

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)

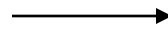
features → analysis

Low-Level



original image

Canny
edge
operator



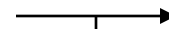
edge image

Mid-Level (Lines and Curves)



edge image

ORT
line &
circle
extraction



data
structure

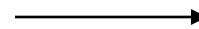


circular arcs and line segments³

Low- to High-Level



low-level



edge image

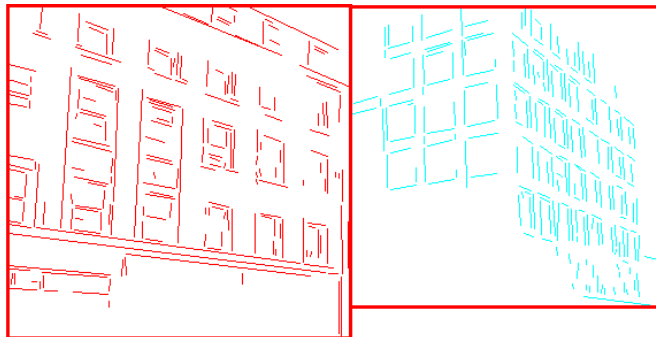
mid-level



consistent
line clusters



high-level



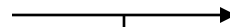
Building Recognition

Mid-level (Regions)

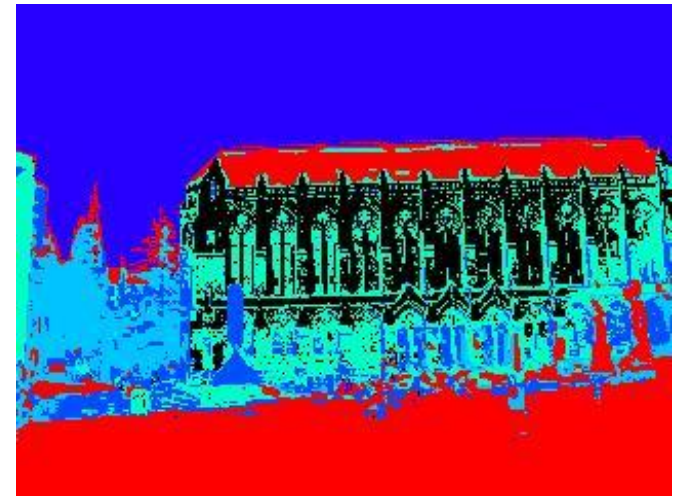


original color image

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

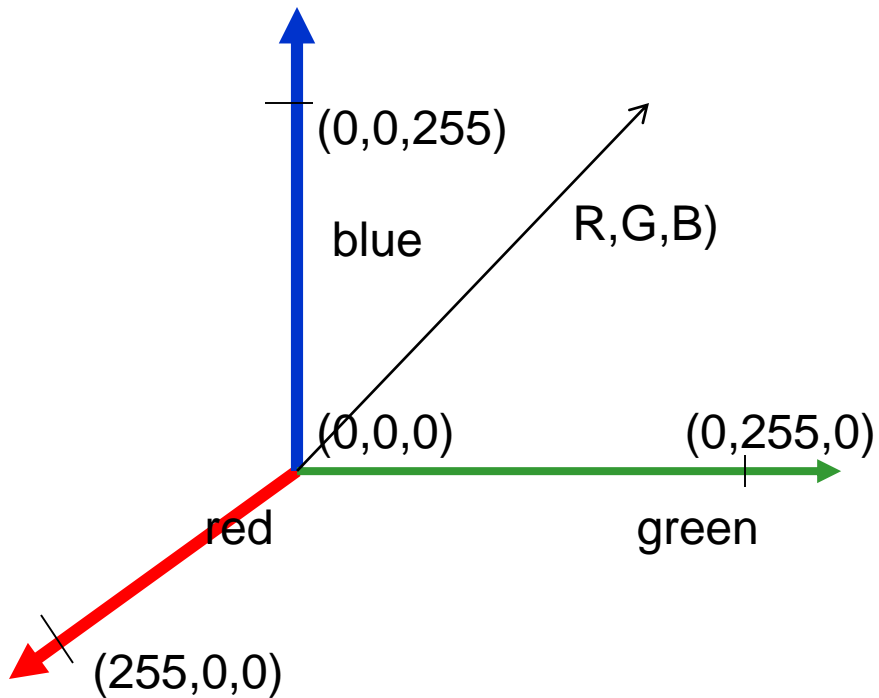
- Retrieve images based on their color and texture attributes
 - Color histograms
 - Texture histograms
 - Both
- This is content-based image retrieval:
CBIR.

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:

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

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

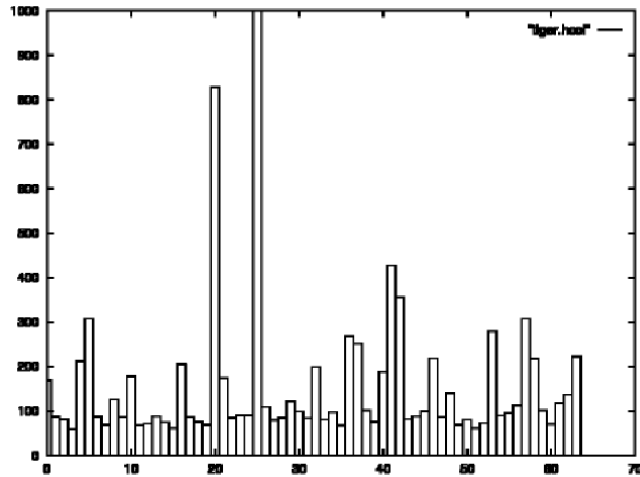
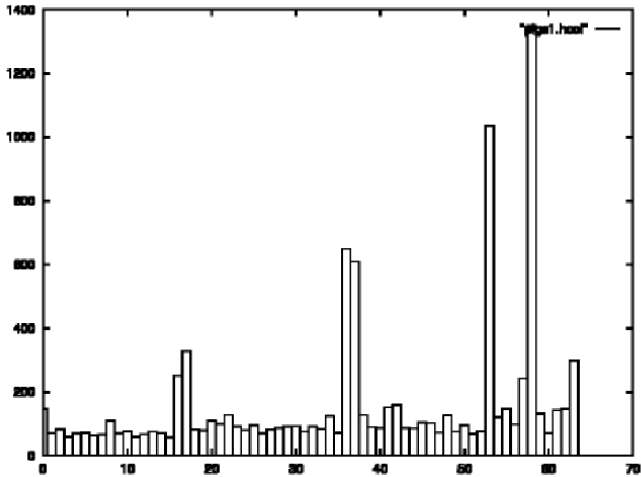
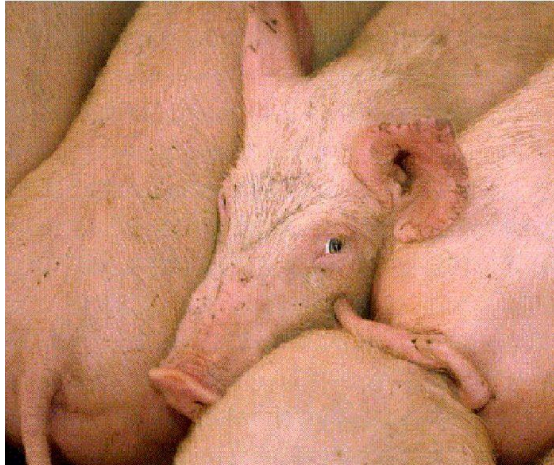
Histograms

- 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.

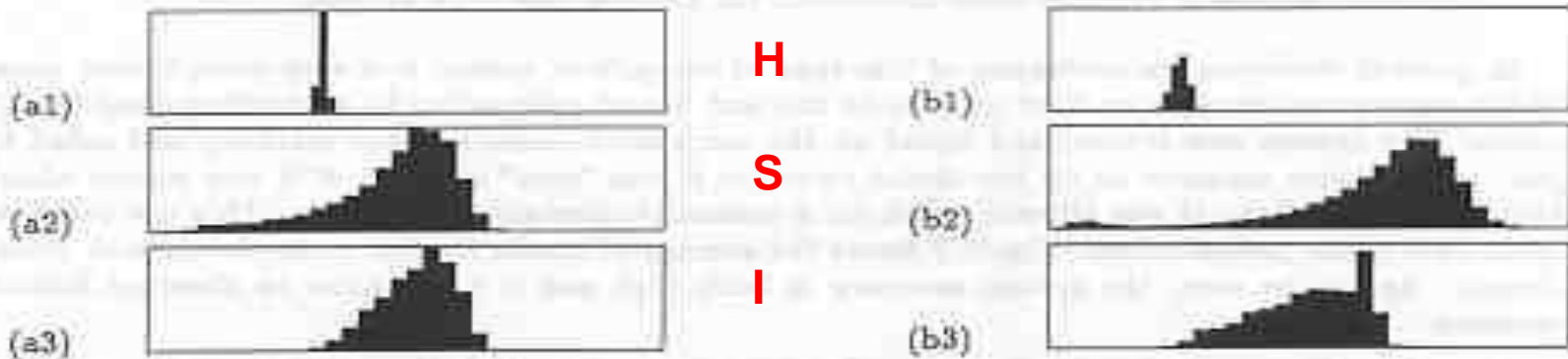
Histograms of two color images



How to make a color histogram

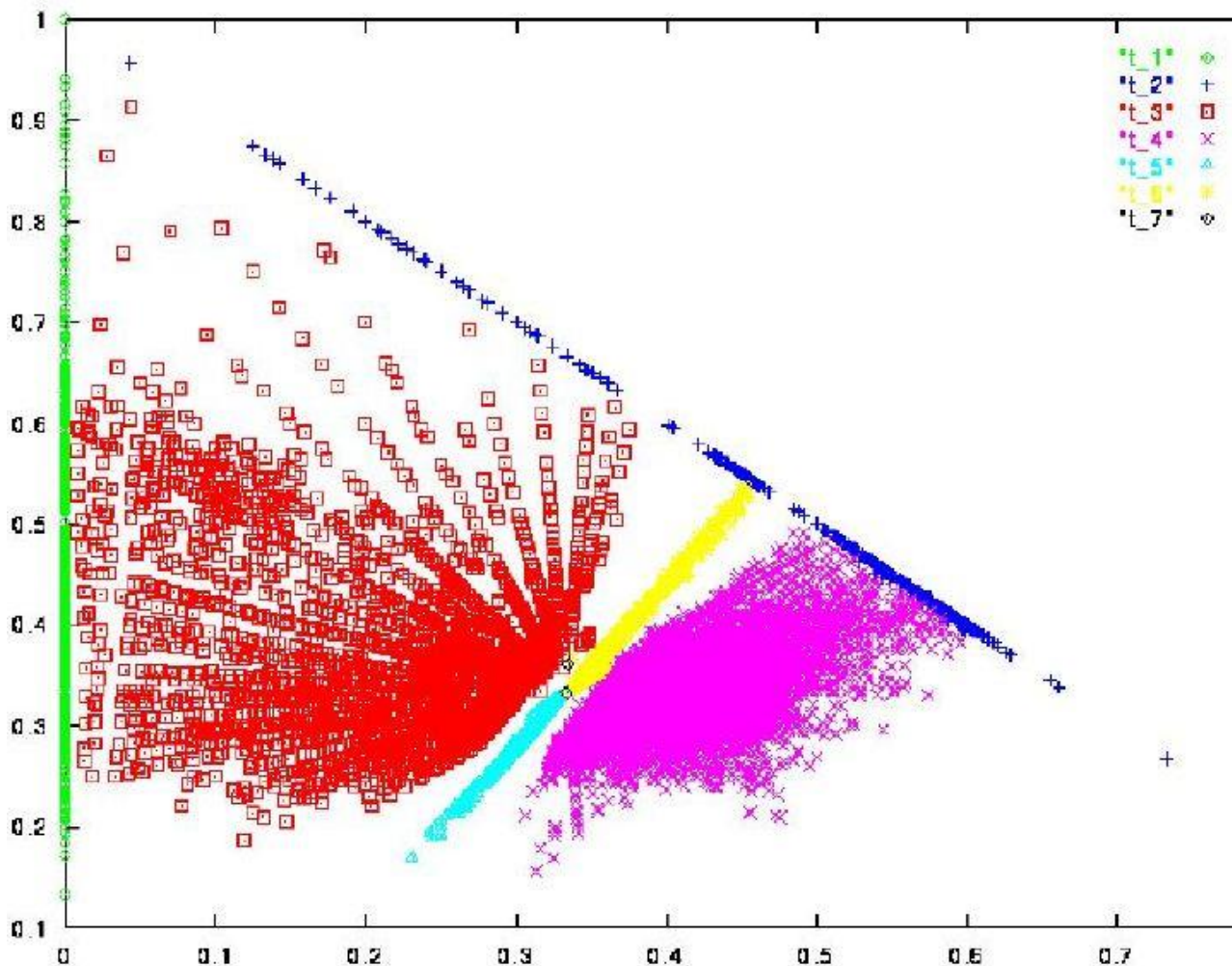
- Make a single 3D histogram. (BIG)
- Make 3 histograms and concatenate them.
(Most Common and what Bindita has)
- Create a single pseudo color between 0 and 255 by using 3 bits of R, 3 bits of G and 2 bits of B
(Old)
- Use normalized color space and 2D histograms.

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)



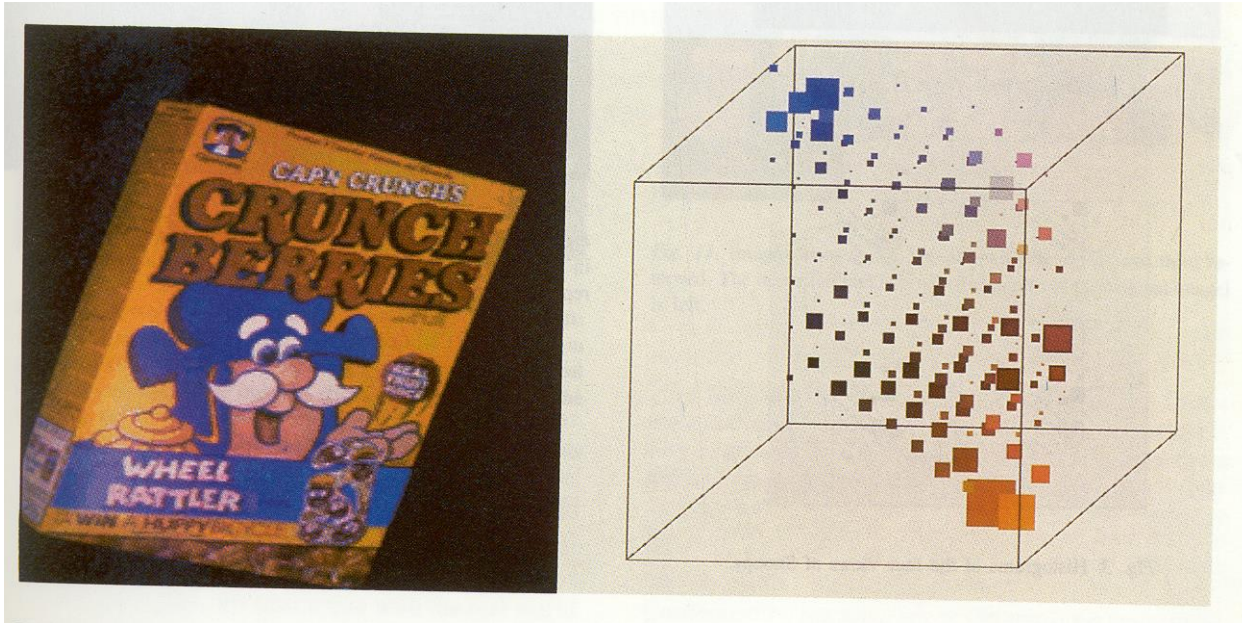
Purple region shows skin color samples from several people. Blue and yellow regions show skin in shadow or behind a beard.

Finding a face in video frame



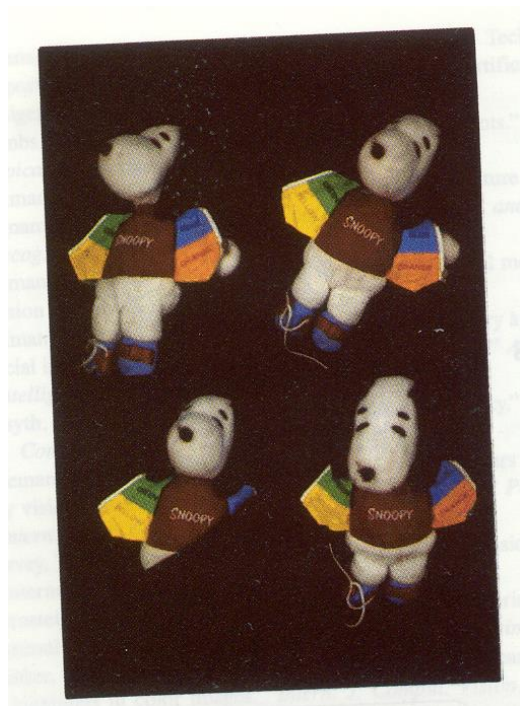
- (left) input video frame
- (center) pixels classified according to RGB space
- (right) largest connected component with aspect similar to a face (all work contributed by Vera Bakic)

(from Swain and Ballard)

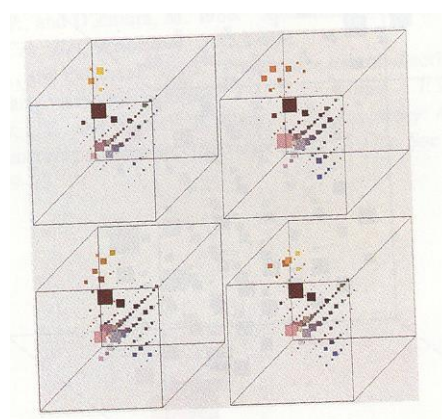


cereal box image

3D color histogram



Four views of Snoopy



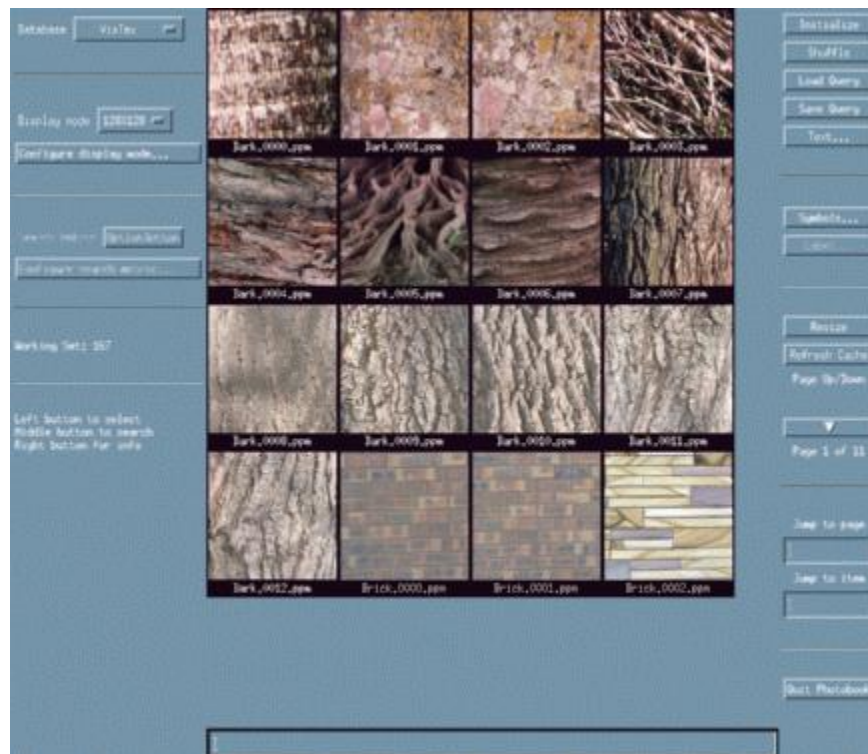
Histograms

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

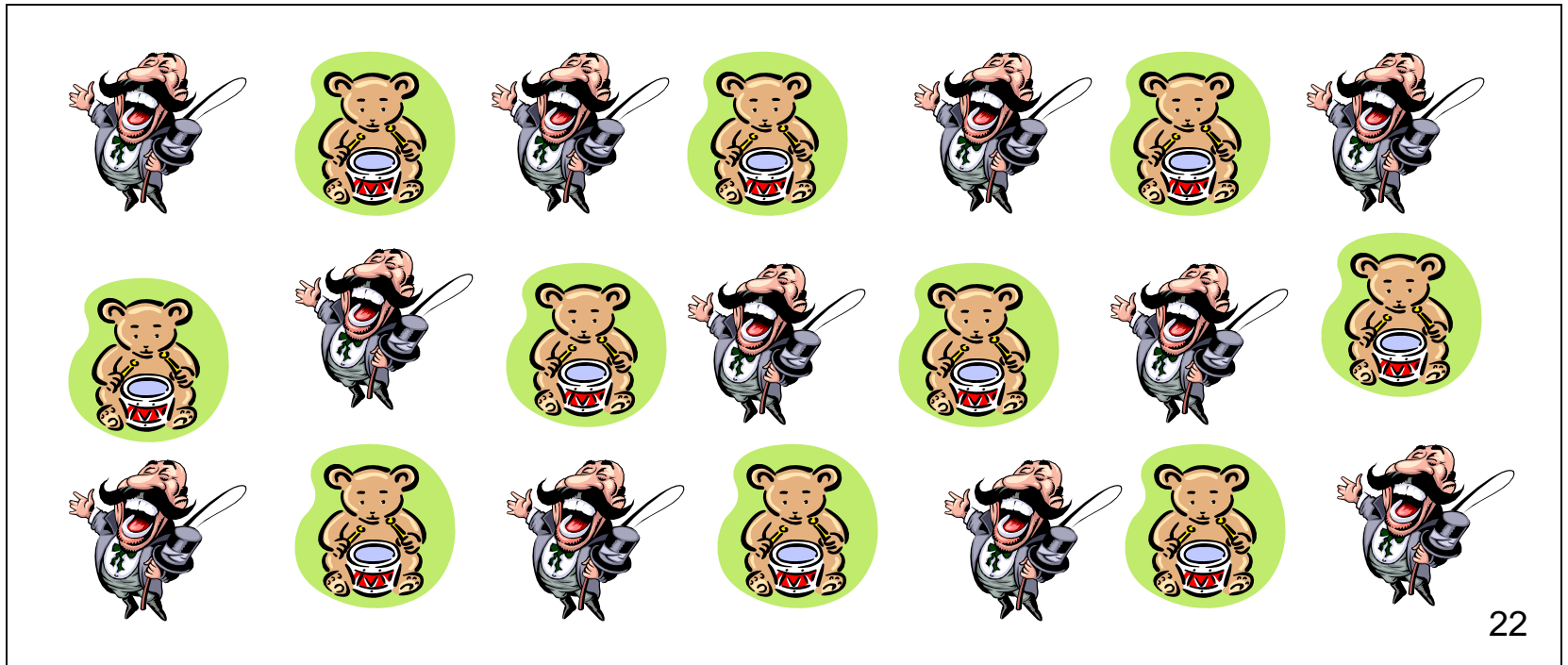
- Color is well defined.
- But what *is* texture?



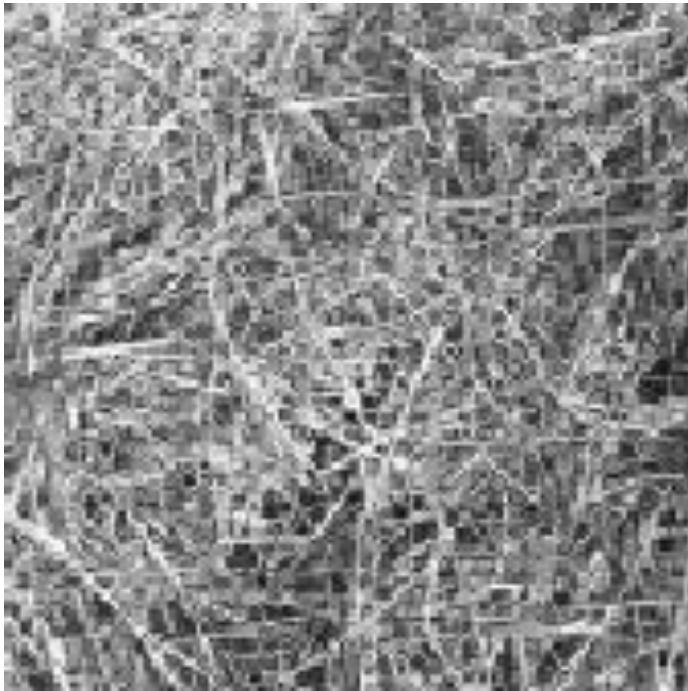
Structural Texture

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

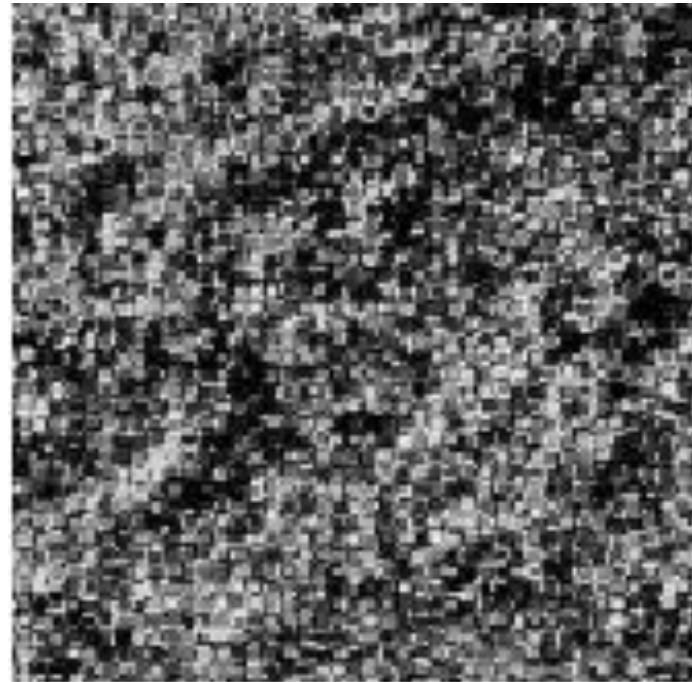
Structural approach: a set of texels in some regular or repeated pattern



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

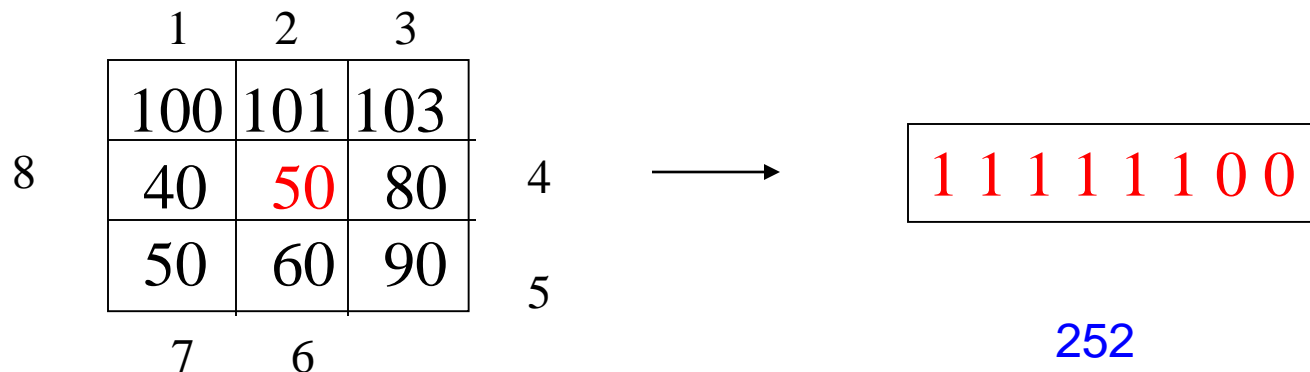


Thresholded
Edge Image



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.



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**.
- $D((x_1, x_2, \dots, x_n), (y_1, y_2, \dots, y_n)) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_n - y_n)^2}$
- 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

The screenshot shows the Fids demo interface. At the top, there is a grid of six images. The first image in the top-left corner is highlighted with a red border. Below the grid, there are navigation buttons: "Random", "Go", "ZoomIn", and a right-pointing arrow. To the right of these buttons, it says "Found 191 matches. Displaying 1 - 6". Below the navigation buttons, there are two columns of distance measures. The first column has checkboxes for "ColorHistL14x4x4", "ColorHist8x8x8", "SobelEdgeHist", "LBPHist", "fleshiness", and "Wavelets". The second column has a "loose ... strict" slider for each measure, with "5" next to it. To the right of the sliders, there is a radio button menu with "And", "Or", and "Sum" options. At the bottom, it says "Server Connected".

Found 191 matches. Displaying 1 - 6

distance measures	loose ... strict
<input type="checkbox"/> ColorHistL14x4x4	5
<input type="checkbox"/> ColorHist8x8x8	5
<input type="checkbox"/> SobelEdgeHist	5
<input checked="" type="checkbox"/> LBPHist	5
<input type="checkbox"/> fleshiness	5
<input type="checkbox"/> Wavelets	5

Server Connected

Example

Fids demo

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

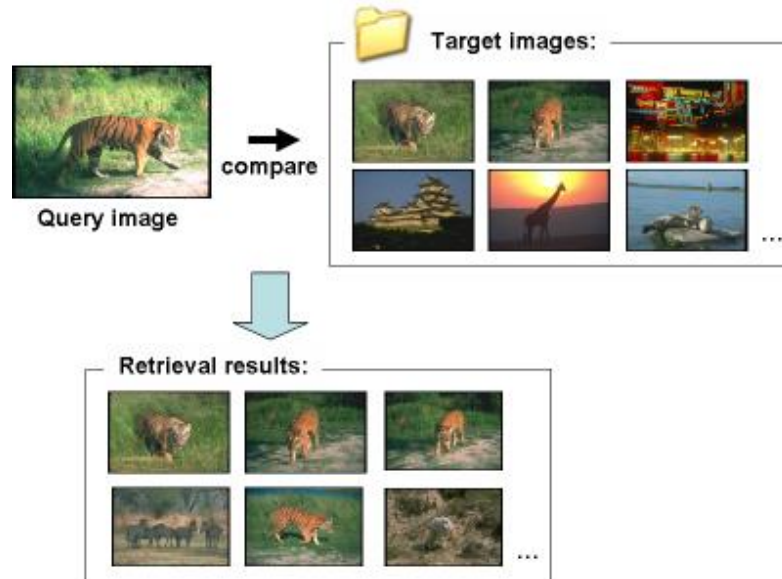
The screenshot shows the Fids demo interface. At the top, there are six image thumbnails arranged in a 2x3 grid. The top-left image is highlighted with a red border. Below the images are two buttons: "Put In Cart" and "Check Out". At the bottom, there are navigation buttons: "Random", "Go", "ZoomIn", and a right-pointing arrow. To the right of these buttons, it says "Found 119 matches. Displaying 1 - 6". Below the navigation buttons, there are two columns of controls. The first column is labeled "distance measures" and contains six checkboxes: "ColorHistL14x4x4", "ColorHist8x8x8", "SobelEdgeHist", "LBPHist" (checked), "fleshiness", and "Wavelets". The second column is labeled "loose ... strict" and contains six sliders, each with a vertical line and the number "5" to its right. To the right of the sliders, there are three radio buttons: "And" (selected), "Or", and "Sum". At the bottom right, there is a box with the text "A double click on an image means:" and two radio buttons: "Set query / Go" (selected) and "Zoom in". At the bottom left, it says "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 2022 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 $H \times W \times C$.
 - 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*).
- <https://www.overleaf.com/learn/latex/Tutorials>
- Turn-in instructions will be given.

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

