

PHILLIPS

Visual Feature Extraction for Content Analysis

Vladimir Nedovic
Storage Systems and Applications

30-05-2006

Overview

- Video Content Analysis
- Visual Feature Spaces
- Our Feature Extraction Methods
- Applications in Cassandra Framework (and some results)
- Conclusions and Future Work

Video Content Analysis

- Tools to manage video data are necessary
 - Easier navigation, indexing, archiving, retrieval
- Video sequence ~ textual document ?
 - Could build an index and a table of contents for a video document
 - Temporal segmentation needed
- Four main processes: **Feature Extraction**, **Temporal Structure Analysis**, **Abstraction** and **Indexing**
- Our emphasis: *feature extraction in visual feature space*

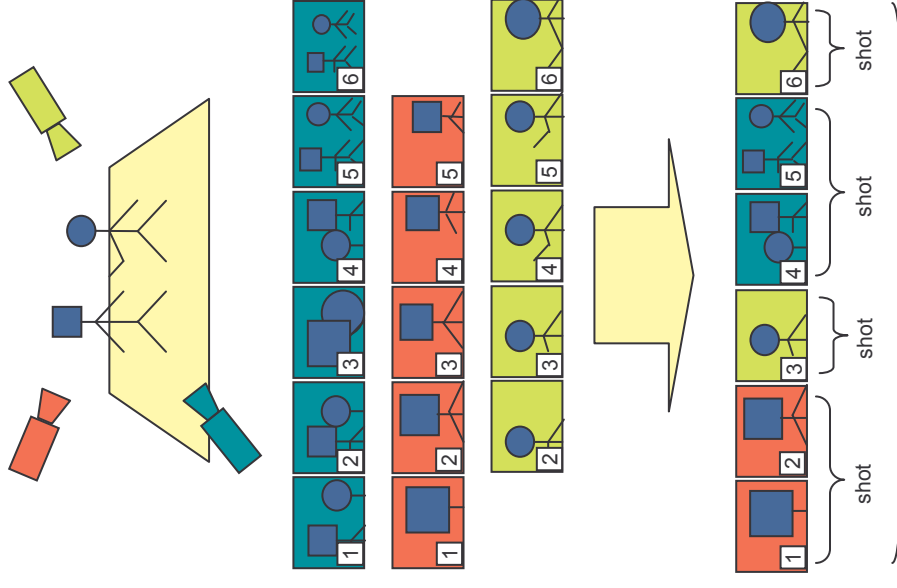
Video Content Analysis (cont.) - Temporal Structure Analysis

- Shots – physical layers in video, single camera take
 - Temporal and physical scene continuity
 - Easy to detect shot transitions – visual discontinuities occur



semantic gap

- Scenes – a collection of shots related *semantically*
 - visual continuity not required anymore



Video Content Analysis (cont.) – Feature Extraction

- Pre-requisite for temporal video segmentation
 - Spatial domain analysis
 - visual features, superimposed text
 - Temporal domain analysis
 - motion, audio, speech
- Features from both domains should be combined for optimal results – machine learning?

Problem Outline

1. Identify and extract salient *visual* features suitable for scene detection
2. Prepare features for input to machine learning algorithms

(Note: input to ML algorithms should contain other features for optimal results)

Motivation

1. Limited visual feature set in CASSANDRA framework (uncompressed domain)
2. New attempt to utilize machine learning on an extensive feature vector
3. Extensions to available framework possible
 - audio and motion features available
 - parallel shot (PS) detector

Visual Feature Spaces

- What constitutes an image (which primitive elements)?
 - *Color* - luminance, colors combined
 - *Spatial Layout* – structure within image
 - *Shapes* – object contours
 - *Texture* – spatial arrangements within objects' surfaces
 - More?
- **Criteria:**
 - Robustness against noise, lots of compression
 - Film-grammar rules



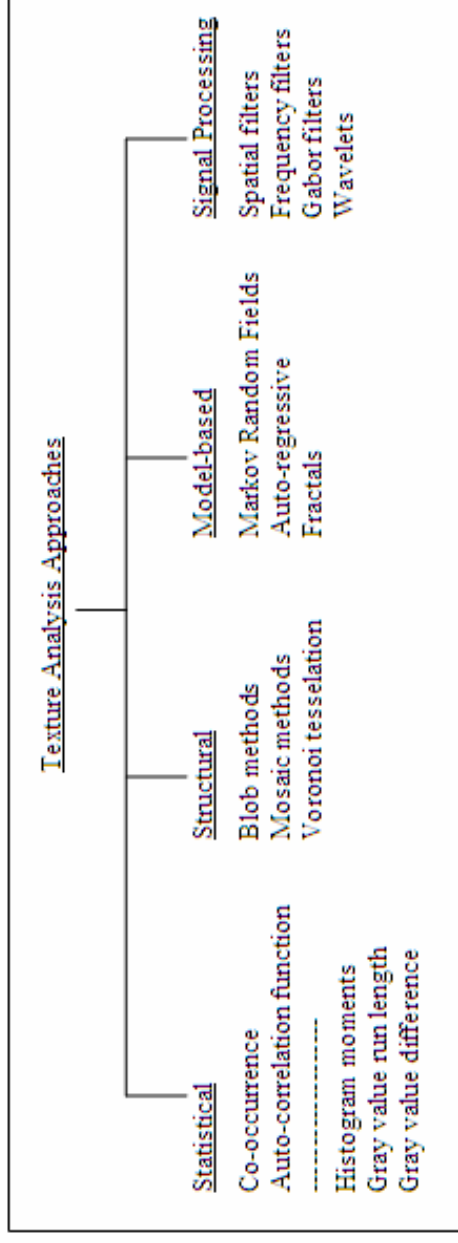
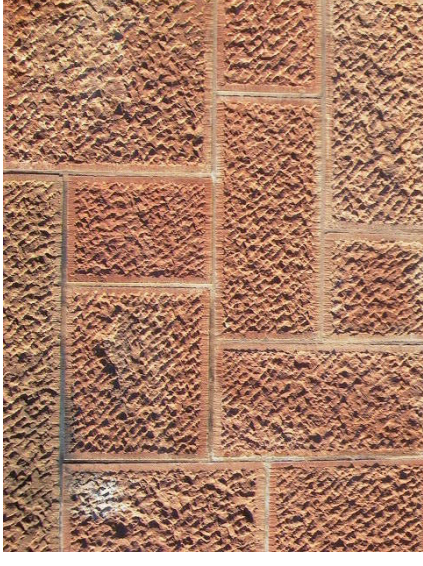
Visual Feature Spaces - Color

- Very expressive
- Depends on the color of the light source
- Multiple Color Spaces
 - perceptual uniformity is an issue
- Many color features exist
 - color histogram, color moments, correlogram, *MPEG-7* features...
- Some features incorporate structure
 - ⇒ we will not consider it separately




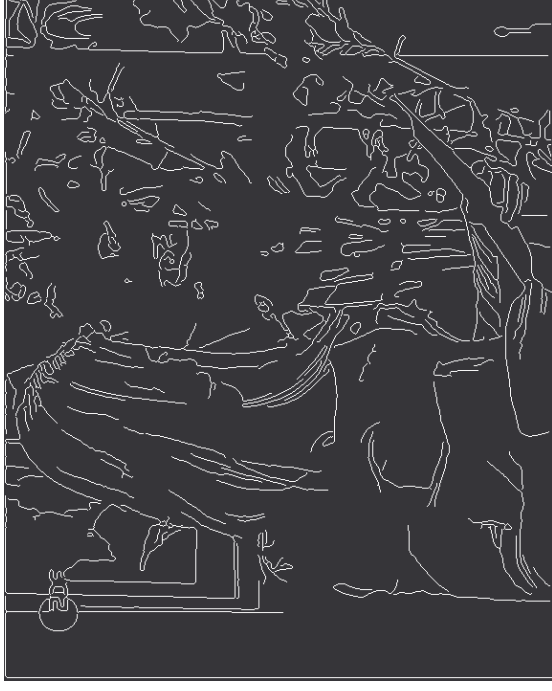
Visual Feature Spaces - Texture

- Very intuitive, but difficult to define formally
- Local spatial arrangement of gray-levels
- Depends on scale
- Many different approaches for analysis



Visual Feature Spaces - Object Shapes

- Start by extracting *edges*
- Link edges to obtain contours
- Very difficult, especially with noise and compression 
- Edges alone sufficient?



Our Visual Features

- **Color:**
 1. luminance histogram
 2. color histogram
 3. color temperature coefficients
 4. *Color Structure*
 5. *Dominant Colors*

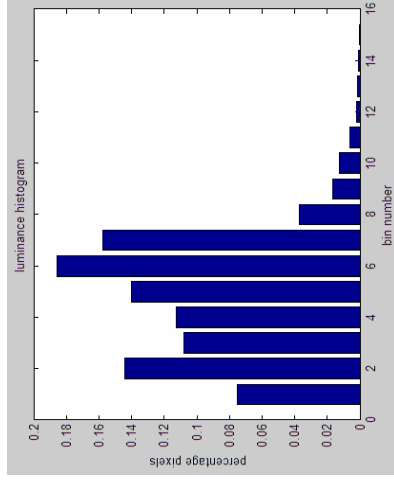
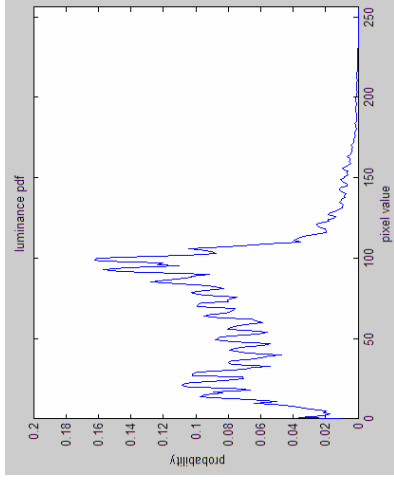
} *MPEG-7 Experimental Model*
- **Texture:**
 1. auto-correlation function →
 2. co-occurrence

Stijn de Waele's work on texture synthesis
- **Edges:**
 1. edge histogram

Color Features – Luminance Histogram

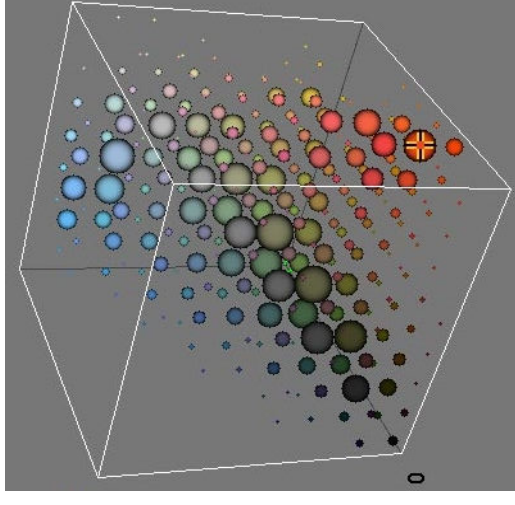
- Represent *distribution* of gray-levels
- Histogram = simplest non-parametric density estimator
 - Estimated *pdf* piecewise linear => discontinuities at bin boundaries (i.e. dependent on lighting)
- Y component (from YUV), 16 bins
- Similarity: histogram intersection

$$d_i(h_r, h_l) = \frac{\sum_{m=1}^M \min(h_l[m], h_r[m])}{\min(\sum_{m_j=1}^M h_l[m_j], \sum_{m_k=1}^M h_r[m_k])}$$



Color Features – Color Histogram

- Same as luminance histogram, but now *joint distribution* of all color components
 - all power of the histogram model, but also its drawbacks
- Color space = *HSV*
 - intuitive for humans => should model similarity well
- Quantization: $8 \times 4 \times 2 = 64$ bins total
- Similarity: histogram intersection



Color Features – Color Temperature

- Color values in an image change with illumination
- One of the main requirements within a scene: continuity of lighting
- Possible to estimate the *deviation* of image's colors from those under *reference* light
 - White-world assumption
 - Grey-world assumption



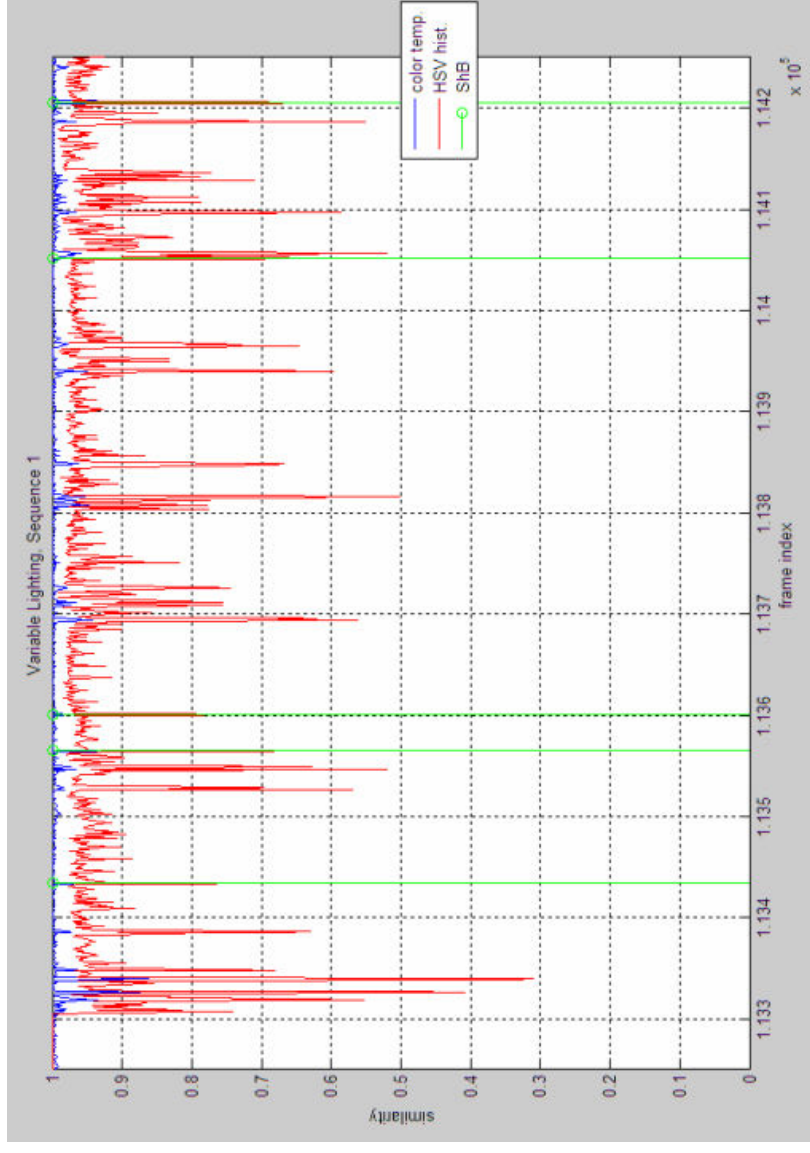
Color Features – Color Temperature (cont.)

- Very easy and fast extraction – three coefficients only

- Similarity measure = inverse of the average (relative) change over color channels

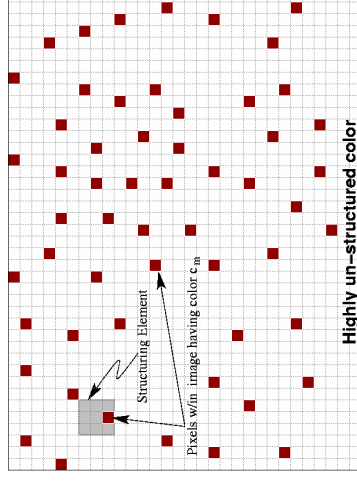
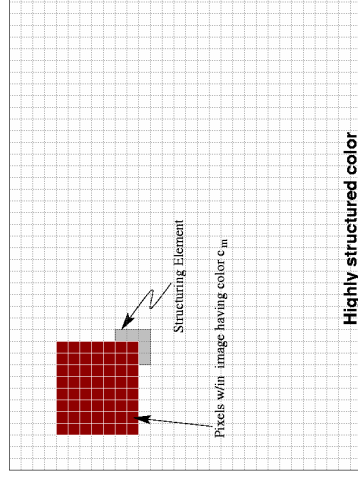
- Robust to lighting changes, can complement color histogram

min. CTS=0.75 @ last ShB



Color Features – Color Structure

- Part of *MPEG-7* set of descriptors
- Describes *structure of color* in the image
- Creates a histogram with counts referring to structuring elements
- *MPEG-7 HMMD* color space
- It is a histogram => histogram intersection for similarity



Figures taken from [ISO01]

Color Features – Dominant Colors

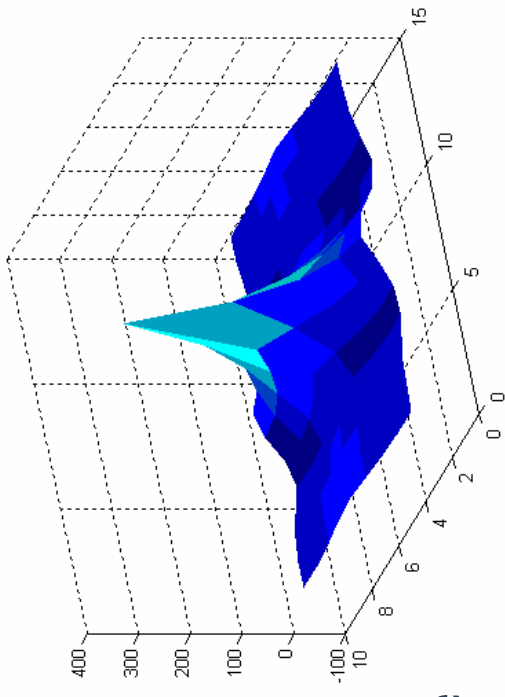
- Also part of *MPEG-7 Experimental Model*
- Describes *representative colors* of an image
- *RGB* color space
- Consists of:
 - spatial coherency of all colors
 - percentage of pixels that each covers
 - each color: 3 color values (i.e. *R, G, B*) & 3 variances
- Its own similarity metric
- Slow

Texture Features – Auto-correlation function (ACF)

- Based on work of Stijn de Waele on texture synthesis
- Describe texture-generating process by its first- and second-order moments (i.e. mean and auto-covariance)
- Correlation = how well two functions match (in an inner-product sense), relative to displacement r
 - auto-correlation = image correlated with itself
- ACF estimated from intermediate image frequencies
 - ⇒ Low (median image) and high (edges) removed first

Texture Features – Auto-correlation function (cont.)

- ACF estimated assuming an autoregressive (AR) model
- Mean estimated and subtracted
⇒ auto-covariance suffices to describe texture *pdf*
- Define cut-off level and derive auto-covariance matrix

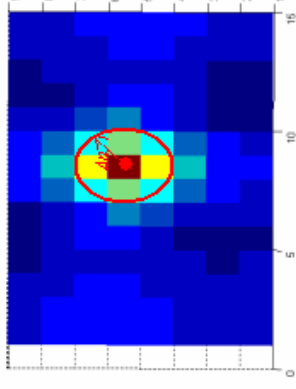


Texture Features – Auto-correlation function (cont.)

- Cut-off = 2 => 3x3 neighborhood = 9 variables
- Auto-covariance matrix 9x9
- Similarity – *Kullback-Leibler Divergence* (i.e. relative entropy)

$$D(p \parallel q) = \sum_x p(x) \log_2 \frac{p(x)}{q(x)} = \frac{1}{2} \left(\log_2 \frac{|\Sigma_2|}{|\Sigma_1|} - d + \text{tr}(\Sigma_2^{-1} \Sigma_1) \right)$$

- No well-defined upper bound => scale with arctangent



(a)

x_1	x_2	x_3	x_4
$R(0,0)$	$R(1,0)$	$R(0,1)$	$R(1,1)$
$R(-1,0)$	$R(0,0)$	$R(-1,1)$	$R(0,1)$
$R(0,-1)$	$R(1,-1)$	$R(0,0)$	$R(1,0)$
$R(-1,-1)$	$R(0,-1)$	$R(-1,0)$	$R(0,0)$

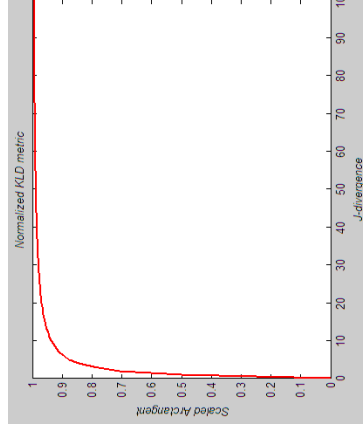
(b)

...	47.9791	173.1938	71.97382	...
...	75.66813	270.5069	75.66813	...
...	71.97382	173.1938	47.9791	...
...

(c)

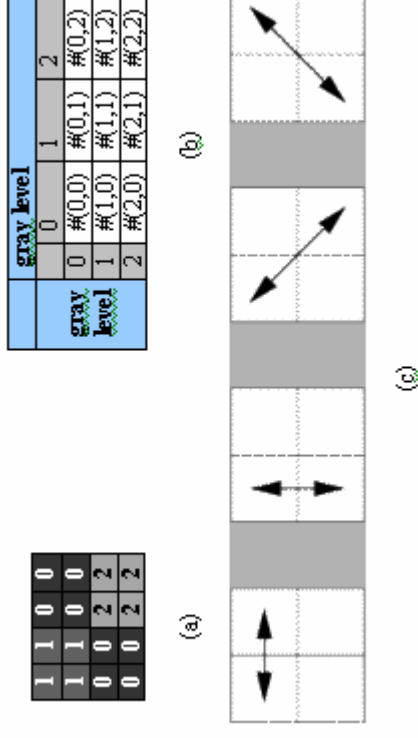
x_1	x_2	x_3	x_4
270.5	75.7	173.2	47.9
75.7	270.5	71.9	173.2
173.2	71.9	270.5	75.7
47.9	173.2	75.7	270.5

(d)



Texture Features – Co-occurrence

- Compute co-occurrences of neighbors' gray-level values
 - four directions: $0^\circ, 90^\circ, 45^\circ, 135^\circ$
 - relative displacement d



- Quantize image to G gray levels
- Create a matrix for each direction

Texture Features – Co-occurrence (cont.)

- Compute statistical features from matrix data

$$\text{Energy: } \sum_{a,b} P_{\phi,r}^2(a,b)$$

$$\text{Entropy: } \sum_{a,b} P_{\phi,r}(a,b) \log_2 P_{\phi,r}(a,b)$$

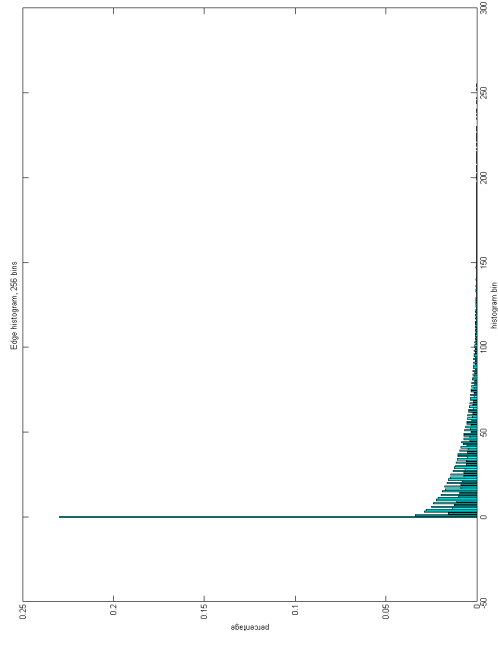
$$\text{Contrast: } \sum_{a,b} |a-b|^k P_{\phi,r}^{\lambda}(a,b)$$

$$\text{Correlation: } \sum_{a,b} \frac{[(ab)P_{\phi,r}(a,b)] - \mu_x \mu_y}{\sigma_x \sigma_y}$$

- **Similarity = Euclidean distance**
 - average over angles
 - overall distance = average over four features

Edge Features – Edge Histogram

- Segmentation difficult => only represent global edge distribution
- Sobel operator
 - horizontal + vertical
- Disregard first bin
 - often more than 90% of all pixels
 - re-normalize histogram
- Similarity – histogram intersection



Possible Applications in CASSANDRA Framework

- *MLC++* machine learning library
 - about 30 simple classifiers
 - cross-validation, learning curves...
- Our probabilistic model
 - Gaussian distribution of frame similarities
- Extension of the Parallel Shot (PS) concept possible
- Preliminary results only – should incorporate other features as well

Applications - Machine Learning with *MLC++*

- Obtained results for shot boundary (ShB) and scene boundary (ScB) detection
- Not possible to include a temporal window
- Results:
 - ShB: recall = **92%**, precision = **77%**
 - possible to increase precision, but at the expense of recall
 - ScB: recall = **51%**, precision = **19%**
 - also 24% and 26%
 - need other features for scene detection !!

Applications – Probabilistic Framework

- Assume frame distances distributed according to a Normal distribution

$$\bar{x} \sim N(\bar{\mu}, \Sigma)$$

- Update model parameters with time

$$\bar{\mu}_{n-1} = \frac{1}{n-1} \sum_{k=1}^{n-1} \bar{x}_k \quad \sigma_{n-1}^{ij} = \frac{1}{n-1} \sum_{k=1}^{n-1} (x_{ki} - \mu_i)(x_{kj} - \mu_j)$$

- For frame n , calculate distances to $n-2, n-1, n+1, n+2$

$$\bar{x}_n = [d_{n, n-2}, d_{n, n-1}, d_{n, n+1}, d_{n, n+2}]$$

Applications – Probabilistic Framework (cont.)

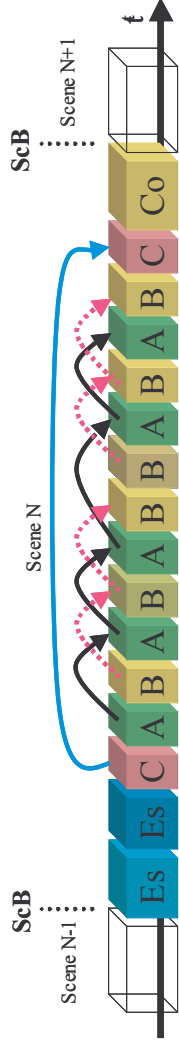
- At time $t=t_0+2$, decide about $t=t_0$
- If m clusters found so far, prob. that frame f_n belongs to cluster C_m is

$$p(f_n \in C_m) = p(\bar{x}_n) = \left(\frac{1}{\sqrt{2\pi}} \right)^r \frac{1}{\sqrt{|\Sigma|}} \exp \left\{ -\frac{1}{2} (\bar{x}_n - \mu_{n-1})^T (\bar{x}_n - \mu_{n-1}) \right\}$$

- If f_n is indeed an ScB, variance should be big, and probability small
- If $p > \tau$, f_n is a ScB
 - $m++$, $n = l$

Applications – Extension to Parallel Shots

- Parallel shots – alternating visually similar shots
 - e.g. dialog
 - never cross a scene boundary
- In CASSANDRA framework, salient-point based algorithms form links between such shots
- On average, 46% of all shots in movies and 72% in series [Nesv06]
 - can be used as pre-processing for ScB detection



Conclusions and Future Work

- Other features certainly necessary for ScB detection (ShB ok, but offline)
- Better results could be obtained with *MLC++* if temporal window included
- Color temperature can complement histogram
- Luminance histogram redundant
- *MPEG-7* descriptors reliable, but slow
- Texture auto-correlation extraction can be optimized
- Lots of parameters could be tweaked

Bibliography

- [Cost06] C. Costaces, N. Nikolaidis and I. Pitas, “Video Shot Detection and Condensed Representation”, *IEEE Signal Processing*, Vol.23, No.2, March 2006.
- [ISO01] *Text of ISO/IEC 15 938-3 Multimedia Content Description Interface – Part 3: Visual. Final Committee Draft, ISO/IEC/JTC1/SC29/WG11*, Doc. N4062, 2001.
- [NL05] J. Nesvadba, Y.S. Joshi and S. Pfundtner, “Parallel Shot Detector”, *Patent No. NL004360*, 2005.
- [Nesv06] J. Nesvadba and Y. Joshi, “Parallel Shot Detection for AV Content Segmentation”, submitted to *IEEE Int'l Conf. on Multimedia and Expo (ICME2006)*, Toronto, Canada, July 2006.

Questions?

