CS330 Image Understanding

Ch. 1

Photon's forms

- *Absorptions*: Black Shirts
- *Diffusions*: Puddle Reflection
- *Reflection*: Mirror
- *Transparency*: Window
- *Refractions*: Objects behind a glass of Water
- *Sampling*: Image digitization means that the function $f(x, y)$ is sampled into a matrix with M rows and N columns.
- *Quantization*: The image quantization assigns to each sample an integer value. The continuous range of the image function $f(x, y)$ is split into k intervals.

Brightness Levels

> **Note**
>
> If b bits are used, then num of **Brightness** levels is $k = 2^b$

- 1 bit means k=2 color (e.g. B&W)
- 8 bits k=256: so gray level range 0-255

Two types of light-sensitive receptors

1. **Cones**: Cone-shaped (lol), used in bright areas
2. **Rods**: used in low-light areas, e.g. seeing in the dark, gray scale

- **Human Cone Sensors**: es gibt 3 sensors that absorb RGB light:
- Long Wavelength `L` (aka red)
- Middle-Wavelength `M` (aka green)
- Short-wavelength `S` (blue)

Types of Resolution

An `8-bit` (*Bit Resolution*) `512` x `512` (*Spatial Resolution*) Image

---

Color Spaces

RGB

Normalized RGB

- Normalized Red = $r = R/(R+G+B)$
- Normalized Green = $g = G/(R+G+B)$
- Normalized Blue = $b = B/(R+G+B)$

HSI/HSV

- *Hue*: encoded as an angle (0-2 `PIE`)
- *Saturation*: is the distance to the vertical axis (0 *from center* to 1)
- *Intensity/Value* is the height along the vertical axis (0 *from bottom* to 1).

CIELAB, Lab, `L*a*b`

One Luminance Channel, and two colors, represented as sphere

How to transform RGB to YIQ

- $Y = 0.30R + 0.59G + 0.11B$
- $I = 0.60R - 0.28G + 0.32B$
- $Q = 0.21R - 0.52G + 0.31B$

RGB to YUV

- $Y = 0.39R + 0.59G + 0.11B$
- $U = 0.493 * (B - Y)$
- $V = 0.877 * (R - Y)$

Color Spaces Summary

- `RGB`: standard for cameras
- `HSI/HSV` hue, saturation, intensity
- `CIE L*a*b` intensity + 2 color channels
- `YIQ` Color TVs, Y is intensity

Pixel Aspect Ratio = describes how the width of a pixel compares to the height

---

Ch. 2

Histograms

- *Intersection* $(h(A), h(B))$ =

$$\sum_{j=1}^{numBins} min(h(A)[j], h(B)[j])$$

- *Similarity Score* $(h(A), h(B))$ =

$$\frac{intersection(h(A), h(B))}{\sum h(A)[j]}$$

> **Note**
> **Intersection** is the minimum between 2 bins, while **Similarity score** is the sum of intersections / the sum of h(A)

Edge Density & Direction

focuses mainly on `num` of edge pixels, and `direction` of the edge pixels

- `Num` produces Gradient Magnitude $Mag(p)$
- `Direction` produces gradient direction $Dir(p)$

EdgenessPerUnitArea: measures busyness, not the orientation

$$F_edgeness = \frac{|\{p|Mag(p) >= T\}|}{N}$$

for some threshold T & N pixels in area of interest

Tl;DR of Edges

- Edgeness per Unit is the `numOfEdges` / `pixelCount`
- Gradient Magnitude focuses on features such as colors, example: dark/light `->` (0.24,0.76)
- Gradient Direction focuses on direction, example: horz/vert/diagonal `->` (0.24,0.76, 0)

Local Binary Partition

- For each pixel p in the image, create an 8-bit number $B=b_0b_1b_2b_3b_4b_5b_6b_7$
- Check the eight neighbour of p

$$b_i = \begin{cases} 1, I_i > I_p \\ 0, else \end{cases}$$

b_0	b_1	b_2
b_3	*	b_4
b_5	b_6	b_7

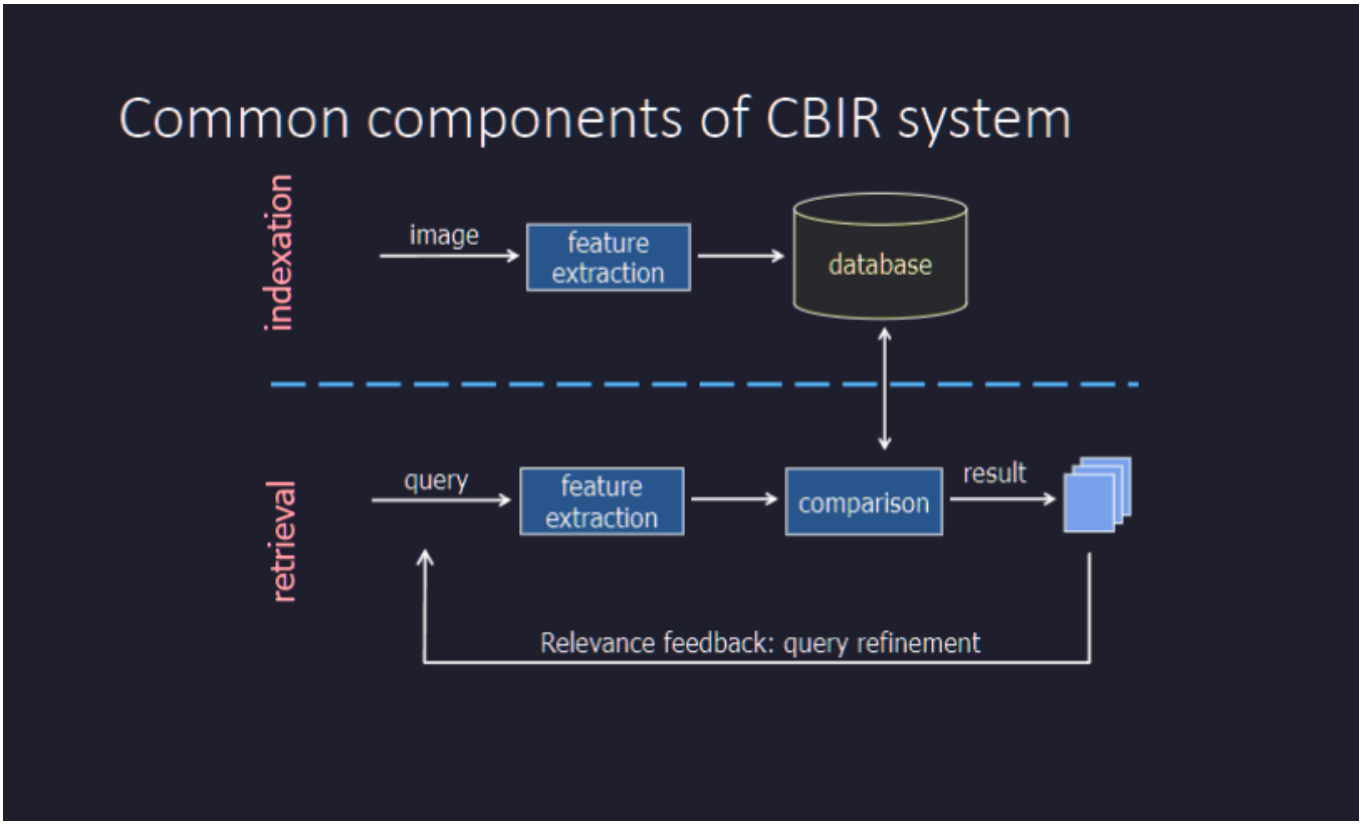
10	12	9
6	7	19
7	10	16

- Texture is represented by a histogram of B
- L1 distance can be used to compare two images

B= | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |

235

Ch. 3



Similarity Measures

used to measure similarity via distance between two feature vectors:

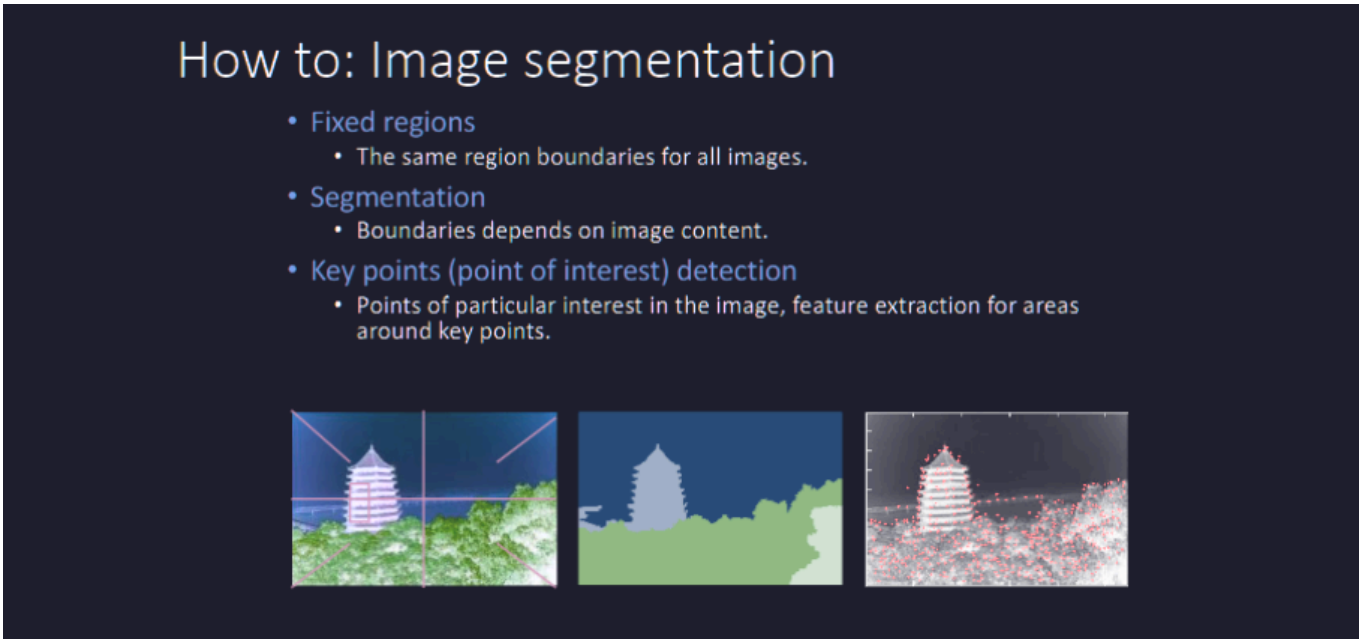
Manhattan Distance

$$MH(a,b) = \sum_{i=1}^{n} |x_i - y_i|$$

Euclidean Distance

$$d(A,B) = \sqrt{\sum_{i=1}^{n} (A_i - B_i)^2}$$

Image Segmentation



Effectiveness Measurement

Precision is also `numOfRelevantDocuments` / `totalDocsRetreived`

$$Precision = \frac{\text{true positive}}{\text{true positive} + \text{false positive}}$$

$$Recall = \frac{\text{true positive}}{\text{true positive} + \text{false negative}}$$

Recall is also `numOfRelevantDocuments` / `totalRelevantDocsInCollection`

$$F1\ \text{Score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

`P@N` or Recall at N are fancy ways of saying being applied only to top-n retrieved images

$$ErrorRate = \frac{numOfNONRelevantImagesRetrieved}{totalNumOfImagesRetrieved}$$

Mean Average Precision

fancy way of saying avg precision for multiple queries

$$MAP = \frac{\sum_Q precision(q_i)}{|Q|}$$

Mean Reciprocal Rank

measures how good the search ranks relevant images

PIL

```
plt.impshow(picArray) # Plots the Image
plt.impshow(picArray[:,:,1], cmap="gray") # Display Green Channel
```

```
suppressedChannel=picArray.copy()

# suppress impact of Red Channel
suppressedChannel[:,:,0]=0
plt.imshow(suppressedChannel)
```