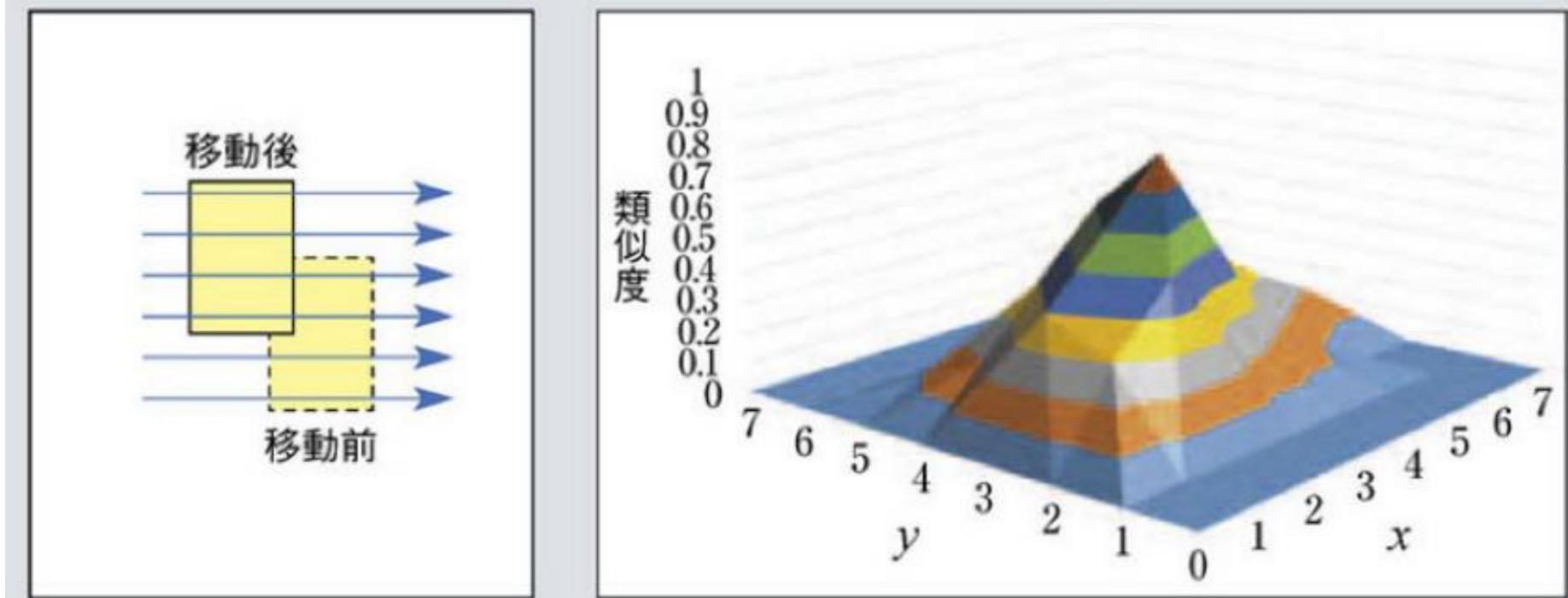
[b] 右上に平行移動

物体追跡1 テンプレートマッチング

テンプレートマッチング

近傍の範囲のみを探索し、類似度最大値を移動位置とする手法

しかし、探索範囲の指定の方法論がなかったり、背景領域にテンプレートに似たものがあると誤って追跡してしまうという欠点がある

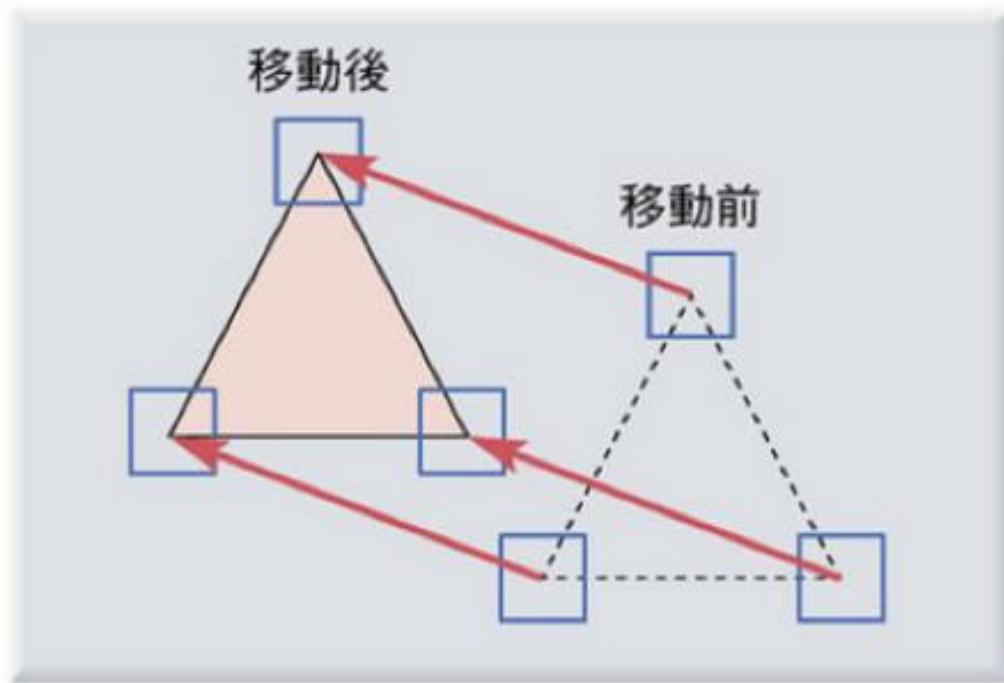


[a] テンプレートマッチング

テンプレートマッチングでは、近傍の領域内のすべての候補位置をスキャンし、テンプレートとの類似度を計算したあと、類似度最大の位置を**物体**の位置として検出する。

物体追跡2 KLTトラッカー

KLTトラッカー
ハリスのコーナー検出で特徴点抽出し、
特徴点のLK法でオプティカルフローを求め物体を追跡する



■図14.16——KLTトラッカーにおける特徴点検出の例

物体追跡3 ミーンシフトトラッキング

ミーンシフトトラッキング

物体がピッタリ収まる矩形領域の色ヒストグラムをモデルとし、探索領域との類似度の極大点を移動先とみなす手法
色ヒストグラムの類似度はバタチャリヤ係数

$$\rho[p(y_0), q] = \sum_{u=1}^m \sqrt{p_u(y_0)q_u}$$

バタチャリヤ係数はヒストグラムインタセクションとなり微分可能

→周辺領域全体をスキャンして類似度最大値とするのではなく、**勾配方向を辿れば良い**ので効率が良い
探索領域が物体移動先領域と重なっている部分の比重を高くして重心を取り、次の探索位置とする

全く重なっていない場合は追跡不可能になることに注意

$$\rho(p(y), q) \approx \frac{1}{2} \sum_{u=1}^m \sqrt{p_u(y_0)q_u} + \frac{1}{2} \sum_{u=1}^m p_u(y) \sqrt{\frac{q_u}{p_u(y_0)}}$$

$$p_u(y) = C_h \sum_{i=1}^{n_h} k\left(\left\|\frac{y - \mathbf{x}_i}{h}\right\|^2\right) \delta(b(\mathbf{x}_i) - u)$$

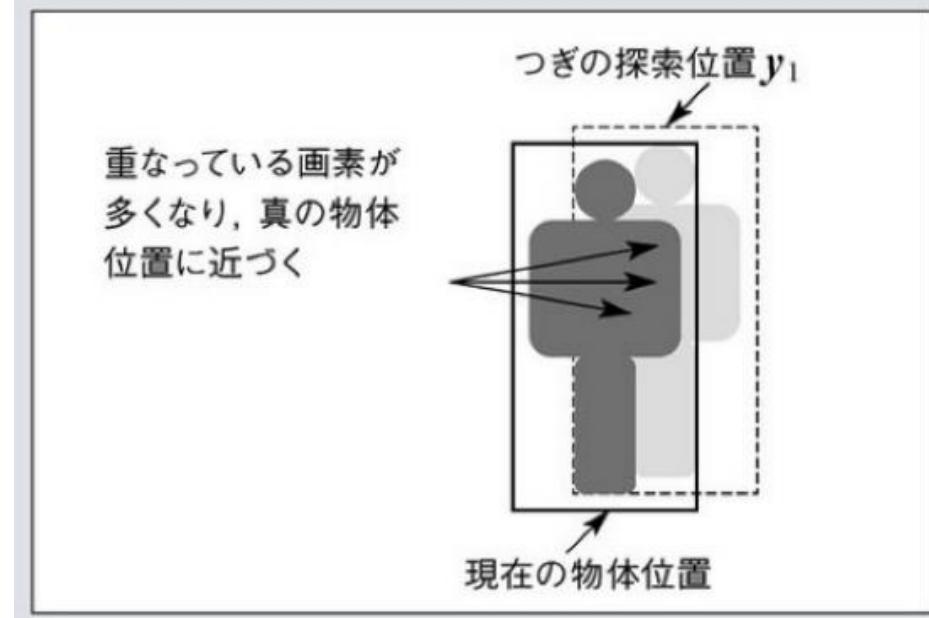
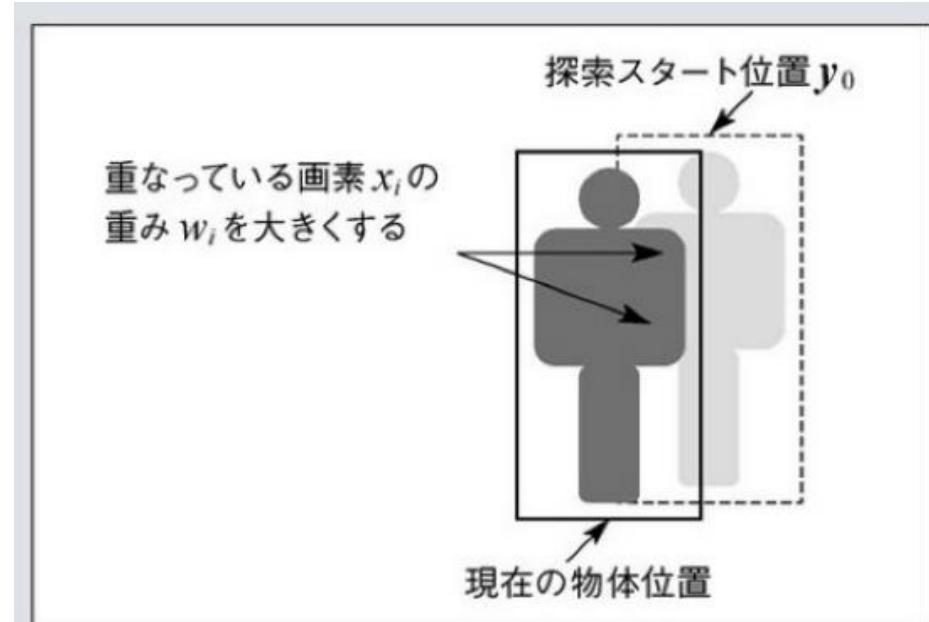
$$\rho(p(y), q) \approx \frac{1}{2} \sum_{u=1}^m \sqrt{p_u(y_0)q_u} + \frac{C_h}{2} \sum_{i=1}^{n_h} w_i k\left(\left\|\frac{y - \mathbf{x}_i}{h}\right\|^2\right)$$

重み計算

モデルは物体がピッタリ収まるように切り取った矩形領域だから、物体上のカラーはモデル領域で最も頻度が高くなるはず
探索領域での注目画素が持つカラーの頻度と、モデル領域での同カラーの頻度を比較
注目画素が物体上の点ならカラー頻度はモデルの方が高く、背景上の点ならカラー頻度は探索領域の方が高い
この性質を利用し、物体上の点の画素位置に重みをつけた加重平均を次の探索位置とする

$$w_i = \sum_{u=1}^m \delta[b(\mathbf{x}_i) - u] \sqrt{\frac{q_u}{p_u(y_0)}}$$

$$y_1 = \frac{\sum_{i=1}^{n_h} \mathbf{x}_i w_i k\left(\left\|\frac{y_0 - \mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n_h} w_i k\left(\left\|\frac{y_0 - \mathbf{x}_i}{h}\right\|^2\right)}$$



物体追跡4 ベイジアンフィルタ 保留

ベイジアンフィルタ

各時刻における移動体の位置を確率分布として求める手法

カルマンフィルタ

ベイジアンフィルタにおいて線形・ガウス型の状態空間モデルにのみ適用可能な最適フィルタ

パーティクルフィルタ

非線形・非ガウス型を含むより一般的な状態空間モデルに適用可能なフィルタ

15.画像からの三次元位置推定 (コンピュータビジョン1)

空間と画像の幾何学的関係

画像座標・ワールド座標の関係

$$\tilde{m} \sim A\tilde{x}$$

$$\sim A(I|0)\tilde{x}$$

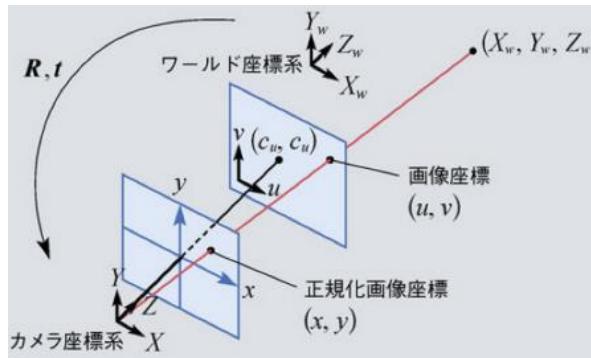
$$\sim A(I|0) \begin{pmatrix} R & t \\ 0^T & 1 \end{pmatrix} \tilde{X}_w$$

$$\sim A(R|t)\tilde{X}_w$$

$$P = A(R|t)$$

$$\tilde{m} \sim P\tilde{X}_w$$

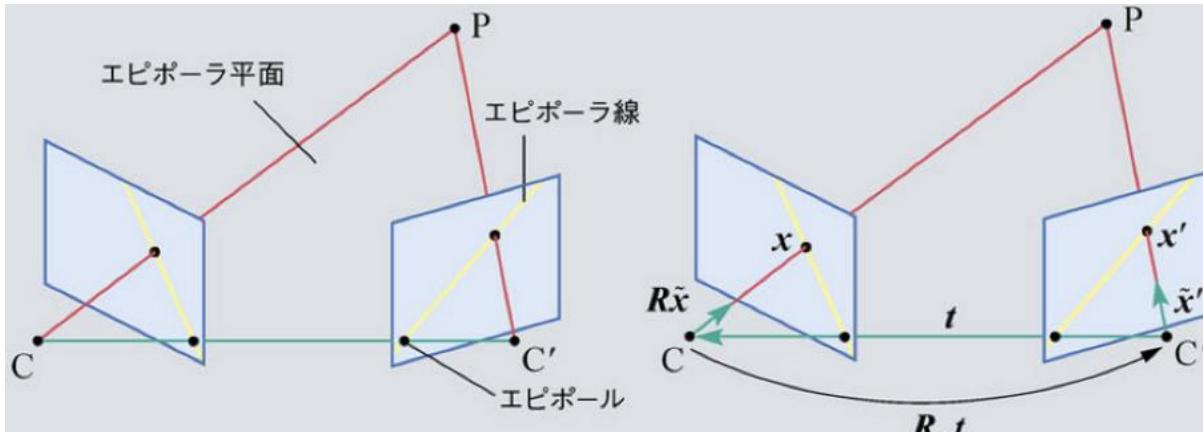
透視投影行列
またはカメラ行列



$$x = \begin{pmatrix} x \\ y \end{pmatrix}, m = \begin{pmatrix} u \\ v \end{pmatrix}, X = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}, X_w = \begin{pmatrix} x_w \\ y_w \\ z_w \end{pmatrix} \quad \tilde{x} \sim \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}, \tilde{m} \sim \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}, \tilde{X} \sim \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}, \tilde{X}_w \sim \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix}$$

~はその両辺が定数倍の違いを許して等しいことを意味する。 $\rightarrow \tilde{m} \sim P\tilde{X}_w$ は空間中における直線の方程式を与えるもの

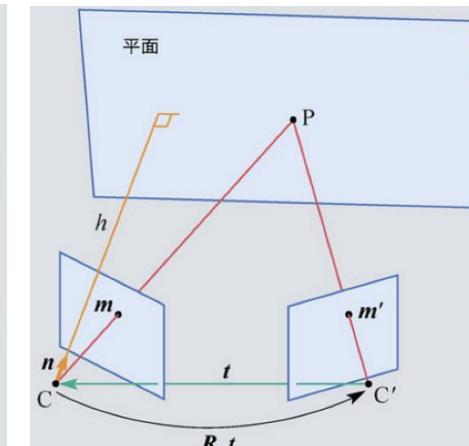
エピポーラ幾何…空間上の任意の点Pの2台のカメラで撮影した画像間の関係



$$\tilde{m}'^T F \tilde{m} = 0$$

$$F = (A'^{-1})^T [t]_{\times} R A^{-1}$$

空間上のある平面上の点Pの場合



$$\tilde{m}' \sim A' \left(R + \frac{tn^T}{h} \right) A^{-1} \tilde{m}$$

内部パラメータ
カメラによって決まる

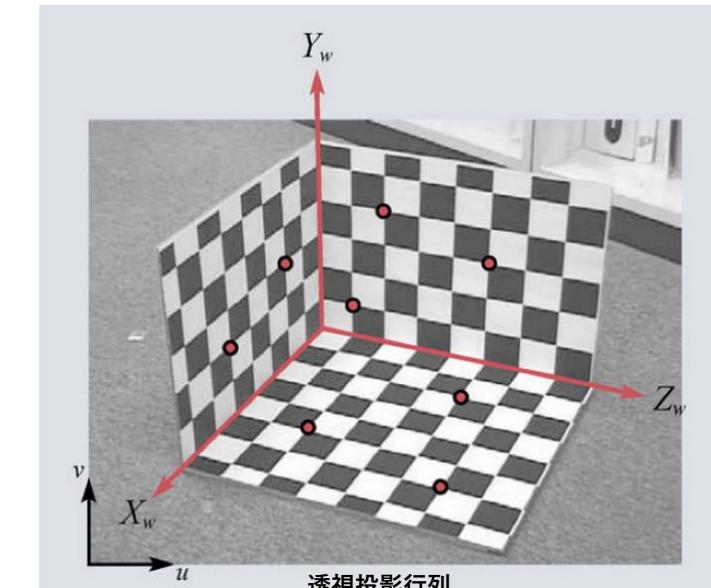
$$A = \begin{pmatrix} \frac{f}{\delta_u} & 0 & c_u \\ 0 & \frac{f}{\delta_v} & c_v \\ 0 & 0 & 1 \end{pmatrix}$$

外部パラメータ R, t
ワールド座標に対するカメラの位置と姿勢で決まる

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = R \begin{pmatrix} X_w \\ Y_w \\ Z_w \end{pmatrix} + t$$

カメラキャリブレーション

空間点の座標が容易に判別できるキャリブレーションターゲットを用いて、ワールド座標 X_w と画像座標 x から透視投影行列 P を求める



透視投影行列

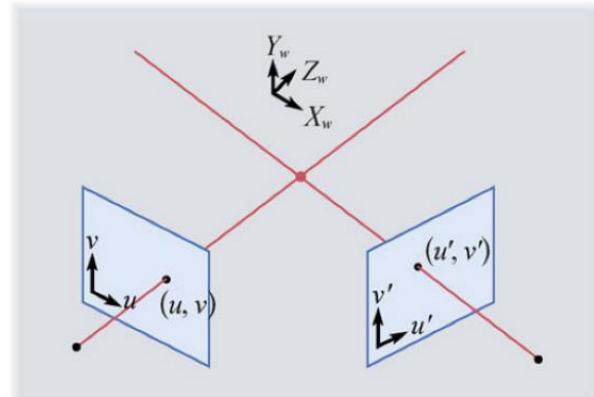
同次座標の不定性より、p34は1と置いてよいので、未知数は11。よってPを一意に決めるには、最低6組の座標が必要

$$P = \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{pmatrix}$$

m, m' から X, Y の推定(ステレオビジョン)

ステレオビジョン

カメラの内部パラメータや、カメラ間の位置関係が任意の視点の異なる画像を利用して、三次元情報を復元する方法

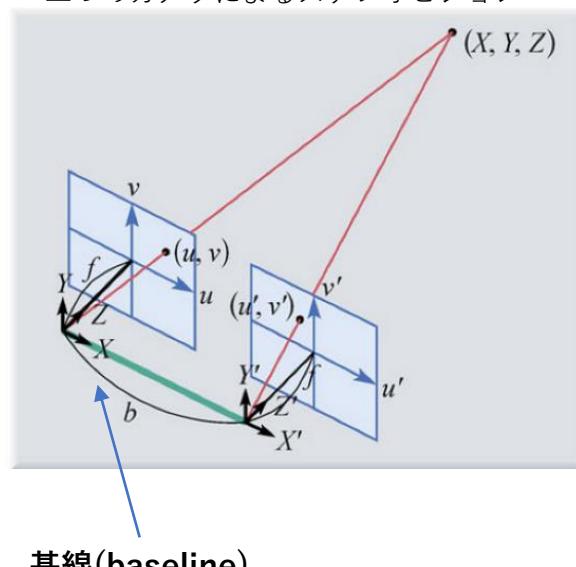


$(u, v), (u', v')$ から (X_w, Y_w, Z_w) を求める

$$\begin{pmatrix} p_{31}u - p_{11} & p_{32}u - p_{12} & p_{33}u - p_{13} \\ p_{31}v - p_{21} & p_{32}v - p_{22} & p_{33}v - p_{23} \\ p'_{31}u' - p'_{11} & p'_{32}u' - p'_{12} & p'_{33}u' - p'_{13} \\ p'_{31}v' - p'_{21} & p'_{32}v' - p'_{22} & p'_{33}v' - p'_{23} \end{pmatrix} \begin{pmatrix} X_w \\ Y_w \\ Z_w \end{pmatrix} = \begin{pmatrix} p_{14} - p_{34}u \\ p_{24} - p_{34}v \\ p'_{14} - p'_{34}u' \\ p'_{24} - p'_{34}v' \end{pmatrix}$$

平行ステレオ

内部パラメータが等しく、互いに光軸が平行で X 軸が同一な二つのカメラによるステレオビジョン



$$X = \frac{bu}{u - u'}$$

$$Y = \frac{bv}{u - u'}$$

$$Z = \frac{bf}{\delta(u - u')}$$

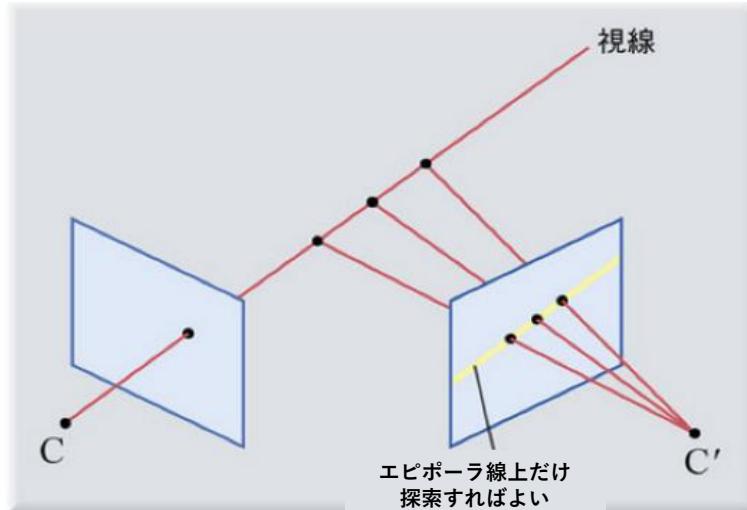
視差 $u - u'$ =水平方向のずれ

Z は視差のみで決まる！
視差と Z は反比例！

m に対する m' の特定(ステレオマッチング)

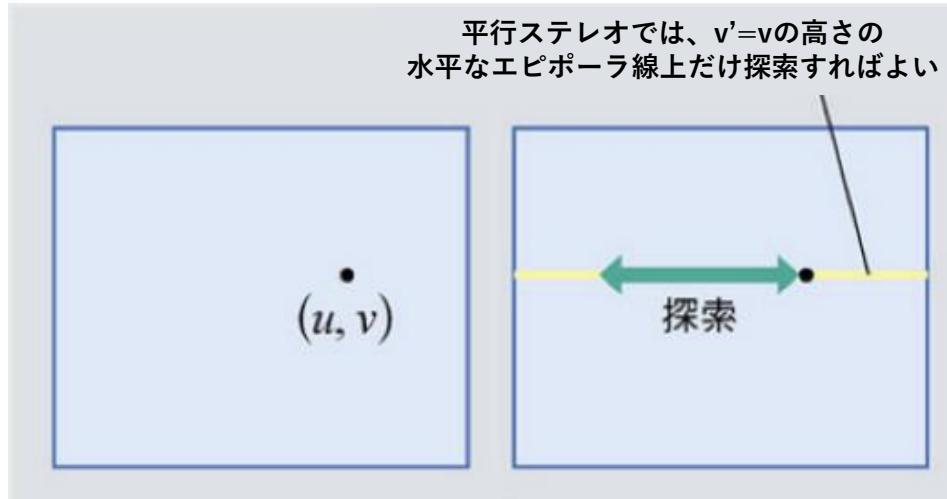
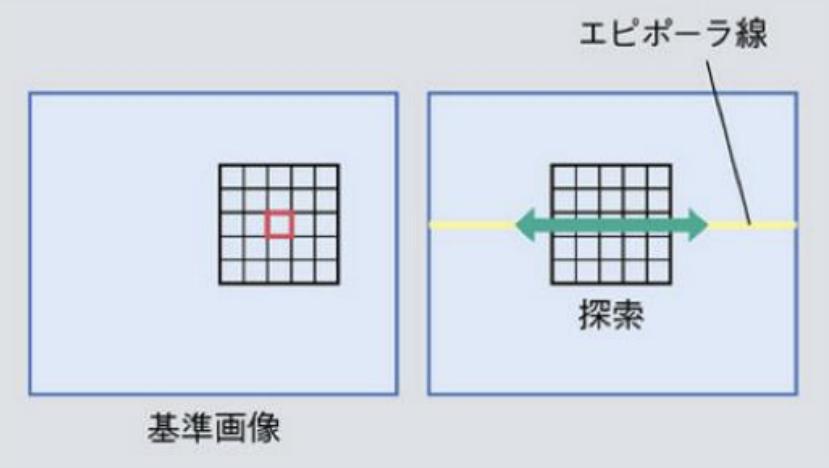
ステレオマッチング

二つの画像間の対応する点を特定すること



ブロックマッチング

注目点の周囲の領域をテンプレートとしたテンプレートマッチング



ステレオ画像の平行化

一般のステレオビジョンでも一旦平行ステレオで撮影した画像に変換して行う



ローカル(ブロック)マッチング

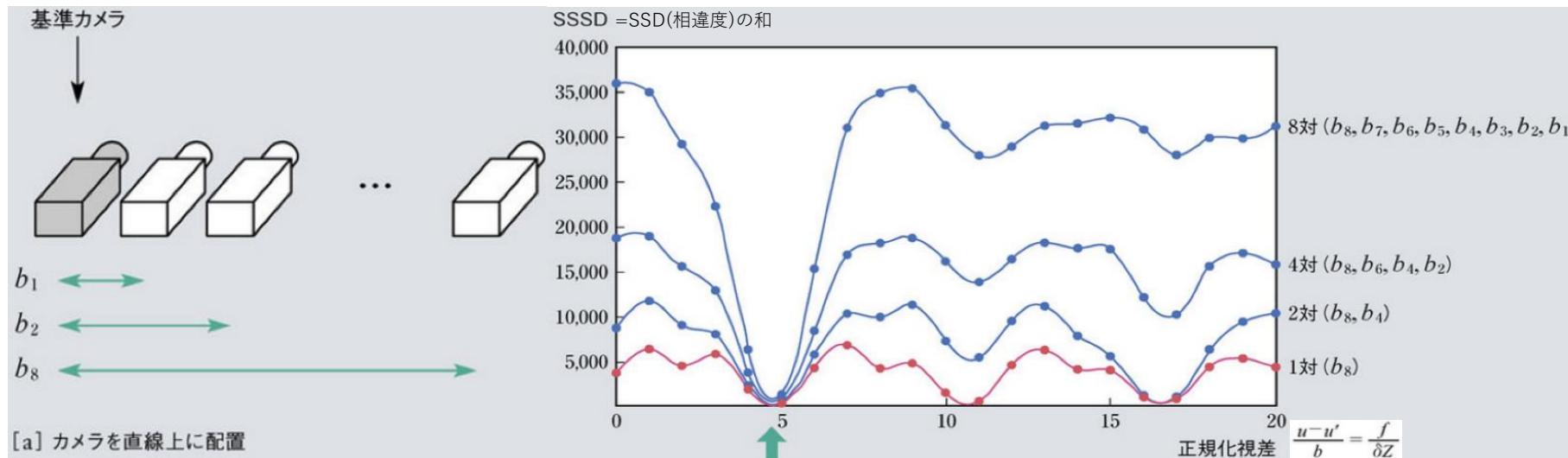
ウィンドウのサイズが小さいと視差の推定が不安定で、大きいとずれたり細かい形状がつぶれたりと精度が低い。

視差画像

(u, v) の位置を $(u-u', v)$ に置き換えた画像。 $u-u'$ は Z に反比例するため、明るいところほどカメラから遠くにあることを意味する。 Z 方向を正しく推定できているかの指標になる。

3台以上のカメラを用いたステレオビジョン(マルチビューステレオ)

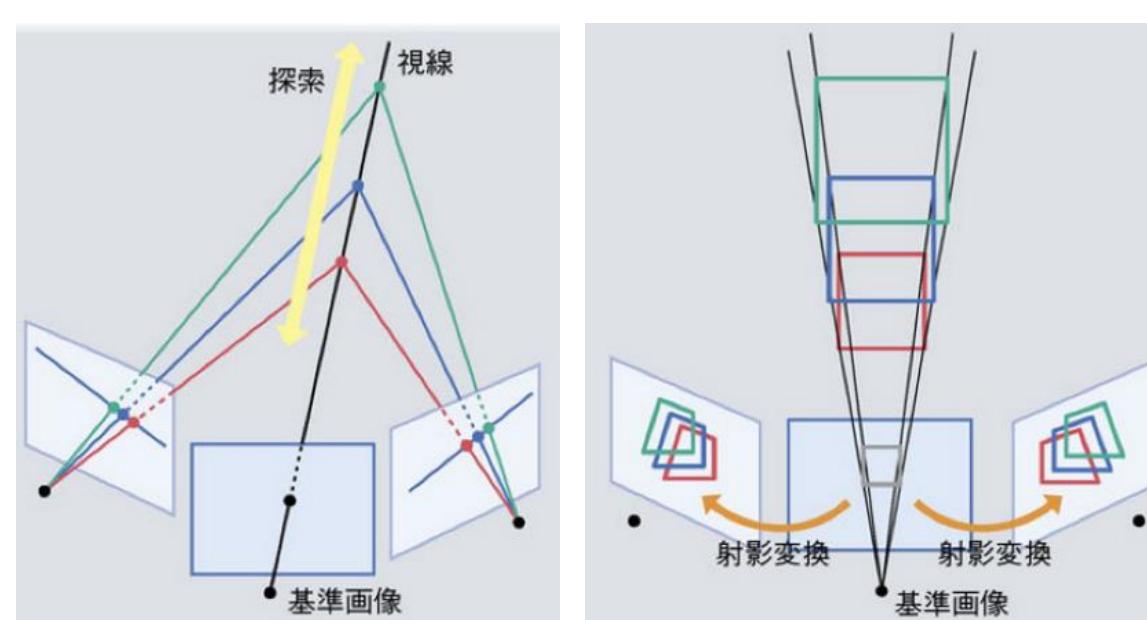
平行配置の場合(マルチベースラインステレオ)



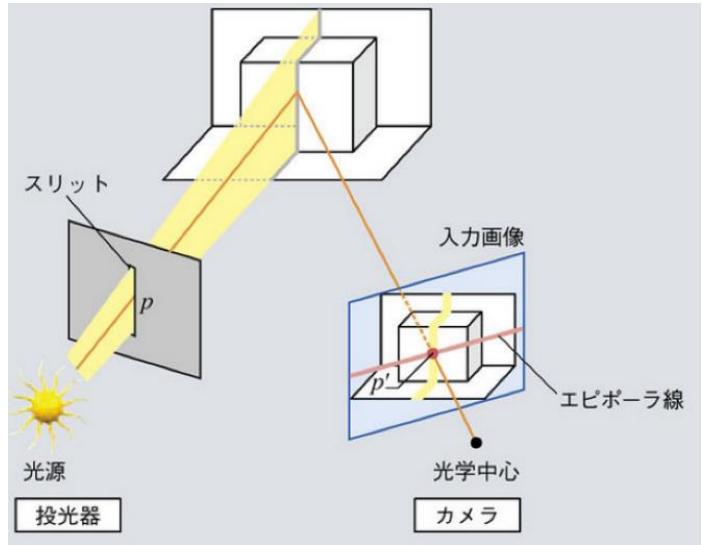
カメラの数が増えるにつれて、最小値がより明確に絞られていく

一般的な配置のマルチビューステレオ

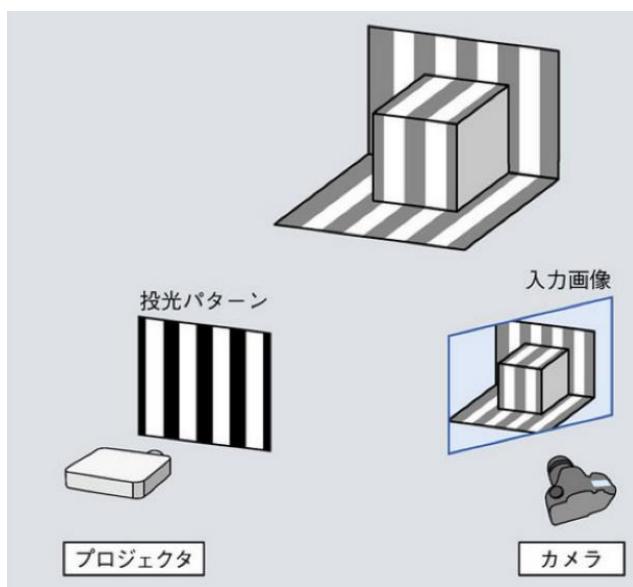
注目点に対する視線を考える
↓
視線上の一点を空間点と仮定
↓
その空間点の
他の複数のカメラへの投影点を参照
↓
各投影点の画素値が
最も一致する位置を空間点の位置と推定



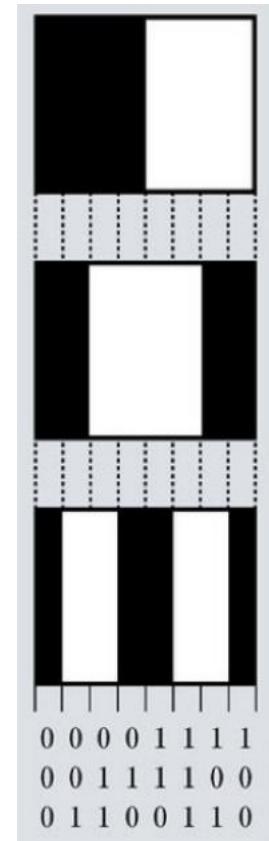
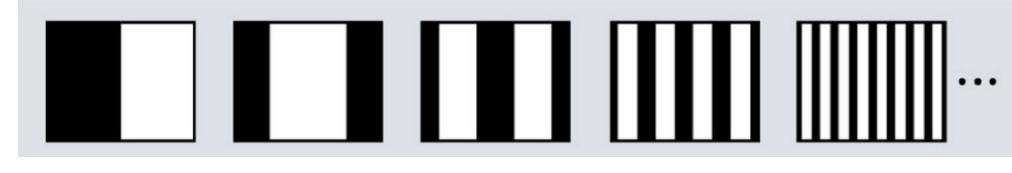
カメラを投光器に置き換えたステレオビジョン(アクティブステレオ)



ライン光投影法
 p に対応する p' =画像上のスリットとエピポーラ線の交点
↓
空間位置の推定



コード化パターン投影法
投光するストライプパターンをコード化
↓
入力画像の各画素に対してコードを検出
↓
各画素に対する投光器からの照射方向がわかる
↓
空間位置の推定



アクティブステレオは画像同士のマッチングが不要なため、より安定的。ただし適用できる対象や環境は限定的

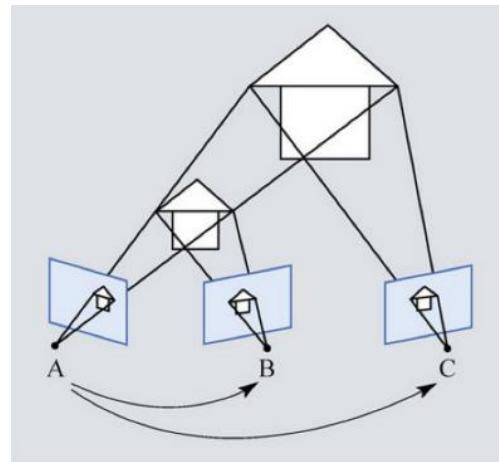
1台のカメラを動かして行うステレオビジョン

PnP問題

X_w と m の組み合わせを n 組知っている時の
方程式からカメラモーション
(カメラの位置と姿勢)を推定する問題

$$x = \frac{r_{11}X_w + r_{12}Y_w + r_{13}Z_w + t_1}{r_{31}X_w + r_{32}Y_w + r_{33}Z_w + t_3}$$

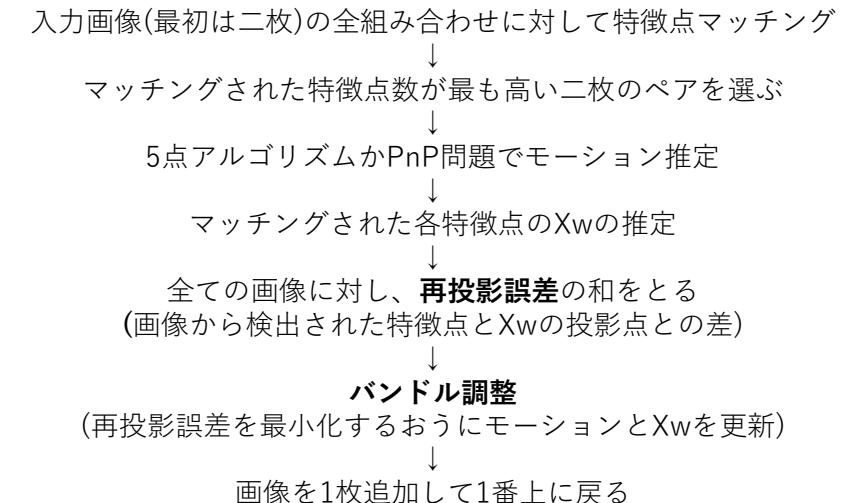
$$y = \frac{r_{21}X_w + r_{22}Y_w + r_{23}Z_w + t_2}{r_{31}X_w + r_{32}Y_w + r_{33}Z_w + t_3}$$



画像対応点既知でも大きさの不定性あり

Aのカメラ位置を基準とした場合、
Bの位置で撮影した画像と、
Cの位置で撮影した画像は、
一切区別がつかない

大量の画像(カメラ位置)を用いた ストラクチャfromモーション



カメラモーション
(カメラ1を基準としたカメラ2の位置・姿勢)の推定



カメラ位置が既知の場合と
同様の X_w の推定

5点アルゴリズム

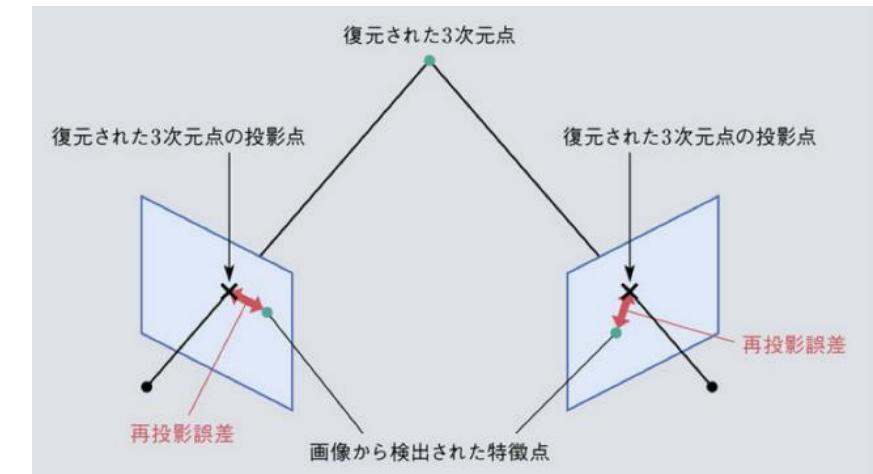
画像対応点を M 組得られている時、
未知数の数は $3M+5$ 、方程式の数は $4M$ となることから、
5組の画像対応点からを求めること。

8点アルゴリズム

2つのカメラ画像間の座標の関係式に
8組の画像対応点を代入し、カメラモーションを求めること。

$$\text{エピポーラ拘束} \quad \tilde{x}'^T E \tilde{x} = 0$$

$$(x' \quad y' \quad 1) \begin{pmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = 0$$

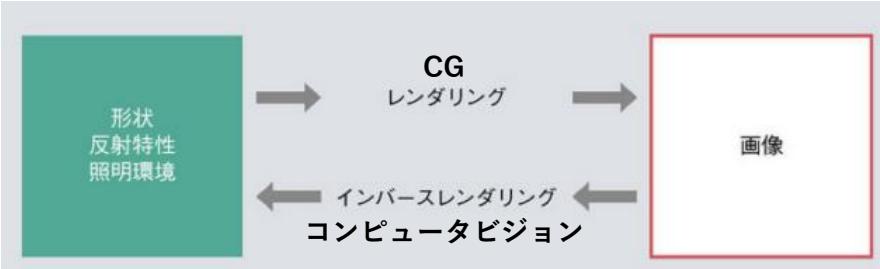
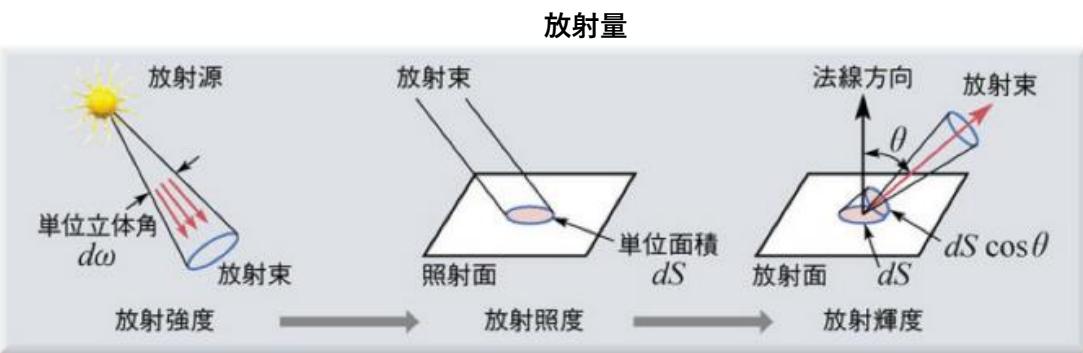
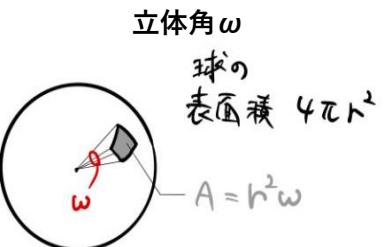


16. 画像からのシーン推定 (コンピュータビジョン2)

シーン推定の基礎知識1

シーン
物体の形状・反射特性(つや、ざらつき)・光源(照明環境)

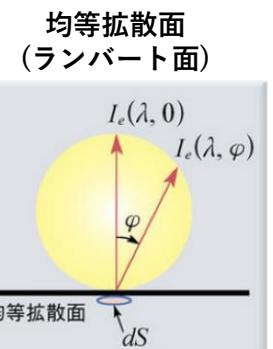
CG
シーンの記述からシーン中の光の伝搬を計算し、
画素値を求め画像を生成すること



$$I = \frac{d\Phi}{d\omega} \quad E_e(\lambda) = \frac{d\Phi_e(\lambda)}{dS} = I_e(\lambda) \frac{\cos \theta}{r^2}$$

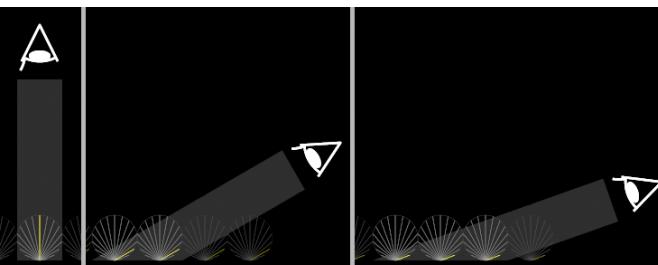
照度の逆二乗則
照度が強度に対して
距離の二乗に反比例

照度の入射角余弦則
照度が強度に対して
 $\cos \theta$ に比例



ランバートの余弦則
強度が $\cos \theta$ に比例して低下する

均等拡散面では、強度 I が $\cos \theta$ で減り、
同一視野の面積 dS は $1/\cos \theta$ で増えるので
増減が相殺し、輝度一定となる



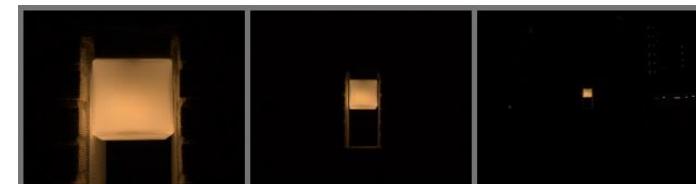
$$I_e(\lambda, \varphi) = I_e(\lambda, 0) \cos \varphi$$

$$L_e(\lambda, \varphi) = \frac{dI_e(\lambda, \varphi)}{dS \cos \varphi}$$

$$I_e(\lambda, \varphi) = I_e(\lambda, 0) \cos \varphi$$

$$L_e(\lambda, \varphi) = dI_e(\lambda, 0) / dS = L_e(\lambda, 0)$$

照度は距離の二乗に反比例するが、
見かけの面積も距離の二乗に反比例するため、
相殺し、輝度一定となる



均等拡散面の輝度は、
光源の方向 θ と放射強度 $I(\lambda)$ だけで決まる
すなわち、強度にも照度にも比例する

$$L_Q(\lambda) = I_e(\lambda) K_d(\lambda) \cos \theta$$

シーン推定の基礎知識2

双方向反射率分布関数 $f_{\text{BRDF}}(\theta_i, \phi_i, \theta_o, \phi_o)$
(Bidirectional Reflectance Distribution function)

不透明物体の反射特性を
入射方向と出射方向で表した4次元関数

Spatially-varying BRDF $f_{\text{svBRDF}}(x, y, \theta_i, \phi_i, \theta_o, \phi_o)$
反射特性が単一なBRDFで表せないような
不均質な不透明物体に対して、その反射特性を
入射位置、入射方向、出射方向で表した6次元関数

すべてのBRDFがもつ性質

エネルギー保存則

出射光エネルギー総量は
入射光エネルギー総量を超えない

$$\int_{\theta_o=0}^{\frac{\pi}{2}} \int_{\phi_o=0}^{2\pi} [f_{\text{BRDF}}(\theta_i, \phi_i, \theta_o, \phi_o) \cos \theta_o] \sin \theta_o d\theta_o d\phi_o \leq 1$$

ヘルムホルツの相反性

入射光の方向と出射光の方向を入れ替えてもBRDFの値は不变

$$f_{\text{BRDF}}(\theta_i, \phi_i, \theta_o, \phi_o) = f_{\text{BRDF}}(\theta_o, \phi_o, \theta_i, \phi_i)$$

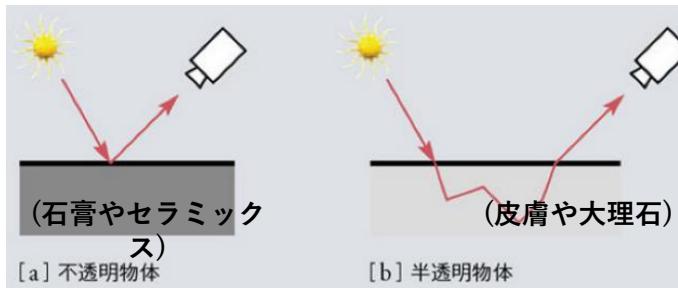
等方性

物体表面上の小領域について、光源方向と視線方向が作る平面を法線周りに回転させても明るさが変化しない性質

異方性

等方性をもたないこと

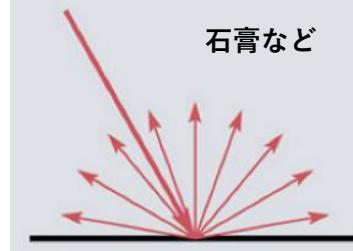
BRDF
入射光の明るさと出射光の明るさの関係を表す関数
→ 出射光の放射輝度と入射光の放射照度の比



反射モデル $f_{\text{BRDF}}(\theta_i, \phi_i, \theta_o, \phi_o) \cos \theta_i$

観測される放射輝度のモデル。
放射強度が法線と光源方向のなす角の余弦に依存することから導出

石膏など



[a] 拡散反射

光沢のある陶磁器など

[b] 鏡面反射

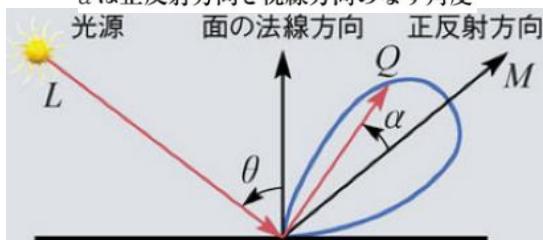
フォンモデル

鏡面反射の経験的なモデル

$$L_{\text{Phong}} = K_s (\cos \alpha)^n$$

K_s は鏡面反射率, n は表面の粗さを表す指数

α は正反射方向と視線方向のなす角度



$$f_{\text{BRDF}}(\theta_i, \phi_i, \theta_o, \phi_o) = \frac{dL(\theta_o, \phi_o)}{dE(\theta_i, \phi_i)}$$

双方向散乱面反射率分布関数 $f_{\text{BSSRDF}}(x, y, \theta_i, \phi_i, x_o, y_o, \theta_o, \phi_o)$
(Bidirectional Scattering Surface Reflectance Distribution function:BSSRDF)

半透明物体の反射特性を入射位置、入射方向、
出射位置、出射方向で表した8次元関数

ウォードモデル

rans-Sparrow のモデルを
表面の粗さが対象軸に平行な方向と
垂直な方向で異なる値を持つように
拡張したモデル

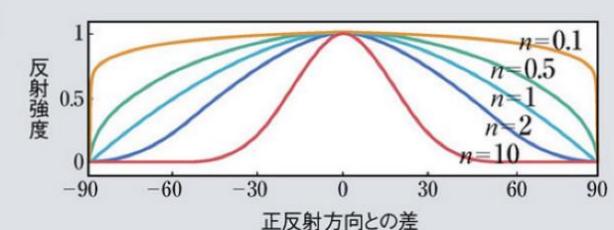
トランス-スパロウモデル

鏡面反射の正確なモデル

$$L_{\text{TS}} = K_s \frac{DGF}{\cos \theta_o}$$

正反射方向で
最も明るい

n が大きいほど
理想的な鏡に近づく



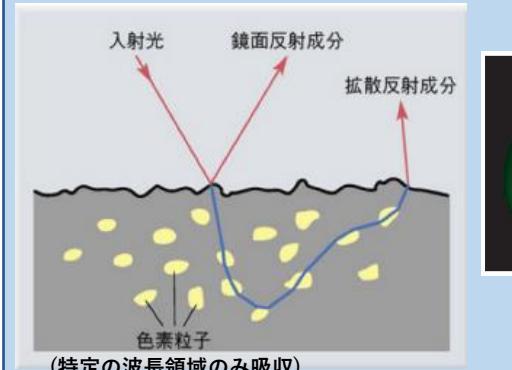
拡散・鏡面反射モデル

	拡散(diffuse)	鏡面(specular)
基本	ランバート(Lambert)	フォン(Phong)
正確	オーレンネイバー	トランススパロウ
拡張		ウォード
覚え方(アルファベット順)	D LO	S PTW

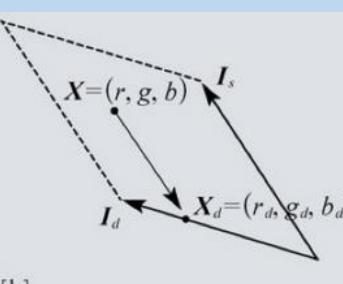
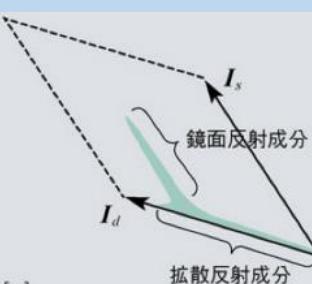
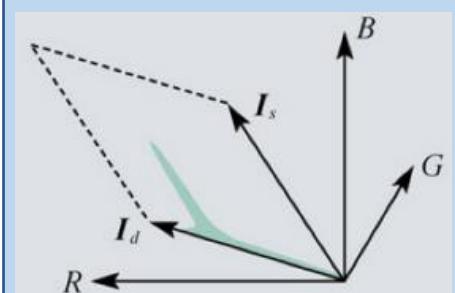
反射成分の分離

コンピュータビジョンアルゴリズムでは、しばしば画像が拡散反射成分のみで構成されることを仮定する。しかし、実際の画像には鏡面反射成分が含まれることが多く、CVの前処理として鏡面反射成分を除去することが必要になる。すなわち、拡散反射成分と鏡面反射成分を分離しなくてはならない。

二色性反射モデル



均質な反射特性をもつ物体が発する光はRGBの比が一定で強度のみ角度に依存する。従って、拡散反射成分の3次元単位ベクトルを I_d 、鏡面反射成分の3次元単位色ベクトルを I_s とする、物体表面の色はこれらの線形結合で表され、全画素値の色分布は I_d と I_s により張られる平面上に存在する。 $X=(r,g,b)$ が得られたら、 X を I_s に平行に射影した先として、 X_d を見つけることができる。



[a] 入力画像



[b] 拡散反射成分

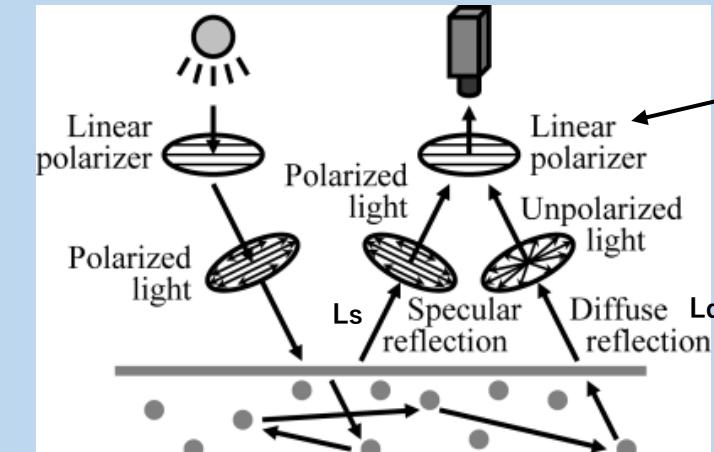


[c] 鏡面反射成分

図16.21——2色性反射モデルに基づく反射成分分離の例 (提供:カーネギーメロン大学 金出武雄)

偏光に基づく分離

鏡面反射成分は、出射光の偏光状態は、入射光の偏光状態に等しい。拡散反射成分は、物体表面層において散乱するため、出射光の偏光状態は、入射光によらず非偏光。



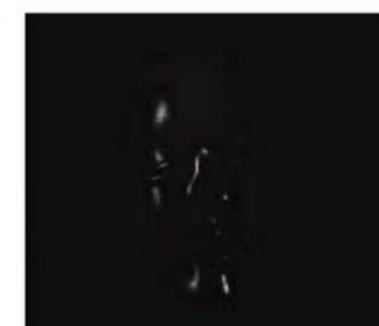
$$L_{\max} = \frac{1}{2} L_d + L_s$$

$$L_{\min} = \frac{1}{2} L_d$$

図のように直線偏光板を二枚配置し、カメラ側の偏光板は回転させる。拡散反射成分は非偏光であるため、明るさは不变だが、鏡面反射成分は偏光しているため、変動する。輝度の最小値の二倍が拡散反射成分となる。



[a] 鏡面反射成分と拡散反射成分からなる画像



[b] 鏡面反射成分



[c] 拡散反射成分

表面の明るさから面の向きを推定(形状復元)

物体表面がランバート面

照度差ステレオ

面法線ベクトルは
その面の高さの勾配から求められる

$$\mathbf{n} = \begin{pmatrix} n_x \\ n_y \\ n_z \end{pmatrix} = \frac{1}{\sqrt{z_x^2 + z_y^2 + 1}} \begin{pmatrix} -z_x \\ -z_y \\ 1 \end{pmatrix}$$

逆に言えば、
面法線ベクトルがわかれれば、
その面の高さ(位置)がわかる

入力画像のすべての画素について
面法線ベクトル(法線と反射率)を推定

法線から勾配を求める。

積分して高さを求める。

3次元形状の復元

面法線ベクトル $\tilde{\mathbf{n}} (=K_d \mathbf{n})$ の推定2

$$i = K_d L \cos \theta \quad K_d = 1$$

$$\theta_1 = \cos^{-1}(i_1/L_1)$$

法線ベクトルと単位球面の交点は、
3つの光源との角度 θ によって作られる
球面上の3円の交点に一意に定まる。

未較正照度差ステレオ

光源の強度と方向が未知の照度差ステレオ。
光源の較正が不要である一方、
法線、反射率、光源ベクトルが
無数に存在してしまうため、
形状復元には曖昧さがある。

面法線ベクトル $\tilde{\mathbf{n}} (=K_d \mathbf{n})$ の推定1

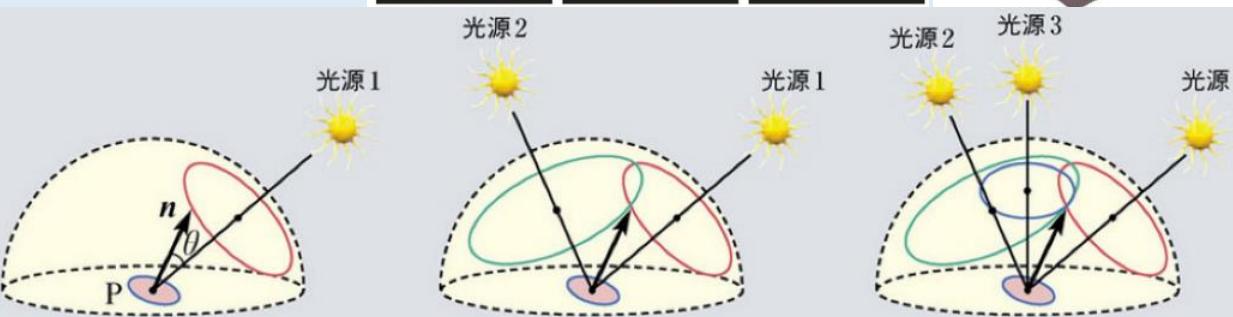
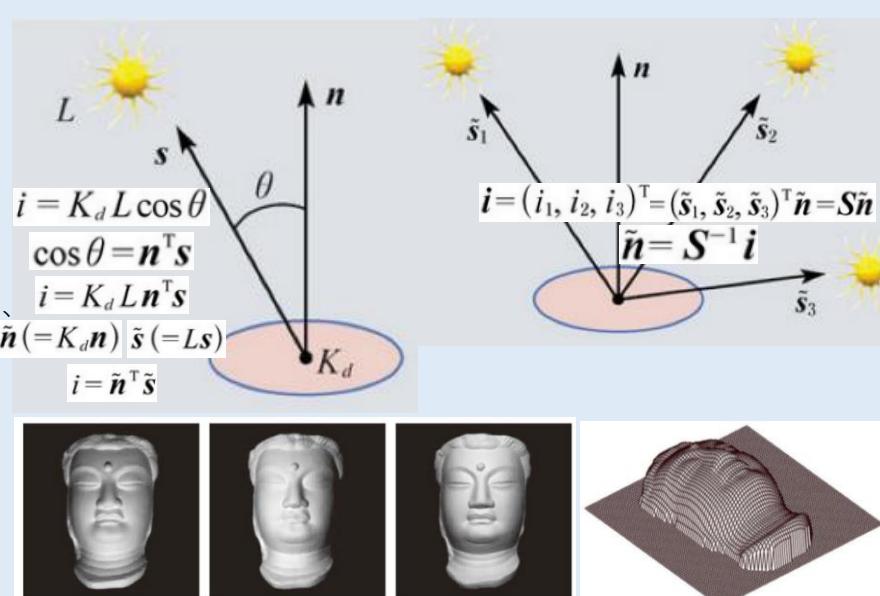
3つの異なる光源で
順に物体を照らす

各光源の強度と方向を表す
光源行列 S は既知とする

それぞれの光源での同一の点の
明るさを要素に持つベクトルを i とし、
これを観測する

n は S の逆行列と i の行列積となる

m 個の光源を利用する
場合は、 S の疑似逆行列を
用いた最小二乗法



シェイププロムシェーディング

1枚の画像のみから形状を復元する手法。
拘束条件が不足するため、
反射率が既知、物体表面の高さが滑らか、
輪郭線における法線が既知などの
拘束条件を追加する必要がある。

物体表面がランバート面ではない場合

参照物体を利用した照度差ステレオ

ランバート面でない場合は、 $i = K_d L \cos \theta$ が
成り立たないため、解析的に解けない。

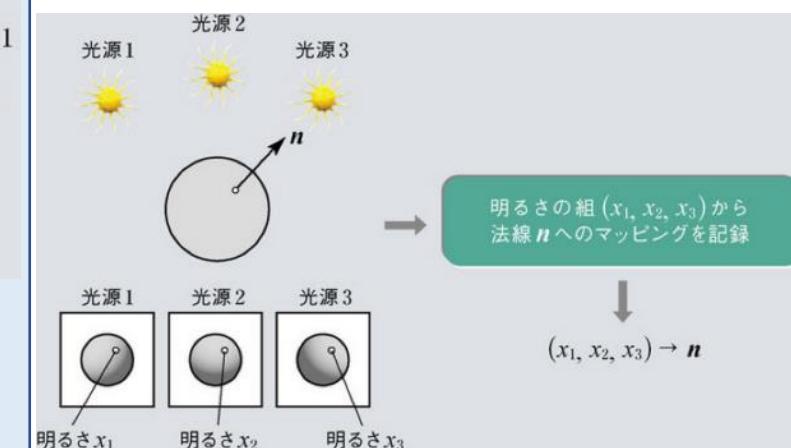
ランバート面の手法では、異なる方向に光源がある時の
明るさの組 (i_1, i_2, i_3) から法線を求めた

対象物体と反射特性が同じかつ法線が既知の物体(球体など)
について、異なる方向に光源をおいて画像を撮影

参照物体の明るさの組 (x_1, x_2, x_3) と面法線ベクトルを
対応させた参照テーブルを作成

参照物体撮影時と同じ位置に光源を置き、対象物体を撮影

対象物体の各画素について明るさの組 (i_1, i_2, i_3) と
参照テーブルから、面法線ベクトルを得る



反射特性(BRDF)の復元

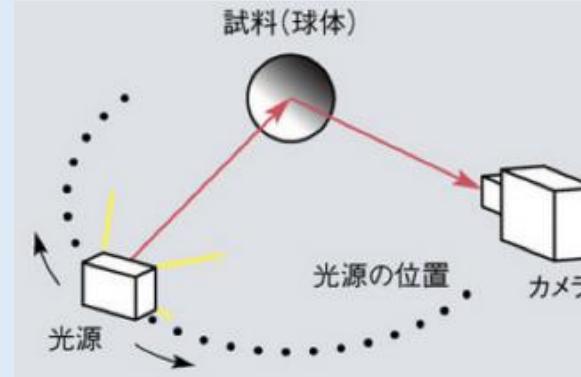
反射特性が均質な物体の場合

ゴニオリフレクトメータ



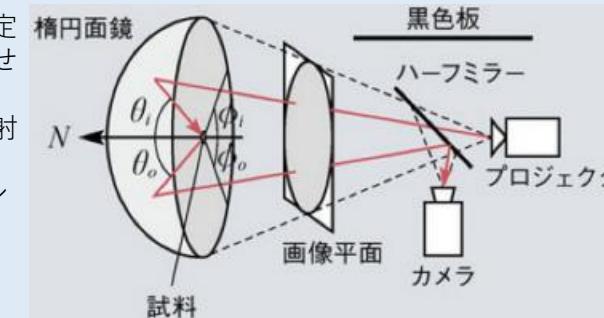
BRDFの計測機器。計測対象の周りで、光源と高度計を回転させながら明るさをサンプリングする。膨大な計測時間を要するという欠点がある。

画像を用いた方法



カメラと試料を固定して光源を回転させる。1枚の画像で、様々な入射光・出射光の方向におけるBRDFを同時にサンプリングできる。

光学系を用いた方法



橢円面鏡の二つの焦点位置に試料とハーフミラー、プロジェクタを配置し試料を様々な方向から同時に照明・観察する。機械的な回転を伴わないため、高速な計測が可能だが、工学系が複雑になるほど較正も大変になる。

反射特性が均質ではない物体の場合



実物体の多くは反射特性が均質ではないが、とはいえ各点のBRDFを計測するのは膨大な時間要する。

反射モデルパラメータ(拡散反射の三つの反射率、鏡面反射の反射率、表面粗さ)を推定。

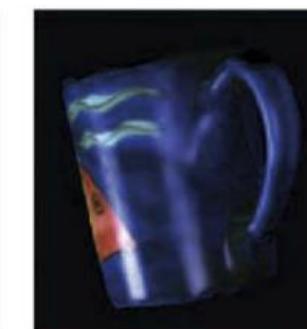
具体的には対象物体を回転させたときに物体表面上のある点で観察される明るさ・色を分離し、最小二乗法を用いる。



[a] 入力画像



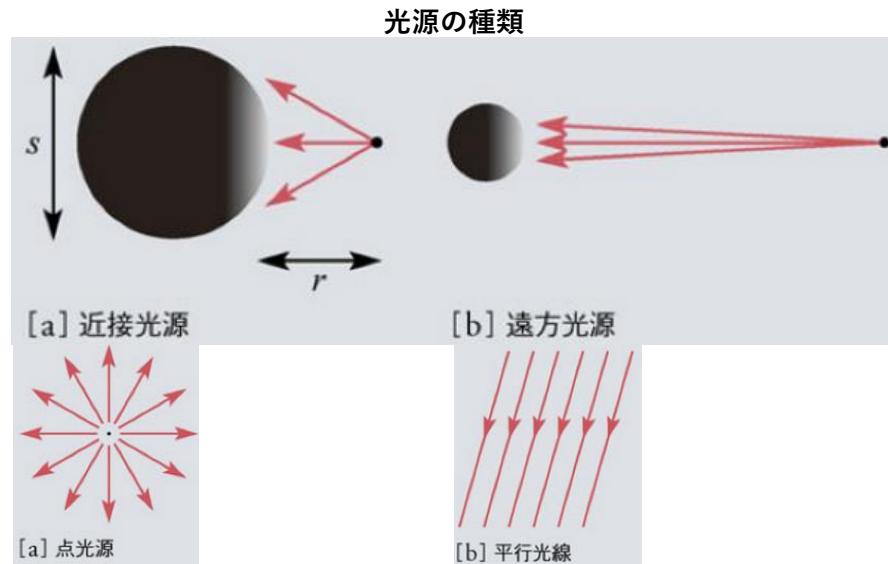
[b] 入力画像の再現



■図16.37——照明条件の異なる生成画像



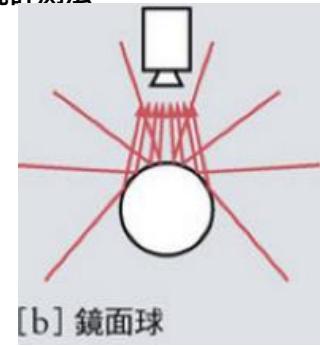
照明環境(光源分布)の復元



遠方光源に照らされたシーンの光源分布 $L(\theta, \phi)$
シーンに (θ, ϕ) から入射する光線の強度

イメージベストライティング(IBL)

実シーンの照明環境を計測して、仮想物体を描画する際に用いることで写実的な画像を生成する。



魚眼レンズ付きカメラで、様々な方向から入射する光の強度(光源分布)を直接得る。光源分布はダイナミックレンジが広いので複数回のハイダイナミックレンジ撮影を行う。

鏡面球の表面では、光は正反射方向のみに反射されるため光線を視点から逆にたどっていけば光源分布を求められる。撮影方法は、魚眼レンズと同様ハイダイナミックレンジ。

シーン中の物体の陰影を用いた照明環境計測法: インバースライティング

光の重ね合わせの原理に基づけば、シーン中の各点の明るさは、シーン中の全光源の明るさの和で表現される。従って、

$$L(\theta, \phi) = \sum_{n=1}^N c_n L^{(n)}(\theta, \phi) \quad c_n \text{は結合係数}$$

n番目の基底光源の下で観測されるシーン中のある点xmの明るさを $R_m(L_n)$ とすると、明るさ i_m は、

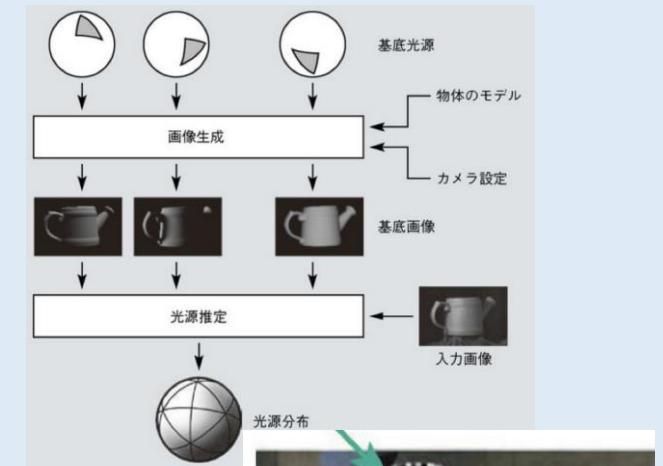
$$i_m = \sum_{n=1}^N c_n R_m(L^{(n)})$$

行列で表すと、

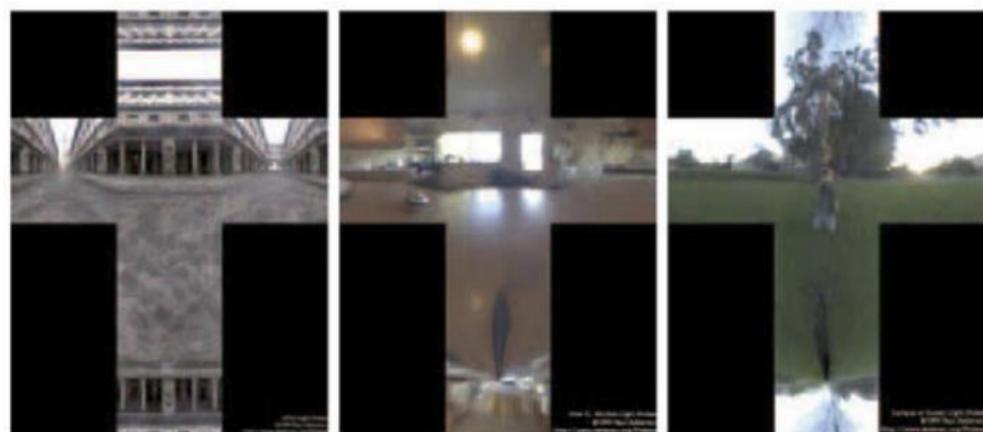
$$\begin{pmatrix} \vdots \\ i_m \\ \vdots \end{pmatrix} = \begin{pmatrix} \cdots & R_m(L^{(1)}) & \cdots \\ & \vdots & \end{pmatrix} \begin{pmatrix} \vdots \\ c_1 \\ \vdots \end{pmatrix}$$

$$i = Rc$$

両辺に R の疑似逆行列をかけば、結合係数 c を推定でき、それを最初の式に代入することで、光源分布が得られる。



環境マップ
仮想物体を取り囲む大きな立方体や球を考え、それらの内側各点に、対応する方向の光源の強度を貼り付け、光源とする。
仮想物体の描画は環境マップを光源とみなして光の伝搬を計算する。



すべての復元

幾何学的解析と光学的解析の融合

ストラクチャラムモーション * ステレオビジョンでシーンの初期形状復元



初期形状とともに法線推定 * 反射特性推定 * 照明環境推定で
詳細形状・拡散反射率・光源分布の復元

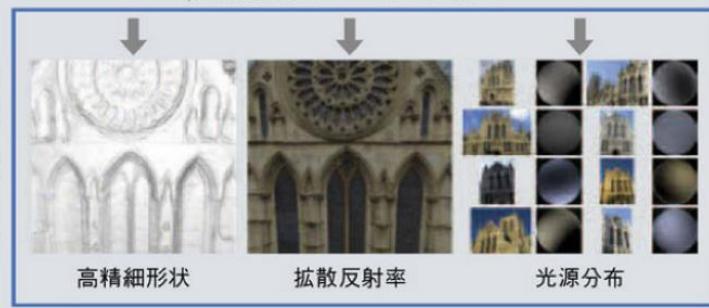
多視点インバースレンダリング



入力画像



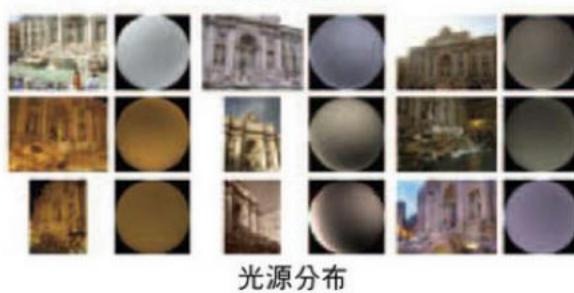
初期形状
(SfM + MVS)



拡散反射率



形状



光源分布

データ駆動型(深層学習)アプローチ

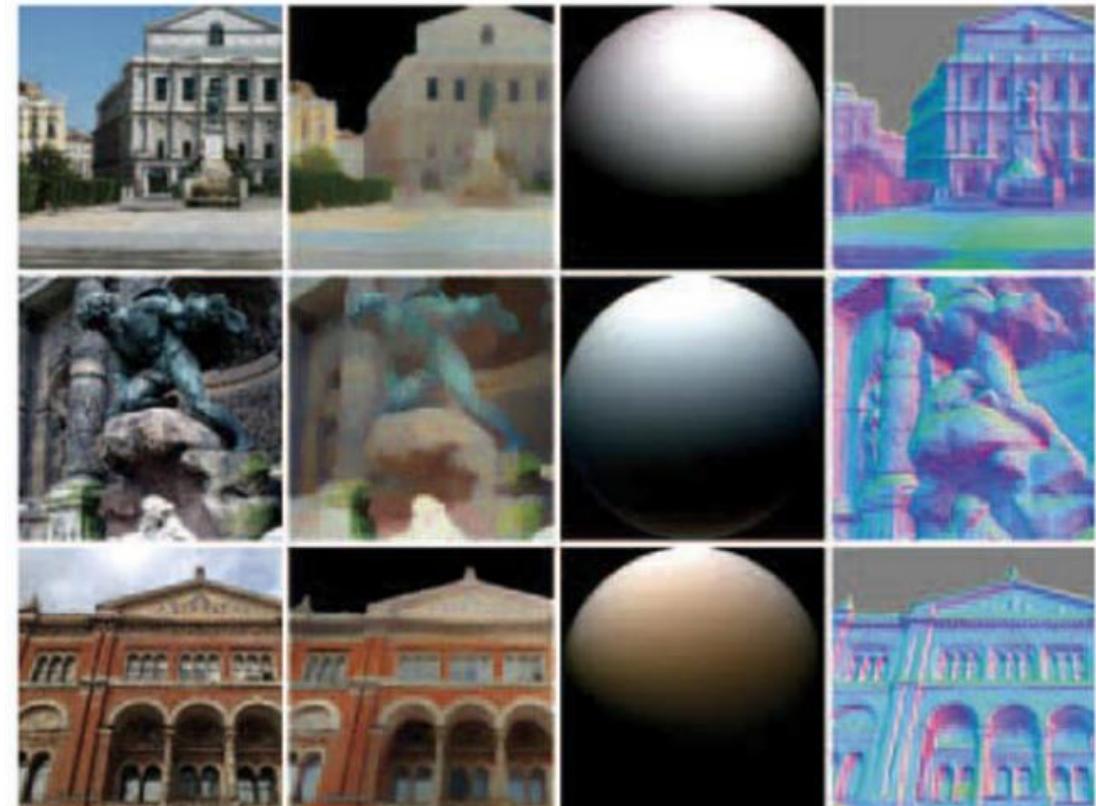
ストラクチャラムモーション・ステレオビジョンにより復元したシーン情報を
正解ラベルとして、単一画像から拡散反射率・法線を出力するよう学習したモデル推論



画素値を反射率で割ることで陰影を得る



陰影と法線を用いて光源分布を推定



入力画像

拡散反射率

光源分布

法線

17. 画像符号化

二値画像の符号化

run

ランレングス符号化(run-length coding)

0 0 0 0 0 0 1 1 1 1 0 0 0 0
0 0 0 0 1 1 1 1 1 1 1 0 0 0
0 0 0 0 1 1 1 1 1 1 1 1 1 0
0 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 0 0 1 1 1 1 1 1
1 1 1 1 1 1 1 0 0 1 1 1 1 0
0 1 1 1 1 1 1 1 1 1 1 1 1 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 1 1 1 1 1 1 1 1 1 0 0
0 0 0 1 1 1 1 1 1 1 1 1 0 0
0 0 0 0 0 0 1 1 1 1 1 1 0 0

[a] 2値画像

0;4, 1;3, 0;4,

0;3, 1;6, 0;2,

0;3, 1;7, 0;1,

0;1, 1;10,

1;4, 0;2, 1;5,

1;5, 0;2, 1;3, 0;1,

0;1, 1;9, 0;1,

0;1, 1;8, 0;2,

0;2, 1;7, 0;2,

0;3, 1;5, 0;3,

0;4, 1;2, 0;5.

4, 3, 4,

3, 6, 2,

3, 7, 1,

1, 10,

0, 4, 2, 5,

0, 5, 2, 3, 1,

1, 9, 1,

1, 8, 2,

2, 7, 2,

3, 5, 3,

4, 2, 5.

[b] ランレングスコード

[c] ランの種類を省略したランレングスコード

1次元符号化方式(MH)

ランレングスの発展

2次元圧縮方式(RAC,MR,MMR)

上下の行の変化は小さいと予測されるから、
上の行と変化しているラインのみを検出し圧縮

参照ラインがNot MH → MR(Modified Read)方式
参照ラインがMH→ MMR(Modified Modified Read)方式

2ライン一括符号化方式
上下に隣接する二行をいっぺんに符号化。

0 0 0 0 1 1 1 1 0 0 0
0 0 0 1 1 1 1 1 1 0 0
0 0 1 1 0 0 0 1 1 1 0
0 1 1 0 0 0 0 0 0 1 1
1 1 0 0 0 0 0 0 0 1 1
1 1 0 0 0 0 0 0 0 1 0
0 1 1 0 0 0 0 0 1 1 0
0 1 1 0 0 0 0 1 1 0 0
0 0 1 1 1 1 1 1 1 0 0
0 0 0 1 1 1 1 1 1 0 0

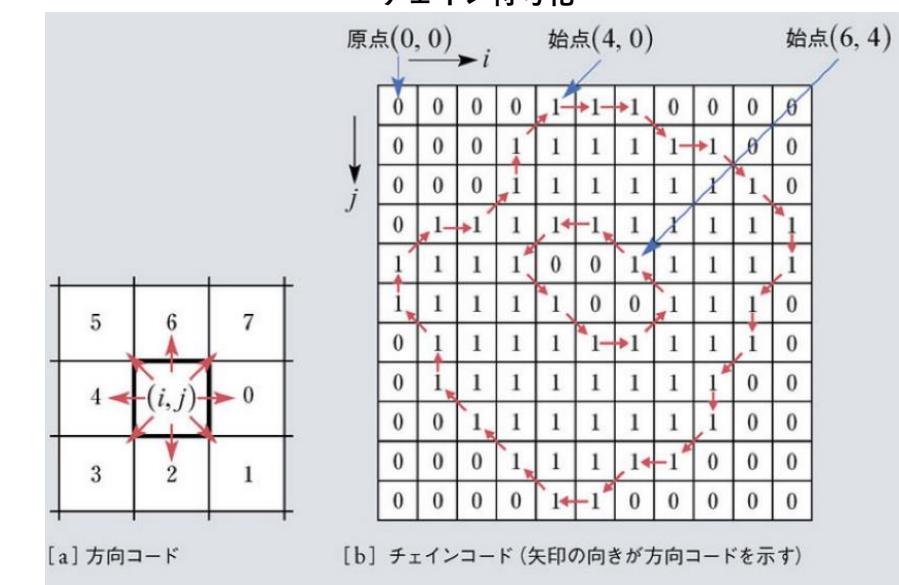
00, 3, 01, 1, 11, 4, 01, 1, 00, 2.

00, 1, 01, 1, 11, 1, 10, 1, 00, 3, 10, 2, 11, 1, 01, 1.

11, 2, 00, 7, 11, 1, 10, 1.

00, 1, 11, 2, 00, 4, 01, 1, 11, 1, 10, 1, 00, 1.

00, 2, 10, 1, 11, 5, 10, 1, 00, 2.

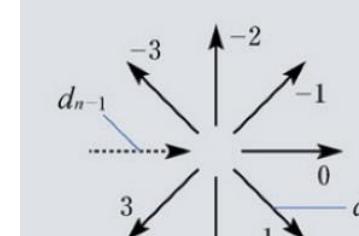


外側輪郭線 (4,0), 0, 0, 1, 0, 1, 1, 2, 3, 2, 3, 2, 3, 4, 3, 4, 5, 5, 5, 6, 5, 6, 7, 0, 7, 6, 7.

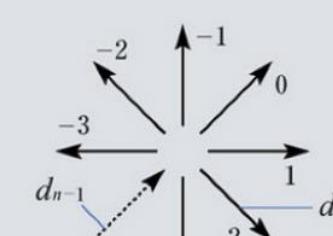
内側輪郭線 (6,4), 5, 4, 3, 1, 1, 0, 7, 5.

差分チェインコード

入る方向と出る方向が一緒なら0、上側は負、下側は正として差分のみをコード化



[a] 前回の方向コード $d_{n-1}=0$ のとき



[b] 前回の方向コード $d_{n-1}=7$ のとき

外側輪郭線 (4,0), 0, 0, 1, -1, 1, 0, 1, 1, -1, 1, 1, -1, 1, 1, 0, 0, 1, -1, 1, 1, 1, -1, 1.

エントロピー —

$$H(p) = \sum_{f=0}^{L-1} p(f) \log_2 \frac{1}{p(f)}$$

画像のエントロピー=画素値の平均情報量

エントロピーが小さいほど高圧縮率といえる。

$p(f)$ は画素値fの出現頻度

濃淡ヒストグラムが同じならエントロピーも同じになる。

平均符号化長をエントロピーより短くすることは理論的に不可能

グレースケール画像の符号化: ハフマン符号化

エントロピー(情報量)

画素値が1種の時0となり、増えるにつれて増大し、ヒストグラムが一様になる時最大となる。すなわち、エントロピーは画像の情報の多さを意味し、**出現確率が高い画素値ほど低いビット数で表現**することで画像を圧縮できる。言い換えれば、エントロピーは圧縮のしにくさを意味する。

$$H(p) = \sum_{f=0}^{L-1} p(f) \log_2 \frac{1}{p(f)}$$

平均符号長(符号長の期待値)

画素ごとにビット数を変えて圧縮すると
き、どれだけ圧縮されたかを示す指標。
この値を小さくするように圧縮する。

$$\lambda = \sum_{f=0}^{L-1} p(f) l(f)$$

可逆符号化

プログラムや数値データなど、圧縮後に元通りに復元することが不可欠なデータに対する符号化。

非可逆符号化

画像データなど、人間にとて知覚しづらい情報を含むデータはに対しそれらを欠損させたより高い圧縮率の符号化。

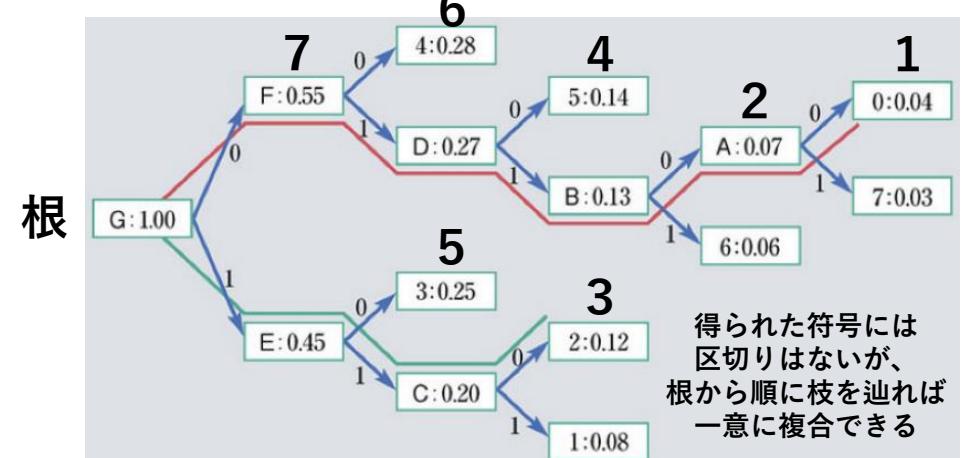
ハフマン符号化

出現確率の低い二つのシンボル(画素値の変換値)を選択

大きい方に符号0、小さい方に符号1を割り当て部分木作成

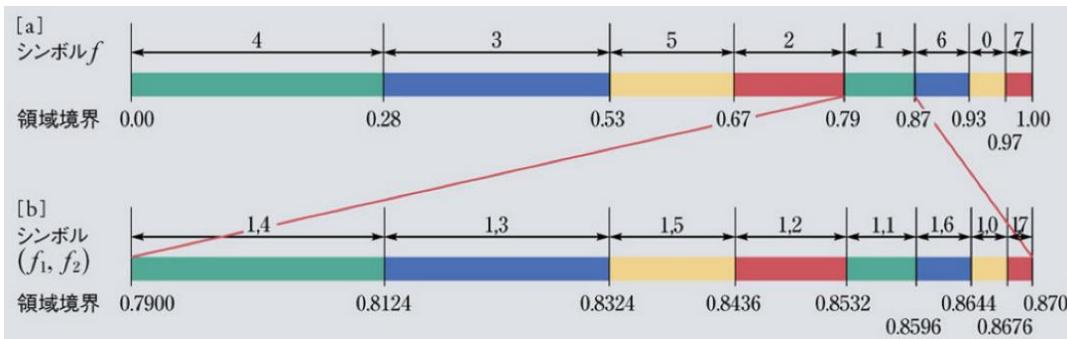
二つの出現確率の和を出現確率とする新たなシンボルに統合

一番最初に戻る



シンボル列(画素値例) 326301 \longleftrightarrow 1011001111001100111

グレースケール画像の符号化: 算術符号化



1シンボルに対して
各シンボルを出現確率の大きい順に並べ直す

↓
出現確率を累積して各シンボルの下限・上限を求める

↓
各シンボルを他と区別するために必要な最小桁数の二進数を符号とする

シンボルの組み合わせに対して

1つ目のシンボルの領域を2つ目のシンボルの出現確率で分割し
それぞれの領域の下限・上限を求める

↓
各領域を他と区別するために必要な最小桁数の二進数を符号とすることで
より圧縮率を高められる

原理的にはハフマン符号化より圧縮率は高いが、
実行時間は長くなる

画素値 f	出現確率 $p(f)$	各シンボルの領域の下限・上限			算術符号 $c(f)$	符号長 $l(f)$
		下限(10進数)	上限(10進数)	下限(2進数)		
4	0.28	0.00	0.28	0.0000000...	00	2
3	0.25	0.28	0.53	0.01000111...	01	2
5	0.14	0.53	0.67	0.10000111...	100	3
2	0.12	0.67	0.79	0.10101011...	101	3
1	0.08	0.79	0.87	0.11001010...	1100	4
6	0.06	0.87	0.93	0.11011110...	1101	4
0	0.04	0.93	0.97	0.11101110...	1110	4
7	0.03	0.97	1.00	0.11111000...	1111	4
平均符号長						2.68ビット

連続する 画素値 f_1, f_2	出現確率 $p(f_1, f_2)$	連続する2つのシンボルの領域の下限・上限			算術符号 $c(f_1, f_2)$	符号長 $l(f_1, f_2)$	符号長 $l(f_1)+l(f_2)$
		下限(10進数)	上限(10進数)	下限(2進数)			
1,4	0.08×0.28	0.7900	0.8124	0.110010100011...	110010	6	6
1,3	0.08×0.25	0.8124	0.8324	0.110011111111...	110011	6	6
1,5	0.08×0.14	0.8324	0.8436	0.110101010001...	1101010	7	7
1,2	0.08×0.12	0.8436	0.8532	0.110101111111...	1101011	7	7
1,1	0.08×0.08	0.8532	0.8596	0.110110100110...	110110	6	8
1,6	0.08×0.06	0.8596	0.8644	0.110111000000...	11011100	8	8
1,0	0.08×0.04	0.8644	0.8676	0.110111010100...	11011101	8	8
1,7	0.08×0.03	0.8676	0.8700	0.110111100001...	1101111	7	8

グレースケール画像の符号化:予測符号化

予測符号化

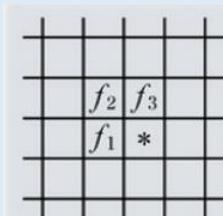
予測値を利用して画素値をエントロピーの小さいシンボルに変換し、複合時は逆変換により元のシンボルに戻す。

DPCM符号化(Differential Pulse Code Modulation)

注目の画素の画素値とその予測値との差分をシンボルとして、値のバリエーションを大幅に減らした予測符号化

近隣画素値

注目画素の1つ左や上の画素値



*:注目画素 $f(i, j)$
 f_1, f_2, f_3 :注目画素に対する近隣画素
予測値: $\hat{f}(i, j) = f_1$

近隣画素値の線形結合

$$\hat{f}(i, j) = f_1/2 + (f_2 + f_3)/4$$

近隣画素値の線形結合の最小二乗解

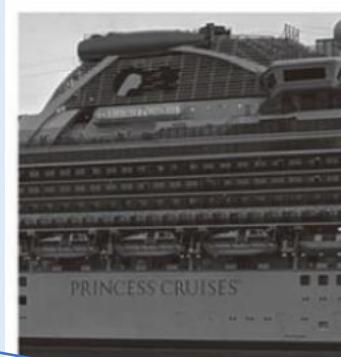
画像全体に対して注目画素とその予測値の差が最小になるような予測係数を求める

$$\hat{f}(i, j) = a_1 f_1 + a_2 f_2 + \dots + a_N f_N$$

端画素値

画像の左端の列や、上端行の画素値

値のバリエーション



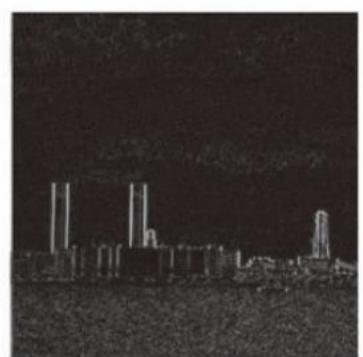
[a] 原画像



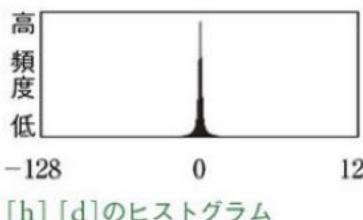
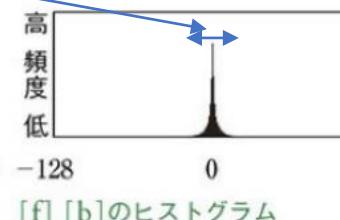
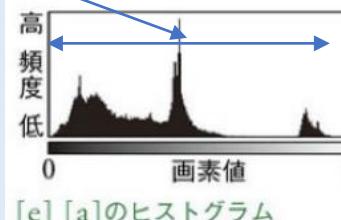
[b] 差分画像



[c] 原画像



[d] 差分画像



グレースケール画像の符号化・変換符号化

変換符号化

画像を周波数領域に類似した領域に変換してから符号化し、複合時に逆変換して元に戻す。

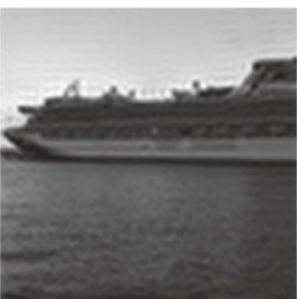
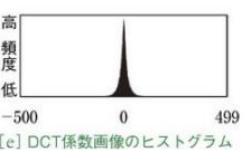
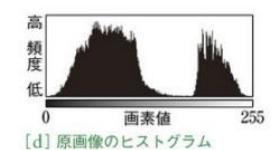
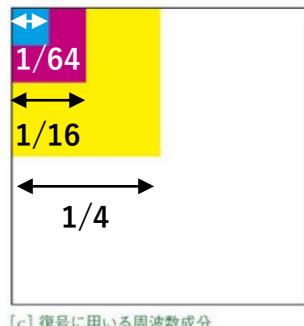
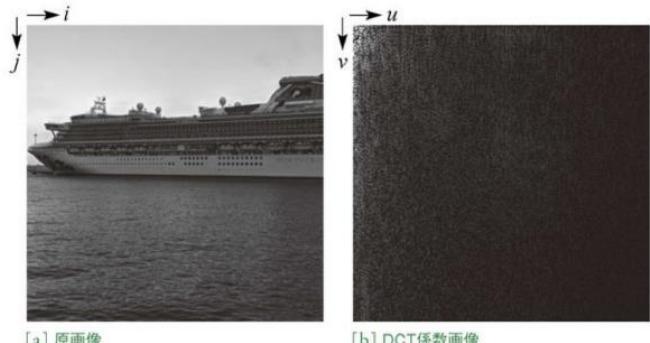
フーリエ変換は変換結果が複素数となるため、代わりに離散コサイン変換、離散ウェーブレット変換を用いる。

離散コサイン変換(Discrete Cosine Transform:DCT)

変換元の関数を水平方向、垂直方向に折り返して、偶関数に拡張してから変換。JPEG,MPEGで用いられている。

$$F(u, v) = \frac{2}{\sqrt{MN}} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} f(i, j) C(u) C(v) \cos \frac{\pi u(2i+1)}{2M} \cos \frac{\pi v(2j+1)}{2N}$$

$$f(i, j) = \frac{2}{\sqrt{MN}} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) C(u) C(v) \cos \frac{\pi u(2i+1)}{2M} \cos \frac{\pi v(2j+1)}{2N}$$

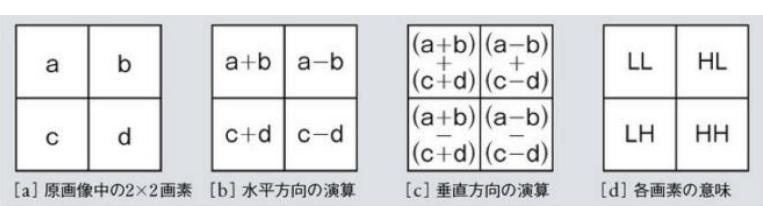


DCTとは異なり、画像上の**特定の位置の周波数、振幅**も得られる変換。ただし、周波数分解能と位置分解能はトレードオフの関係。1つの平均と複数の差分に変換することで圧縮できる。

$$W(a, b) = \int_{-\infty}^{\infty} f(x) \phi_{a,b}(x) dx \quad f(x) = \frac{2}{C_\phi} \int_0^{\infty} \left(\int_{-\infty}^{\infty} W(a, b) \phi_{a,b}(x) db \right) \frac{da}{a^2} \quad \phi_{a,b}(x) = \frac{1}{\sqrt{a}} \phi\left(\frac{x-b}{a}\right)$$

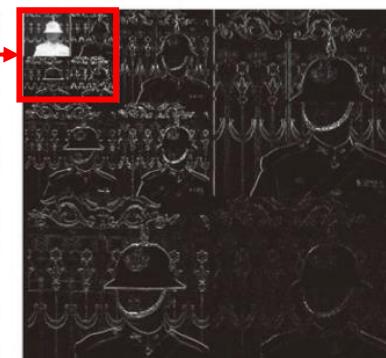
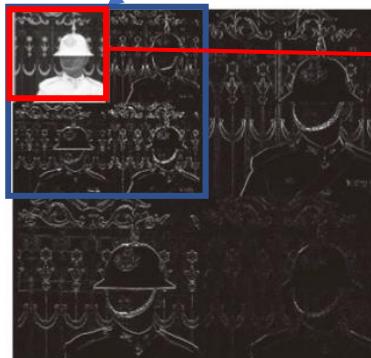
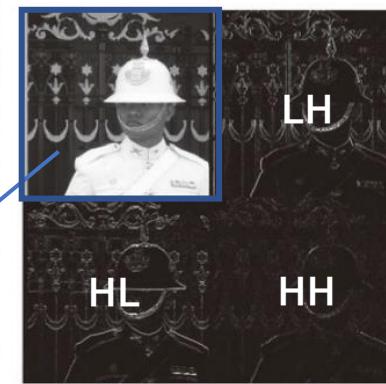
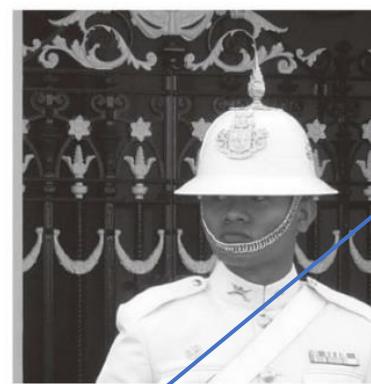
多重解像度分析

離散化したウェーブレット変換(Discrete Wavelet Transformation:DWT)を用いて、信号を異なるスケールに分解して解析すること。



上の処理を全画素にわたって行い、LLを左上に揃えれば、平均と差分に変換した画像が得られる。さらに得られた平均画像に対し、変換を繰り返せばより圧縮できる。平均画像の圧縮率が $1/2^j$ の時、 j をスケールという。LL,HL,LH,HHはXYにHigh、Lowを対応させた表記。すなわち、位置と周波数成分の対応がわかる。

JPEGは、画像を 8×8 画素のブロックごとにDCTを行い、ブロック全体を異なる周波数で同じ振幅の正弦波の重ね合わせで表すため、特有のノイズが表れる。改良版のJPEG2000では、DWTが使われノイズが改善されている



離散ウェーブレット変換と多重解像度分析

ウェーブレット変換による符号化の特徴

- ①画像全体の 2×2 領域ごとの平均、横、縦、斜めの差分を抽出するため、元のデータを完全に再構成可能
- ②DCTのようにブロックに分けて処理を行うわけではないのでブロックノイズは生じない
- ③各位置でのエッジを抽出するため、局所領域での特徴を捉えられる。
- ④平均と各差分画像は元の画像の半分のサイズ
- ⑤差分画像を全て0にした状態で逆変換を行っても、ある程度は復元可能

$\begin{array}{ c c } \hline a & b \\ \hline c & d \\ \hline \end{array}$	$\begin{array}{ c c } \hline a+b & a-b \\ \hline c+d & c-d \\ \hline \end{array}$	$\begin{array}{ c c } \hline (a+b) + (c+d) & (a-b) + (c-d) \\ \hline (a+b) - (c+d) & (a-b) - (c-d) \\ \hline \end{array}$	$\begin{array}{ c c } \hline LL & HL \\ \hline LH & HH \\ \hline \end{array}$
[a] 原画像中の 2×2 画素	[b] 水平方向の演算	[c] 垂直方向の演算	[d] 各画素の意味



カラー画像の符号化

JPEG

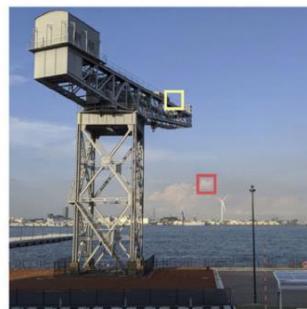


色空間の変換

RGBを輝度Yと色差Cb,Crに変換。

色差は比較的小さい値になるため、平均符号長を短くできる。さらに、人間の色の空間分解能は明るさの空間分解能よりも低いため、色差成分は輝度成分よりも低い周波数成分を割り当てても気づきにくい。

$$\begin{cases} Y = 0.299R' + 0.587G' + 0.114B' \\ C_b = 0.564(B' - Y) = -0.169R' - 0.331G' + 0.500B' \\ C_r = 0.713(R' - Y) = 0.500R' - 0.419G' - 0.081B' \end{cases}$$



2次元DCT

8×8ブロックに分けて行う。これによりブロック境界が不連続になったり(ブロックノイズ)、高周波が失われることで細い線の周辺でもやもや(モスキートノイズ)する。

量子化

2次元DCTで得られるのは浮動小数点の係数。これに対し周波数成分によって異なる値(量子化テーブル)を除算して、整数値のみ取り出す。

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

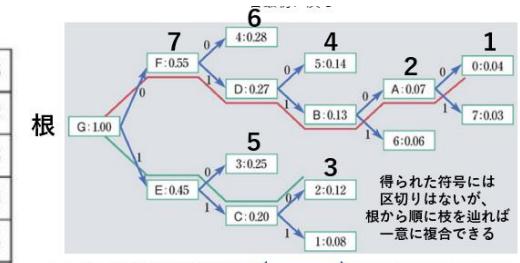
[a] 輝度成分に対する量子化テーブル

[b] 色差成分に対する量子化テーブル

ハフマン符号化

シンボルの並びは、左上の直流成分は隣接ブロックとの差分を始点としそれ以外はジグザグスキヤンの順番。
これにより後半はほとんど0になり、より圧縮率が上がる。

→ 水平方向空間周波数 <i>u</i>							
0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63



動画像の符号化

フレーム間符号化

動画像においてフレーム間で生じた差分のみを符号化

オブジェクト符号化

対象物体と背景を分けて符号化

動き補償符号化

動きが速い対象物体を含む画像間の差分は大きな値になってしまうため、直前のフレームからテンプレートマッチングを用いて対象物体の動き推定を行い、フレームの予測画像を作成し差分をとる符号化

MPEG

基本的には動き補償符号化とDCT符号化の組み合わせで構成されており、DCT符号化はJPEGと同様。これに加えて、隣接フレームはほとんど同じであるという仮定のもと圧縮している。

GOP(Group of Pictures) 符号化された動画像のフレームのならび。「B B I B B P B B P B B P」

Iピクチャ

(Intra coded picture)

- JPEG符号化された画像
- 単独で復号できる。
- ところどころ(0.5sに一枚程度)に含むことで途中からの再生が可能
- 早送りはIピクチャのみを再生している
- データ量は大きい

Pピクチャ

(Predictive coded picture)

- 直前のIピクチャかPピクチャからのマクロブックごとの順方向動き補償符号化フレーム
- データ量はIピクチャの半分程度
- 動きベクトルが求まらない場合はJPEG符号化

Bピクチャ

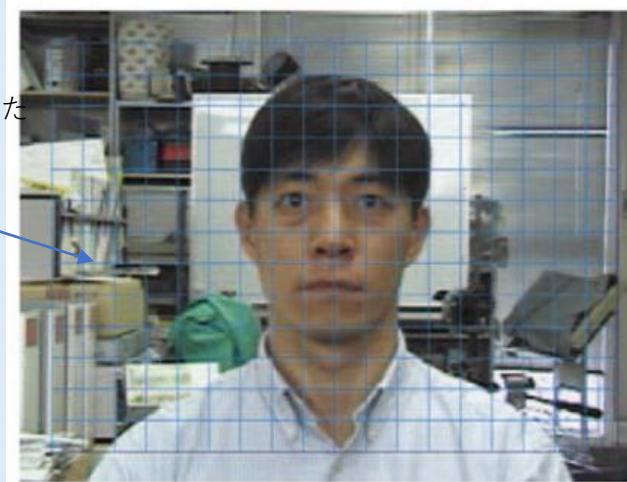
(Bidirectionally-predictive coded picture)

- 直前と直後のIピクチャかPピクチャからのマクロブックごとの動き補償符号化フレーム
- ①順方向 ②逆方向 ③①と②の平均 ④JPEGの内、最もデータ量が低いものを採用
- 他のピクチャの復号に使われないため、予測誤差を粗く量子化でき、データ量はIの1/4

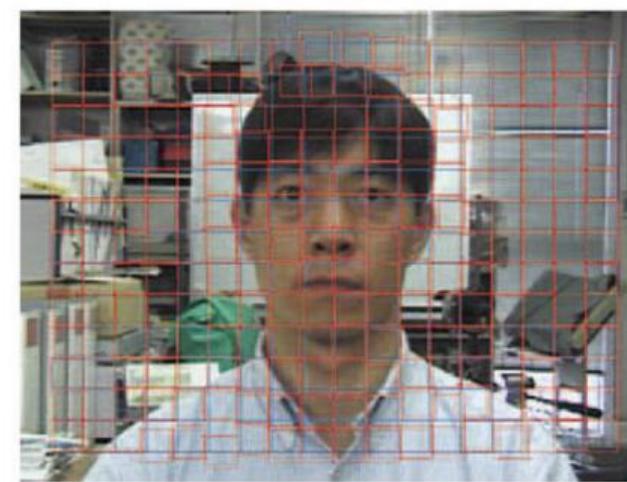
人物が手前に移動

マクロブック

8×8画素のブロックを4個合わせた
16×16画素のブロック



〔a〕Iピクチャの例



〔b〕符号化前のPピクチャにおける推定した動きベクトル

18. 知的財産権

産業財産権・不正競争防止法

知的財産権

人間が知的な創造活動によって生み出した成果に対する権利の総称

保護対象		保護法	権利名	保護期間
著作権 目的は文化の発展	著作物（小説、音楽、舞踊、絵画、建築、地図、映画、写真、プログラムなど） 実演、レコード、放送	著作権法 (詳細は表a.11、表a.12を参照)	著作権 著作隣接権	著作者の死後70年 (法人、映画は公表後70年) 実演、発売後70年、放送後50年
	発明（「物」、「方法」、「物の生産方法」の発明で高度なもの）	特許法	特許権	出願日から20年
産業財産権 目的は産業の発展	考案（物品の形状、構造または組み合わせにかかる考案で高度性は不要）	実用新案法	実用新案権	出願日から10年（無審査）
	意匠（物品のデザイン、画像デザイン）	意匠法	意匠権	出願日から ^{*49} 25年
	商標（トレードマーク、サービスマーク）	商標法	商標権	設定登録日から10年 (10年ごとに更新可能)
営業秘密（ノウハウ、顧客データなど）、著名な商品表示、形態など		不正競争防止法	-	-

産業財産権

産業の発展を目的とした権利。各権利を取得するために特許庁には出願する必要がある。

不正競争防止法

権利を認めて保護する著作権や特許権とは逆で、他人の成果に便乗した不正行為を禁止する法律。

不正行為とは、他のブランド名、商品名を利用して混同させたり、営業秘密を不正に入手するなどがある

産業財産権					著作権	
権利	実用新案	商標	特許	意匠	著作隣接	著作
対象	形状の考案	マーク、ネーミング	物、方法の発明	デザイン	実演、レコード、放送	著作物
期間 [年]	10	10(更新可)	20	25	実演、発売後70年、放送後50年	死後70年 無名、団体、映画は公表後70年
覚え方	実証が得意な隣の長さん					

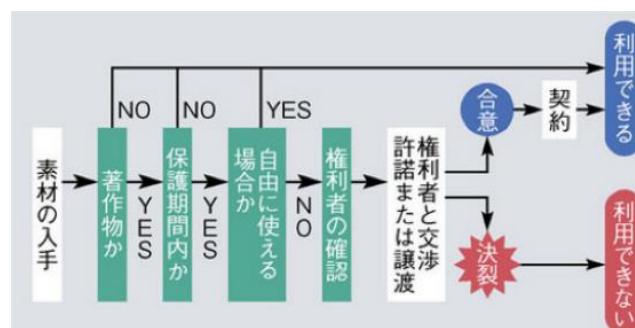
著作権

著作権

著作物を創作した著作者に認められる著作権(著作財産権、著作人格権)と、著作物を公衆に伝えるものに認められる著作隣接権がある。

	著作者の権利(著作物を創作した著作者に認められる権利)		著作権		著作隣接権(著作物を公衆に伝達する者に認められる権利)		実演家人格権
権利の発生(無方式主義)	著作権(著作財産権)	著作者人格権	権利の発生(無方式主義)	権利の性質	権利者	実演を行った時点、音を最初に固定(録音)した時点、放送を行った時点で自動的に発生	著作隣接権
権利の性質	財産的権利、譲渡可	人格的権利、譲渡不可	権利の性質	放送事業者	有線放送事業者	財産的権利 譲渡可	人格的権利 譲渡不可
権利者	著作者(著作権者)	著作者	権利者	放送事業者	有線放送事業者	レコード製作	実演家
権利の内容	複製権 上演権・演奏権 上映権 公衆送信権・伝達権 口述権 展示権 頒布権(映画の著作物のみ) 譲渡権(映画以外の著作物) 貸与権(映画以外の著作物) 翻訳権、翻案権など 二次的著作物の利用に関する権利	公表権 (未公表の自分の著作物を公表するかしないかを決定する権利) 氏名表示権 (自分の著作物を公表するときに名前を表示するかしないか、表示する場合は実名か変名かを決定する権利) 同一性保持権 (著作物の性質ならびにその利用の目的および態様に照らしてやむを得ないと認められる場合などを除き、自分の著作物の内容、題号を自分の意に反して勝手に改変されない権利)	権利の内容	複製権 再放送権 有線放送権 ・テレビジョン放送の伝達権 ・送信可能化権	複製権 ・複製権 ・放送権、 再有線放送権 ・有線テレビジョン放送の伝達権 ・送信可能化権	複製権 ・複製権 ・譲渡権 ^{*51} ・貸与権(レコード 発売後1年間のみ) ・送信可能化権	・録音権、録画権 ・放送権、 有線放送権 ・譲渡権 ・貸与権(レコード 発売後1年間のみ) ・送信可能化権
保護期間	一般著作物は原則として、創作のときから著作者の死後70年間(詳細は、下記*50を参照のこと)	著作者の生存中 (著作者の死後も著作者人格権の侵害となるべき行為をしてはならない)	保護期間	放送後50年	有線放送後50年	(報酬請求権) ・商業用レコードの二次使用料を受ける権利 ・貸レコードについて報酬を受ける権利 (貸与権消滅後69年間)	実演家の生存中 (実演家の死後も実演家人格権の侵害となるべき行為をしてはならない)
著作物の保護期間	・実名(周知の変名を含む)の著作物:著作者の死後70年(原則的保護期間) ・無名・変名の著作物:公表後70年(死後70年の経過が明らかであれば、その時点まで) ・団体名義の著作物:公表後70年(創作後70年以内に公表されなければ、創作後70年) ・映画の著作物:公表後70年(創作後70年以内に公表されなければ、創作後70年)						

利用のルール



自由に使える場合

- ・私的利用
- ・写りこみ
- ・検討過程
- ・非享受利用
- ・引用
- ・営利目的じゃない上演
- ・公開著作物
- ・オークションの商品紹介用
- ・新たな知見を創出する情報処理

著作権侵害にあたらない場合

- ・アイディアやありふれた表現
- ・偶然の一一致

著作物

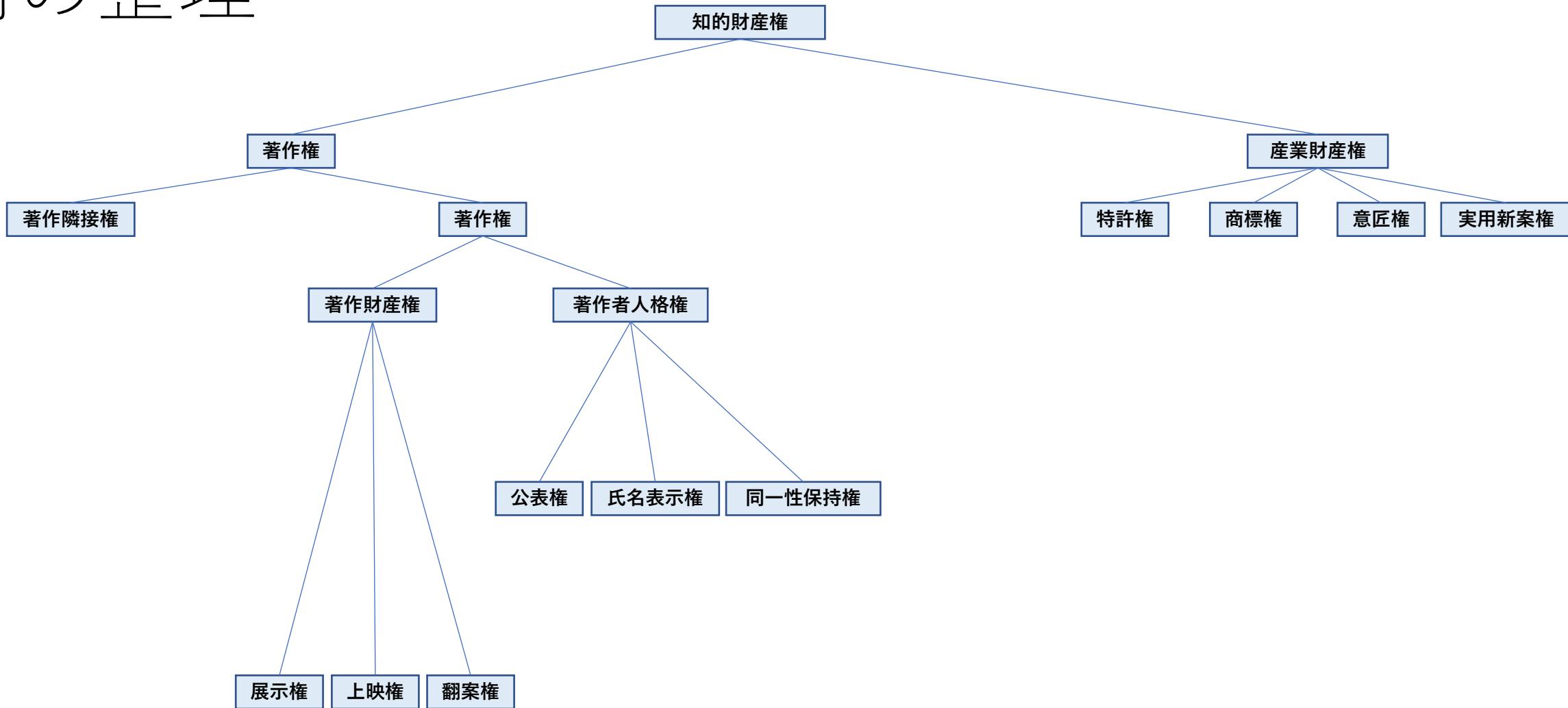
思想又は感情を創作的かつ具体的に外部に表現されたものであって、文芸、学術、美術又は音楽の範囲に属するもの。どんな画像や動画であっても著作物になりうる。

CG、コンピュータプログラム、データベースも著作物。

ただし、アイディア、コンセプトなど外部に表現されていないものは著作物の対象じゃない！

逆に表現されているものは未発表でも著作物！

権利の整理



著作物の利用

原則NG

著作財産権

1. 複製
2. 上演
3. 送信
4. 口述
5. 展示
6. 頒布
7. 譲渡
8. 貸与
9. 翻訳
10. 翻案

著作人格権

1. 公表
2. 氏名表示
3. 同一性保持(加工されない権利)

権利者に許諾を得ればOK

自由に使える場合

1. 家庭内用の複製
2. 写りこみ
3. 利用検討段階
4. 非享受利用(学習用データなど)
5. 引用
6. 非営利上演
7. 公開美術品
8. オークション用
9. 新たな知見創出のためのパソコンでの軽微な利用

著作者・著作権者

著作者 = 著作物を創作したもの

著作権者 = 著作権をもつもの

著作人格権については、
権利の譲渡ができないため、
著作権者 = 著作者

著作財産権については、
著作権者 = 著作者 + 権利譲渡されたもの

共同著作物

二人以上で創作したもの
全員が著作者兼著作権者になり、
著作権の行使は全員の合意が必要

職務著作物

会社の命により、会社の職務上創作したもの
著作者および著作権者は法人

CG・プログラム・データベース

CG

ディスプレイモニタ上の表現と生成用プログラムを別個にとらえて、それぞれ著作物の定義に該当する場合は著作物となる

データベース

以下の条件を満たすデータベースは著作物

1. コンピューターで検索できる
2. 創作性を有する

プログラム

1. 言語、規約、アルゴリズムは保護されない
2. 法人名義で公表されてなくても職務著作となる
3. デバッグ、バージョンアップは同一性保持権の適用外
4. 必要な範囲内で複製できる

©マーク

ベルヌ条約…国境を越えて著作権を保護するための条約。日本を含む多くの国が加盟済。

無方式主義…著作権は、著作物の創作と同時に発生するため、商標権のような登録手続きは不要という主義。ベルヌ条約は無方式主義
方式主義…著作権の享受に何らかの方式を必要とする

万国著作権条約…「方式主義」を採用する国においても、「無方式主義」を採用するベルヌ条約加盟国の著作権を保護するための条約。日本も加盟。

©制度

「方式主義」を採用している万国著作権条約の締結国において、
「無方式主義」の国の著作物が保護を受けるための条件として、

複製物に「最初の発行時から」「著作者の名」「最初の発行年」とともに「©」表示することを要求する制度。

現在、「方式主義」を採用する国はごく少数であるため、
「©」を表示する必要性は乏しい

ただし、現在でも、著作権を主張する者が、自分が著作者であることを示すために「©」を用いている。

表示例 ©2020 CG-ARTS All rights reserved.
©CG-ARTS 2020 All rights reserved.