



2. 画像データベース

(マルチメディアデータベース序論, 全6回)

<https://www.kkaneko.jp/de/multimedadb/index.html>

金子邦彦



今日のトピックス



- 画像データの特徴量の例

- 色ヒストグラム
- テクスチャ
- など

} 画素値から取り出される
特徴的な情報

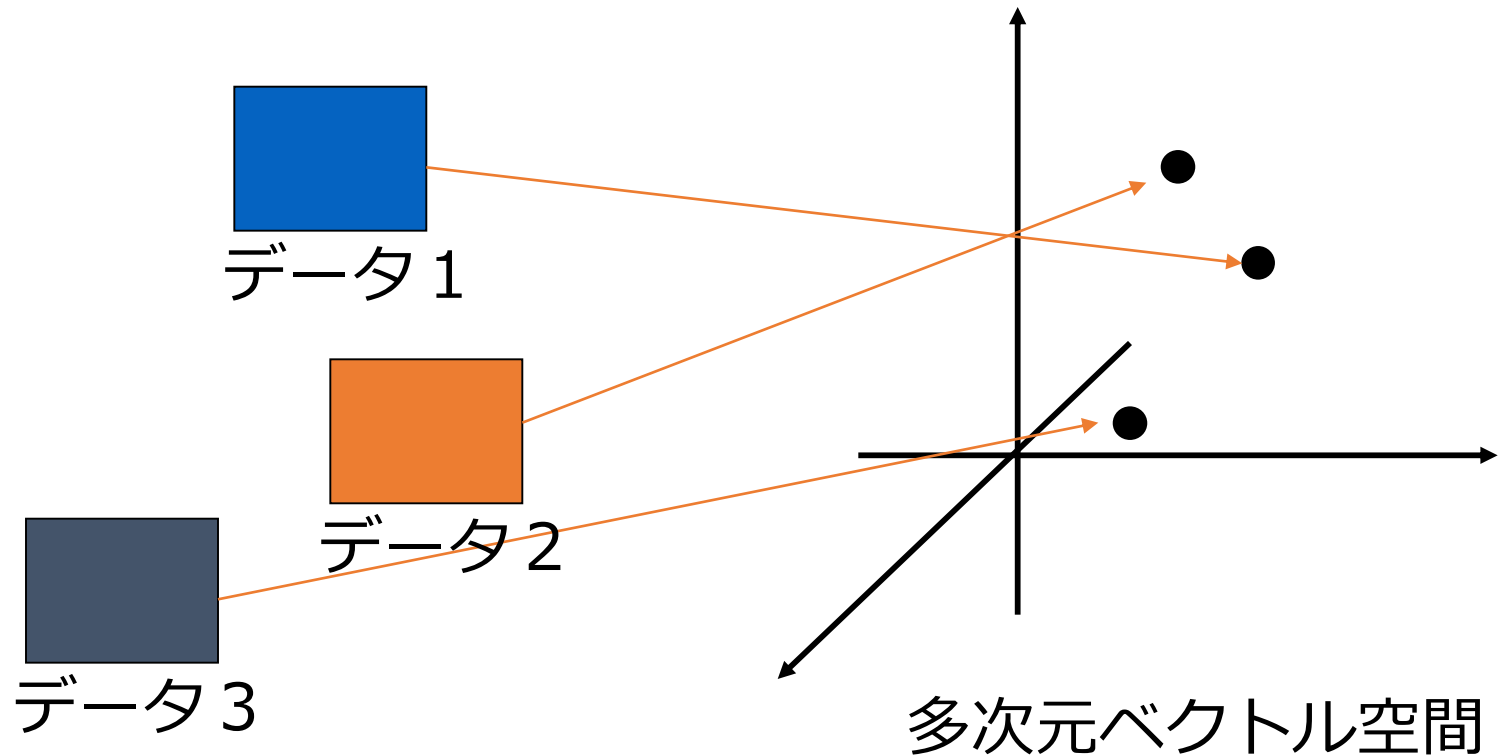
- 画像データ内容検索のための技術の例

- 画像を $(m \times n)$ 個の「セル」に分割し、セル単位で内容情報（色ヒストグラム等）を取り出す
- ものの形を近似する

特徴量



- 特徴量は、「多次元のベクトル」データ



特徴量の課題



- 「内容検索」が出来るだけの十分な情報を含むこと
 - 多次元ベクトルの長さが、十分に長い
 - 特徴量の取り出し方が十分に吟味されている
- データの類似性が考慮されていること
 - 似ているデータは、多次元ベクトル空間の「近く」のベクトルに写像される

画像データベースの内容検索



- 問い合わせの入力として、画像が与えられる
- ユーザは、似ている画像を得たい
- システムは、入力画像とデータベース内の画像とのマッチングを行う



- 画像データベースでの内容検索
 - ユーザの手元に、1枚の写真があって、「この写真に写っている人に似ている写真」を、データベースの中から探す

画像の性質 (1 / 2)

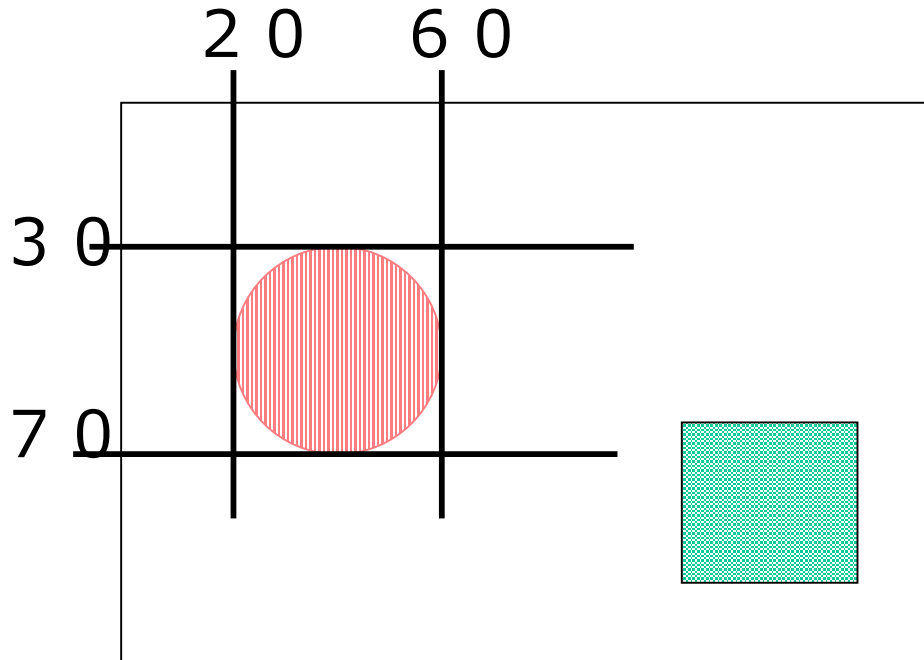


- 画像には、「もの」が写っている
 - 形
 - 場所
 - 色、明るさ
 - 色や明るさは、「画素」が持つ性質
 - しかし、「画素」は、直接扱うのは手間がかかる
 - 「ものが写っている長方形領域（セル）」を単位として扱うと、扱いやすい

画像の性質 (2 / 2)



- 領域 : $XLB=20, XUB=60, YLB=30, YUB=70$
- 色 : $RED=5, GREEN=3, BLUE=1$



画像の内容情報



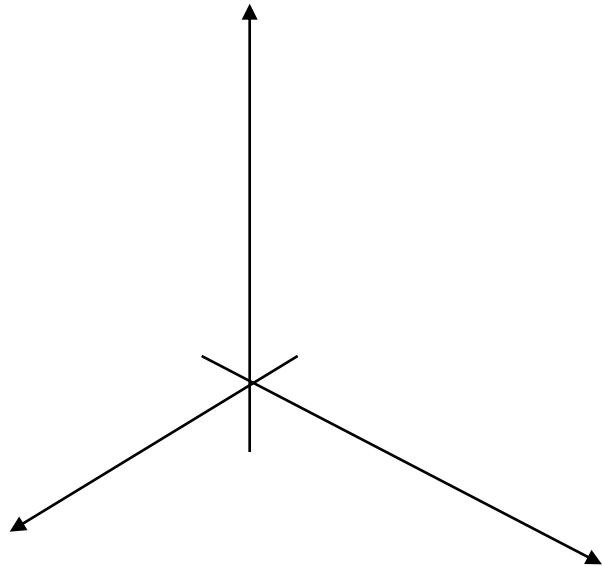
- 色 (color)
- テクスチャ (texture)
- 形 (shape)
- エッジ (edge)

色情報

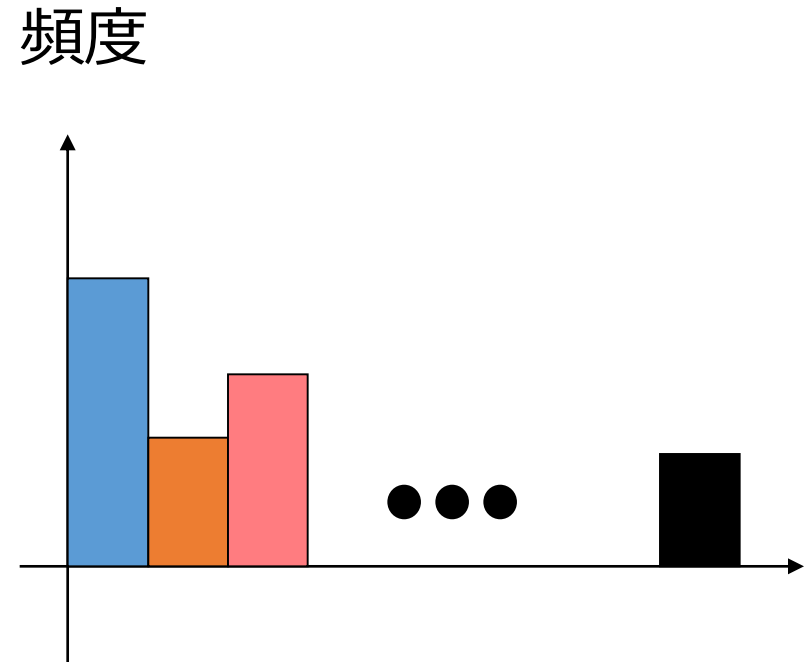
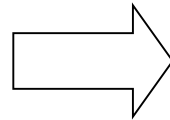


- 明るさの平均 (mean brightness)
- 色ヒストグラム (color histogram)
- 色のばらつき具合 (statistical moments)
- 主に登場する色の種類 (dominant colors)

色ヒストグラム



色空間

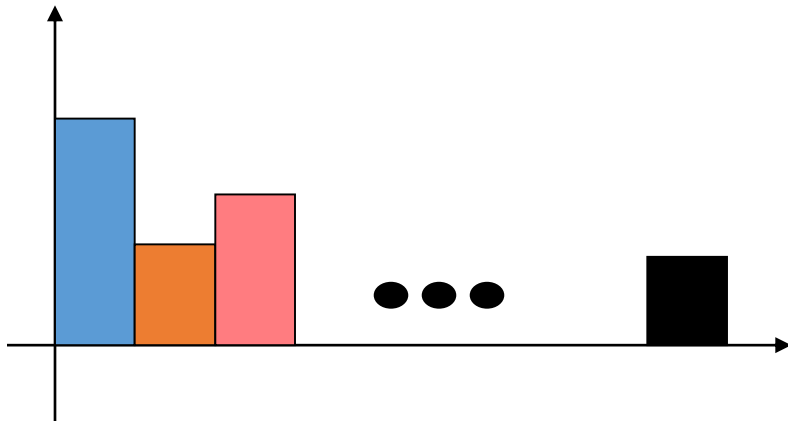


数十個程度の
クラスタ

色ヒストグラムの類似度

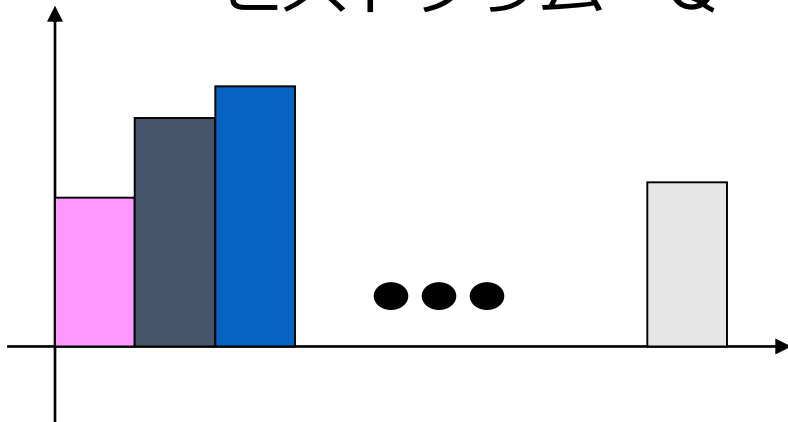


ヒストグラム I



$$\sum_{i=1}^N \sum_{j=1}^N a_{ij} (l_i - Q_i) (l_j - Q_j)$$

ヒストグラム Q



a_{ij} : Iの*i*番目の色と,
Qの*j*番目の色の類
似度

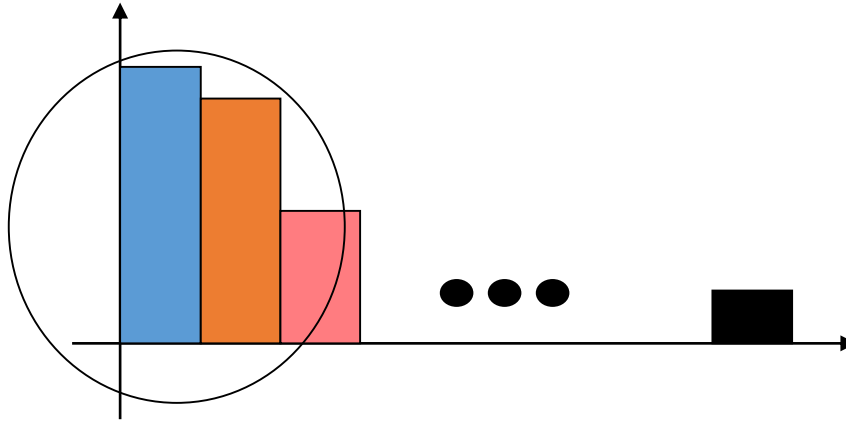
$$0 < a_{ij} < 1$$

l_i, Q_i : 頻度

主に登場する色の種類



- ヒストグラムを使って求める



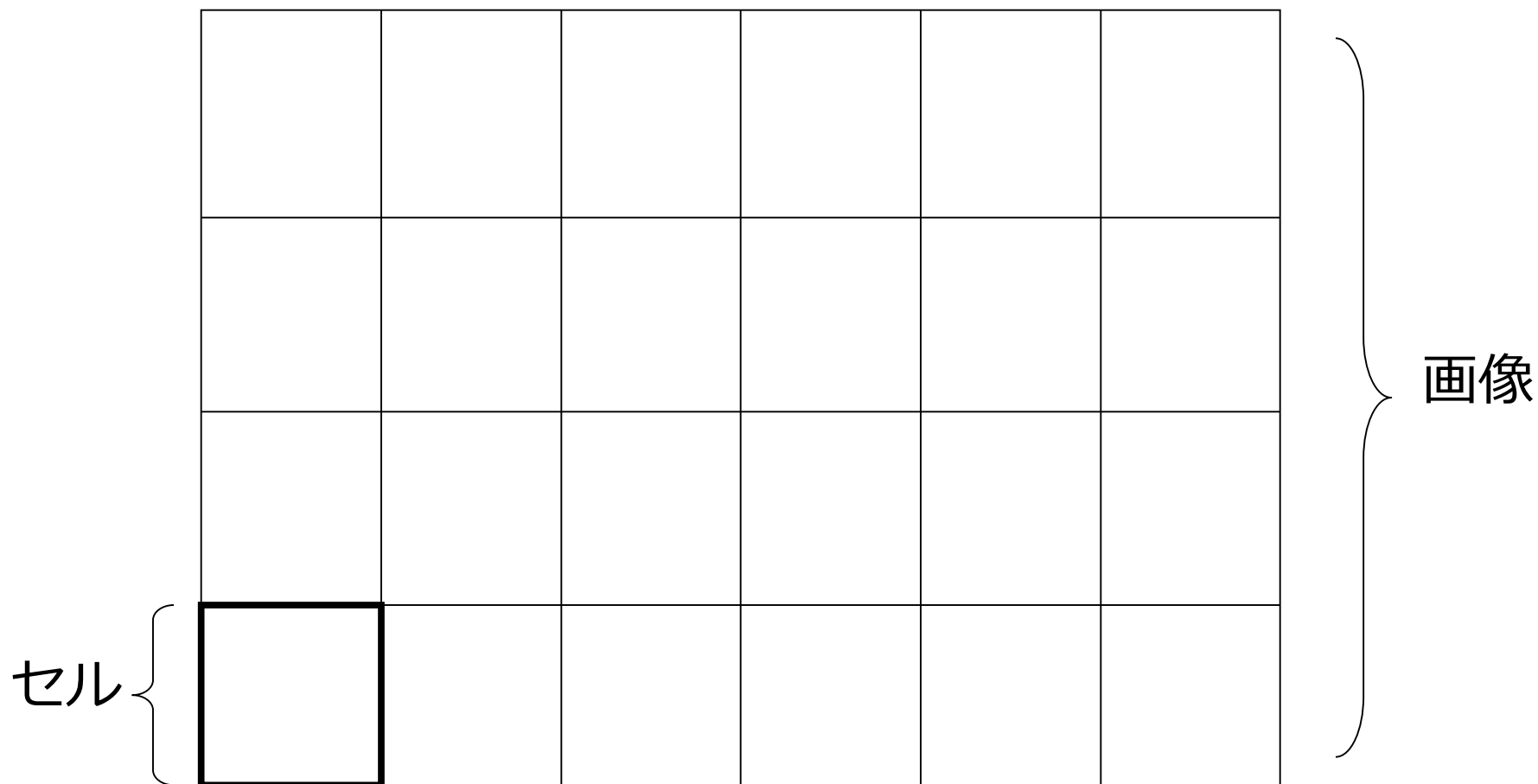
- 一般に、おおむね数個の色で、画像の大部分を埋めつくすことができる
- 「あまり登場しない色」を無視する

Tamura feature



- Contrast
 - 画素の明るさの分散
- Directionality
 - 画素のグラディエントの向き
- Coarseness
 - テクスチャの粗さ.

画像のセル分割



セルの扱い

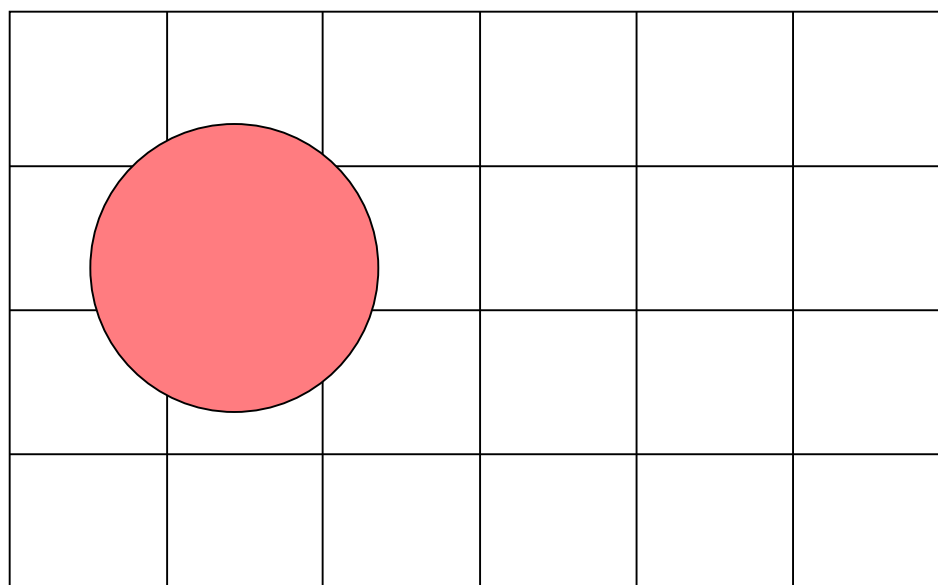


- 画像を $(m \times n)$ 個のセルに分割
 - セルは、すべて同じ大きさ
 - セル分割した画像を image grid という
- 各セルには、属性付けを行う
 - 2値画像の属性： $(\{b,w\}, \text{bwalgo})$
 - 画素は $\{b,w\}$ の値をとる
 - プロシージャ bwalgo は、セル番号を入力として、 $\{b,w\}$ の割合を返すようなプロシージャ
 - 濃淡画像の属性： $([0,1], \text{grayalgo})$



画像のセル分割での 形と場所の扱い

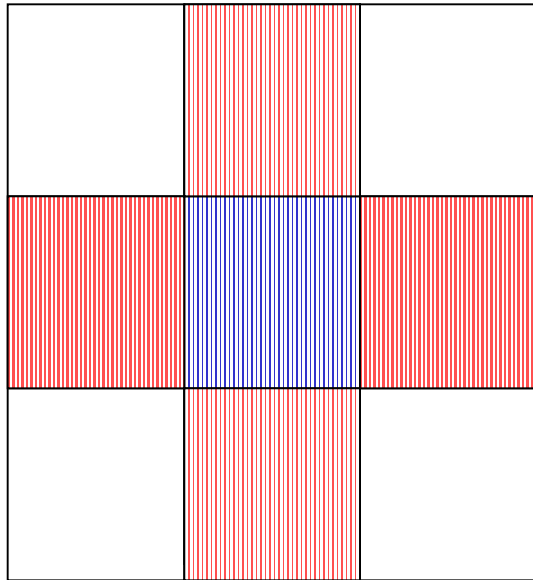
- 場所： セル番号（の集まり）
- 形： セルの集まり方



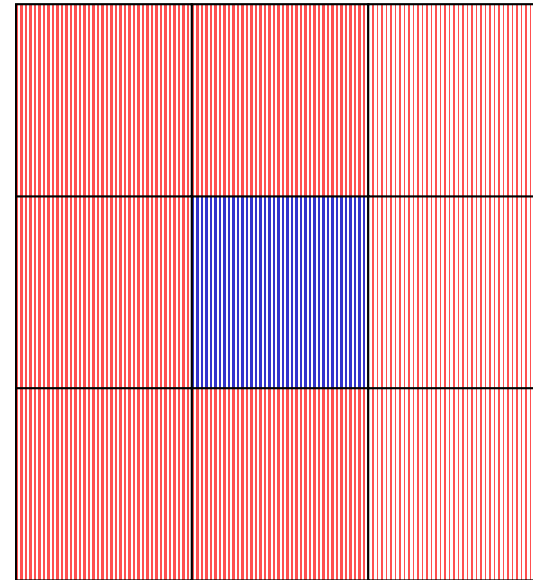
セルの隣接



- 4 隣接



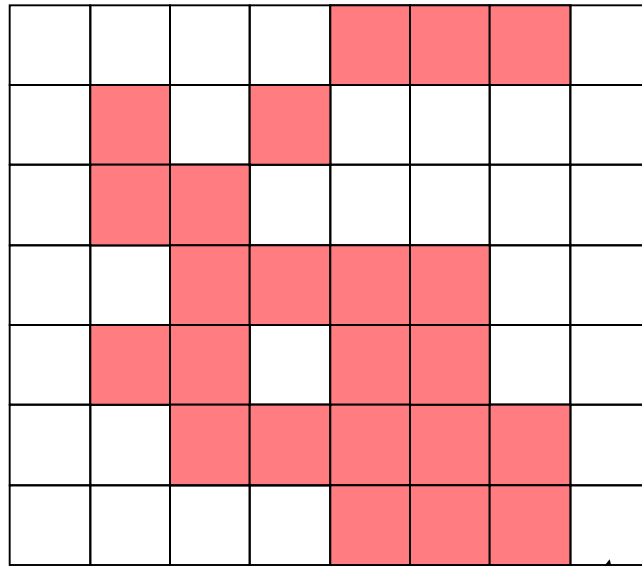
- 8 隣接



形



- ものの形 ⇒ 画素の集まり



画素

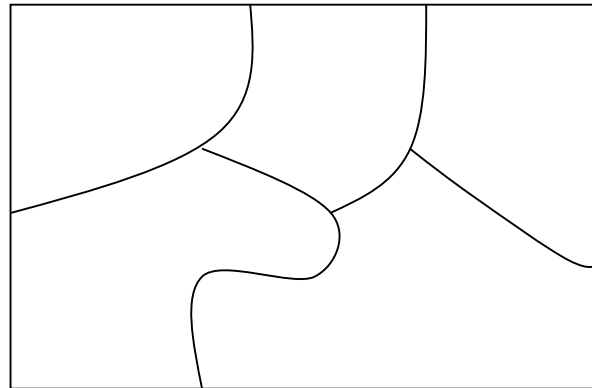
セグメント分割



- 「画像に写っているもの」を，人手でコンピュータに教えるのは，しばしば困難.
- 画像処理（セグメント分割）を使って，「画像に写っているもの」の「形」の候補を取り出すことは可能

セグメント分割 (1 / 2)

- 画像を、互いに素ないくつかの部分に分割



セグメント分割 (2 / 2)



0. 1 5	0. 2 5	0. 5 5	0. 5 5
0. 0 5	0. 3 0	0. 6 5	0. 6 5
0. 3 5	0. 3 0	0. 5 5	0. 8 5
0. 6 5	0. 6 3	0. 8 5	0. 9 0

セグメントの意味



- ある「規則」でセグメントを作る
「平均値からの誤差 0.05 以内」など
- $R_i \cap R_j = \emptyset$ (i, j は異なる)
- $I = R_1 \cup \dots \cup R_k$

split アルゴリズム



- 画像全体から始める

$$R1 = I$$

- それぞれの領域 R_i が「規則」を満たしているか？
 - 満たしていれば終了
 - 満たしていなければ R_i を分割
- 全ての領域が、規則を満たすようになるまで続ける

merge アルゴリズム



- split が終了したら、 R_i と R_j がマージできないか、判定する
 - マージしても規則が満たされるならマージする



- 画像データの単位

- 画素
- セル
- セグメント
- フレーム（1枚の画像）

- これら単位ごとに「特徴量」を取り出し、内容検索を行う

ベクトルデータのオペレーション



- 特徴量は, 一般に多次元のベクトル
- 多次元ベクトルのオペレーション
 - 基準値との比較
 - 区間との比較
 - 距離
 - など
- 広がりを持ったベクトルデータのオペレーション
 - 基準値との比較
 - 区間との比較
 - など



- 演算子

意味

<

左辺が右辺より小さい

<=

左辺が右辺以下

>

左辺が右辺より大きい

>=

左辺が右辺以上

==

左辺が右辺と等しい

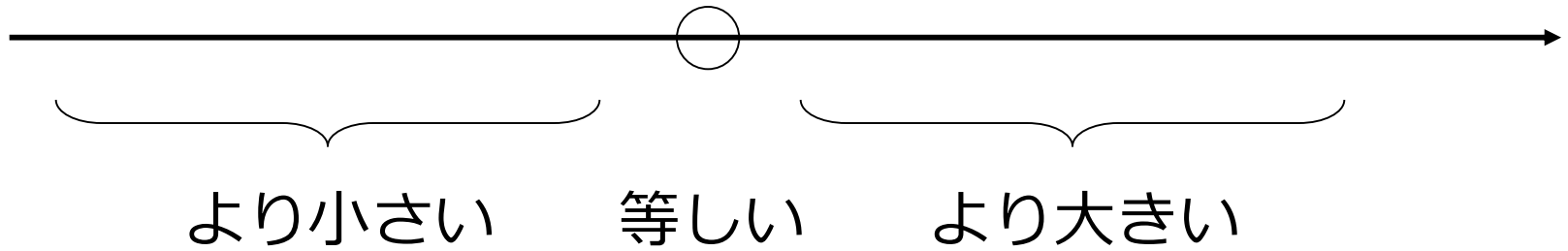
!=

左辺が右辺と等しくない

比較可能な値の演算



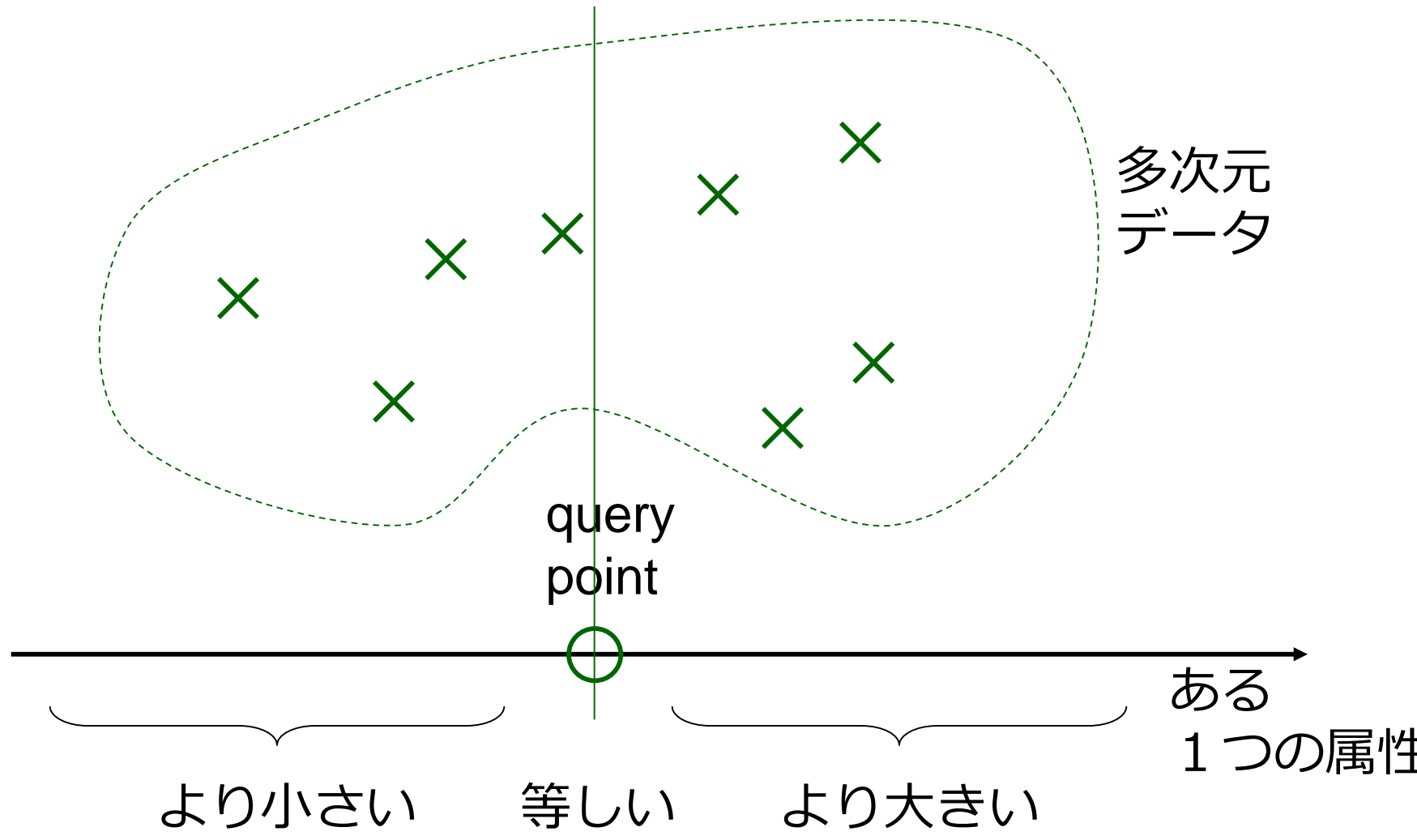
数値, 文字, 文字列は比較可能



	より小さい	等しい	より大きい
<	×		
<=	×	×	
>			×
>=		×	×
=		×	
!=	×		×

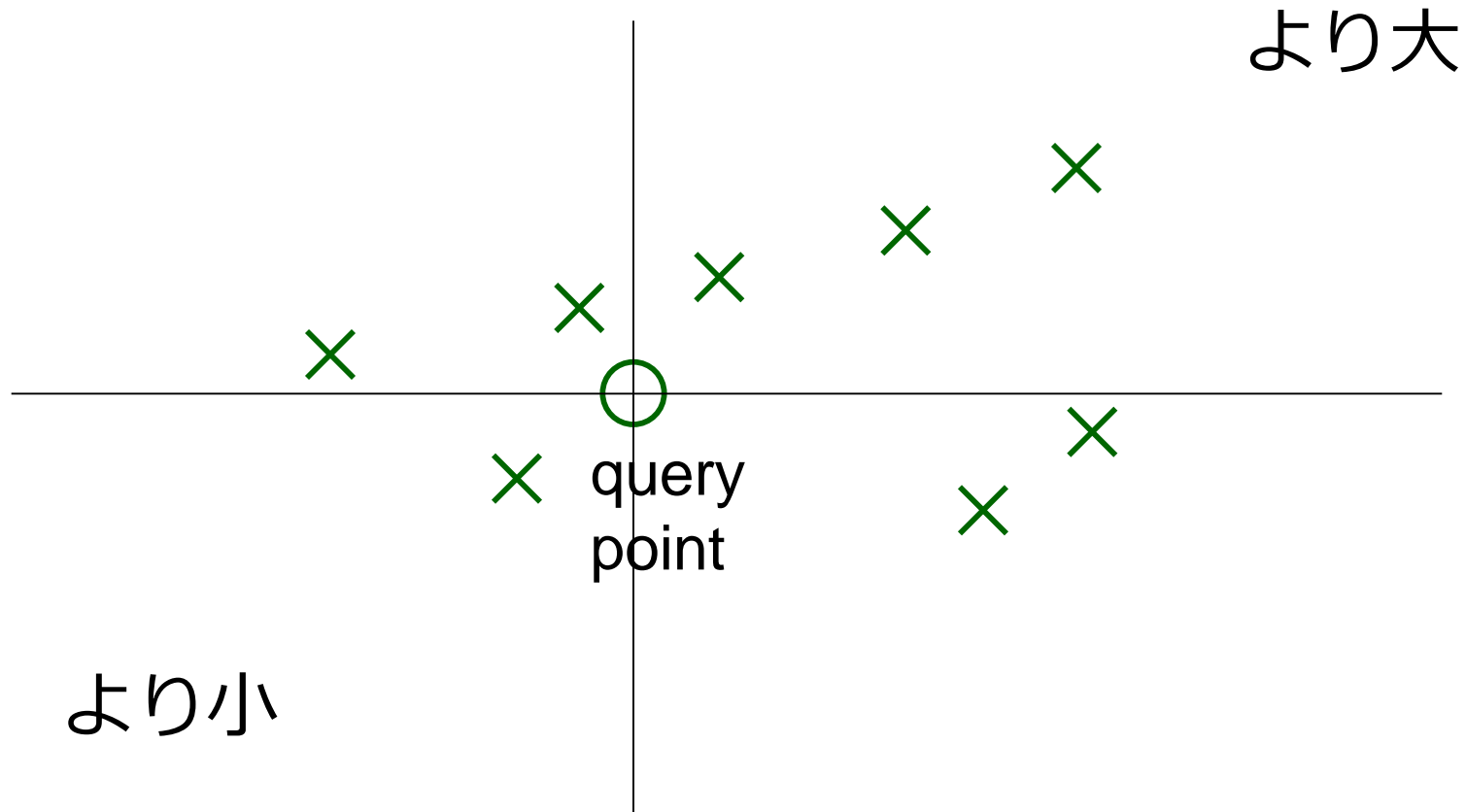


- 「多次元の値」について、ある軸に投影して、比較を行う
 - query point はスカラー値





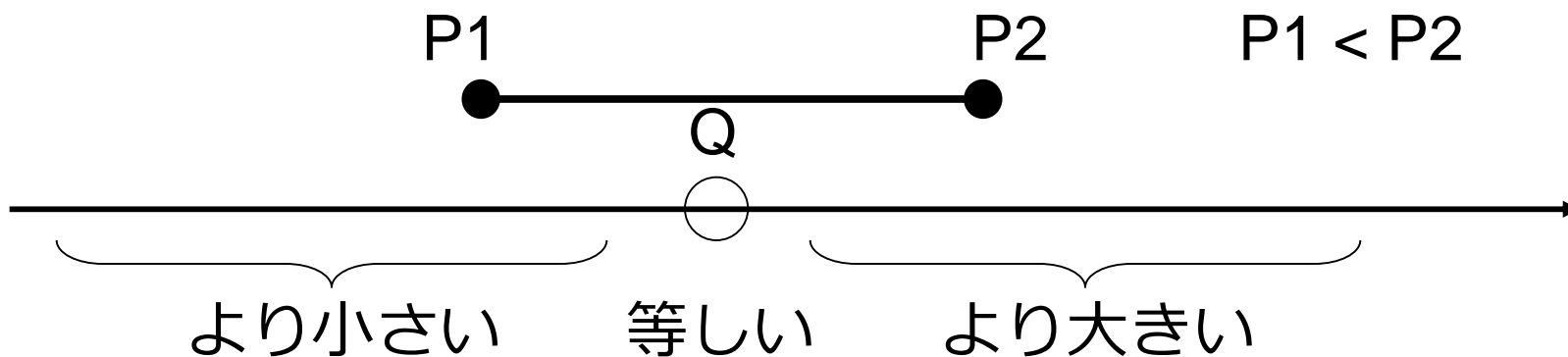
- 「多次元の値」を，そのまま扱う
 - query point は多次元





「区間」データの述語

区間と、ある値との関係



P1 < Q	P2 < Q	
P1 < Q	P2 = Q	
P1 < Q	P2 > Q	
P1 = Q	P2 < Q	ありえない
P1 = Q	P2 = Q	ありえない
P1 = Q	P2 > Q	
P1 > Q	P2 < Q	ありえない
P1 > Q	P2 = Q	ありえない
P1 > Q	P2 > Q	

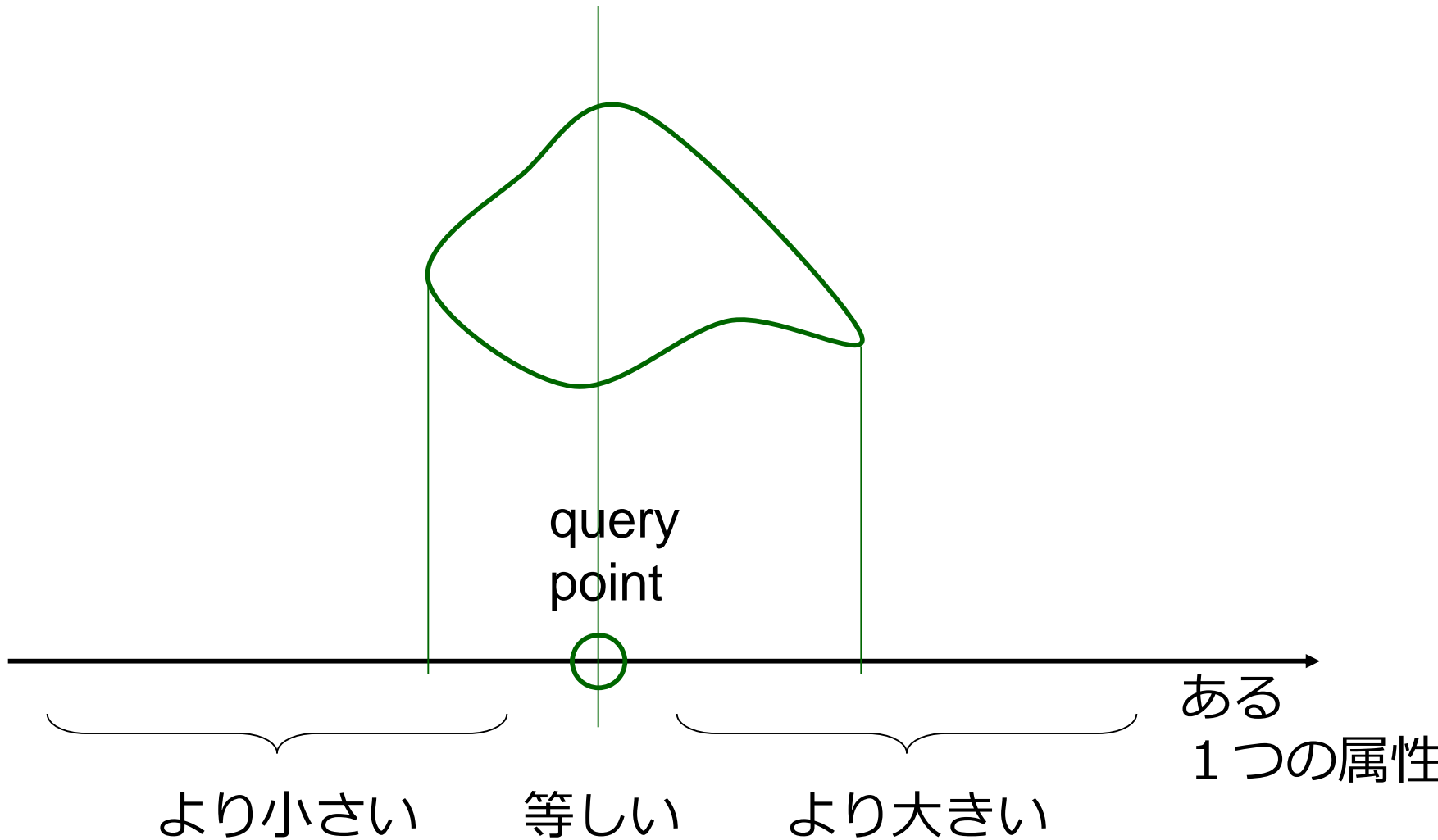
「区間」データの述語



$P1 < Q$	$P2 < Q$	
$P1 < Q$	$P2 = Q$	
$P1 < Q$	$P2 > Q$	
$P1 = Q$	$P2 > Q$	
$P1 > Q$	$P2 > Q$	



- 「領域」について，ある軸に投影して，比較を行う
 - query point はスカラー値





- 「多次元の値」を，そのまま扱う
 - query point は多次元

