Locally Embedded Subspaces for Efficient Video Indexing and Retrieval

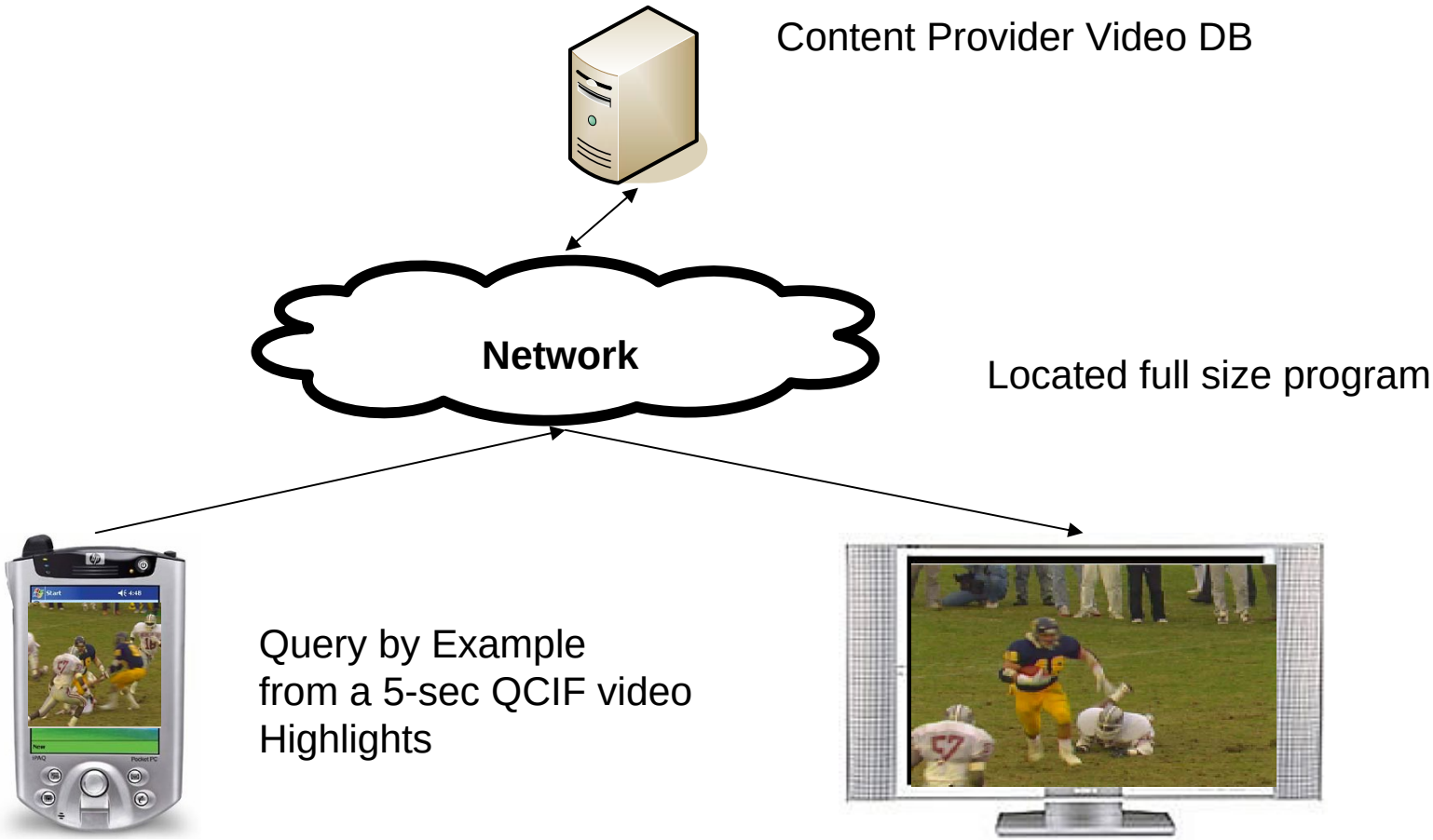
Aggelos K. Katsaggelos

Dept of EECS, Northwestern University, Evanston, IL
www.eecs.northwestern.edu/~aggk

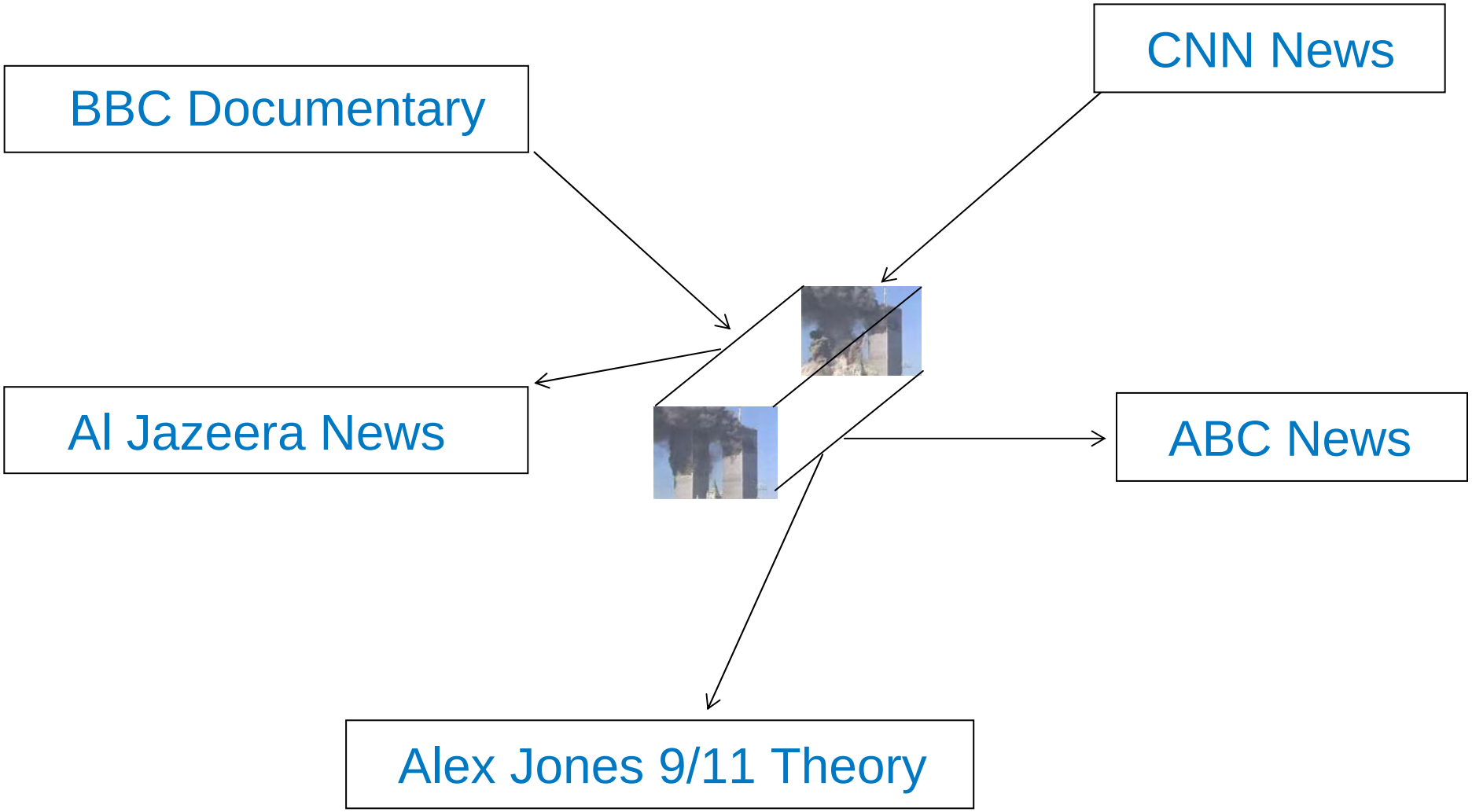
The Grand Challenges

- **How to manage large video repositories ?**
  - Video content, both personal and produced, are growing explosively.
  - Need to facilitate efficient searching and browsing of video content with very large collections
  - Provide linkage among video programs via common video segments
  - Repeat clip detection, copyright protection

Motivation


Content Provider Video DB
Network
Located full size program
Query by Example
from a 5-sec QCIF video
Highlights
Mobile
TV Set

Motivation – Video Repository Linkage



Interpretable Image Features Based Approaches

- **Color, Shape, Texture**
  - Color histogram, Color moments, Color Layout
  - Shape (who's to do segmentation ?)
  - Texture (wavelet decomposition based)
- **Motion**
  - Motion vector statistics
  - Camera motion
- **Object level information**
  - Object detection and tacking
- **Problems**
  - Computational complexity
  - High dimensionality of feature space

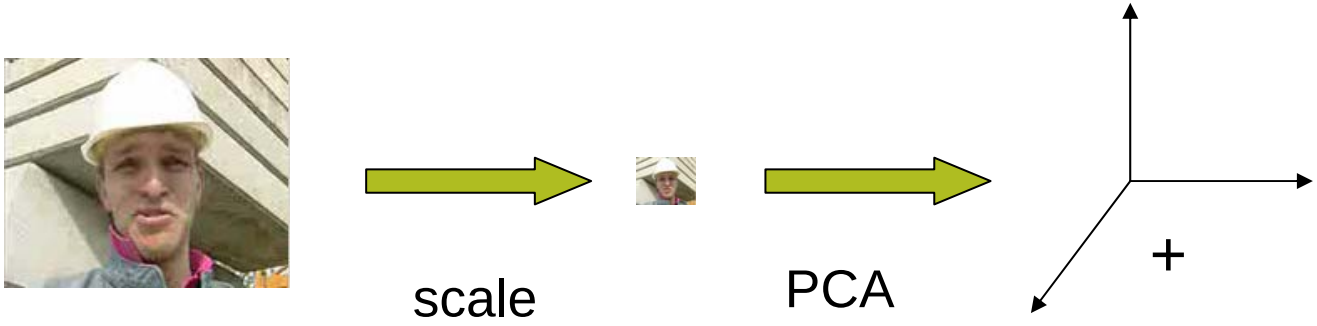
Proposed Approach

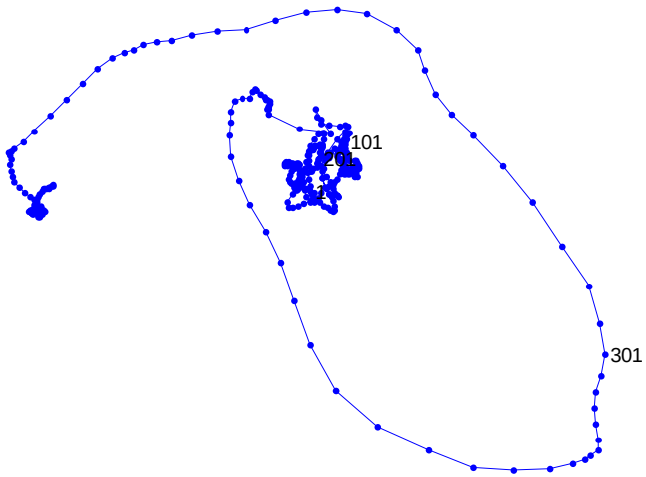
- **Matching by luminance field trace**

- **A video sequence of frame size WxH, F(k), can be captured as a trace in R^{WxH}**

- **Find a compact approximation of trace as, x(k), of F(k), in some lower dimensional space R^d, such that this trace is indexing efficient, search robust**

Luminance Field Trace (LUFT)



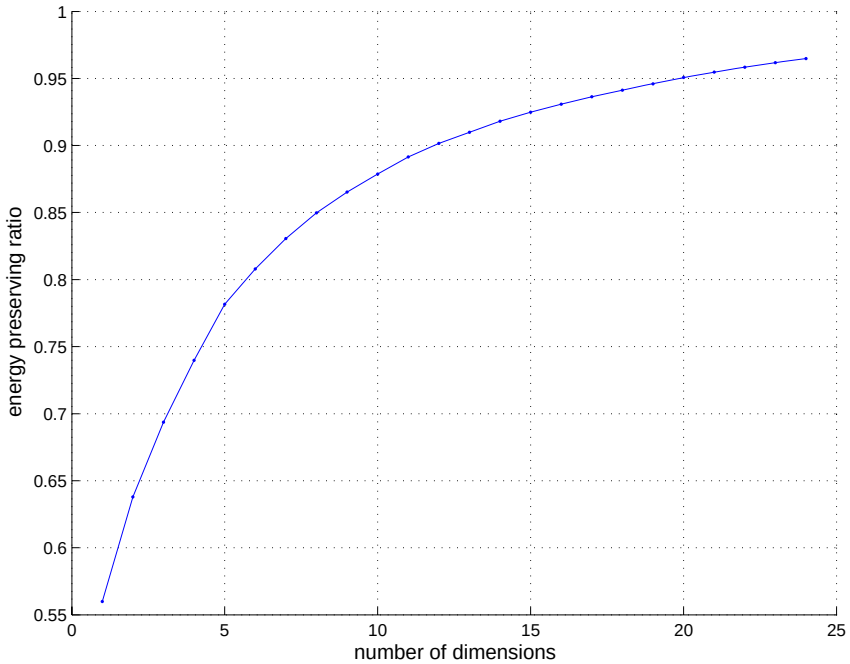


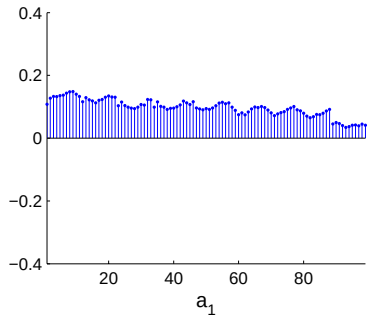
- Scaling to a common spatial scale, for example,11x9 for noise reduction and handling frame size variation
- PCA to identify the trace residing subspace in R^{11x9}.

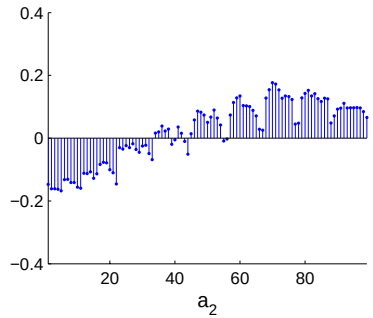
"foreman" seq in 2D (1st and 2nd component) PCA space

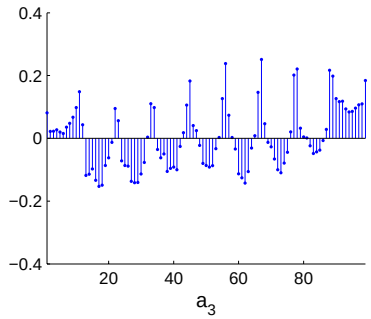
LUFT PCA Basis

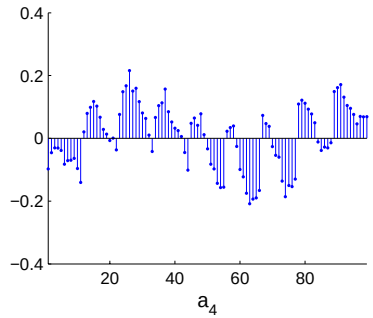
- **PCA basis functions**
  - Scale: w=11, h=9
  - For *d*=20, 95% info preserved





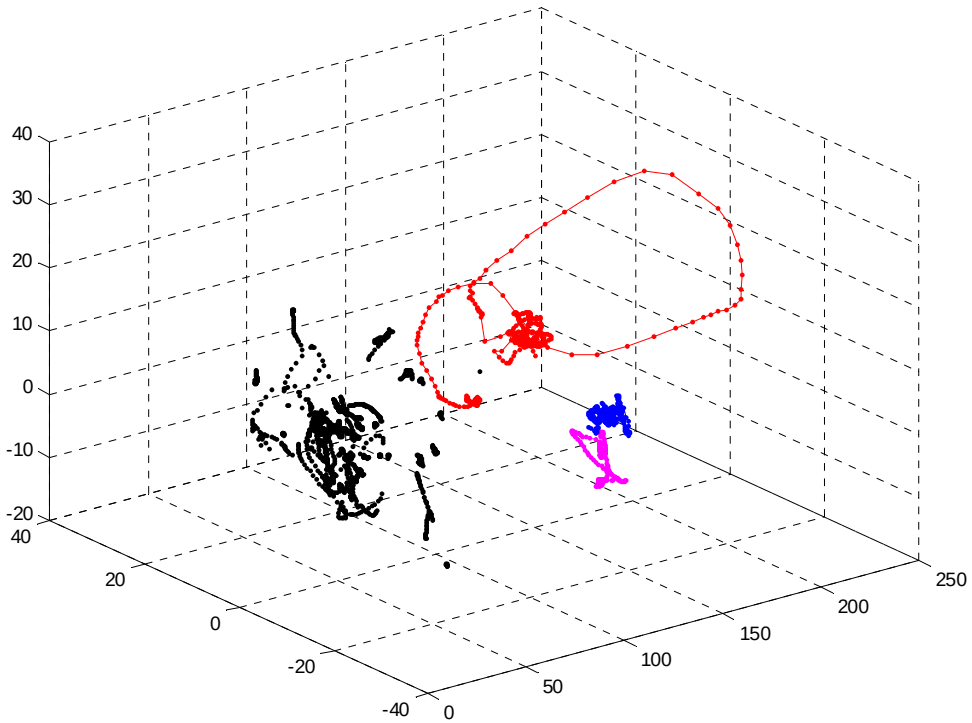






Video Trace Examples

video as trace in PCA space with 1st,2nd and 3rd components



- "foreman" : 400 frames
- "stefan" : 300 frames
- "mother-daughter": 300 frames
- "mixed": 40 shots of 60 frames each from randomly selected sequences.

. "foreman", . "stefan", . "mother-daughter" , . "mixed"

Matching metrics

- **For two video traces, *m*-frame Q and *n*-frame T, the projection distance is given by:**

$$d(Q,T) = \frac{1}{m} \min_{k_1,k_2,\cdots,k_m} \sum_{j=1}^{m} \| q_j - t_{k_j} \|$$

$$= \frac{1}{m} \min_{k} \sum_{j=1}^{m} \| q_j - t_{k+j-1} \|$$

- **The matching is found by:**

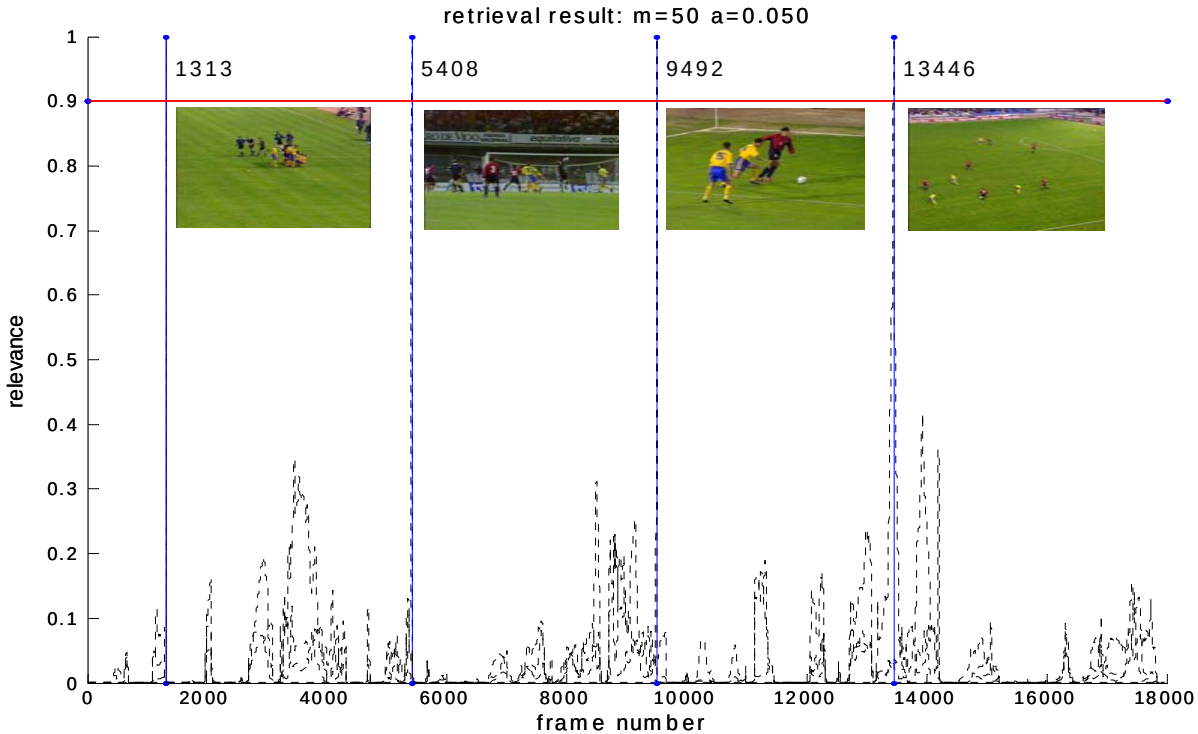
$$d(k;Q,T) = \frac{1}{m} \sum_{j=1}^{m} \| q_j - t_{k+j-1} \|$$

$$k^* = \begin{cases} \arg\min_k d(k;Q,T), & if\ d(Q,T) < d_{\min} \\ not\ exist \end{cases}$$

Matching Metric Example

- **Some query clips and their normalized projection distances**
  - Relevance value R_k (in [0..1.0])

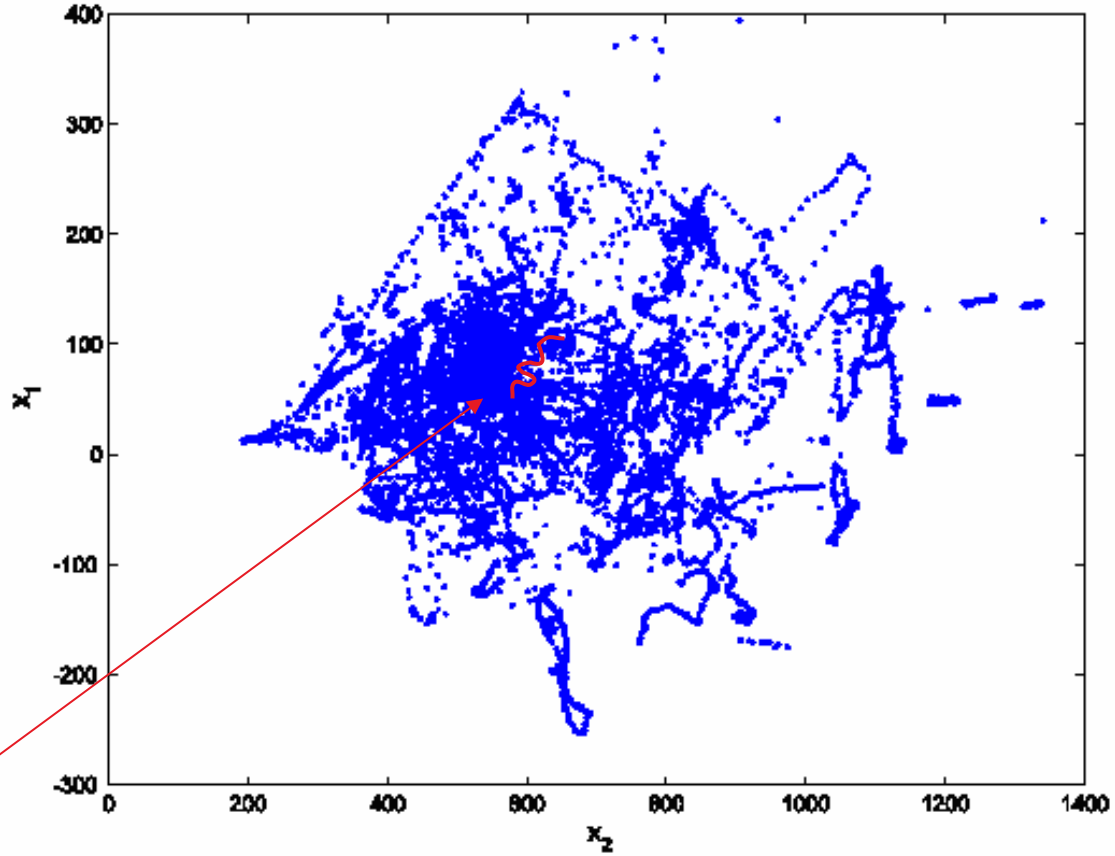
$$R_k = e^{-\partial d_k(Q,T)}$$



- **Four 2-sec query examples from an 18000-frame soccer game program**

Indexing Scheme

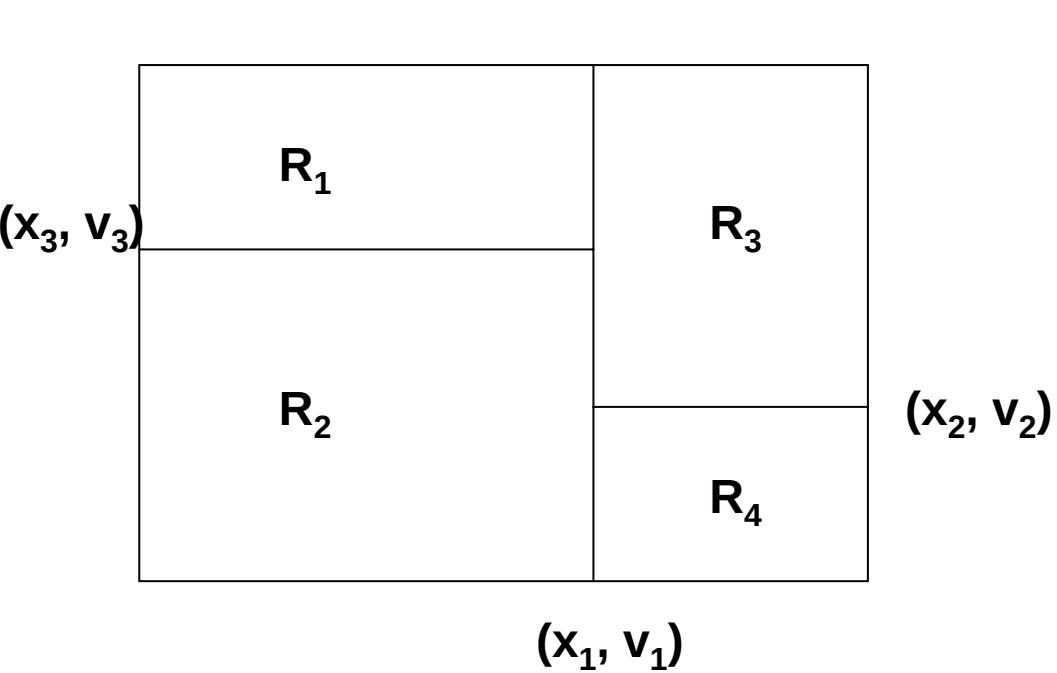
- **For large video collections, exhaustive search is not efficient**
- **Need to have efficient indexing scheme**

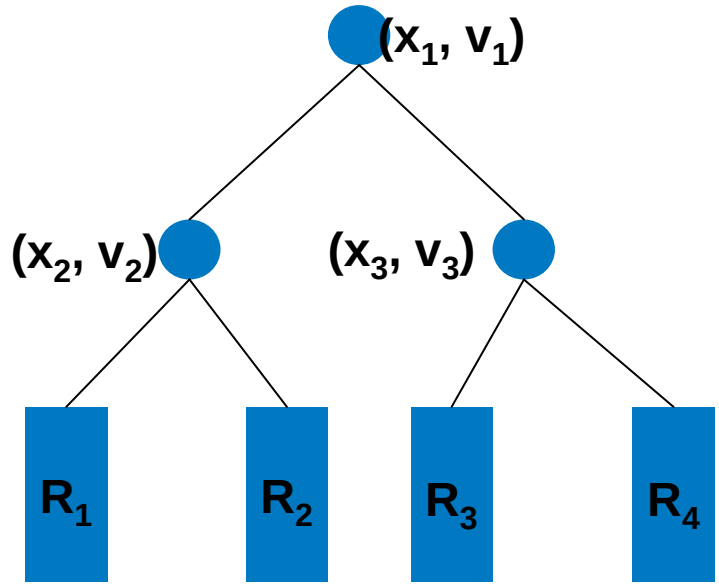


Example of video traces of 50K frames from TRECVID

Kd-Tree Type Scheme

- **Iterative max variation cut at median value**
- **After each cut, compute an MBB – Min Bounding Box for each child node**
- **Store cutting plane and value, as well as MBB at each node**

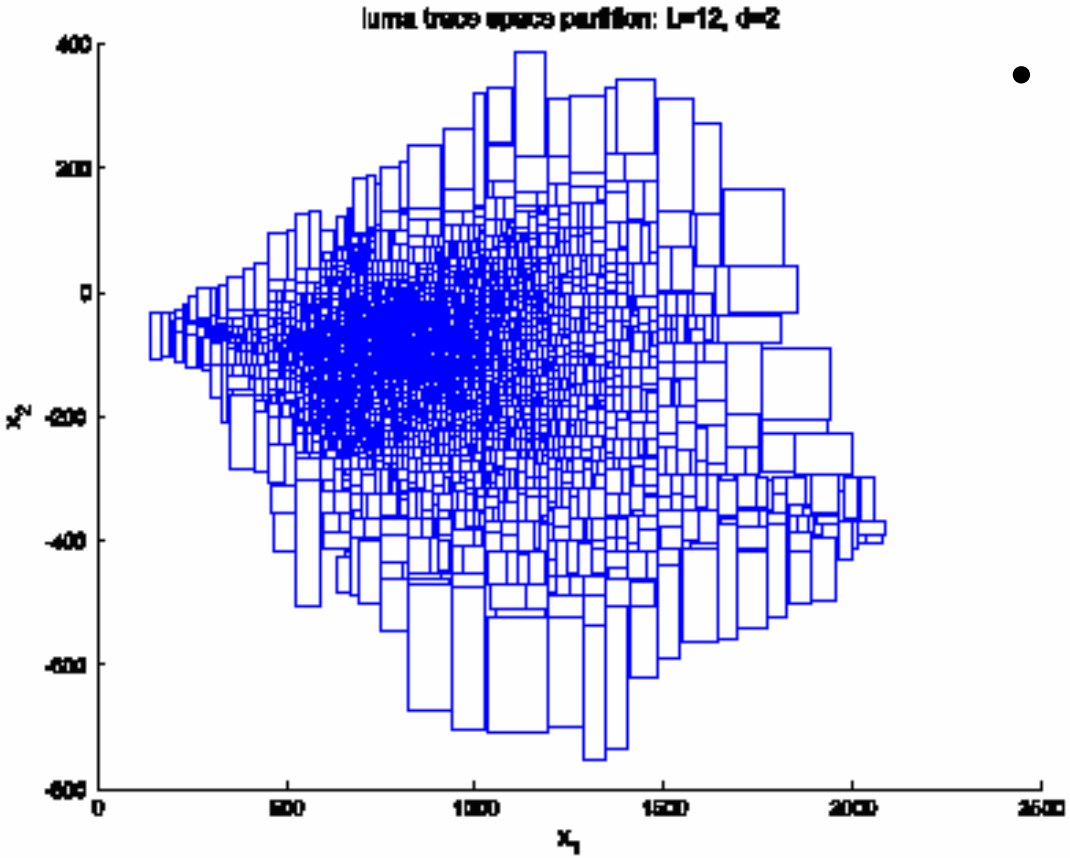




- **At retrieval time, query clip is traversing the tree by MBB intersections and splits**

Indexing Scheme

- **MBB partition of the LUFT space**
  - Always split at max variation dimension.



- **Example:**
  - For 5 hours of video from NIST TRECVID
  - An index tree of 12 levels, and 4096 leaf nodes level MBBs are plotted. Each node has about 132 frames
  - Indexing space dimension shown d=2.

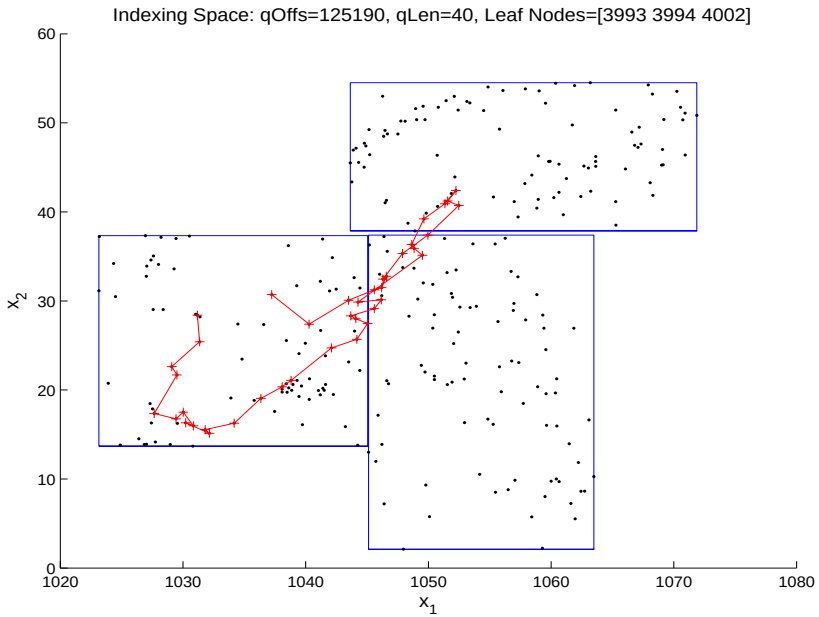
  - Time to build this index:530 sec on an 2.4GHz Celeron/256M RAM Laptop in Matlab, not bad at all.

Retrieval with Locally Embedded Metrics

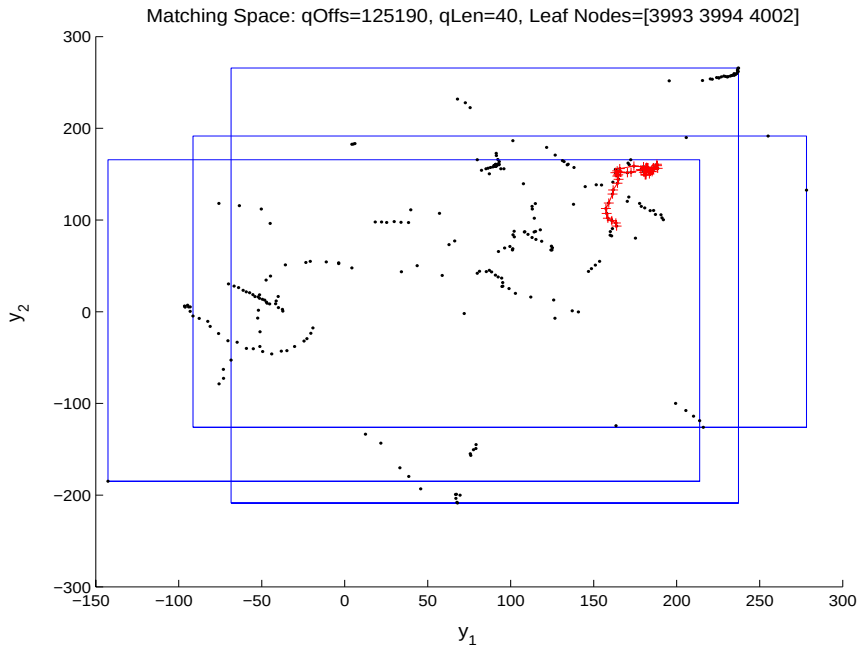
- **Retrieval space may not need to be the same as indexing space**
- **For a query clip, the index tree helps identify a number of leaf nodes that were traversed by the query clip**
  - a coarser process that can afford larger errors in the video trace
- **The actual computation of the retrieval distance can be locally adjusted**
  - For better accuracy and robustness
  - e.g., through a locally trained PCA subspace
  - More advanced technique that find a best trace separating subspace via LDA, Graph Laplacian framework

Locally Embedded PCA subspace example

- **Train a PCA metric based on a sub set (sub tree) of traces**
- **Gives better resolution by localization**



Traces in global indexing space



Traces in locally embedded PCA space

Indexing and Retrieval Performance

- **Indexing Efficiency:**
  - For query clip *Q*, of length *m*, over matched clips length, *N*(*Q*) -- perfect efficiency: 1.0

$$\eta(Q) = \frac{m}{N(Q)}$$

- **Retrieval Complexity:**

$$C(m, L, \eta) = C_1 + C_2 = t_1 O(mL) + t_2 O(m^2 / \eta)$$

  - Query clip length, m
  - Index tree levels, L
  - C_1: related to traversing the index tree to locate leaf nodes
  - C_2: actual matching complexity

Retrieval Performance

- **Retrieval Performance:**
  - Data set 1: 200,000 frames, indexed Data set 2: 100,000 frames, as negative query
  - Randomly set up 800 queries with query clip of 0.5 to 4 sec in length
  - T_1: time to traverse the indexing tree
  - T_2: actual trace matching
  - Error Rate < 1%
  - Average retrieval time: 10 to 55 ms.

Retrieval Performance

Table 1. Indexing/Retrieval Performance, $L=12$, $d_{indx}=2$

Data Set	m	ErrRate (e/200)	T_1 (ms)	T_2 (ms)	mean(η)
1	*15*	*1/200*	*1.0*	*15.9*	*0.036*
1	*30*	*1/200*	*3.1*	*27.5*	*0.042*
1	*45*	*0/200*	*3.0*	*43.8*	*0.045*
1	*60*	*0/200*	*5.5*	*49.4*	*0.051*
2	*15*	*0/200*	*1.7*	*8.8*	*0.049*
2	*30*	*0/200*	*3.2*	*19.3*	*0.060*
2	*45*	*0/200*	*4.0*	*25.3*	*0.060*
2	*60*	*0/200*	*7.7*	*34.4*	*0.072*

Table 2. Indexing/Retrieval Performance, $L=12$, $d_{indx}=3$

Data Set	m	ErrRate (e/200)	T_1 (ms)	T_2 (ms)	mean(η)
1	*15*	*0/200*	*1.3*	*9.5*	*0.049*
1	*30*	*0/200*	*2.1*	*19.5*	*0.064*
1	*45*	*0/200*	*2.7*	*23.2*	*0.070*
1	*60*	*0/200*	*6.5*	*32.7*	*0.079*
2	*15*	*0/200*	*1.9*	*5.2*	*0.849*
2	*30*	*0/200*	*2.6*	*9.2*	*0.115*
2	*45*	*0/200*	*4.6*	*14.5*	*0.125*
2	*60*	*0/200*	*7.1*	*19.8*	*0.126*

Retrieval Performance

- **Larger set: 5 hrs TRECVID**
  - Error Rate < 1% all

d	m	T_1 (ms)	T_2 (ms)
2	*30*	*1.7*	*25.3*
2	*60*	*3.1*	*56.6*
4	*30*	*2.0*	*14.3*
4	*60*	*4.3*	*31.8*

Summary

- **LUFT is an efficient and robust scheme for video shot retrieval**
- **The indexing scheme developed scales well with large collection data bases**
- **Applications in QBE, repeated clips detection, piracy detection, video clip linkage.**
- **Future work:**
  - A bottom up indexing approach to have better organized leaf nodes traces.
  - Localized linear/kernel space modeling for better robustness in retrieval
  - Handle editing induced variations, eg., graphical logos, cuts
  - Application in video data mining (NIST TRECVID)

Rate-Distortion Optimla Video Summarization

Aggelos K. Katsaggelos

Dept of EECS, Northwestern University, Evanston, IL
www.eecs.northwestern.edu/~aggk

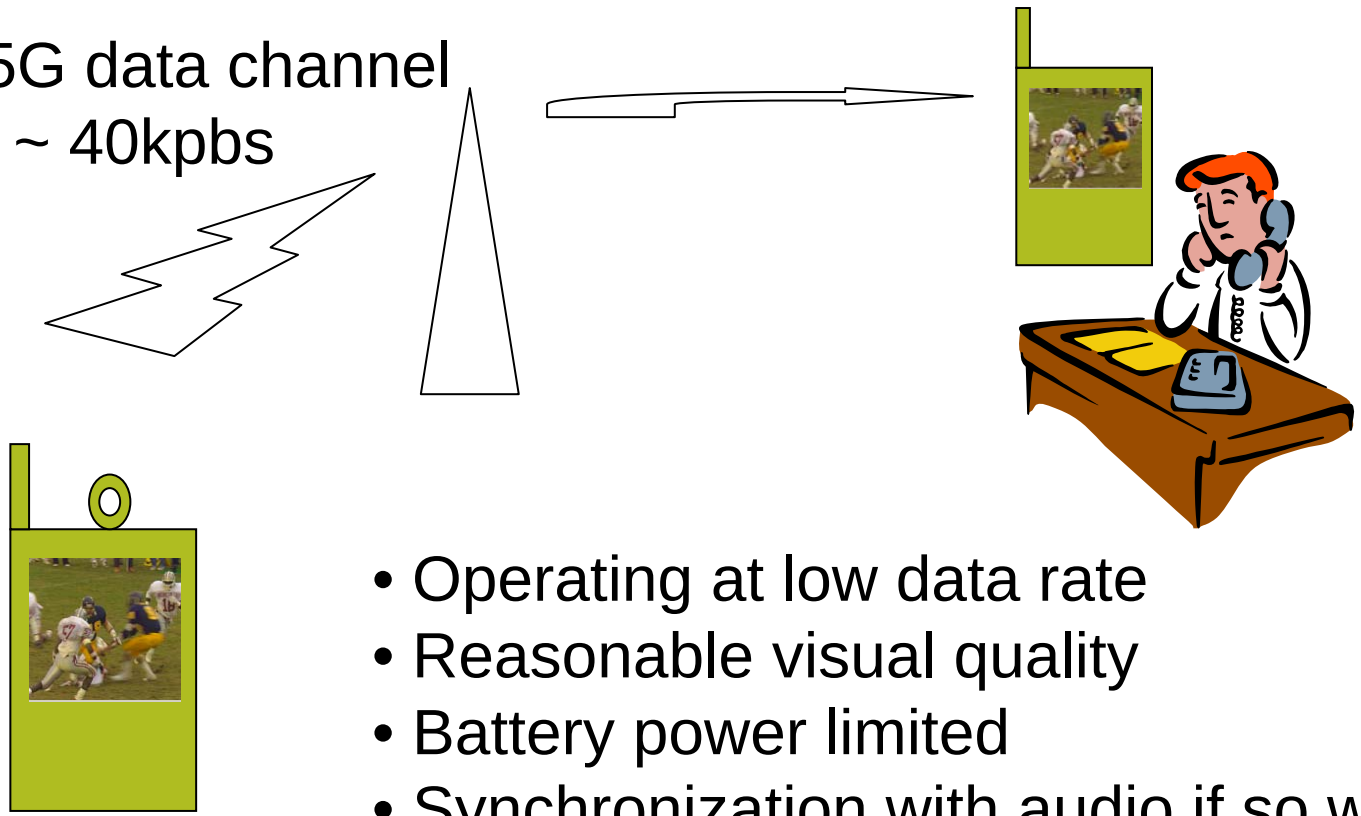
Introduction

Why Video Summary ?

- Viewing time constraint, a shorter version is more desirable in some applications, for example, a 2-min summary of an one-hour surveillance video.
- Bit rate constraint, Storage and bandwidth limit the bit rate of a video sequence. A shorter version with better SNR quality conveys more useful information.
- Energy constraint, mobile communication devices' operations are battery energy limited. How to communicate more information with less energy ? (circuits vs communication energy cost)

An example

2G/2.5G data channel
4kpbs ~ 40kpbs



- Operating at low data rate
- Reasonable visual quality
- Battery power limited
- Synchronization with audio if so wish

Organization of the presentation

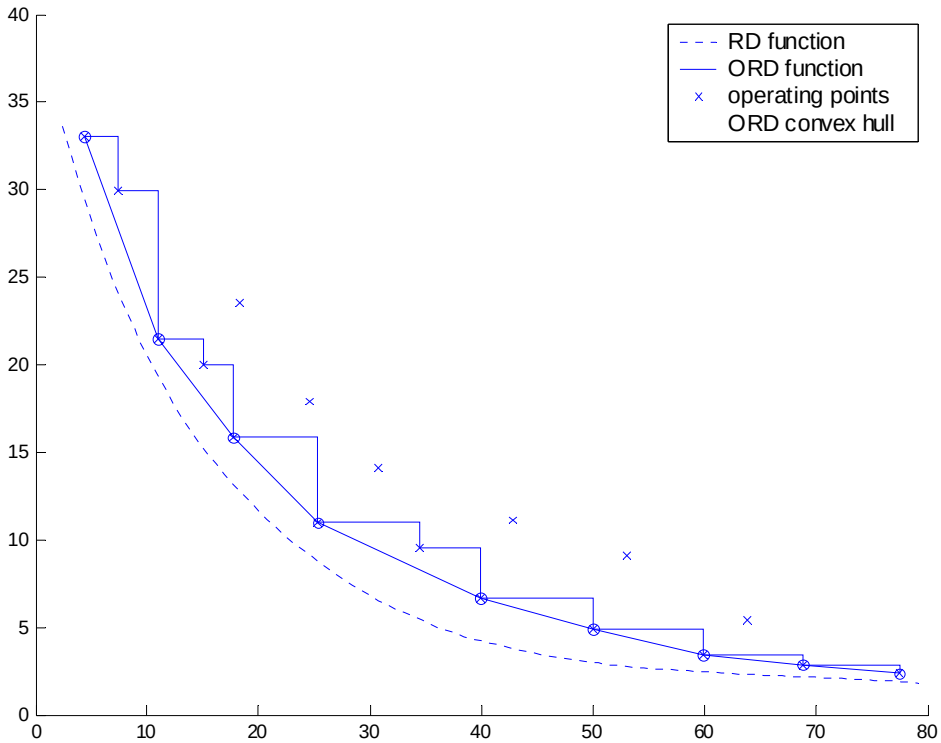
Video summarization

- Rate-distortion optimization based formulations
- Solution to the MINAVG problem
- Solution to the MINMAX problem

Frame distortion metric and shot segmentation

- Frame distortion metric
- Shot segmentation based on order statistics

Operational Rate-Distortion Theory



$$R_{op}(D) = \min_{Q_j} R(Q_j),\ s.t.\ D(Q_j) \le D$$

- Gives the "operational" R-D performance curve.
- $\{Q_j\}$: operating points associated with coding decisions and parameters
- Optimization: select operating points that minimizes distortion for a given rate, or minimizing rate for a given distortion.

Definitions and assumptions

Video Sequence (*n*-frame): V= $\{f_0, f_1, \cdots f_{n-1}\}$

Video Summary (*m*-frame): $S = \{f_{l_0}, f_{l_1}, \cdots f_{l_{m-1}}\}$

Reconstructed Sequence: $V_S' = \{f_0', f_1', \cdots f_{n-1}'\}$

Where: $f_k' = f_{i=\max(l):s.t.\ l\in\{l_0,l_1,\cdots,l_{m-1}\},\ i\le k}$

Definitions and assumptions

Summary Distortion (Average):

$$D(S) = \frac{1}{n}\sum_{j=0}^{n-1} d(f_j, f_j')$$

Summary Distortion (Maximum):

$$D(S) = \max_k d(f_k, f_k')$$

Summary Rate (Temporal):

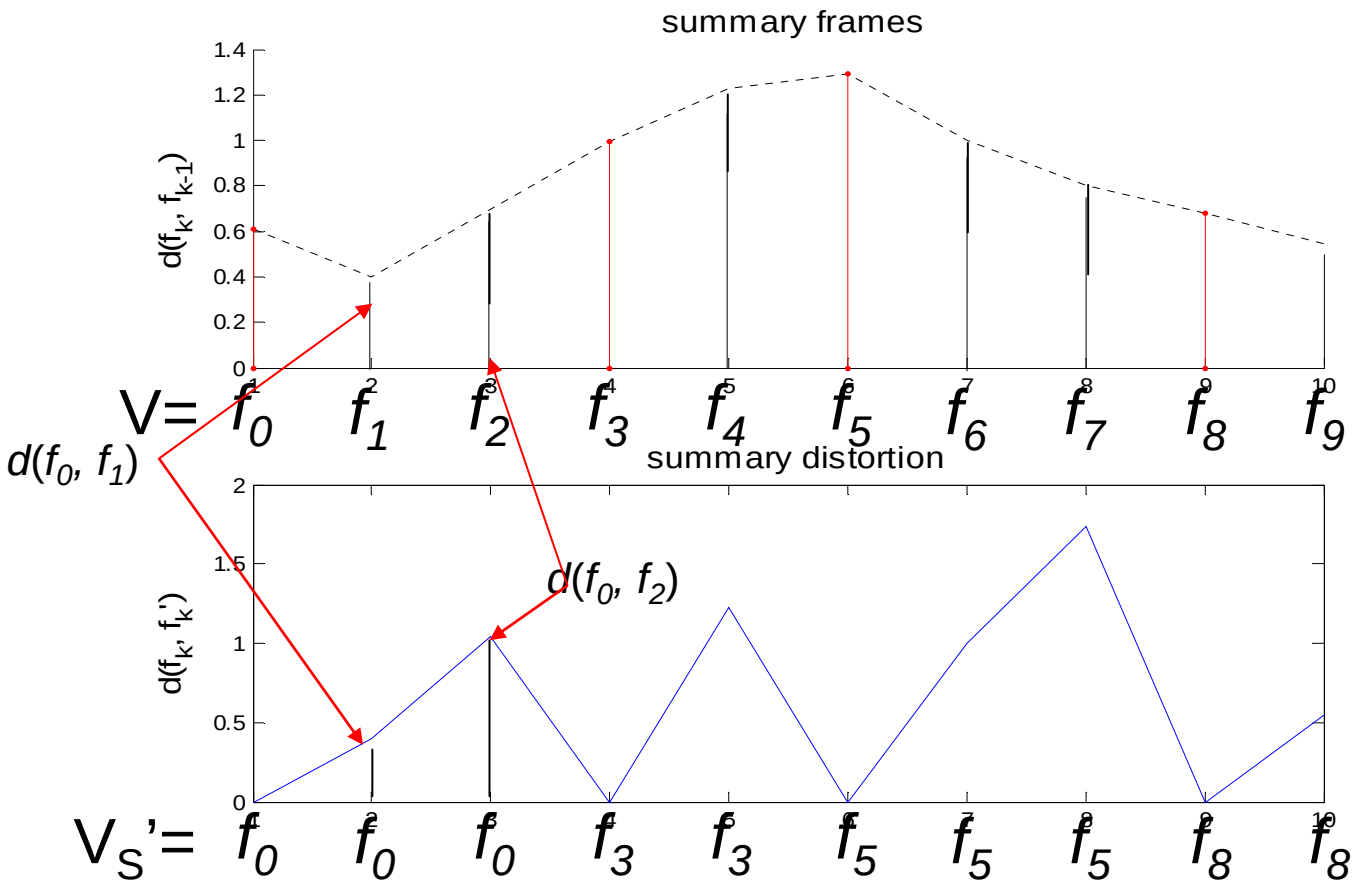
$$R(S) = \frac{m}{n}$$

Summary Rate (Bit):

$$R(S) = \sum_{j=0}^{m-1} b(f_{l_j})$$

An example:

n=10, **S={f_0, f_3, f_5, f_8 }** ***, m=4, D^{avg}=0.6, D^{max}=1.73,***



R-D Optimization Formulation

Summarization as a rate-distortion optimization problem

- MDOS (Min Distortion Optimal Summary) formulation:

$$S^* = \arg\min_S D(S),\ s.t.\ R(S) \le R_{\max}$$

- MROS (Min Rate Optimal Summary) formulation:

$$S^* = \arg\min_S R(S),\ s.t.\ D(S) \le D_{\max}$$

- Frame skip constrained MDOS and MROS:

$$S^* = \arg\min_S D(S),\ s.t.\ R(S) \le R_{\max}, \text{ and } l_k - l_{k-1} \le K_{\max} + 1,\ \forall k$$

$$S^* = \arg\min_S R(S),\ s.t.\ D(S) \le D_{\max}, \text{ and } l_k - l_{k-1} \le K_{\max} + 1,\ \forall k$$

MINAVG Optimal Summarization

When the average frame distortion is used as summarization distortion:

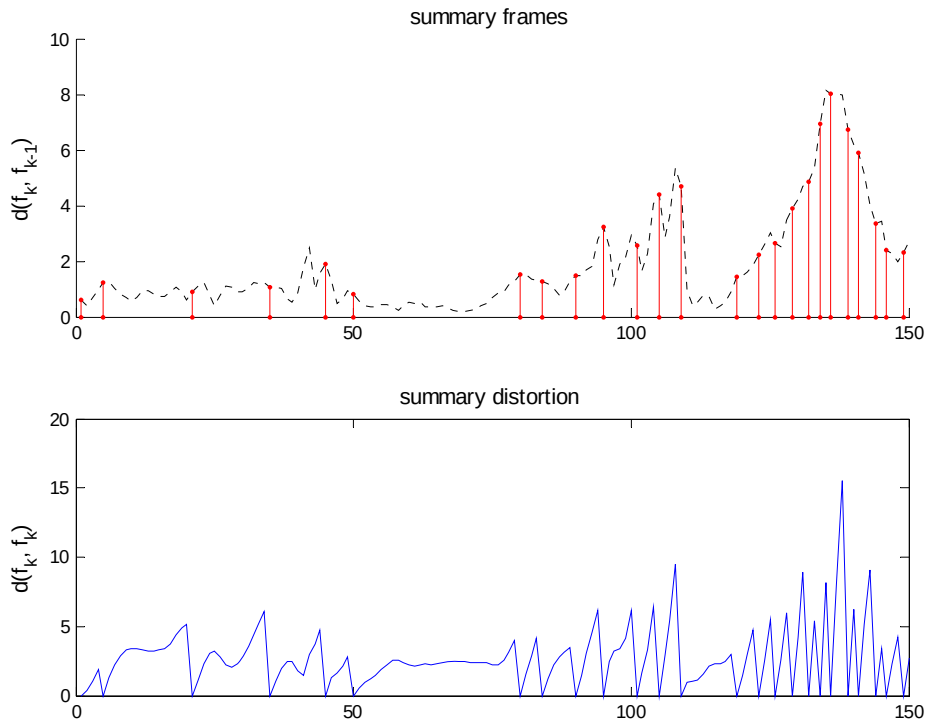
- Temporal rate based formulations:
  - •Dynamic Programming (DP) solution to the MDOS formulation
  - • Bi-section searching on the Operational RD function to solve MROS formulation
- Bit rate based formulations:
  - •Lagrangian relaxation + DP solution to the MDOS and MROS formulations
  - •Skip frame constraint

Key to the solution:

- Distortion state recursion
- Viterbi algorithm like DP solution: compute optimal solution at each stage and at the final stage, back track for the optimal solution.

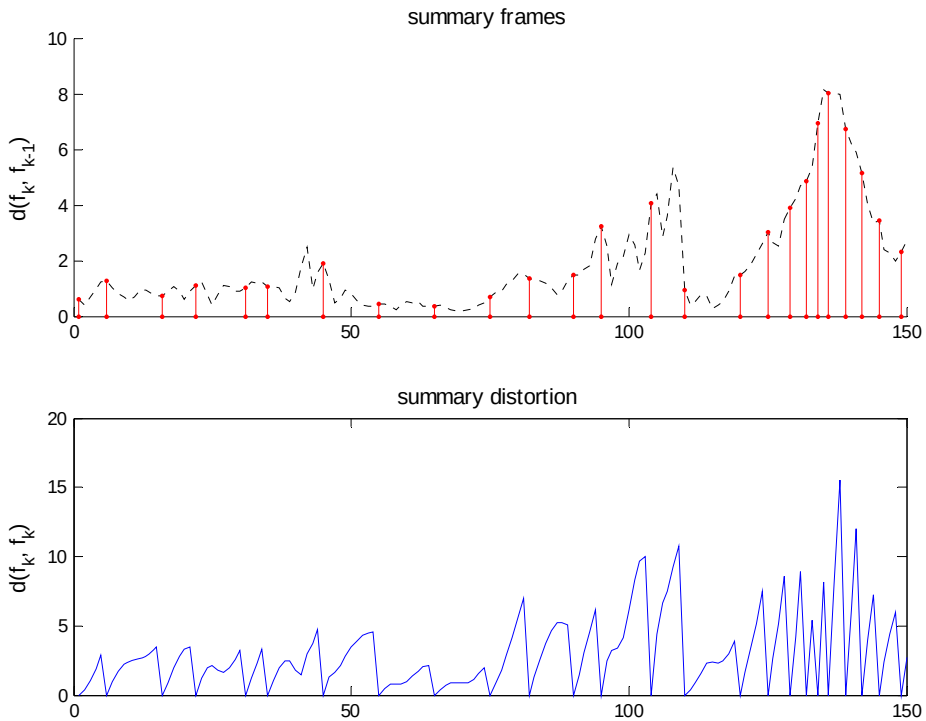
Simulation results :

MDOS: “foreman” sequence, frames 150~299,n=150, m=25, K_{max}= no constrain. Result: D(S)=2.68



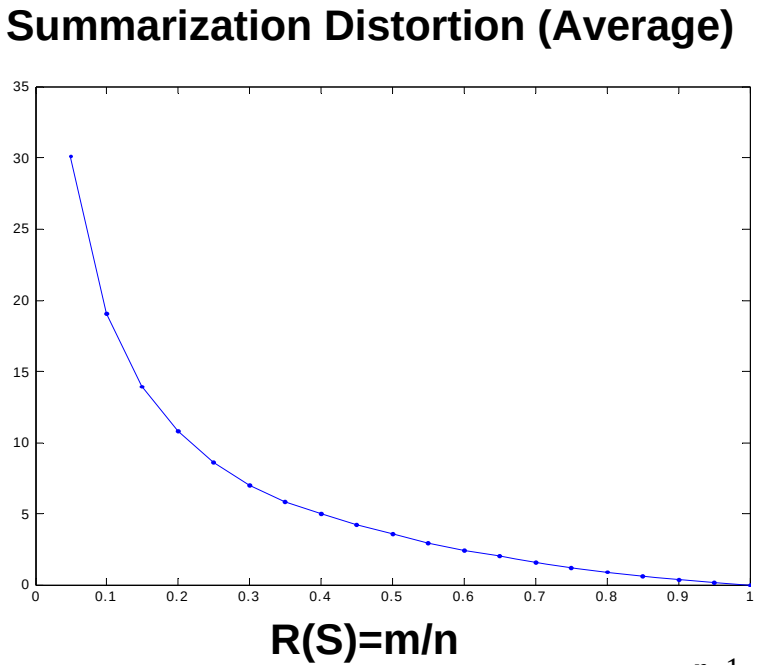
Simulation results : Frame Skip Constrained

MDOS: “foreman” sequence, frames 150~299,n=150, m=25, K_{max}= 10. Result: D(S)=2.90



Solution to the MROS formulation

Bi-Section searching on the operational D-R (ODR) function:



- The ODR is non-increasing
- Bi-section search on the rate, and solve for each rate with MDOS formulation and DP.
- Will converge to the optimal solution R^*.

$$D^*(R) = D^*(m/n) = \min_{l_1, l_2, \cdots, l_{m-1}} (1/n) \sum_{j=0}^{n-1} d(f_j, f_j')$$

Bit constrained MDOS formulation:

- The Lagrangian relaxation:

$$S_{\lambda}^{*} = \arg\min_{S}\{D(S) + \lambda R(S)\}$$

- Solution to the original MDOS formulation:

  Find the λ^{*} such that, $R(S_{\lambda^{*}}^{*}) = R_{\max}$, or closest to the rate constraint R_{max}.

- Solve the relaxed problem by DP, bi-section search on the Lagrangian multiplier for the solution to the original formulation.

Distortion and rate state definitions

- Summary Segment Distortion:

$$G_{l_t}^{l_{t+1}} = \sum_{j=l_t}^{l_{t+1}-1} d(f_{l_t}, f_j)$$

- Distortion State D_t^k, for summary with t frames ending with f_k,

$$D_t^k = \min_{l_1, l_2, \cdots l_{t-2}} \{G_0^{l_1} + G_{l_1}^{l_2} + \cdots + G_{l_{t-2}}^{k} + G_k^n\}$$

- Rate for D_t^k,

$$R_t^k = \sum_{j=0}^{t-1} b(f_{l_j})$$

- The relaxed problem,

$$J_\lambda^{t,k} = \min_{l_1, l_2, \cdots l_{t-2}} \{D_t^k + \lambda R_t^k\}$$

Recursion for the relaxed problem

$$
\begin{aligned}
J_{\lambda}^{t+1,k} &= \min_{l_1,l_2,\cdots l_{t-1}} \{D_{t+1}^{k} + \lambda R_{t+1}^{k}\} \\
&= \min_{l_1,l_2,\cdots l_{t-1}} \{G_0^{l_1} + \cdots + G_{l_{t-1}}^{k} + G_k^{n} + \lambda[b(f_0) + b(f_1) \\
&\qquad + \cdots b(f_{l_{t-1}}) + b(f_k)]\} \\
&= \min_{l_1,l_2,\cdots l_{t-1}} \{G_0^{l_1} + \cdots + G_{l_{t-2}}^{l_{t-1}} + G_{l_{t-1}}^{n} - G_{l_{t-1}}^{n} + G_{l_{t-1}}^{k} + G_k^{n} \\
&\qquad + \lambda[b(f_0) + b(f_1) + \cdots b(f_{l_{t-1}})] + \lambda b(f_k)\} \\
&= \min_{l_1,l_2,\cdots l_{t-1}} \{G_0^{l_1} + \cdots + G_{l_{t-1}}^{n} + \lambda[b(f_0) + \cdots b(f_{l_{t-1}})] \\
&\qquad - [\underbrace{G_{l_{t-1}}^{n} - (G_{l_{t-1}}^{k} + G_k^{n})}_{e^{l_{t-1},k}}] + \lambda b(f_k)\} \\
&= \left\{ \begin{array}{ll} \min\limits_{l_{t-1}} \{J_{\lambda}^{t,l_{t-1}} - e^{l_{t-1},k} + \lambda r^{k}\}, & \text{if intra coding} \\ \min\limits_{l_{t-1}} \{J_{\lambda}^{t,l_{t-1}} - e^{l_{t-1},k} + \lambda r_{l_{t-1}}^{k}\}, & \text{if inter coding} \end{array} \right\}
\end{aligned}
$$

Edge cost for the relaxed problem

$$J_{\lambda}^{t+1,k} = \min_{l_{t-1}} \{ J_{\lambda}^{t,l_{t-1}} - e^{l_{t-1},k} + \lambda b(f_k) \}$$

$$= \begin{cases} \min_{l_{t-1}} \{ J_{\lambda}^{t,l_{t-1}} - e^{l_{t-1},k} + \lambda r^k \}, & \text{if intra coding} \\ \min_{l_{t-1}} \{ J_{\lambda}^{t,l_{t-1}} - e^{l_{t-1},k} + \lambda r_{l_{t-1}}^k \}, & \text{if inter coding} \end{cases}$$

Distortion edge cost $e^{j,k}$ is the same as before,
Additional edge cost in rate:

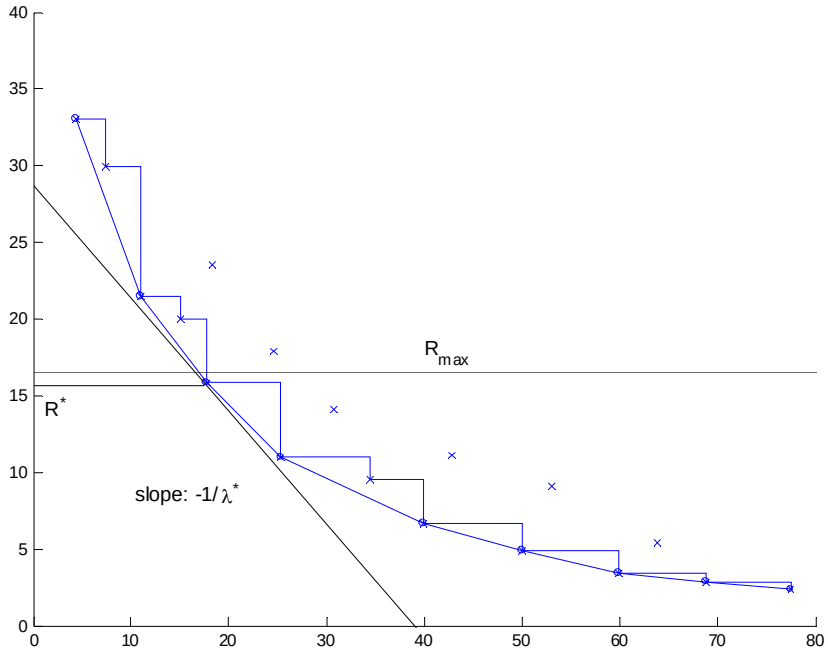
$$\lambda b(f_k) = \begin{cases} \lambda r^k, & \text{if intra coding } f_k \\ \lambda r_{l_{t-1}}^k, & \text{if inter coding } f_k \text{ with MC on } f_{l_{t-1}} \end{cases}$$

Assuming the rates are given by rate profiler [Z.He, CSVT '02, *p*-domain modeling] to achieve constant PSNR quality.

Solution to the original problem

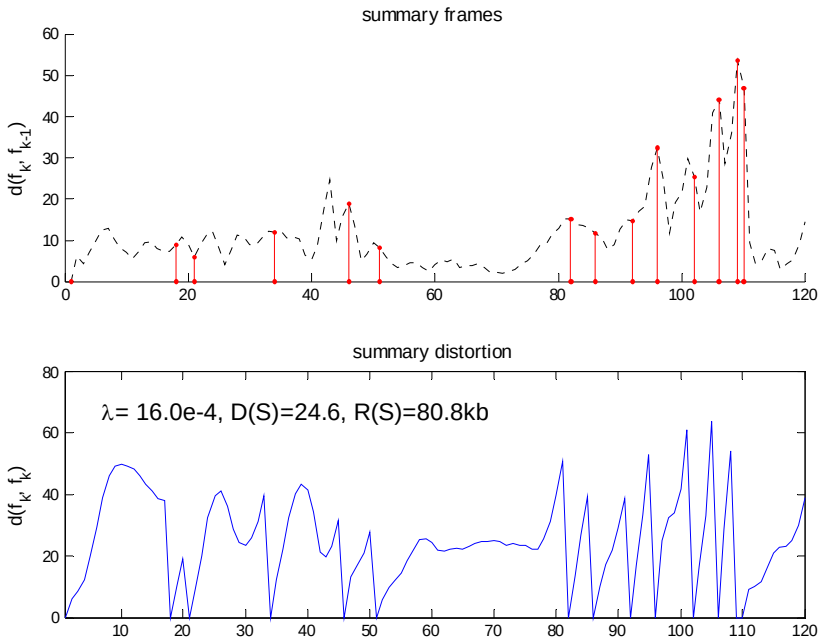
DP Trellis for a given λ :

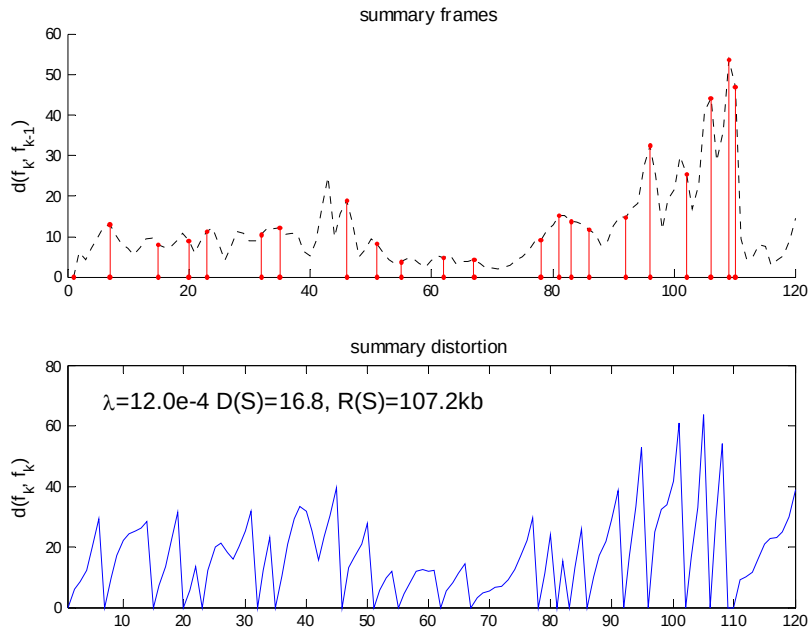
- Solves the relaxed problem with DP, which is a convex hull solution.
- Solve the original by searching on λ for the tightest bound on the rate constraint R_{max}.
- The MROS formulation can be solved similarly



Simulation results :

MDOS: “foreman” sequence, frames 150~279,n=120

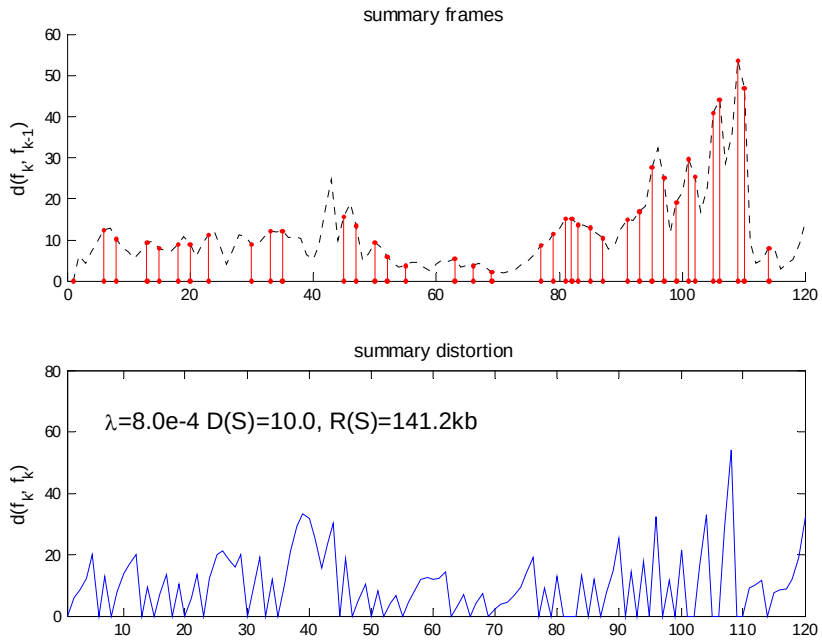




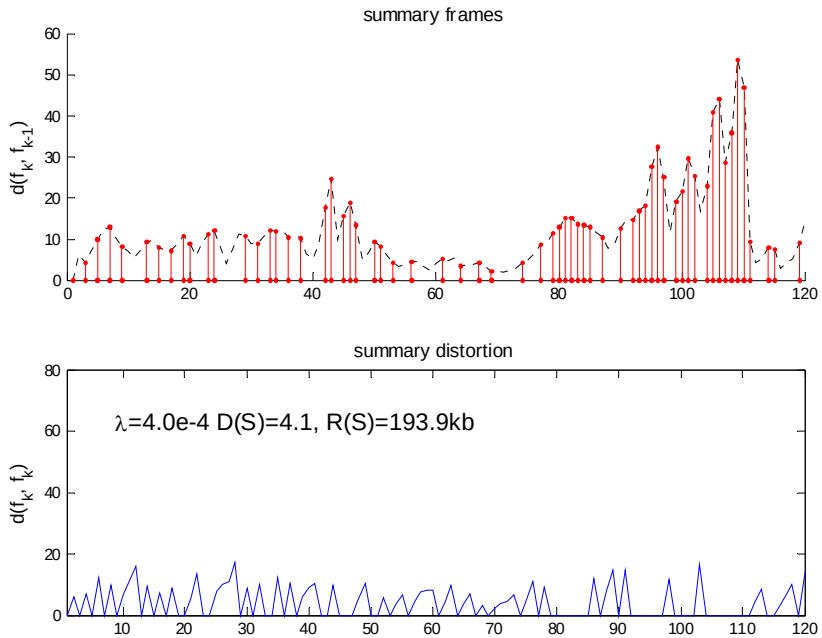
(1) *L*=1.6e-5, D(S)=24.6, R(S)=80.8kb (2) *L*=1.2e-5, D(S)=16.8, R(S)=107.2kb

Simulation results :

MDOS: “foreman” sequence, frames 150~279,n=120



(3) *L*=0.8e-5, D(S)=10.0, R(S)=141.2kb



(4) *L*=0.4e-5, D(S)=4.1, R(S)=193.9kb