COMPUTER VISION
Introduction

Computer vision is the analysis of digital images by a computer for such applications as:

- Industrial: part localization and inspection, robotics
- Medical: disease classification, screening, planning
- Military: autonomous vehicles, tank recognition
- Intelligence Gathering: face recognition, video analysis
- Security: video analysis
- Science: classification, measurement
- Document Processing: text recognition, diagram conversion

The Three Stages of Computer Vision

- low-level (image processing)

  image → image

- mid-level (feature extraction)

  image → features

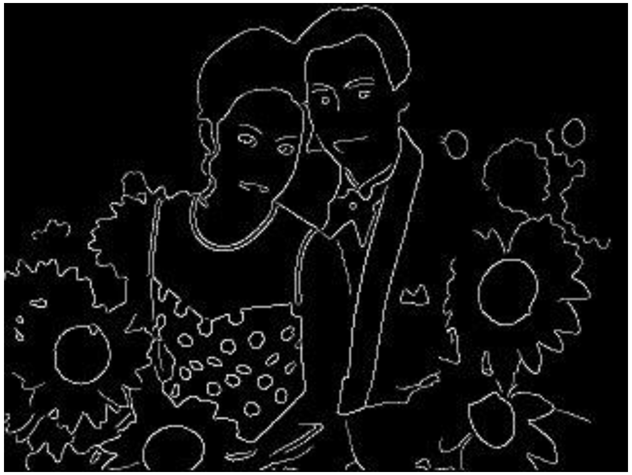
- high-level (the intelligent part)

  features → analysis

Low-Level

original image

Canny edge operator



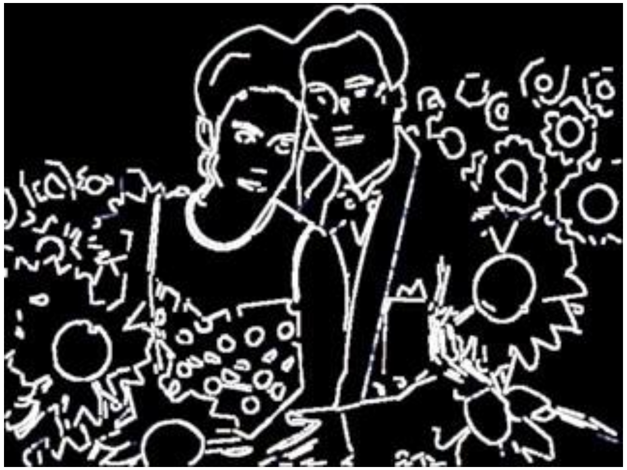
edge image

Mid-Level (Lines and Curves)

edge image

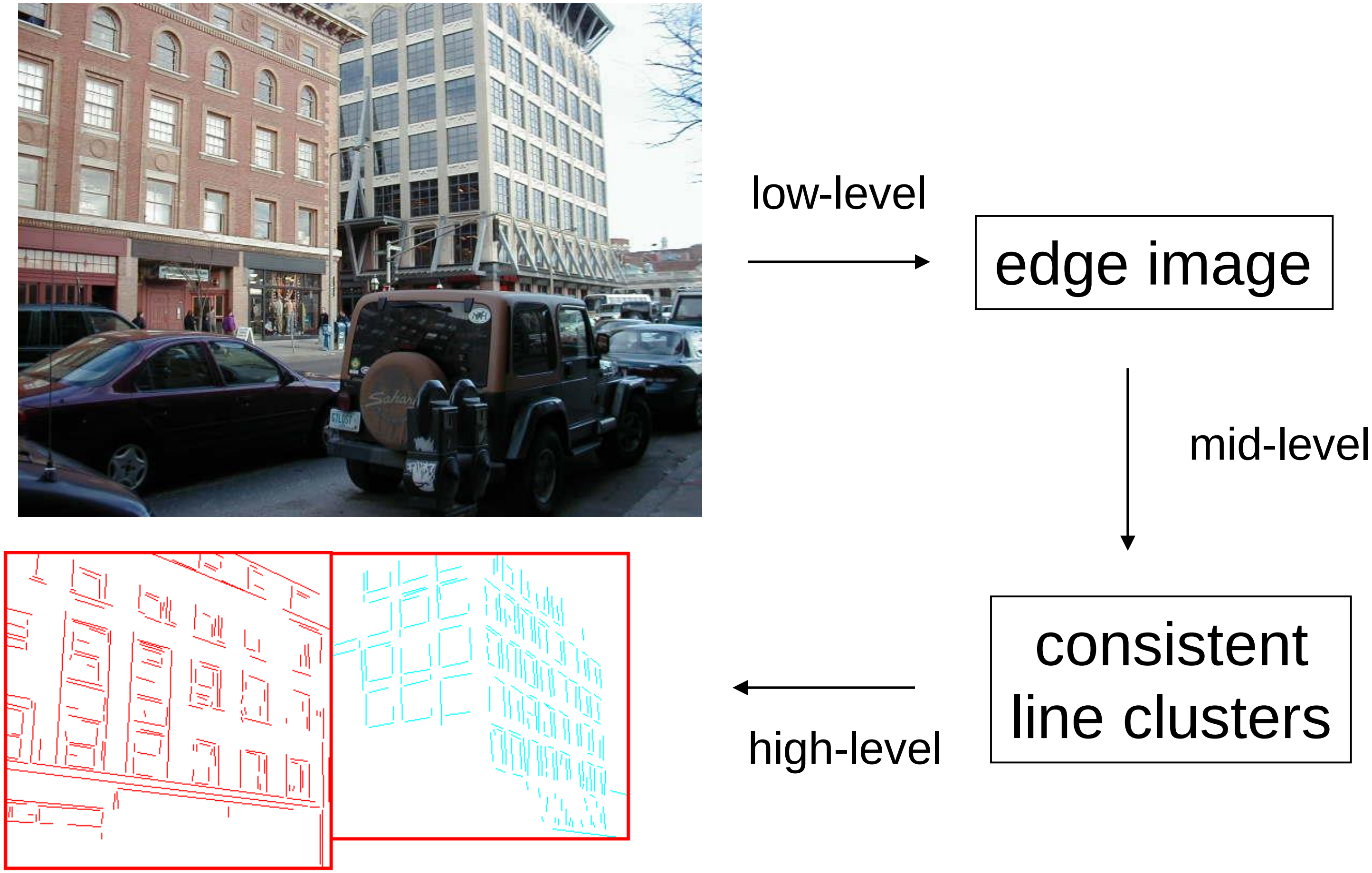
ORT line & circle extraction

data structure



circular arcs and line segments

Low- to High-Level



low-level

edge image

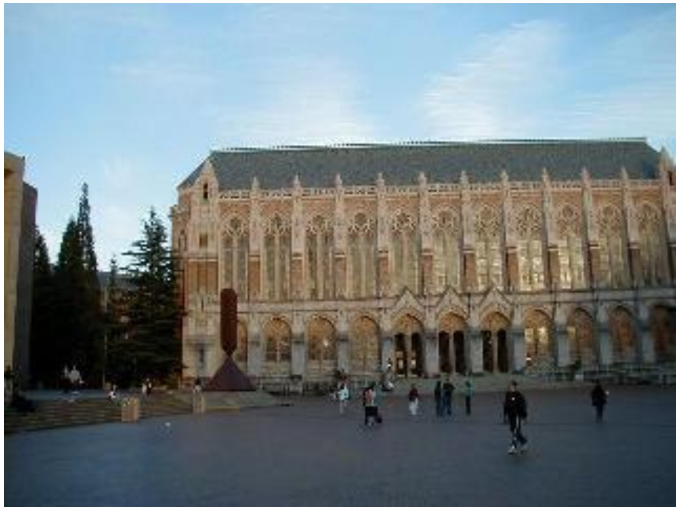
mid-level

consistent line clusters

high-level

Building Recognition

Mid-level (Regions)

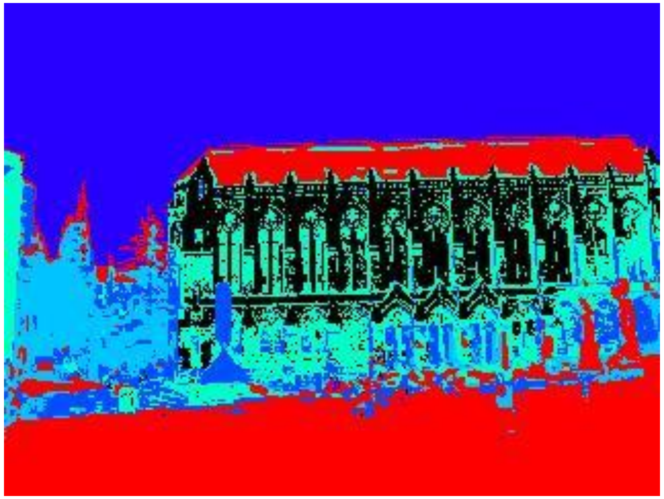


original color image

K-means clustering

(followed by connected component analysis)

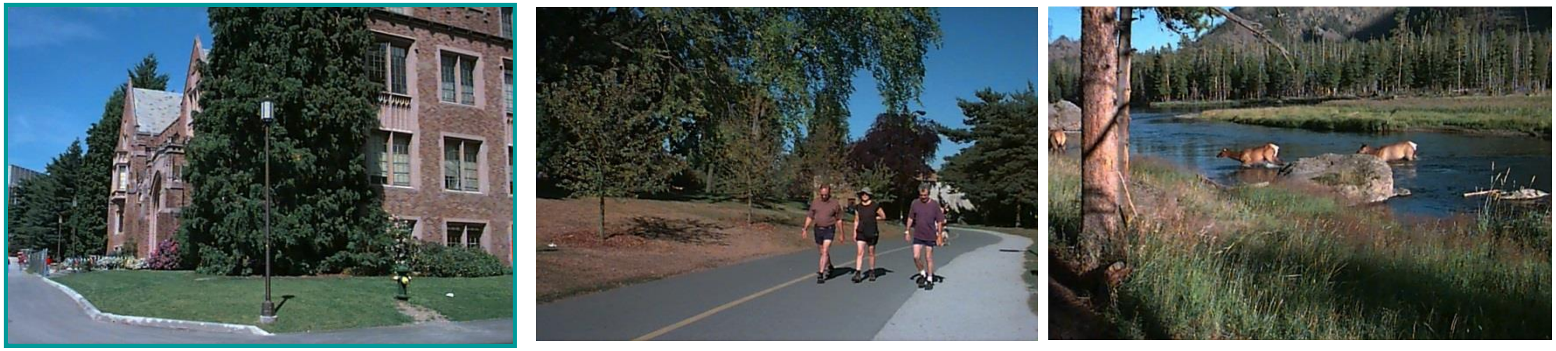
data structure



regions of homogeneous color

Image Databases:

Images from my Ground-Truth collection.



- Retrieve all images that have trees.
- Retrieve all images that have buildings.
- Retrieve all images that have antelope.

Simpler

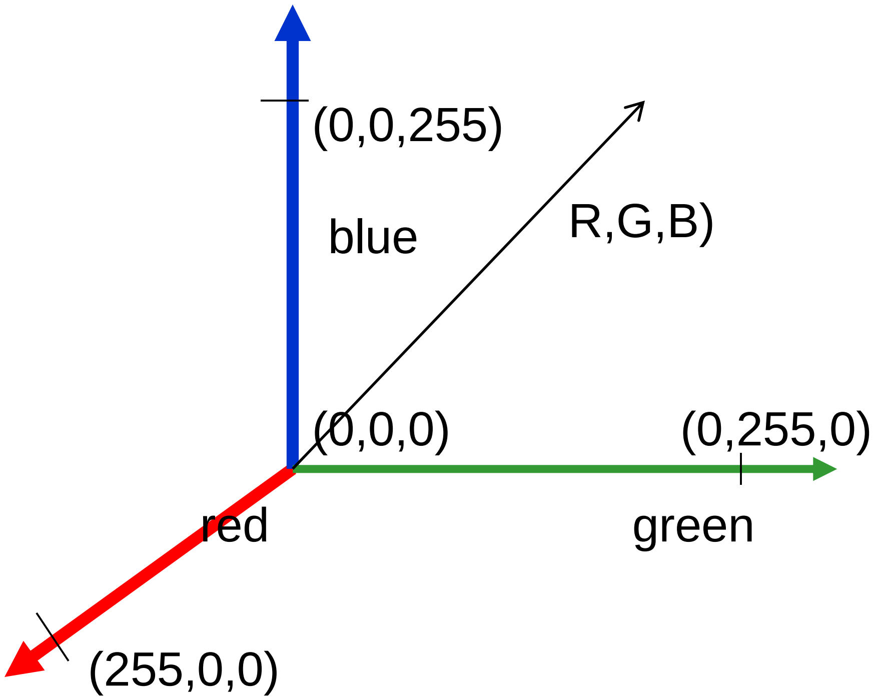
- Retrieve images based on their color and texture attributes
  - Color histograms
  - Texture histograms
  - Both
- This is content-based image retrieval: CBIR.

Color Spaces

• RGB	standard for cameras
• HSI/HSV	hue, saturation, intensity
• CIE L*a*b	intensity plus 2 color channels
• YIQ	color TVs, Y is intensity
• and more	

RGB Color Space

Absolute



Normalized

Normalized red $r = R/(R+G+B)$

Normalized green $g = G/(R+G+B)$

Normalized blue $b = B/(R+G+B)$

Conversion from RGB to YIQ

An approximate linear transformation from RGB to YIQ:

$$
\begin{array}{rrcrcrcr}
luminance & Y & = & 0.30R & + & 0.59G & + & 0.11B \\
R-cyan & I & = & 0.60R & - & 0.28G & - & 0.32B \\
magenta-green & Q & = & 0.21R & - & 0.52G & + & 0.31B
\end{array}
$$

We often use this for color to gray-tone conversion.

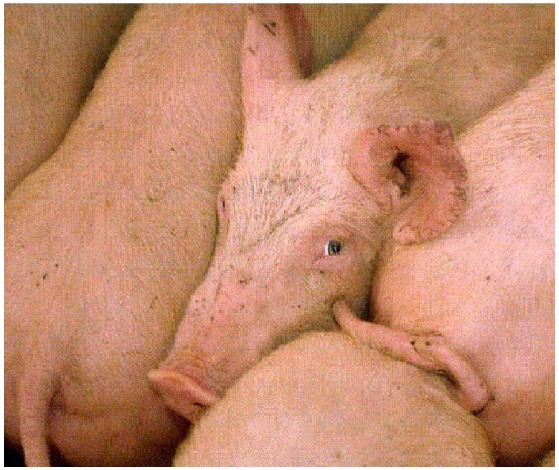
Histograms

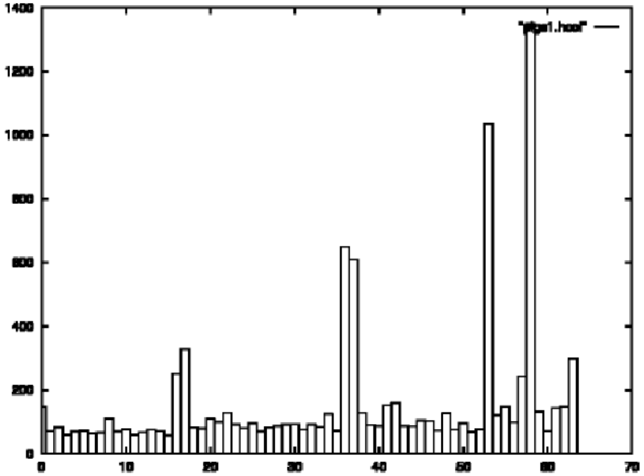
- A histogram of a gray-tone image is an array H[*] of bins, one for each gray tone.
- H[i] gives the count of how many pixels of an image have gray tone i.
- P[i] (the normalized histogram) gives the percentage of pixels that have gray tone i.

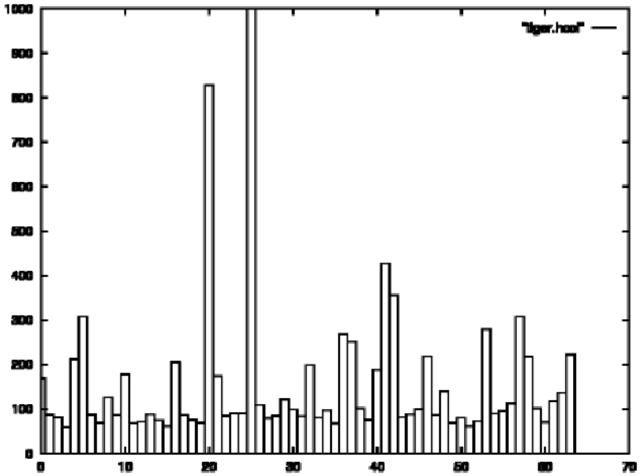
Color histograms can represent an image

- Histogram is fast and easy to compute.
- Size can easily be normalized so that different image histograms can be compared.
- Can match color histograms for database query or classification.

Histograms of two color images



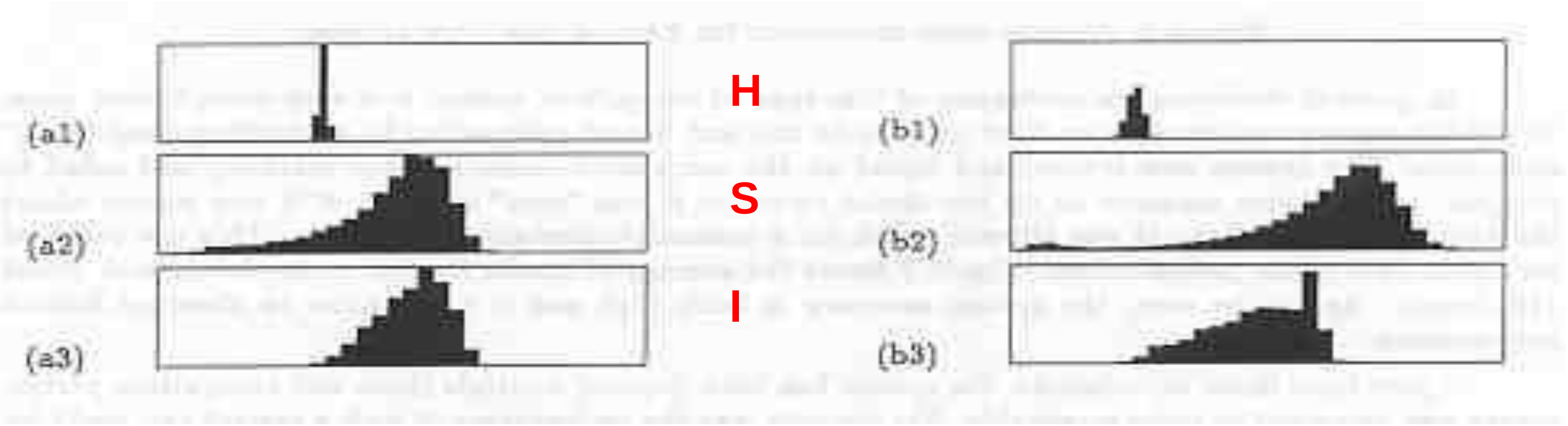




How to make a color histogram

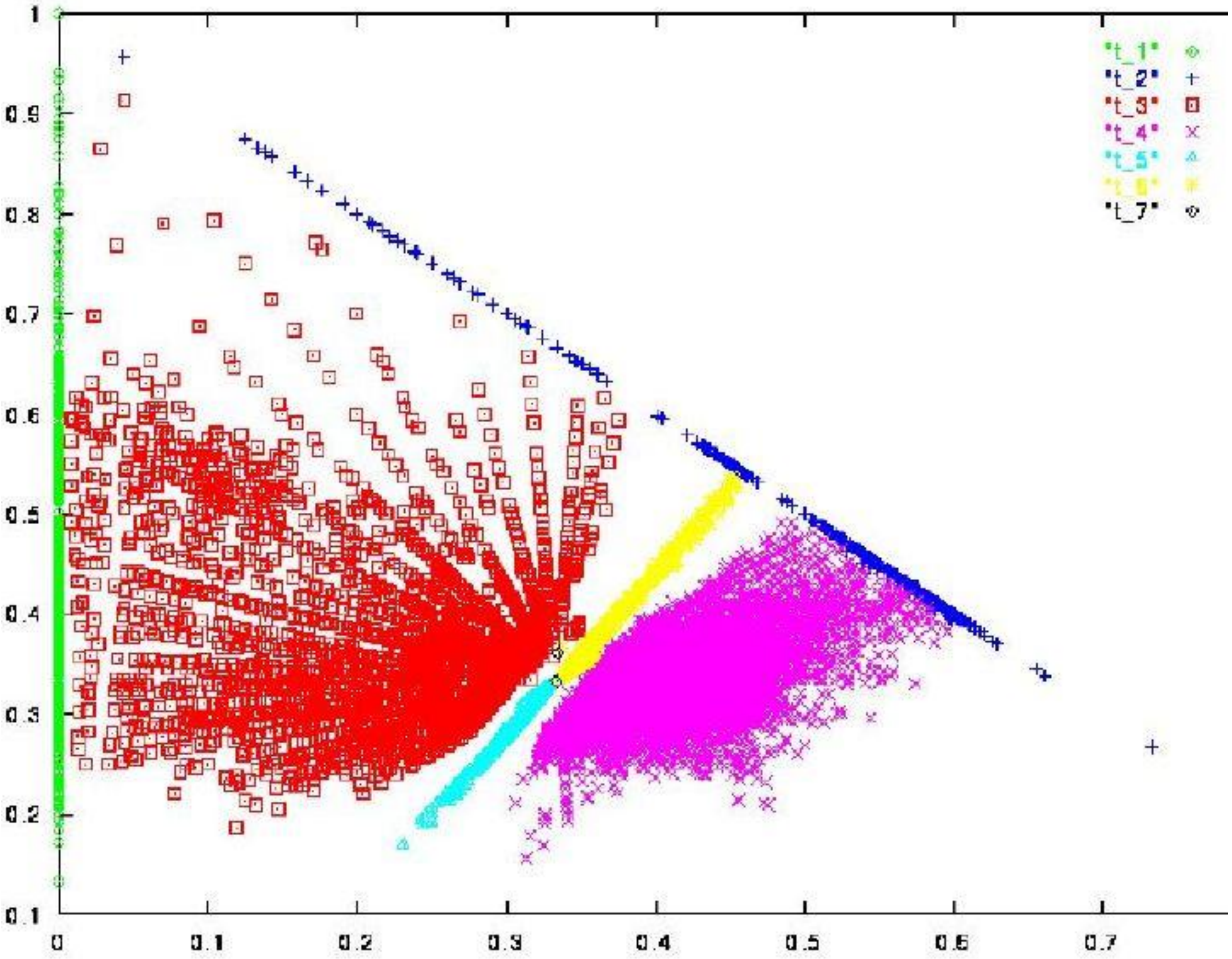
- Make a single 3D histogram. (BIG)
- Make 3 histograms and concatenate them. (Most Common and what Bindita has)
- Create a single pseudo color between 0 and 255 by using 3 bits of R, 3 bits of G and 2 bits of B (Old)
- Use normalized color space and 2D histograms.

Apples versus Oranges



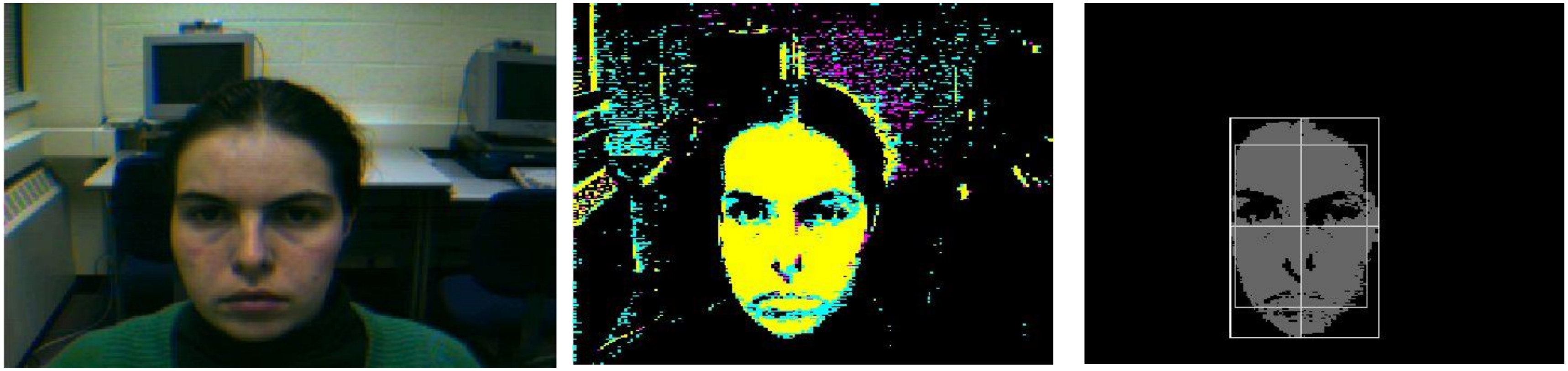
Separate HSI histograms for apples (left) and oranges (right) used by IBM's VeggieVision for recognizing produce at the grocery store checkout station (see Ch 16).

Skin color in RGB space (shown as normalized red vs normalized green)



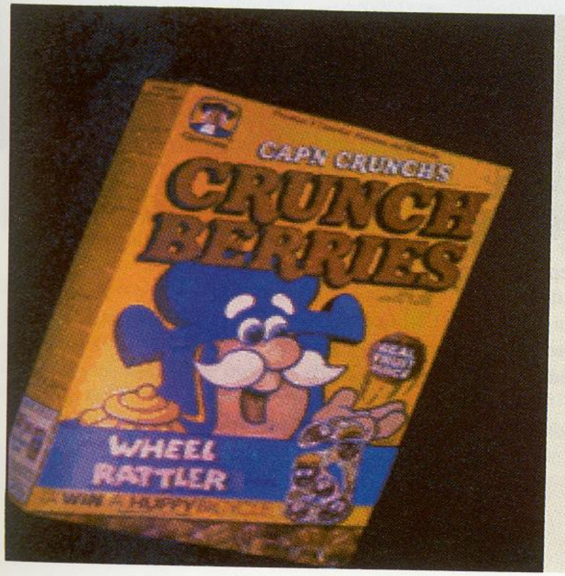
Purple region shows skin color samples from several people. Blue and yellow regions show skin in shadow or behind a beard.

Finding a face in video frame

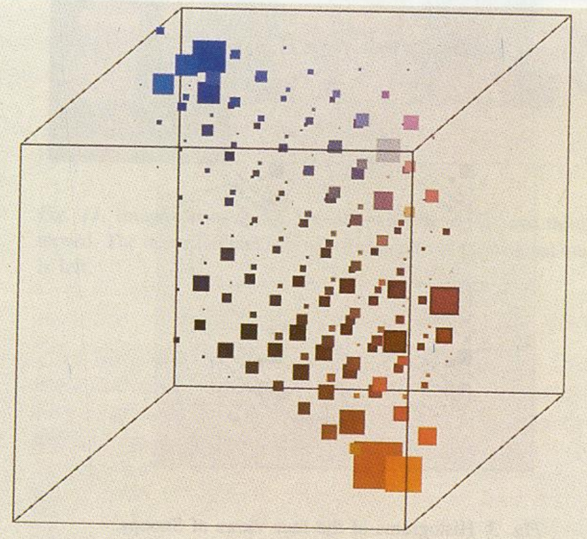


- (left) input video frame
- (center) pixels classified according to RGB space
- (right) largest connected component with aspect similar to a face (all work contributed by Vera Bakic)

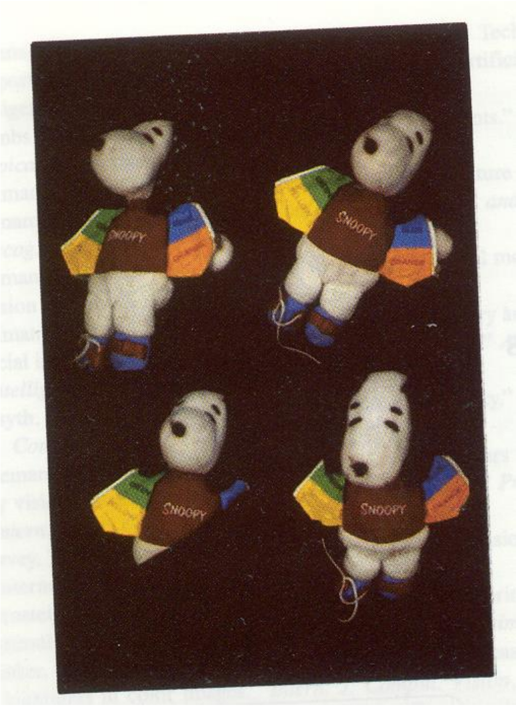
(from Swain and Ballard)

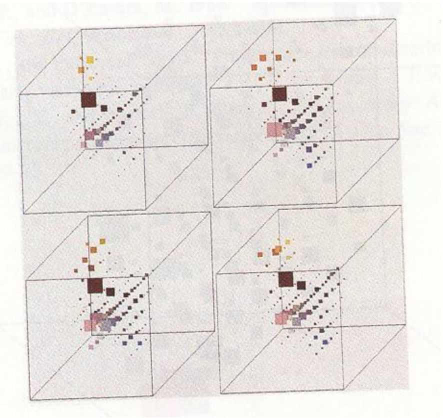


cereal box image



3D color histogram





Four views of Snoopy

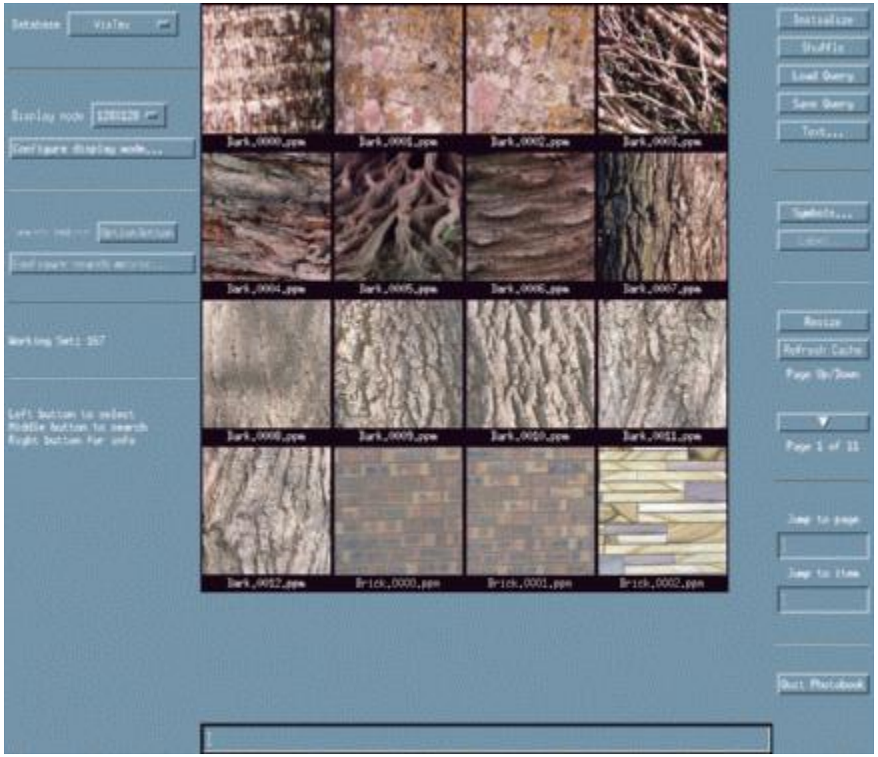
Histograms

Uses

- Although this is an extremely simple technique, it became the basis for many content-based image retrieval systems and works surprisingly well, both alone, and in conjunction with other techniques.

Texture

- Color is well defined.
- But what *is* texture?



Database VisTex

Display mode 128x128

Working Set: 167

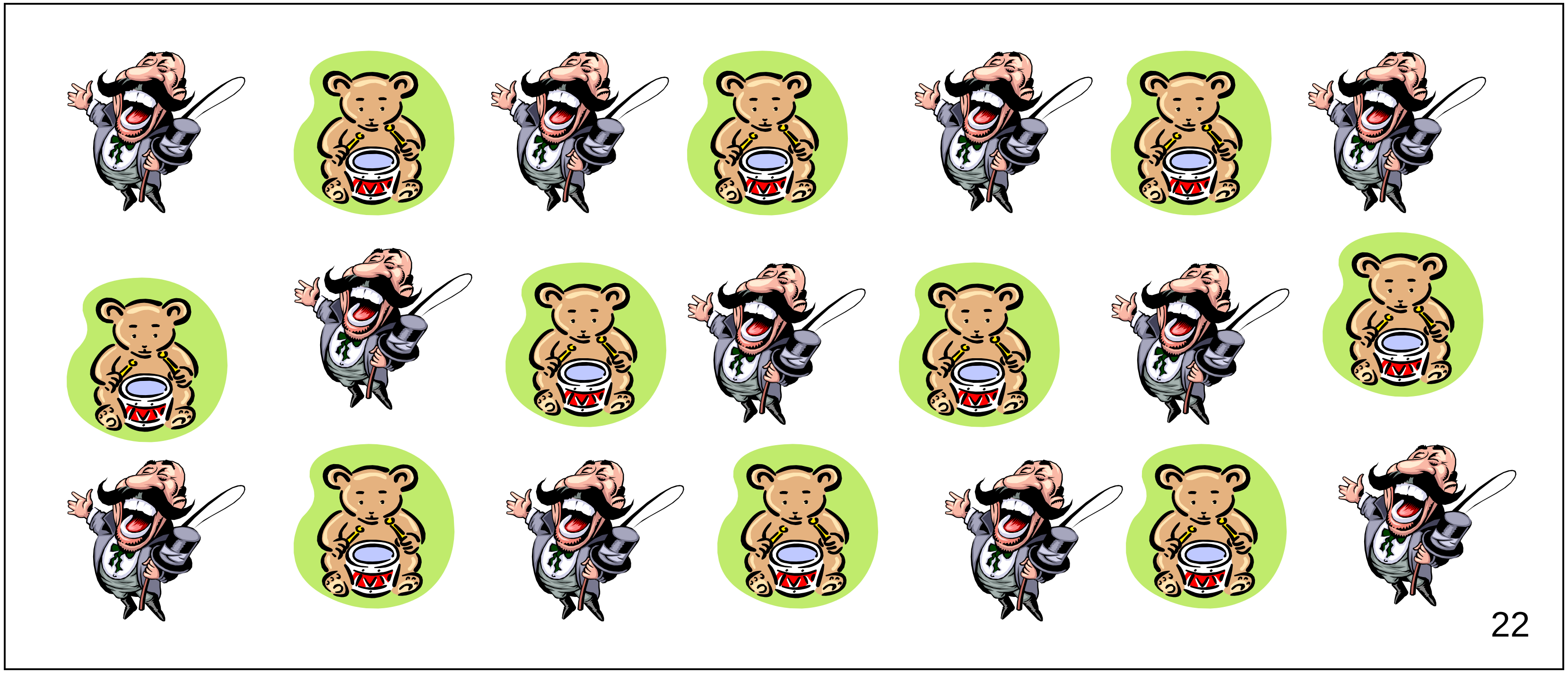
Left button to select
Middle button to search
Right button for info

Page 1 of 11

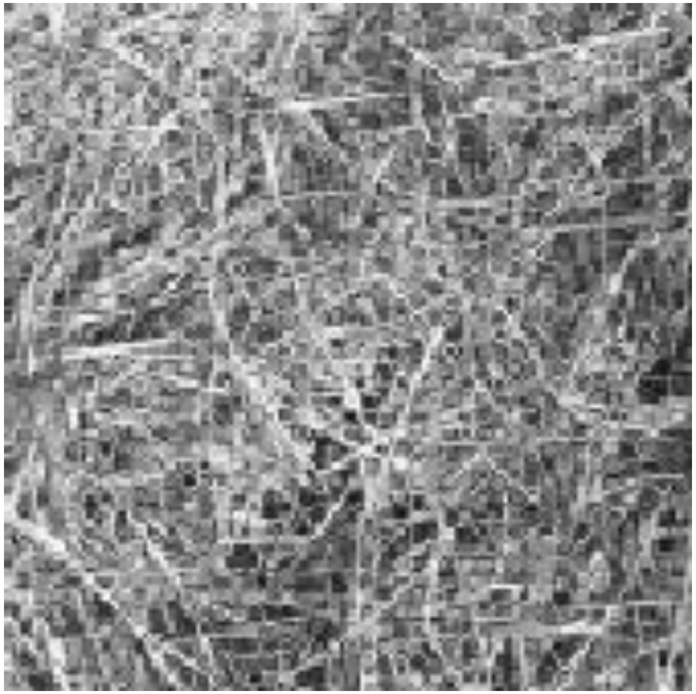
Structural Texture

Texture is a description of the spatial arrangement of color or intensities in an image or a selected region of an image.

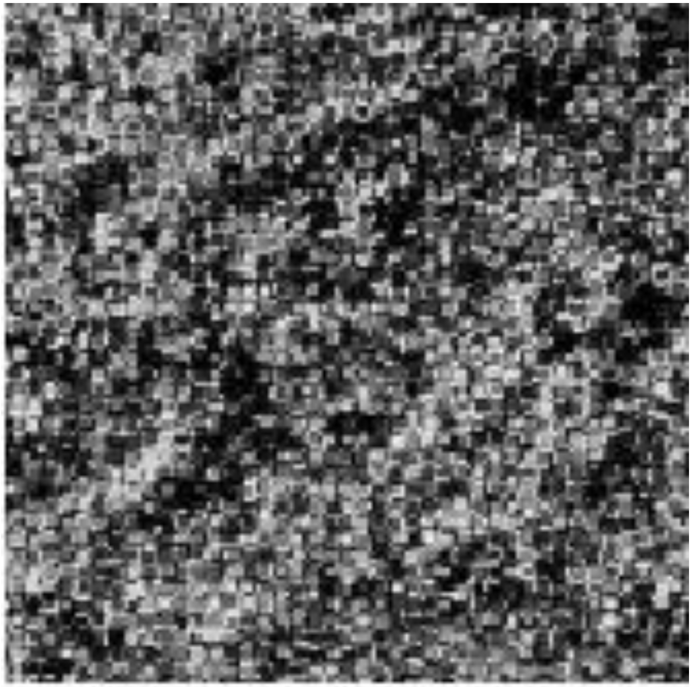
Structural approach: a set of texels in some regular or repeated pattern



Natural Textures from VisTex



grass



leaves

What/Where are the texels?

The Case for Statistical Texture

- Segmenting out texels is difficult or impossible in real images.
- Numeric quantities or statistics that describe a texture can be computed from the gray tones (or colors) alone.
- This approach is less intuitive, but is computationally efficient.
- It can be used for both classification and segmentation.

Some Simple Statistical Texture Measures

Edge Density and Direction

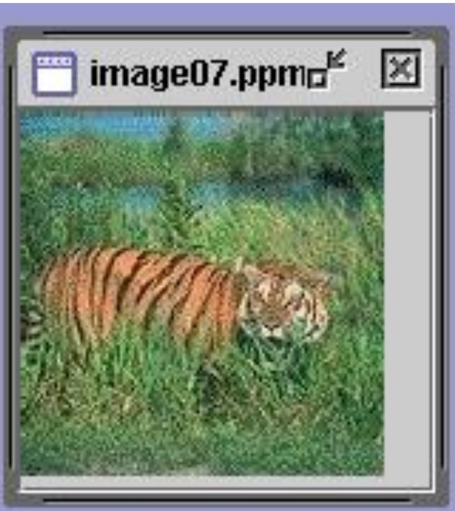
- Use an edge detector as the first step in texture analysis.

- The number of edge pixels in a fixed-size region tells us how busy that region is.

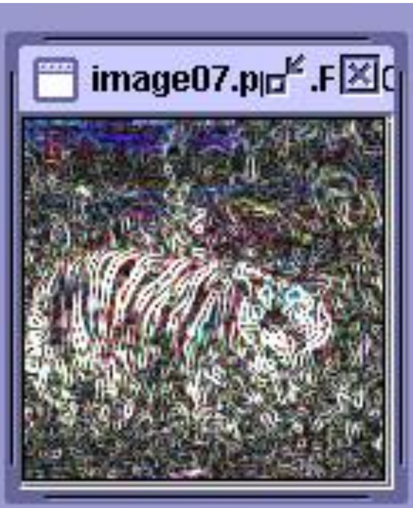
- The directions of the edges also help characterize the texture

Example

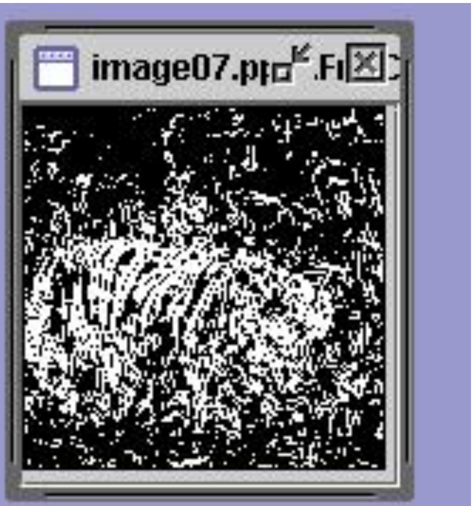
Original Image



Frei-Chen
Edge Image



Thresholded
Edge Image

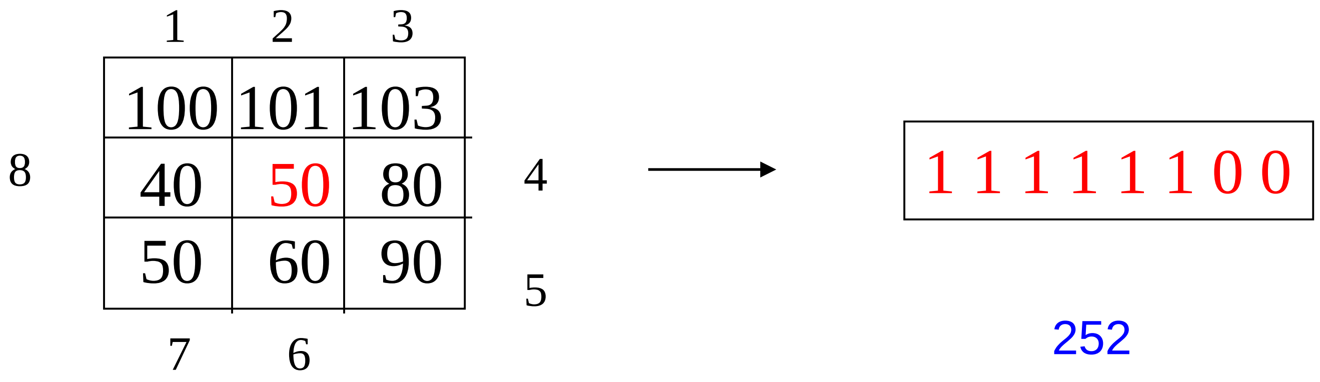


Local Binary **Pattern** Measure

- For each pixel p, create an 8-bit number b_1 b_2 b_3 b_4 b_5 b_6 b_7 b_8, where b_i = 0 if neighbor i has value less than or equal to p's value and 1 otherwise.

- Convert these 8-bit strings to integers.

- Represent the texture in the image (or a region) by the histogram of these numbers.



	1	2	3	
	100	101	103	
8	40	50	80	4
	50	60	90	5
	7	6		

1 1 1 1 1 1 0 0

252

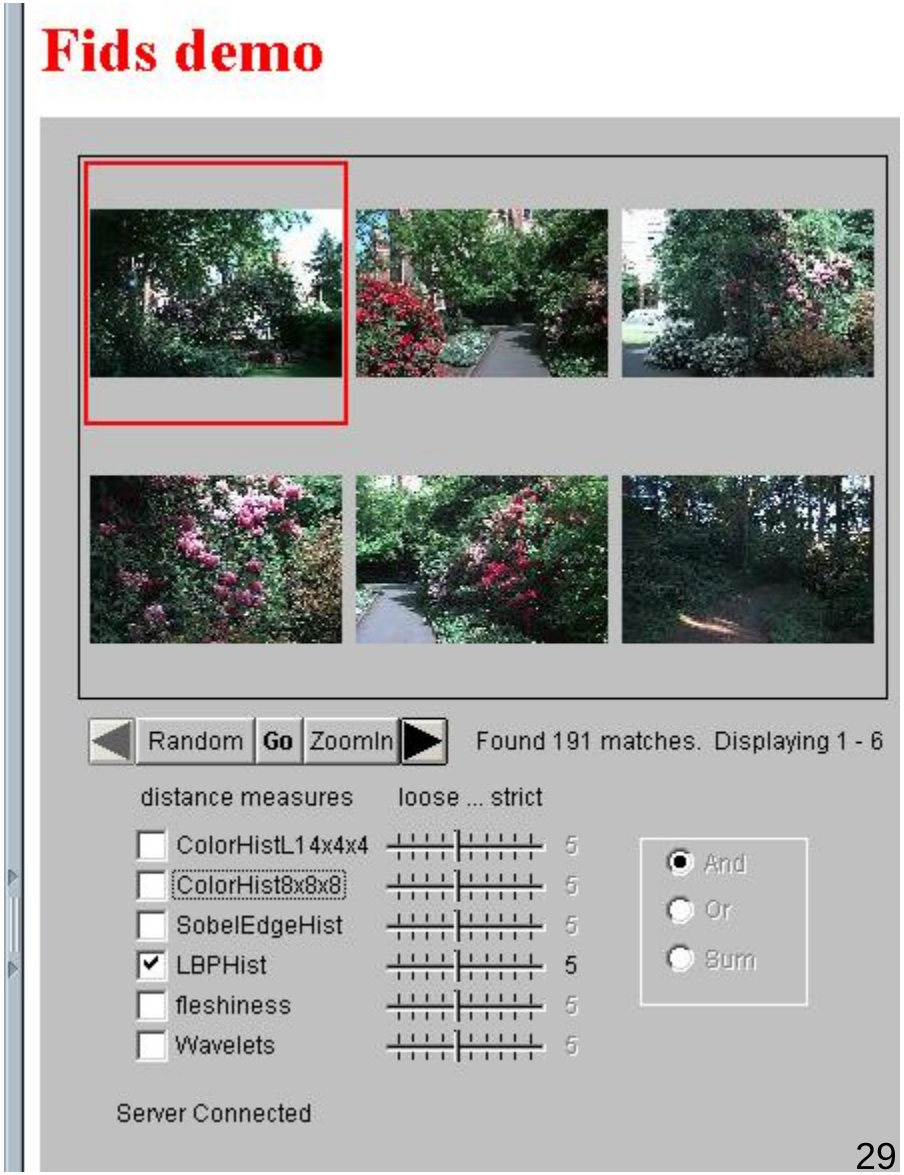
Distance Measures

- In order to retrieve images from a database, we need to use distance measures to measure how far is a query image from each database image.
- The most common distance measure in a vector space is Euclidean Distance.
- D((x1,x2,…xn),(y1,y2,…yn)) =

  $\mathrm{sqrt}[(x1\text{-}y1)^2 + (x2\text{-}y2)^2 + \ldots +(xn\text{-}yn)^2]$
- HW4 asks you to use this plus 3 others.

Example

Fids (Flexible Image Database System) is retrieving images similar to the query image using LBP texture as the texture measure and comparing their LBP histograms



Fids demo

Found 191 matches. Displaying 1 - 6

distance measures loose ... strict

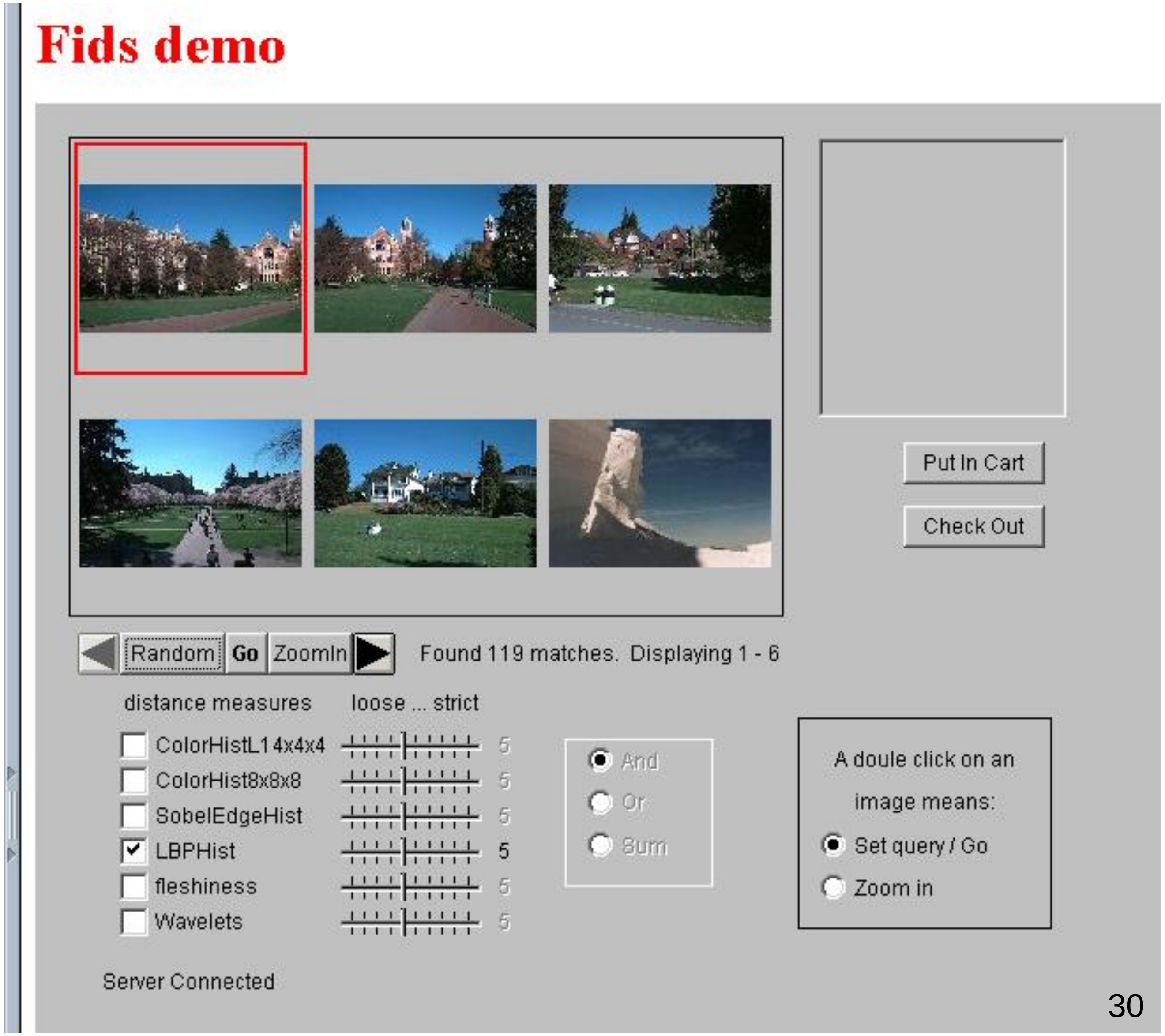
□ ColorHistL14x4x4 5
□ ColorHist8x8x8 5
□ SobelEdgeHist 5
☑ LBPHist 5
□ fleshiness 5
□ Wavelets 5

● And
○ Or
○ Sum

Server Connected

Example

Low-level measures don’t always find semantically similar images.



Fids demo

Found 119 matches. Displaying 1 - 6

distance measures loose ... strict

□ ColorHistL14x4x4	5
□ ColorHist8x8x8	5
□ SobelEdgeHist	5
☑ LBPHist	5
□ fleshiness	5
□ Wavelets	5

● And
○ Or
○ Sum

A doule click on an image means:
● Set query / Go
○ Zoom in

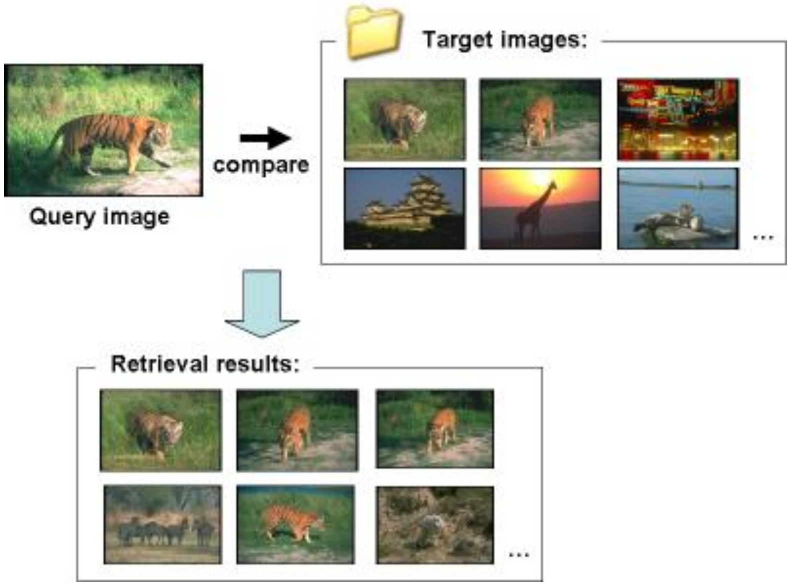
Server Connected

What else is LBP good for?

- We found it in a paper for classifying deciduous trees.
- We used it in a real system for finding cancer in Pap smears.
- We are using it to look for regions of interest in breast and melanoma biopsy slides.
- You will use it for HW4.

CSE 473 Winter 2019 Homework 4

Content-based Image Retrieval



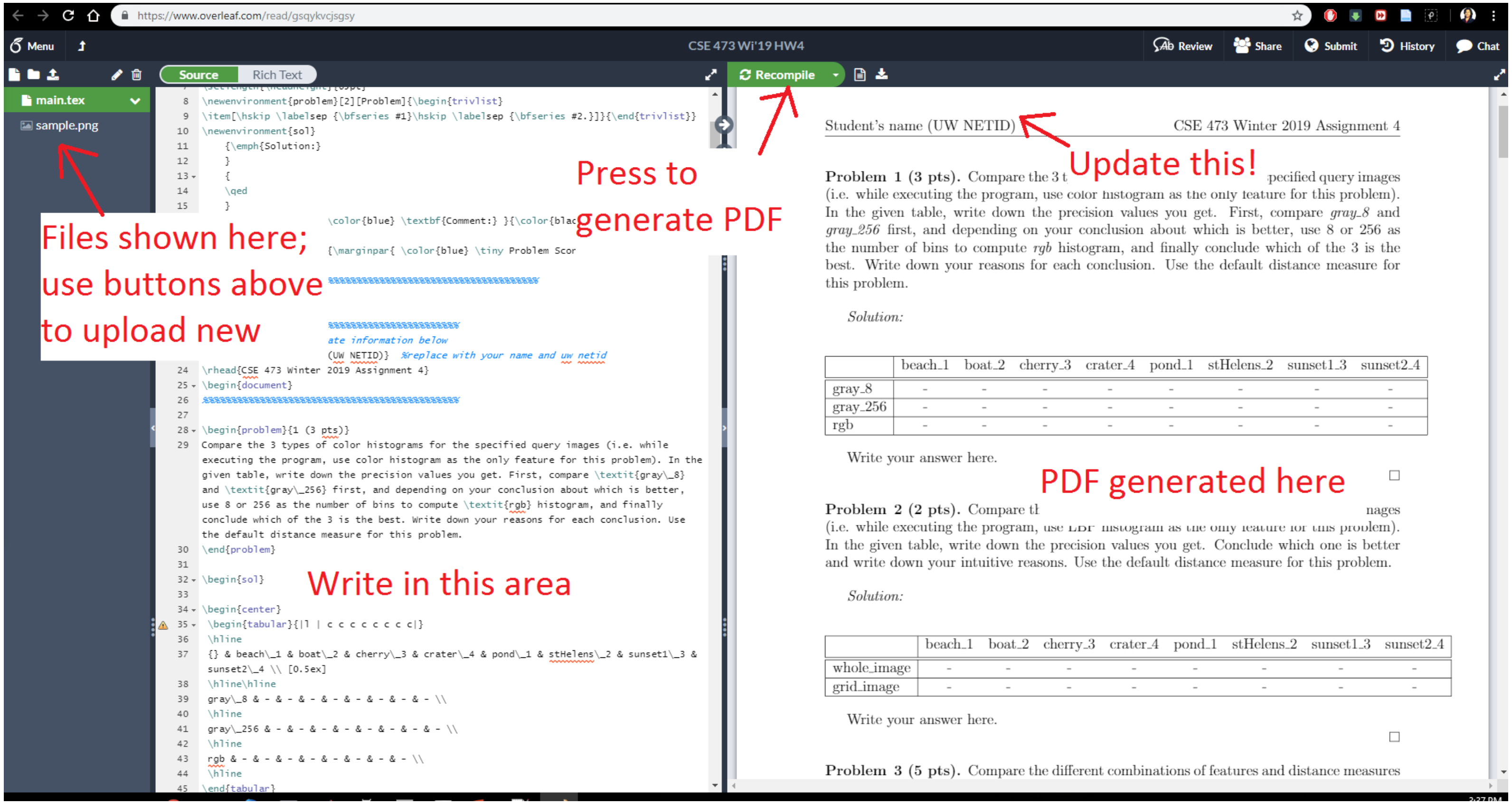
Introduction

- Given a database of images (40 images belonging to 8 classes with 5 images per class) and a query image, retrieve the images from the database which are most similar to the query image.

- In python, an image with height H, width W and number of channels C (3 for RGB images) is represented by a 3D matrix of shape H x W x C.
  - To access a particular pixel (h,w) of a particular channel (c) of the 'image' variable, use image[h,w,c]

- Write your code in the code file (*main.py*) and your comments and results on the Overleaf (Latex) document as a report (*main.pdf*). Submit these 2 files.

Overleaf



Files shown here; use buttons above to upload new

Press to generate PDF

Update this!

Write in this area

PDF generated here

```
\newenvironment{problem}[2][Problem]{\begin{trivlist}
\item[\hskip \labelsep {\bfseries #1}\hskip \labelsep {\bfseries #2.}]}{\end{trivlist}}
\newenvironment{sol}
    {\emph{Solution:}
    }
    {
    \qed
    }
                  \color{blue} \textbf{Comment:} }{\color{blac

                  {\marginpar{ \color{blue} \tiny Problem Scor

                  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


                  %%%%%%%%%%%%%%%%%%%%%%%%%%%
                  ate information below
                  (UW NETID)}  %replace with your name and uw netid
\rhead{CSE 473 Winter 2019 Assignment 4}
\begin{document}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

\begin{problem}{1 (3 pts)}
Compare the 3 types of color histograms for the specified query images (i.e. while executing the program, use color histogram as the only feature for this problem). In the given table, write down the precision values you get. First, compare \textit{gray\_8} and \textit{gray\_256} first, and depending on your conclusion about which is better, use 8 or 256 as the number of bins to compute \textit{rgb} histogram, and finally conclude which of the 3 is the best. Write down your reasons for each conclusion. Use the default distance measure for this problem.
\end{problem}

\begin{sol}

\begin{center}
 \begin{tabular}{|l | c c c c c c c c|}
 \hline
 {} & beach\_1 & boat\_2 & cherry\_3 & crater\_4 & pond\_1 & stHelens\_2 & sunset1\_3 & sunset2\_4 \\ [0.5ex]
 \hline\hline
 gray\_8 & - & - & - & - & - & - & - & - \\
 \hline
 gray\_256 & - & - & - & - & - & - & - & - \\
 \hline
 rgb & - & - & - & - & - & - & - & - \\
 \hline
\end{tabular}
```

Student's name (UW NETID) CSE 473 Winter 2019 Assignment 4

Problem 1 (3 pts). Compare the 3 t pecified query images (i.e. while executing the program, use color histogram as the only feature for this problem). In the given table, write down the precision values you get. First, compare *gray_8* and *gray_256* first, and depending on your conclusion about which is better, use 8 or 256 as the number of bins to compute *rgb* histogram, and finally conclude which of the 3 is the best. Write down your reasons for each conclusion. Use the default distance measure for this problem.

Solution:

	beach_1	boat_2	cherry_3	crater_4	pond_1	stHelens_2	sunset1_3	sunset2_4
gray_8	-	-	-	-	-	-	-	-
gray_256	-	-	-	-	-	-	-	-
rgb	-	-	-	-	-	-	-	-

Write your answer here. □

Problem 2 (2 pts). Compare tl nages (i.e. while executing the program, use LBP histogram as the only feature for this problem). In the given table, write down the precision values you get. Conclude which one is better and write down your intuitive reasons. Use the default distance measure for this problem.

Solution:

	beach_1	boat_2	cherry_3	crater_4	pond_1	stHelens_2	sunset1_3	sunset2_4
whole_image	-	-	-	-	-	-	-	-
grid_image	-	-	-	-	-	-	-	-

Write your answer here. □

Problem 3 (5 pts). Compare the different combinations of features and distance measures

Instructions

- Compute a feature vector for each image. Use two types of features:

  - ***Color Histogram***: Compute 3 variations
    - histogram with grayscale values and 8 bins
    - histogram with grayscale values and 256 bins
    - histogram with RGB values and *n* bins, where n depends on which of the above 2 you found better

  - ***LBP (Local Binary Pattern) Histogram***: Compute 2 variations
    - histogram from the entire image
    - histogram by concatenating histograms from each 16x16 region

Instructions

- Use a distance measure to compare the feature vectors and retrieve images in the order of increasing distance (images with lesser distance values are more similar to the query image)
  - Compare different distance measures as asked in the report.

- To run the starter code, open your terminal in the homework folder and type:

```
$ python main.py -q beach_1
```

  - Further instructions are written in the starter code.
  - Questions are directly written in the report. Write your solutions where asked.

Sample (random) output for query beach_1