

# 4. 画像の内容検索

(マルチメディアデータベース序論, 全6回)

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

金子邦彦



# Content-Based Image Retrieval

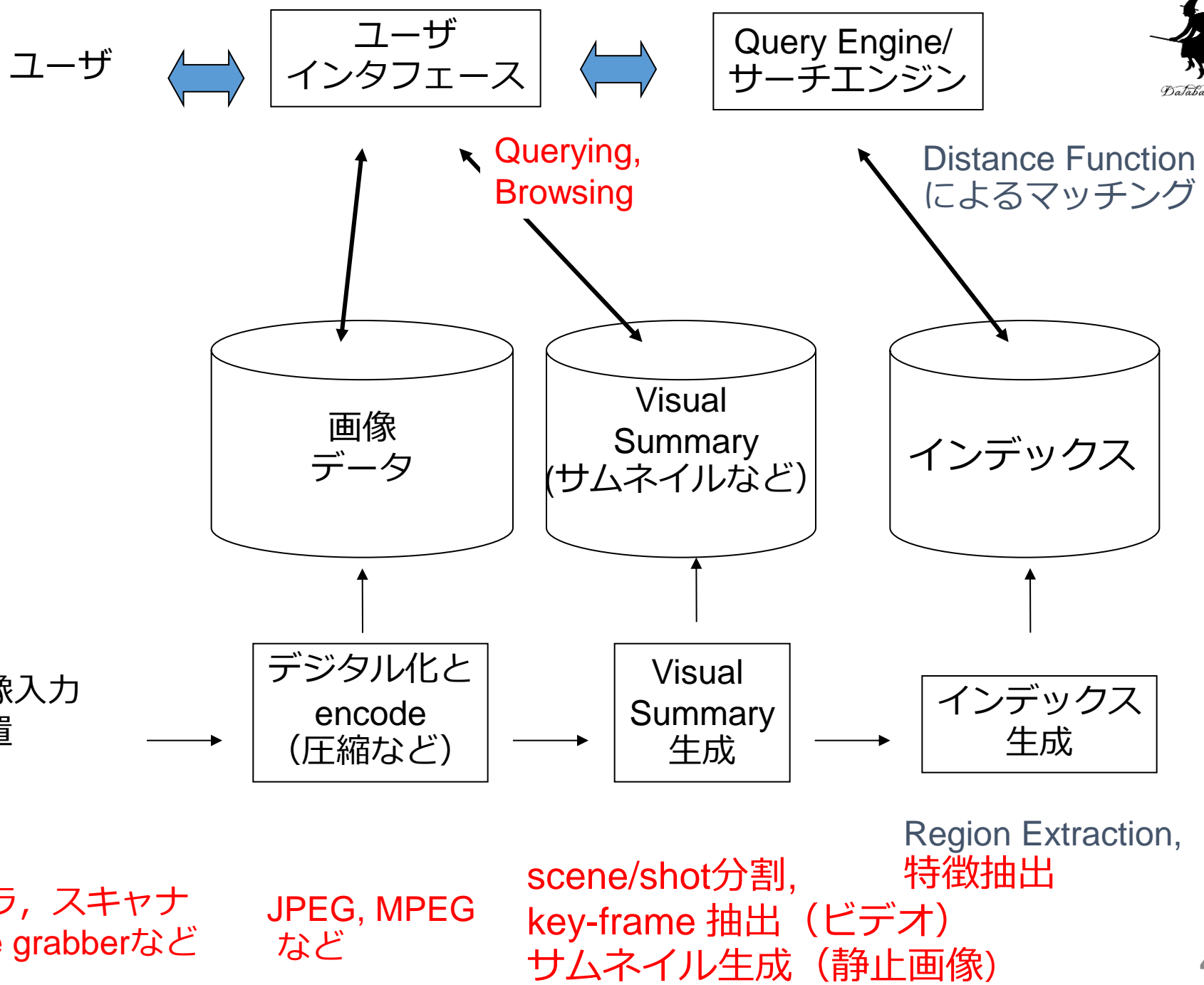


- 画像（静止画像／ビデオ）の「内容」による検索
  - 検索条件
  - Perceptual property
    - Color, Texture, Shape
    - Spatial Layout, Motion Feature(ビデオ)
  - Semantic Primitives
    - Objects, Roles, Scenes,
  - Subjective Attributes
    - Impressions, Emotions

# 静止画像の Perceptual Property



- Visual property
  - Color
  - Texture
  - Shape
  - Motion Feature
- Spatial Layout
  - 位置(absolute position)
  - 大きさ (size)
  - 関係(spatial relationship) : 相対的な位置関係



カメラ, スキャナ  
frame grabberなど

JPEG, MPEG  
など

scene/shot分割,  
key-frame 抽出 (ビデオ)  
サムネイル生成 (静止画像)

Region Extraction,  
特徴抽出

# Query by Perceptual Properties



- Perceptual property
  - (例) ビデオから「赤」が50%のcutを探せ
  - (例) 「丸いもの」が写っている画像を探せ

# Query by Spatial Layout

- Visual property

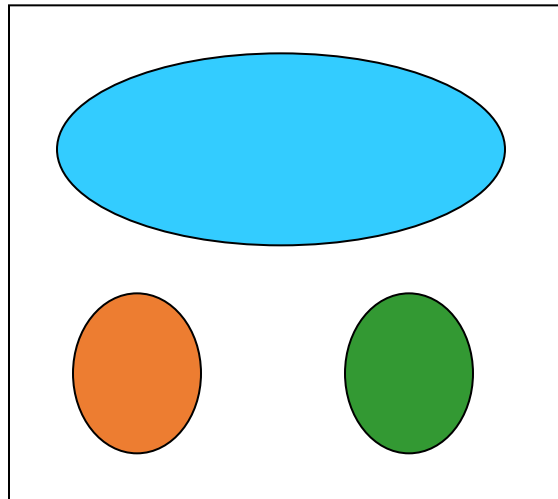
- Color

青、緑、水色

- Spatial Layout

- 位置(absolute position)

左下、右下、上



# Query by 'X'



- an Image Example
  - ユーザが例示画像を与える。最も似た画像を探す
- a Visual Sketch
  - ユーザが「欲しい」画像のスケッチを書く
- a Specification of Visual Features
  - Perceptual Properties を直に記述する
- a Keyword or Complete Text
  - データベース内の画像すべてに、あらかじめキーワードを（人手で）つけておく

# Content-Based Image Retrieval での Query Engine/ Search Engine



- ユーザは, 自分の「見たい」画像について：
  - うまく記述できない
  - 「見たい」ものが何かを決めていない（偶然の発見を楽しむ）ことさえある
- 解)
  - 画像データを「分類」しておいて, 似た画像をグループ化しておく
  - ユーザに見やすく表示.
  - ユーザにとっては, 「余分な画像」が表示されないの  
で便利



# Content-Based Image Retrieval での Query Engine/ Search Engine



- 検索の精度の向上
- 解)
  - 画像を, 題材によって階層的に分類しておく
  - ユーザは, 自分の欲しい画像のカテゴリ (「動物」, 「ネコ」, 「風景」など) を探した後に, そのカテゴリ内で検索
  - ⇒ 検索範囲は, データベース全体でなく, 限定された部分. 検索の精度が上がる

# Features



- Low-level Feature:
  - 既存の image processing, computer vision の技術で抽出できる
  - color, texture など
- Middle-level Features:
  - 技術の確立には至っていない
  - 登場物の形など
- High-level Features:
  - 人手が必要
  - Semantic Primitives, Subjective Attributesなど

# 「色」と「色分布」

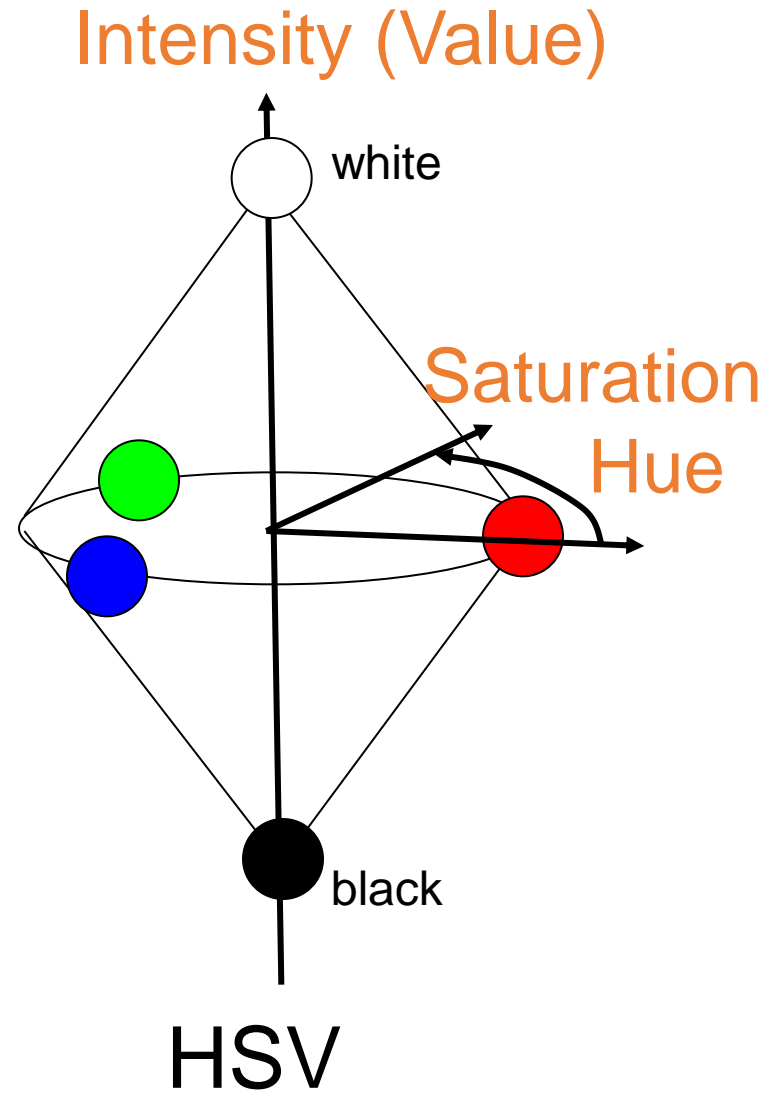
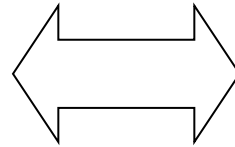
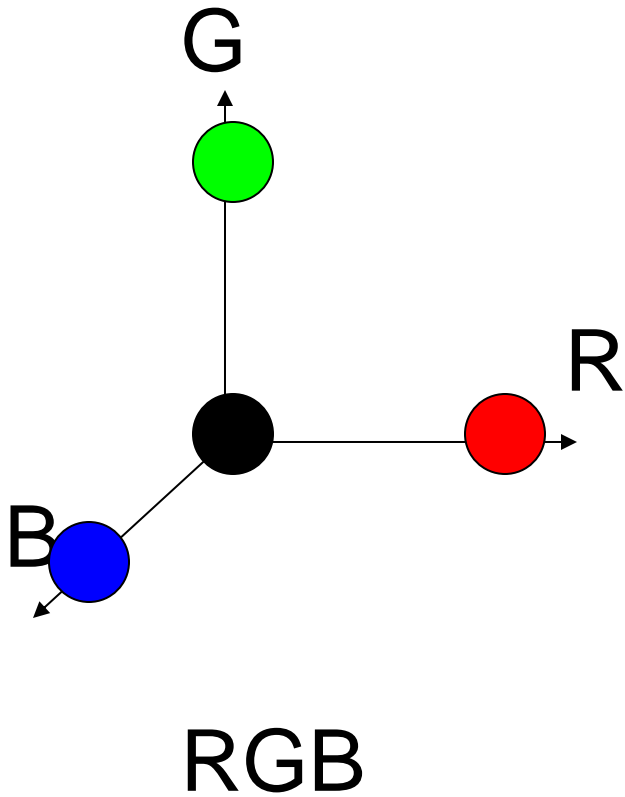


- Regional Color
  - 「画素」や「ある部分」の色のこと
  - 例) この部分は「深い緑色」である
- Global Color
  - 画像全体の色分布のこと
  - 例) この画像は、およそ50%が青で、30%が黄色
- 色と色分布は区別する

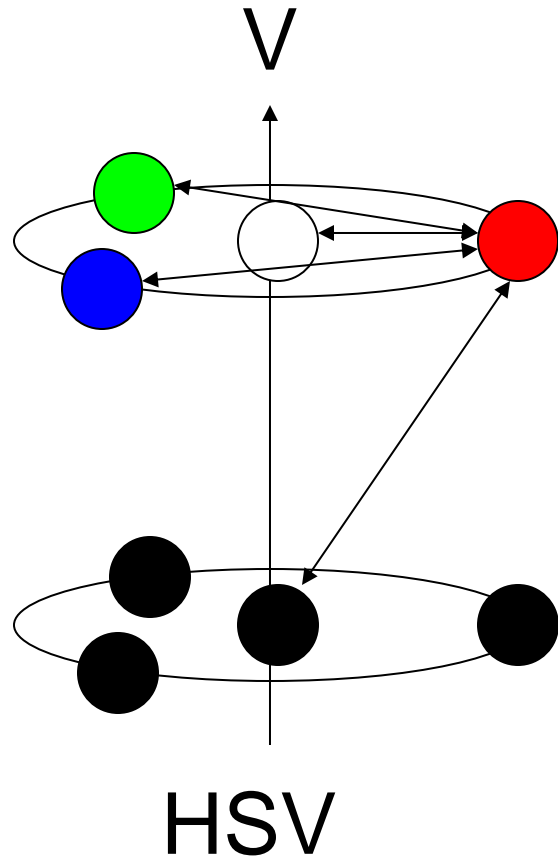
# 虹の七色

名前	英名	波長範囲
		770 nm
赤	Red	
		645 nm
橙色 (オレンジ)	Orange	
		580 nm
黄色	Yellow	
		550 nm
緑	Green	
		490 nm
青	Blue	
藍 (インジゴ)	Indigo	
		450 nm
紫	Violet	
		360 nm

# HSV と RGB



# 色の HSV 表現での距離



円筒空間での  
距離

# CIE 表色系



- CIEは「国際照明委員会」のこと
  - RGB系, XYZ系, UVW系, LAB系を提案
- R, G, B の加法混色で表色できないスペクトルがある
  - 実験では, 550nmより波長の短い緑色や青色のスペクトルの色は, R, G, B をどんな割合にしても表色できない
- スペクトルに無い光の色を, RGBの混合で作ることができる
  - 赤紫色

# L\* a\* b\* (エルスター、エースター、 ビースター) 表色系

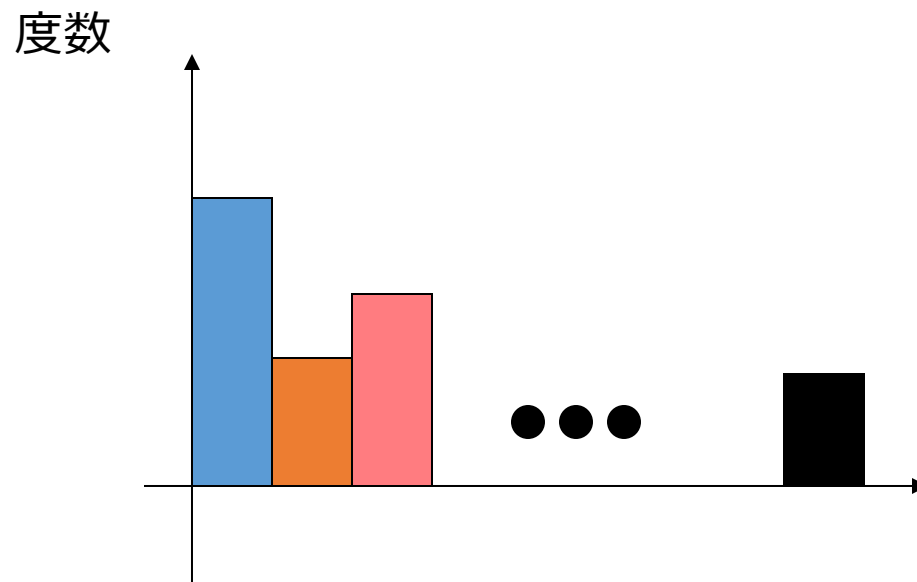
- L軸: 明るさ (Lightness)
- a\*軸: の+側は赤領域、-側は緑領域
- b\*軸: の+側は黄領域、-側は青領域



# カラーヒストグラム



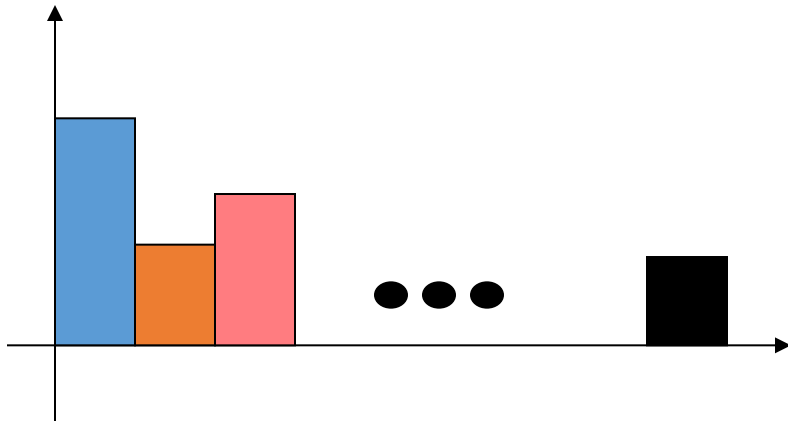
色分布を表現する手法の1つ



# カラーヒストグラムのマッチング

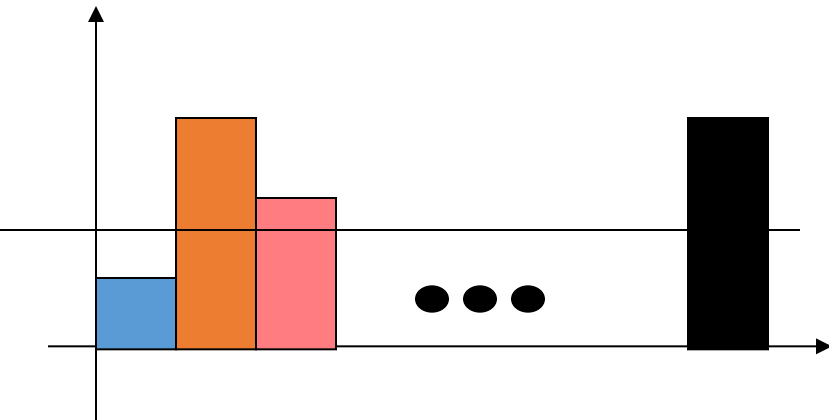


カラーヒストグラム C



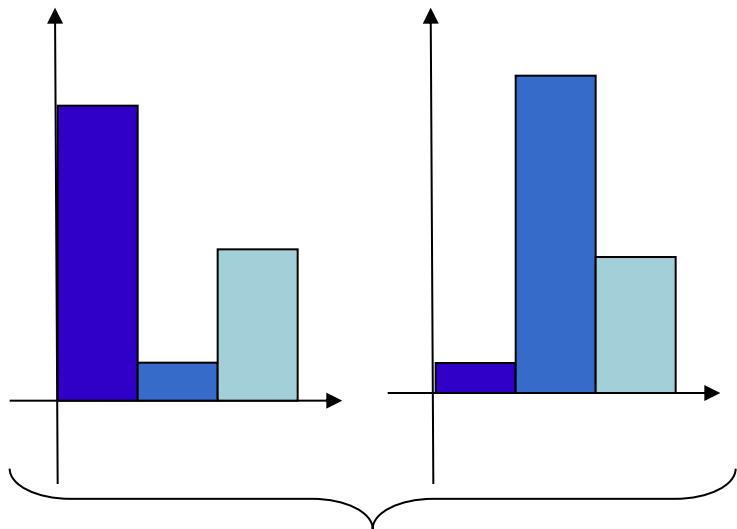
2つのカラーヒストグラムの  
マッチングには,  
2つのベクトル間の  
距離関数を使う

カラーヒストグラム t



## ユークリッド距離

$$\sum_i^N (c_i - t_i)^2$$



ユークリッド距離は遠いが、  
実際には似ている

## 色の correlation を考慮した 距離の例

$$\sum_i^N \sum_j^N \mathbf{a}_{ij} (c_i - t_i) (c_j - t_j)$$

$\mathbf{a}_{ij}$ : 2つのヒストグラムの  
 $i$ 番目の色と,  
 $j$ 番目の色の類似度

$$0 < \mathbf{a}_{ij} < 1$$

# カラーヒストグラムのコンパクトな表現



- 「0/1」 のベクトル表現
  - カラーヒストグラムから、しきい値を使って、「0/1」のベクトルを求める

