Computer Vision

CSE/EE 576
Content-Based Image Retrieval and the EM Algorithm

Linda Shapiro
Professor of Computer Science & Engineering
Professor of Electrical & Computer Engineering

Content-Based Image Retrieval

- Queries
- Commercial Systems
- Retrieval Features
- Indexing in the FIDS System
- Lead-in to Object Recognition

Content-based Image Retrieval (CBIR)

Searching a large database for images that *match* a query:

- What kinds of databases?
- What kinds of queries?
- What constitutes a match?
- How do we make such searches efficient?

Applications

– Art Collections

  e.g. Fine Arts Museum of San Francisco

– Medical Image Databases

  CT, MRI, Ultrasound, The Visible Human

– Scientific Databases

  e.g. Earth Sciences

– General Image Collections for Licensing

  Corbis, Getty Images

– The World Wide Web

  Google, Microsoft, etc

What is a query?

– an image you already have

– a rough sketch you draw

– a symbolic description of what you want
e.g. an image of a man and a woman on a beach

Some Systems You Can Try

- Corbis ~~sells~~ sold high-quality images for use in advertising, marketing, illustrating, etc. Corbis was sold to a Chinese company, but

- Getty images now provides the image sales.

- http://www.gettyimages.com/search/2/image?excludenudity=true&sort=best

Google Image

- Google Images
http://www.google.com/imghp

Try the camera icon.

Microsoft Bing

- http://www.bing.com/

Problem with Text-Based Search

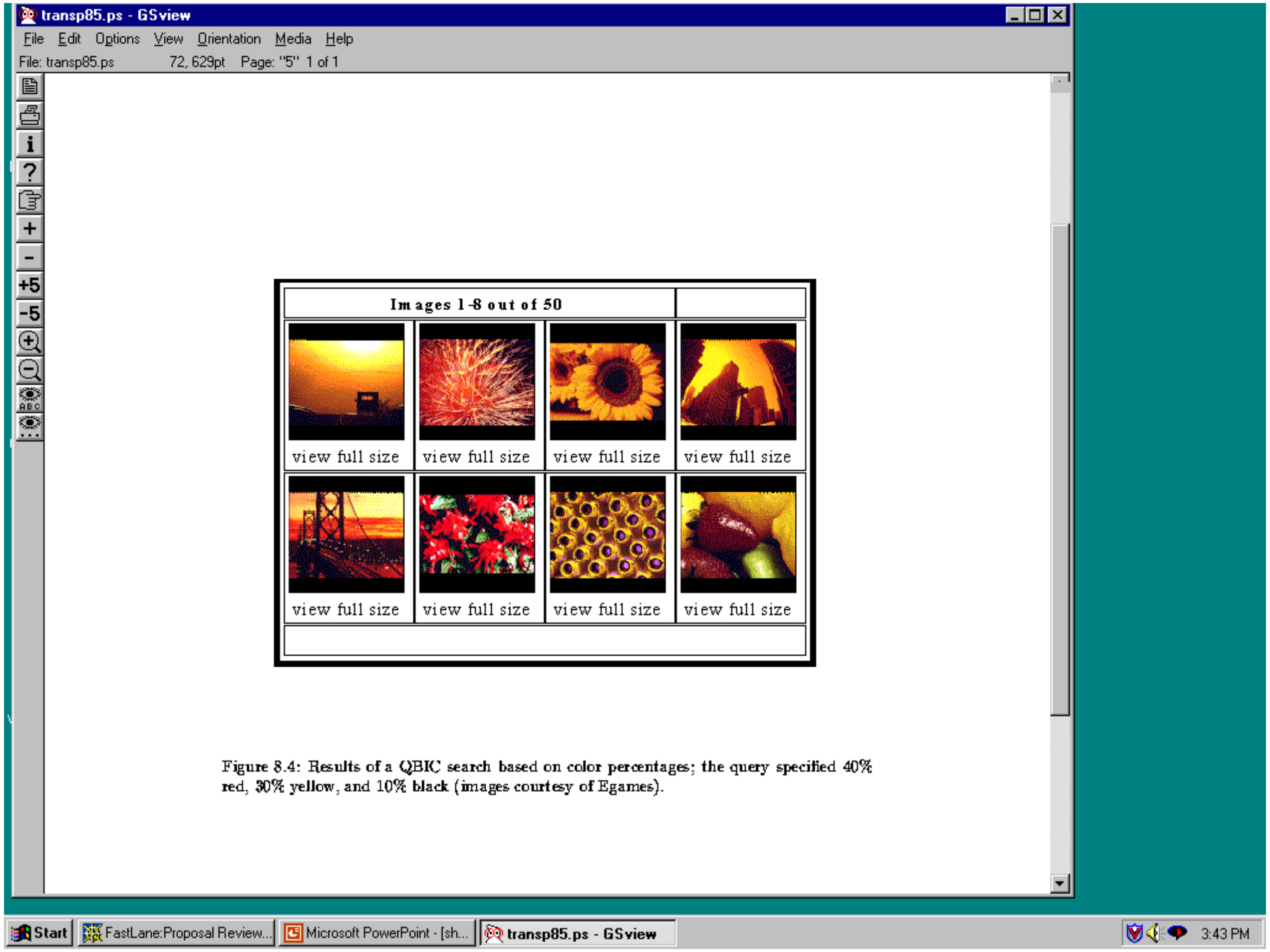
- Retrieval for pigs for the color chapter of my book
- Small company (was called Ditto)
- Allows you to search for pictures from web pages

Features

- Color (histograms, gridded layout, wavelets)
- Texture (Laws, Gabor filters, local binary pattern)
- Shape (first segment the image, then use statistical or structural shape similarity measures)
- Objects and their Relationships

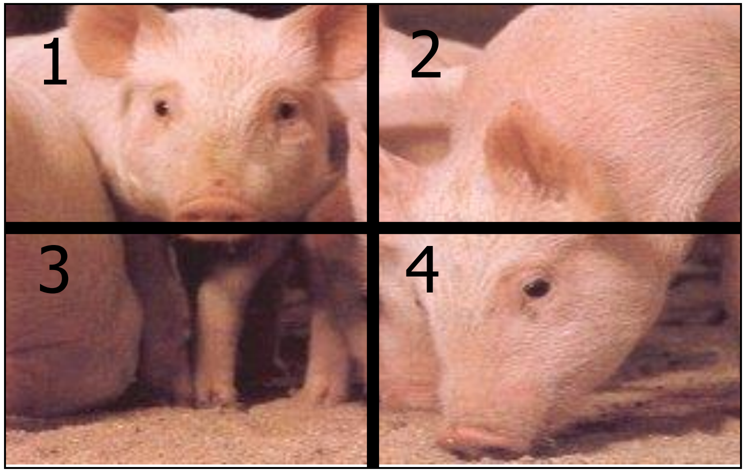
This is the most powerful, but you have to be able to recognize the objects!

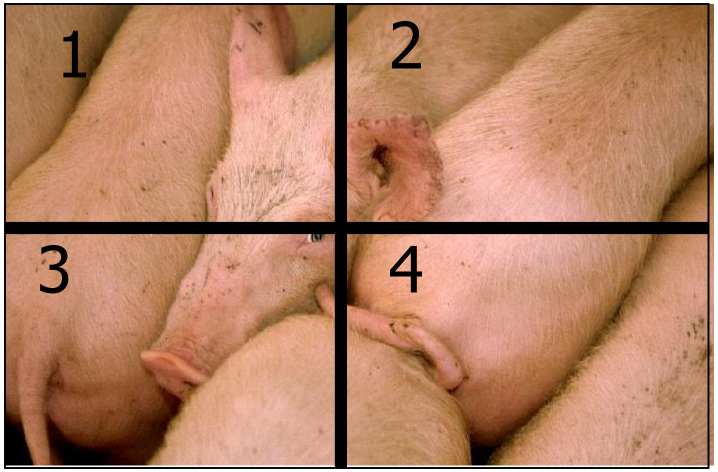
Color Histograms



Gridded Color

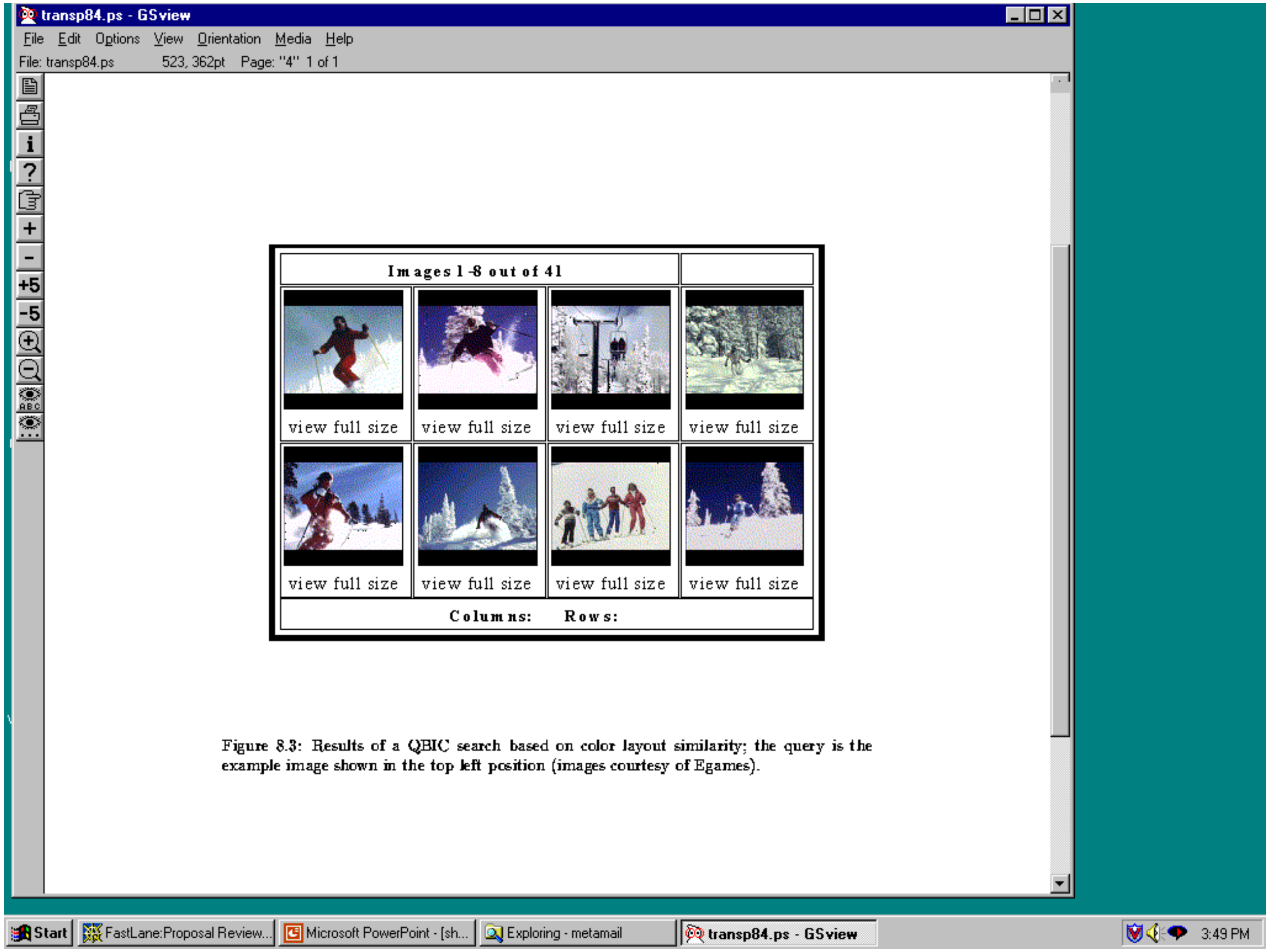
Gridded color distance is the sum of the color distances in each of the corresponding grid squares.





What color distance would you use for a pair of grid squares?

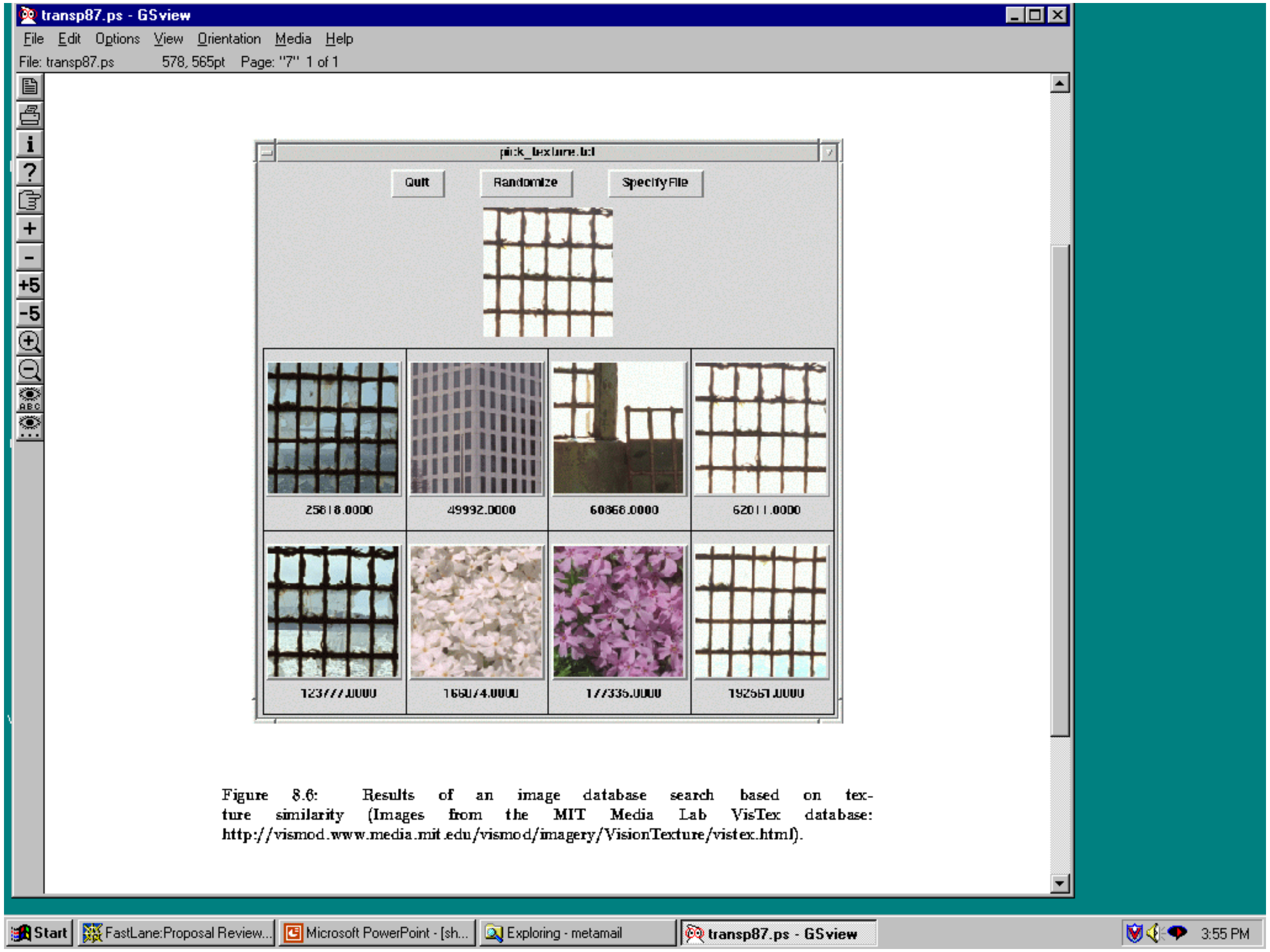
Color Layout
(IBM's Gridded Color)



Texture Distances

- Pick and Click (user clicks on a pixel and system retrieves images that have in them a region with similar texture to the region surrounding it.
- Gridded (just like gridded color, but use texture).
- Histogram-based (e.g. compare the LBP histograms).

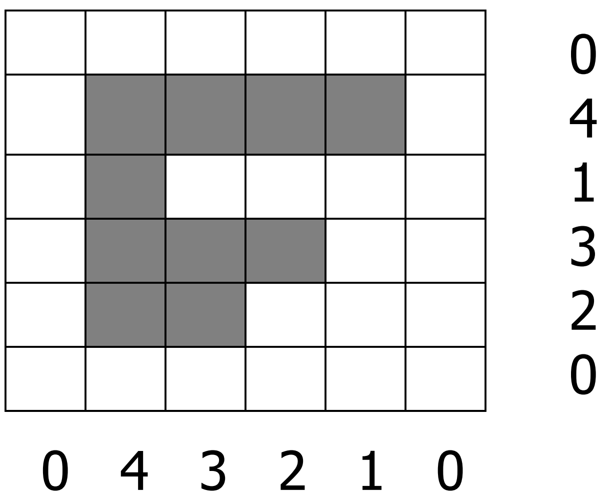
Laws Texture



Shape Distances

- Shape goes one step further than color and texture.
- It requires identification of regions to compare.
- There have been many shape similarity measures suggested for pattern recognition that can be used to construct shape distance measures.

Global Shape Properties: Projection Matching

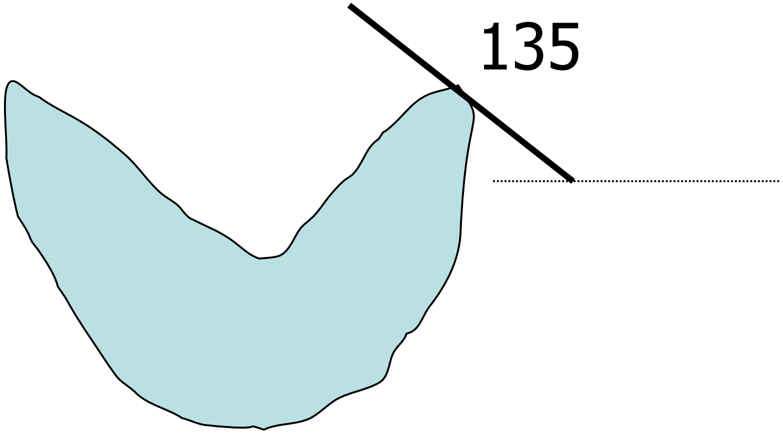


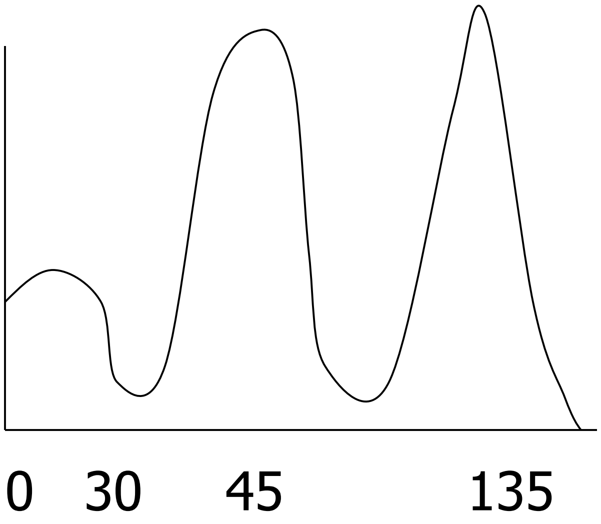
Feature Vector
(0,4,1,3,2,0,0,4,3,2,1,0)

In projection matching, the horizontal and vertical projections form a histogram.

What are the weaknesses of this method? strengths?

Global Shape Properties: Tangent-Angle Histograms





Is this feature invariant to starting point?
Is it invariant to size, translation, rotation?

Boundary Matching

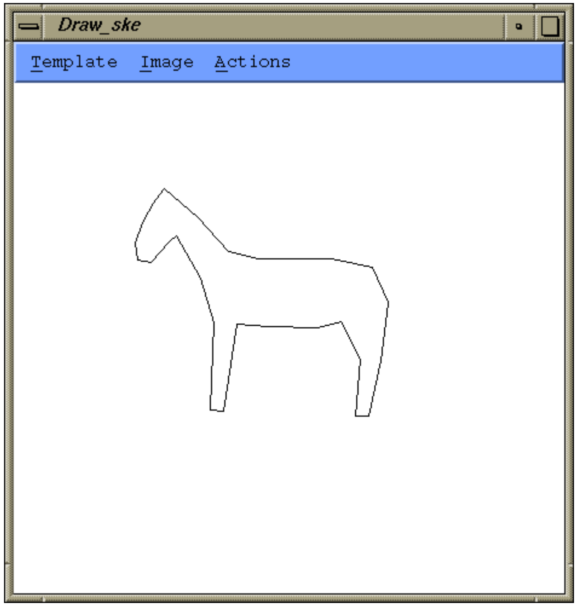
- Fourier Descriptors
- Sides and Angles
- Elastic Matching

  The distance between query shape and image shape has two components:

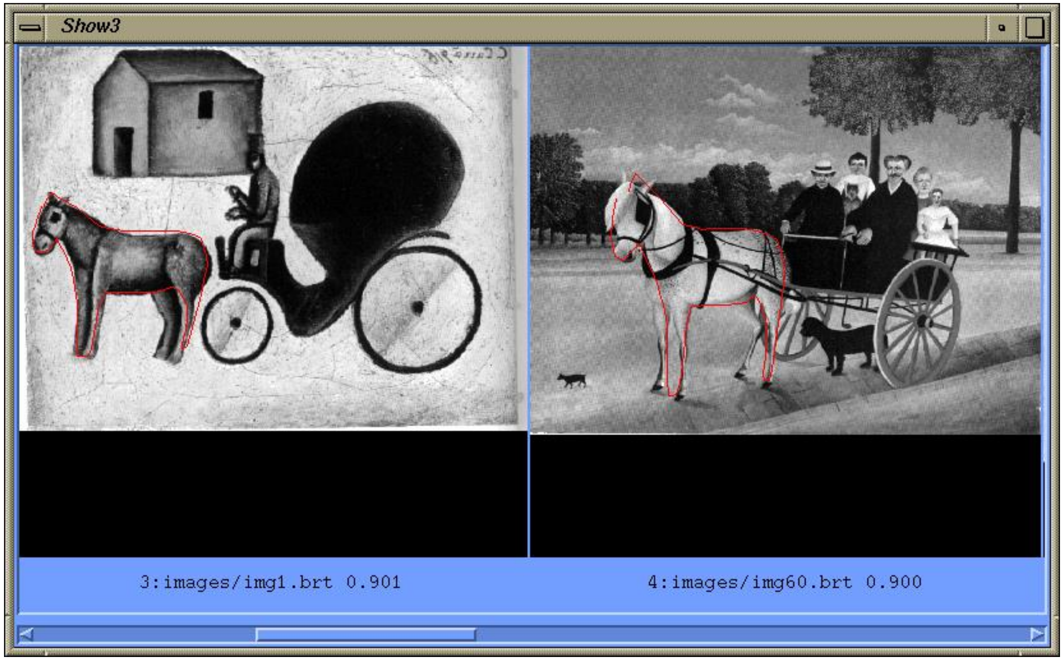
  1. energy required to deform the query shape into one that best matches the image shape

  2. a measure of how well the deformed query matches the image

Del Bimbo Elastic Shape Matching



query



retrieved images

Regions and Relationships

- Segment the image into regions
- Find their properties and interrelationships
- Construct a graph representation with nodes for regions and edges for spatial relationships
- Use graph matching to compare images

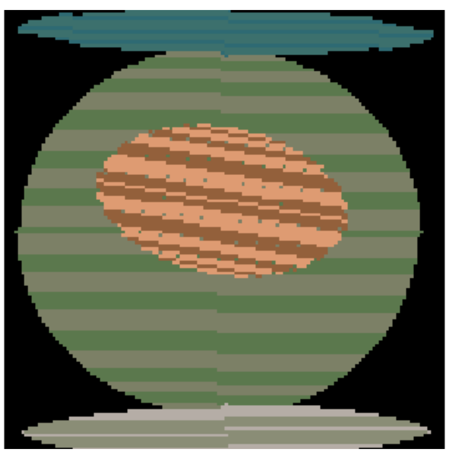
Like what?

Blobworld (Carson et al, 1999)

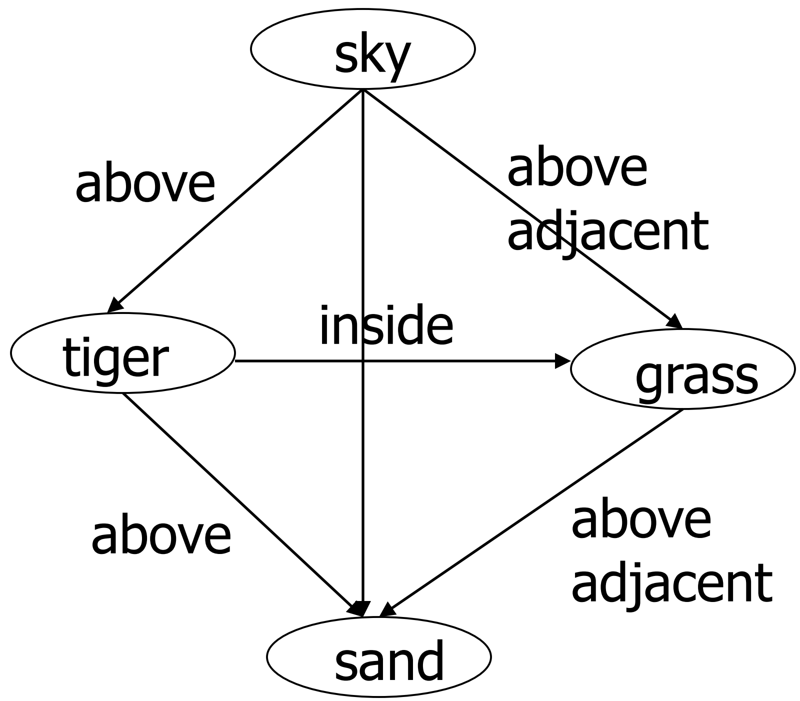
- Segmented the query (and all database images) using EM on color+texture
- Allowed users to select the most important region and what characteristics of it (color, texture, location)
- Asked users if the background was also important

Tiger Image as a Graph (motivated by Blobworld)

image



abstract regions

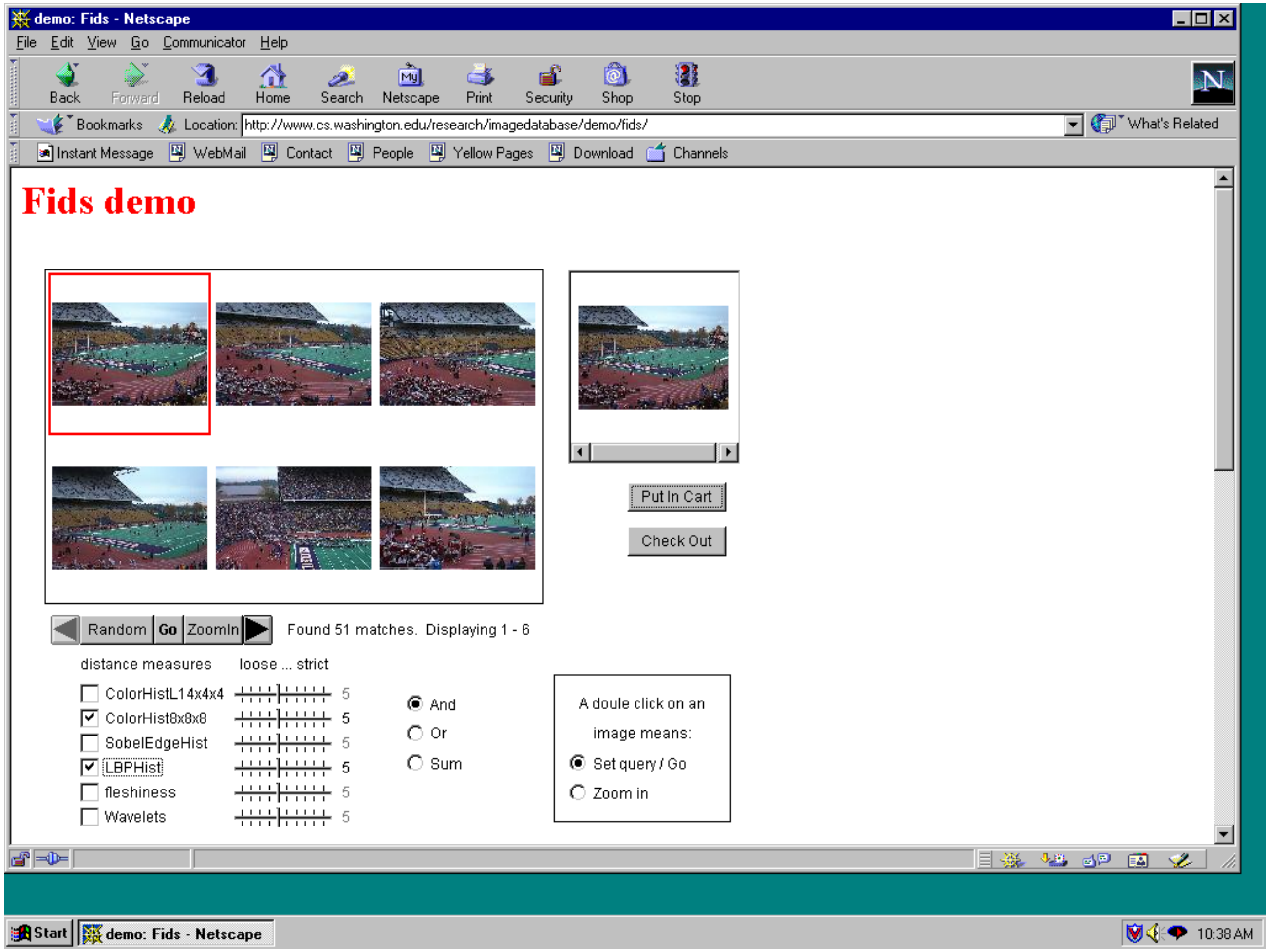


Andy Berman's FIDS System

multiple distance measures

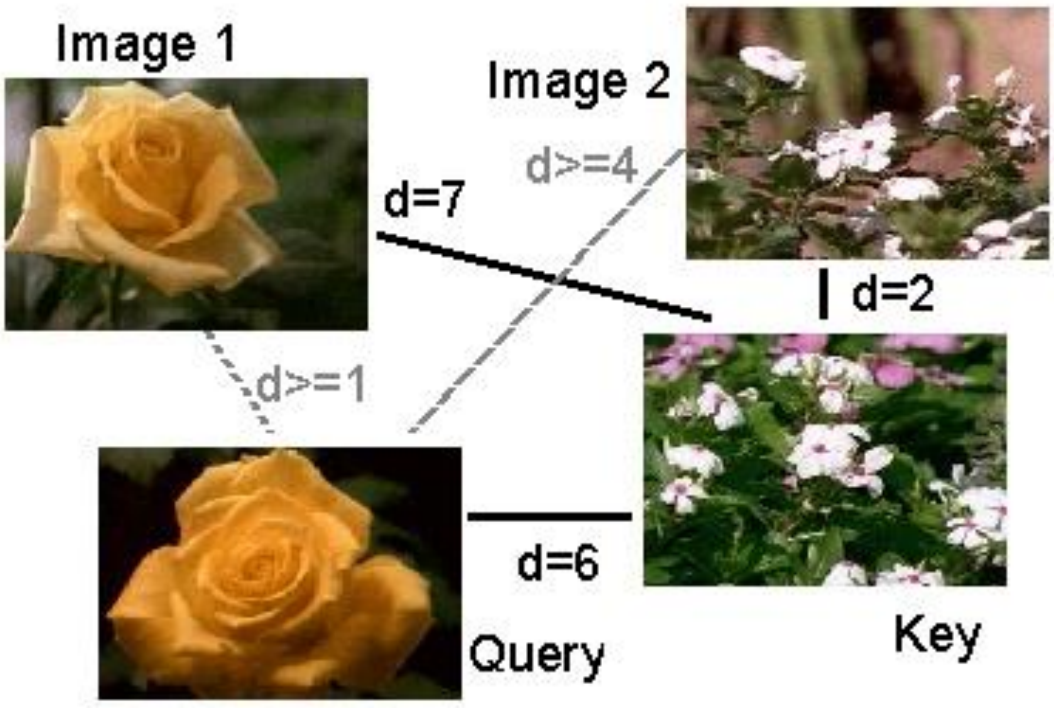
Boolean and linear combinations

efficient indexing using images as keys



Andy Berman's FIDS System:

Use of key images and the triangle inequality for efficient retrieval. d(I,Q) >= |d((I,K) – d(Q,K)|



Andy Berman's FIDS System:

Bare-Bones Triangle Inequality Algorithm

Offline

1. Choose a small set of key images
2. Store distances from database images to keys

Online (given query Q)

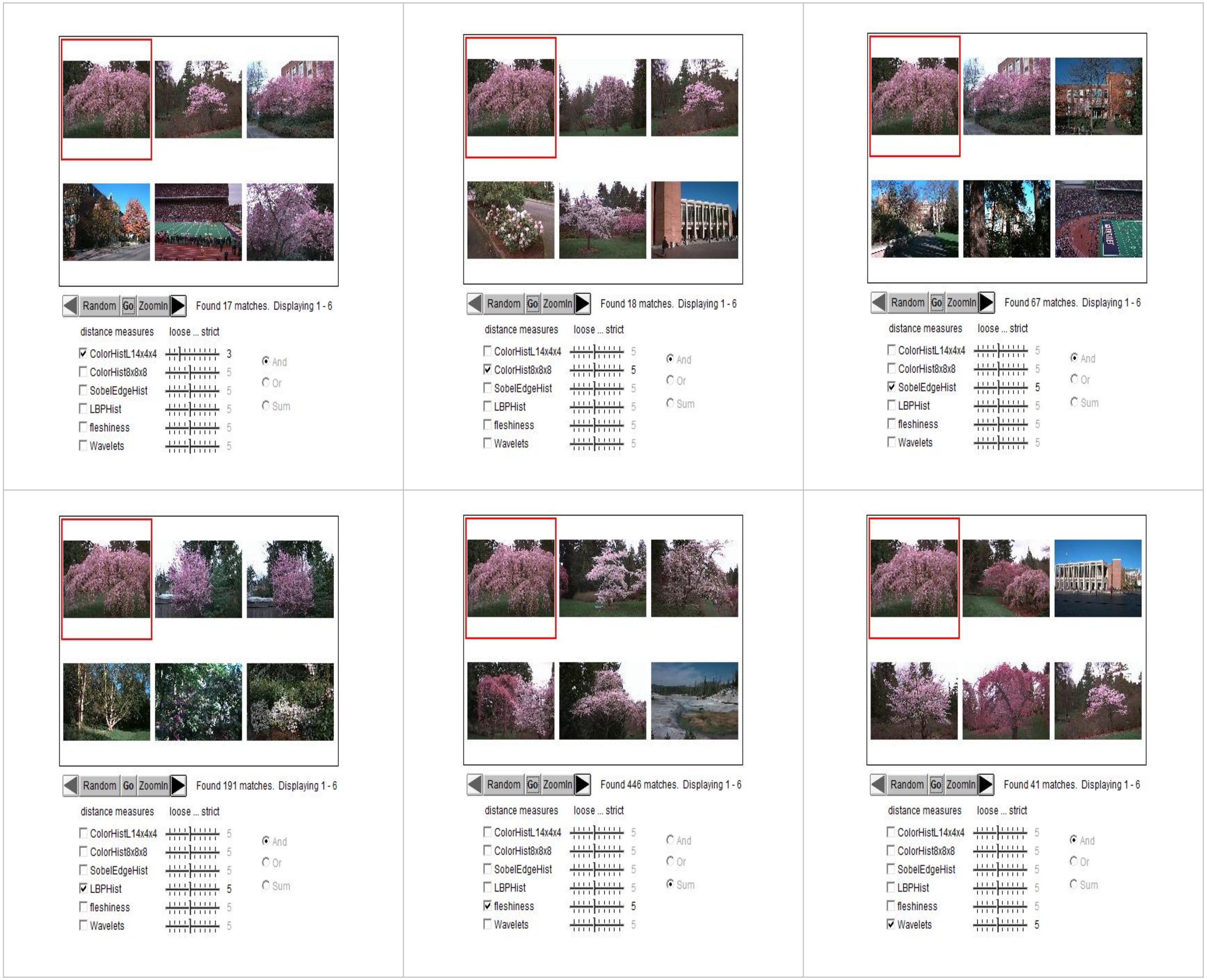
1. Compute the distance from Q to each key
2. Obtain lower bounds on distances to database images
3. Threshold or return all images in order of lower bounds

Andy Berman’s FIDS System:

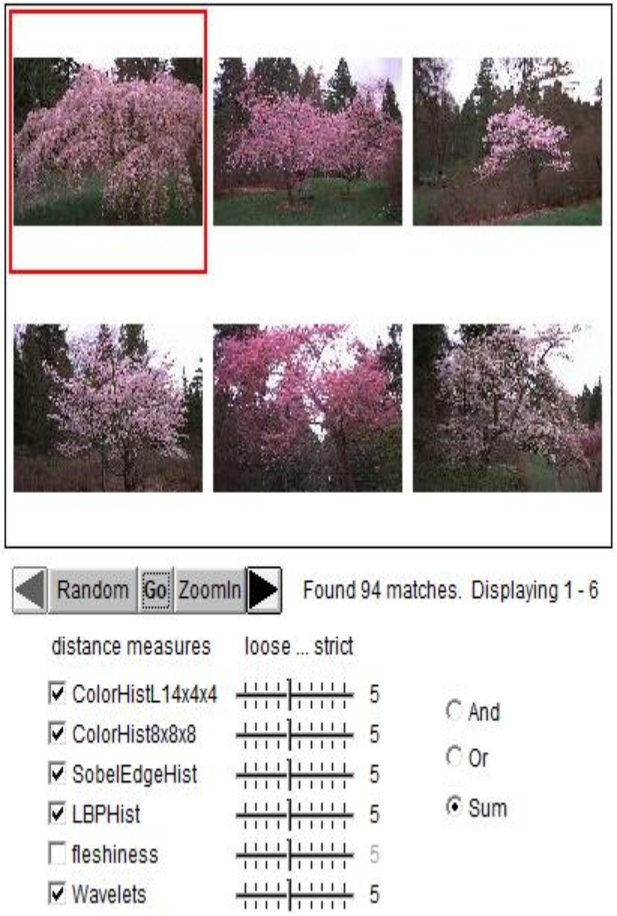
Flexible Image Database System: Example

An example from our system using a simple color measure.
images in system: 37,748
threshold: 100 out of 1000
images eliminated: 37,729

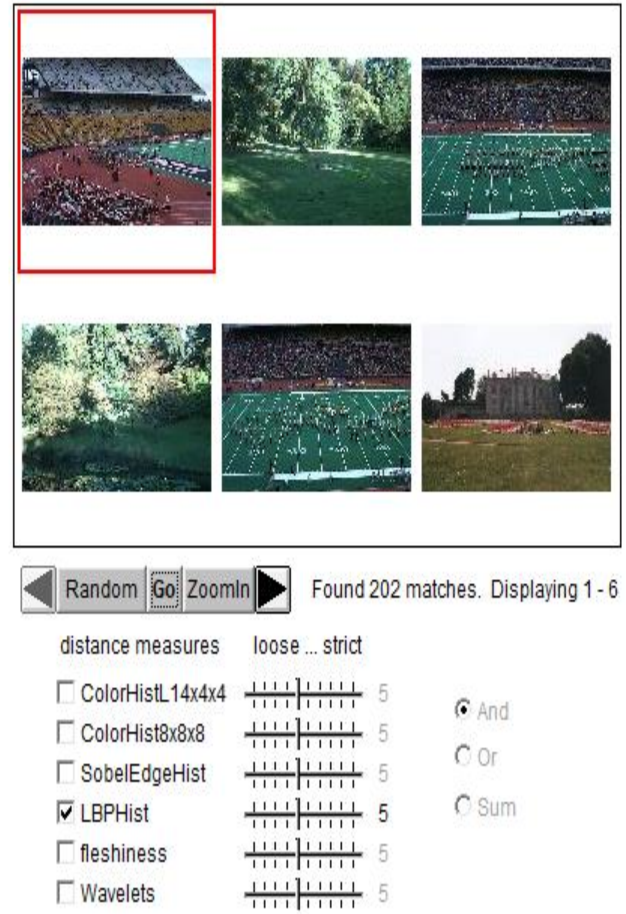
Different Features



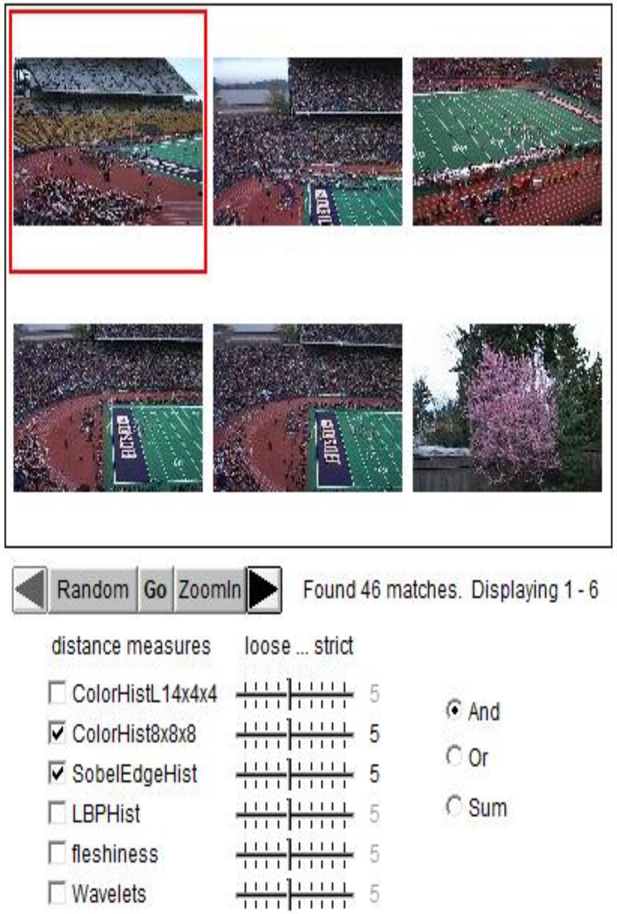
Combined Features

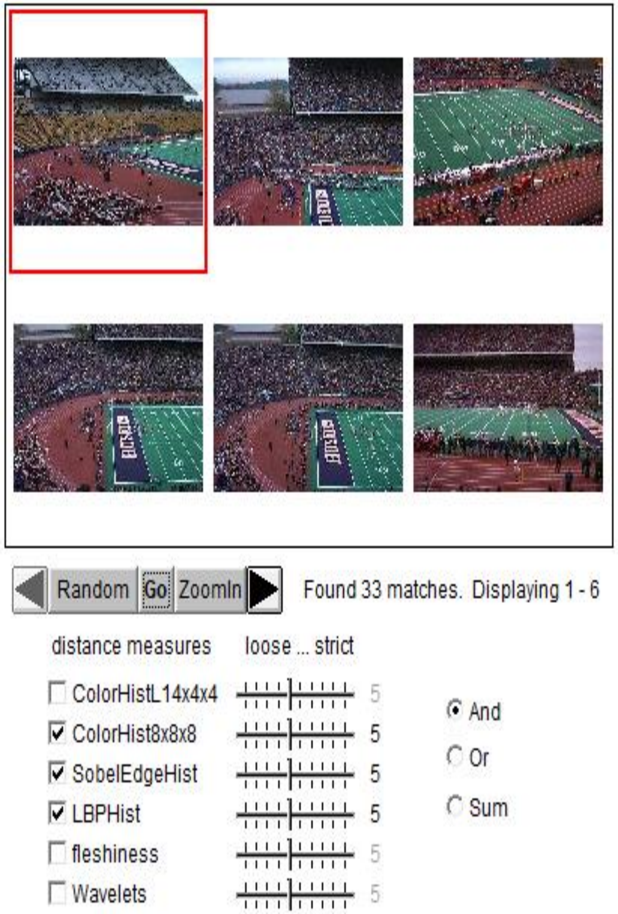


Another example: different features

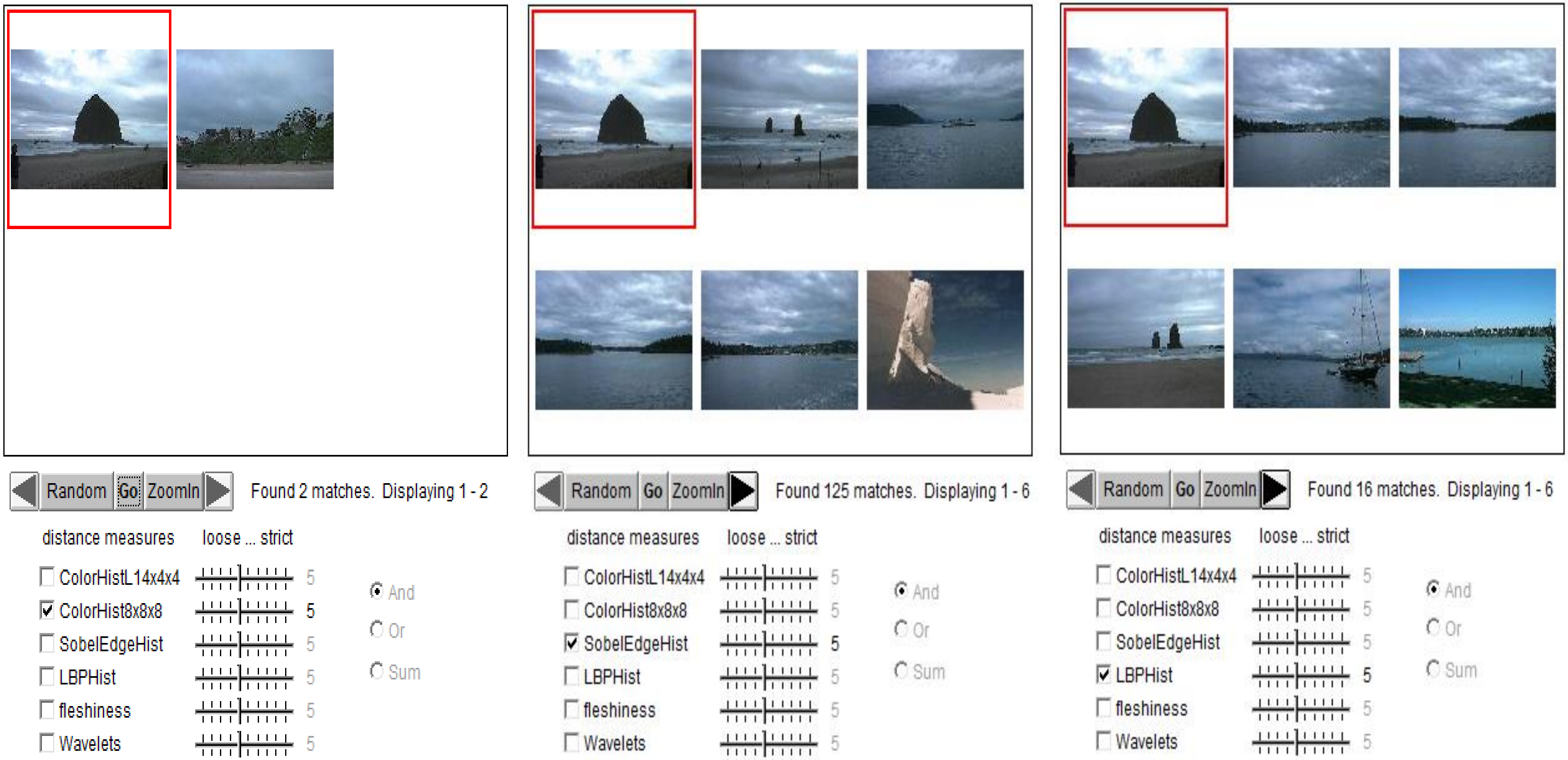


Combined Features

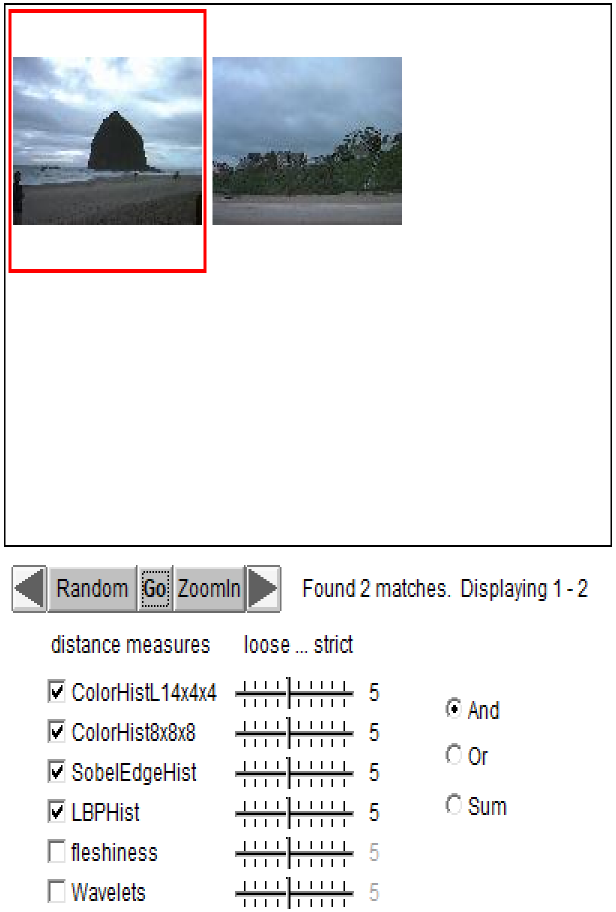


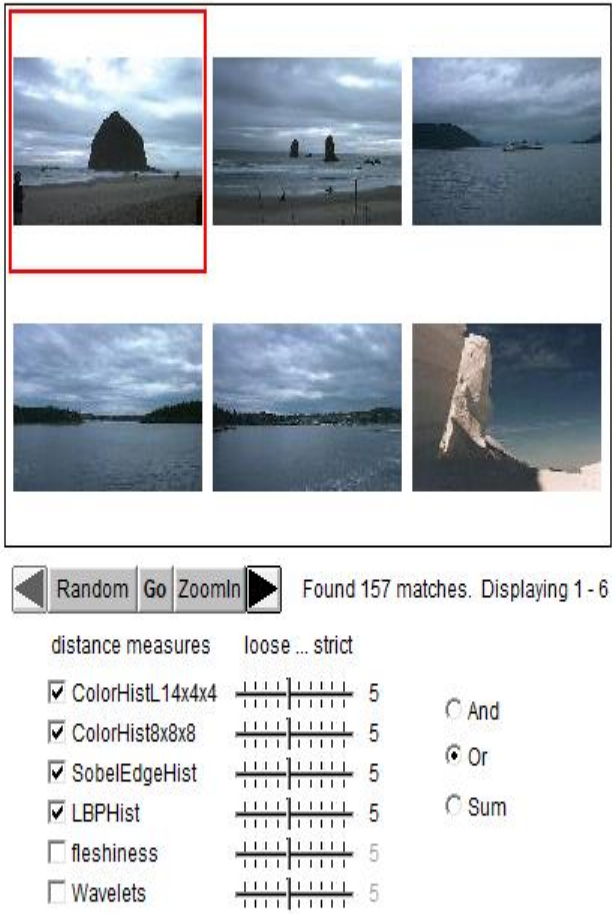


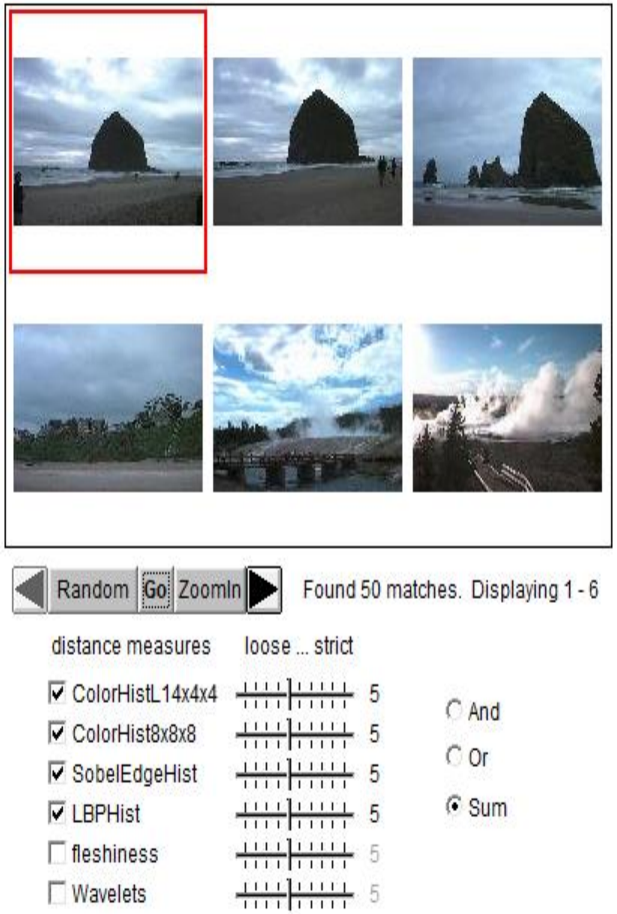
Another example: different features



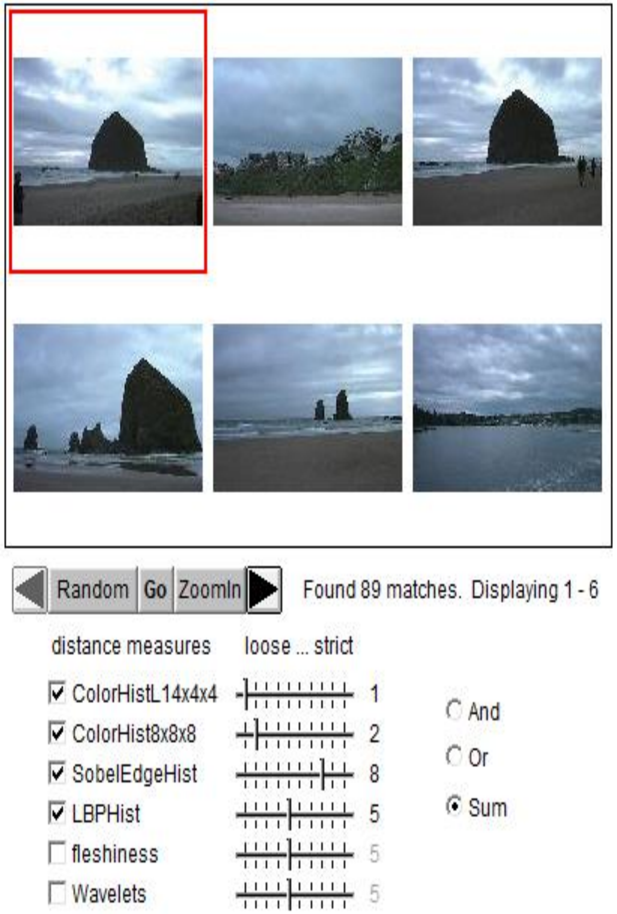
Different ways for combination



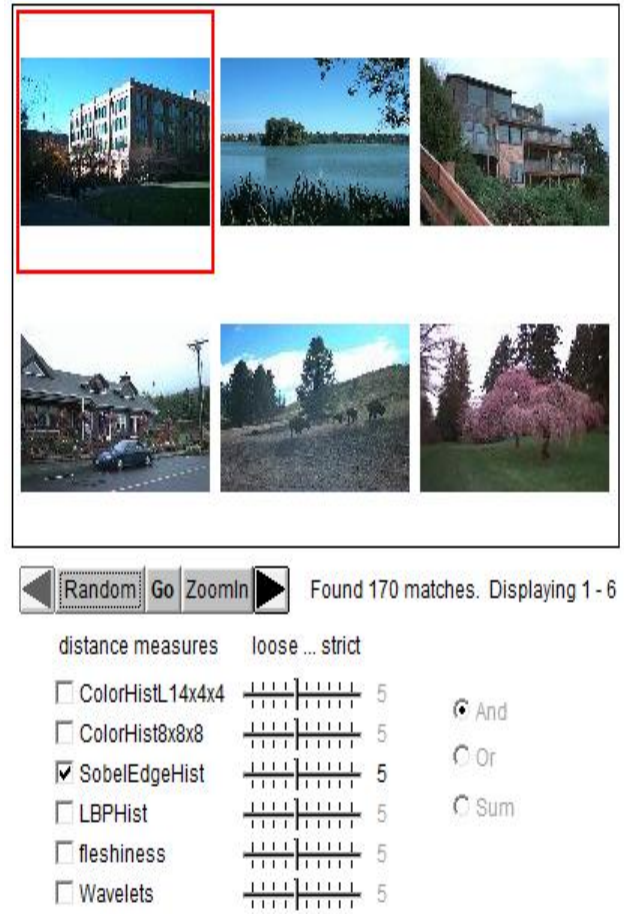


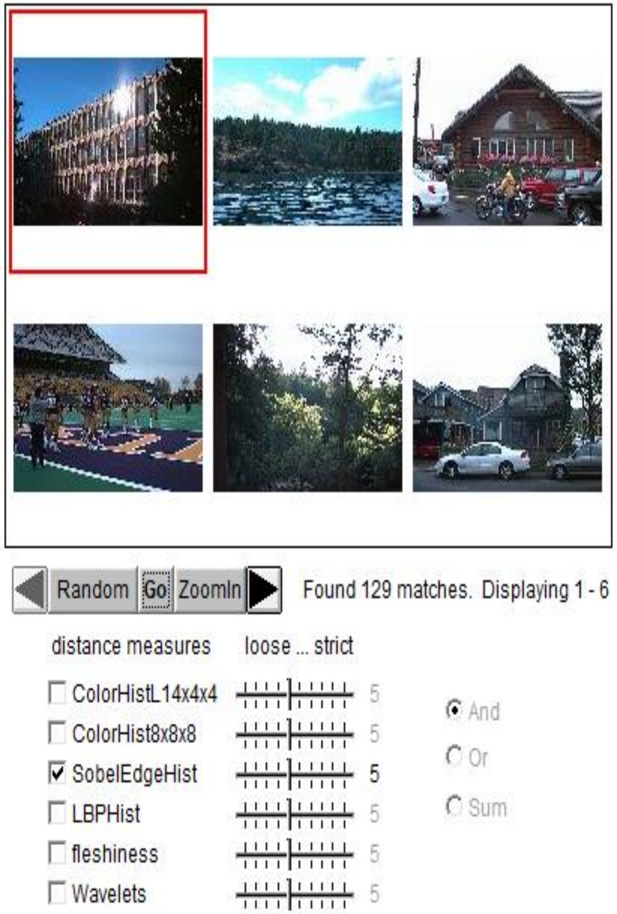


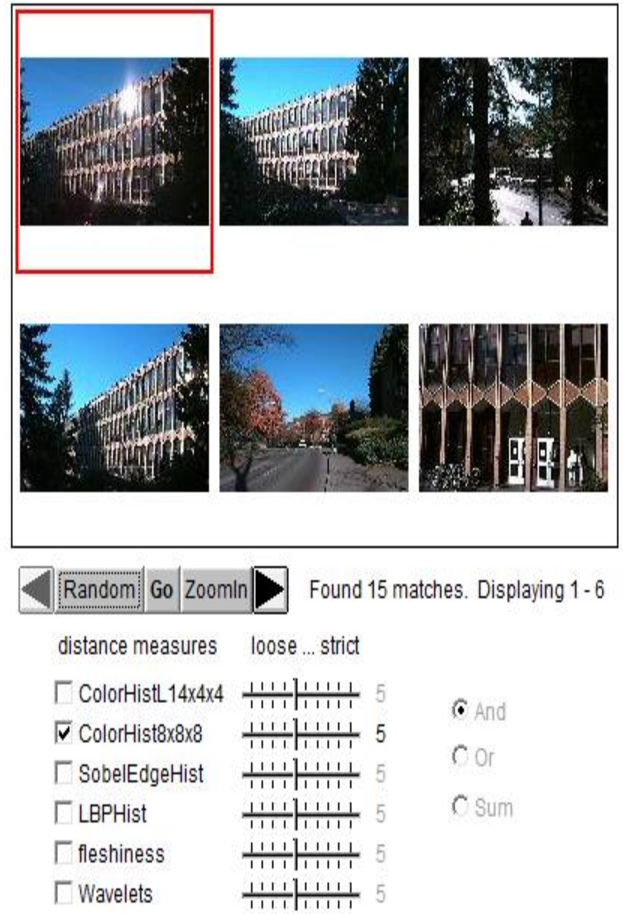
Different weights on features




Random
Go
ZoomIn
Found 170 matches. Displaying 1 - 6
distance measures
loose ... strict
ColorHistL14x4x4
5
ColorHist8x8x8
5
SobelEdgeHist
5
LBPHist
5
fleshiness
5
Wavelets
5
And
Or
Sum


Random
Go
ZoomIn
Found 170 matches. Displaying 1 - 6
distance measures
loose ... strict
ColorHistL14x4x4
5
ColorHist8x8x8
5
SobelEdgeHist
5
LBPHist
5
fleshiness
5
Wavelets
5
And
Or
Sum


Random
Go
ZoomIn
Found 129 matches. Displaying 1 - 6
distance measures
loose ... strict
ColorHistL14x4x4
5
ColorHist8x8x8
5
SobelEdgeHist
5
LBPHist
5
fleshiness
5
Wavelets
5
And
Or
Sum


Random
Go
ZoomIn
Found 15 matches. Displaying 1 - 6
distance measures
loose ... strict
ColorHistL14x4x4
5
ColorHist8x8x8
5
SobelEdgeHist
5
LBPHist
5
fleshiness
5
Wavelets
5
And
Or
Sum

Weakness of Low-level Features

- Can't capture the high-level concepts



Yi Li's Overall Approach

- Develop object recognizers for common objects
- Use these recognizers to design a new set of both low- and mid-level features
- Design a learning system that can use these features to recognize classes of objects

Building Features: Consistent Line Clusters (CLC)

A **Consistent Line Cluster** is a set of lines that are homogeneous in terms of some line features.

- **Color-CLC**: The lines have the same color feature.
- **Orientation-CLC**: The lines are parallel to each other or converge to a common vanishing point.
- **Spatially-CLC**: The lines are in close proximity to each other.

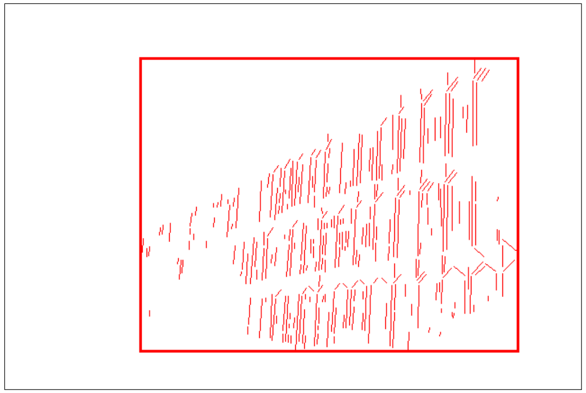
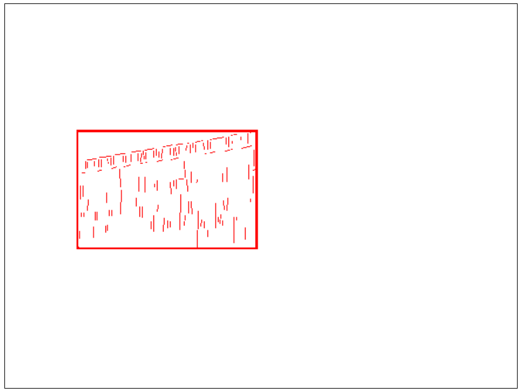
Experimental Evaluation

- Object Recognition
  - 97 well-patterned buildings (bp): 97/97
  - 44 not well-patterned buildings (bnp): 42/44
  - 16 not patterned non-buildings (nbnp): 15/16 (one false positive)
  - 25 patterned non-buildings (nbp): 0/25
- CBIR

Experimental Evaluation

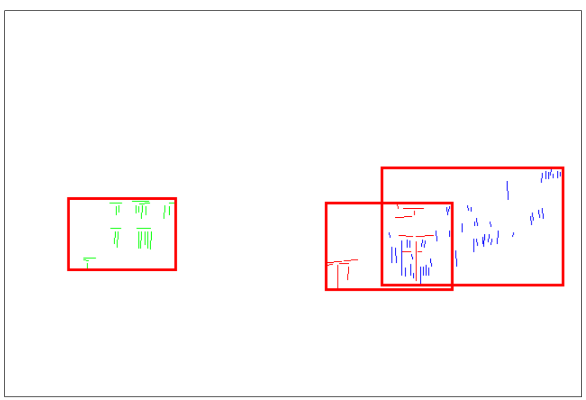
Well-Patterned Buildings

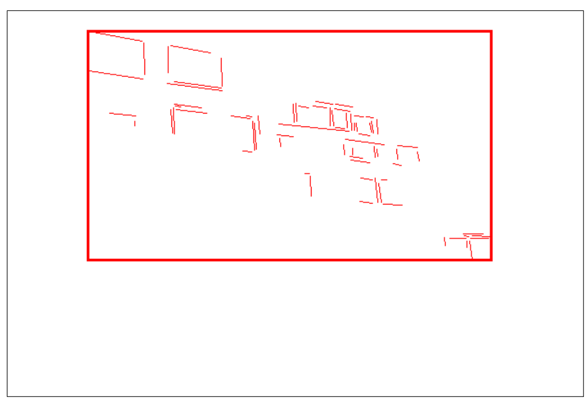



Experimental Evaluation

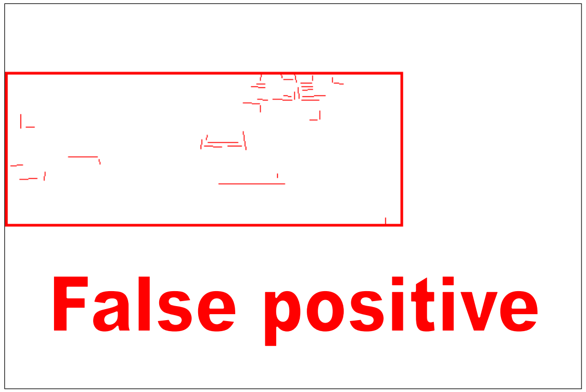
Non-Well-Patterned Buildings





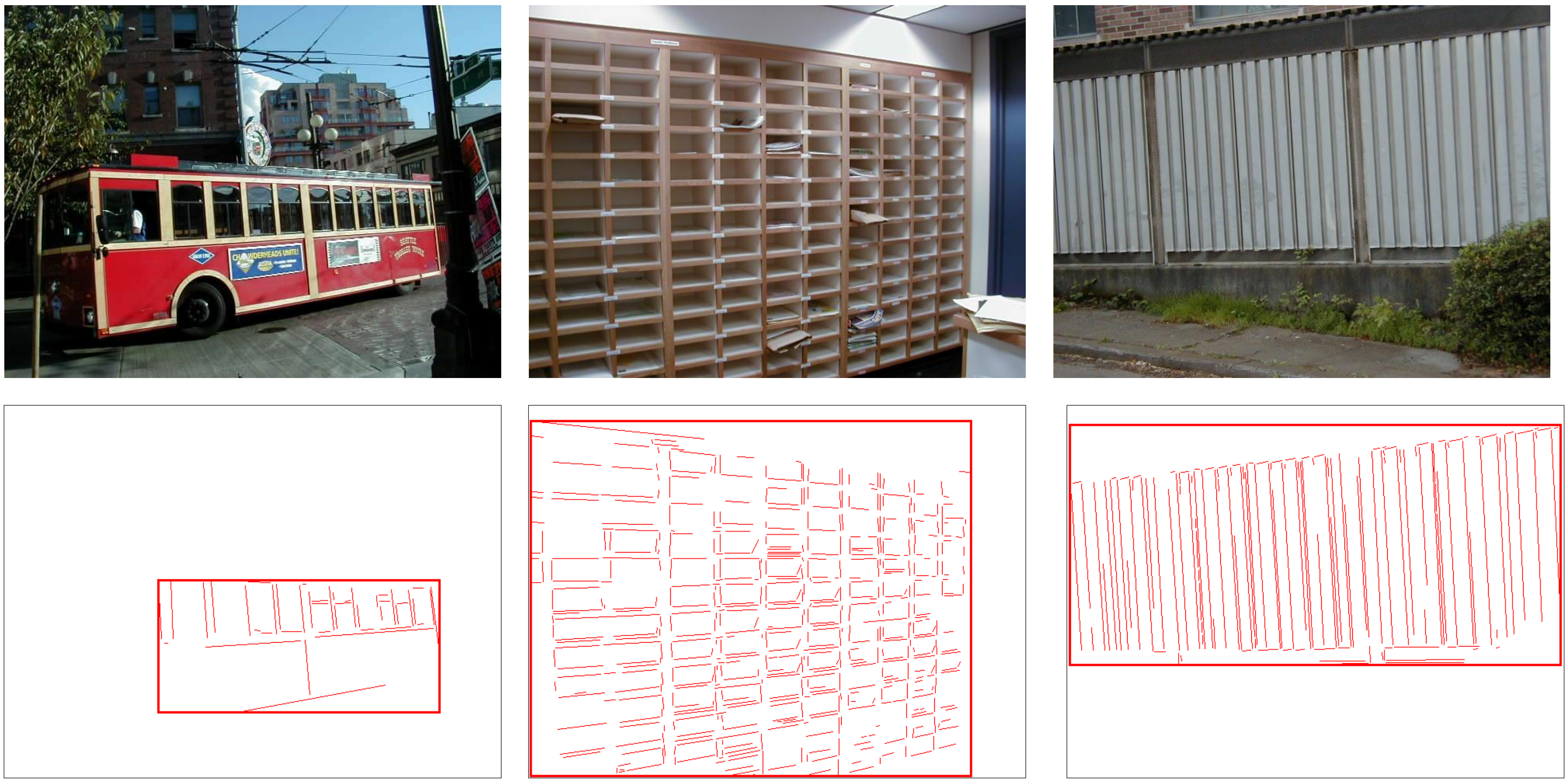
Experimental Evaluation

Non-Well-Patterned Non-Buildings



Experimental Evaluation

Well-Patterned Non-Buildings (false positives)



Experimental Evaluation (CBIR)

	Total Positive Classification (#)	Total Negative Classification (#)	False positive (#)	False negative (#)	Accuracy (%)
Arborgreens	0	47	0	0	100
Campusinfall	27	21	0	5	89.6
Cannonbeach	30	18	0	6	87.5
Yellowstone	4	44	4	0	91.7

Experimental Evaluation (CBIR)

False positives from Yellowstone




Machine Learning!

- Unsupervised (given the data, no class labels)
- Supervised (given data with class labels)
- We will look at two unsupervised methods today
  - K-means
  - EM
- We saw that EM was used in Rob Fergus's work.

Clustering

- There are K clusters $C_1, \ldots, C_K$ with means $m_1, \ldots, m_K$.

- The **least-squares error** is defined as

$$D = \sum_{k=1}^{K} \sum_{x_i \in C_k} \| x_i - m_k \|^2 .$$

- Out of all possible partitions into K clusters, choose the one that minimizes D.

Why don't we just do this?
If we could, would we get meaningful objects?

K-Means Clustering

Form K-means clusters from a set of n-dimensional vectors

1. Set ic (iteration count) to 1

2. Choose randomly a set of K means $m_1(1)$, …, $m_K(1)$.

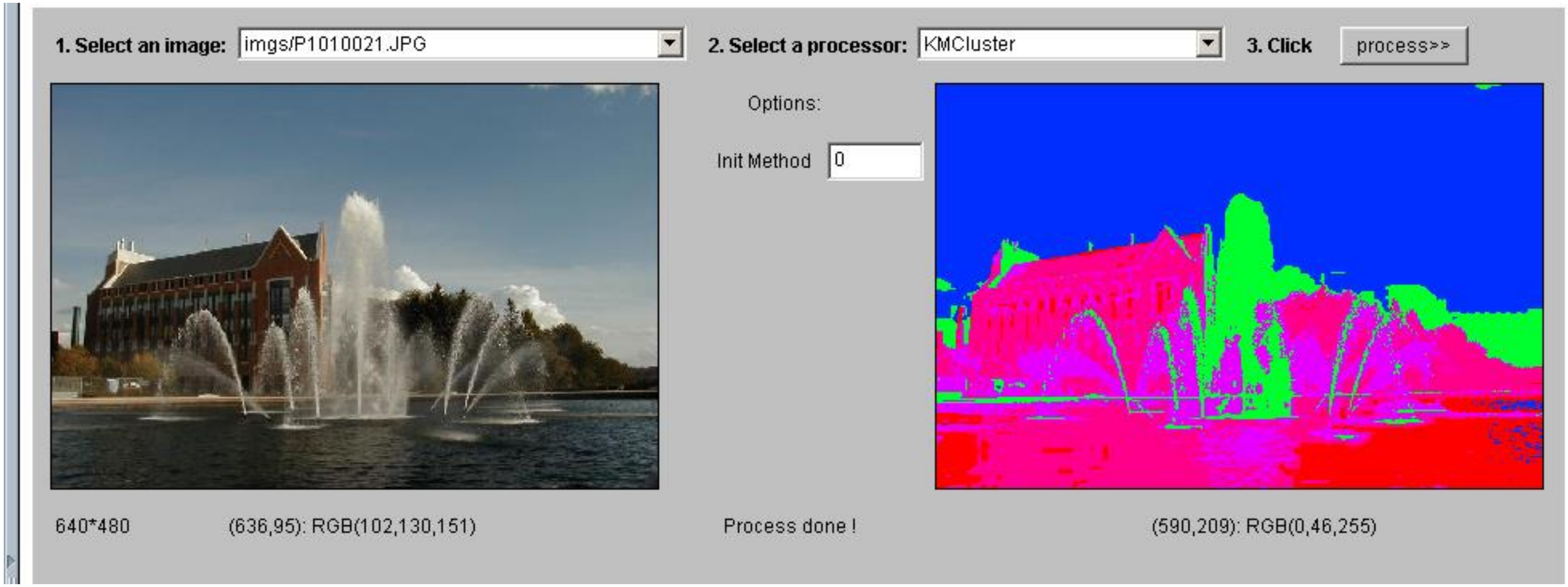
3. For each vector x_i compute $D(x_i , m_k(ic))$, k=1,…K
   and assign x_i to the cluster C_j with nearest mean.

4. Increment ic by 1, update the means to get $m_1(ic)$,…,$m_K(ic)$.

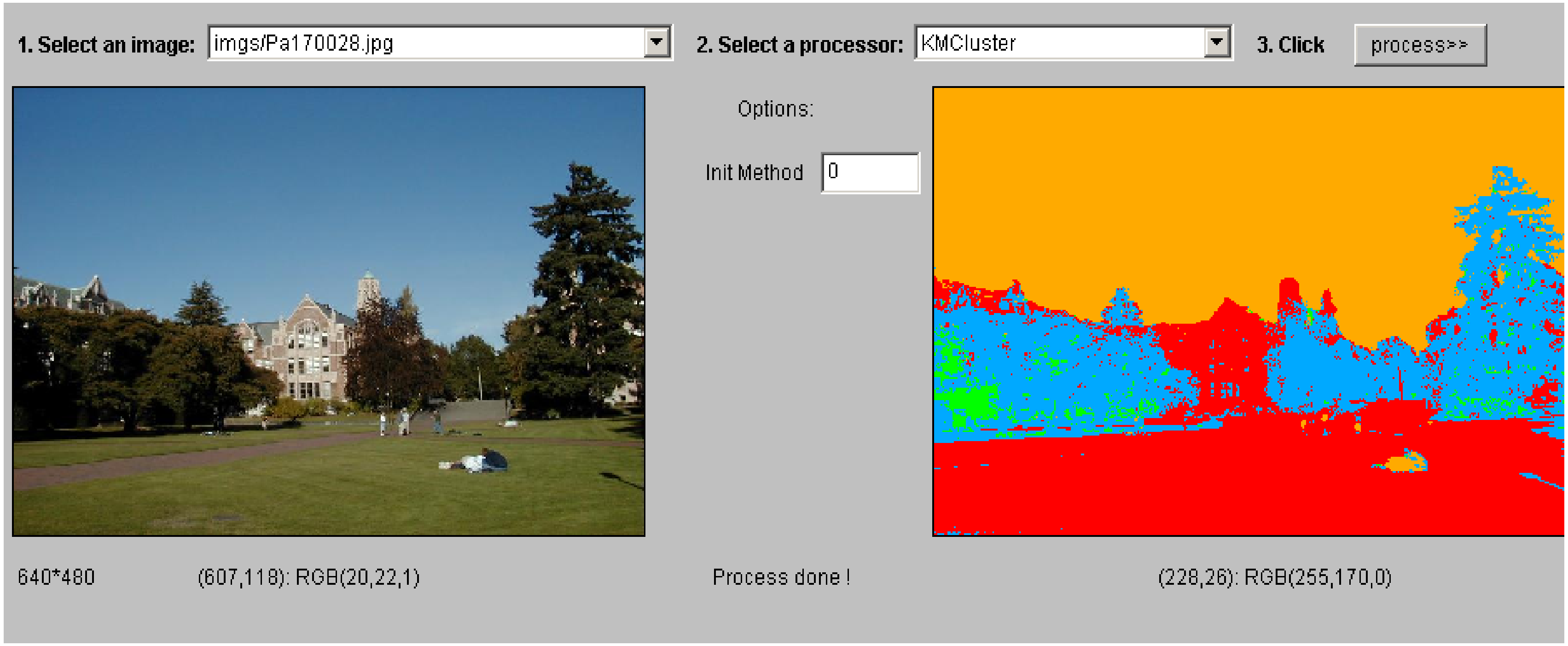
5. Repeat steps 3 and 4 until $C_k(ic) = C_k(ic+1)$ for all k.

K-Means Example 1

K-Means Example 2



K-Means Example 3



K-means Variants

- Different ways to initialize the means
- Different stopping criteria
- Dynamic methods for determining the right number of clusters (K) for a given image

- The EM Algorithm: a probabilistic formulation

K-Means

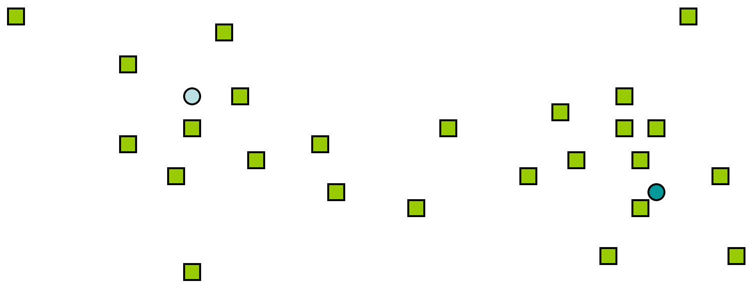
- Boot Step:
  - Initialize K clusters: $C_1, \ldots, C_K$
    Each cluster is represented by its mean m_j
- Iteration Step:
  - Estimate the cluster for each data point

$$x_i \Longrightarrow C(x_i)$$

  - Re-estimate the cluster parameters

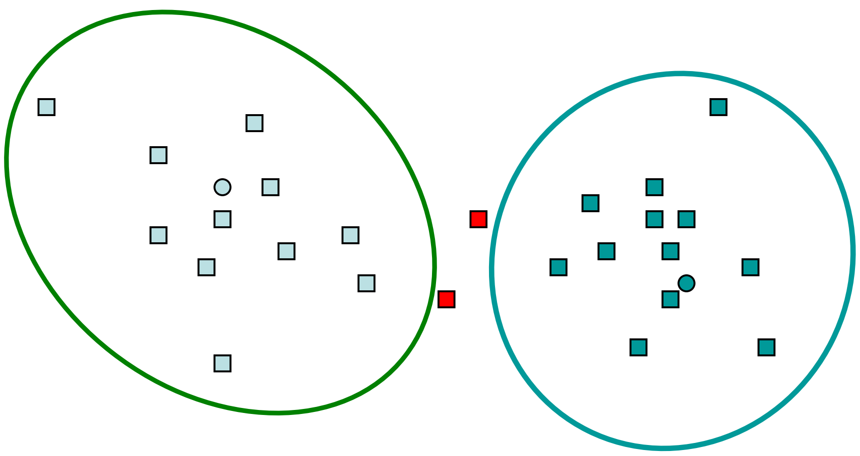
$$m_j = mean\{x_i \mid x_i \in C_j\}$$

K-Means Example



K-Means Example

Where do the red points belong?



K-Means $\rightarrow$ EM

- Boot Step:
  - Initialize K clusters: $C_1, \ldots, C_K$

    (μ_j, Σ_j) and $P(C_j)$ for each cluster j.
- Iteration Step:
  - Estimate the cluster of each data point

    $p(C_j \mid x_i)$ ⇨ Expectation
  - Re-estimate the cluster parameters

    $(\mu_j, \Sigma_j), p(C_j)$ For each cluster j ⇨ Maximization

What is a covariance matrix

- For a multidimensional distribution of n dimensions (X_1, X_2, … X_n):
- Its mean μ is a vector $\mu = (x_1, x_2, \ldots x_n)$

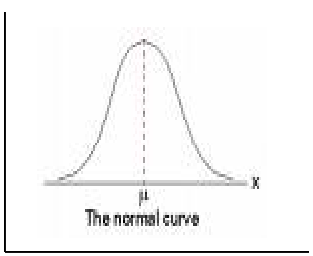
  Example: $\mu = (r, g, b)$
- Its covariance matrix gives the variances and covariances for pairs of variables:

  Σ = a matrix in which $\Sigma_{ii} = \sigma_i^2$ (variance)

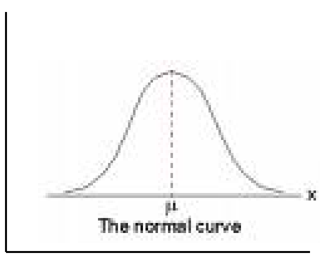
  and $\Sigma_{ij} = \mathrm{Cov}(X_i, X_j)$ (covariance of two)

1-D EM with Gaussian Distributions

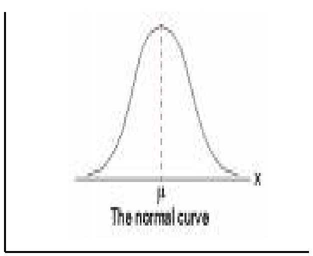
- Each cluster C_j is represented by a Gaussian distribution $N(\mu_j , \sigma_j)$.
- Initialization: For each cluster C_j initialize its mean μ_j , variance σ_j, and weight α_j.



$N(\mu_1 , \sigma_1)$
$\alpha_1 = P(C_1)$



$N(\mu_2 , \sigma_2)$
$\alpha_2 = P(C_2)$



$N(\mu_3 , \sigma_3)$
$\alpha_3 = P(C_3)$

Expectation

- For each point x_i and each cluster C_j compute $P(C_j \mid x_i)$.
- $P(C_j \mid x_i) = P(x_i \mid C_j)\, P(C_j) \,/\, P(x_i)$
- $P(x_i) = \sum_j P(x_i \mid C_j)\, P(C_j)$
- Where do we get $P(x_i \mid C_j)$ and $P(C_j)$?

1. Use the pdf for a normal distribution:

$$P(x_i \mid C_j) = \frac{1}{\sqrt{2\pi}\,\sigma_j} e^{-\frac{(x_i - \mu_j)^2}{2\sigma_j^2}}$$

2. Use $\alpha_j = P(C_j)$ from the current parameters of cluster C_j.

Maximization

- Having computed $P(C_j \mid x_i)$ for each point x_i and each cluster C_j, use them to compute new mean, variance, and weight for each cluster.

$$\mu_j = \frac{\sum_i p(C_j \mid x_i) \cdot x_i}{\sum_i p(C_j \mid x_i)}$$

$$\Sigma_j = \frac{\sum_i p(C_j \mid x_i) \cdot (x_i - \mu_j) \cdot (x_i - \mu_j)^T}{\sum_i p(C_j \mid x_i)}$$

$$p(C_j) = \frac{\sum_i p(C_j \mid x_i)}{N}$$

Multi-Dimensional Expectation Step for Color Image Segmentation

Input (Known)		Input (Estimation)		Output
$x_1=\{r_1, g_1, b_1\}$ $x_2=\{r_2, g_2, b_2\}$ … $x_i=\{r_i, g_i, b_i\}$ …	+	Cluster Parameters (μ_1,Σ_1), $p(C_1)$ for C_1 (μ_2,Σ_2), $p(C_2)$ for C_2 … (μ_k,Σ_k), $p(C_k)$ for C_k	⇨	Classification Results $p(C_1\|x_1)$ $p(C_j\|x_2)$ … $p(C_j\|x_i)$ …

$$p(C_j \mid x_i) = \frac{p(x_i \mid C_j)\cdot p(C_j)}{p(x_i)} = \frac{p(x_i \mid C_j)\cdot p(C_j)}{\sum_j p(x_i \mid C_j)\cdot p(C_j)}$$

Multi-dimensional Maximization Step for Color Image Segmentation

Input (Known)		Input (Estimation)		Output
$x_1=\{r_1, g_1, b_1\}$ $x_2=\{r_2, g_2, b_2\}$ … $x_i=\{r_i, g_i, b_i\}$ …	+	Classification Results $p(C_1\|x_1)$ $p(C_j\|x_2)$ … $p(C_j\|x_i)$ …	$\Rightarrow$	Cluster Parameters (μ_1,Σ_1), $p(C_1)$ for C_1 (μ_2,Σ_2), $p(C_2)$ for C_2 … (μ_k,Σ_k), $p(C_k)$ for C_k

$$\mu_j = \frac{\sum_i p(C_j \mid x_i) \cdot x_i}{\sum_i p(C_j \mid x_i)} \qquad \Sigma_j = \frac{\sum_i p(C_j \mid x_i) \cdot (x_i - \mu_j) \cdot (x_i - \mu_j)^T}{\sum_i p(C_j \mid x_i)} \qquad p(C_j) = \frac{\sum_i p(C_j \mid x_i)}{N}$$

Full EM Algorithm
Multi-Dimensional

- Boot Step:
  - Initialize K clusters: $C_1, \ldots, C_K$

    (μ_j, Σ_j) and $P(C_j)$ for each cluster j.

- Iteration Step:
  - Expectation Step

$$p(C_j \mid x_i) = \frac{p(x_i \mid C_j) \cdot p(C_j)}{p(x_i)} = \frac{p(x_i \mid C_j) \cdot p(C_j)}{\sum_j p(x_i \mid C_j) \cdot p(C_j)}$$

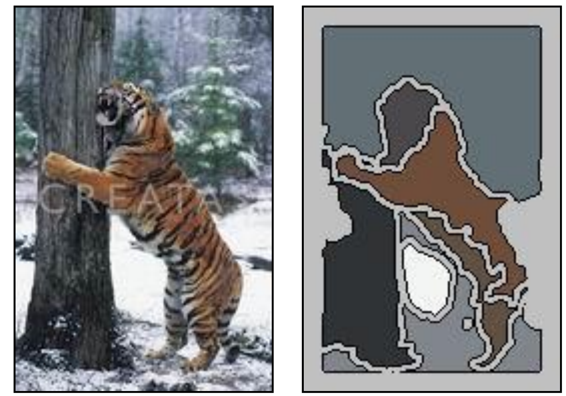
  - Maximization Step

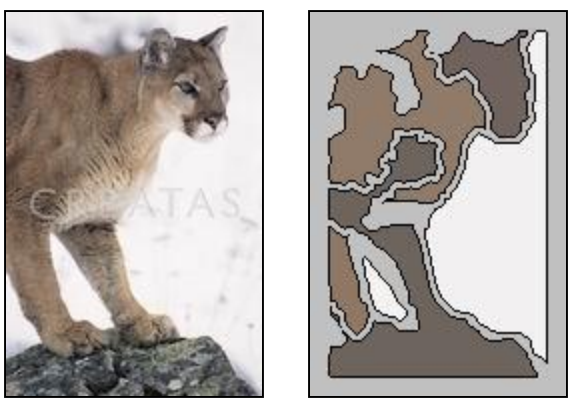
$$\mu_j = \frac{\sum_i p(C_j \mid x_i) \cdot x_i}{\sum_i p(C_j \mid x_i)} \qquad \Sigma_j = \frac{\sum_i p(C_j \mid x_i) \cdot (x_i - \mu_j) \cdot (x_i - \mu_j)^T}{\sum_i p(C_j \mid x_i)} \qquad p(C_j) = \frac{\sum_i p(C_j \mid x_i)}{N}$$

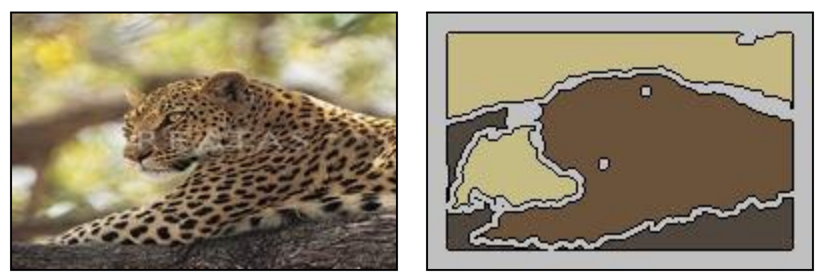
EM Applications

- Blobworld: Image segmentation using Expectation-Maximization and its application to image querying
- Used both color and texture features with the EM algorithm.

Blobworld: Sample Results







EM Classifier Approach

Object Class Recognition using Images of Abstract Regions

Yi Li, Jeff A. Bilmes, and Linda G. Shapiro

Department of Computer Science and Engineering

Department of Electrical Engineering

University of Washington

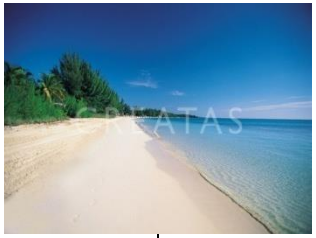
Problem Statement

Given: Some images and their corresponding descriptions

{trees, grass, cherry trees}

{cheetah, trunk}

{mountains, sky}



{beach, sky, trees, water}

•••

To solve: What object classes are present in new images

?

?

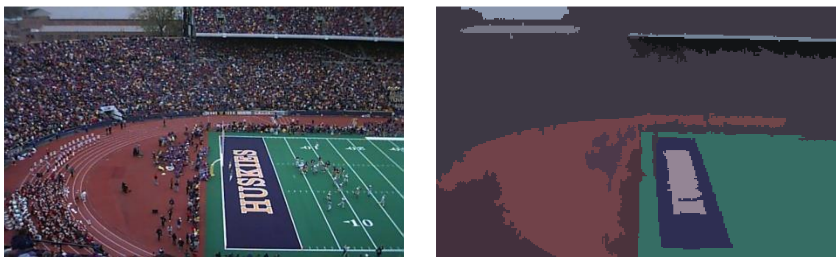
?

?

•••

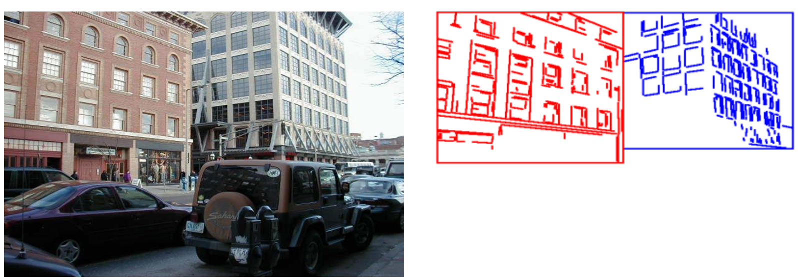
Image Features for Object Recognition

- Color

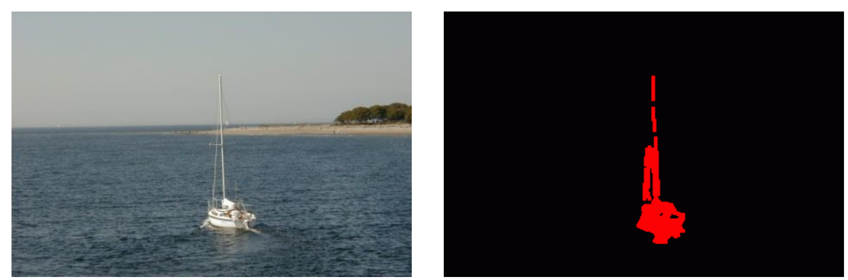


- Texture

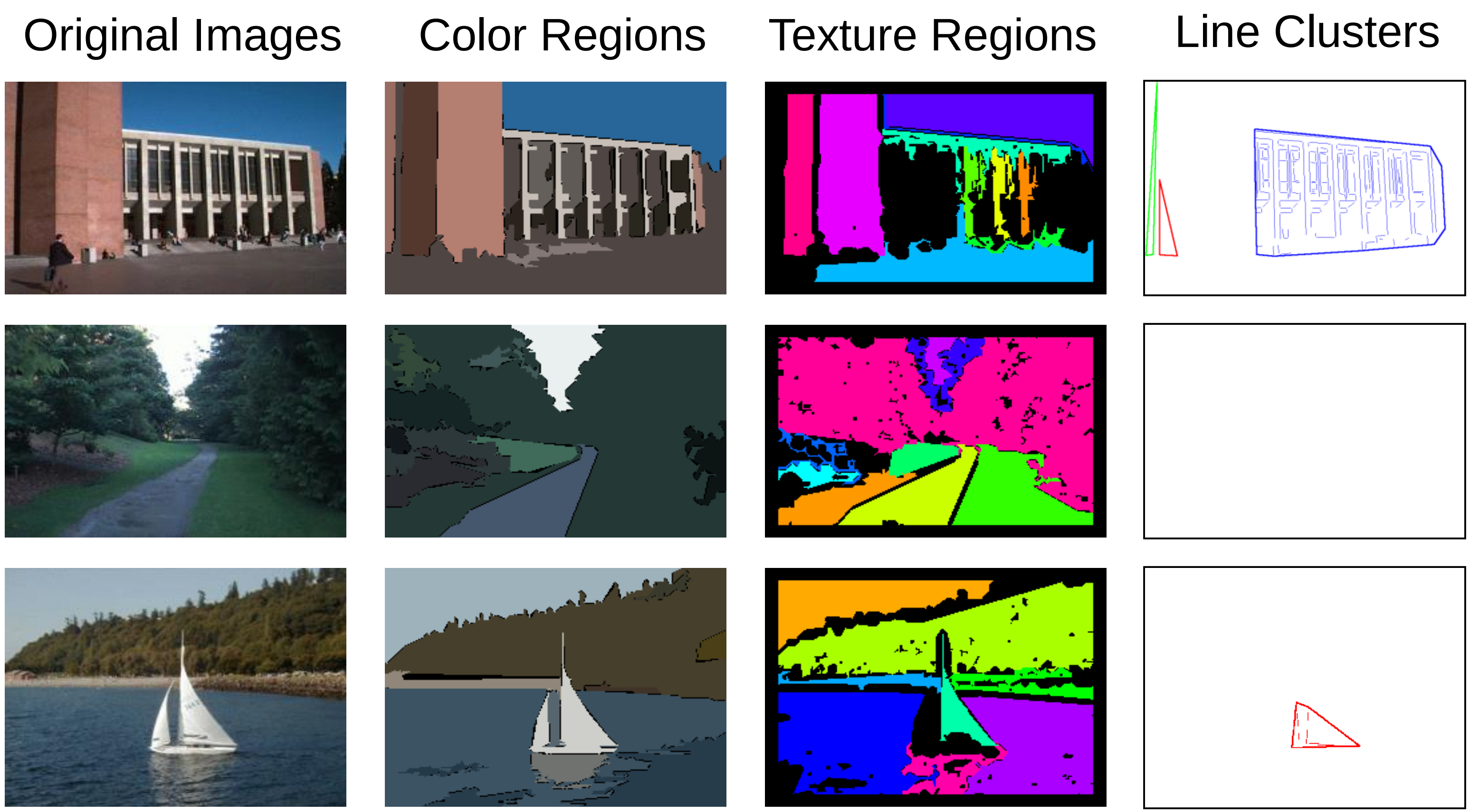
- Structure



- Context



Abstract Regions



Abstract Regions

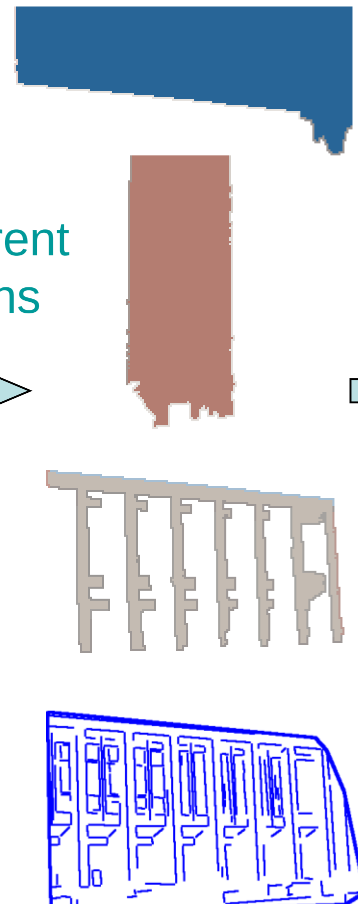
Multiple segmentations whose regions are not labeled; a list of labels is provided for each training image.

image

labels

{sky, building}

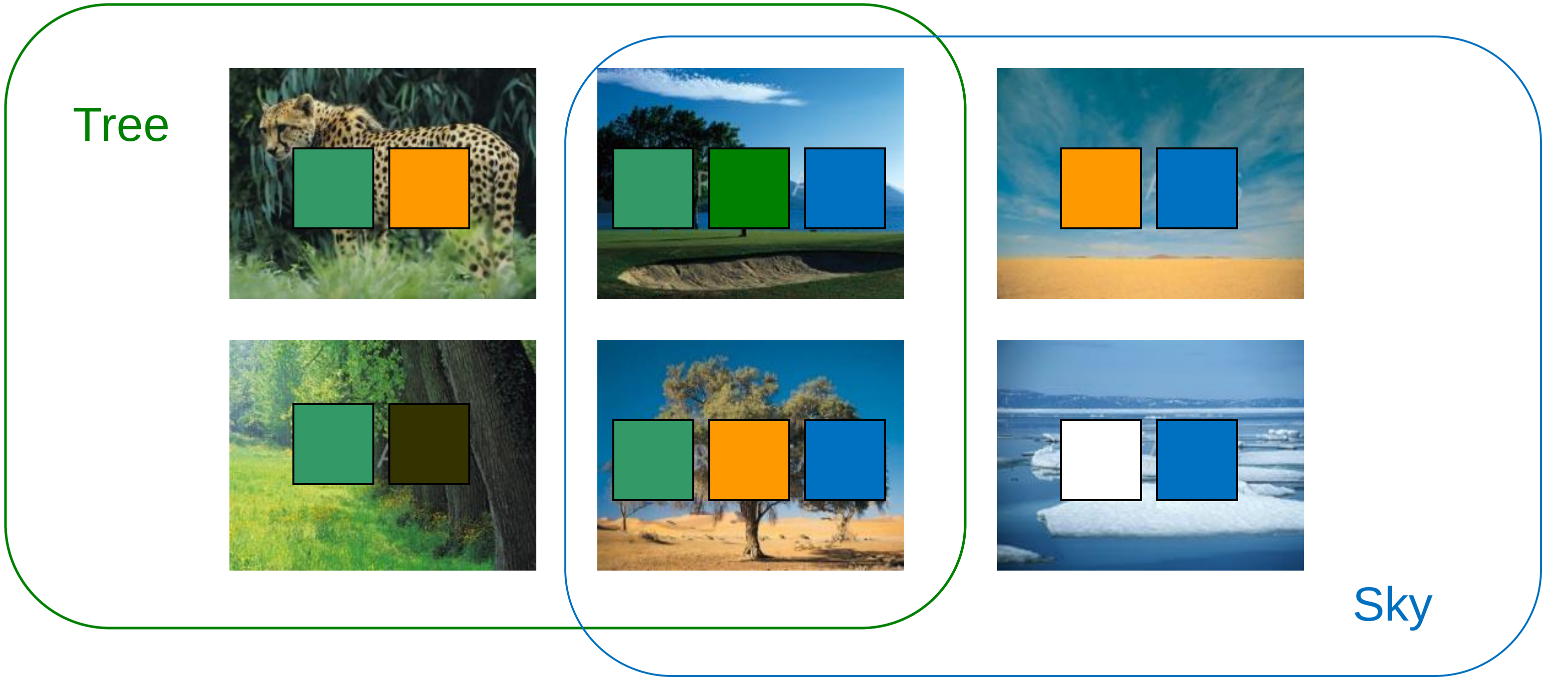
various different segmentations



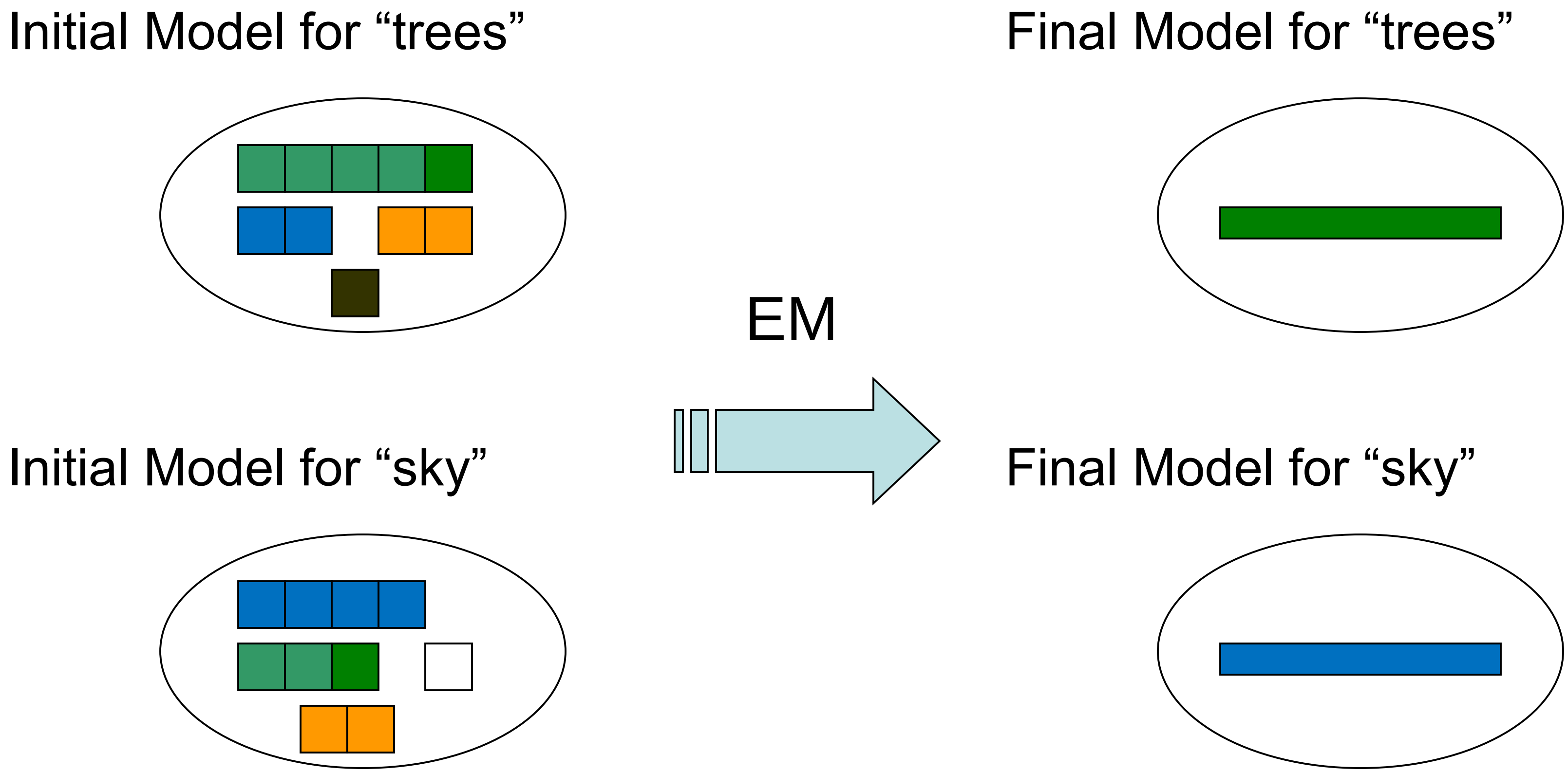
region attributes from several different types of regions

Model Initial Estimation

- Estimate the initial model of an object using all the region features from all images that contain the object



EM Classifier: the Idea

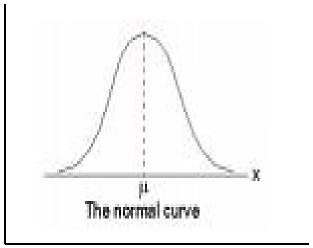


EM Algorithm

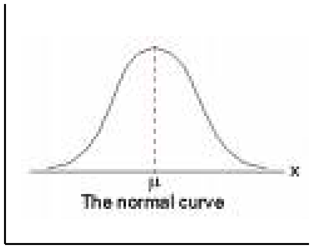
- Start with K clusters, each represented by a probability distribution
- Assuming a Gaussian or Normal distribution, each cluster is represented by its mean and variance (or covariance matrix) and has a weight.
- Go through the training data and soft-assign it to each cluster. Do this by computing the probability that each training vector belongs to each cluster.
- Using the results of the soft assignment, recompute the parameters of each cluster.
- Perform the last 2 steps iteratively.

1-D EM with Gaussian Distributions

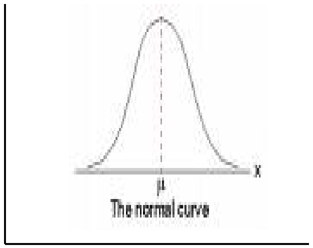
- Each cluster C_j is represented by a Gaussian distribution $N(\mu_j , \sigma_j)$.
- Initialization: For each cluster C_j initialize its mean μ_j , variance σ_j, and weight α_j.



$N(\mu_1 , \sigma_1)$
$\alpha_1 = P(C_1)$



$N(\mu_2 , \sigma_2)$
$\alpha_2 = P(C_2)$



$N(\mu_3 , \sigma_3)$
$\alpha_3 = P(C_3)$

- With no other knowledge, use random means and variances and equal weights.

Standard EM to EM Classifier

- That's the standard EM algorithm.
- For n-dimensional data, the variance becomes a co-variance matrix, which changes the formulas slightly.
- But we used an EM variant to produce a classifier.
- The next slide indicates the differences between what we used and the standard.

EM Classifier

1. Fixed Gaussian components (one Gaussian per object class) and fixed weights corresponding to the frequencies of the corresponding objects in the training data.

2. Customized initialization uses only the training images that contain a particular object class to initialize its Gaussian.

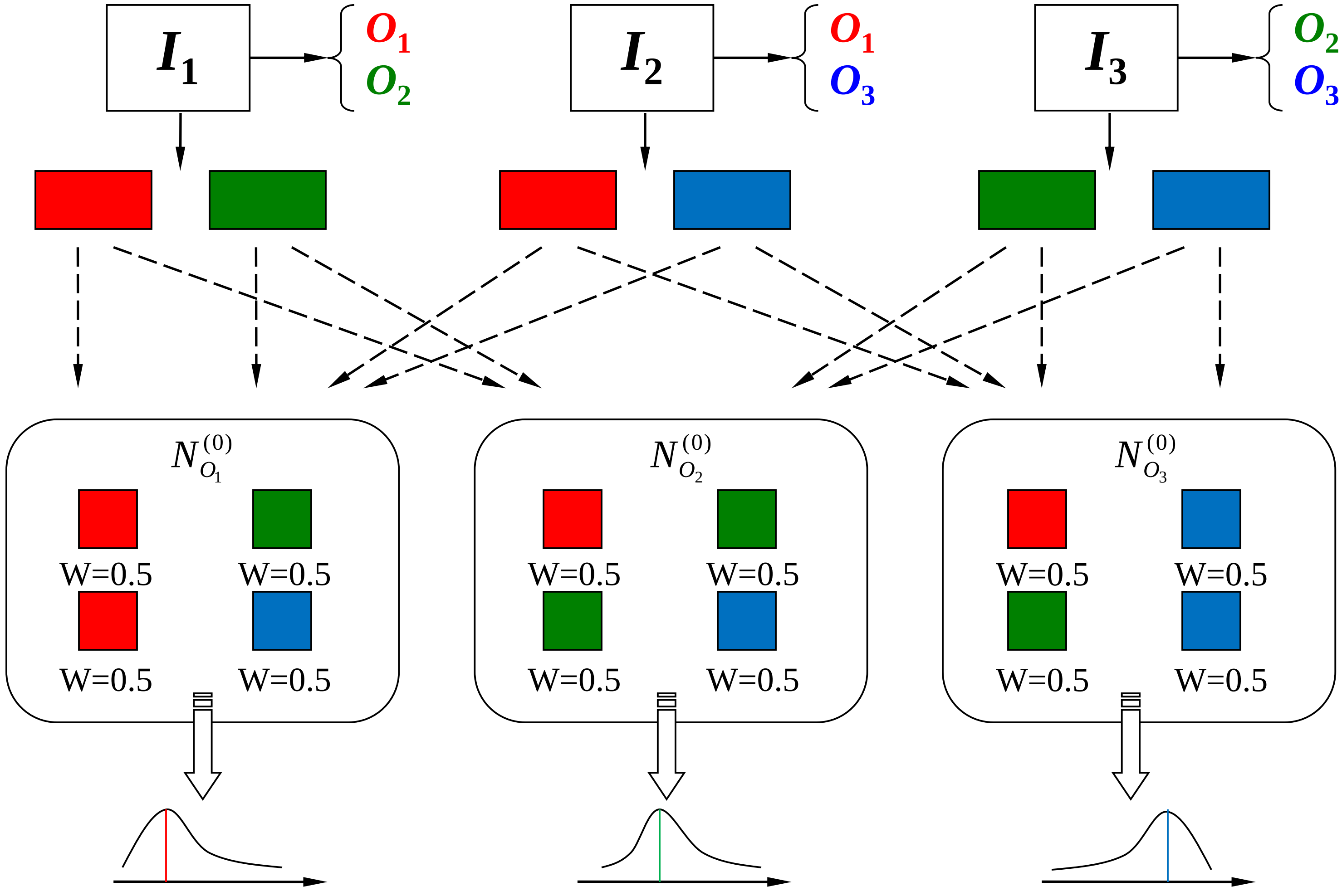
3. Controlled expectation step ensures that a feature vector only contributes to the Gaussian components representing objects present in its training image.

4. Extra background component absorbs noise.

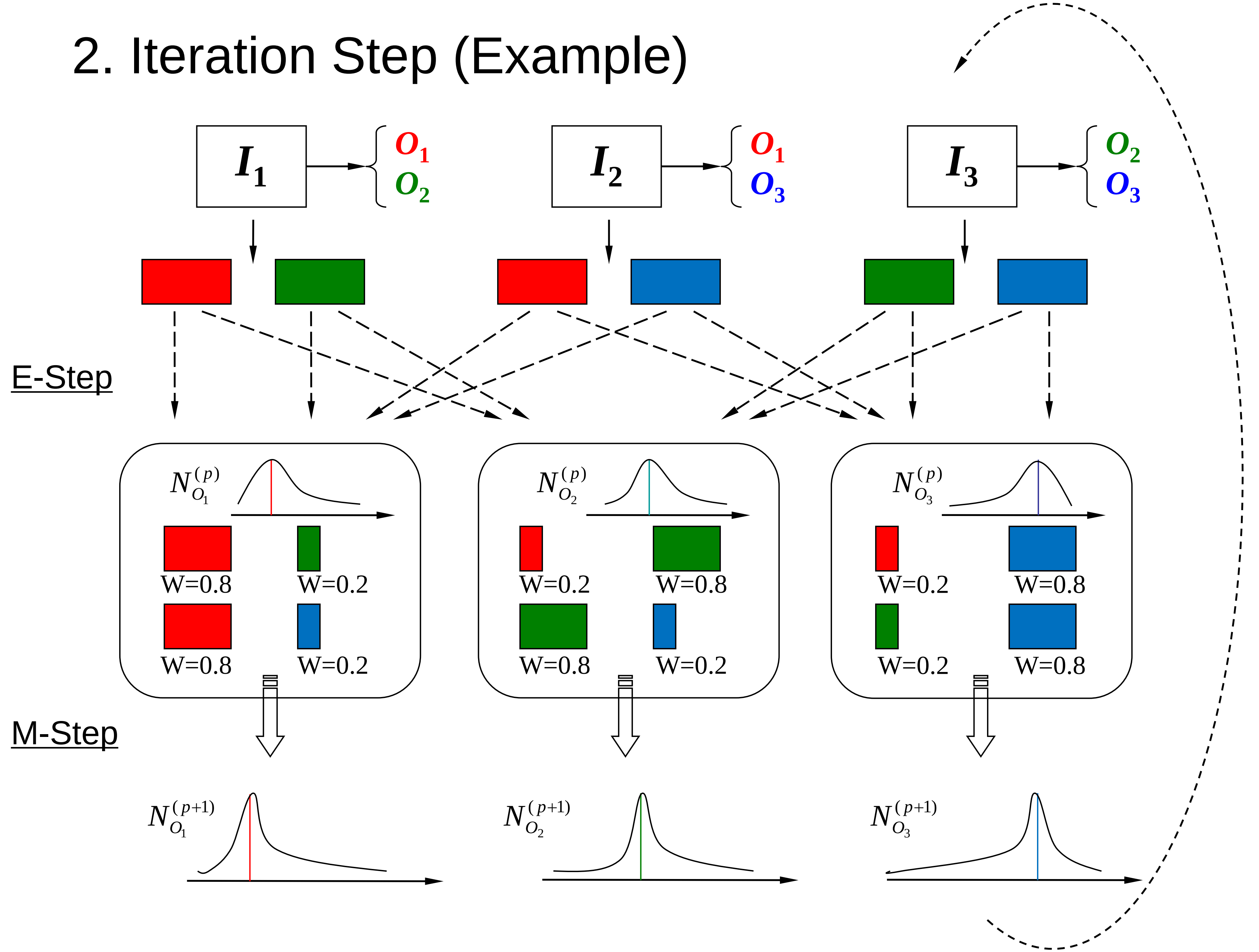
Gaussian for trees	Gaussian for buildings	Gaussian for sky	Gaussian for background

1. Initialization Step (Example)

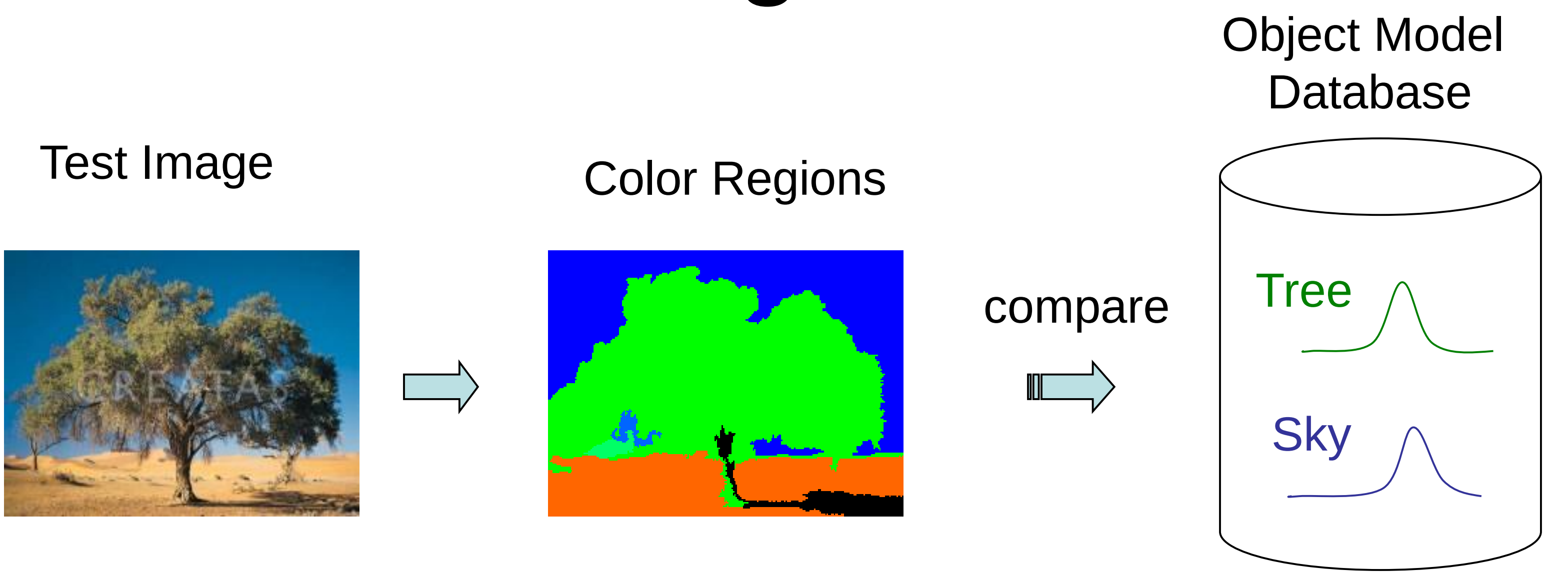
Image & description



2. Iteration Step (Example)



Recognition



How do you decide if a particular object is in an image?

To calculate $p(tree \mid image)$

$$p(tree \mid image) = f\left(\begin{array}{l} p(\,tree \mid \quad) \\ p(\,tree \mid \quad) \\ p(\,tree \mid \quad) \\ p(\,tree \mid \quad) \end{array} \right)$$

f is a function that combines probabilities from all the color regions in the image.

e.g. max or mean

Combining different types of abstract regions: First Try

- Treat the different types of regions independently and combine at the time of classification.

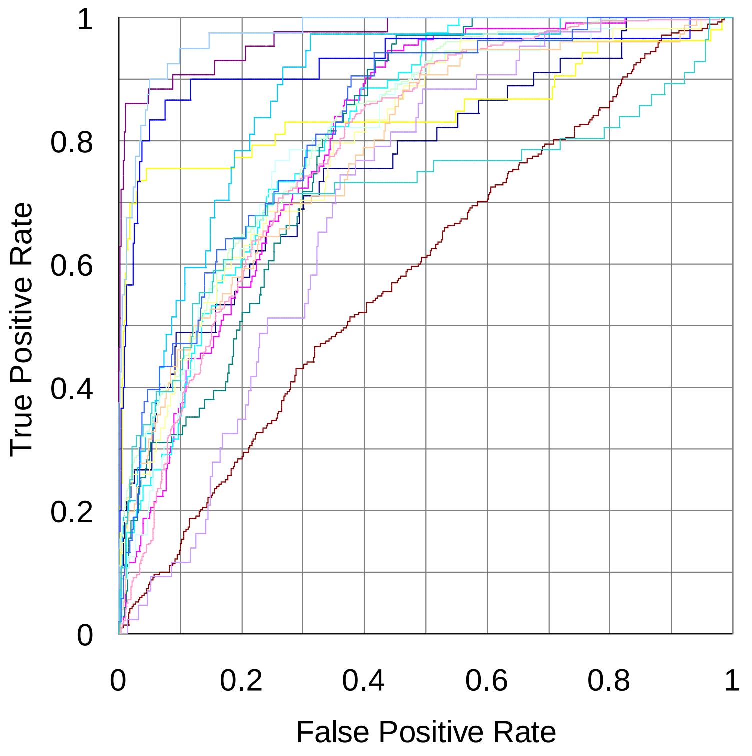
1. $P(\text{object}|\ a_1, a_2,..,a_n) = P(\text{object}|a_1)*..*P(\text{object}|a_n)$

2. Form intersections of the different types of regions, creating smaller regions that have both color and texture properties for classification.

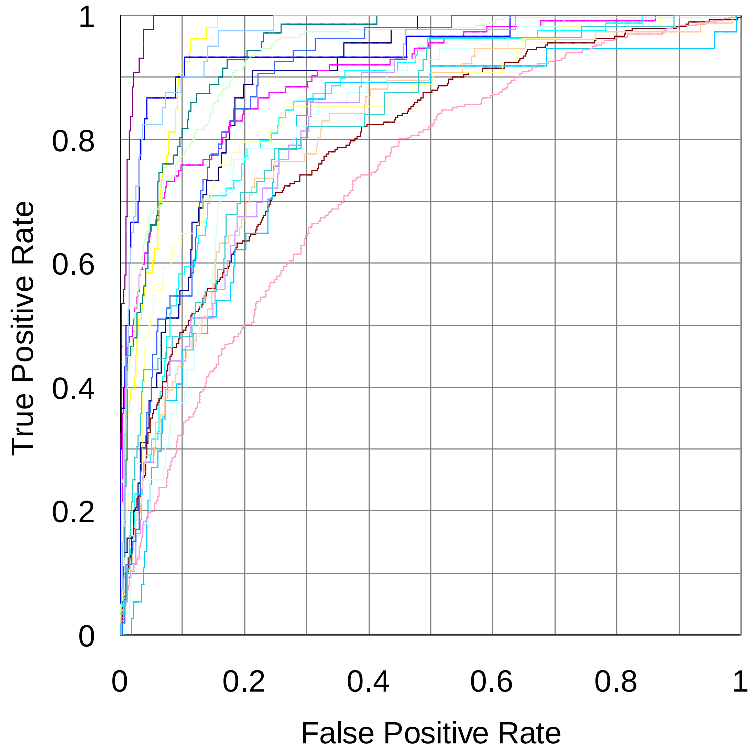
Experiments (on 860 images)

- 18 keywords: mountains (30), orangutan (37), track (40), tree trunk (43), football field (43), beach (45), prairie grass (53), cherry tree (53), snow (54), zebra (56), polar bear (56), lion (71), water (76), chimpanzee (79), cheetah (112), sky (259), grass (272), tree (361).
- A set of cross-validation experiments (80% as training set and the other 20% as test set)
- The poorest results are on object classes "tree," "grass," and "water," each of which has a high variance; a single Gaussian model is insufficient.

ROC Charts:
True Positive vs. False Positive



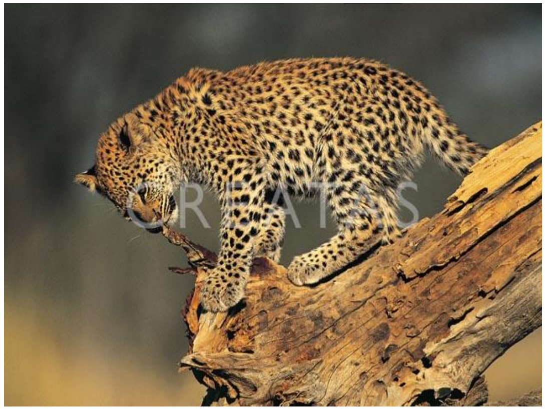
Independent Treatment of
Color and Texture



Using Intersections of
Color and Texture Regions

Sample Retrieval Results

cheetah



Sample Results (Cont.)

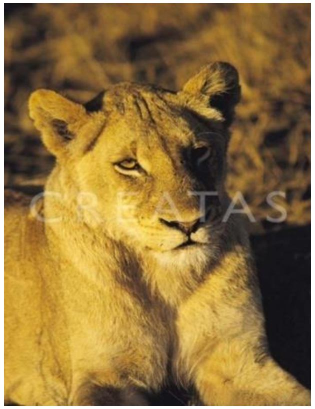
grass

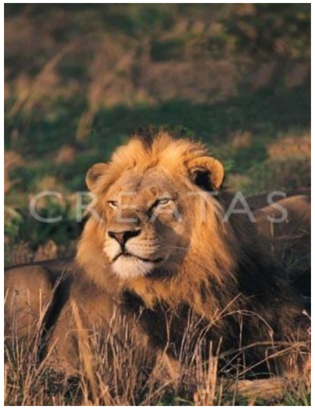
Sample Results (Cont.)

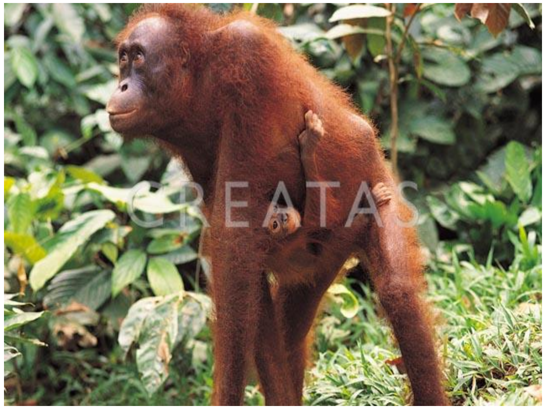
cherry tree

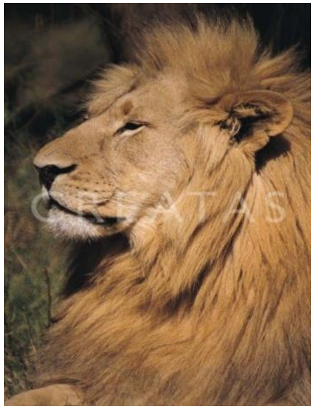
Sample Results (Cont.)

lion









Summary

- Designed a set of abstract region features: color, texture, structure, . . .
- Developed a new semi-supervised EM-like algorithm to recognize object classes in color photographic images of outdoor scenes; tested on 860 images.
- Compared two different methods of combining different types of abstract regions. The intersection method had a higher performance

Weakness of the EM Classifier Approach

- It did not generalize well to multiple features

- It assumed that object classes could be modeled as Gaussians