

## 12回 パターン検出と画像特徴

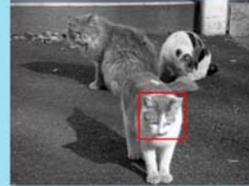
- テンプレートマッチング
- 領域分割
- 画像特徴

# テンプレート マッチング

## テンプレートマッチング

(図形・画像などの) 型      照合

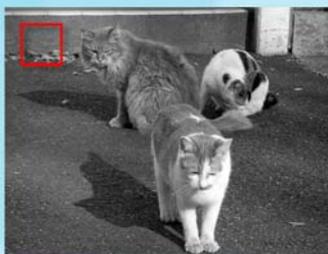
### ● Template Matching



- テンプレートと呼ばれる小さな一部の画像領域と同じパターンが画像全体の中に存在するかどうかを調べる方法
- 画像内にある対象物体の位置検出、物体数のカウント、物体移動の検出などに使われる

## テンプレートマッチングの計算

- テンプレートを画像の中で順番に移動させながら、テンプレートとテンプレートに重なる部分の画像の類似度を計算していく



画像



テンプレート

### 類似度計算

- SSD (Sum of Squared Difference)
  - 画素値の差の2乗和
  - ユークリッド距離の2乗
  - (ユークリッド距離の2乗和)
- SAD (Sum of Absolute Difference)
  - 画素値の差の絶対値の和
  - 市街地距離
  - (市街地距離の和) **高速**

## 類似度の計算

### ● SSD (Sum of Squared Difference)

$$SSD = \sum_{x=0}^{X-1} \sum_{y=0}^{Y-1} (I(x, y) - T(x, y))^2$$

- 画像の輝度値を  $I(x, y)$
- テンプレート画像の輝度値を  $T(x, y)$

### ● SAD (Sum of Absolute Difference)

$$SAD = \sum_{x=0}^{X-1} \sum_{y=0}^{Y-1} |I(x, y) - T(x, y)|$$

- ◆ 値が小さい程、テンプレートとテンプレートに重なる部分の画像が似ている
- ◆ 完全に一致している場合には、値は0になる

## テンプレートマッチングの計算

194	183	173	173	176	185	189	176
191	163	142	148	160	169	172	173
191	167	114	108	146	153	158	158
199	183	145	96	105	125	132	140
203	193	172	119	80	61	77	109
197	196	181	153	107	105	71	80
186	185	183	170	132	88	77	75

画像

180	145	90
190	170	120
195	180	150

テンプレート

### 類似度の値 (SSD, SAD)

- 計算結果が0となる場所を発見  
→ テンプレートと**同じパターン**が検出
- 計算結果が0に近い小さな値  
→ テンプレートと**類似のパターン**を検出

## テンプレートマッチング (SSD)

194	183	173	173	176	185	189	176
191	163	142	148	160	169	172	173
191	167	114	108	146	153	158	158
199	183	145	96	105	125	132	140
203	193	172	119	80	61	77	109
197	196	181	153	107	105	71	80
186	185	183	170	132	88	77	75

画像

180	145	90
190	170	120
195	180	150

テンプレート

$$SSD = \sum_{x=0}^{X-1} \sum_{y=0}^{Y-1} (I(x, y) - T(x, y))^2$$

SSD =

$$\begin{aligned} & (163 - 180)^2 + (142 - 145)^2 + (148 - 90)^2 \\ & + (167 - 190)^2 + (114 - 170)^2 + (108 - 120)^2 \\ & + (183 - 195)^2 + (145 - 180)^2 + (96 - 150)^2 \\ & = 11756 \end{aligned}$$

- SSDは値が大きくなる

## テンプレートマッチング (SAD)

194	183	173	173	176	185	189	176
191	163	142	148	160	169	172	173
191	167	114	108	146	153	158	158
199	183	145	96	105	125	132	140
203	193	172	119	80	61	77	109
197	196	181	153	107	105	71	80
186	185	183	170	132	88	77	75

画像

180	145	90
190	170	120
195	180	150

テンプレート

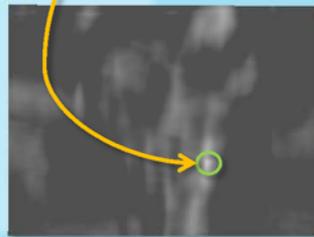
$$SAD = \sum_{x=0}^{X-1} \sum_{y=0}^{Y-1} |I(x, y) - T(x, y)|$$

$$\begin{aligned} SAD &= |163-180| + |142-145| + |148-90| \\ & + |167-190| + |114-170| + |108-120| \\ & + |183-195| + |145-180| + |96-150| \\ &= |-17| + |-3| + |58| \\ & + |-23| + |-56| + |-12| \\ & + |-12| + |-35| + |-54| \\ &= 17+3+58+23+56+12+12+35+54 \\ &= 270 \end{aligned}$$

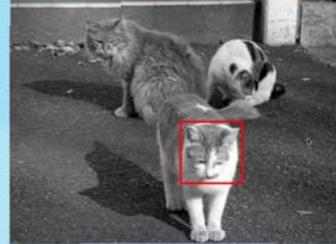
## テンプレートマッチングと類似度



画像



類似度を示す画像



検出結果



テンプレート

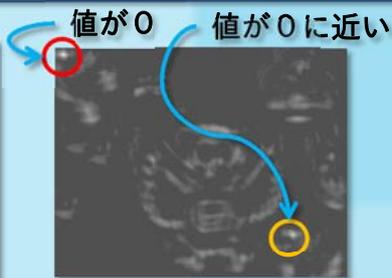
1. テンプレートを移動しながら類似度を計算
2. 出力画像へ類似度を書き込む
3. 出力画像を正規化する (階調を補正)

類似度が最も高い場所

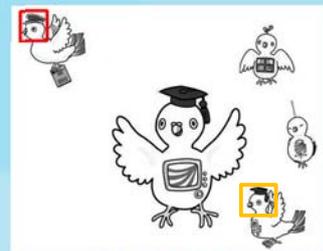
## テンプレートマッチングの例1



画像



類似度を示す画像



検出結果



テンプレート

- 類似度順に候補を  
列挙できる



第1候補



第2候補

帽子が違う

## テンプレートマッチングの例2



- 単純なテンプレートマッチングでは、  
拡大・縮小、回転した対象は検出が難しい

### 解決法

- ✓ 複数の拡大縮小を行ったテンプレートを用意
- ✓ 拡大縮小や回転の影響を受けにくい画像特徴に変換

## テンプレートマッチングの高速化

- テンプレートや探索対象となる画像のサイズが大きくなると、  
類似度の計算が爆発的に増加

→ 高速にテンプレート画像を検出することが困難

### 解決法

- ◆ 画像ピラミッドを利用した粗密探索 (Coarse-to-Fine Search)  
様々な解像度のテンプレートと探索対象画像を用意し、  
粗い画像から密度の高い詳細画像に探索を行う
- ◆ 計算の打ち切り  
類似度計算の過程で、ある条件を満たしたら、  
類似度計算を終了し、次の探索を進める方法
- ◆ 計算の並列化やハードウェア化  
マルチコアCPU (中央演算処理装置) の利用  
GPGPU (General-purpose computing on graphics processing units) の利用

# 領域分割

## 領域分割 (Region Segmentation)

- 画像を特徴の一様な部分領域に分割し、対象物の占める領域を見つける方法

### 応用分野

- 医療画像解析
- 衛星画像解析
- 顔認識
- 交通監視システム
- 指紋認識

## 領域分割 (Region Segmentation)



原画像



領域分割後

- 色・輝度
- 形状
- テクスチャ

■ 花の部分

■ 花以外の部分

- 画像内容を解析し認識するためには、対象物を抽出しなければならない  
→ **領域分割**は画像認識において重要な処理の一つ

## カラー画像の減色

### カラー画像の減色

- 画像の階調数を減らす処理
- 画像データサイズを減らす効果などがある

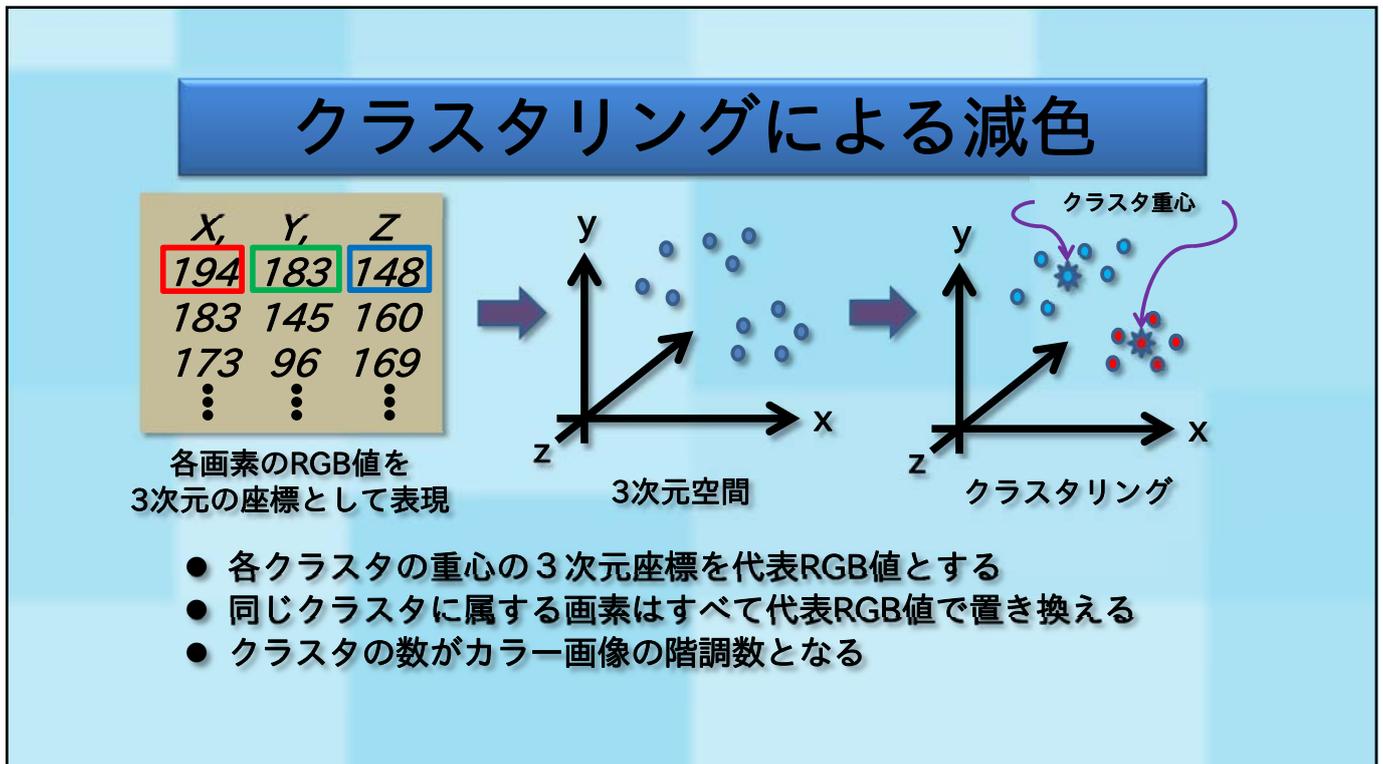
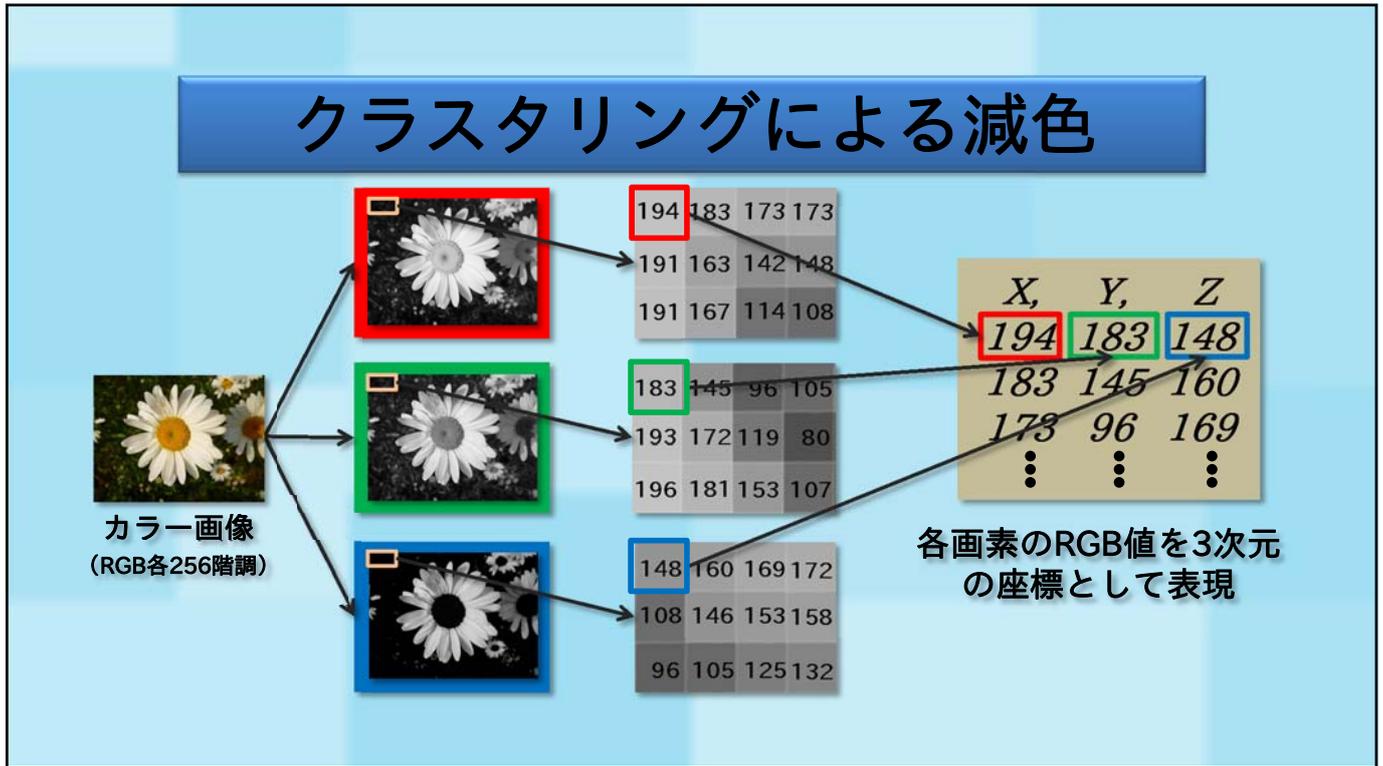


原画像  
(RGB各256階調)



減色後の画像  
(4階調)

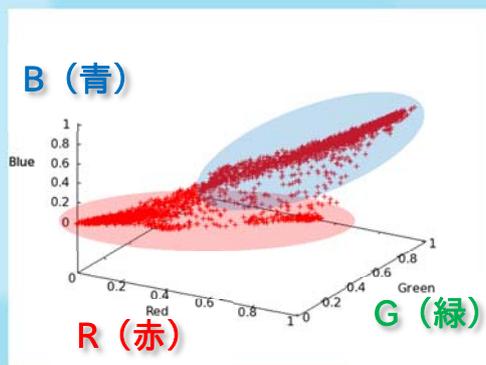
- k平均法による減色は、色による画像の領域分割



## クラスタリングによる減色



画像サイズ : 80×60  
点の数 : 4800



各画素のRGB値を3次元の座標として表現

## クラスタリングによる減色 (例1)



原画像



2階調



4階調



8階調



16階調

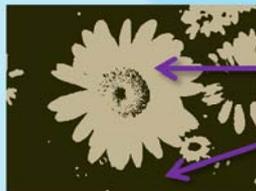


32階調

## カラー画像の減色による領域分割



原画像



2階調

C1 ■ 194, 181, 92

C2 ■ 27, 27, 5

## クラスタリングによる減色 (例2)



原画像



2階調



4階調

C1 ■ 246, 244, 209

C2 ■ 203, 190, 81

C3 ■ 92, 94, 52

C4 ■ 90, 63, 24



8階調



16階調



32階調

## 様々な領域分割アルゴリズム

- 画像ピラミッドを用いた画像領域分割
- 平均値シフト法による画像領域分割
- Watershedアルゴリズムによる画像領域分割

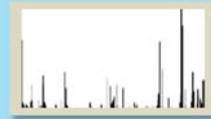
画像特徴

## 画像特徴



画像

特徴抽出  
(Feature Extraction)



画像特徴

- 色
- 形状
- テクスチャ

- 画像特徴量 → 画像の特徴や性質を表す数値
- パターン検出、画像分類、類似検索、画像改善などへ応用

## 広域特徴と局所特徴



- 広域特徴 (Global Features)
  - ◆ 一つの画像全体から計算される



- 局所特徴 (Local Features)
  - ◆ 画像の一部分から計算される
  - ◆ 画像の一部である部品
    - パッチ (Patch、小さな領域) とよばれる

# 局所特徴



格子

- 画像を格子状に区切って分割し、分割した各領域をパッチとする



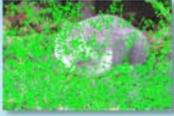
ランダム

- 乱数を発生。乱数を画像内の点の位置情報に変換し、各点の周辺領域をパッチとする



領域

- 画像を、色、テクスチャなど、何らかの特性に基づいて領域分割し、各領域をパッチとする



注目点

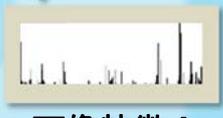
- 画像内のエッジやコーナーなど画像の目立った場所などを画像内から探し、パッチとする

# 画像特徴の不変性 (Invariance)



↓ 画像 A

画像特徴計算



画像特徴 A

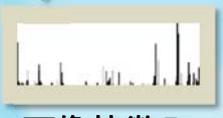


鳥の大きさが異なる



↓ 画像 B

画像特徴計算



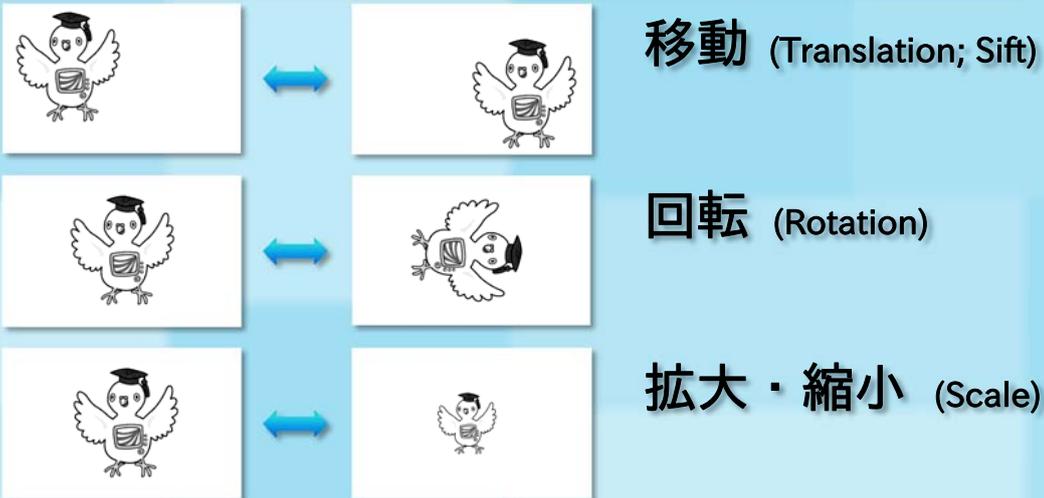
画像特徴 B



画像特徴が同じ!

画像の拡大・縮小  
に対し不変な特徴  
(Scale Invariant)

## 画像特徴の不変性 (Invariance)



## SIFT特徴

拡大縮小 不変 特徴 変換

- SIFT (Scale-Invariant Feature Transform)
- 1999年にDavid Loweが提案した手法
- 局所特徴の検出と記述を行うアルゴリズム
- 画像の拡大・縮小、回転、照明変化などの影響を受けにくい特徴
- SIFT特徴の計算を行うソフトウェアが多数存在

## SIFT特徴の例

- SIFT特徴は一つの画像から、複数のキーポイント (特徴点) と呼ばれる位置で計算
- 各キーポイントから128次元の特徴



画像



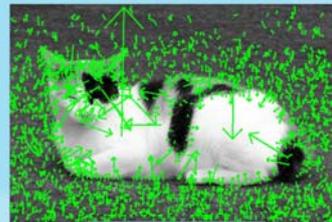
SIFT特徴

- 矢印の始点はキーポイントの位置
- 矢印の方向は特徴の方向
- 矢印の長さは特徴の大きさ

## SIFT特徴の例



画像

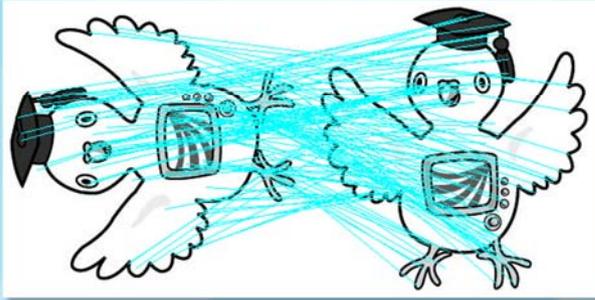


SIFT特徴

- SIFT特徴の計算には、DoG (Difference of Gaussian)、画像の微分、ヒストグラム計算など、複数の画像処理アルゴリズムが使われる
- キーポイントは画像の特徴を代表する部分
- キーポイントの数は画像の内容によって異なる

## SIFT特徴による画像の対応付け

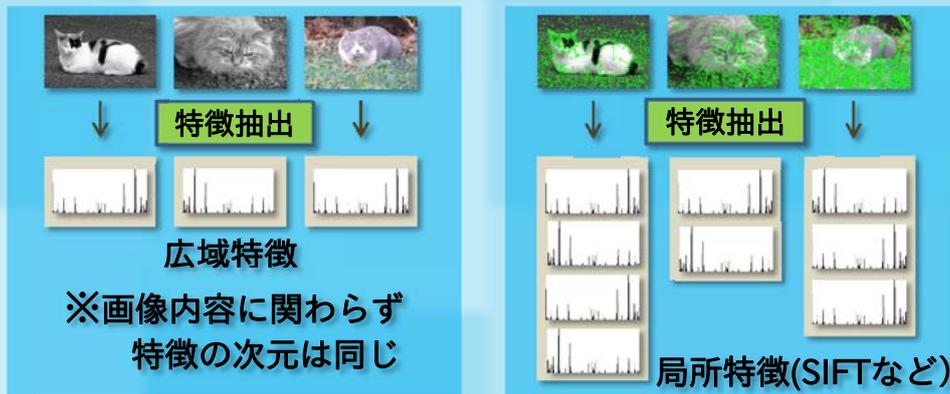
- SIFT特徴の類似度を計算することによって、画像間で対応付け（マッチング）を行うことができる。



1. 同じ画像を90度回転
2. 画像からSIFT特徴を計算
3. 2枚の画像のキーポイントを比較
4. 類似しているキーポイントを直線で結ぶ

回転に強い：回転しても同じキーポイントを計算、画像の対応付けを実現

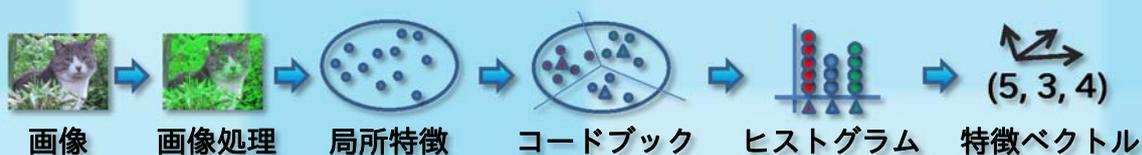
## 画像特徴の次元



- SIFT特徴の数は画像内容によって異なる ⇒ 特徴の次元が異なる
- 特徴の次元が異なると比較が困難 ⇒ B o F モデルの利用

## BoF (Bag of Features)

- BoF (バッグ オブ フィーチャーズ)
- 幾つかの呼び方がある
  - Bag of Words (BoW)
  - Bag of Visual Words (BoVW)
  - Bag of Key points (BoK)
- 自然言語処理分野で発達した手法
- コードワードと呼ばれる幾つかの局所特徴を代表するような特徴を作成
- コードワードの出現頻度からなるヒストグラムで表現



## まとめ (第12回 パターン検出と画像特徴)

- テンプレートマッチング
  - SSD、SAD
- 画像からの特徴抽出
  - SIFT、SURF
- BoFモデル
  - BoF、コードワード、コードブック、K-Means