
Efficient Video Similarity Measurement & Search

Presented at University of Kentucky on February 27, 2004

Sen-ching Samson Cheung, sccheung@ieee.org
University of California, Berkeley
(Now at Lawrence Livermore National Laboratory)

Outline

- Problem, motivation, and overview
- System components:
 - video signature
 - fast similarity search
 - signature clustering
- Search engine demo
- Summary and Future work

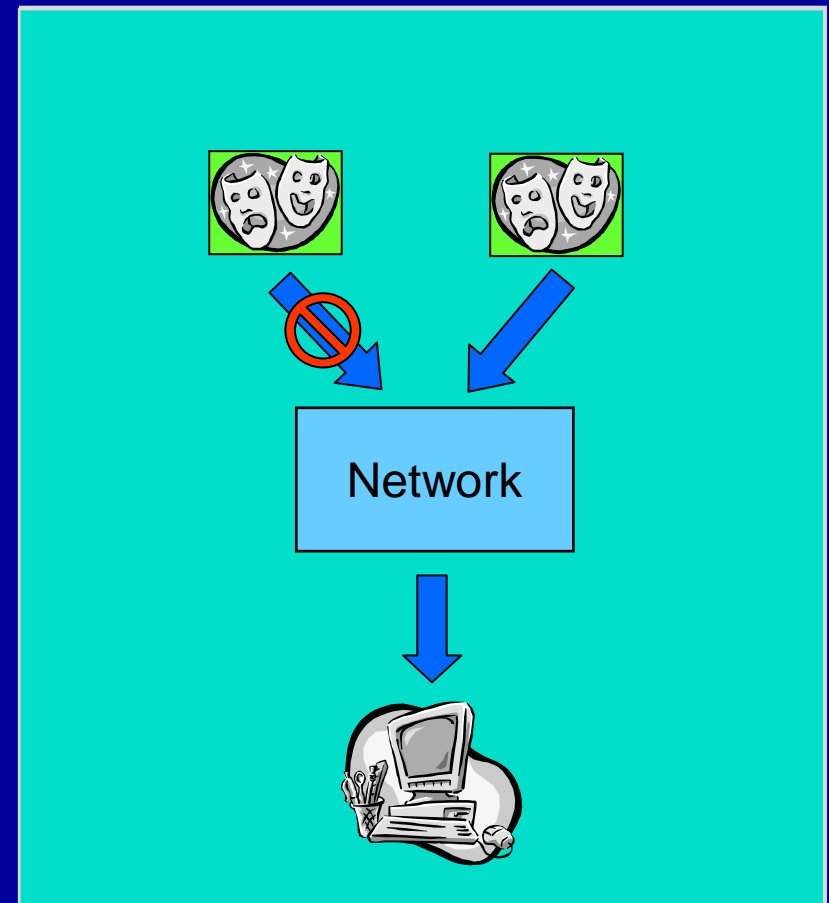
Similar video clips on the web

Same content but undergone format conversion, minor editing, or summarization, etc.



Why do we care?

- Content identification without modification
- Better organization of search results
- Fault-tolerant delivery



Problem

Design efficient algorithms to measure video similarity and to perform video similarity search in large databases

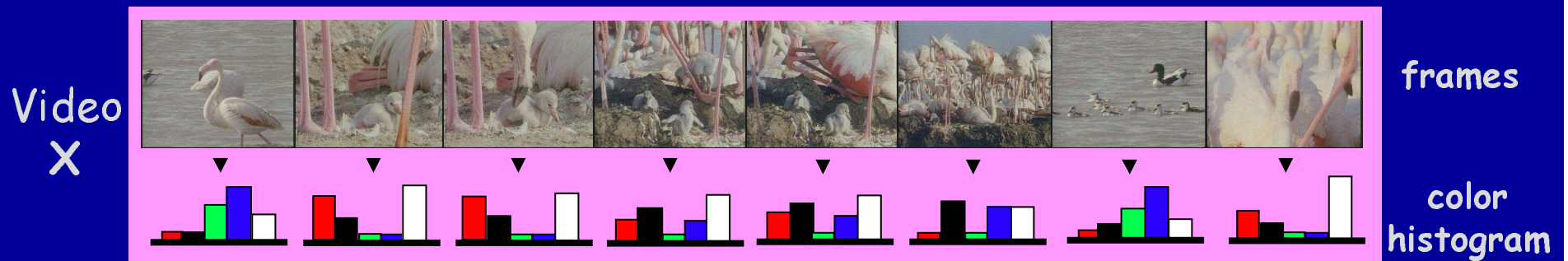
Overview of Solutions

- Efficient representation for similarity measurement
 - Focus on complexity reduction on similarity measurement
 - Summarize video into video signature
- Fast similarity search on signatures
 - Novel feature extraction scheme for metric spaces
- Signature Clustering
 - Improved retrieval performance
 - Intuitive organization

Outline

- Problem, motivation and overview
- System components:
 - video signature
 - fast similarity search
 - signature clustering
- Search engine demo
- Summary and Future work

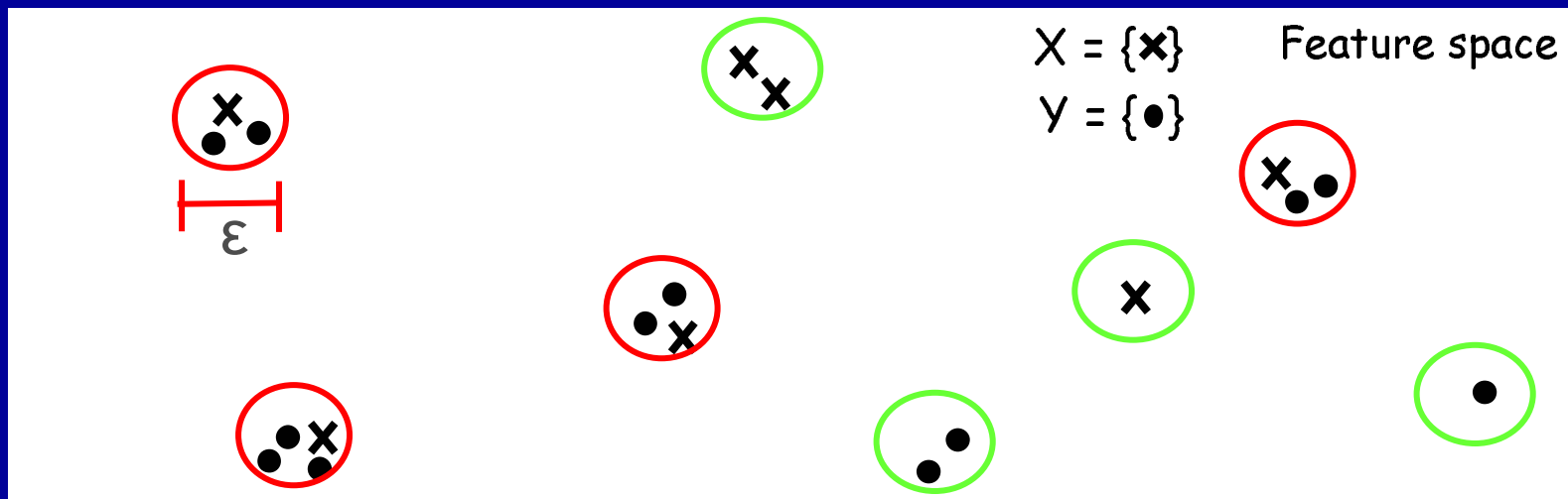
Bag-of-Frame Video Model



- Represent each frame as a feature vector with a metric $d(x,y)$ so that,
frame x and y are similar if $d(x,y) \leq \epsilon$
- Bag-of-vector video model: $X = \{x_i, i = 1, \dots, T\}$
 - robust against any reordering

Ideal video similarity

Define Ideal Video Similarity, $IVS = \frac{\# \text{red circle}}{\# \text{red circle} + \# \text{green circle}}$

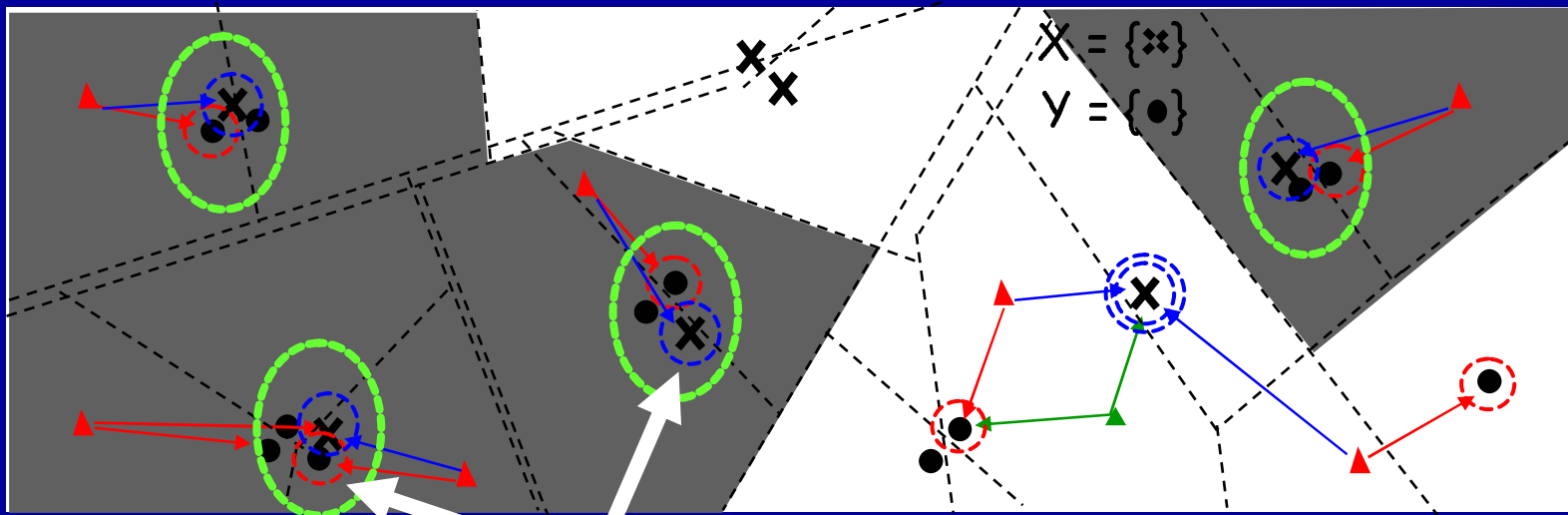


- *Need to store the whole video*
- *Complexity $\sim O(|X| \cdot |Y|)$*

Voronoi video similarity

- Voronoi diagram - partition of F based on proximity to vectors
- Intersection = $\{s: \text{closest frames to } s \text{ from } X \text{ and } Y \text{ are similar}\}$
- Voronoi video similarity = Relative volume of intersection
– estimated by # of "seed vectors" in intersection

$$VVS \cong \frac{\# \text{○}}{\# \text{▲}}$$



Use these vectors to represent video \Rightarrow Video signature

Video Signature

1. Select a set of random seed vector:

$$S = \{s_1, s_2, \dots, s_m\}$$

2. For each video X in database, compute its video signature:

$$\text{ViSig}(X) = (x_1, x_2, \dots, x_m)$$

where $x_i = \arg \min_{x \in X} d(x, s_i)$

3. Video clips X and Y are similar $\Leftrightarrow d(x_i, y_i) \leq \varepsilon$ for more than $m/2$ i 's

Why video signature?

1. Easy to generate - single pass on the video
2. Fixed size - each signature has m vectors
3. Scalable - To achieve an error probability less than δ in a database of N videos,

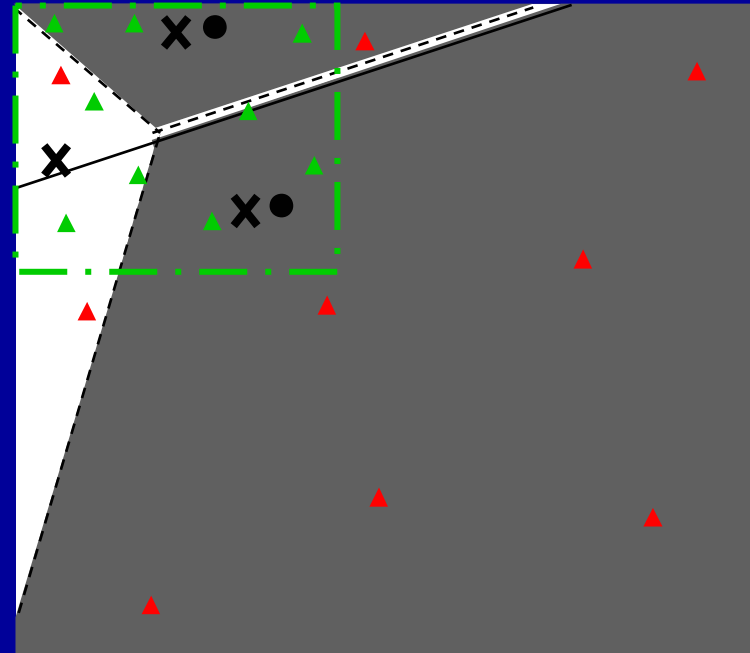
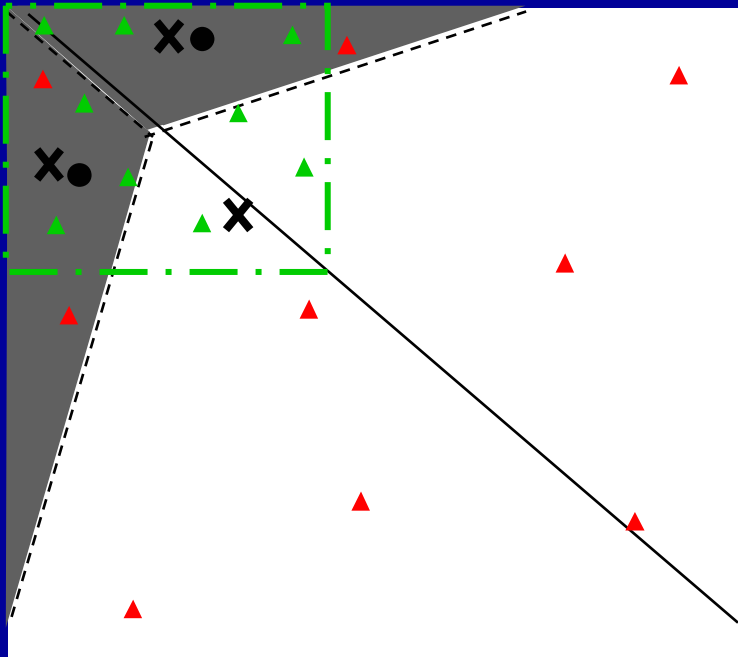
$$|\text{ViSig}(X)| = O(\ln N - \ln \delta)$$

- accuracy does not depend on the length of the video!
- scalable to large database



1. VVS \neq IVS in general
2. Sensitive to ϵ - "Voronoi Gap Problem"

Seed vector distribution



Theorem: If ϵ is small, and all seed vectors have equal prob. to be in every Voronoi Cell of $X \cup Y$,

$$IVS = 2/3$$

$$IVS = 2/3$$

$$VSS = 3/9$$

$$VSS = 7/9$$

$$E[VSS_b(X, Y; \epsilon)] = IVS(X, Y; \epsilon)$$

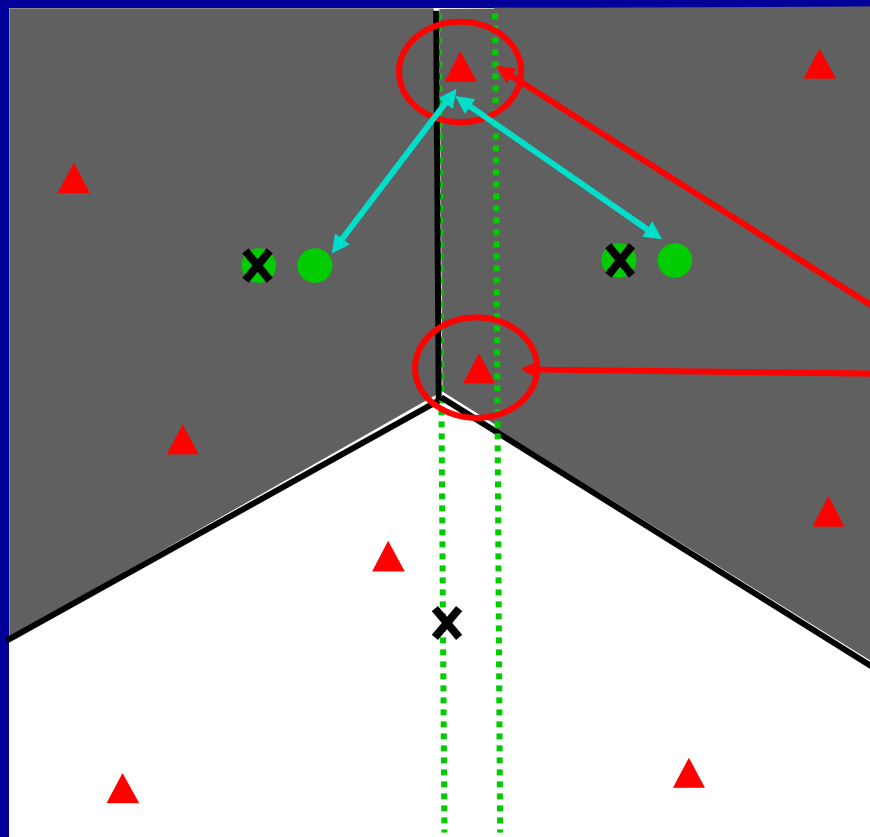
$$VSS = 6/9$$

$$VSS = 6/9$$



Randomly sample seed vectors from clusters of training data.

Voronoi gap problem



→
Perturbation

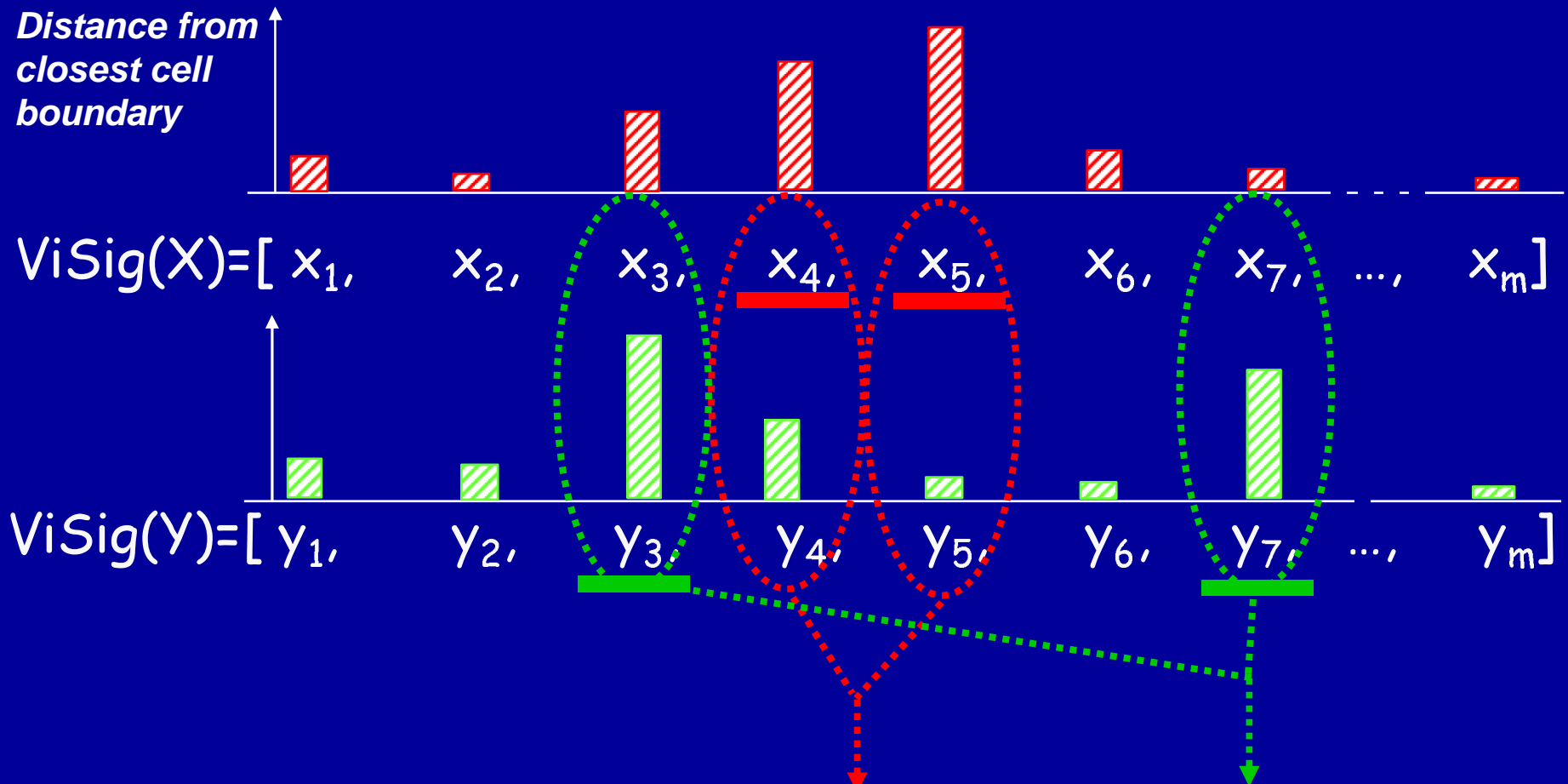
$$VSS = \frac{\cancel{6}}{\cancel{9}} \frac{4}{9}$$

Thm: s in VG \Rightarrow s
must be within 2ϵ
from cell boundary



*Over-sample and
use seed vectors
that are far away
from boundary!*

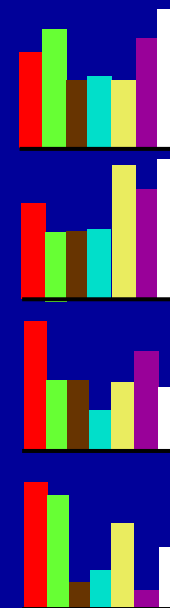
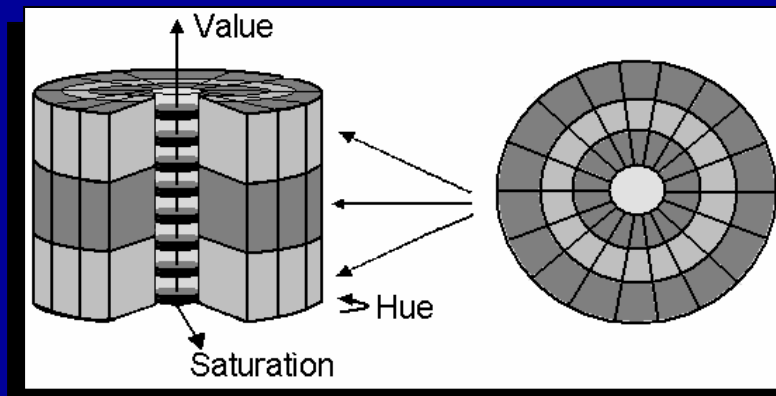
Ranked signature similarity



Only used top "ranked" signature vectors in similarity measurement!

Feature vector

HSV Color Histogram [Smith 97], MPEG7

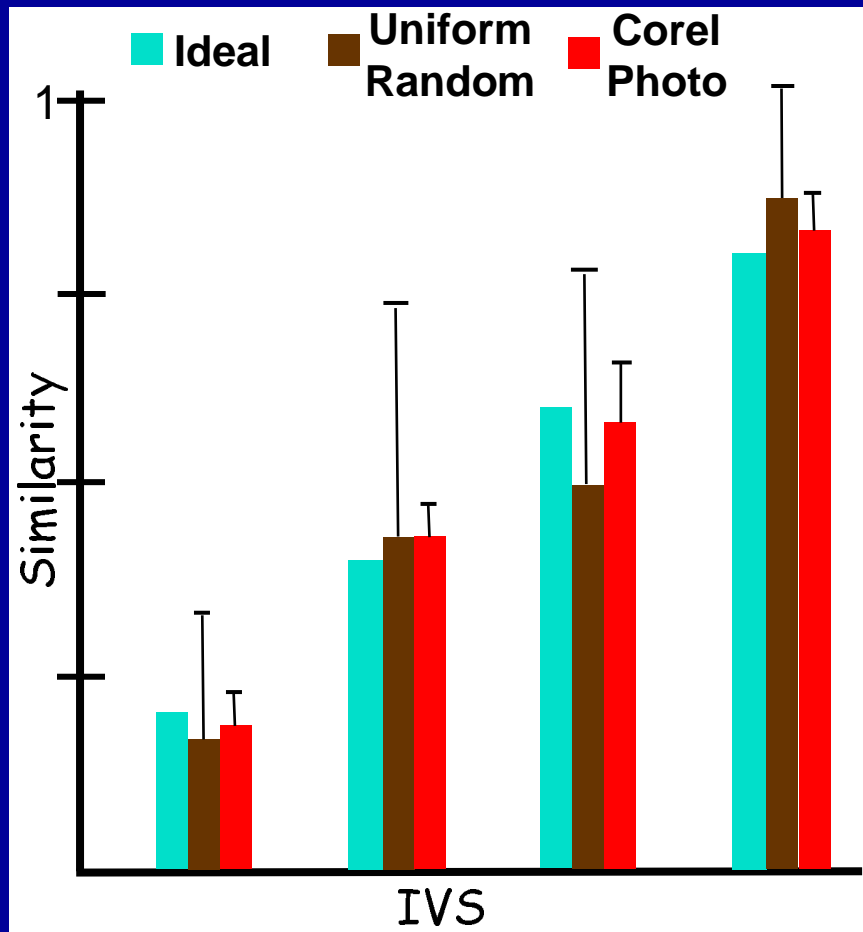


- Dimension = $(3 \times 18 \times 3 + 16)$ bins \times 4 quadrants = 712
- Remove common dominant color - slide show, plots
- l_1 -distance
- 100 seed vectors, top 2 - 14 ranked vectors for measurement

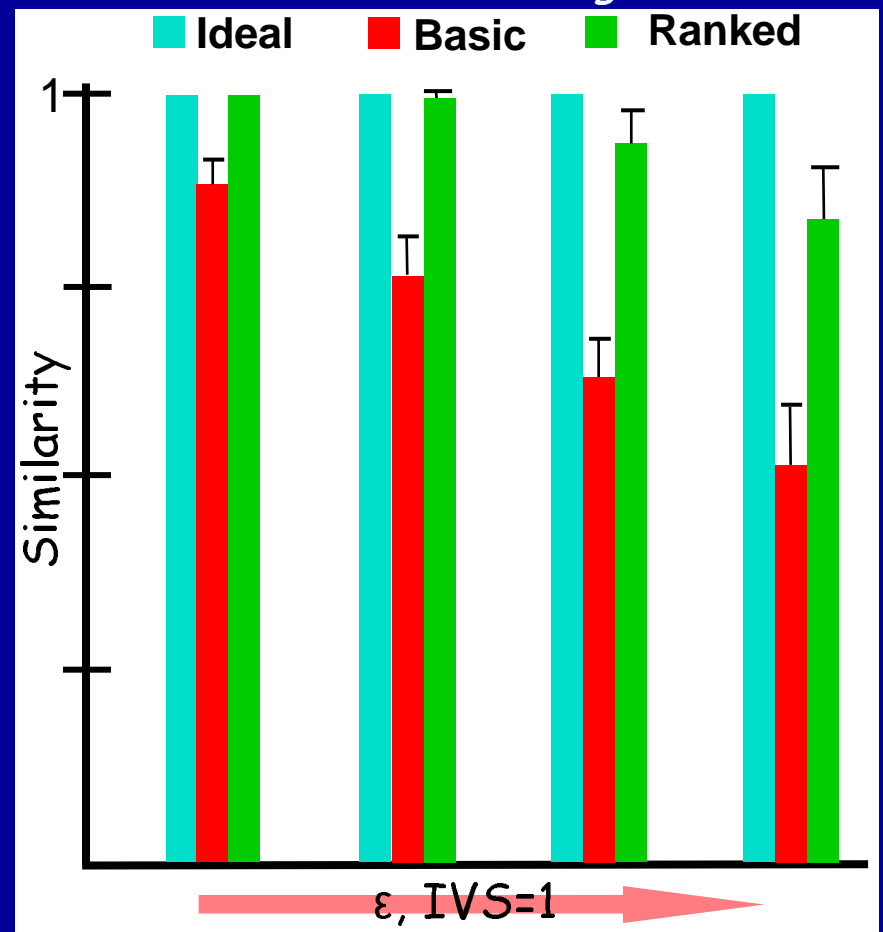
Simulations with artificial similarity

- Average results of 15 MPEG-7 test sequences

Seed Vector Distribution

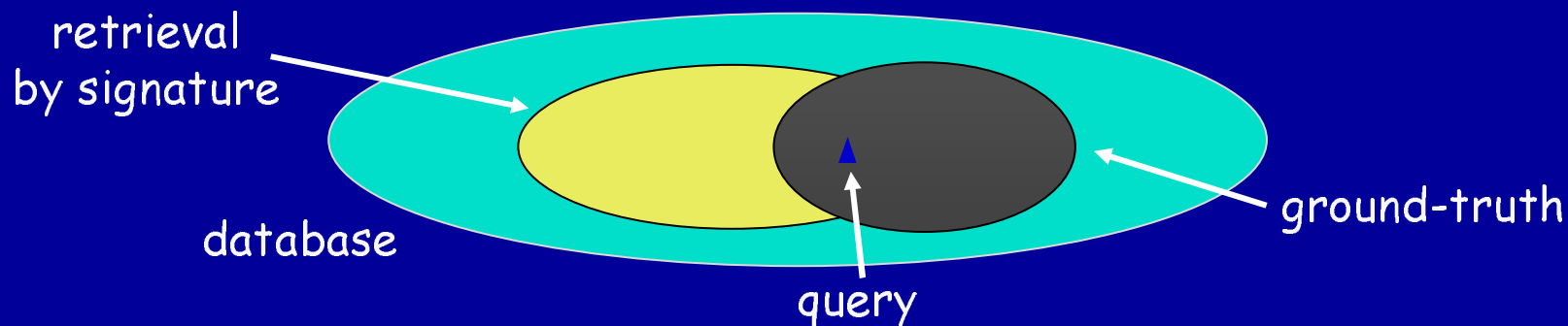


Basic vs. Ranked Signatures

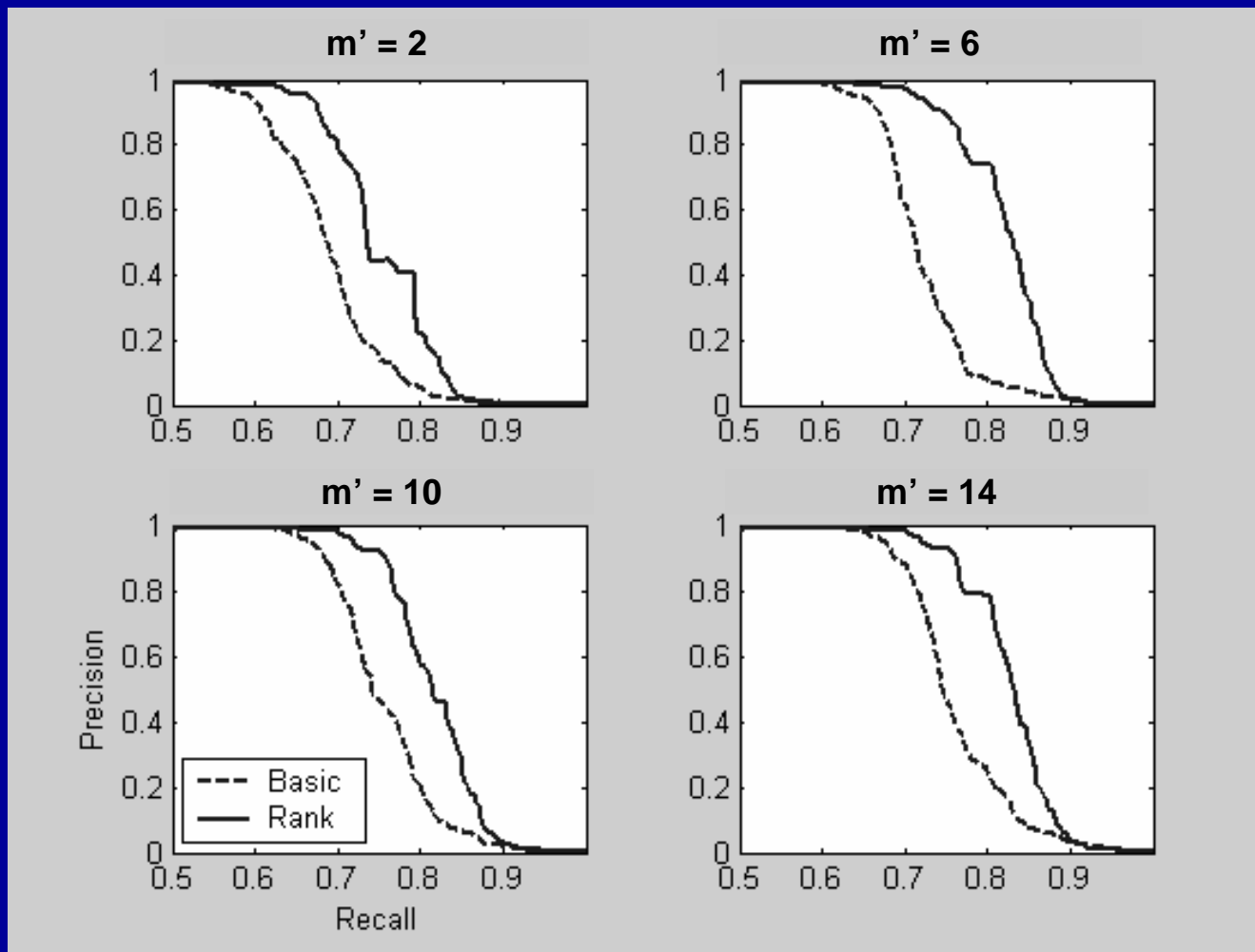


Real web experiment

- 46,356 (1800 hours) video clips crawled from the web
- Statistical pooling methods to generate ground-truth
 - meta-data + medoid + manual examinations
- Ground-truth G has 107 clusters of similar video totaling 443 clips
- Performance measurements at different ϵ :
 - Recall = % of ground-truth retrieved = $|\text{yellow} \cap \text{red}| / |\text{red}|$
 - Precision = % of retrieval that are ground-truth = $|\text{yellow} \cap \text{red}| / |\text{yellow}|$

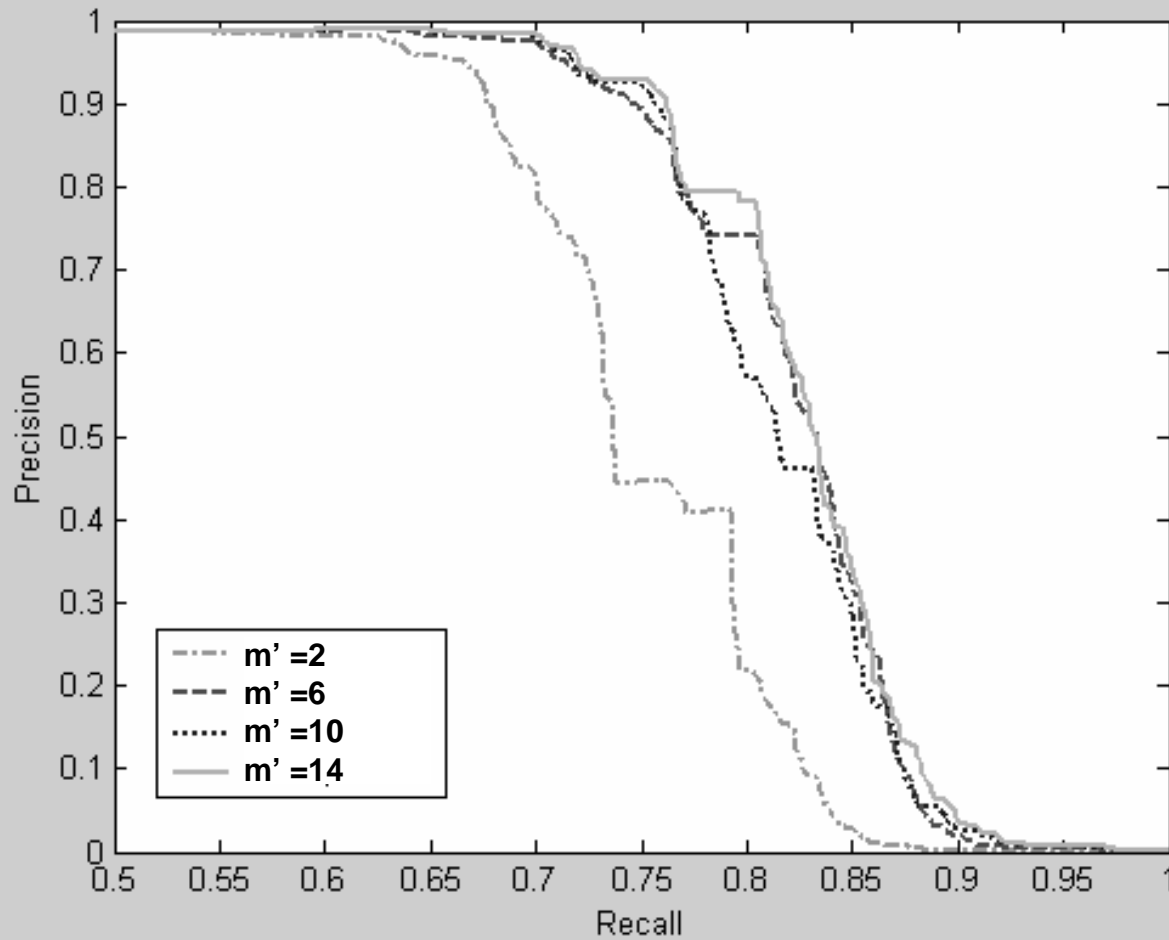


Experiment : Basic vs. Ranked VSS



Ranked signatures use $m=100$ vectors.

Experiment : Number of seeds

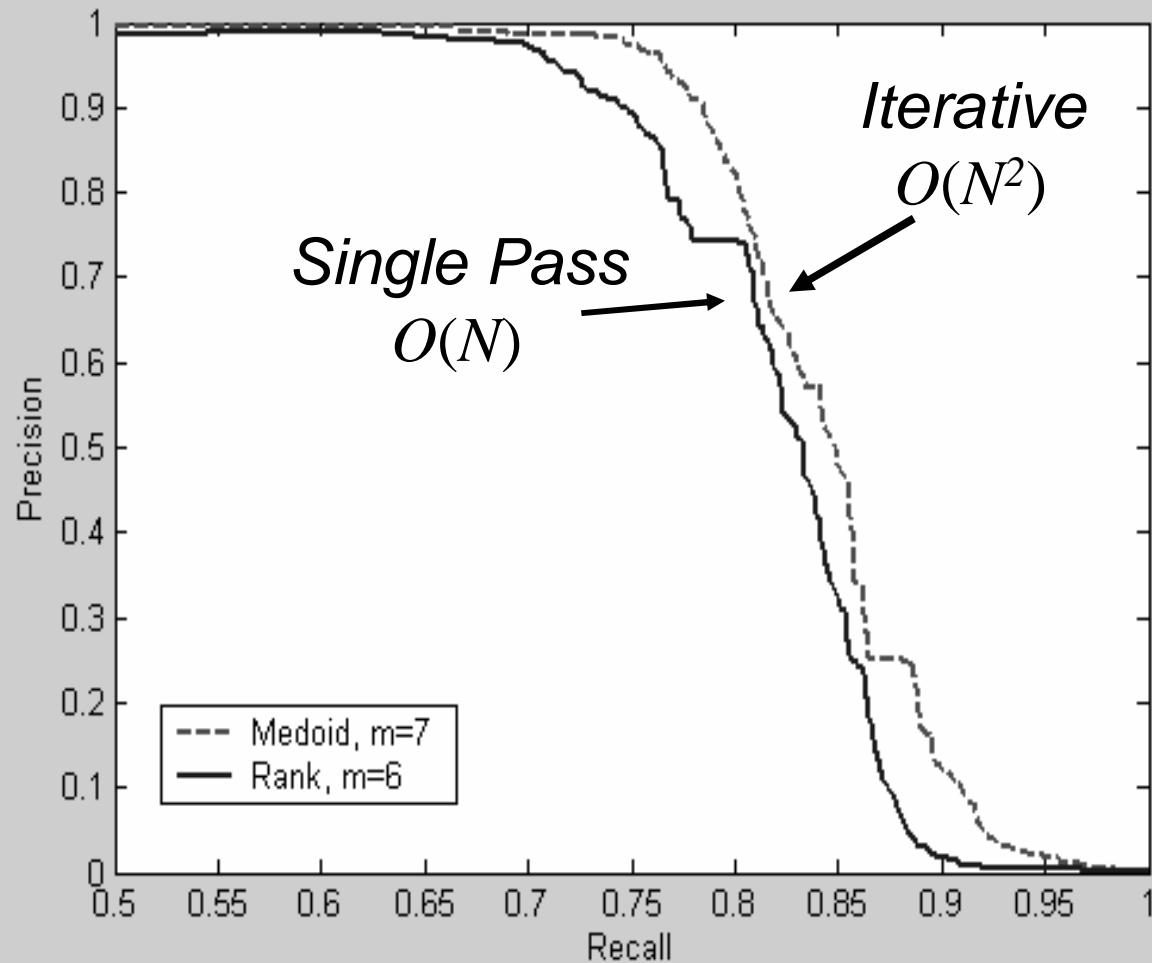


Each signature has 100 frames.

Find the number of top-ranked frames required for comparison

Experiment : Compare w/ Medoid

Medoid summarizes a video by finding m of its frames that are “closest” to the entire video [Chang et.al 98]



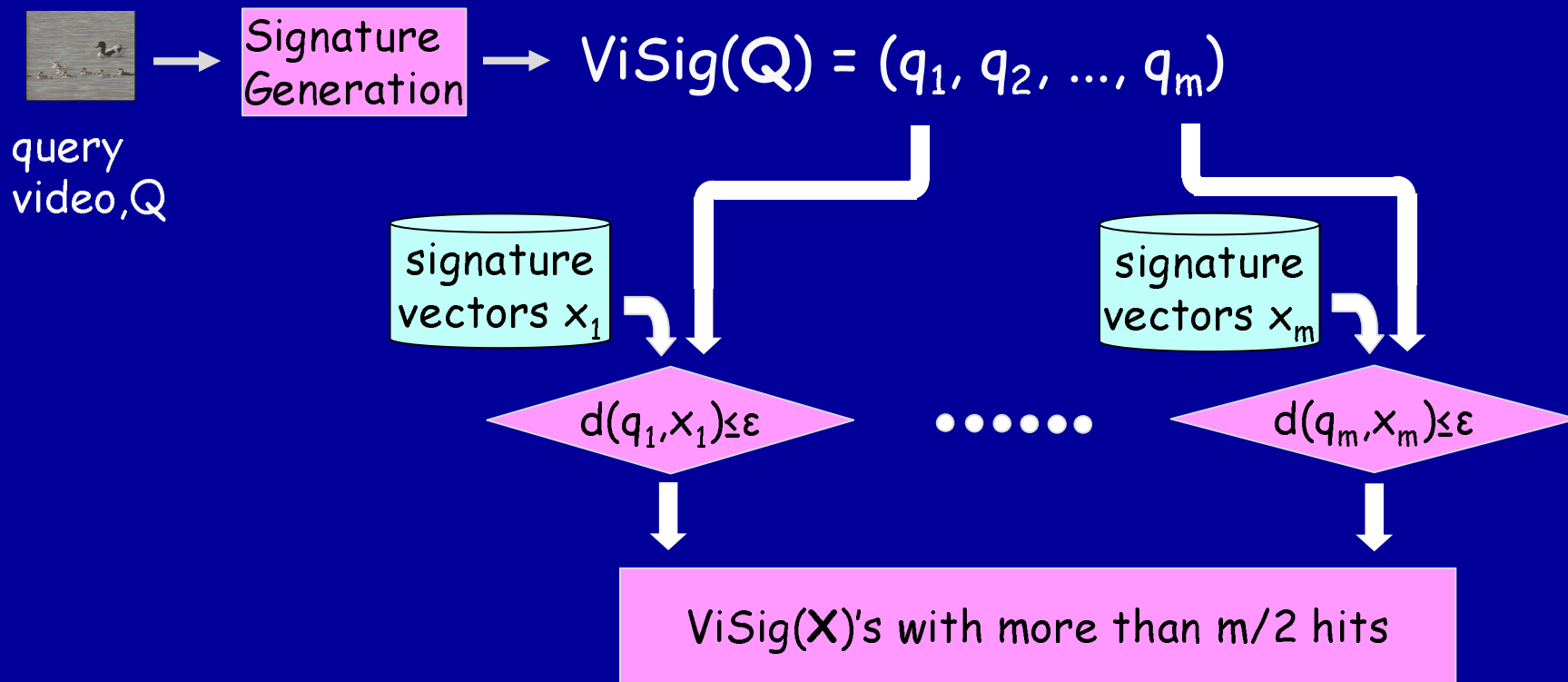
Video similarity literature

- Compact representations of shots not entire video
 - object-based, motion trajectory [Chang et.al 98], MPEG7
 - distribution of color or texture [Iyengar 99], MPEG7
 - piecewise polynomial [Naphade et al. 00]
 - low dimensional bounding box [Kobla et al. 97]
- Key-frames for summarization, not similarity measurement
 - browsing [Zhang 95, Yeng 95, Gonsel 97, Sun 98, Giensohn 00]
 - iterative frame-constrained clustering [Chang et al. 99]
- Need measurement insensitive to temporal edit
 - warping/edit distance [Lienhart 98, Adjero 99, Naphade et.al 01]
 - time series of shot duration [Indyk et.al 99]

Outline

- Problem, motivation and overview
- System components:
 - video signature
 - fast similarity search
 - signature clustering
- Search engine demo
- Summary and Future work

Similarity Search



Key Step :

For a given q , find all x with $d(x, q) \leq \epsilon$ in a large database

Curse of dimensionality

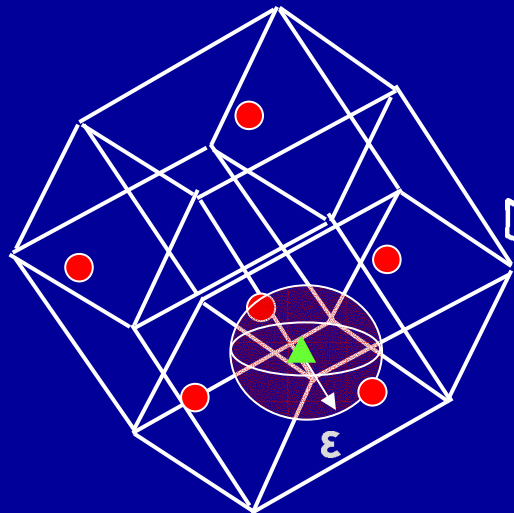
Similarity search on high-dimensional vectors is hard !

Typical indexing techniques reduce to sequential search for 10 or higher dimensions .

[Weber, Schek, Blott98]

- Average distance between closest neighbors increase with dimension
- # bounding regions grows exponentially with dimension
- Volume of each region shrinks exponentially with dimension

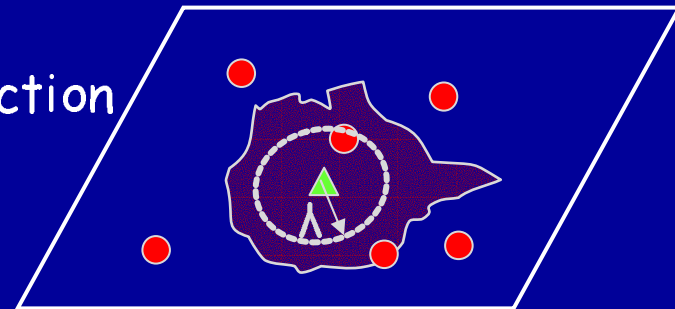
Dimension reduction for fast search



High dimensional space
 $d(x,y)$



Dimension Reduction
Mapping T

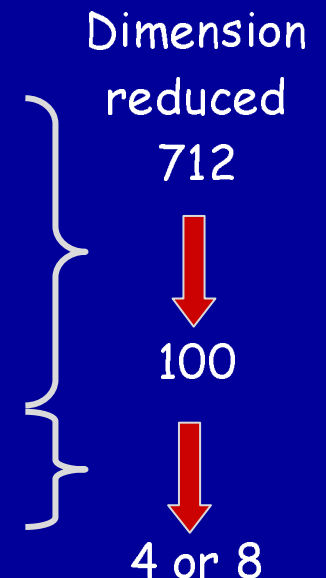


Low dimensional space
 $d'(T(x),T(y)) + \text{Indexing}$

Goal: Need to find mapping T and metric $d'(\cdot, \cdot)$ that can preserve distance relationship

Propose new dim. reduction mapping

- Applicable to general metric
- Better performance than PCA, Wavelet, Fastmap, etc.
- Proposed dimension reduction scheme
 1. Utilizes both upper and lower bounds of the Triangle Inequality
 2. Combines them into a single metric function called "Difference of Squares"
 3. Further reduces dimension by PCA



Use Triangle Inequality for dimension reduction

Basic Triangle inequality:

$$|d(x,s)-d(q,s)| \leq d(x,q) \leq [d(x,s)+d(q,s)]$$

Use multiple seeds to obtain tighter bounds:

$$\max_i |d(x,s_i)-d(q,s_i)| \leq d(x,q) \leq \min_i [d(x,s_i)+d(q,s_i)]$$

- $T(x) = [d(x,s_1), d(x,s_2), \dots, d(x,s_m)]$
- Added benefit: $d(x,s_i)$ for all s_i 's are already computed by Video Signature

Triangle Inequality Pruning (TIP)

Dimension Reduction:

For each signature vector x , compute $T(x)$ given by:

$$T: x \rightarrow [d(x, s_1), d(x, s_2), \dots, d(x, s_m)]$$

Similarity Search for q :

1. **Pruning:** Remove x in the database that satisfy

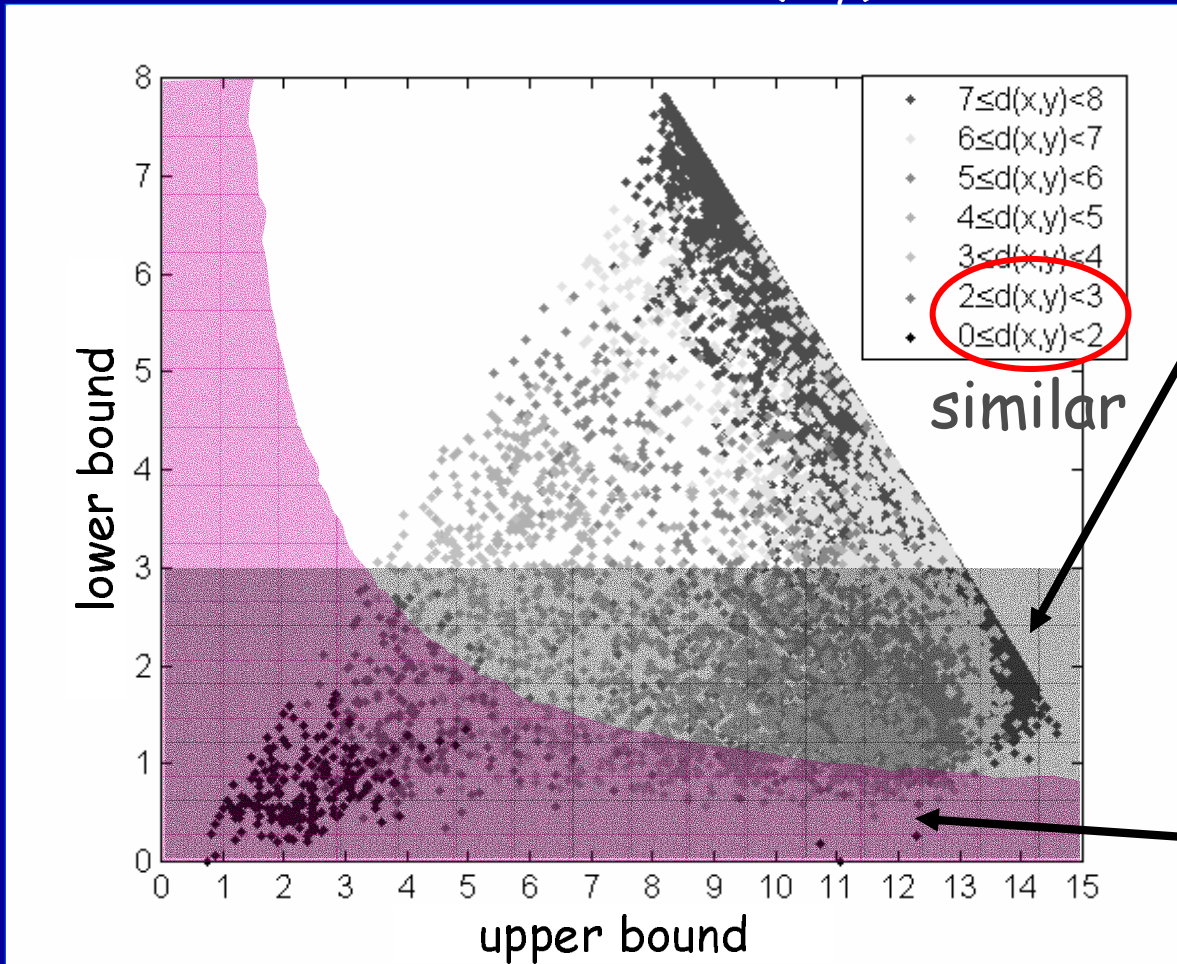
$$l_\infty(T(x), T(q)) = \max_i |d(x, s_i) - d(q, s_i)| > \lambda$$

2. **Refinement:** Identify x that satisfy $d(x, q) \leq \varepsilon$ in the remainder of the database.

Only the lower bound is used!

Use both bounds for pruning

10,000 random $d(x,y)$



Lower bound ≤ 3

Accuracy = 100%

Pruning Amt = 70%

Product of two bounds ≤ 9

Accuracy = 98%

Pruning Amt = 99%!

Heuristic justification

Simple algebra :

$$d(x,q) < \varepsilon \Rightarrow \text{upper-bound} < \varepsilon + 2 \cdot \min_i \{d(q,s_i), d(x,s_i)\}$$

- Not a tight bound \Rightarrow no performance guarantee
- Neither the upper bound nor the product of bounds forms a metric for indexing. Try "*Difference of square*":

$$\begin{aligned} & \sum_i [|d(x,s_i) - d(q,s_i)| \cdot (d(x,s_i) + d(q,s_i))]^2 \\ &= \sum_i [d(x,s_i)^2 - d(q,s_i)^2]^2 \end{aligned}$$



l_2 between $T^*(x)$ and $T^*(q)$ where
 $T^*(x) = [d(x,s_1)^2, \dots, d(x,s_m)^2]$

Proposed Technique (Version 1)

Dimension Reduction:

For each signature vector x , compute $T^*(x)$ given by

$$T : x \rightarrow (d(x, s_1)^2, d(x, s_2)^2, \dots, d(x, s_m)^2)$$

712-d histogram

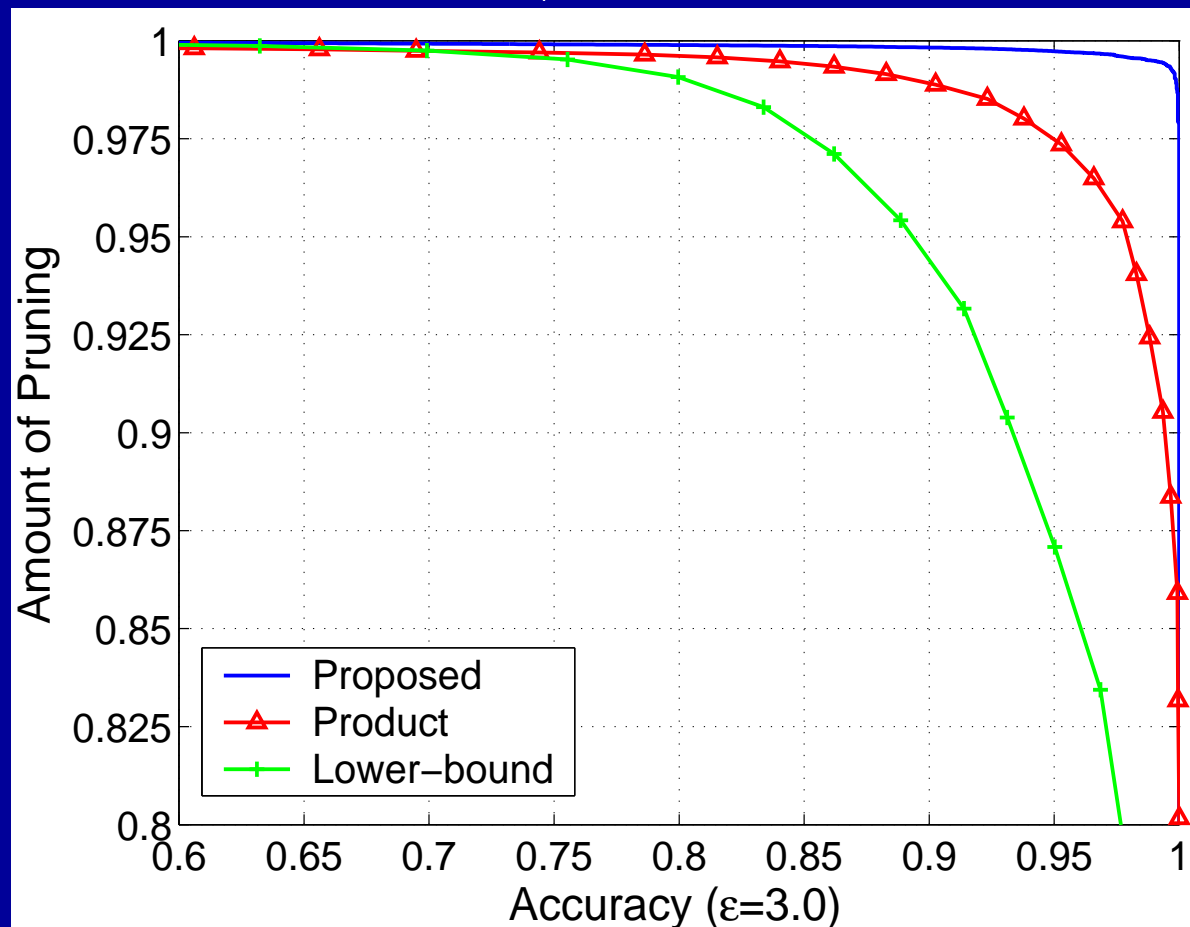
100-d vector

Dimension Reduction:

1. **Pruning:** remove x in the database that satisfy $l_2(T^*(x), T^*(q)) > \lambda$
2. **Refinement:** Identify x that satisfy $d(x, q) \leq \varepsilon$ in the remainder of the database.

Performance of Version 1

1000 random queries on a database
of 46,331 web video



$\dim(T^*(x))$ is still too high! Proposed Technique (Version 2)

Dimension Reduction:

1. For each signature vector x , compute $T^*(x)$ given by

$$T : x \rightarrow \underbrace{(d(x,s_1)^2, d(x,s_2)^2, \dots, d(x,s_m)^2)}_{100\text{-d vector}}$$

712-d histogram

2. Project each $T^*(x)$ into $T^{**}(x)$ with PCA

$$T^{**} : T^*(x) \rightarrow T^{**}(x)$$

100-d vector 4 to 8-d vector

Similarity Search for q :

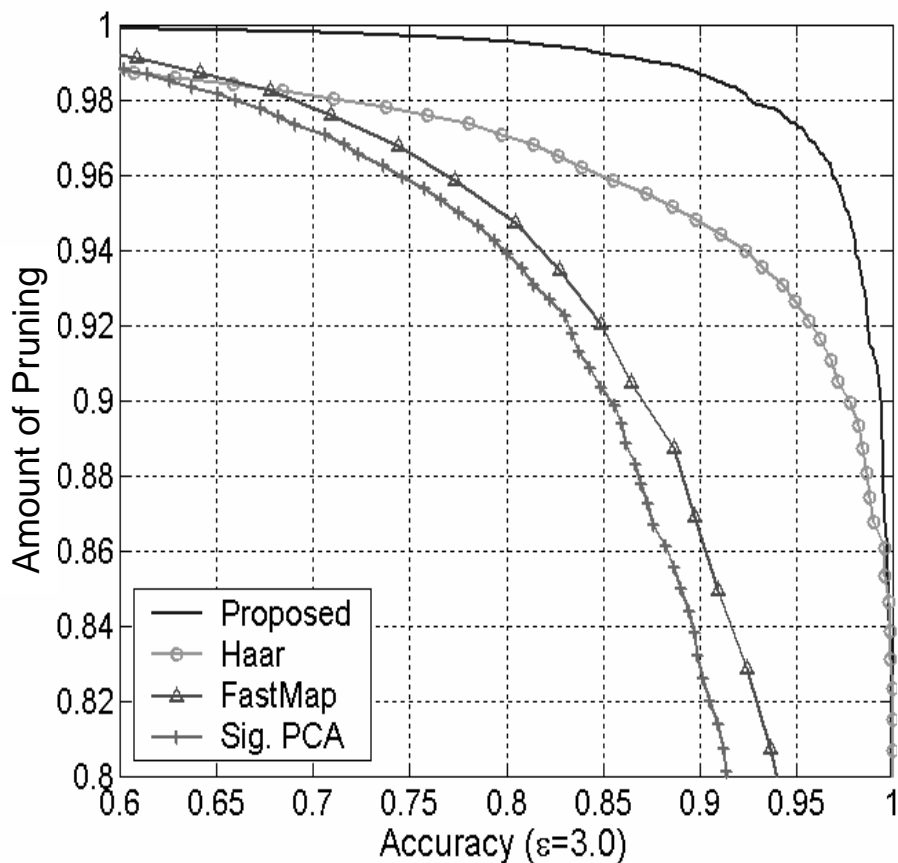
1. **Pruning:** remove x in the database that satisfy $l_2(T^{**}(x), T^{**}(q)) > \lambda$
2. **Refinement:** Identify x that satisfy $d(x,q) \leq \varepsilon$ in the remainder of the database.

Comparisons

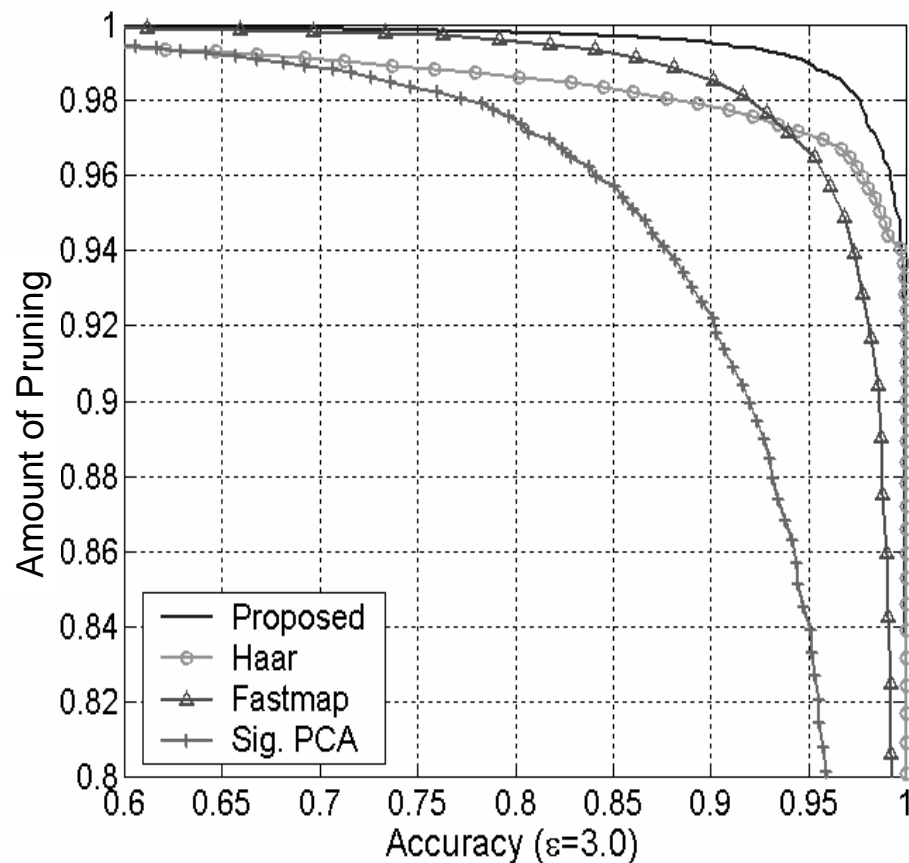
- Apply PCA directly on signature vectors
- Fastmap [Faloutsos, Lin 95] - another randomized technique for general metric
- Haar Wavelet [MPEG-7] - fixed transform for color histogram with l_1 metric
- Experiments: compress 100-d vectors into 4-d and 8-d vectors

Performance of Version 2

4-D Feature Extraction



8-D Feature Extraction



Speed test on Intel Xeon (550Mhz, 1GB)

Average of 100 random queries on a database of 23,206 entries

<i>Scheme</i>	<i>Accuracy</i>	<i>Pruning Time (uses 8-d vectors)</i>	<i>Refinement Time (uses 712-d vectors)</i>
Sequential search (histogam)	100%	--	6730 ± 35 ms
PCA	89%	130 ± 1.4 ms	401 ± 75 ms
Haar	92%	152 ± 1.3 ms	123 ± 28 ms
Fastmap	91%	131 ± 1.5 ms	75 ± 11 ms
Proposed	89%	131 ± 0.8 ms	33 ± 8 ms

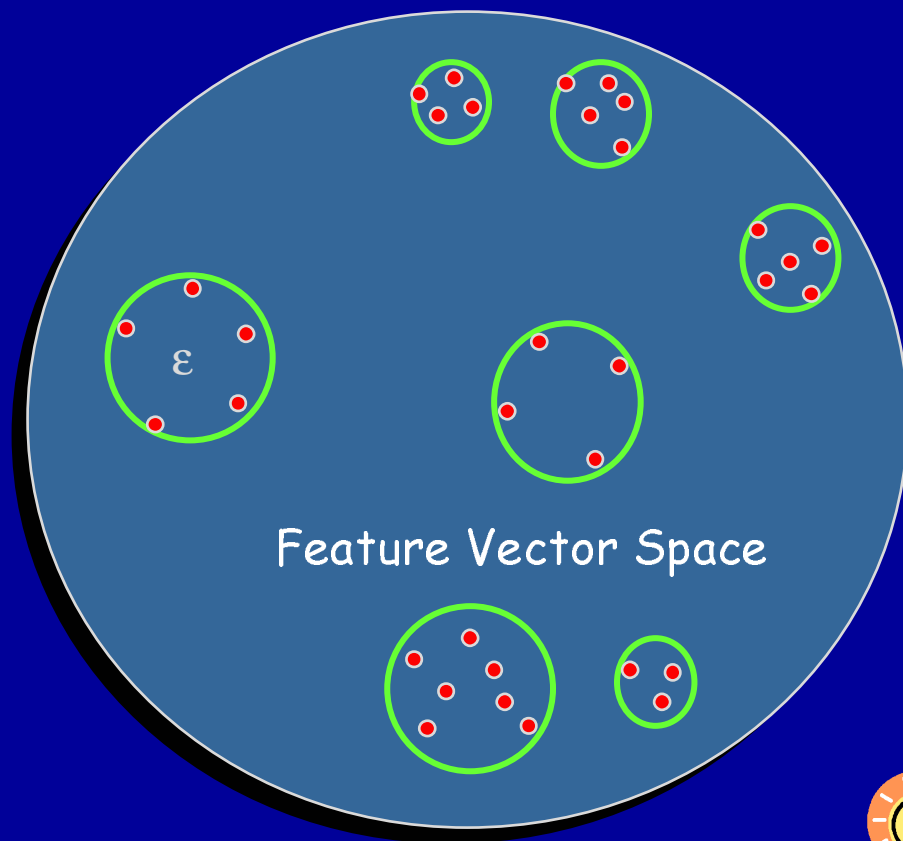
Outline

- Problem, motivation and overview
- System components:
 - video signature
 - fast similarity search
 - signature clustering
- Search engine demo
- Summary and Future work

Why Clustering?

- Intuitive organization of similar data
- Sum is bigger than parts - use data to rectify error:
 - Possible sampling error in signature similarity
 - Possible pruning error in fast search
- Select threshold based on local statistics of data
- Number of clusters very large and unknown
 - difficult to apply optimization-based clustering like K-means
 - Compare with hierarchical clustering

Cluster Definition



A cluster of similar videos C :

- $d(V, W) \leq \epsilon_C$ for most $V, W \in C$
 - $d(V, U) > \epsilon_C$ for all $V \in C, U \notin C$
- ϵ_C varies from cluster to cluster.



“Almost complete”
“Complete” connected
components at different
distance thresholds.

Almost-complete Clustering

Need data structure to keep track of connectedness

Thm: Clustering-forming thresholds must be part of the *Minimum Spanning Forest*



Edge Density $\geq \lambda$

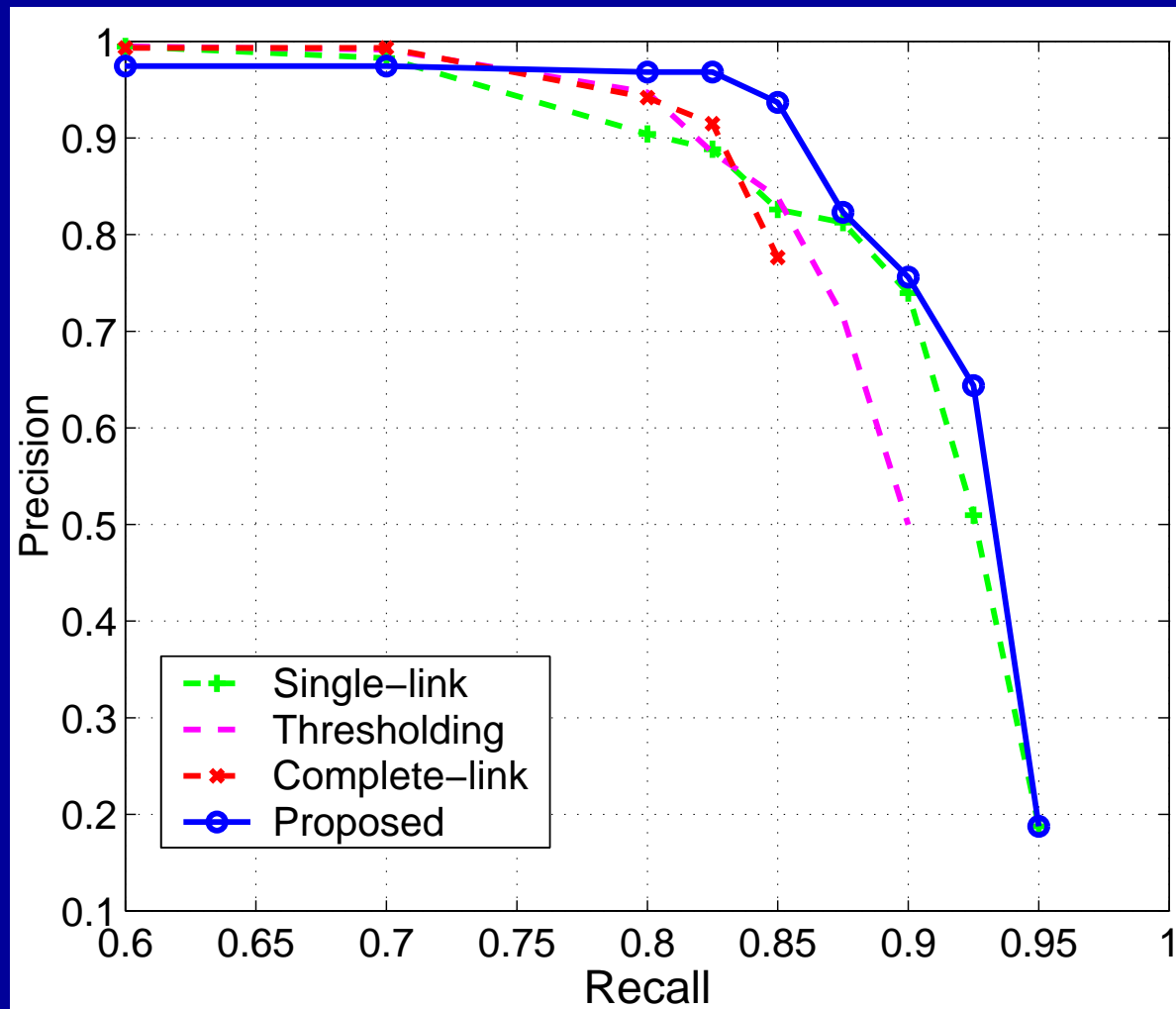
Build sparse graph by similarity search on each point with ϵ_0



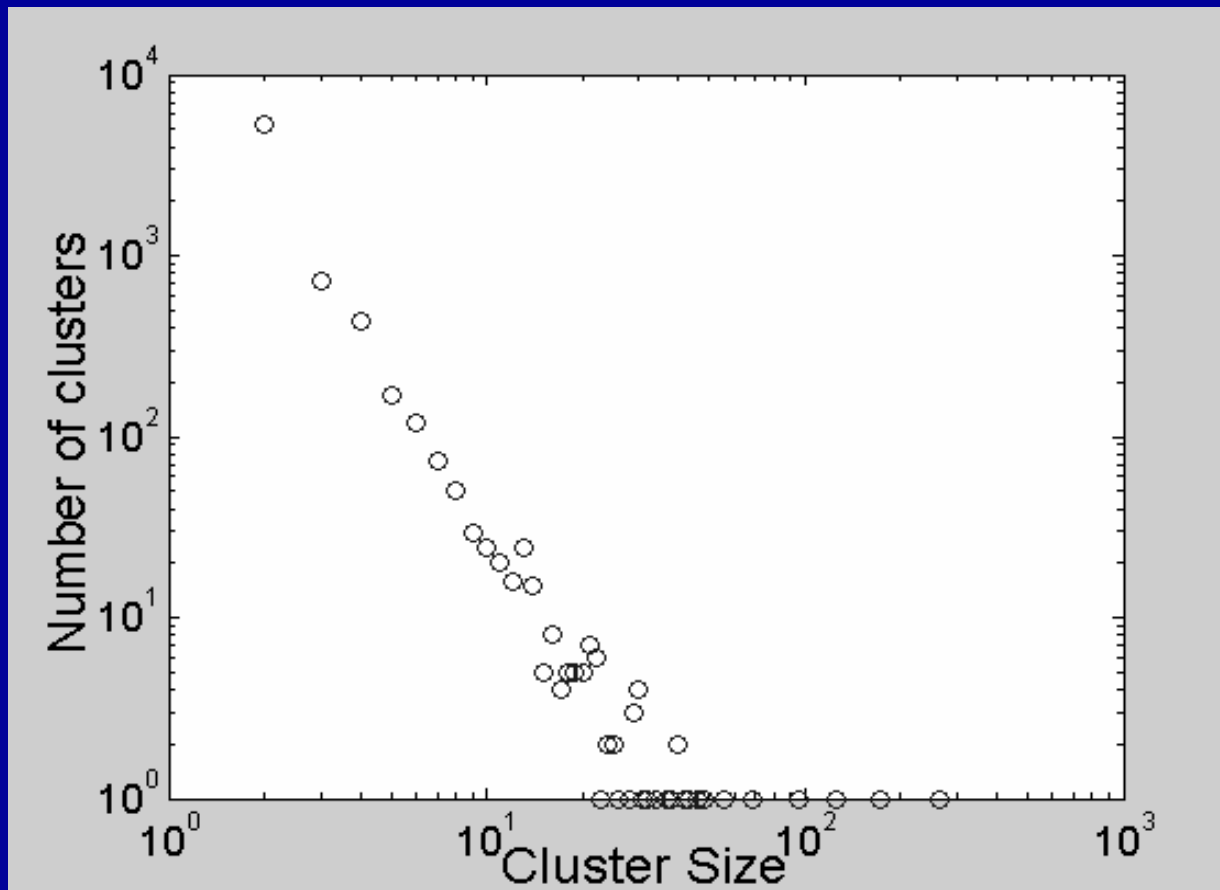
Implementation using MSF

1. Use modified Kruskal Algorithm to compute MSF
 - Sort edges and examine in increasing order
 - Keep track of edge densities on either side of a new branch (at no additional cost)
2. Consider components on either side of the next longest branch length of the MSF.
3. Output components as clusters if their edge densities $\geq \lambda$.
4. Back to step 2 until no component remains.

Retrieval Results

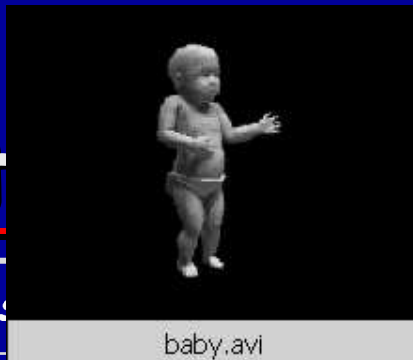


Video Cluster Statistics

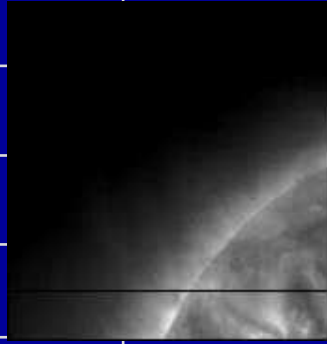


Average number of similar copies = 2.95

Ten largest



Size	Location Diversity	Misclassified	Description
263	0.12	Y	Share red-letter intro
172	0.70	N	Sharing Baby from "Ally McBeal"
126	0.43	Y	Share "MTV News" intro
95	0.01	Y	Share "Chv.Net" intro
68	0.01	N	For message from Chv.Net
56	0.98	N	Angry Man hitting computer
48	0.19	N	Mathematical plots of wave equation
46	0.09	N	Different segments of Clinton's testimony
43	0.42	N	SOHO Astronomical Video
42	0.08	N	Synthetic Aperture Radar Video

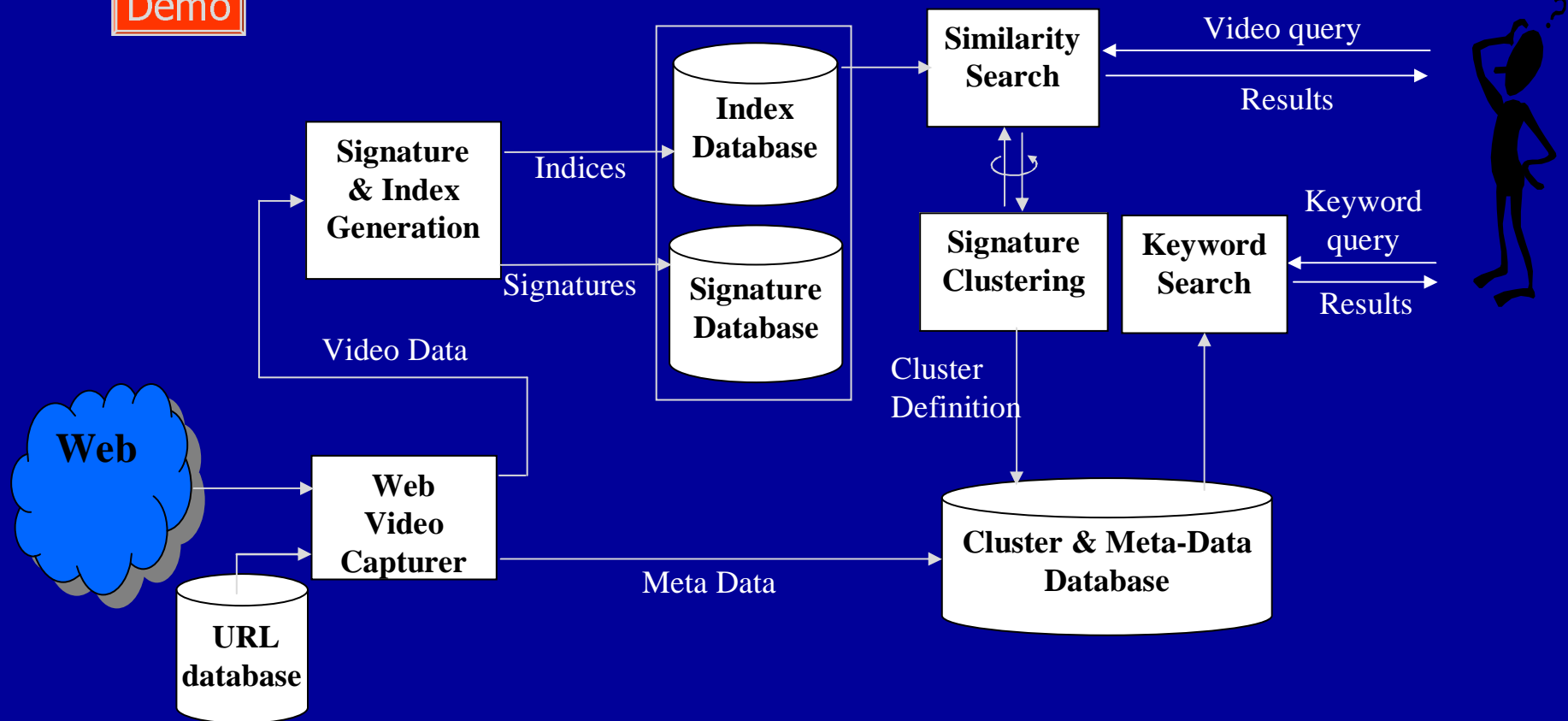


Outline

- Problem, motivation and overview
- System components:
 - video signature
 - fast similarity search
 - signature clustering
- Search engine demo
 - <http://www.eecs.berkeley.edu/~cheungsc/cluster>
- Summary and Future work

Prototype System Architecture

Demo



Computation time

<i>Processing</i>	<i>Time (500 MHz Xeon, 1GB)</i>
Decoding [◆] + Signature	~ 1.6 hr/1 hr of video
Similarity Search	140,000 sig./second
Clustering of 46,000 sig. ($\epsilon_0=4.0$)	4 minutes [▲]

- ◆ 3 frames per second
- ▲ Excluding signature distance computations

Summary

- Video Signature for fast similarity measurement
 - results on simulated data and web data
- Triangle-inequality based dimension reduction
 - out-perform Fastmap, Haar wavelet, Lower-bound
- Graph-theoretical clustering algorithm
 - out-perform thresholding and Hierarchical clusterings
- Web video statistics and search engine
- Acknowledgement: Professor Avideh Zakhor, funding from AFOSR, NSF, CA DIMI, Hughes Research

Future Work

- Media signature/finger-printing/hashing for content identification and measurement
 - more general and less intrusive than watermarking
- Signature as a general approach for streaming data summarization
 - network intrusion, virus scanning (real & computer)
- Sub-sequence identification and statistics
 - identify the most often occurred subsequences
- Randomized vs. Statistical representation
 - signature vs. PCA+Mixture of Gaussian

Other work (current/future)

- Scientific image processing and computer vision
 - for comparing simulation and experimental results in computational fluid dynamics
 - remote sensing
- Event mining in video
 - robust background subtraction
 - traffic analysis