Searching non-text information objects

Non-text digital objects
- Music
- Speech
- Images
- 3D models
- Video
- ?

Ways to query for something
1. Query by category/theme
   - easiest - work done ahead of time
2. Query by describing content
   - text-based query
   - text-based retrieval?
3. Query by example
   - “similar to”
   - imprecise example - sketch
   - query text docs and non-text objects with 2
   - don’t often do doc search by 3
   - big move to do music, images by 3

Query by describing content
- text-based queries
- where get text-based content?
  - author labels
  - metadata
  - URLs
  - text near imbedded objects
    - html pages
  - group tagging
    - folksonomy
    - Flickr

Query by example
- How represent objects?
  - features of a class of objects (e.g. image)
  - how compare features?
  - what data structures?
  - what computational methods?
- Issues
  - large number of objects
  - accuracy of representation
  - large size of representation
  - complexity of computations

Features
- typically vector of numbers characterizing object representation
- “similar to” = close in vector space
  - threshold
  - Euclidean distance?
  - other choices for distance metric
Example: content-based image search

First example method: color histogram

- k colors
- histogram: % pixels each color
- k×k matrix A of color similarity weights
- histogram defines feature vectors
- \(\text{dist}_{\text{histo}}(x, y) = (x-y)^tA(x-y)\)
  \[= \sum_{i=1}^{k} \sum_{j=1}^{k} a_{ij}(x_i - y_i)(x_j - y_j)\]
- cross-talk: quadratic terms needed
  - not Euclidean distance

Second example method: a region-based representation

- region-based features of images
- query processed in same way as collection
- space-conscious: use bit vectors
- levels of representation:
  - store bit vector for each region
  - store bit vector for each image
- get close candidates: compare image bit vectors
- compare top k candidates using region bit vectors

Processing images of collection & query

- segment into homogeneous regions
  - 14 dimensional feature vectors
- threshold and transform
  - high-dimensional bit vectors - store
  - XOR for distance between regions
- build image feature vector
  - n region bit-vectors + weights \(\rightarrow\) 1 m-dimensional real-valued image feature vector
  - \(L_1\) distance between feature vectors
- transform image vector
  - one high-dimensional bit vector for image - store

color histograms: reducing complexity

- compute RED\(_{\text{avg}}\), GREEN\(_{\text{avg}}\), BLUE\(_{\text{avg}}\):
  - over all pixels
- use to construct 3D-vector
- use Euclidean distance
- get close candidates
- examine close candidates with full histogram metric

color histograms: observations

- works for certain types of images
  - sunset canonical example
- color histogram global property

- this only small part of work:
  QBIC system, IBM, 1995
Components region feature vector

- color moments - 9 dim
  - role similar to histogram
- bounding box region - 5 dim
  - ln(aspect ratio)
  - ln (bounding box size)
  - density = # pixels / bounding box size
  - centroid x
  - centroid y

weight regions proportional to sq. root of area

Observations: region based

- Example of one regional method
  - lots of research, lots of places!
- This method uses sampling heavily
  - produce bit vectors
- Part of larger project - multiple media
  - CASS, Princeton, 2004

Third example method: Combining simple ideas

- Goals
  - reduce search space
  - reduce disk I/O cost
- Simple ideas
  - K-means clustering of image database
  - B+ trees
  - heuristic search limits
- New ideas
  - search beyond cluster containing query image
  - limit search within each cluster

Image representation

- Input: non-texture RGB images
- Process
  - resize to uniform 128x128 pixels
  - transform to different color space
    - relate to human perception
  - Apply Daubechies wavelet tranformation
    - use several applications
    - obtain 964 dimensional feature vector

Data space representation

- Cluster data space using K-means
  - search for "most cost effective" K
    - cluster validity indexes
    - majority vote
- Find cluster centroids
- For each cluster build a B+ tree
  - B+ tree represent each image in cluster
  - search key for ith image in cluster is distance of feature vector of ith image to cluster center
Search space for query

- don’t search things know probably too far
- don’t limit search to just cluster containing query

- Chose similarity threshold $c$ for data set
- search images in outer shell of cluster
  - range $d-c$ to $d+c$ for $d$=distance query to its centroid

- Same principle whether q in boundry of a cluster or not
  - but use different $c$ : $c_{same}$, $c_{diff}$

Choosing $c_{same}$, $c_{diff}$

- Initially
  $c_{same} = \text{avg. of distances all images to their centers}$
  $c_{diff} = 0$
- iteratively search for values give best gain
  - factors in gain
    - improved average distance found
    - reduced size of search space
    - compared to K-means
    - with linear search bounding
      - shortest distance
      - largest search space

Results

- find best 5 matches to a query image
- most interesting result: resources used versus value find
- sample numbers (1000 images):
  - average distance
    - K-means & B+ tree 51.887
    - K-means 52.212
    - linear search 50.881
  - size search space
    - K-means & B+ tree 147
    - K-means 92.39
    - linear search 900

Other Results

- visually:
  - not beating other methods for image quality
- calculate precision of top 5 returns
  - 10 pre-existing image categories
    - crude
  - sample numbers:
    - them 0.568, linear search 0.576

Observations

- dynamic capability of B+ trees
- color based
- no region analysis of images
- image representation and data space representation independent

“Integrating wavelets with clustering and indexing for effective content-based image retrieval” 2012

Fourth example method: Image ranking

- given similarity measures
- use PageRank style
- define
  \[ \mathbf{v} = \alpha(1/n) + (1-\alpha)\mathbf{Sv} \]
- where
  - $n$ is the number of images to be ranked
  - $\mathbf{S}$ is a matrix of image-image similarities
    - column normalized, symmetric
  - $\mathbf{v}$ is the vector of VisualRanks
  - $\alpha$ is the usual parameter
**Observations: Image rank**

- intention to use on images returned by other means
  - e.g. text based
- graph undirected
- tested on Google image search
  - VisualRank, Google, 2008
- Deployed?

**Table 1: Relevancy Study**

<table>
<thead>
<tr>
<th>“Irrelevant” images per product query</th>
<th>VisualRank</th>
<th>Google</th>
</tr>
</thead>
<tbody>
<tr>
<td>Among top 10 results</td>
<td>0.47</td>
<td>2.82</td>
</tr>
<tr>
<td>Among top 5 results</td>
<td>0.30</td>
<td>1.31</td>
</tr>
<tr>
<td>Among top 3 results</td>
<td>0.20</td>
<td>0.81</td>
</tr>
</tbody>
</table>

**Image search: Summary of techniques**

- Techniques seen
  - aggregate/average features
  - sample
  - course screening followed by more accurate
- **Goals**
  - reduce dimension
  - reduce complexity of distance metric
  - reduce space
Image search:
Commercial search engines

• Use everything you can afford to use
• Text still king!? 