RuSSIR
Russian Summer School
in Information Retrieval
2008

LABS hp

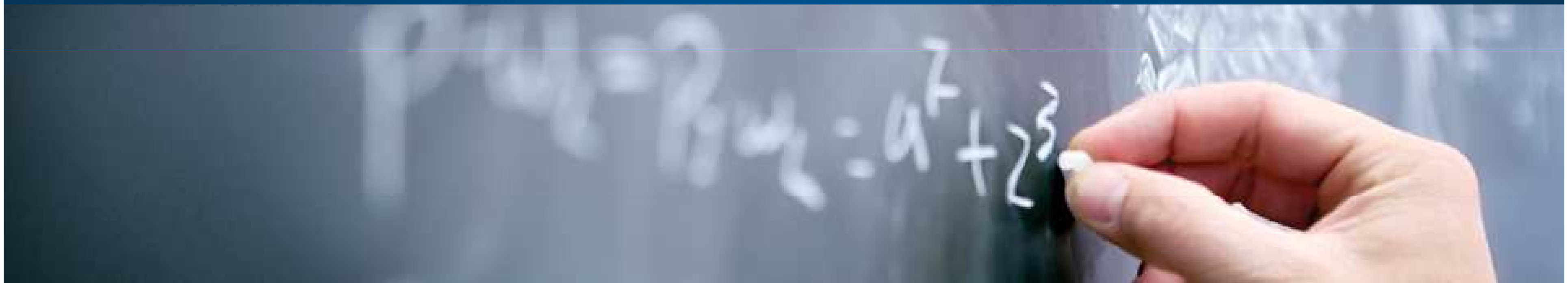
Content Based Image Retrieval

Natalia Vassilieva
nvassilieva@hp.com
HP Labs Russia

Tutorial outline

- Lecture 1
  - Introduction
  - Applications
- Lecture 2
  - Performance measurement
  - Visual perception
  - Color features
- Lecture 3
  - Texture features
  - Shape features
  - Fusion methods
- Lecture 4
  - Segmentation
  - Local descriptors
- Lecture 5
  - Multidimensional indexing
  - Survey of existing systems

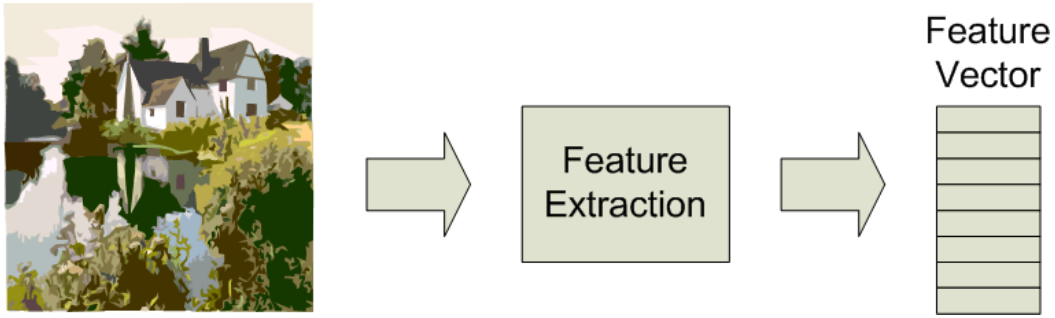
Lecture 5
Multidimensional indexing
Survey of existing systems



Lecture 5: Outline

- Multidimensional indexing
  - Tree structures
    - VP-tree
  - Locality Sensitive hashing
- Survey of existing systems

Need of multidimensional indexing



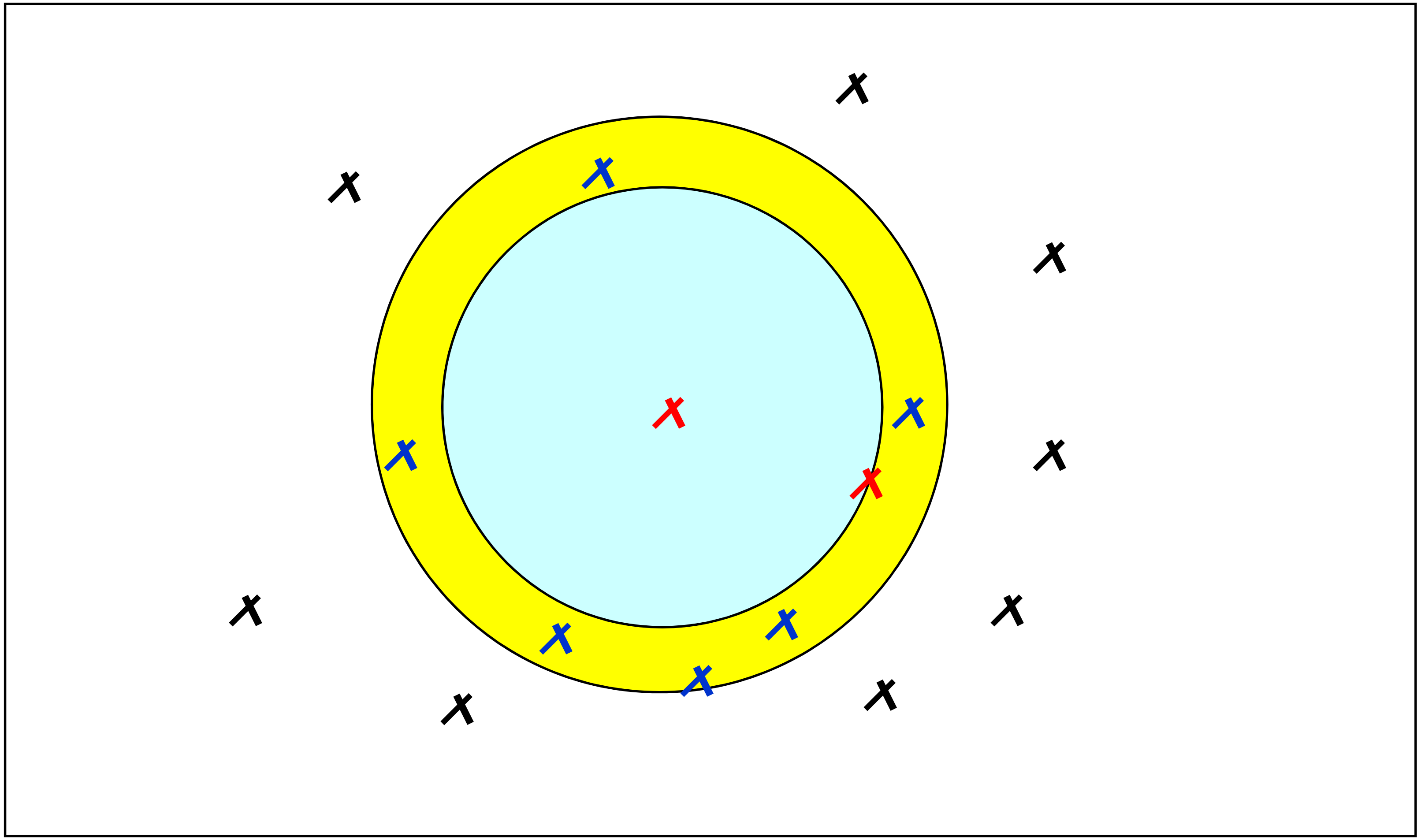
- High-dimensional data
  - Mean Color = RGB = 3 dimensional vector
  - Color Histogram = 256 dimensions
  - ICA-based texture = 21*30 dimensions
- Effective storage and speedy retrieval needed
- Similarity search, Nearest neighbour

Problem Description

- ε - Nearest Neighbor Search (ε - NNS)
  - Given a set P of points in a normed space , preprocess P so as to efficiently return a point $p \in P$ for any given query point q, such that
    - $dist(q,p) \leq (1 + \varepsilon) \times \min_{r \in P} dist(q,r)$

- Generalizes to K- nearest neighbor search (K >1)

Problem Description



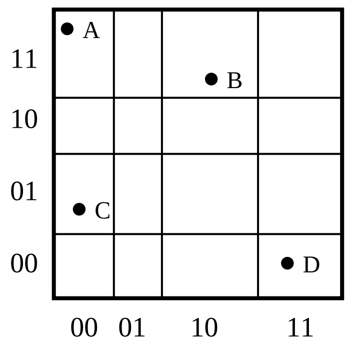
Lecture 5: Outline

- Multidimensional indexing
  - Tree structures
    - VP-tree
  - Locality Sensitive hashing
- Survey of existing systems

Some known indexing techniques

- Trees
  - R-tree – low dimensions (2D), overlap
  - Quad-tree – low dimensions (2D), inefficient for skewed data
  - k-D tree - inefficient for high dimensional skewed data
  - VP tree (metric trees)
- VA-file – not good for skewed data
- Hashing

data space



(a)

approximation

A	0011
B	1011
C	0001
D	1100

(b)

Spheres vs. Rectangles

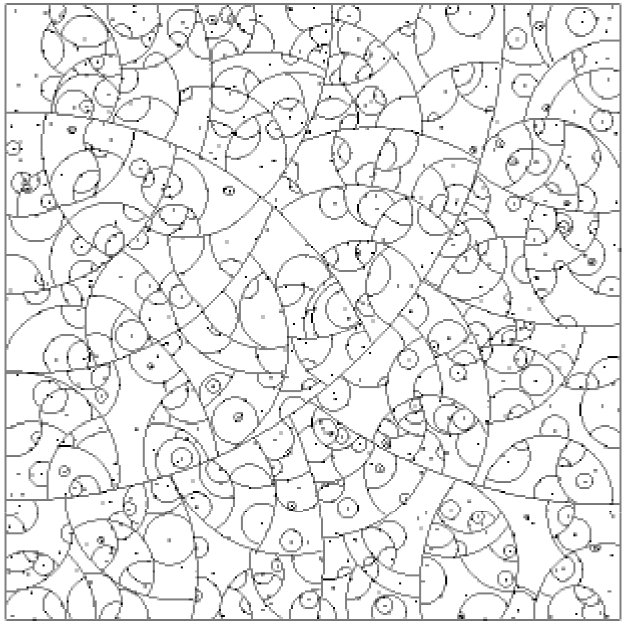


Figure 1: vp-tree decomposition

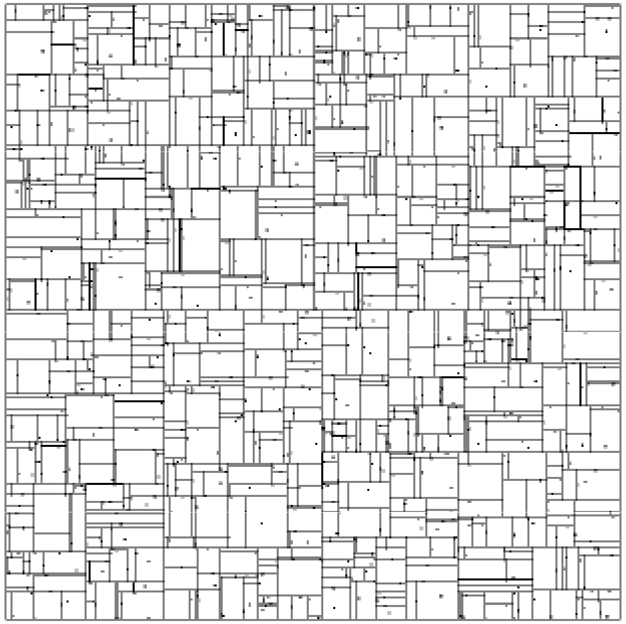


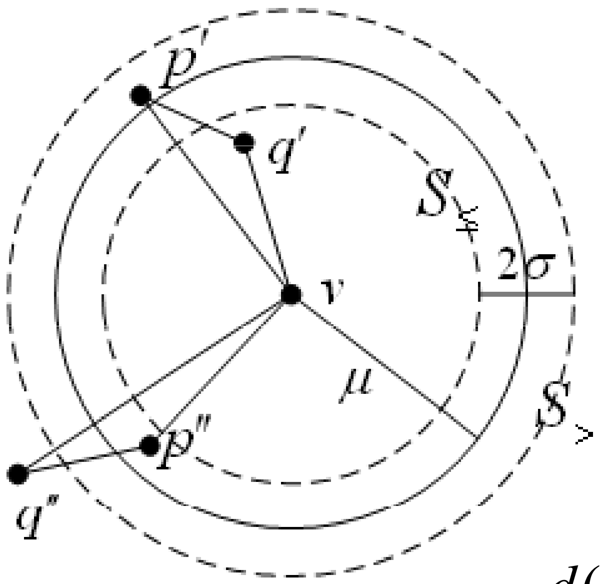
Figure 2: kd-tree decomposition

- $$\text{ratio} = \frac{\text{Volume(Sphere)}}{\text{Volume(Cube)}} \leq 1$$
- $$\text{dimensionality} \uparrow \Rightarrow \text{ratio} \uparrow$$
- relative distances

Lecture 5: Outline

- Multidimensional indexing
  - Tree structures
    - VP-tree
  - Locality Sensitive hashing
- Survey of existing systems

Vantage point method



$$d(v,q) \le \mu - \sigma \qquad p \in S_{>}$$

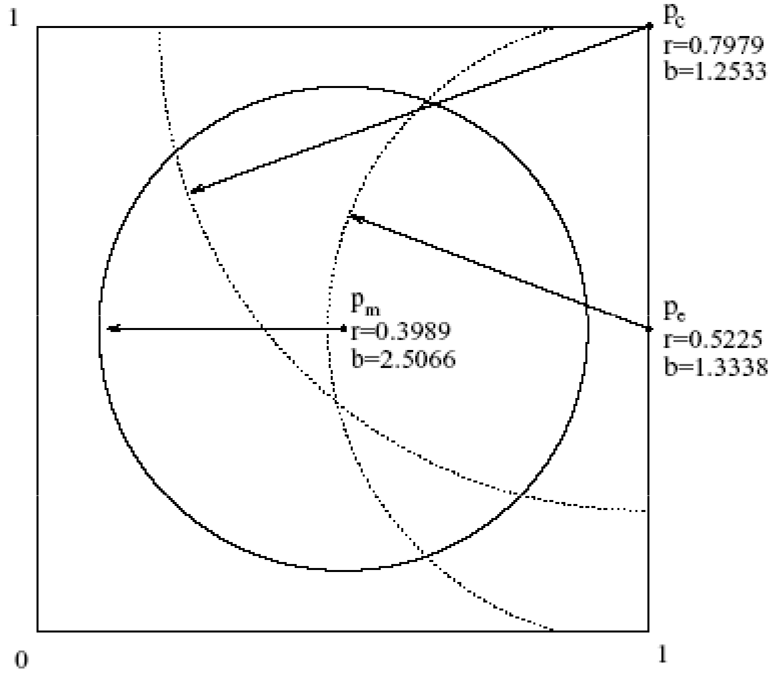
$$d(q,p) \ge | d(v,p) - d(v,q) | > | \mu - (\mu - \sigma) | = \sigma$$

$$d(v,q) > \mu + \sigma \qquad p \in S_{\le}$$

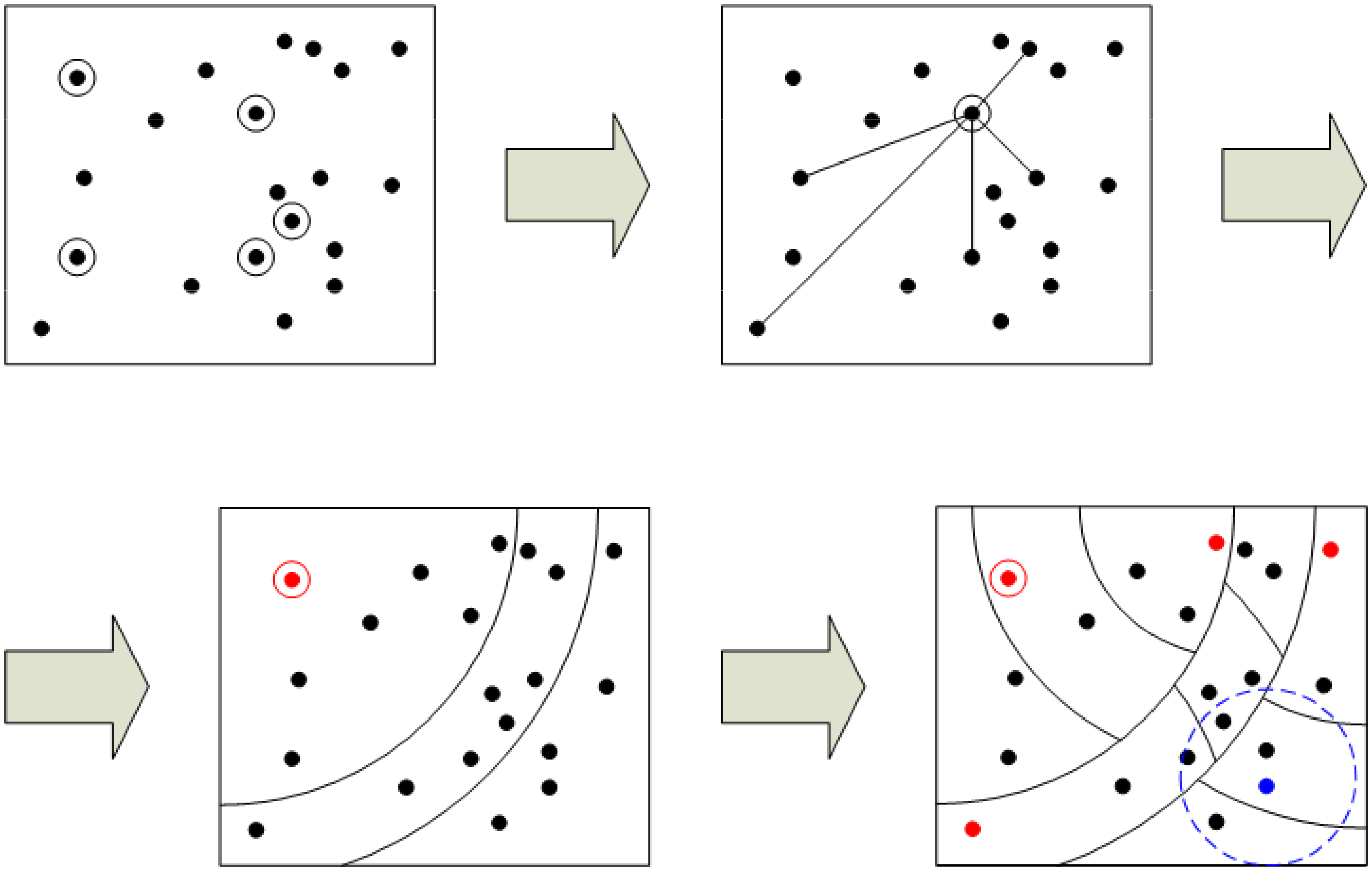
$$d(q,p) \ge | d(v,q) - d(v,p) | > | (\mu + \sigma) - \mu) | = \sigma$$

Conditions

- Minimum circuit
- "Corners" of the space
- Balanced tree
- Maximum standard deviation



Algorithms



Lecture 5: Outline

- Multidimensional indexing
  - Tree structures
    - VP-tree
  - Locality Sensitive hashing
- Survey of existing systems

LSH: Motivation

- Similarity Search over High-Dimensional Data
  - Image databases, document collections etc
- Curse of Dimensionality
  - All space partitioning techniques degrade to linear search for high dimensions
- Exact vs. Approximate Answer
  - Approximate might be good-enough and much-faster
  - Time-quality trade-off

LSH: Key idea

- Locality Sensitive Hashing (LSH) to get *sub-linear* dependence on the data-size for high-dimensional data
- Preprocessing :
  - Hash the data-point using several LSH functions so that probability of collision is higher for closer objects

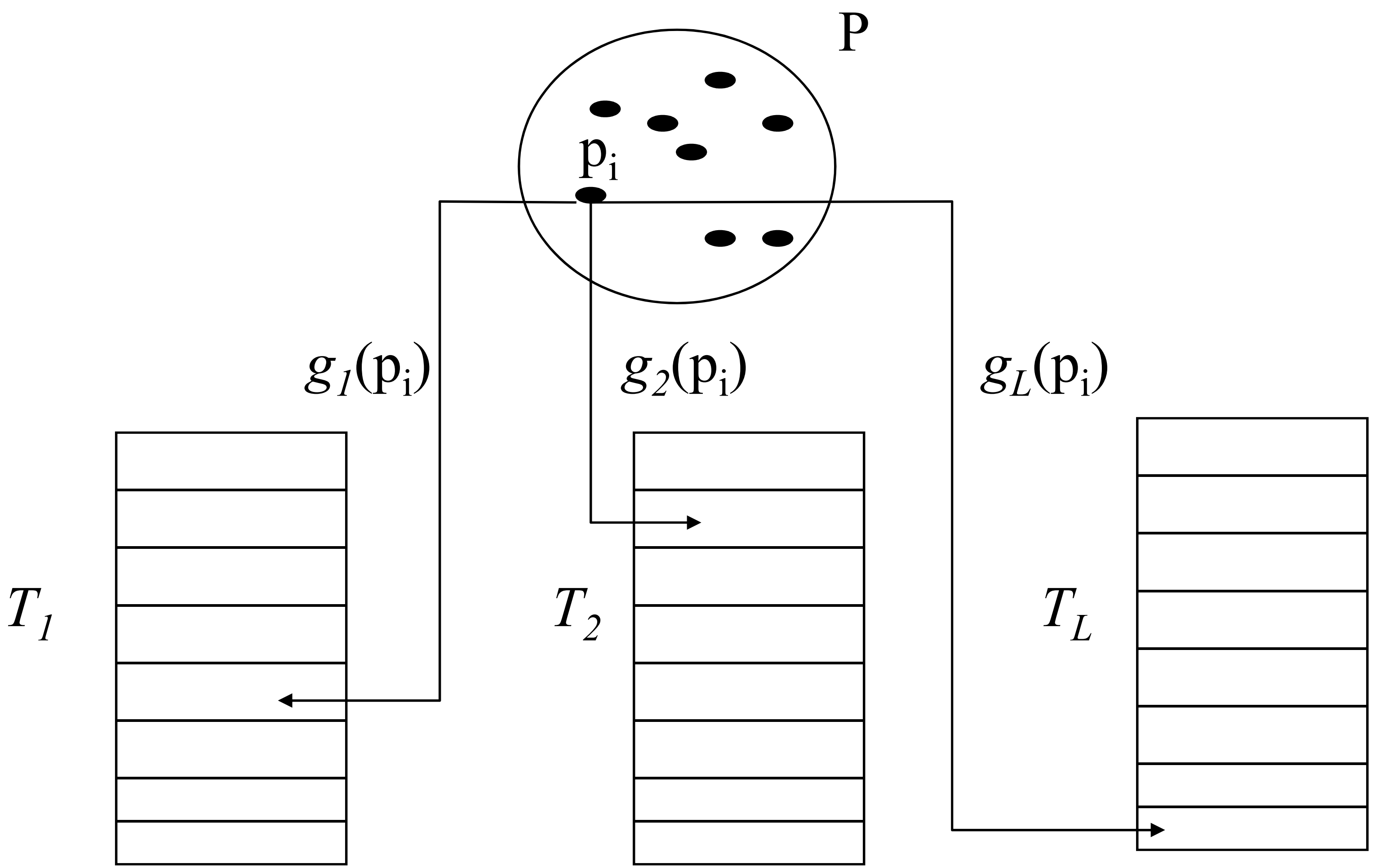
LSH: Algorithm

- Input
  - Set of N points $\{ p_1 , \ldots\ldots\ldots p_n \}$
  - L (number of hash tables)
- Output
  - Hash tables T_i , $i = 1 , 2, \ldots. L$
- Foreach $i = 1 , 2, \ldots. L$
  - Initialize T_i with a random hash function $g_i(.)$
- Foreach $i = 1 , 2, \ldots. L$

  Foreach $j = 1 , 2, \ldots. N$

  Store point p_j on bucket $g_i(p_j)$ of hash table T_i

LSH: Algorithm



LSH: ε - NNS Query

- Input
  - Query point q
  - K (number of approx. nearest neighbors)
- Access
  - Hash tables T_i , $i = 1 , 2, \ldots. L$
- Output
  - Set S of K (or less) approx. nearest neighbors
- $S \leftarrow \varnothing$

  Foreach $i = 1 , 2, \ldots. L$
  - $S \leftarrow S \cup$ { points found in $g_i(q)$ bucket of hash table T_i }

LSH: Analysis

- Family *H* of (r_1, r_2, p_1, p_2)-sensitive functions, $\{h_i(.)\}$
  - $dist(p,q) < r_1 \Rightarrow Prob_H [h(q) = h(p)] \geq p_1$
  - $dist(p,q) \geq r_2 \Rightarrow Prob_H [h(q) = h(p)] \leq p_2$
  - $p_1 > p_2$ and $r_1 < r_2$
- LSH functions: $g_i(.) = \{ h_1(.) \ldots h_k(.) \}$
- For a proper choice of *k* and *l*, a simpler problem, (r,ε)-Neighbor, and hence the actual problem can be solved
- Query Time : $O(d \times n^{[1/(1+\varepsilon)]})$
  - d : dimensions , n : data size

LSH: Applications

- To index local descriptors
  - Near duplicate detection
  - Sub image retrieval

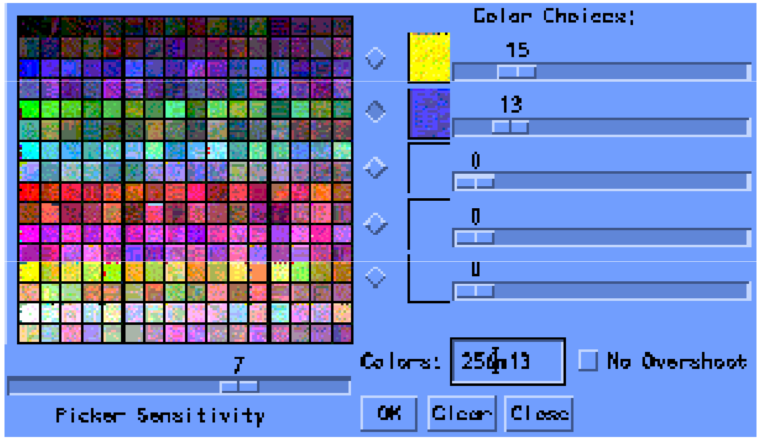
Lecture 5: Outline

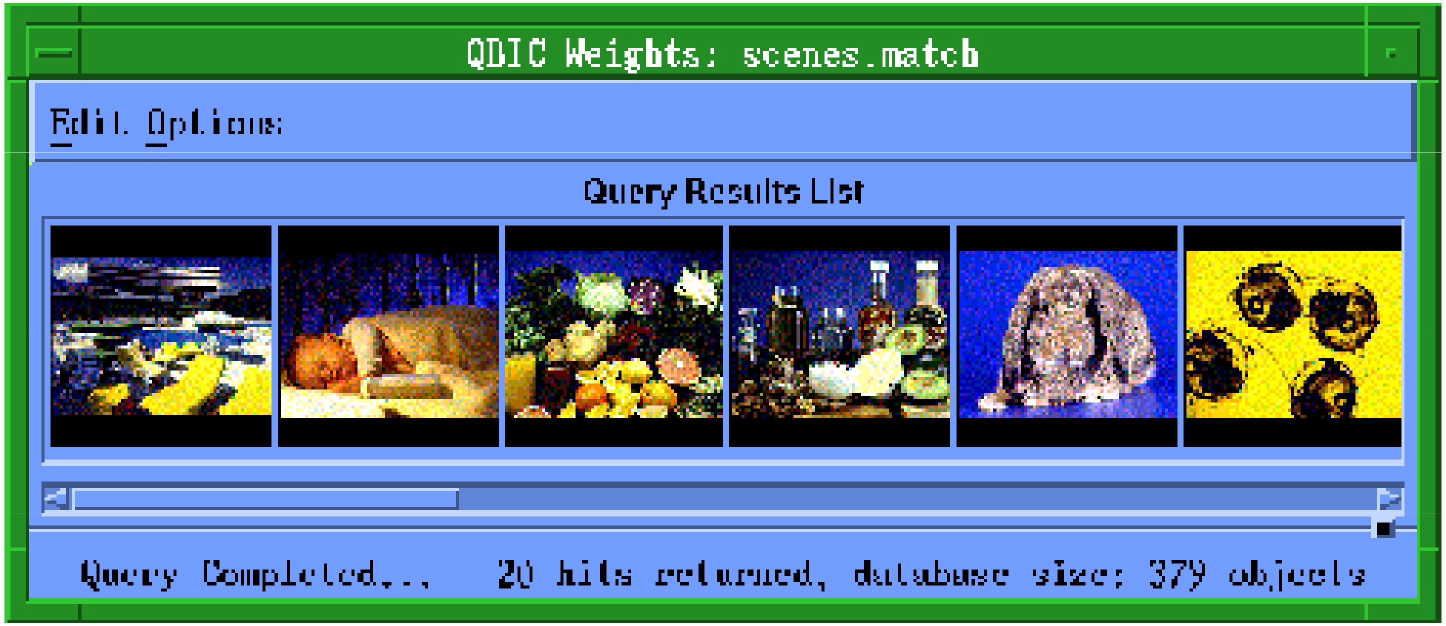
- Multidimensional indexing
  - Tree structures
    - VP-tree
  - Locality Sensitive hashing
- Survey of existing systems

IBM's QBIC

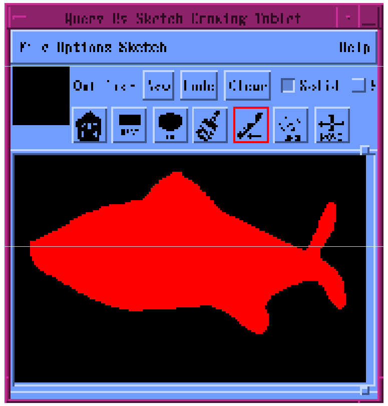
- http://wwwqbic.almaden.ibm.com/
- QBIC – Query by Image Content
- First commercial CBIR system.
- Model system – influenced many others.
- Uses color, texture, shape features
- Text-based search can also be combined.
- Uses R*-trees for indexing

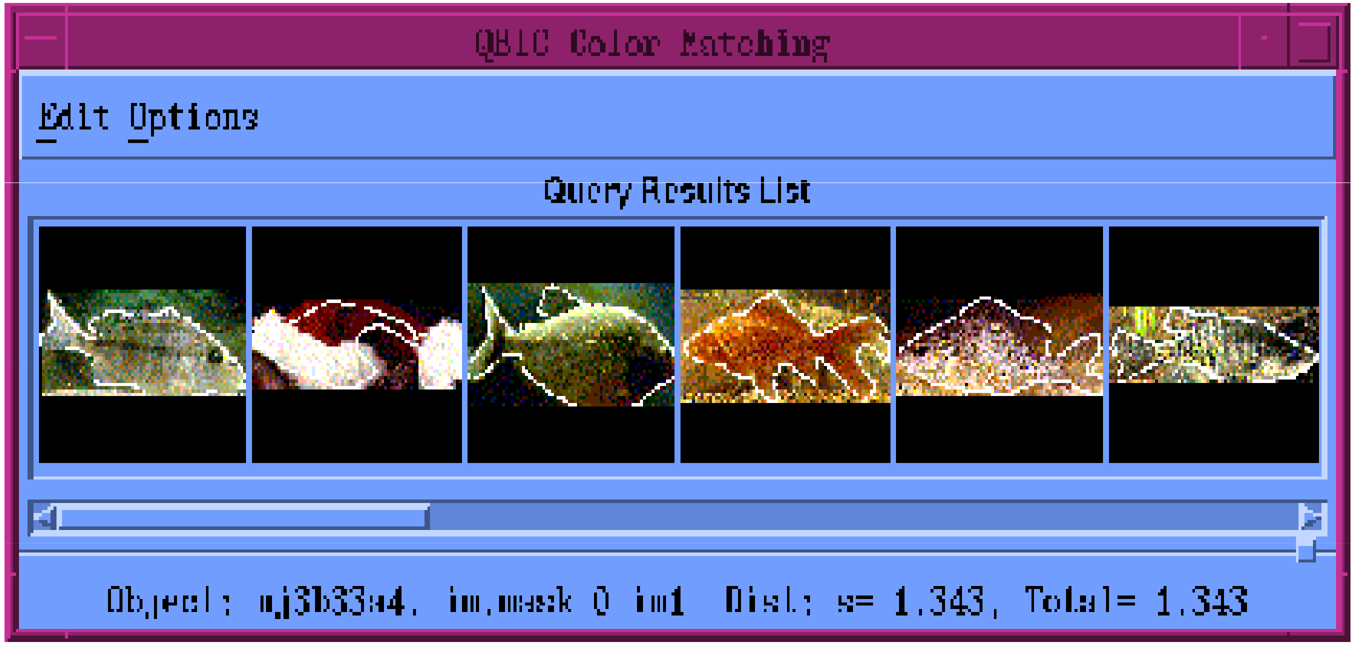
QBIC – Search by color





QBIC – Search by shape





QBIC – Query by sketch

Virage

- http://www.virage.com/home/index.en.html
- Developed by Virage inc.
- Like QBIC, supports queries based on color, layout, texture
- Supports arbitrary combinations of these features with weights attached to each
- This gives users more control over the search process

VisualSEEk

- http://www.ee.columbia.edu/ln/dvmm/researchProjects/MultimediaIndexing/VisualSEEk/VisualSEEk.htm
- Research prototype – University of Columbia
- Mainly different because it considers spatial relationships between objects.
- Global features like mean color, color histogram can give many false positives
- Matching spatial relationships between objects and visual features together result in a powerful search.

Features in some existing systems

	Color	Texture	Shape
QBIC	Histograms (HSV) $dist^2 = H_1 A H_2^T$	Tamura Image, Euclid dist	Boundary geometrical moments + Invariant moments
VisualSEEk	Histograms (HSV), Color Sets, Location info		
Netra	Histograms (HSV), Color codebook, Clusterisation	Gabor filters	Fourier-based
Mars	Histograms, HSV $dist = 1 - \sum_{i=1}^{N} \min(H_1(i), H_2(i))$	Tamura Image, 3D Histo	MFD (Fourier)

Other systems

- xCavator by CogniSign
  http://xcavator.net/
- CIRES
  http://amazon.ece.utexas.edu/~qasim/samples/sample_buildings5.html
- MFIRS by University of Mysore
  http://www.pilevar.com/mfirs/
- PIRIA
  http://www-list.cea.fr/fr/programmes/systemes_interactifs/labo_lic2m/piria/w3/pirianet.php?bdi=coil-100&cide=cciv&up=1&p=1

Other systems

- IMEDIA
  http://www-rocq.inria.fr/cgi-bin/imedia/circario.cgi/v2std
- TILTOMO
  http://www.tiltomo.com/
- The GNU Image-Finding Tool
  http://www.gnu.org/software/gift/
- Behold
  http://www.beholdsearch.com/about/#features
- LTU technologies
  http://www.ltutech.com/en/
- …

LABS hp

Lecture 5: Resume

- Multidimensional indexing
  - VP trees can be used
  - LSH is great for near duplicates and sub image retrieval
- There are a lot of systems
  - Research projects
  - Commercial projects (usually combined with text-based retrieval)
  - CBIR is a very active area: research is moving to commercialize projects just now

Lecture 5: Bibliography

- Christian Böhm, Stefan Berchtold, Daniel A. Keim. Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. ACM Computing Surveys 2001.
- Volker Gaede, Oliver Günther. Multidimensional Access Methods. ACM Computing Surveys 1998.
- Roger Weber, Hans-Jörg Schek, Stephen Blott. A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces. International Conference on Very Large Data Bases (VLDB) 1998.
- Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In SCG '04, pp 253-262, 2004.
- Kave Eshgi, Shyamsundar Rajaram. Locality Sensitive Hash Functions Based on Concomitant Rank Order Statistics. In Proc. of ACM KDD, 2008.

Tutorial outline

- Lecture 1
  - Introduction
  - Applications
- Lecture 2
  - Performance measurement
  - Visual perception
  - Color features
- Lecture 3
  - Texture features
  - Shape features
  - Fusion methods
- Lecture 4
  - Segmentation
  - Local descriptors
- Lecture 5
  - Multidimensional indexing
  - Survey of existing systems