Διδακτορική Διατριβή

Τεχνικές Μηχανικής Μάθησης για Διαχείριση Γνώσης σε Πολυμεσικά Δεδομένα

Βασίλειος Χασάνης

Τμήμα Πληροφορικής, Πανεπιστήμιο Ιωαννίνων

Ιούνιος 2009

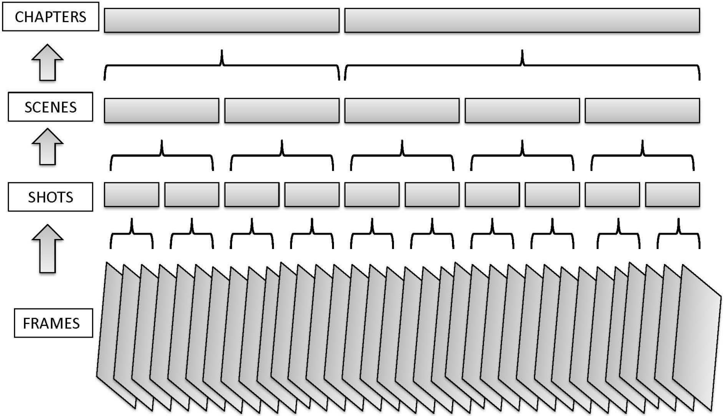
Outline

1. Shot Transitions Detection and Classification Using Support Vector Machines
2. Key-frame Extraction Using Spectral Clustering
3. Segmentation of Videos into Scenes Using Spectral Clustering and Sequence Alignment
4. High-level Movie Segmentation
5. Rushes Summarization
6. Event Detection and Classification in Video Surveillance Sequences

Video

Video: A sequence of frames

A three-dimensional signal, in which two dimensions reveal the visual content in the horizontal and vertical frame direction, and the third one reveals the variations of the visual content over the time axes.



Shot Transitions Detection and Classification Using Support Vector Machines

Shot

The smallest physical segment of a video is the **shot** and is defined as an unbroken sequence of frames recorded from one camera.

Shot Transitions

- Hard cut
- Gradual Transitions
  - Dissolve
  - Fade in, Fade out
  - Other effects

- V. Chasanis, A. Likas, and N. Galatsanos. "Simultaneous detection of abrupt cuts and dissolves in videos using support vector machines". **Pattern Recognition Letters**, 30(1):55-65, 2009.

Visual Examples

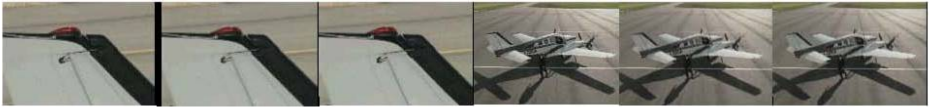


Figure: Hard cut

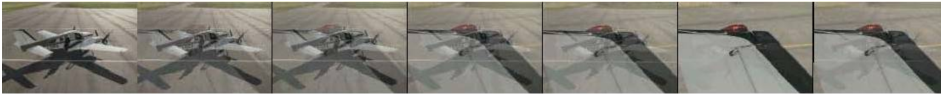


Figure: Dissolve

Challenges,Problems and Contribution

Challenges

- Frame representation.
- Frames dissimilarity.
- Classification using dissimilarity vectors.

Problems

- Different visual content.
- Simultaneous detection of all types of shot transitions.

Contribution

- Features that describe the variation between adjacent frames and the contextual information at the same time.
- Threshold independent method.
- A single classifier for detection and classification of all types of shot transitions.

Contribution

Features selection

- The common practice of identifying the transitions between shots is to first calculate the discontinuity (distance) values of adjacent frames.
- Gradual transitions spread over several frames, thus the differences between adjacent frames are not sufficient to characterize them.
- The differences between non-adjacent frames is also employed in the definition of the proposed features.

Classification

- Support Vector Machines
- A single classifier that classifies transitions into:
  - Hard cuts
  - Gradual Transitions
  - No transitions

Color Histogram and Dissimilarity Value

Color Histogram

- Normalized RGB histograms with 256 bins for each one of the RGB component defined as H_R, H_G and H_B.
- Concatenation of three histograms $H = [H_R H_G H_B]$.

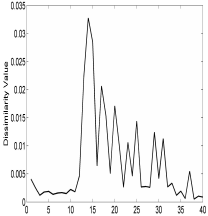
Dissimilarity Value

$$d(I_i, I_{i+l}) = \frac{1}{3} \sum_{k=1}^{768} \frac{(H_i(k) - H_{i+l}(k))^2}{H_i(k) + H_{i+l}(k)}$$

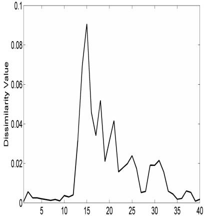
Inter-frame Distance

Dissimilarity value is computed between adjacent frames and frames with time distance l (**inter-frame distance**).

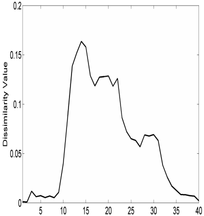
- $l = 1$. To identify hard cuts between two consecutive frames.
- $l = 2$. During a gradual transition two consecutive frames may be the same or very similar to each other, the dissimilarity value will tend to zero.
- $l = 6$. A gradual transition stretches along several frames, while the difference value between consecutive frames is smaller.



$l = 1$



$l = 2$



$l = 6$

Definition of feature vectors

- For each value of l, the **dissimilarity values** form three vectors:

$$
\begin{aligned}
D^{l=1} &= [d(I_1, I_2), \ldots, d(I_i, I_{i+1}), \ldots, d(I_{N-1}, I_N)], \\
D^{l=2} &= [d(I_1, I_3), \ldots, d(I_i, I_{i+2}), \ldots, d(I_{N-2}, I_N)], \\
D^{l=6} &= [d(I_1, I_6), \ldots, d(I_i, I_{i+6}), \ldots, d(I_{N-6}, I_N)].
\end{aligned}
$$

- For each frame i we define a **window** of length w that is centered at this frame and contains the dissimilarity values:

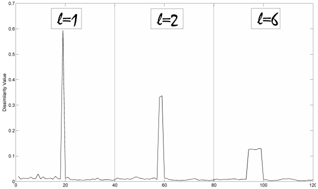
$$
\begin{aligned}
W^{l=1}(i, 1:w) &= [D^{l=1}(i - w/2), \ldots, D^{l=1}(i), \ldots, D^{l=1}(i + w/2 - 1)] \\
W^{l=2}(i, 1:w) &= [D^{l=2}(i - w/2), \ldots, D^{l=2}(i), \ldots, D^{l=2}(i + w/2 - 1)] \\
W^{l=6}(i, 1:w) &= [D^{l=6}(i - w/2), \ldots, D^{l=6}(i), \ldots, D^{l=6}(i + w/2 - 1)]
\end{aligned}
$$

- Normalize dissimilarity values

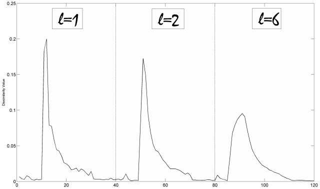
$$
\tilde{W}^{l=k}(i, j) = \frac{W^{l=k}(i, j)}{\sum_{j=1}^{w} W^{l=k}(i, j)}, \quad k = 1, 2, 6.
$$

- Feature vector of frame F_i: $\tilde{W}_{F_i} = [\tilde{W}^{l=1}\, \tilde{W}^{l=2}\, \tilde{W}^{l=6}]$

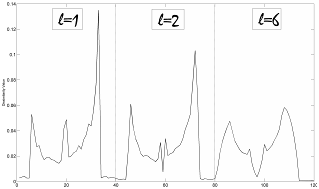
Feature vector examples



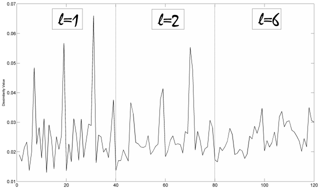
Hardcut



Dissolve



Dissolve



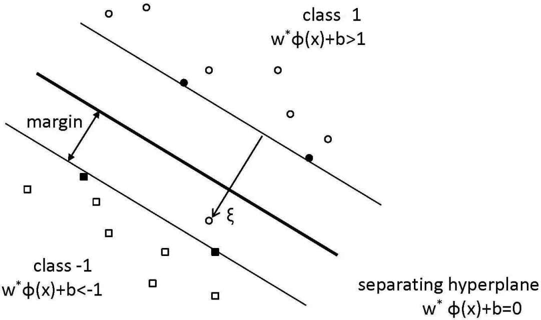
Normal

Classification of Feature Vectors

- **Classify** feature vectors using a machine learning approach.
- Build a training set of annotated examples of shot transitions.
- Employ **Support Vector Machine Classifier** .

Support Vector Machine Classifier

- Suppose we are given a training set of m vectors $x_i \in \mathbb{R}^n$, i=1, . . . , m and a vector $y \in \mathbb{R}^m$ with $y_i \in \{$1,-1$\}$ denoting the class of vector x_i.
- The classical SVM classifier:
  - Non-linear **mapping** (function $\phi(x)$) of input data to a feature space of higher dimension.
  - Finds an **hyperplane** ($w^T\phi(x) + b =$ 0) which separates data points of two classes.



Support Vector Machine Classifier

$$\begin{aligned} \min_{w,b,\xi} \quad & \tfrac{1}{2} w^T w + C \textstyle\sum_{i=1}^{m} \xi_i \\ \text{subject to} \quad & y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i \\ & \xi_i \geq 0, \ i = 1, \dots, m \end{aligned}$$

- C is the penalty parameter for **misclassified** data introduced by error variables ξ_i.
- The decision function is:

$$sign(\sum_{i=1}^{m} w_i K(x_i, x) + b), \text{ where } K(x_i, x_j) = \phi^T(x_i)\phi(x_j) \ .$$

Support Vector Machine Classifier

Support Vectors

Training patterns x_i that contribute to the decision function and have $w_i \neq 0$.

RBF Kernel

$$K(x_i, x_j) = exp(-\gamma \|x_i - x_j\|^2)$$

"One-against-one" approach

- For a k-class problem, $k(k-1)/2$ **binary** classifiers are constructed and each one is trained to discriminate data from two classes.
- The decision of each binary classifier is considered as a **vote** for its proposed class.
- The class with the **maximum** number of votes is selected.

Performance Criteria

$$\text{Recall} = \frac{N_c}{N_c + N_m}, \text{Precision} = \frac{N_c}{N_c + N_f}, F_1 = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}},$$

- N_c stands for the number of **correct** detected shot boundaries.
- N_m stands for the number of **missed** ones.
- N_f stands for the number of **false** detections.

- Final performance measure:

$$F_1 = \frac{\alpha}{\alpha + b} F_{1C} + \frac{b}{\alpha + b} F_{1D}$$

  - α is the number of true hard cuts
  - b the number of true dissolves

Video Data for Shot Detection Problem

- The video sequences were taken from TV-series, documentaries and educational films.
- Nine videos (70000 frames), manually annotated by a human observer.
- Sample uniformly negative examples due to **imbalanced** training set.
- 9 “rounds” of testing (8 videos are used for training and the 9-th for testing).

Video ID	Frames	Cuts	Dissolves	Genre
T1	6318	36	23	Comedy
T2	9466	28	16	Action
T3	11807	4	6	Drama
T4	1535	14	8	Educational
T5	17982	146	7	Action
T6	1665	1	19	Comedy
T7	14993	105	11	Drama
T8	9840	12	41	Documentary
T9	6355	9	11	Documentary
Total	69334	355	142	-

Table: Characteristics of videos used for the shot detection problem.

Results

Transition type	N_c	N_m	N_f	Recall (%)	Precision (%)	F_1 (%)
Cuts	351	4	9	98.87	97.50	98.18
Dissolves	127	15	33	89.44	79.38	84.11
Average	-	-	-	96.18	92.32	94.21

Table: $w = 40$, $l = 1$, $l = 2$ and $l = 6$.

Transition type	N_c	N_m	N_f	Recall (%)	Precision (%)	F_1 (%)
Cuts	352	3	8	99.15	97.78	98.46
Dissolves	130	12	25	91.55	83.87	87.54
Average	-	-	-	96.98	93.80	95.37

Table: $w = 50$, $l = 1$, $l = 2$ and $l = 6$.

Transition type	N_c	N_m	N_f	Recall (%)	Precision (%)	F_1 (%)
Cuts	353	2	4	99.44	98.88	99.16
Dissolves	127	15	25	89.44	83.55	86.39
Average	-	-	-	96.58	94.50	95.53

Table: $w = 60$, $l = 1$, $l = 2$ and $l = 6$.

Comparison

METHOD	CUTS		
	Recall (%)	Precision(%)	F_1(%)
w = 40, l=1, l=2 and l=6.	98.87	97.50	98.18
w = 40, l=2 and l=6.	98.87	97.50	98.18
w = 40, l=1, l=2 and l=6 (HSV, x^2).	99.44	98.89	99.16
w = 40, l=1, l=2 and l=6 (HSV, KL).	99.15	98.60	98.92
Pair-wise comparison	85.07	84.83	84.95
Likelihood ratio	94.37	86.12	90.05
Twin-comparison	89.30	88.05	88.92
Wavelets	97.18	91.57	94.29

Table: Comparative results using Recall, Precision and F_1 measures for cuts detection.

Comparison

METHOD	DISSOLVES		
	Recall (%)	Precision(%)	F_1(%)
$w = 40$, $l=1$, $l=2$ and $l=6$.	89.44	79.38	84.11
$w = 40$, $l=2$ and $l=6$.	88.73	80.77	84.56
$w = 40$, $l=1$, $l=2$ and $l=6$ (HSV, x^2).	88.03	81.17	84.46
$w = 40$, $l=1$, $l=2$ and $l=6$ (HSV, KL).	85.92	79.74	82.73
Pair-wise comparison	-	-	-
Likelihood ratio	-	-	-
Twin-comparison	70.42	64.94	67.57
Wavelets	74.64	81.53	77.93

Table: Comparative results using Recall, Precision and F_1 measures for dissolves detection.

Key-frame Extraction Using Spectral Clustering

- Key-frame extraction is an important task in video processing and analysis that is used to create a **video summary**.
- Indexing and retrieval algorithms are usually applied on **key-frames**.

Key-frames

Most representative frames of a shot describing the whole content.

- V. Chasanis, A. Likas, and N. Galatsanos. "Efficient video shot summarization using an enhanced spectral clustering approach." In **Proceedings of the 18th International Conference on Artificial Neural Networks, Part I**, pp. 847-856, Prague, Czech Republic, September 2008.

Requirements and Approaches

Requirements

- Must represent the whole video content without missing important information.
- Must not be similar, in terms of video content information, thus containing redundant information.

Typical Approaches

- Detect abrupt changes in the similarity between successive frames.
- Perform clustering of shot frames into groups and select a representative frame of each group as key-frame.

Contribution

- Spectral analysis of frame similarity matrix.
- Estimate the number of key-frames from the eigenvalues of the similarity matrix.
- Employ global k-means (instead of k-means).

Spectral Clustering

Given a set of frames $F = \{F_1, \dots, F_N\}$ to be partitioned into M groups.

- Compute similarity matrix $A = [a(i,j)]$, with $a(i,j) = \text{sim}(F_i, F_j)$.
- Eigenvalue computation of a matrix $\Phi(A)$ (normalized cut).
- Construct the eigenvector matrix $U = [u_1, \dots, u_M]$ (top eigenvectors).
  - each frame F_k is represented by an M-dimensional real vector y_k corresponding to the k-th row of U.
- Cluster the rows of U into M groups using k-means.

Spectral Clustering - Analysis

- $Z = [Z_1, Z_2, \ldots, Z_M]$:**partition matrix** representing a clustering solution.
    - Column vector Z_j is the binary indicator vector for group j:

$$
\begin{aligned}
Z(i,j) = 1 &\ : \ if \ \ i \in G_j \\
Z(i,j) = 0 &\ : \ otherwise \\
Z^T Z = I_M &
\end{aligned}
$$

- **Clustering objective** (depending on Φ)

$$
\max_{Z} \ trace(Z^T \Phi Z),
$$
$$
s.t. \ Z^T Z = I_M \text{ and } Z(i,j) \in \{0,1\}.
$$

Spectral Clustering - Analysis

- The **spectral approach** (for M clusters) provides solution to the following continuous optimization problem (**relaxation**):

$$\max_{Y} \; trace(Y^T \Phi Y), \\ s.t. \; Y^T Y = I_M.$$

- Optimal solution: $Y^* = U_M = [u_1, \ldots, u_M]$
  - u_i are the eigenvectors corresponding to the ordered top M eigenvalues λ_i of Φ.
- Use k-means to obtain Z^* from Y^* (discretization).

Number of Key-frames

- The **optimal value** of the objective function for M clusters is:

$$sol(M) = \max_{Y^T Y = I_M} trace(Y^T \Phi Y) = \lambda_1 + \lambda_2 + \ldots + \lambda_M$$

- The improvement from adding cluster $M+1$ is:

$$sol(M+1) - sol(M) = \lambda_{M+1}.$$

- When λ_{M+1} is lower than a threshold, the improvement is **negligible** and the number of clusters is assumed to be M.
- The proposed approach:
  - Compute and sort eigenvalues: $\lambda_1 \geq \lambda_2 \geq \ldots \lambda_N$
  - Determine the largest eigenvalue $\lambda_{M+1} < T$ (T=0.005 in all experiments).
  - Select M as the number of clusters.

Proposed Method

- Frame similarity matrix A based on normalized HSV **color histograms**.
- Matrix Φ: $\Phi = I - D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ (normalized cut).
- **Eigenanalysis** of Φ: estimation of number of key-frames M.
- Define the matrix $U = [u_1, \dots, u_M]$ and cluster its rows into M groups:
  - **global k-means** [Likas et al., PR 2003] is used (instead of typical k-means) to overcome the initialization problem of k-means.
- The **medoid** frame of each group is selected as key-frame.

Evaluation metrics

- The **evaluation** of extracted key-frames is a difficult issue.
- Which frames are the best representatives is rather **subjective**.
- Objective **quality measures** have been proposed:
  - Average Fidelity (AF)
  - Shot Reconstruction Degree (SRD)
- Given:
  - The frame sequence $F = \{F_1, F_2, \ldots, F_N\}$.
  - The set of key-frames $KF = \{KF_1, KF_2, \ldots, KF_M\}$

Average Fidelity and Shot Reconstruction Degree

Average Fidelity

- Average similarity of frames to closest key-frames.

Shot Reconstruction Degree

- Reconstruct each frame from neighboring KFs in the sequence, using a frame **interpolation algorithm** IA.

$$\tilde{F}_n = \mathsf{IA}(KFn_j, KFn_{j+1}),\ n_j \leq n < n_{j+1}$$

- Compute the similarity between the original and reconstructed sequence.
- The Shot Reconstruction Degree (SRD):

$$SRD(F, KF) = \sum_{n=0}^{N-1} Sim(F_n, \tilde{F}_n)$$

Dataset

Frame Sequence	No. Frames	Genre
F_1	633	Comedy
F_2	144	Basketball
F_3	145	Basketball
F_4	146	Basketball
F_5	225	Football
F_6	300	Football
F_7	172	Football

Table: Key-Frame extraction dataset characteristics.

Comparison

Algorithms

- k-means on the histogram vectors
  - 20 restarts keeping the one with minimum clustering error.
- Typical spectral clustering algorithm (with 20 k-means restarts)
- Adaptive key-frame extraction method (AKF) [Rasheed et al., TMM 2005]:
  - The middle frame is selected as the first key-frame.
  - Next, each frame in the sequence is compared with the current set (KF) of key-frames.
  - If it differs more than a threshold value from every key-frame in the current set KF, then it is added to KF as a new key-frame.

Number of key-frames

Assumed to be the same as selected using the proposed estimation algorithm.

Results

ASF	Algorithm			
Frame Seq.	Our method	K-means	AKF	Spectral
F_1	**0.973**	0.9549	0.9616	0.9619
F_2	**0.9437**	0.9278	0.8913	0.9235
F_3	**0.9506**	0.9344	0.9268	0.9253
F_4	**0.9557**	0.948	0.9405	0.9462
F_5	**0.9673**	0.9467	0.9550	0.9625
F_6	**0.9558**	0.931	0.9424	0.9318
F_7	**0.9782**	0.9654	0.9672	0.9675

Table: Comparative results of the tested key-frame extraction algorithms using Average Shot Fidelity measure on dataset A.

Results

SRD	Algorithm			
Frame Seq.	Our method	K-means	AKF	Spectral
F_1	**1859.66**	1533.34	1693.1	1620.6
F_2	**424.72**	369.87	292.43	362.64
F_3	**502.76**	430.78	374.23	431.32
F_4	**528.09**	356.46	340.89	393.02
F_5	**843.10**	808.2	758.23	780.33
F_6	**855.44**	753.75	813.1	791.2
F_7	**707.92**	648.71	642.97	663.15

Table: Comparative results of the tested key-frame extraction algorithms using SRD measure on dataset A.

Representation (solution for M=1,..,5)

FRAME 398-ITERATION 1

FRAME 398-ITERATION 2

FRAME 853-ITERATION 2

FRAME 254-ITERATION 3

FRAME 398-ITERATION 3

FRAME 805-ITERATION 3

FRAME 254-ITERATION 4

FRAME 494-ITERATION 4

FRAME 716-ITERATION 4

FRAME 805-ITERATION 4

FRAME 273-ITERATION 5

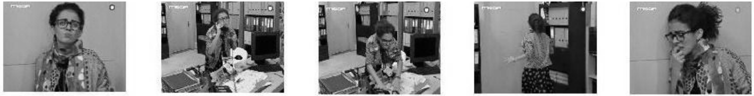
FRAME 494-ITERATION 5

FRAME 716-ITERATION 5

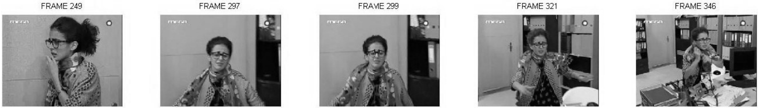
FRAME 805-ITERATION 5

FRAME 864-ITERATION 5

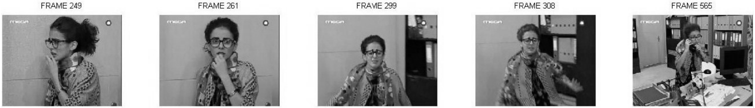
Representation



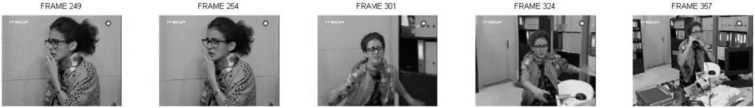
Our method



K-means

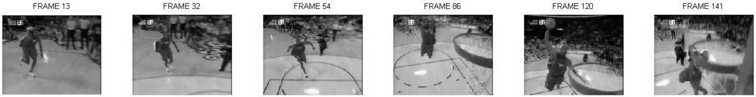


AKF

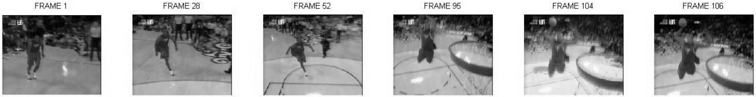


Spectral

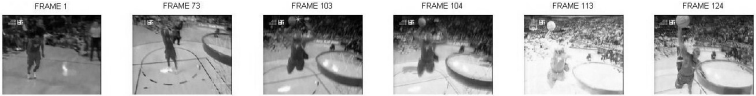
Representation



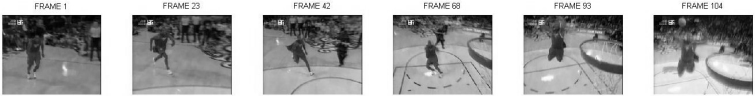
Our method



K-means



AKF



Spectral

Segmentation of Videos into Scenes Using Spectral Clustering and Sequence Alignment

Scene

A group of **successive** shots that take place in a fixed physical setting (e.g. a dialogue detection in a room) or a group of successive shots that describe an action or event (e.g. a car chase by police cars).

- V. Chasanis, A. Likas, and N. Galatsanos. “Scene detection in videos using shot clustering and sequence alignment”. **IEEE Transactions on Multimedia**, 11(1):89-100, January 2009.
- V. Chasanis, A. Likas, and N. Galatsanos. “Scene detection in videos using shot clustering and symbolic sequence segmentation”. In **Proceedings of IEEE 9th Workshop on Multimedia Signal Processing**, pp. 187-190, Chania, Greece, October 2007.

Scene Detection Algorithm

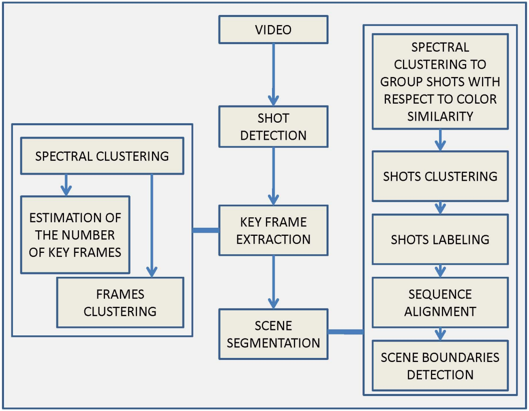


Figure: The main steps of our scene segmentation method.

Contribution

Contribution

- Shots are clustered into groups using **only visual similarity**, while time adjacency is treated in a distinct processing phase.
- Shots are **labeled** according to the cluster they are assigned.
- Sequence alignment to detect changes in the **pattern** of sequences of shot labels that correspond to scene boundaries.

Shot Similarity

- Existing approaches consider the temporal distance of shots as an extra feature that is taken into account when computing the similarity between two shots for shot clustering into scenes.
- In our approach, shots are clustered into groups using **only visual similarity**.
- Time adjacency is treated in a distinct processing phase.
- **Visual similarity** between a pair of shots i and j:

$$VisSim(i,j) = \max_{p \in K_i, q \in K_j} ColSim(p,q),$$

where K_i and K_j are the sets of key-frames of shots i and j respectively.

$$ColSim(i,j) = \sum_{h \in bins} \min(H_i(h), H_j(h)),$$

where H_i, H_j are the HSV normalized color histograms of frames f_i and f_j respectively.

Shots Clustering

Suppose there is a set $V = \{v_1, v_2, \ldots, v_N\}$ of N shots, ordered in time, to be segmented.

- A $N \times N$ similarity matrix A is computed.

$$a(i,j) = VisSim(v_i, v_j), \ \ v_i, v_j \in V$$

- Spectral clustering algorithm employing fast-global k-means is used to group shots into clusters.
- Number of shot clusters is **not equal** to the number of scenes in the video, but to the principal color distributions of the video.

Symbolic Shot Sequence

- Spectral clustering algorithm has provided a partition of the shots into K clusters $\{C_1, C_2, \ldots, C_K\}$.
- A label is assigned to each shot according to the cluster it belongs.
- A symbolic sequence of labels is produced.

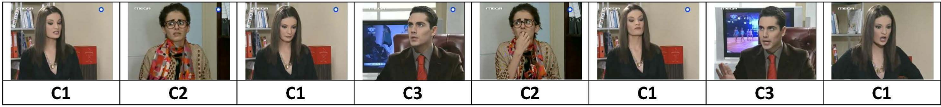
V_{01}	V_{02}	V_{03}	V_{04}	V_{05}		V_{06}	V_{07}	V_{08}	V_{09}		V_{10}	V_{11}	V_{12}	V_{13}	V_{14}	V_{15}	V_{16}		V_{17}	V_{18}	V_{19}	V_{20}	V_{21}
C_1	C_1	C_1	C_1	C_1	$\downarrow$	C_2	C_2	C_2	C_2	$\downarrow$	C_3	C_5	C_3	C_5	C_3	C_5	C_3	$\downarrow$	C_4	C_4	C_2	C_4	C_4

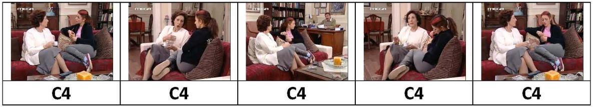
Figure: Video sequence of labels.

Patterns of shot labels

Patterns of repetitive shots:

- A dialogue between two or more persons.
    - Camera switches from one person to another.
    - Sequence of shots $C_1\,C_2\,C_1\,C_3\,C_2\,C_1\,C_3\,C_1$.
- Different captions of the same setting.
    - Cameras recording from different angles.
    - Sequence of shots $C_4^1\,C_4^2\,C_4^1\,C_4^3\,C_4^3\,C_4^2$.





Sequences Comparison

- A scene changes when a change in such patterns occurs.
- Comparison of **successive non-overlapping** windows of shot labels.

$$\overbrace{\begin{array}{ccccccccccccccccccccc} \mathrm{V}_{01} & \mathrm{V}_{02} & \mathrm{V}_{03} & \mathrm{V}_{04} & \mathrm{V}_{05} & \mathrm{V}_{06} & \mathrm{V}_{07} & \mathrm{V}_{08} & \mathrm{V}_{09} & \mathrm{V}_{10} & \mathrm{V}_{11} & \mathrm{V}_{12} & \mathrm{V}_{13} & \mathrm{V}_{14} & \mathrm{V}_{15} & \mathrm{V}_{16} & \mathrm{V}_{17} & \mathrm{V}_{18} & \mathrm{V}_{19} & \mathrm{V}_{20} & \mathrm{V}_{21} \\ \mathrm{C}_1 & \mathrm{C}_1 & \mathrm{C}_1 & \mathrm{C}_1 & \mathrm{C}_1 & \mathrm{C}_2 & \mathrm{C}_2 & \mathrm{C}_2 & \mathrm{C}_2 & \mathrm{C}_3 & \mathrm{C}_5 & \mathrm{C}_3 & \mathrm{C}_5 & \mathrm{C}_3 & \mathrm{C}_5 & \mathrm{C}_3 & \mathrm{C}_4 & \mathrm{C}_4 & \mathrm{C}_2 & \mathrm{C}_4 & \mathrm{C}_4 \end{array}}^{OriginalSequence}$$

$$\text{Iteration } 1 \longrightarrow \underbrace{(C_1\,C_1\,C_1\,C_1)}_{X_1^1}\underbrace{(C_1\,C_2\,C_2\,C_2)}_{X_2^1}$$

$$\text{Iteration } 2 \longrightarrow \underbrace{(C_1\,C_1\,C_1\,C_1)}_{X_1^2}\underbrace{(C_2\,C_2\,C_2\,C_2)}_{X_2^2}$$

$$\text{Iteration } 3 \longrightarrow \underbrace{(C_1\,C_1\,C_1\,C_2)}_{X_1^3}\underbrace{(C_2\,C_2\,C_2\,C_3)}_{X_2^3}$$

Figure: Sub-sequences to be compared.

Global Sequence Alignment Algorithm

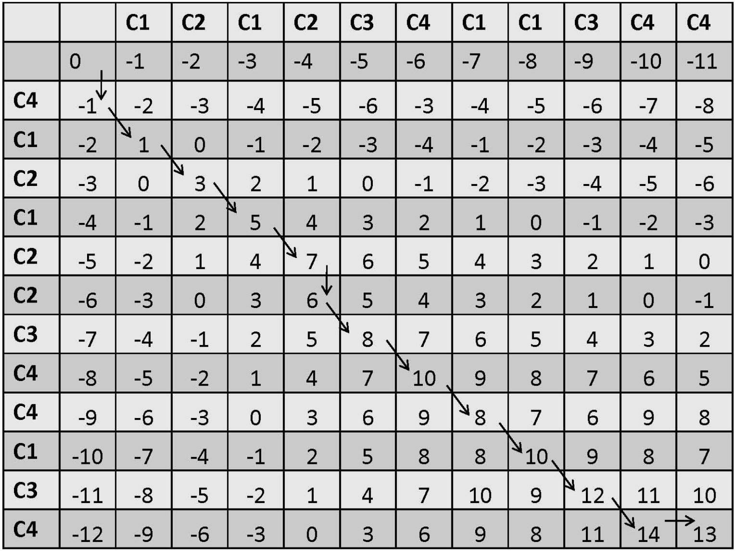
“Needleman-Wunsch” algorithm

- An example of dynamic programming commonly used in bioinformatics to align protein or nucleotide sequences.
- Performs global alignment on two sequences and is guaranteed to find the alignment with the maximum score.

- Given two sequences of length w: $X_1 = L_1 L_2 \dots L_w$ and $X_2 = M_1 M_2 \dots M_w$.
- The labels $L_i, M_i,\ i = 1, \dots, w$ belong to some alphabet of K symbols.
- To align these sequences, a $w \times w$ matrix N is constructed where :

$$N(i,j) = \begin{cases} N(i-1,j-1) + S(X_1(i), X_2(j)) \\ N(i-1,j) - d \\ N(i,j-1) - d \end{cases}$$

Global Sequence Alignment Algorithm



		C1	C2	C1	C2	C3	C4	C1	C1	C3	C4	C4
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10	-11
C4	-1	-2	-3	-4	-5	-6	-3	-4	-5	-6	-7	-8
C1	-2	1	0	-1	-2	-3	-4	-1	-2	-3	-4	-5
C2	-3	0	3	2	1	0	-1	-2	-3	-4	-5	-6
C1	-4	-1	2	5	4	3	2	1	0	-1	-2	-3
C2	-5	-2	1	4	7	6	5	4	3	2	1	0
C2	-6	-3	0	3	6	5	4	3	2	1	0	-1
C3	-7	-4	-1	2	5	8	7	6	5	4	3	2
C4	-8	-5	-2	1	4	7	10	9	8	7	6	5
C4	-9	-6	-3	0	3	6	9	8	7	6	9	8
C1	-10	-7	-4	-1	2	5	8	8	10	9	8	7
C3	-11	-8	-5	-2	1	4	7	10	9	12	11	10
C4	-12	-9	-6	-3	0	3	6	9	8	11	14	13

Global Sequence Alignment Algorithm

- The **traceback** from $N(w,w)$ to $N(0,0)$ defines the optimal alignment of X_1 and X_2.
- The output of the alignment algorithm is an alignment matrix containing:
  - Matches (M)
  - Mismatches (m)
  - Gaps (G)

$$
\begin{array}{lcl}
Seq_1 & : & C_1\,C_2\,C_1\,C_2\,C_3\,C_4\,C_1\,C_1\,C_3\,C_4\,C_4 \\
Seq_2 & : & C_4\,C_1\,C_2\,C_1\,C_2\,C_2\,C_3\,C_4\,C_4\,C_1\,C_3\,C_4
\end{array}
$$

Output : (Alignment matrix)

Seq_1	_	C_1	C_2	C_1	C_2	_	C_3	C_4	C_1	C_1	C_3	C_4	C_4
Seq_2	C_4	C_1	C_2	C_1	C_2	C_2	C_3	C_4	C_4	C_1	C_3	C_4	_
Type	G	M	M	M	M	G	M	M	m	M	M	M	G

Substitution Matrix S

Substitution Matrix S

$S(i,j)$ expresses how similar are shot labels C_i and C_j in terms of color and position.

- Color similarity between shot labels can be defined from the similarity of their respective clusters medoids.

$$CSM(i,j) = VisSim(m_i, m_j),\ \ m_i, m_j \in Med$$

- Position similarity is expressed through the possibility that a shot label i precedes or follows a shot label j.

$$PPM(i,j) = \frac{1}{N-1}\{\#\ \text{pairs}(L_1 = C_i, L_2 = C_j)\}$$

-

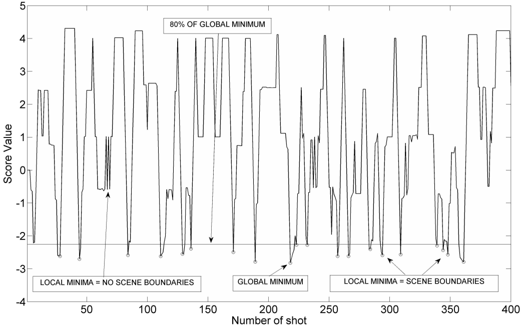
$$S(i,j) = \begin{cases} CSM(i,j) + PPM(i,j) & , i = j \\ -\alpha(1 - CSM(i,j)) - (1-\alpha)(1 - PPM(i,j)) & , i \neq j \end{cases}$$

Scoring Sequence

Scoring Function

F=(score of matches)-(score of mismatches)-(score of gaps).

- Find global minimum.
- Scene boundaries are the local minima that are less than 80% of the global minimum value.



Dataset

Video	Duration(min)	Shots	Scenes	Genre
V_1	22	404	15	comedy
V_2	31	591	18	comedy
V_3	30	587	16	comedy
V_4	23	437	13	comedy
V_5	27	633	14	drama
V_6	26	454	17	drama
V_7	32	377	15	comedy
V_8	45	608	25	drama
V_9	31	714	25	action
V_{10}	26	246	19	action

Table: Dataset B characteristics.

Results

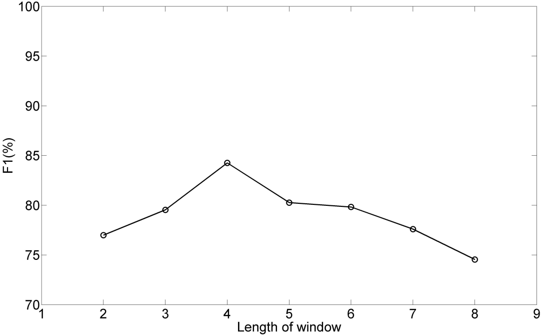


Figure: Average performance results for different values of the window parameter w.

Results

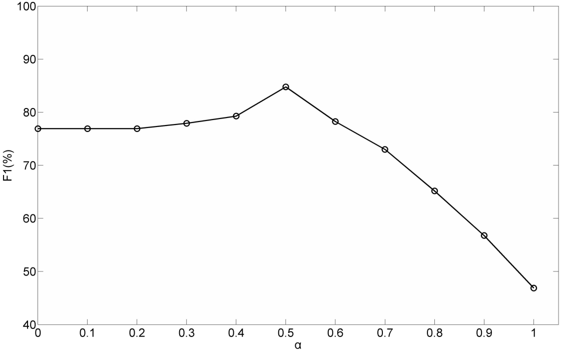


Figure: Average performance results for different values of the a parameter and w $= 4$, $Th = 0.8$.

Comparison

- Spectral clustering on a color and motion based similarity matrix weighted by a decreasing function of the temporal distance between shots. [Rasheed et al., TMM 2005]
- Construction of a scene transition graph based on visual characteristics and temporal dynamics. [Yeung et al., CVIU 1998]

Comparative results of the tested scene detection algorithms

		C2009	R2005	Y1998
V_1	R(%)	86.67	86.67	60.00
	P(%)	92.85	61.90	81.82
	F_1(%)	**89.70**	72.22	69.23
V_2	R(%)	100.00	83.33	72.22
	P(%)	90.00	62.50	68.42
	F_1(%)	**94.74**	71.43	70.27
V_3	R(%)	87.50	81.25	87.50
	P(%)	73.68	52.00	70.00
	F_1(%)	**80.00**	63.41	77.78
V_4	R(%)	76.92	92.31	76.92
	P(%)	83.33	60.00	71.43
	F_1(%)	**80.00**	72.73	74.07
V_5	R(%)	85.71	92.86	78.57
	P(%)	92.31	63.16	64.71
	F_1(%)	**88.89**	75.18	70.97
V_6	R(%)	82.35	76.47	70.59
	P(%)	93.33	61.90	66.67
	F_1(%)	**87.50**	68.42	68.57
V_7	R(%)	86.67	86.67	80.00
	P(%)	81.25	61.90	75.00
	F_1(%)	**83.87**	68.42	77.42
V_8	R(%)	76.00	80.77	71.43
	P(%)	95.00	70.00	74.07
	F_1(%)	**84.44**	75.00	72.73
V_9	R(%)	72.00	80.77	64.00
	P(%)	75.00	55.26	59.26
	F_1(%)	**73.47**	65.63	61.54
V_{10}	R(%)	70.00	75.00	68.42
	P(%)	93.33	75.00	72.22
	F_1(%)	**80.00**	75.00	70.27

Results

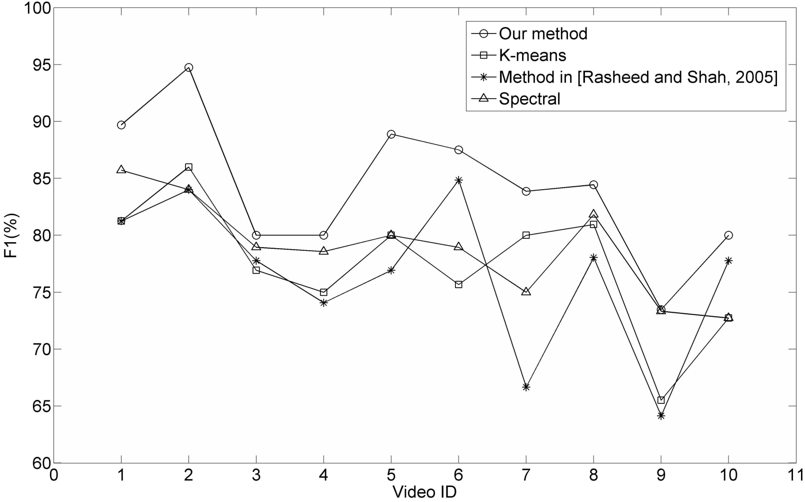


Figure: Scene detection results (using F_1 measure) comparing four key-frame extraction algorithms.

High-level Movie Segmentation

Chapter

A more compact representation/segmentation of a video is the merging of successive scenes into chapters (logical story units).

- V. Chasanis, A. Kalogeratos, and A. Likas. “Movie segmentation into scenes and chapters using locally weighted bag of visual words”. **In Proceedings of ACM International Conference on Image and Video Retrieval**, Santorini, Greece, July 2009.

Challenges and Contribution

Challenges

- **Constant change** in the color distribution of the shots due to different shots taken at different places.
- Color histograms are inefficient to describe scenes with constant changing content.
- **Distinctive points** are repeated in consecutive shots during the progress of such an event.

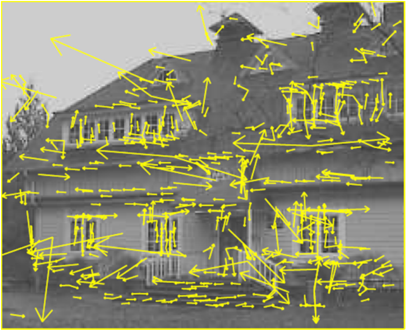
Contribution

- Employ **Locally invariant descriptors** to provide sufficient description of interest points and their possible transformations.
  - **SIFT** (Scale-Invariant Feature Transforms) descriptors
  - **CCH** (Contrast Context Histogram) descriptors
- Shot representation with **visual word histograms**.
- Temporally histogram **smoothing** (using a gaussian kernel) with respect to **neighboring** histograms.

Scale-Invariant Feature Transforms

Scale-invariant feature transforms [Lowe, IJCV 2004]

- Find **keypoints** in image.
- Compute scale-invariant coordinates relative to a **neighborhood** of each keypoint.
- 128-dimensional feature vector serves as the descriptor for each keypoint.

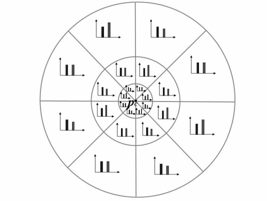


Contrast Context Histogram Descriptors

- Contrast context histogram [Huang et al., PR 2008] represents the contrast distributions of a local region around an **interest point** and serves as a local descriptor for this region.
- The contrast of a point p in this area is given from the following equation:

$$C(p) = I(p) - I(p_c).$$

- For each **salient** point p_c a region R is defined in a quantized log-polar coordinate system.



Contrast Context Histogram Descriptors

- For each sub-region $R_{ij} = (r_i \theta_j)$, a **positive** and a **negative** bin of the contrast values are computed.

$$H_{R_{ij}^{+}}(p_c) = \frac{\sum\{C(p)|p \in R_{ij} \;\; and \;\; C(p) \geq 0\}}{\#R_{ij}^{+}},$$

$$H_{R_{ij}^{-}}(p_c) = \frac{\sum\{C(p)|p \in R_{ij} \;\; and \;\; C(p) < 0\}}{\#R_{ij}^{-}},$$

- $\#R_{ij}^{+}$ and $\#R_{ij}^{-}$ define the number of positive and negative positive contrast values in R_{ij}
- CCH descriptor is a 64-dimensional features vector.

$$CCH\, p_c = (H_{R_{00}^{+}}, H_{R_{00}^{-}}, \ldots, H_{R_{rl}^{+}}, H_{R_{rl}^{-}}).$$

Bag of Visual Words

- Given a shot s_t and its corresponding set of n key-frames $KF = \{kf_1, \ldots, kf_n\}$.
- For each key-frame kf_i, $i = 1, \ldots, n$, a set of descriptors D_{kf_i} is extracted (SIFT or CCH).
- The shot is described by the concatenation of all descriptors:

$$D_{s_t} = D_{kf_1} \bigcup \ldots \bigcup D_{kf_n}.$$

- The descriptors for all N video shots $D_S = D_{s_1} \bigcup D_{s_2} \bigcup \ldots \bigcup D_{s_N}$ are clustered into k groups (**visual words**) $\{C_1, C_2, \ldots, C_k\}$ using the k-means algorithm.
- Each element of the set of descriptors D_{s_t} is assigned to one of the k visual words (clusters).
- Given shot s_t with D descriptors $d_{t_1}, \ldots, d_{t_D}$, the **visual word histogram** VH_t for this shot is defined as:

$$VH_t(l) = \frac{\#\{d_{t_j} \in C_l,\ j = 1, \ldots, D\}}{D},\ \ l = 1, \ldots, k.$$

Similarities Between Video and Text Documents

Text

Words, paragraphs and logical story units (book chapters).

Movie

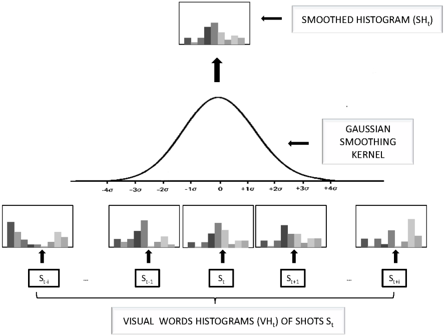
Shots, scenes and logical story units (DVD chapters).

- Locally Weighted Bag of Words (Lowbow) [Lebanon et al., JMLR 2007] has been proposed for text document representation and segmentation.
- The main idea is to represent a text document by describing several locations in its word sequence using histograms.
- Each histogram is **smoothed** with the neighbor histograms using a gaussian kernel.

Smoothing Process

- The visual word histogram of a shot is **temporally** smoothed with respect to the histograms of neighboring shots.
- A normalized discretized gaussian kernel K_σ with zero mean and standard deviation σ is used.

$$SH_t = \sum_{n=-\infty}^{\infty} VH_n \cdot K_\sigma(t-n),$$


SMOOTHED HISTOGRAM (SH_t)
GAUSSIAN SMOOTHING KERNEL
-4σ
-3σ
-2σ
-1σ
0
+1σ
+2σ
+3σ
+4σ
S_{t-i}
...
S_{t-1}
S_t
S_{t+1}
...
S_{t+i}
VISUAL WORDS HISTOGRAMS (VH_t) OF SHOTS S_t

Segmentation

- By adjusting the value of σ, we can preserve **contextual** information at different time scales:
    - A low value of σ results in small scale smoothing (scene detection).
    - A higher value of σ results in large scale smoothing (chapter detection).
- The **boundaries** between different video segments separate video parts containing different visual words distributions.
- Boundaries are the **local maxima** of the distance between successive smoothed histograms.

Scene Detection

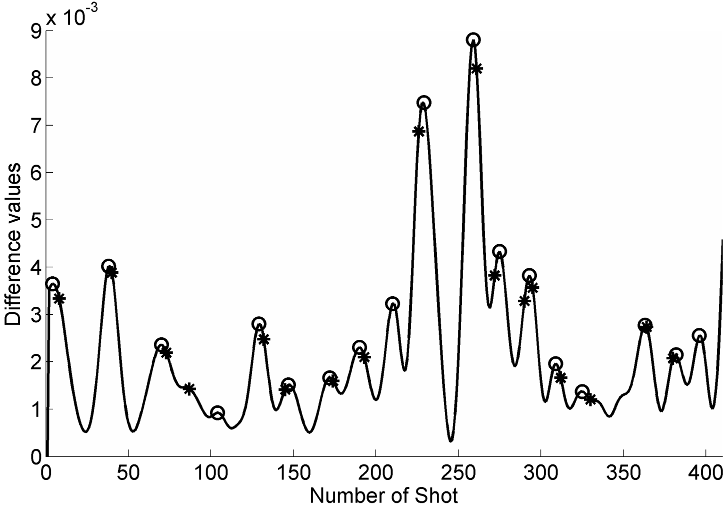


Figure: Difference values of the smoothed histograms using $\sigma = 8$ (scene detection).

Chapter Detection

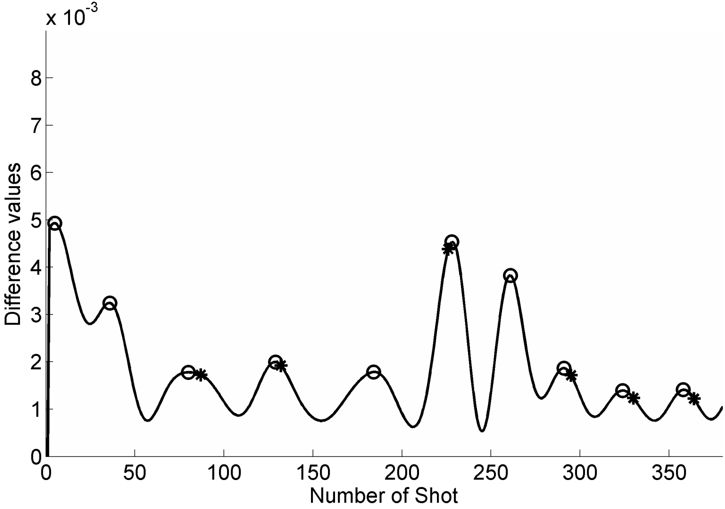


Figure: Difference values of the smoothed histograms using $\sigma = 16$ (chapter detection).

Dataset

Movie	ID	Duration(min)	Shots	Scenes	Chapters	Genre
A Beautiful Mind	M1	36	421	18	7	Biography | Drama
Sex and the City	M2	70	1217	45	19	Comedy | Romance
Gone in 60 seconds	M3	80	1788	74	23	Action | Crime | Thriller
Goldeneye	M4	74	1218	46	20	Action | Adventure
Top Gun	M5	74	1113	48	16	Action | Romance

Table: Movies characteristics.

Performance w.r.t σ

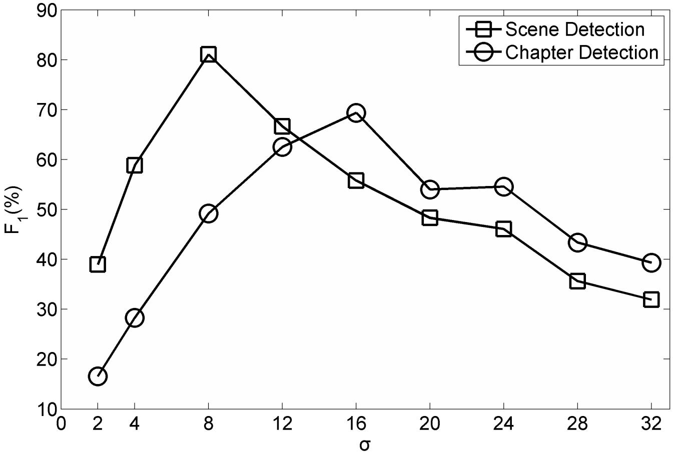


Figure: Average performance results (on all movies) for different values of the smoothing parameter σ for the scene and chapter detection problems, using SIFT descriptors and a vocabulary of 500 visual words.

Results

		SIFT	CCH	SEQAL	NCUT	GRAPH	HSV
M1	R(%)	88.89	83.33	77.78	83.33	77.78	72.22
	P(%)	88.89	88.24	82.35	60.00	60.87	72.22
	F_1(%)	**88.89**	85.71	80.00	69.77	68.29	72.22
M2	R(%)	91.11	80.00	80.00	71.71	64.44	73.33
	P(%)	83.67	73.47	67.92	46.24	56.86	63.46
	F_1(%)	**87.23**	76.60	73.47	56.22	60.42	68.04
M3	R(%)	82.43	77.03	79.73	74.32	62.16	71.62
	P(%)	72.62	69.51	68.60	55.56	54.12	67.09
	F_1(%)	**77.22**	73.08	73.75	63.58	57.86	69.28
M4	R(%)	88.89	77.78	88.89	80.00	68.89	68.89
	P(%)	68.97	63.94	64.52	53.73	48.44	58.49
	F_1(%)	**77.67**	70.00	74.77	64.29	56.88	63.27
M5	R(%)	75.00	70.83	75.00	72.92	70.83	52.08
	P(%)	73.47	72.34	70.59	53.85	45.33	58.14
	F_1(%)	**74.23**	71.58	72.73	61.95	55.28	54.95

Table: Comparative results using Recall(R), Precision(P) and F_1 measures for the scene detection problem for movies M_1-M_5, (SEQAL - Chasanis et al.,2009 - C2009, NCUT - Rasheed and Shah - R2005, GRAPH - Yeng et al., 1998).

Results

		SIFT	CCH	SEQAL	NCUT	GRAPH	HSV
M1	R(%)	100.00	85.71	42.86	71.43	71.43	71.43
	P(%)	63.64	60.00	33.33	41.67	31.25	50.00
	F_1(%)	**77.78**	70.59	37.50	52.63	43.49	58.82
M2	R(%)	89.47	84.21	68.42	57.89	63.18	73.68
	P(%)	68.00	59.26	41.94	45.83	37.50	50.00
	F_1(%)	**77.27**	69.57	52.00	41.16	47.06	59.57
M3	R(%)	78.26	82.61	65.22	60.87	60.87	65.22
	P(%)	45.00	43.18	30.62	40.00	40.00	30.61
	F_1(%)	**57.14**	56.72	41.67	48.28	48.28	41.67
M4	R(%)	80.00	75.00	95.00	40.00	45.00	60.00
	P(%)	61.54	53.57	30.65	42.10	33.33	42.86
	F_1(%)	**69.57**	62.50	46.34	41.02	38.29	50.00
M5	R(%)	87.50	75.00	68.75	56.25	81.25	56.25
	P(%)	51.85	50.00	32.33	36.00	26.00	39.13
	F_1(%)	**65.12**	60.00	44.00	43.90	39.39	46.15

Table: Comparative results using Recall(R), Precision(P) and F_1 measures for the chapter detection problem for movies M_1-M_5, (SEQAL - Chasanis et al.,2009 - C2009, NCUT - Rasheed and Shah - R2005, GRAPH - Yeng et al., 1998).

Rushes Summarization

Video summary

A **condensed** version of the initial video where judgements about the video content can be made in less time and effort than using the initial video.

Video rushes

Unedited video footage containing many **redundant** information and **useless** frames.

- V. Chasanis, A. Likas, and N. Galatsanos. "Video rushes summarization using spectral clustering and sequence alignment". **In Proceedings of the 2nd ACM TRECVid Video Summarization Workshop**, Vancouver, Canada, October 2008.

Challenges and Contribution

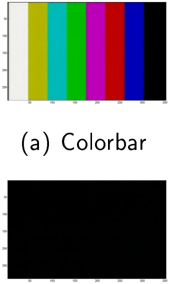
Challenges

- Useless frames detection and removal.
- Removal of redundant information (repetitive shots).

Contribution

- Edge direction histograms and SIFT descriptors to detect useless frames.
- Local sequence alignment to detect similar shots.

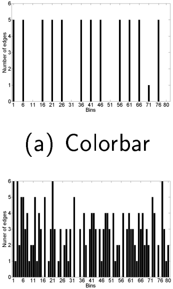
Useless Frames Detection



(a) Colorbar

(b) Monochrome

Figure: Useless frames.



(a) Colorbar

(b) Typical frame

Figure: Edge direction histograms.

Edge Direction Histogram

Captures the spatial distribution of edges.

- Edges are grouped into five categories: vertical, horizontal, 45 diagonal, 135 diagonal, and isotropic (nonorientation specific).

Visual Shot Similarity Metric

- Rushes often contain **repetitive** information, since the same scene is usually taken many times until the desired result is produced.
- Our goal is to detect similar segments which in our case are shots and keep only one representative for each group of similar shots.
- Two shots that describe the **same scene** are considered **similar** and their key frames will follow the same order.
- Either a segment of one shot or the whole shot will also appear in the other shot.

Visual Shot Similarity Metric

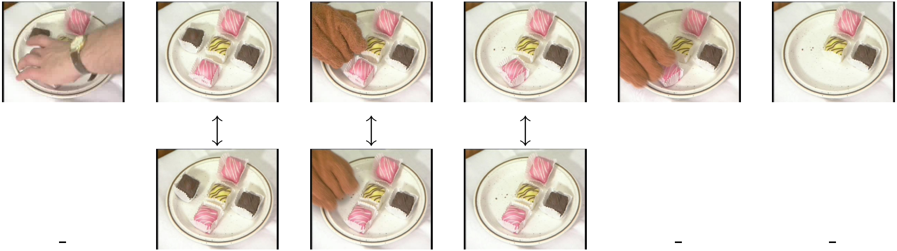
- Employ a **local** sequence alignment algorithm between the sets of their key-frames.
- A key-frame is "matched" with the most similar (visually) key-frame of the other set of key-frames, while also taking into consideration the temporal order of key-frames.
- The **score** of the sequence alignment constitutes the final shot similarity metric.

Visual Shot Similarity Metric

- To align two sequences we use the “Smith-Waterman” algorithm.
- Given two shots S_i and S_j and $KF_i = \{KF_i^1,\ KF_i^2, \ldots,\ KF_i^m\}$, $KF_j = \{KF_j^1,\ KF_j^2, \ldots,\ KF_j^n\}$ their corresponding key-frame sets.

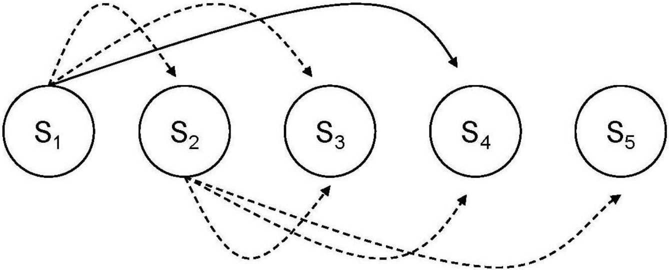
$$SM(m,n) = VisSim(KF_i^m, KF_j^n)$$

- The **substitution matrix** of the sequence alignment is given by similarity matrix SM.



Repetitive Shot Detection

- To find groups of repetitive and similar shots we compared each shot with the next three.
- If one of the three shots is similar with the shot under consideration then all the shots between these two shots and the shots under comparison, form a group. If none of the shots is similar then a new group of shots is considered and the algorithm continues until all shots are examined.



Clapboard Removal

- Rushes also contain **clapboards** to indicate the current number of the shot.
- To detect clapboards we compute for each key-frame the scale-invariant feature transforms (**SIFT**).
- In order to detect whether a key-frame contains a clapboard, we compute its SIFT descriptors and compare them with the SIFT descriptors of the database (TRECVID 2007 Development Data).

(a) Clapboard (b) Sift descriptors

Figure: Clapboard and its sift descriptors.

Summarization

- The final stage of our summarization method involves the production of the final **video summary**.
- The summary of a video can be a set of key-frames or a video of a smaller duration than the original video.
- Once the repetitive shots have been detected, the shot with the largest duration is selected as their representative.
- Select a number of frames around each key-frame to generate the final video summary.

Experiments

- We have tested our method on TRECVID 2008 Test Data (40 videos) under the Rushes Summarization competition of NIST TRECVID 2008.
- The goal of this task is to produce video summaries with duration less than or equal to $p = 2\%$ of the duration of the original video.

Evaluation metrics

Three humans at Dublin City University have judged each summary. The quality of each summary was evaluated directly by subjective and objective measures.

- Subjective measures
  1. The fraction of inclusions found in the summary (IN) ranging from 0 to 1.
  2. Lack of junk (colorbars, clapboars and monochrome frames) (JU). The lack of junk score was an integer ranging from 1 (worst) to 5 (best).
  3. Whether the summary had a pleasant tempo/rythm (TE). Score ranges from 1 (worst) to 5 (best).
  4. Whether the summary contained lots of duplicate video (RE). Score ranges from 1 (worst) to 5 (best).

Results and comparison

	Our method		All	
	Mean	Median	Avg.(Mean)	Avg.(Median)
IN (0-1)	0.53	0.56	0.44	0.44
JU (1-5)	3.31	3.33	3.17	3.21
TE (1-5)	2.50	2.33	2.76	2.75
RE (1-5)	3.16	3.33	3.3	3.36

Table: Performance of our video rushes summarization method.

Event Detection and Classification in Video Surveillance Sequences

Event

Time interval where a person performs an activity.

- Individual activities performed in an indoor environment using a standing camera.

Challenges and Contribution

Challenges

- Video sequence segmentation into segments/events.
- Classification in predefined categories.

Contribution

- Background substraction and video segmentation using SIFT descriptors.
- Event representation using a set of visual word histograms.
- To define event dissimilarity we employ Dynamic Time Warping distance.
- Supervised methods to classify events in predefined categories.

Background Substraction

- Compute and store set D_b of SIFT descriptors of frames describing the background.
- For each frame j compute set D_j SIFT descriptors.
- Each descriptor of set D_j is compared with the descriptors of set D_b to find a “match”.

Background Substraction

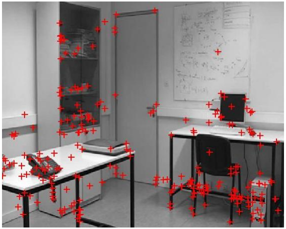


Figure: Video frame of the background and the location of the extracted descriptors.

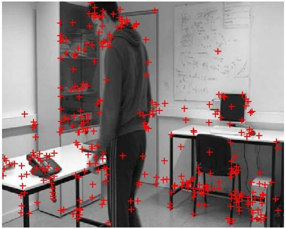
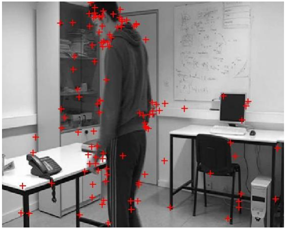


Figure: Video frame with its descriptors.



Video Segmentation into Events

- A video event is defined as the time interval where a person performs an activity.
- When someone enters the room under surveillance, new descriptors will appear that do not correspond to background.
- Analyze a vector that corresponds to the number of "foreground" descriptors for each frame.

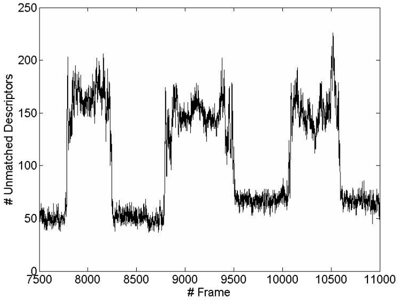
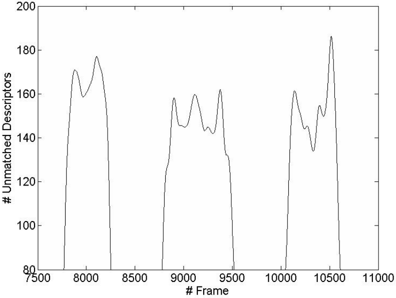


Figure: Number of foreground descriptors of a video surveillance sequence.

Video Segmentation into Events

- In order to detect the beginning and the end of a video event, this vector is smoothed with the discretized gaussian kernel.
- Discard low values of the smoothed signal to remove noise (background descriptors that have not been removed).



Event Representation

- The descriptors of all event frames are clustered into a predefined number of clusters K using the k-means algorithm.
- K denotes the size of the visual words vocabulary.

Event-based representation

A visual word histogram VHE_i of an event i is constructed by assigning each descriptor of all the event frames to one of the K visual words (clusters).

Frame-based representation

A visual word histogram VHF_j^i of a frame j of event i is constructed by assigning the frame's descriptors to one of the K visual words (clusters).

Event Dissimilarity

Event-based dissimilarity

To compute a distance value between two events E_i and E_l we compare their corresponding visual word histograms VHE_i and VHE_l.

Frame-based dissimilarity

- We compare the visual word histograms VHF of their frames.
- Given events $E_i = \{f_1^i, \ldots, f_{N_i}^i\}$ and $E_l = \{f_1^l, \ldots, f_{N_l}^l\}$.
- $N_i \neq N_l$, thus **Dynamic Time Warping** (DTW) distance is employed to compare two events.

Dynamic Time Warping

Dynamic Time Warping

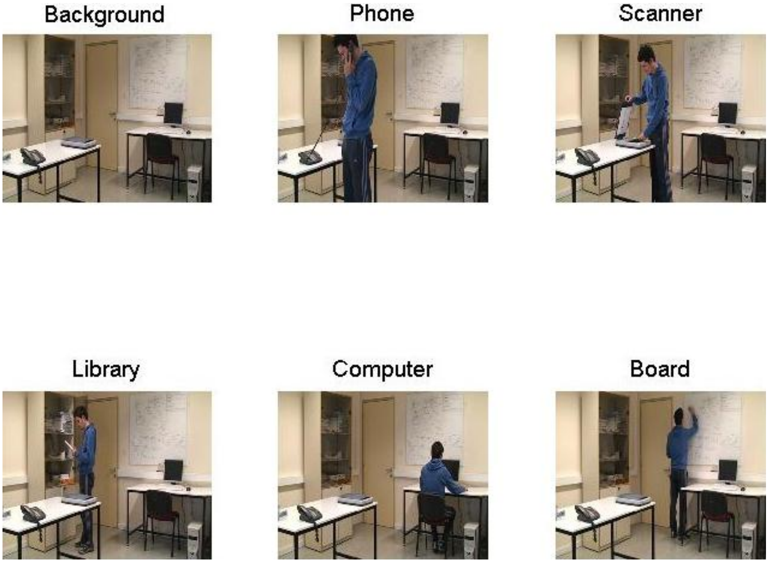
- Similar to global sequence alignment algorithm.
- Uses dynamic programming to compare sequences of continuous data.
- The distance between two video segments/events E_i and E_l is the average DTW distance of their K-multidimensional signals:

$$D(E_i, E_l) = \frac{1}{K} \sum_{m=1}^{K} DTW(VHF^i(m), VHF^l(m))$$

Experimental Results

- **Individual** activities performed in an indoor environment.
- Performed activities are **not overlapping**, in the sense that a person enters the room performs a set of basic actions and leaves the room.
- 20 activities/events are performed that are divided in five categories:
  1. **Phone**:a person enters the room, goes to the phone and makes a conversation.
  2. **Scanner**:a person enters the room, goes to the scanner and scans a document.
  3. **Library**:a person enters the room, goes to the library and opens it.
  4. **Computer**:a person enters the room, goes to the computer and works with it.
  5. **Board**:a person enters the room, goes to the board and writes something.
- The result of the automatic segmentation was **optimal**, since no over-segmentation or under-segmentation was performed and all 20 events were detected as unique.

Video surveillance sequence



Background

Phone

Scanner

Library

Computer

Board

Figure: Sample frames of the background and the five categories of events.

Classification Results

- Nearest neighbor classifier (with 1, 3, and 5 neighbors).
- Support Vector Machine classifier.
  - Radial basis function (RBF) kernel.
  - Cross-validation for parameters **C**, γ.
- The classification accuracy was estimated using the leave-one-out approach.

K	1-NN		3-NN		5-NN		SVM	
	DTW	EV	DTW	EV	DTW	EV	DTW	EV
10	80%	85%	80%	85%	65%	65%	75%	65%
20	90%	90%	95%	90%	90%	80%	95%	95%
50	95%	95%	95%	95%	95%	90%	100%	95%
100	95%	90%	100%	100%	100%	95%	100%	95%

Table: Classification results for the first video sequence.

Conclusions

- Shot Detection
  - Features that describe the variation between adjacent frames and the contextual information at the same time.
  - Example based, threshold independent method.
  - Simultaneous detection of all types of shot transitions.
- Key-frame Extraction
  - Unique key-frames that summarize efficiently the video content.
  - Estimation of number of key-frames.
- Scene Detection
  - Shots are clustered into groups using only visual similarity, while time adjacency is treated in a distinct processing phase.
  - Shots are labeled according to the cluster they are assigned.
  - Sequence alignment to detect changes in the pattern of sequences of shot labels that correspond to scene boundaries.

Conclusions

- High-Level Movie segmentation
  - Shot representation with visual word histograms.
  - Temporally smoothing (gaussian kernel) of visual word histograms.
  - By adjusting the smoothing parameter of the gaussian kernel we can detect both scene and chapter boundaries of each movie.
- Rushes Summarization
  - Edge direction histograms and SIFT descriptors to detect useless frames.
  - Local sequence alignment to detect similar shots.
- Event Detection and Classification
  - Background substraction and video segmentation using SIFT descriptors.
  - Event representation using a set of visual word histograms.
  - Event dissimilarity using Dynamic Time Warping distance.
  - Supervised methods to classify events in predefined categories.

Future work

- Video retrieval using relevance feedback.
- Apply temporally smoothed visual word histograms in shot detection and key-frame extraction problems.
- Build a visual word vocabulary by comparing video shots to existing semantic detectors.
- Test high-level segmentation algorithm in a variety of video genres, i.e sports and tv-news.
- In video surveillance there are several open problems that deserve further investigation.
    - Detection of multiple events in a video surveillance sequences.

Journal Publications

1. V. Chasanis, A. Likas, and N. Galatsanos. “Simultaneous detection of abrupt cuts and dissolves in videos using support vector machines”. **Pattern Recognition Letters**, 30(1):55-65, 2009.
2. V. Chasanis, A. Likas, and N. Galatsanos. “Scene detection in videos using shot clustering and sequence alignment”. **IEEE Transactions on Multimedia**, 11(1):89-100, January 2009.
3. V. Chasanis, A. Kalogeratos, A. Likas, and N. Galatsanos. “High-Level Movie Segmentation Using Temporally Smoothed Histograms of Visual Words”. **IEEE Transactions on Multimedia**, Under 2nd review.

Conference Publications

1. V. Chasanis, A. Likas, and N. Galatsanos. "Scene detection in videos using shot clustering and symbolic sequence segmentation". In **Proceedings of IEEE 9th Workshop on Multimedia Signal Processing**, pp. 187-190, Chania, Greece, October 2007.
2. V. Chasanis, A. Likas, and N. Galatsanos. "A support vector machine approach for video shot detection". In **Proceedings of the 1st International Symposium on Intelligent Interactive Multimedia Systems and Services**, pp. 45-54, Peraias, Greece, July 2008.
3. V. Chasanis, A. Likas, and N. Galatsanos. "Efficient video shot summarization using an enhanced spectral clustering approach." In **Proceedings of the 18th International Conference on Artificial Neural Networks, Part I**, pp. 847-856, Prague, Czech Republic, September 2008.
4. V. Chasanis, A. Likas, and N. Galatsanos. "Video rushes summarization using spectral clustering and sequence alignment". **In Proceedings of the 2nd ACM TRECVid Video Summarization Workshop**, Vancouver, Canada, October 2008.
5. V. Chasanis, A. Kalogeratos, and A. Likas. "Movie segmentation into scenes and chapters using locally weighted bag of visual words". **In Proceedings of ACM International Conference on Image and Video Retrieval**, Santorini, Greece, July 2009.