# CS598:Visual information Retrieval

**Lecture II: Image Representation:**

**Color, Texture, and Shape**

# Re-cap of Lecture I

- What is visual information retrieval?
- Why do we care about it?
- What are the fundamental challenges?

# Lecture II: Part I

Color, Texture, Shape Descriptors

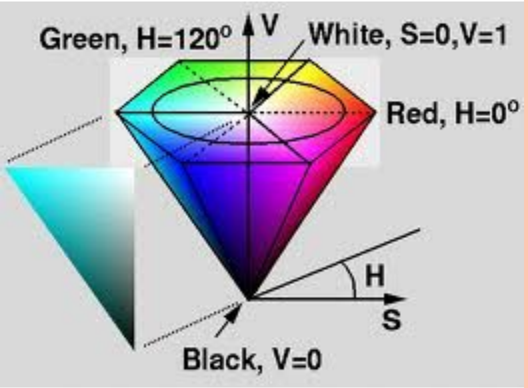
# Quiz:

- How would you describe the image?
- How would we make a computer to numerically encode such a description?

# Outline

- Color
  - Color histogram
  - Color correlogram
- Texture
  - Local binary pattern
- Shape
  - Histogram of oriented gradient

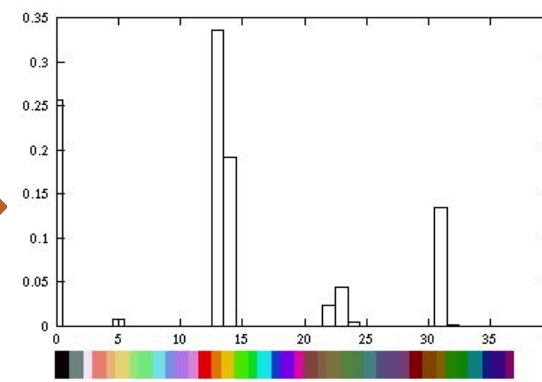
# Basics of color image




- For a color image of size W×H
  - Each pixel is represented by a (r,g,b) tuple,
  - The r, g, b represent the red, green, and blue component respectively,
  - The luminance of a color pixel can be calculated as
    - L=0.30r+0.59g+0.11b
  - Normalized RGB
    - nr=r/(r+g+b), ng=g/(r+g+b) when r+g+b≠0, 0 otherwise
  - HSV color model
    - RGB conversion to HSV
      - http://en.wikipedia.org/wiki/HSL_and_HSV

# Color histogram (1)

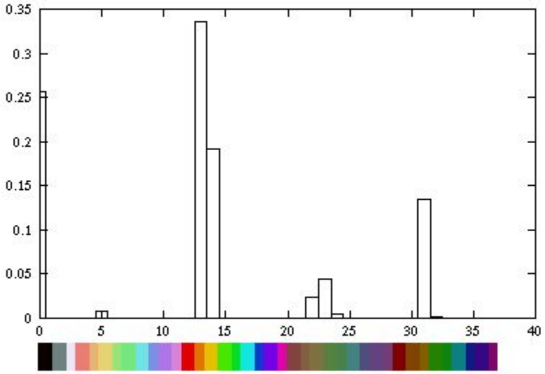
- The color histogram defines the image color distribution.
  - Partition the color space
  - Count of pixels for each color zone




*Color histogram*

# COLOR HISTOGRAM (2)

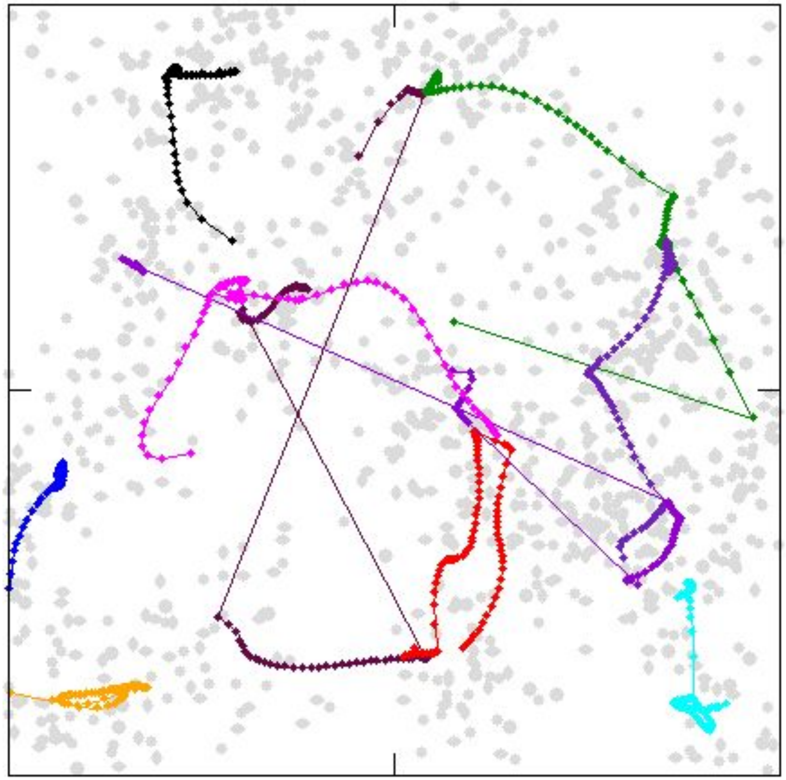
- How do we partition the color space?
  - Uniform partition of the color space
  - Clustering the color pixels




*Color histogram*

# Review of K-means clustering

- K-means clustering



# K-means clustering

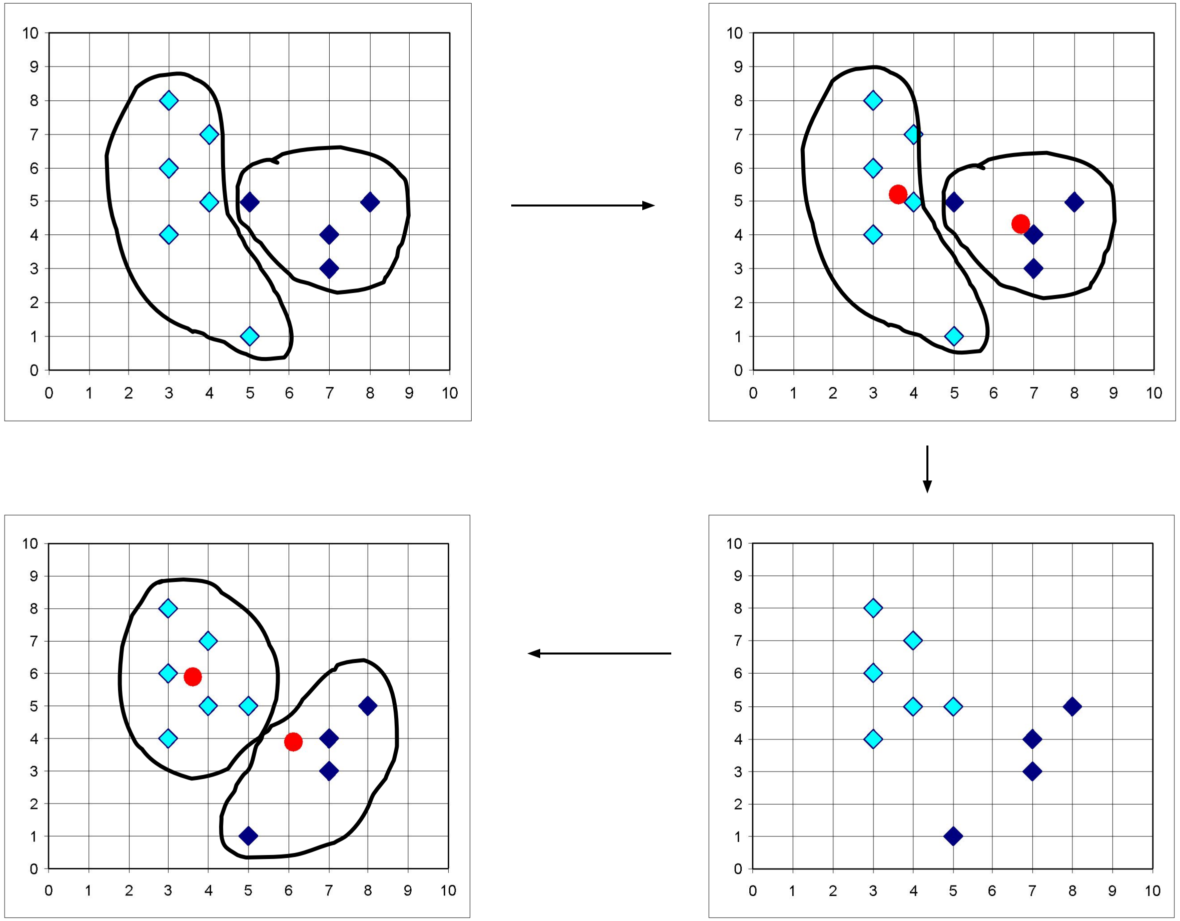
- Input
  - The number of cluster K.
  - The data collection $X_1, X_2, \dots, X_n$.
- Step 1: Choose a set of K instances as initial cluster center, i.e., $\mu_1, \mu_2, \dots, \mu_K$.
- Step 2: Iterative refinement until convergence
  - Assign each data sample $X_i$ to the closest cluster center
  - Recalculate the cluster center by averaging all the data samples assigned to it
- Output:
  - A set of refined cluster center

# Quiz: k-means

- Is K-means optimizing any objective function?

$$\boxed{\min_{\mu_1,\mu_2,\ldots,\mu_K} \sum_{h=1}^{K} \sum_{X_{ih} \in c(\mu_h)} \|X_{ih} - \mu_h\|^2}$$

# K-means: Example

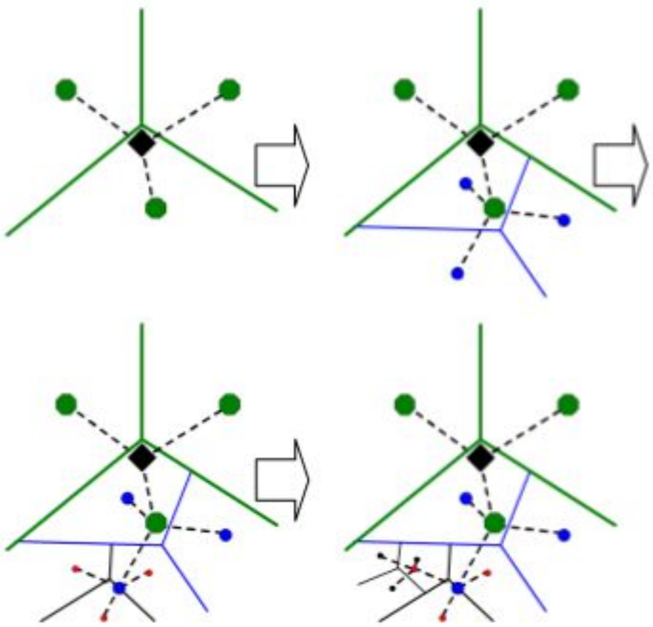


# Discussion: Variations of K-means

- How to select the initial K means?
- What distances to use?
- Strategies to calculate cluster means?

# Extension: Hierarchical K-means

- Hierarchical k-means
  - Building a tree on the training features
  - Children nodes are clusters of k-means on the parent node
  - Treat each node as a “word”, so the tree is a hierarchical codebook



Nister and Stewenius, “Scalable Recognition with a Vocabulary Tree,” CVPR 2006

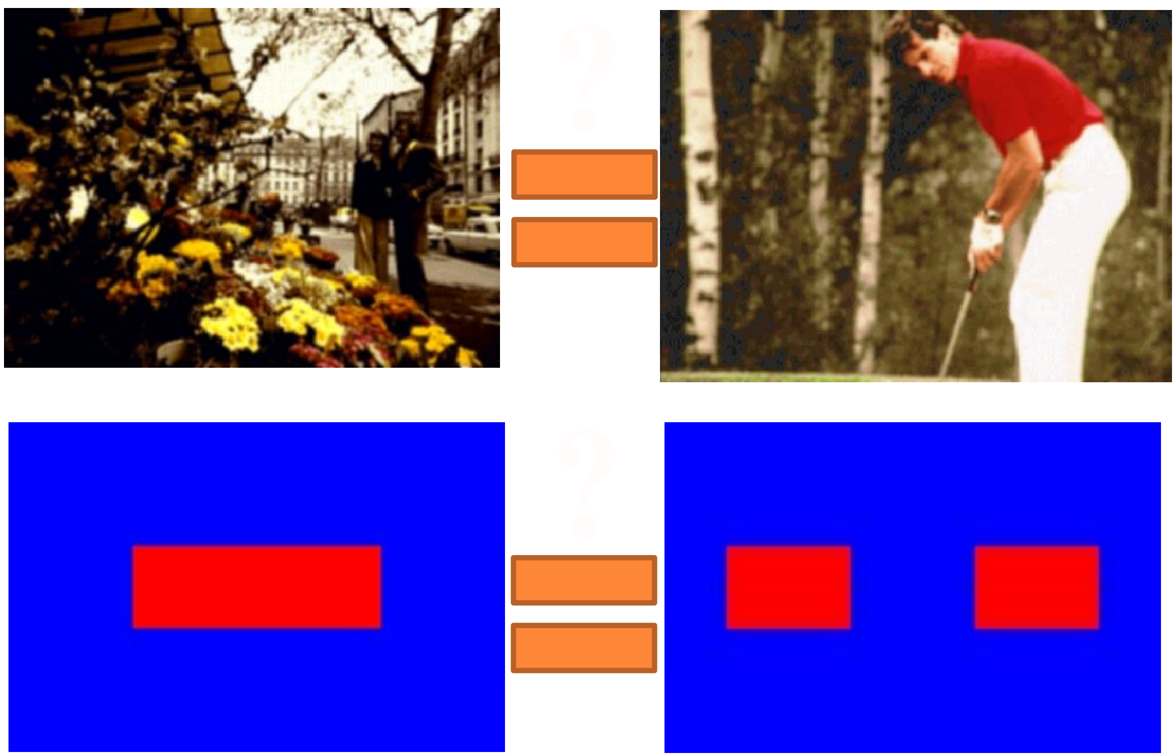
# Using k-means for color histogram

- Image independent clustering
  - Gather a large set of training images
  - Run K-means on all color pixels from all images
  - Use the learned cluster center to calculate the histogram
    - No need to encode the cluster centers in the histogram

- Image dependent clustering
  - Run K-means on the color pixels from each image
  - Use the learned clusters from each image to calculate the histogram for that image
    - The cluster centers need to be encoded together with the histogram.
    - It is also called a color signature of the image.

# Quiz: Color histogram

- What is the shortcoming of color histogram?



# Quiz: Color histogram

- How do you calculate the dominant color of an image?

# Outline

- Color
  - Color histogram
  - Color correlogram
- Texture
  - Local binary pattern
- Shape
  - Histogram of oriented gradient

# Color correlogram

- Color Correlogram is
  - a variant of histogram that accounts for local spatial correlation of colors;
  - based on stimation of the probability of finding a pixel of color *j* at a distance k from a pixel of color *i* in an image.

| | |
|---|---|
| Image: | $I$ |
| Quantized colors: | $c_1, c_1, \ldots\ldots\ldots c_m,$ |
| Distance between two pixels: | $\lvert p_1 - p_2 \rvert = max \quad \lvert x_1 - x_2 \rvert, \lvert y_1 - y_2 \rvert$ |
| Pixel set with color $c$: | $I_c = \; p \mid I(p) = c$ |
| Given distance: | $k$ |

# Color correlogram

- The color correlogram is defined as

$$\gamma_{c_i,c_j}^{(k)}(I) = \Pr_{p_1 \in I_{c_i}, p_2 \in I}[p_2 \in I_{c_j} \,||\, p_1 - p_2 | = k]$$

- The auto-correlogram is

$$\alpha_c^{(k)}(I) = \gamma_{c,c}^{(k)}(I)$$

# QUIZ

- What is the color auto-correlogram when k=0?

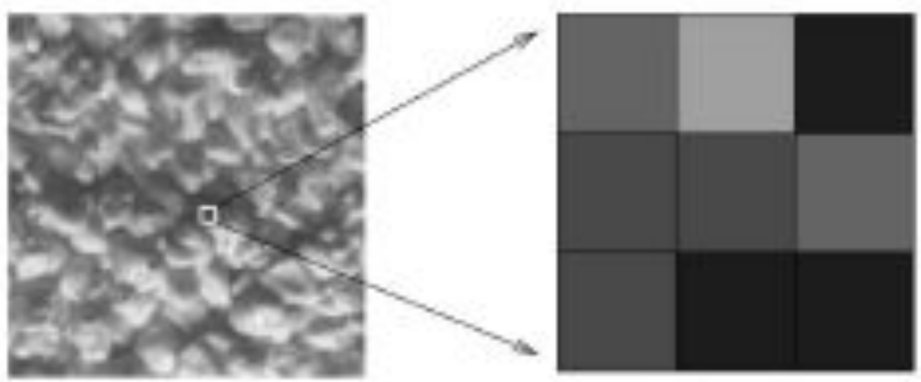
# Outline

- Color
  - Color histogram
  - Color correlogram
- Texture
  - Local binary pattern
- Shape
  - Histogram of oriented gradient

## Summary of Local Binary

- A 2D surface texture descriptor
- Simple to compute
- Invariant w.r.t. illumination changes
- Invariant w.r.t. spatial rotation of objects

# Local binary pattern (LPB)



Original Image

Sub-Image

| | | |
|---|---|---|
| 85 | 99 | 21 |
| 54 | 54 | 86 |
| 57 | 12 | 13 |

Sub-Image

| | | |
|---|---|---|
| 1 | 1 | 0 |
| 1 | | 1 |
| 1 | 0 | 0 |

11010011

11101001

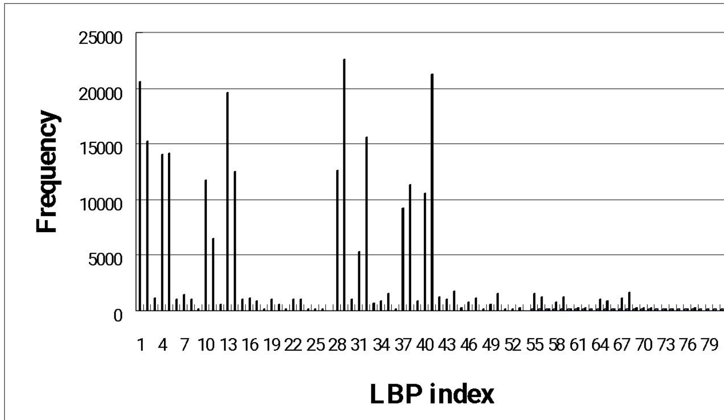
11110100

⋮

10100111

$(00111101)_2 \rightarrow (61)_{10}$

LBP index

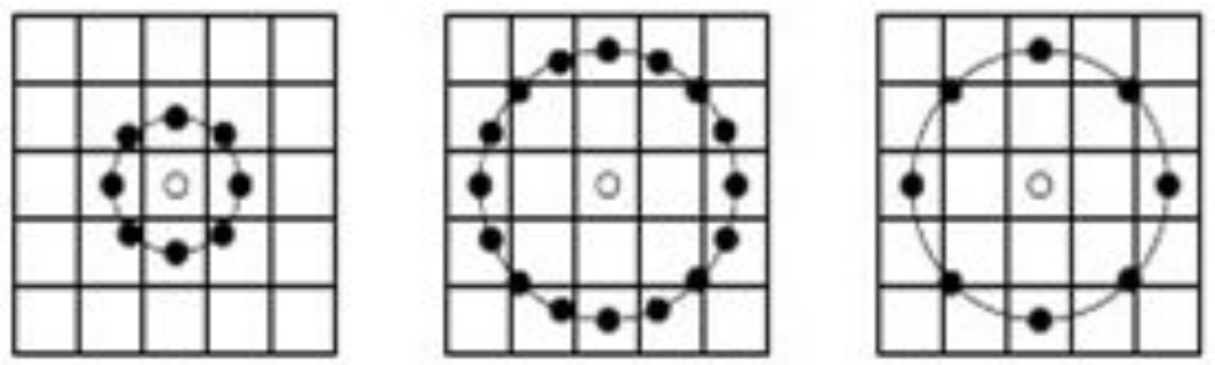


# Quiz

- Why is LBP illumination invariant?
- How to describe texture at different scale?

# LBP at different scale

- Extend LBP operator to use neighborhoods of different sizes
- Defining the local neighborhood as a set of sampling points evenly spaced on a circle centered at the pixel
- If a sampling point does not fall in the center of a pixel using bilinear interpolation.



# LBP: Uniform pattern

- Uniform Pattern
  - A local binary pattern is called uniform if the binary pattern contains at most two circular bitwise transitions from 0 to 1 or vice versa
- Examples:
  - 00000000 (0 transitions) is uniform
  - 01110000 (2 transitions) is uniform
  - 11001111 (2 transitions) is uniform
  - 11001001 (4 transitions) is NOT uniform
  - 01010011 (6 transitions) is NOT uniform
- There are 58 uniform patterns for 8bit LBP

# LBP HISTOGRAM OF UNIFORM PATTERNS

- Each uniform pattern corresponds to one bin.
- All non-uniform patterns are mapped to one bin
- A 59 dimensional histogram can thus be constructed for each image with 8bit LBP

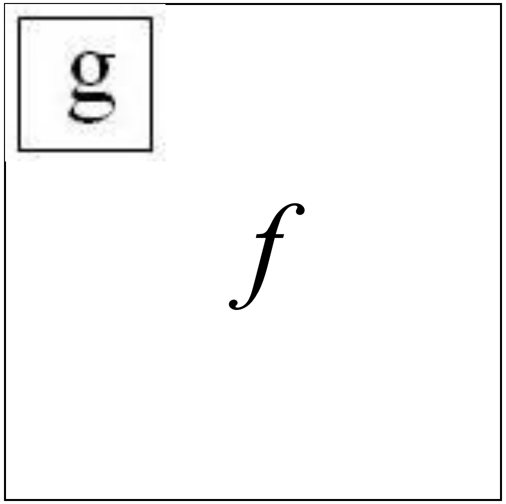
# Outline

- Color
  - Color histogram
  - Color correlogram
- Texture
  - Local binary pattern
- Shape
  - Histogram of oriented gradient

# Basics of image filtering

- Let $f$ be the image and $g$ be the kernel. The output of convolving $f$ with $g$ is denoted $f * g$.

$$(f * g)[m,n] = \sum_{k,l} f[m-k, n-l]\, g[k,l]$$




- MATLAB functions: conv2, filter2, imfilter

Source: F.

# Derivatives with convolution

For 2D function f(x,y), the partial derivative is:

$$\frac{\partial f(x,y)}{\partial x} = \lim_{\varepsilon \to 0} \frac{f(x+\varepsilon, y) - f(x,y)}{\varepsilon}$$

For discrete data, we can approximate using finite differences:

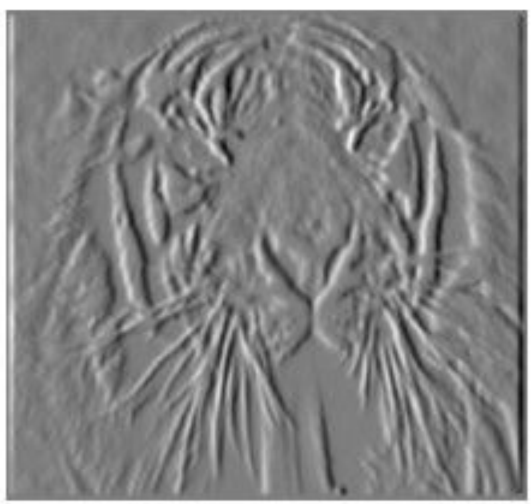
$$\frac{\partial f(x,y)}{\partial x} \approx \frac{f(x+1, y) - f(x,y)}{1}$$

To implement above as convolution, what would be the associated filter?

Source: K. Grauman

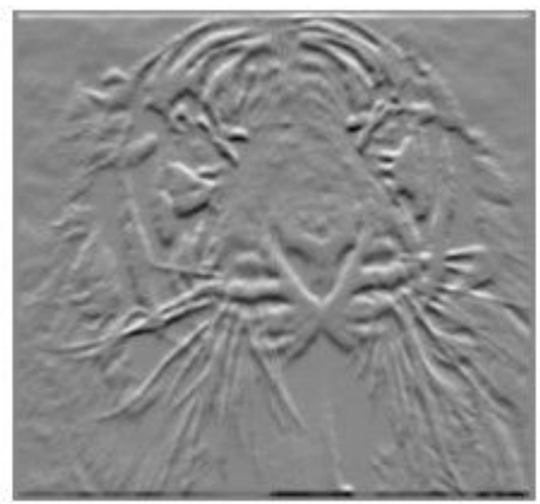
# Partial derivatives of an image

$$\frac{\partial f(x, y)}{\partial x}$$



| | |
|---|---|
| -1 | 1 |

$$\frac{\partial f(x, y)}{\partial y}$$



| |
|---|
| -1 |
| 1 |

or

| |
|---|
| 1 |
| -1 |

Which shows changes with respect to x?

# Finite difference filters

- Other approximations of derivative filters exist:

**Prewitt:** $M_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$ ; $M_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$

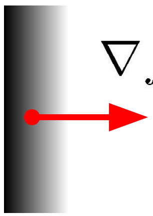
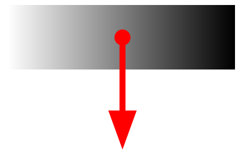
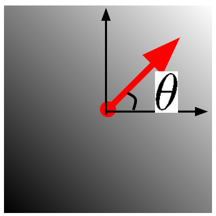
**Sobel:** $M_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$ ; $M_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$

**Roberts:** $M_x = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$ ; $M_y = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$

Source: K. Grauman

# Image gradient

- The gradient of an image: $$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$$

- 
$\nabla f = \left[\frac{\partial f}{\partial x}, 0\right]$

$\nabla f = \left[0, \frac{\partial f}{\partial y}\right]$


$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$

The gradient points in the direction of most rapid increase in intensity

- How does this direction relate to the direction of the edge?

The gradient direction is given by $\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$

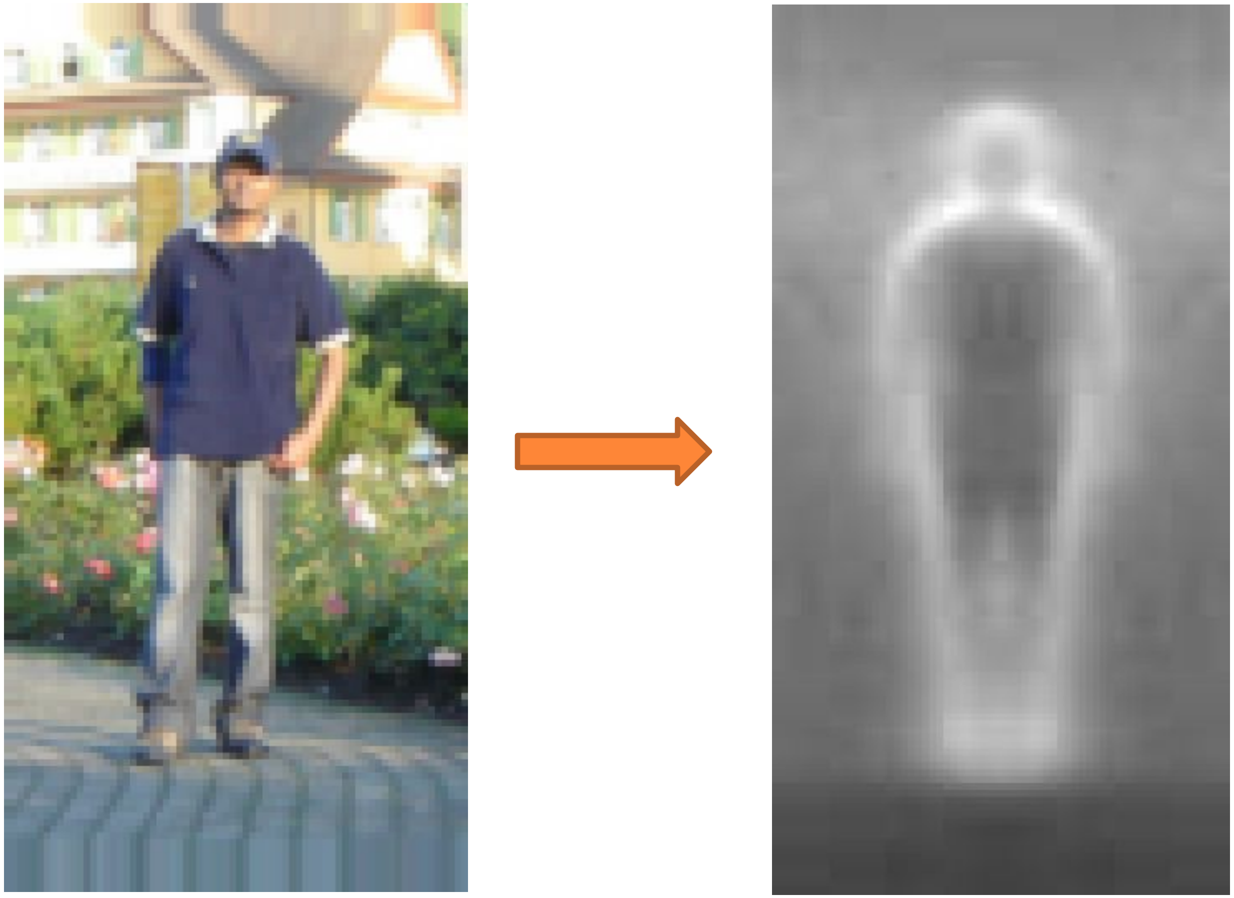
The *edge strength* is given by the gradient magnitude

$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

Source: Steve Seitz

# Histogram of oriented gradient (1)

- Calculate the gradient vectors at each pixel location

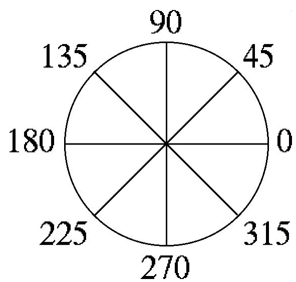


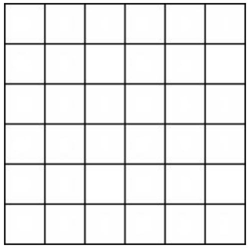
Source: P. Barnum

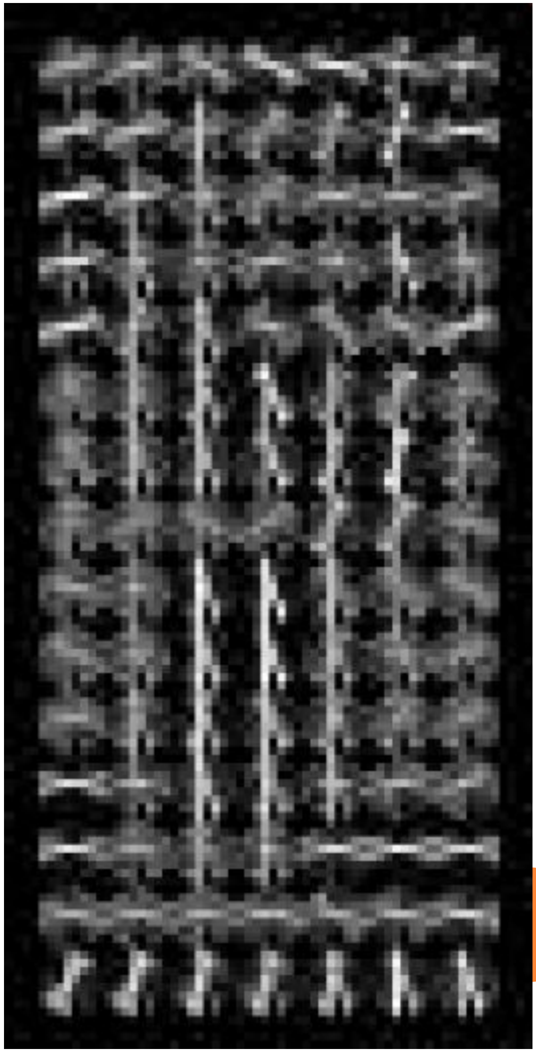
# Histogram of oriented gradient (2)

- Histogram of gradient orientations

  -Orientation -Position

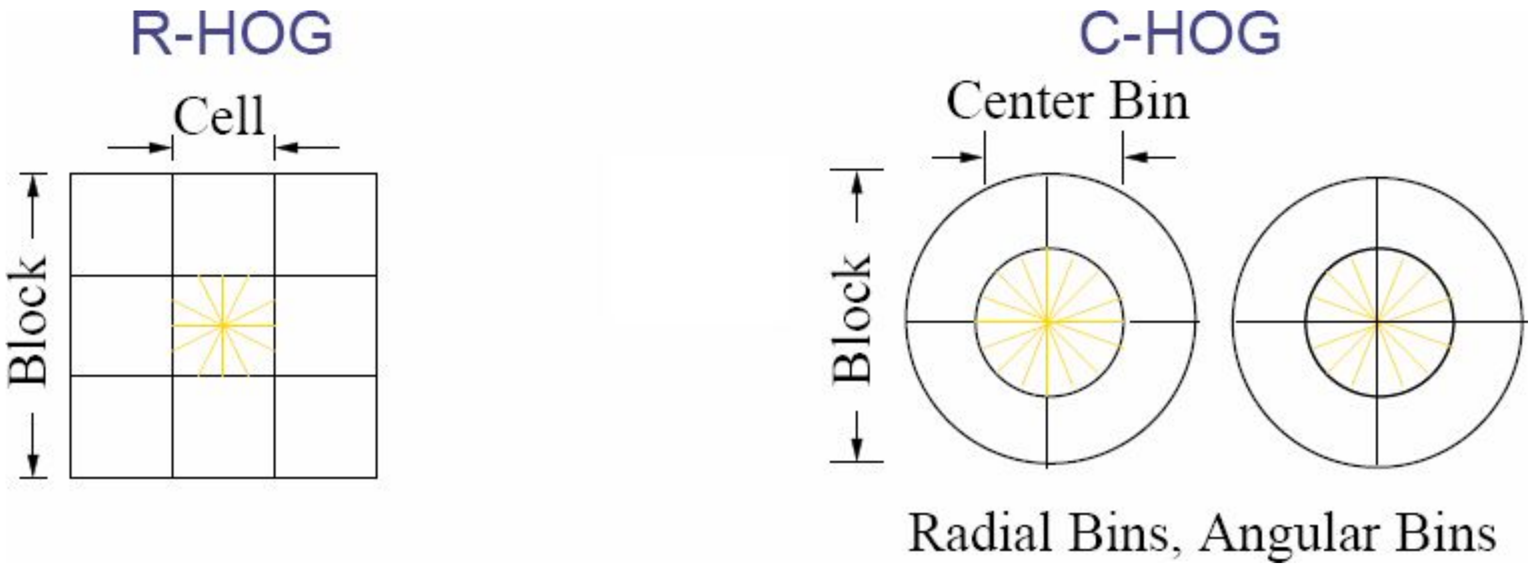




- Weighted by magnitude

# Histogram of oriented gradient (3)




$L1-norm : v \longrightarrow v/(||v||_1 + \epsilon)$

$L2-norm : v \longrightarrow v/\sqrt{||v||_2^2 + \epsilon^2}$

$L1-sqrt : v \longrightarrow \sqrt{v/(||v||_1 + \epsilon)}$

$L2-hys$ : L2-norm, plus clipping at .2 and renomalizing

# From HoG to Global Image Description

- Strategy 1: Pooling
  - Average Pooling: Take the average the HoG vectors from all blocks to form a single vector
  - Max Pooling: Take the max of the HoG vectors from all blocks to form a single vector

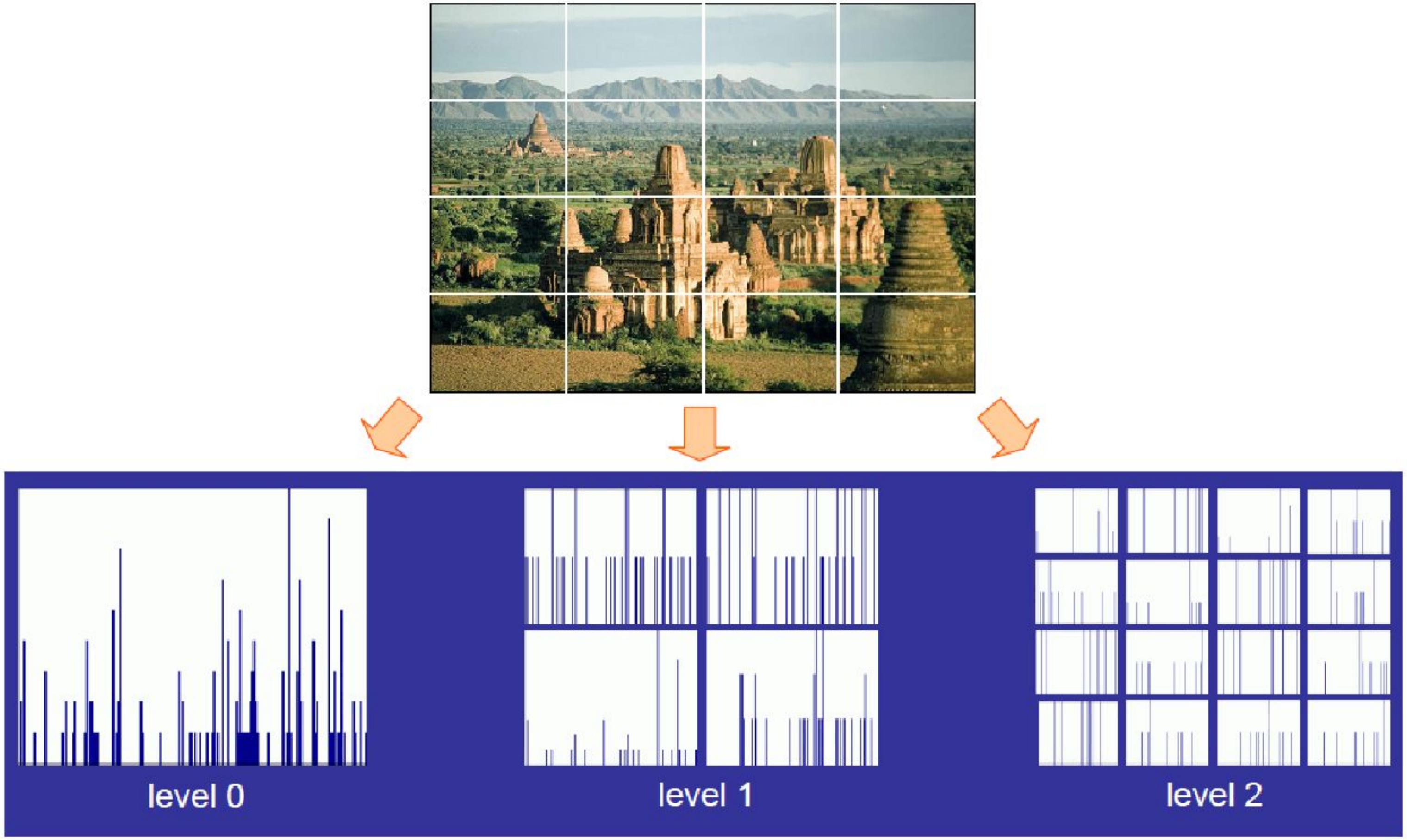
- Strategy 2:
  - Concatenate all HoG vectors to form a single vector

# Quiz

- How do you compare these two strategies?

# Coarse-to-fine spatial match

- Spatial pyramid




Source: S. Lazebnik

# Lecture II: Part I I

Similarity and distance measures

# Outline

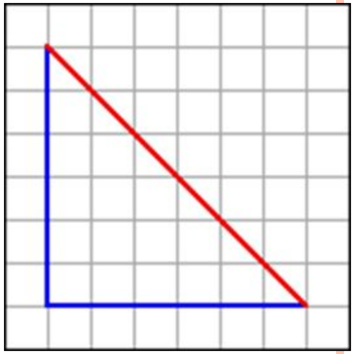
- Distance measures
  - Euclidean distances (L2) distances
  - L1 and Lp distances
  - Chi-square distances
  - Kullback-Liebler divergence
  - Earth mover distances

- Similarity measures
  - Cosine similarity
  - Histogram intersection

# Distance between two histograms

- Let's start with something familiar to you…

$$D(I,J) = \sqrt{\sum_i (H_I(i) - H_J(i))^2}$$



where $H_I$ and $H_J$ are the global histogram descriptors extracted from image I and J respectively

- How do we usually call it?
  - Euclidean distance or $L_2$ distance

# Quiz

- What is the potential issue with $L_2$ distance?

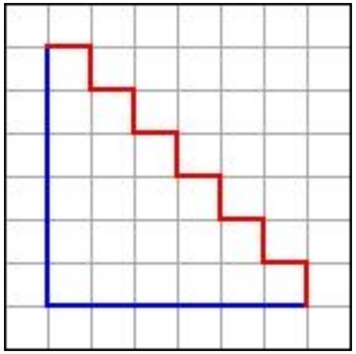
$$\boxed{D(I,J) = \sqrt{\sum_i |H_I(i) - H_J(i)|^2}}$$

- Is it robust to noise?

# $L_1$ DISTANCE

- Now consider another distance…

$$D(I,J) = \sum_i |H_I(i) - H_J|$$



- How do we usually call it?
  - $L_1$ distance
  - Absolute distance
  - City-block distance
  - Manhattan distance

- Is it more robust to noise?

# $L_P$: Minkowski-form distance

- $L_p$ distance

$$D(I,J) = \left( \sum_i |H_I(i) - H_J(i)|^p \right)^{1/p}$$

- Spatial cases
  - $L_1$ : absolute, cityblock, or mahattan
  - $L_2$ : Euclidean distance
  - $L_\infty$ : Maximum value distance

# Outline

- Distance measures
  - Euclidean distances (L2) distances
  - L1 and Lp distances
  - <u>Chi-square distances</u>
  - Kullback-Liebler divergence
  - Earth mover distances

- Similarity measures
  - Cosine similarity
  - Histogram intersection

# Chi-square distances

- Motivated from nonparametric Chi-Square ($\chi^2$) test statistics

$$\boxed{D(I,J) = \sum_i \frac{2(H_I(i) - H_J(i))^2}{H_I(i) + H_J(i)}}$$

- It penalize dimensions which are large in value!
  - More robust to noise…..

# Outline

- Distance measures
  - Euclidean distances (L2) distances
  - L1 and Lp distances
  - Chi-square distances
  - Kullback-Liebler divergence
  - Earth mover distances

- Similarity measures
  - Cosine similarity
  - Histogram intersection

# KL Divergence

- Motivated from Information Theory
  - Cost of encoding one distribution as another

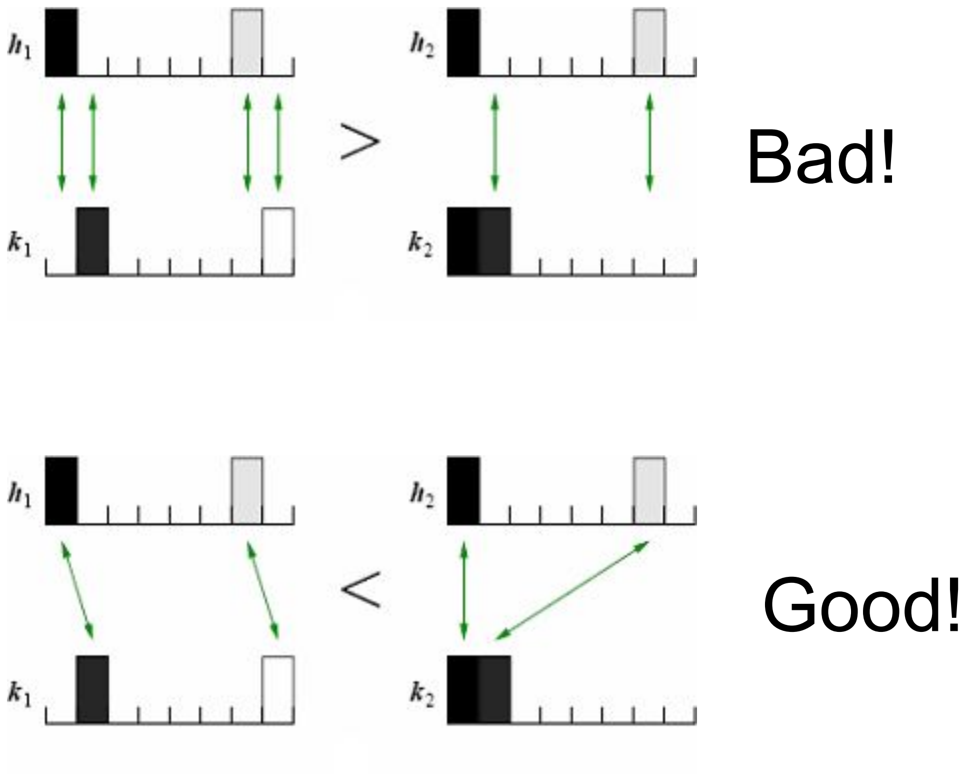
$$\boxed{KL(I,J) = \sum_i H_I(i) log \frac{H_I(i)}{H_J(i)}}$$

  - Unfortunately it is not symmetric
    - i.e., $KL(I,J) \neq KL(J,I)$
  - KL distances

$$\boxed{D(I,J) = \frac{KL(I,J) + KL(J,I)}{2}}$$

# Quiz:

- What is the problem of all the distance measures?



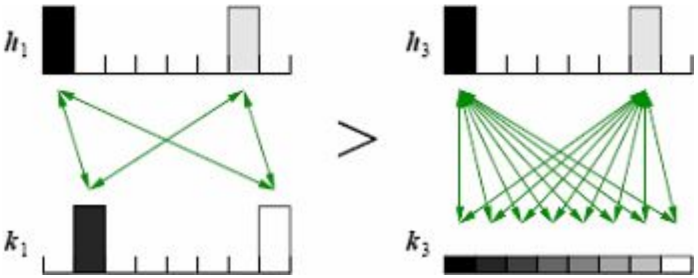
# Outline

- Distance measures
  - Euclidean distances (L2) distances
  - L1 and Lp distances
  - Chi-square distances
  - Kullback-Liebler divergence
  - <u>Earth mover distances</u>

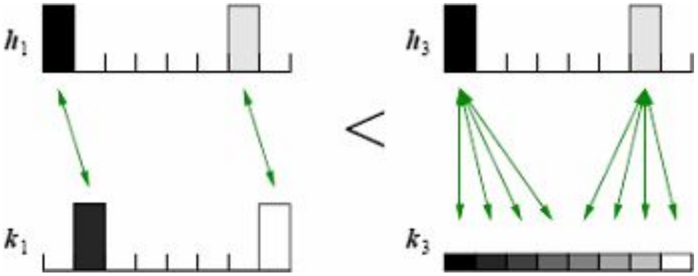
- Similarity measures
  - Cosine similarity
  - Histogram intersection

# A GOOD CROSS BIN MATCHING ALGORITHM

- Which of the following two is better?




Bad!




Good!

Slides credit on EMD : Frederik Heger

# Earth mover's distance (EMD)

- Earth mover distance is defined as

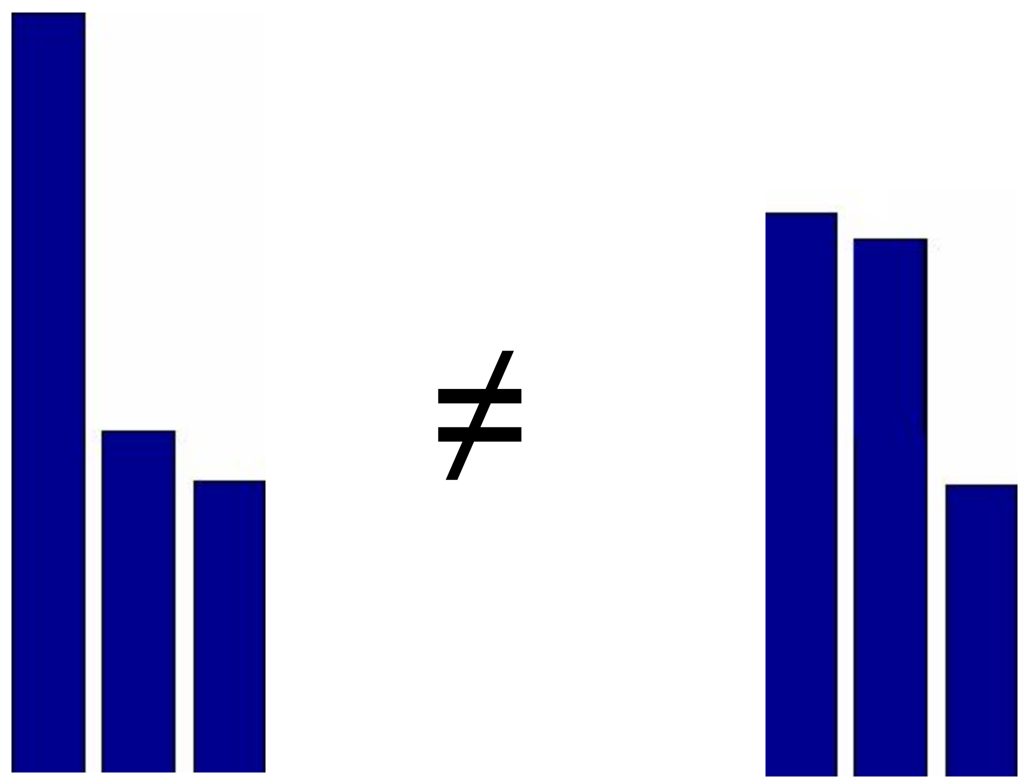
$$\boxed{D(I,J) = \frac{\sum_{i,j} f_{ij} d_{ij}}{\sum_{i,j} f_{ij}}}$$

where $d_{ij}$ is the distance between bin $i$ in $H_I$ and bin j in $H_J$, and $\{f_{ij}\}$ is transportation of value from different bins to make the two distribution equal.

- Notes
  - $\{f_{ij}\}$ is obtained by solving a linear optimization problem, the transportation problem
  - Minimal cost to transform one distribution to the other
  - Total cost = sum of costs from individual features

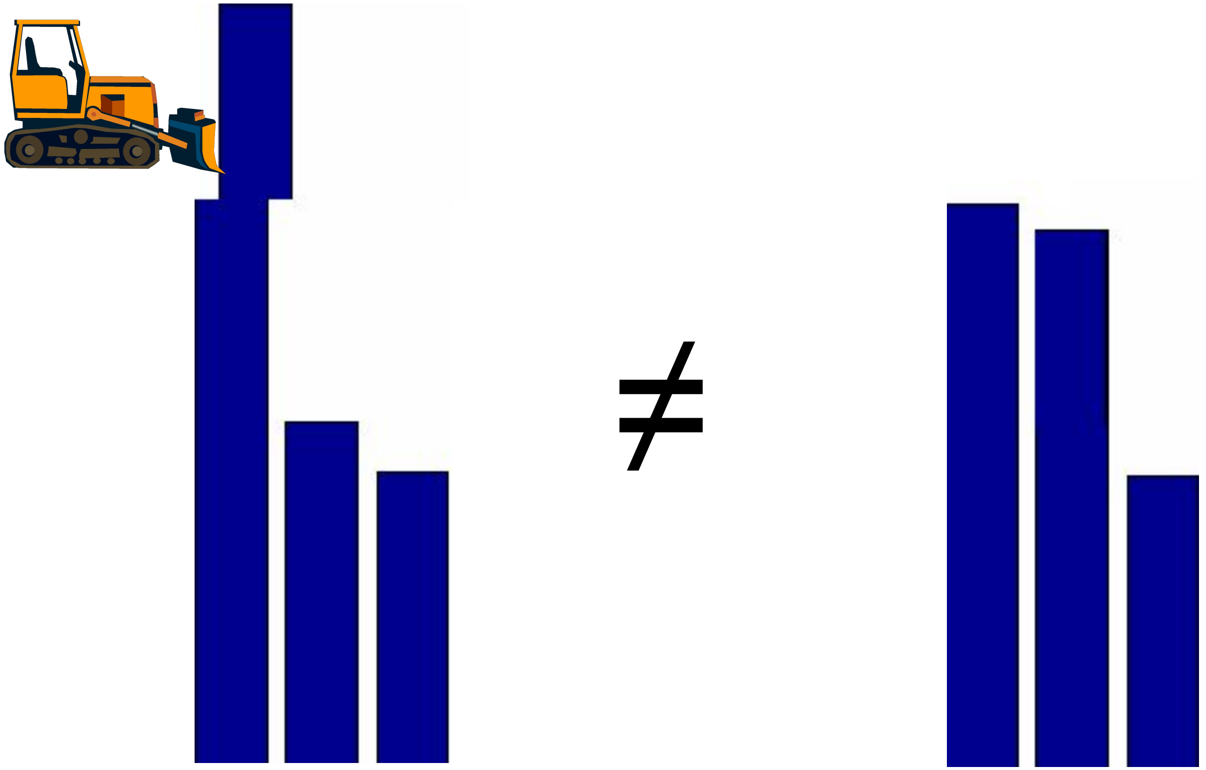
Slides credit on EMD : Frederik Heger

# EMD



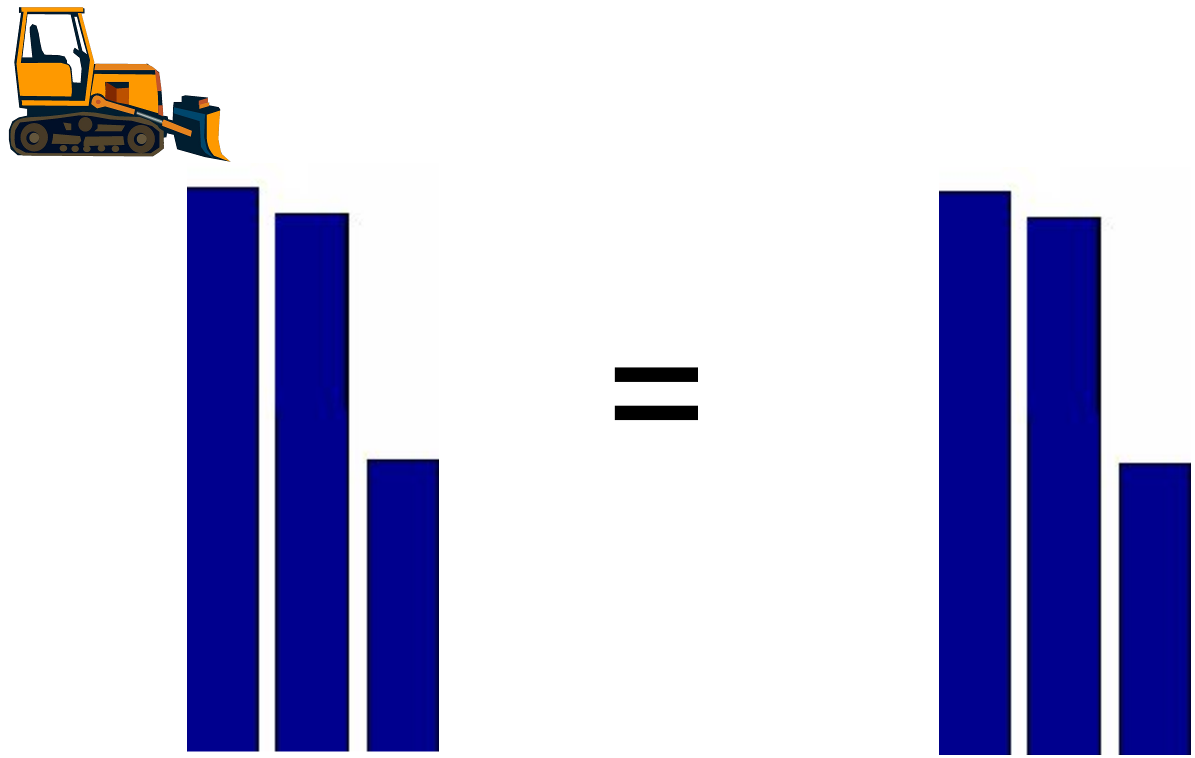

Slides credit on EMD : Frederik Heger

# EMD



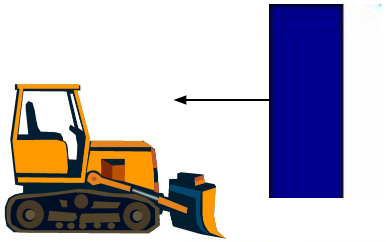
Slides credit on EMD : Frederik Heger

# EMD

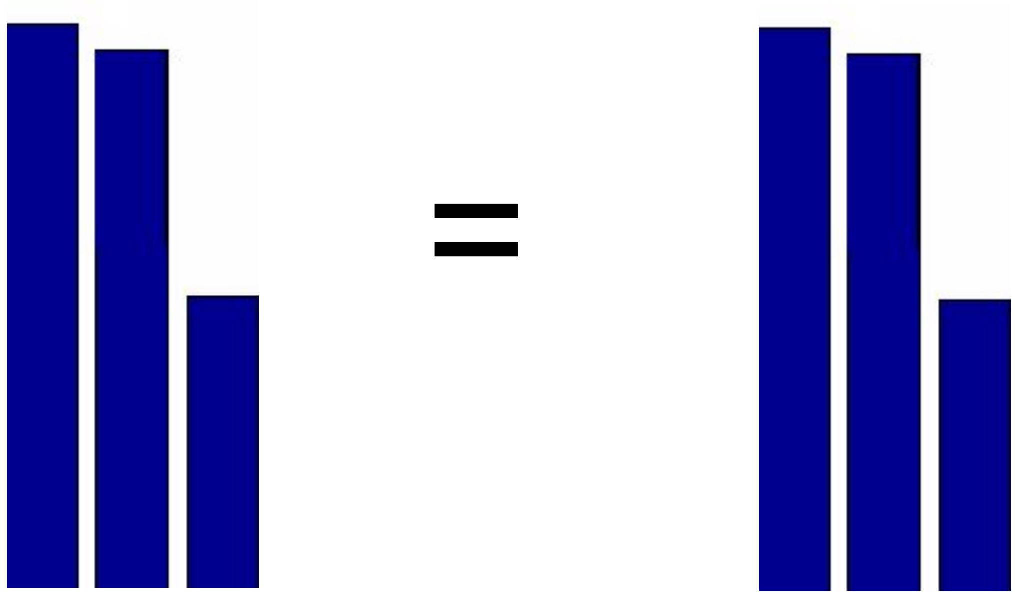


Slides credit on EMD : Frederik Heger

# EMD



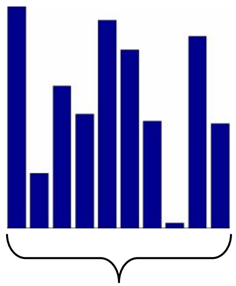
(amount moved) * (distance moved)



Slides credit on EMD : Frederik Heger

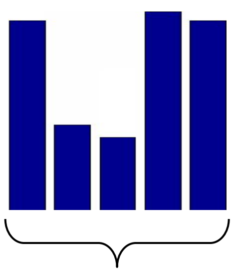
# EMD

I



m clusters

J



n clusters

$$\sum_{\text{All movements}} (\text{distance moved}) * (\text{amount moved})$$

$$\sum_{i=1}^{m} \sum_{j=1}^{n} (\text{distance moved}) * (\text{amount moved})$$

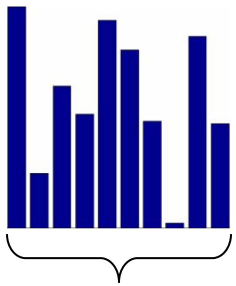
$$\sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij} * (\text{amount moved})$$

$$\sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij} f_{ij} = \text{WORK}$$

Slides credit on EMD : Frederik Heger

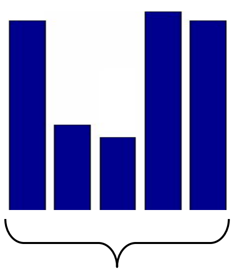
# EMD

I



m clusters

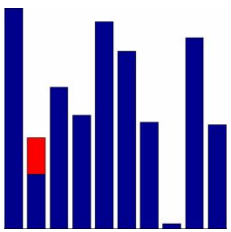
J



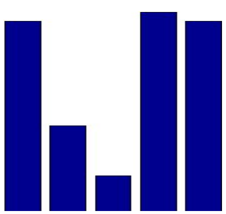
n clusters

Move earth only from P to Q

I'



J'

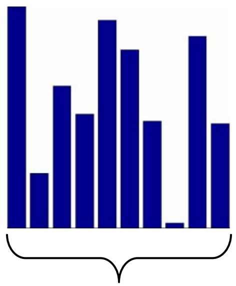


$f_{ij} \geq 0$

Slides credit on EMD : Frederik Heger

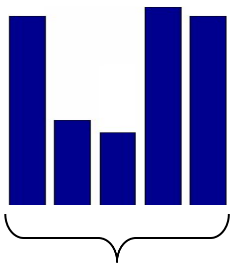
# EMD

I



m clusters

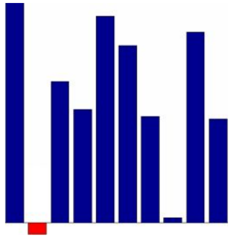
J



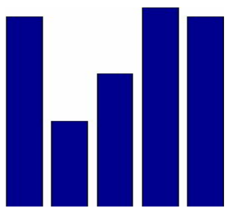
n clusters

I cannot send more earth than there is

I'



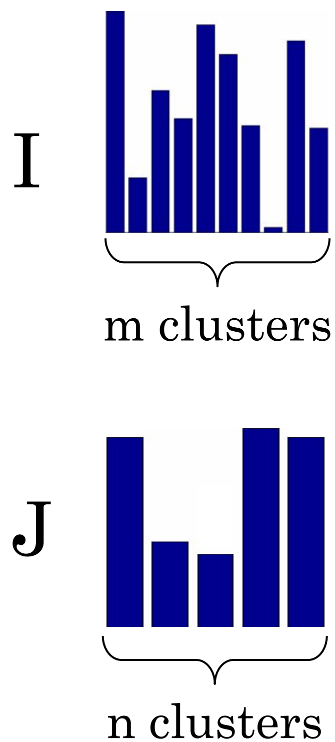
J'



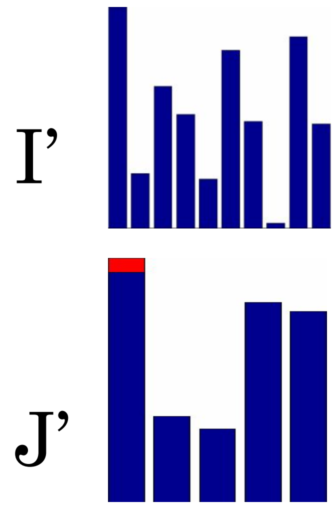
$$\sum_{j=1}^{n} f_{ij} \leq \mathrm{w}_{\mathrm{I}_\mathrm{i}}$$

Slides credit on EMD : Frederik Heger

# EMD




J cannot receive more earth than it can hold

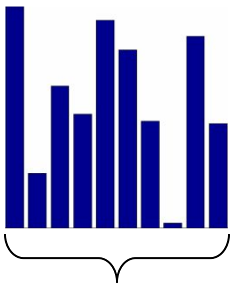



$$\sum_{i=1}^{m} f_{ij} \leq \mathrm{w}_{J_j}$$

Slides credit on EMD : Frederik Heger

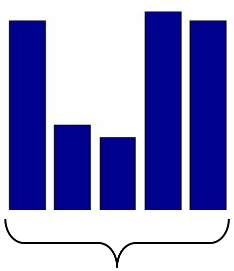
# EMD

I



m clusters

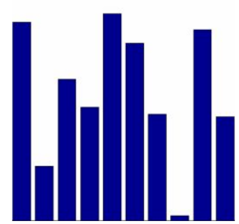
J



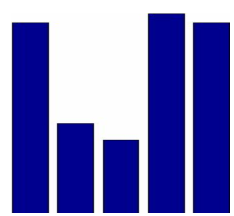
n clusters

As much earth as possible must be moved

I’



J’



$$\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} = \min\left(\sum_{i=1}^{m} w_{I_i}, \sum_{j=1}^{n} w_{J_j}\right)$$

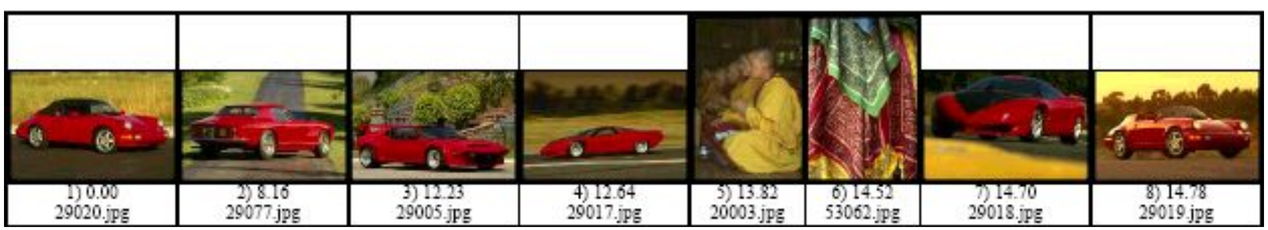
Slides credit on EMD : Frederik Heger

# Color-based image retrieval


L1 distance


Jeffrey divergence


$\chi^2$ statistics




Earth Mover Distance

*Y. Rubner, J. Puzicha, C. Tomasi and T.M. Buhmann*, ***"Empirical Evaluation of Dissimilarity Measures for Color and Texture"***, CVIU'2001

Slides credit on EMD : Frederik Heger

# Outline

- Distance measures
  - Euclidean distances (L2) distances
  - L1 and Lp distances
  - Chi-square distances
  - Kullback-Liebler divergence
  - Earth mover distances

- Similarity measures
  - Cosine similarity
  - Histogram intersection

# Histogram intersection

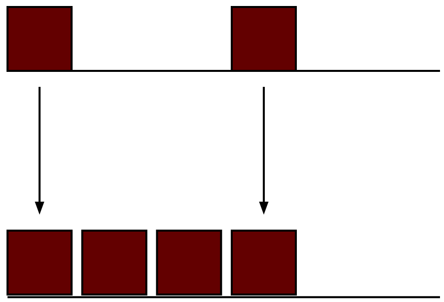
- Measuring how much overlap two histograms have

$$S(I,J) = \sum_i \min(H_I(i), H_J(i))$$

- It defines a proper kernel function…..

# Issues with EMD

- High computational complexity
  - Prohibitive for texture segmentation
- Features ordering needs to be known
  - Open eyes / closed eyes example
- Distance can be set by very few features.
  - E.g. with partial match of uneven distribution weight



EMD = 0, no matter how
many features follow

Slides credit on EMD : Frederik Heger

# Summary

- Color, texture, descriptors
  - Color histogram
  - Color correlogram
  - LBP descriptors
  - Histogram of oriented gradient
  - Spatial pyramid matching
- Distance & Similarity measure
  - $L_p$ distances
  - Chi-Square distances
  - KL distances
  - EMD distances
  - Histogram intersection