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

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

SYSTEMS

Some Systems You Can Try

Corbis Stock Photography and Pictures

http://pro.corbis.com/

- Corbis sells high-quality images for use in advertising, marketing, illustrating, etc.
- Search is entirely by keywords.
- Human indexers look at each new image and enter keywords.
- A thesaurus constructed from user queries is used.

QBIC

IBM's QBIC (Query by Image Content)

http://wwwqbic.almaden.ibm.com

- The first commercial system.
- Uses or has-used color percentages, color layout, texture, shape, location, and keywords.

Blobworld

UC Berkeley's Blobworld

http://elib.cs.berkeley.edu/photos/blobworld

- Images are segmented on color plus texture
- User selects a region of the query image
- System returns images with similar regions
- Works really well for tigers and zebras

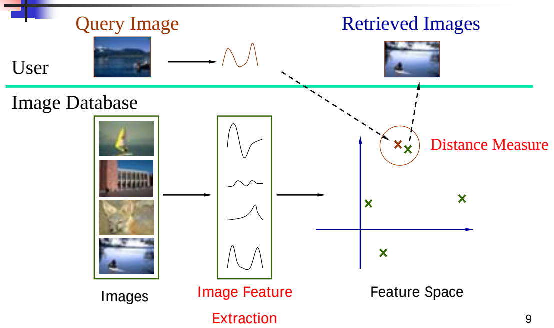
Ditto

Ditto: See the Web

http://www.ditto.com

- Small company
- Allows you to search for pictures from web pages

Image Features / Distance Measures

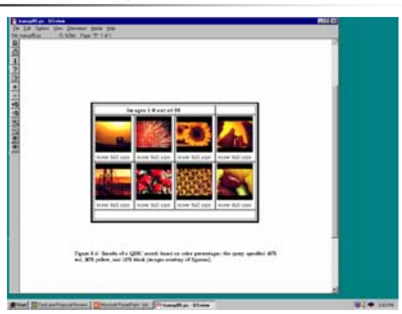


Features

- Color (histograms, gridded layout, wavelets)
- Texture (Laws, Gabor filters, local binary partition)
- 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



QBIC's Histogram Similarity

The QBIC color histogram distance is:

$$d_{hist}(I,Q) = (h(I) - h(Q))^T \mathbf{A} \, (h(I) - h(Q))$$

- $h(I)$ is a K-bin histogram of a database image
- $h(Q)$ is a K-bin histogram of the query image
- A is a K x K similarity matrix

Similarity Matrix

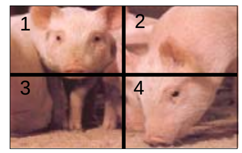
	R	G	B	Y	C	V
R	1	0	0	.5	0	.5
G	0	1	0	.5	.5	0
B	0	0	1		?	
Y				1		
C		?			1	
V						1

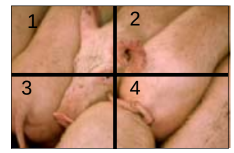
How similar is blue to cyan?

13

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?

14

Color Layout (IBM's Gridded Color)

15

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).

16

Laws Texture

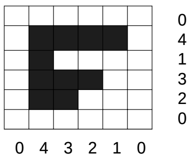
17

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.

18

Global Shape Properties: Projection Matching

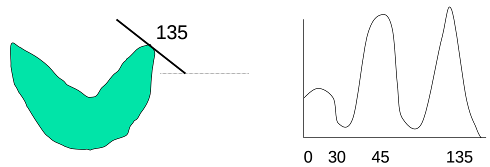


Feature Vector
(0,4,1,3,2,0,0,4,3,2,1,0)

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

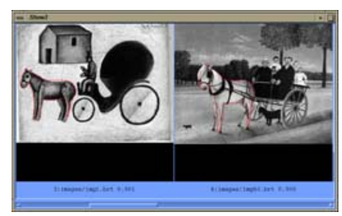
  The distance between query shape and image shape has two components:

  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

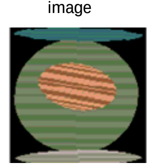
Regions and Relationships

- 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

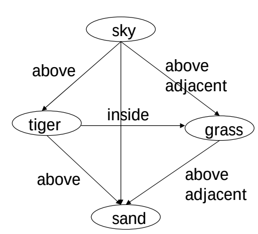
Like what?

Tiger Image as a Graph

image



abstract regions



Object Detection: Rowley's Face Finder

1. convert to gray scale
2. normalize for lighting*
3. histogram equalization
4. apply neural net(s) trained on 16K images

What data is fed to the classifier?

32 x 32 windows in a pyramid structure

* Like first step in Laws algorithm, p. 220

25

Fleck and Forsyth's Flesh Detector

See Transparencies

The "Finding Naked People" Paper

- Convert RGB to HSI
- Use the intensity component to compute a texture map
  texture = med2 (| I - med1(I) |)   median filters of radii 4 and 6
- If a pixel falls into either of the following ranges, it's a potential skin pixel

  texture < 5, 110 < hue < 150, 20 < saturation < 60
  texture < 5, 130 < hue < 170, 30 < saturation < 130

Look for LARGE areas that satisfy this to identify pornography.

26

Wavelet Approach

Idea: use a wavelet decomposition to represent images

What are wavelets?

- compression scheme
- uses a set of 2D basis functions
- representation is a set of coefficients, one for each basis function

27

Jacobs, Finkelstein, Salesin Method for Image Retrieval (1995)

1. Use YIQ color space
2. Use Haar wavelets
3. 128 x 128 images yield 16,384 coefficients x 3 color channels
4. Truncate by keeping the 40-60 largest coefficients (make the rest 0)
5. Quantize to 2 values (+1 for positive, -1 for negative)

28

JFS Distance Metric

$$d(I,Q) = w_{00} \mid Q[0,0] - I[0,0] \mid + \sum_{ij} w_{ij} \mid Q'[i,j] - I'[i,j] \mid$$

where the w's are weights,

Q[0,0] and I[0,0] are scaling coefficients related to average image intensity,

Q'[i,j] and I'[i,j] are the truncated, quantized coefficients.

29

Experiments

20,558 image database of paintings

20 coefficients used

User "paints" a rough version of the painting he /she wants on the screen.

See Video

30

Relevance Feedback

In real interactive CBIR systems, the user should be allowed to interact with the system to "refine" the results of a query until he/she is satisfied.

Relevance feedback work has been done by a number of research groups, e.g.

- The Photobook Project (Media Lab, MIT)
- The Leiden Portrait Retrieval Project
- The MARS Project (Tom Huang's group at Illinois)

Information Retrieval Model*

- An IR model consists of:
  - a document model
  - a query model
  - a model for computing similarity between documents and the queries
- Term (keyword) weighting
- Relevance Feedback

*from Rui, Huang, and Mehrotra's work

Term weighting

- Term weight
  - assigning different weights for different keyword(terms) according their relative importance to the document
- define w_{ik} to be the weight for term t_k ,k=1,2,…,N, in the document *i*
- document *i* can be represented as a weight vector in the term space

$$D_i = [w_{i1}; w_{i2}; \ldots; w_{iN}]$$

Term weighting

- The query Q also is a weight vector in the term space

$$Q = [w_{q1}; w_{q2}; \ldots; w_{qN}]$$

- The similarity between D and Q

$$Sim(D,Q) = \frac{D \cdot Q}{\|D\| \, \|Q\|}$$

Using Relevance Feedback

- The CBIR system should automatically adjust the weight that were given by the user for the relevance of previously retrieved documents
- Most systems use a statistical method for adjusting the weights.

The Idea of Gaussian Normalization

- If all the relevant images have similar values for component *j*
  - the component *j* is relevant to the query
- If all the relevant images have very different values for component *j*
  - the component *j* is not relevant to the query
- the inverse of the standard deviation of the related image sequence is a good measure of the weight for component *j*
- the smaller the variance, the larger the weight

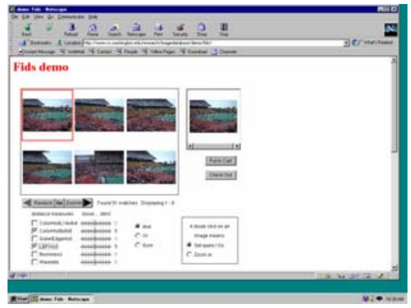
Leiden Portrait System

The Leiden Portrait Retrieval System is an example of the use of relevance feedback.

http://ind156b.wi.leidenuniv.nl:2000/

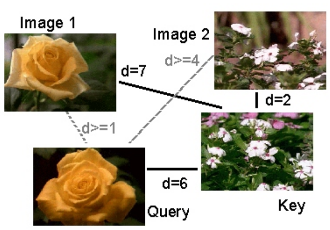
Andy Berman's FIDS System

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.



Andy Berman's FIDS System:

Bare-Bones Triangle Inequality Algorithm

Offline

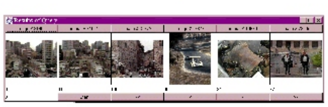
1. Choose a small set of key images
2. Store distances from database images to keys

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

Andy Berman's FIDS System:

Bare-Bones Algorithm with Multiple Distance Measures

Offline

1. Choose key images for each measure
2. Store distances from database images to keys for all measures

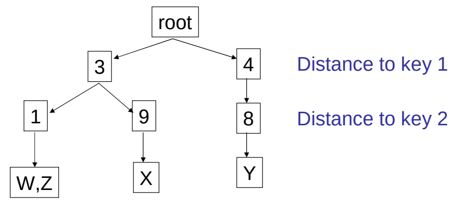
Online (given query Q)

1. Calculate lower bounds for each measure
2. Combine to form lower bounds for composite measures
3. Continue as in single measure algorithm

Andy Berman's FIDS System:

Triangle Tries

A triangle trie is a tree structure that stores the distances from database images to each of the keys, one key per tree level.



Andy Berman's FIDS System:

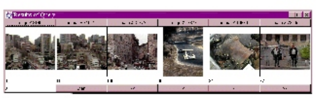
Triangle Tries and Two-Stage Pruning

- First Stage: Use a short triangle trie.
- Second Stage: Bare-bones algorithm on the images returned from the triangle-trie stage.

The quality of the output is the same as with the bare-bones algorithm itself, but execution is faster.

Andy Berman's FIDS System:

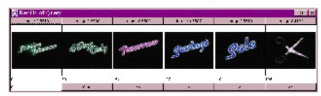
Flexible Image Database System: Example



\# of images in system: 37,748
Depth of triangle trie: 6
\# of images eliminated by trie: 30,300
\# images eliminated by second-stage: 7429
19 images remaining, as before

Andy Berman's FIDS System:

Flexible Image Database System: Example



Example from our system using a combination color+texture measure
\# images in system: 37,748
\# images from color trie: 3,676
\# images from texture trie: 497
\# images in merged set: 3,785
\# images eliminated: 33,963

Andy Berman's FIDS System:

Performance on a Pentium Pro 200-mHz

Step 1. Extract features from query image. ($.02s \le t \le .25s$)

Step 2. Calculate distance from query to key images.
($1\mu s \le t \le .8ms$)

Step 3. Calculate lower bound distances.
($t \approx 4ms$ per 1000 images using 35 keys, which is about 250,000 images per second.)

Step 4. Return the images with smallest lower bound distances.

Andy Berman's FIDS System:

Speed Comparisons

Image-query comparisons per second

Distance Measure	Direct Calculation	Bare-bones^⊛ Algorithm	Two-stage^* Pruning Algorithm
Sobel	24937	250,000	12,000,000
Color	2174	250,000	2,200,000
Wavelet	115	250,000	950,000
LBP	3623	250,000	700,000
Flesh	833,333	250,000	650,000

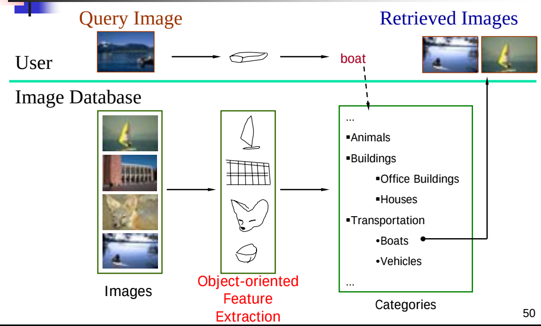
^2 digits of accuracy, not including key-comparison times.
⊛ 35 keys
* Best trie found, depth and binning varies, threshold chosen by hand

Weakness of Low-level Features

- Can't capture the high-level concepts

49

Current Research Objective



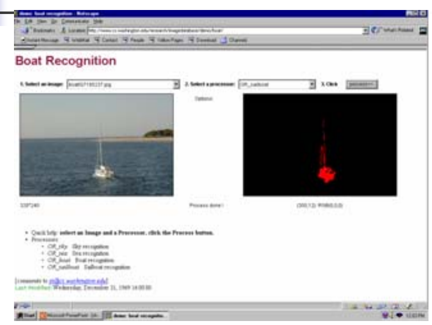
50

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

51

Boat Recognition



52

Vehicle Recognition

53

Building Recognition

54

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.

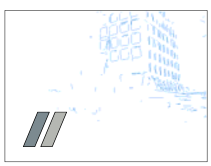
55

Color-CLC

- Color feature of lines: color pair (c_1,c_2)
- Color pair space:
  - RGB (256^3*256^3) Too big!
  - Dominant colors (20*20)
- Finding the color pairs:
  - One line $\rightarrow$ Several color pairs
- Constructing Color-CLC: use clustering

56

Color-CLC



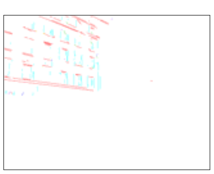
57

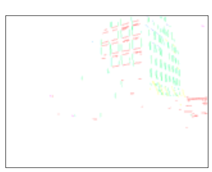
Orientation-CLC

- The lines in an Orientation-CLC are parallel to each other in the 3D world
- The parallel lines of an object in a 2D image can be:
  - Parallel in 2D
  - Converging to a vanishing point (perspective)

58

Orientation-CLC

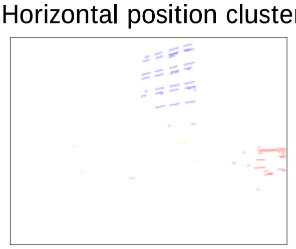




59

Spatially-CLC

- Vertical position clustering
- Horizontal position clustering

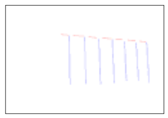


60

Building Recognition by CLC

Two types of buildings → Two criteria

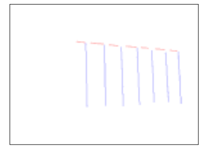
- Inter-relationship criterion
- Intra-relationship criterion



61

Inter-relationship criterion

$(N_{c1} > T_{i1}$ or $N_{c2} > T_{i1})$ and $(N_{c1}+N_{c2}) > T_{i2}$



N_{c1} = number of intersecting lines in cluster 1

N_{c2} = number of intersecting lines in cluster 2

62

Intra-relationship criterion

$|S_o| > T_{j1}$ or $w(S_o) > T_{j2}$

S_0 = set of heavily overlapping lines in a cluster

63

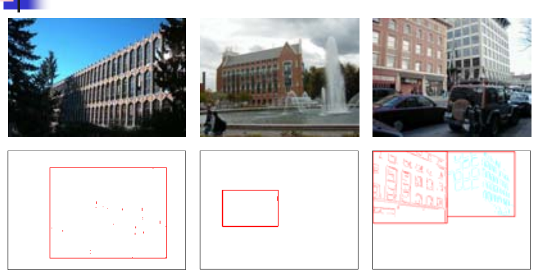
Experimental Evaluation

- 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

64

Experimental Evaluation

Well-Patterned Buildings



65

Experimental Evaluation

Non-Well-Patterned Buildings

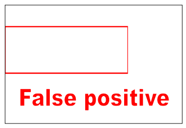
66

Experimental Evaluation

Non-Well-Patterned Non-Buildings



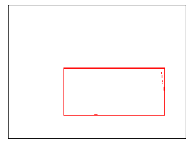
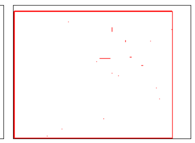
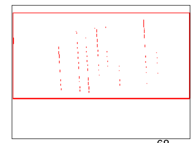


Experimental Evaluation

Well-Patterned Non-Buildings (false positives)



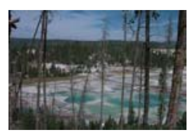
Experimental Evaluation (CBIR)

	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






Future Work

- Future Work
  - Constructing hierarchically structured clusters
  - Using CLC on other objects
  - Combining CLC with other features
  - Developing a learning approach using hierarchical, multiple classifiers (Chou 2000)