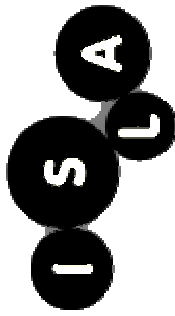
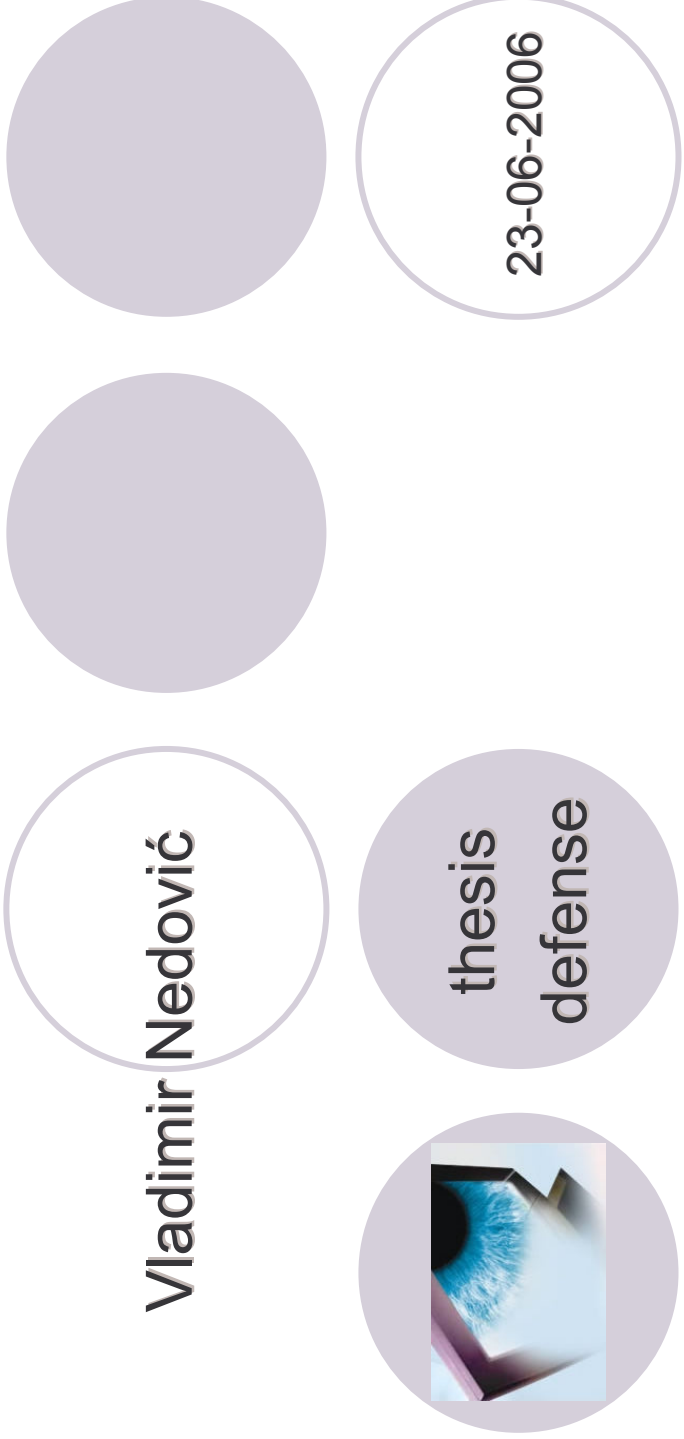


Visual Feature Extraction for Content Analysis



Intelligent Systems Lab
Amsterdam (ISLA)



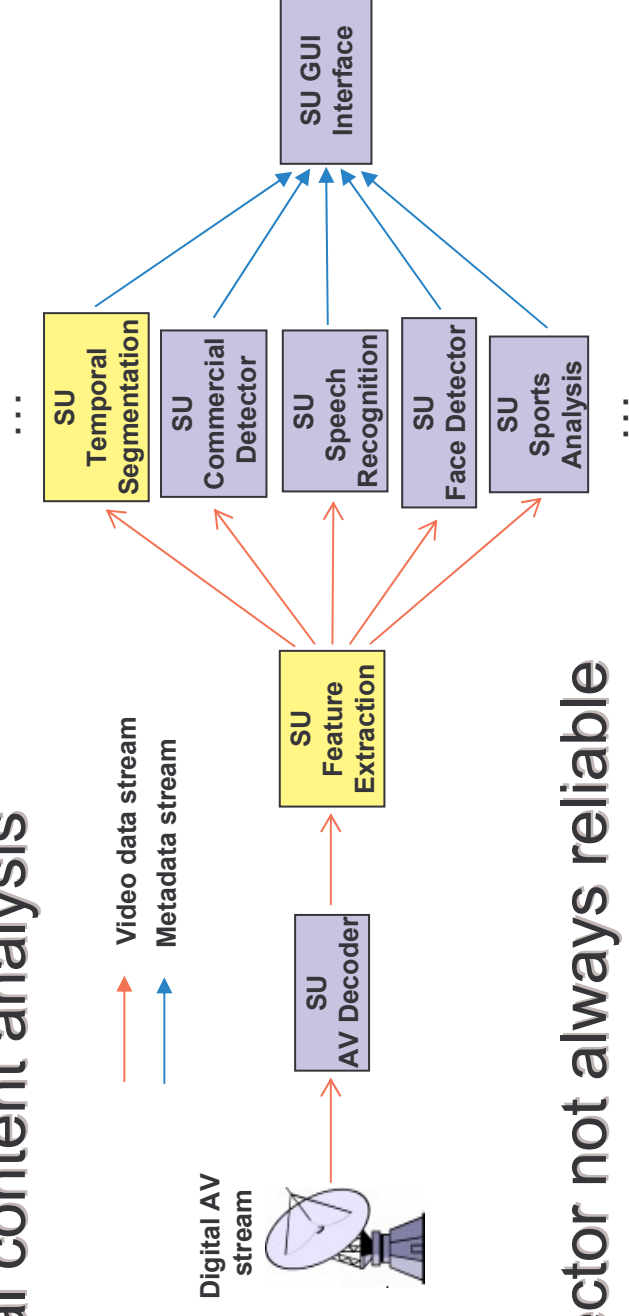
Storage Systems and
Applications Group



- **Problem Outline**
- Motivation
- Our Visual Features
- Applications in CASSANDRA Framework
(Philips Research) and some results
- Conclusions and Future Work

Problem Outline – The context

- CASSANDRA Framework – Distributed processing system for audio-visual content analysis



- Shot-cut detector not always reliable
 - Problems with high amount of motion
 - Problems in conditions of variable lighting
- Scene boundary detector needs much improvement

Problem Outline - Motivation

- An extensive feature set for temporal segmentation necessary in CASSANDRA framework
 1. Detection of *difficult shot boundaries*
 - visual features might suffice
 2. Detection of *scene boundaries*
 - features from multiple domains should probably be incorporated
- Motion and audio features available
 - ⇒ Focus on salient *visual* features
- For both tasks, fuse features and input to machine learning

Problem Outline - Criteria

- Main criterion: robustness to intra-shot variations, but sensitivity to changes across shot and scene boundaries
 - i.e. reliable metrics of visual similarity
 - Choose features based on the properties of the visual system
 - Refer to film-grammar rules
 - e.g. continuity of lighting required within entire semantic scenes
- Other criteria:
 - Robustness against noise, high compression rates
 - most of the material are TV broadcasts
 - Computational cost
 - constraints of the real-world system within Philips Research



- Problem Outline
- Motivation
- **Our Visual Features**
- Applications in CASSANDRA Framework
(Philips Research) and some results
- Conclusions and Future Work

Our Visual Features

- Color – most expressive:
 1. luminance histogram } global color distribution
 2. color histogram }
 3. color temperature coefficients → illumination color **NOVEL!**
 4. *Color Structure* } *MPEG-7 Experimental Model*
 5. *Dominant Colors* } (open source)
- Texture – statistical approach:
 1. auto-correlation matrix → prior work on texture synthesis within Philips Research **NOVEL!**
 2. co-occurrence features
- Edges – segmentation difficult => no shapes, edge pdf only:
 1. edge histogram

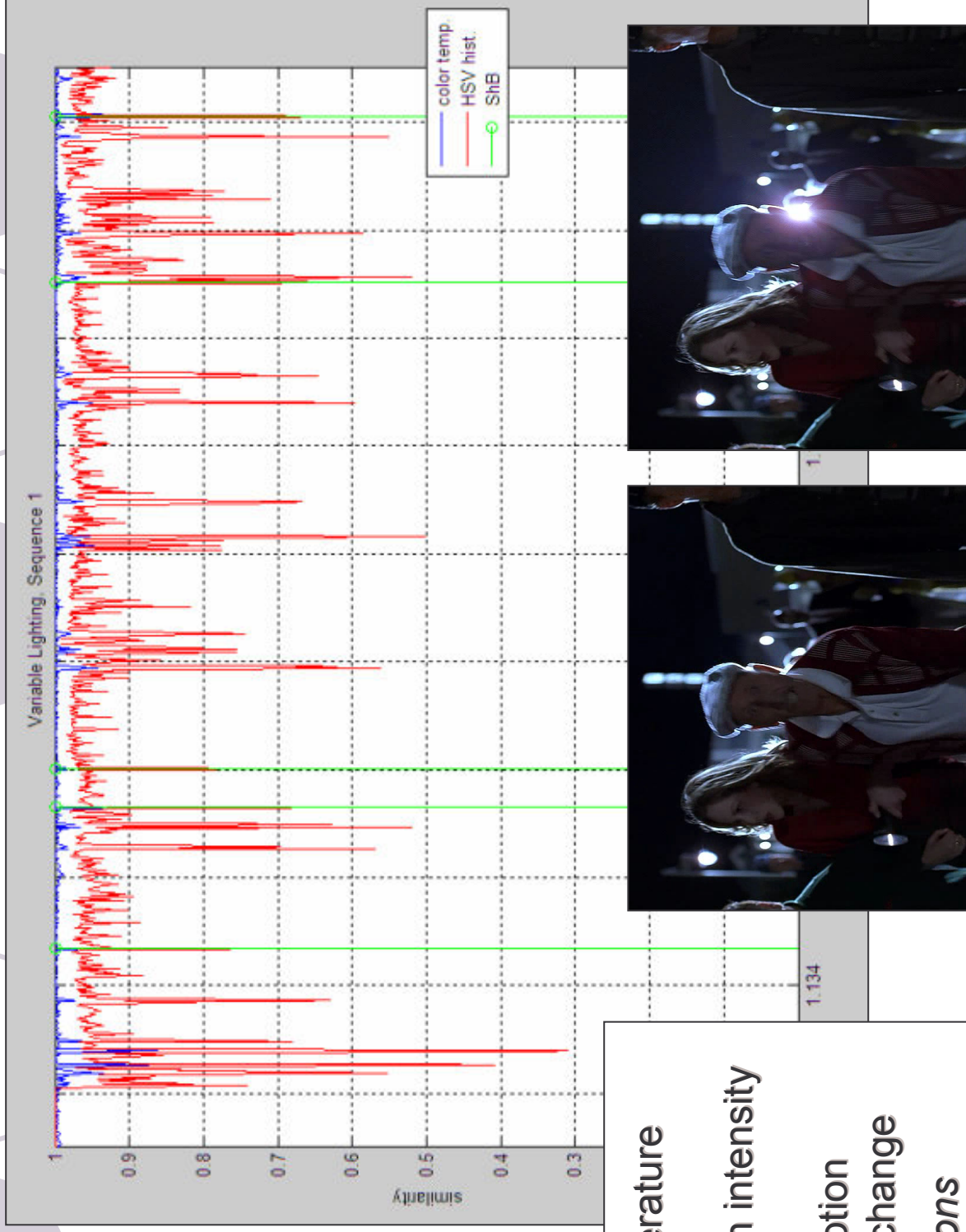
Color Temperature

- One of the main requirements within a scene: continuity of lighting
- *pdf* estimated in a histogram = piecewise linear => dependent on illumination
 - color temperature = robust to lighting changes, can complement color histogram
- Color temperature coefficients = *deviation* of image's colors from those under estimated *reference* light
- Very easy and fast extraction – three coefficients only



Color Temperature (cont.)

- Variable lighting scene -

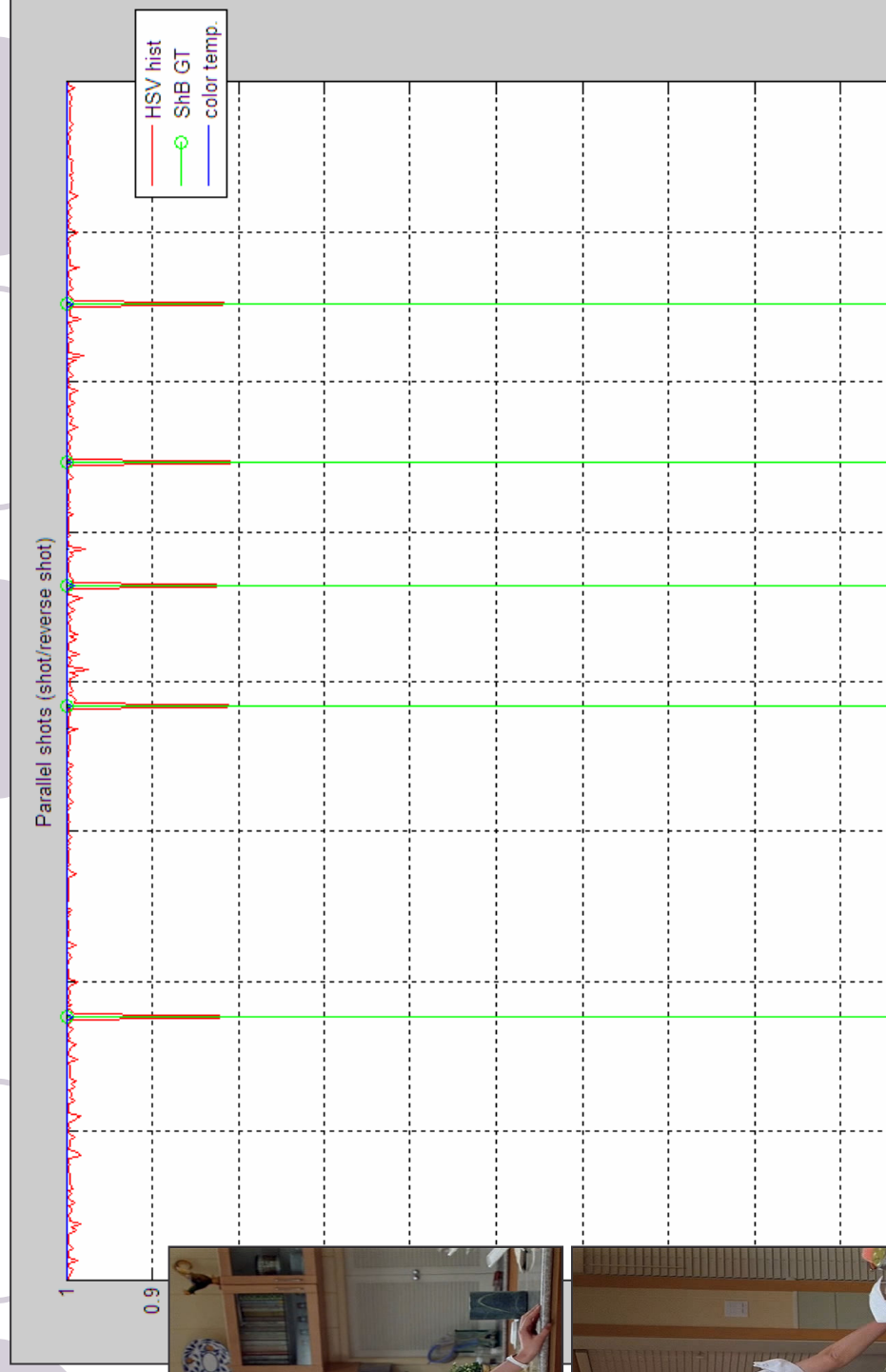


- Color temperature invariant to:
 - illumination intensity
 - highlights
 - camera motion
 - viewpoint change

⇒ *shot transitions*

Color Temperature (cont.)

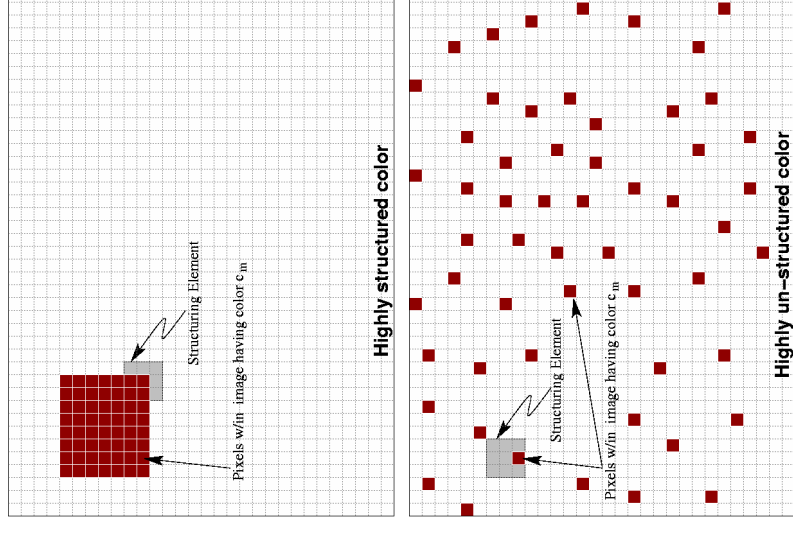
- Dialog scene -



- Color temperature invariant to viewpoint change
- Will respond to physical scene changes because of new light source
 - ⇒ *invariant to shot changes, dependent on scene changes*

MPEG-7 color descriptors

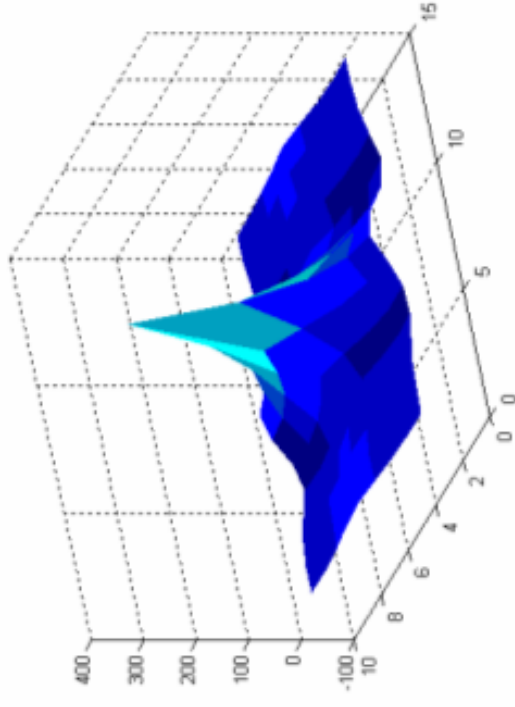
- **Color Structure**
 - *structure of color in an image*
 - a histogram of colors in structuring elements
 - reliable
 - not too practical, slow
- **Dominant Colors**
 - *representative colors in an image*
 - reliable, but not too transparent
 - slow



Color Structure figure taken from [ISO01]

Texture auto-correlation matrix

- Based on prior work on texture synthesis
 - describe the texture-generating stochastic process by its first- and second-order moments (i.e. mean and auto-covariance)
- Auto-correlation: comparison of the image with itself, as a function of relative displacement
 - correlation = normalized covariance
- Texture most prevalent in intermediate image frequencies
 - remove low frequencies (median image)
 - remove high frequencies (edges)

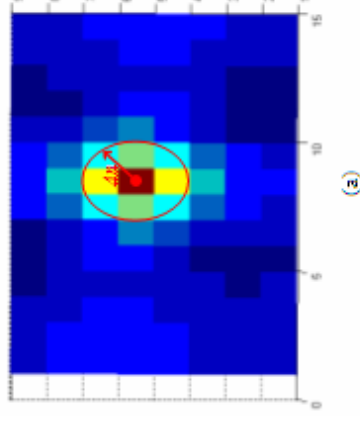


Texture auto-correlation matrix (cont.)

- Mean estimated and subtracted => auto-covariance suffices to describe Gaussian texture pdf
- Define cut-off level and derive auto-covariance matrix
- Similarity: *Kullback-Leibler Divergence* (i.e. relative entropy)

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

- No well-defined upper bound for *KLD* values => scale with arctangent



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

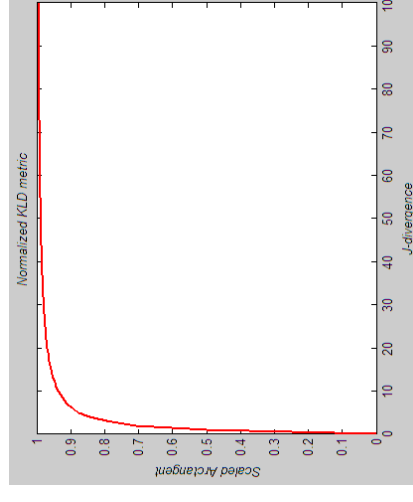
(b)

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

(d)





- Problem Outline
- Motivation
- Our Visual Features
- Applications in CASSANDRA Framework
(Philips Research) and some results
- Conclusions and Future Work

Possible Applications in CASSANDRA Framework

- *MLC++* machine learning library (open source)
 - Test features for shot and scene boundary detection
 - Note: results for scene detection are preliminary – should incorporate other features as well
- Our probabilistic model
 - Assume Gaussian distribution of frame similarities
- Extension of the parallel-shot concept possible
 - Form links between intra-scene shots

Applications – Machine learning with *MLC++*

- About 30 simple classifiers
 - Naïve Bayes, Decision trees, Nearest Neighbors, voting classifiers...
- Prepare input in C4.5 format
 - input = vector of feature similarity values (i.e. early fusion)
 - similarity values between successive frames
 - normalized to a [0..1] range
 - i varies over features, j varies over samples:

$$x'_{ij} = \frac{x_{ij} - \mu_i}{3\sigma_i}$$

0.999223,	0.99425,	0.999178,	1,	0.99056,	0.995302,	0.995321,	0.999901,	notShBnd
Color temperature similarity	Texture correlation similarity	Texture co-occurrence similarity	Luminance histogram similarity	Color histogram similarity	Edge histogram similarity	Dominant Colors similarity	Color Structure similarity	class label

Applications – Machine learning with MLC++ (cont.)

- Tested features on a two-hour video *Master and Commander*:
 - dim lights, lots of smoke, fast motion, etc.
 - current shot detector:
 - temporal window = 15 frames
 - recall = **90%**, precision = **84%**
- Shot boundary (ShB) results:
 - recall = **92%**, precision = **77%** (Bagging with Perceptron)
 - possible to increase precision, but at the expense of recall
 - but good results with *color histogram alone*:
 - R = **87%**, P = **72%** (Winnow)
 - **76%** each (Bagging with Perceptron)

Applications – Machine learning with MLC++ (cont.)

- Preliminary scene boundary (ScB) results:
 - recall = **51%**, precision = **19%** (Naïve Bayes)
 - with kNN: 24% and 26%, respectively
 - with *color temperature only*:
 - R = 47%, P = 22% (Perceptron)
 - R = 28%, P = 29% (Bagging with Perceptron)
- In both tasks, a particular single feature can perform as well as the whole set
 - ⇒ no apparent contribution from fusion (of other *visual* features)
- Could incorporate a temporal window for better results
- Results might also improve for other parameter settings

Applications – Probabilistic Framework

- Assume frame distances are normally distributed
- For frame n , calculate distances to $n-2, n-1, n+1, n+2$
 \Rightarrow at time $t=t_0+2$, decide about t_0

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

- Update model parameters with time

$$\bar{\mu}_{n-1} = \frac{1}{n-1} \sum_{k=1}^{n-1} \bar{x}_k$$

$$\sigma_{n-1}^2 = \frac{1}{n-1} \sum_{k=1}^{n-1} (x_{nk} - \mu_j)(x_{nk} - \mu_j)$$

- If m clusters found so far, prob. that frame f_n belongs to cluster C_m is

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

- If f_n is indeed an ScB, variance should be big, and probability small
 - If $p > \tau$, f_n is a ScB $\Rightarrow m++$, $n=1$

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 a pre-processing step for ScB detection
 - links could be made between the PS portion and the rest of the shots on the basis of visual features
 - not tested

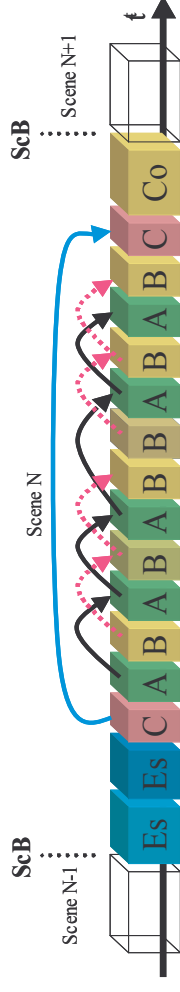


Figure taken from [NL05]



- Problem Outline
- Motivation
- Our Visual Features
- Applications in CASSANDRA Framework
(Philips Research) and some results
- **Conclusions and Future Work**



Conclusions

- Visual features alone not sufficient for ScB detection
- For ShB detection, results similar to the shot detector currently in the framework
 - processing is done offline => would be practical only for difficult video content
- Better results could be obtained with *MLC++* if temporal window is included
 - maybe also with other parameter settings
- Luminance histogram might be redundant
- *MPEG-7* descriptors reliable, but slow

Conclusions (cont.)

- However...
 - color histogram as efficient in ShB detection as the whole feature set
 - &
 - color temperature alone can achieve the same (or better) results as the whole set in ScB detection
 - =>
 - other visual features are redundant?
 - late fusion of features instead of early fusion?
-
- Color histogram and color temperature could be very inexpensive but efficient early indicators of shot & scene boundaries
 - could be combined for real-time processing

Conclusions – Contribution

- Development of a robust visual feature set within CASSANDRA project at Philips Research Eindhoven
 - will be used further for temporal video segmentation, content description & analysis, etc.
 - novel features developed
 - color temperature, texture auto-correlation matrix
- Identification of the importance of color temperature
 - simple, but powerful
- Color histogram + temperature = good early indicators of shot/scene changes
 - ⇒ color information very powerful

Conclusions – Future Work

- Machine Learning is the way to proceed
- other features should be incorporated for ScB detection
 - motion, audio & speech features are available
- Principal Component Analysis should be done first?
 - reduce complexity
 - eliminate redundant features
- late fusion instead of early fusion?
 - use simple features such as color histogram and color temperature as early indicators
 - fuse results later

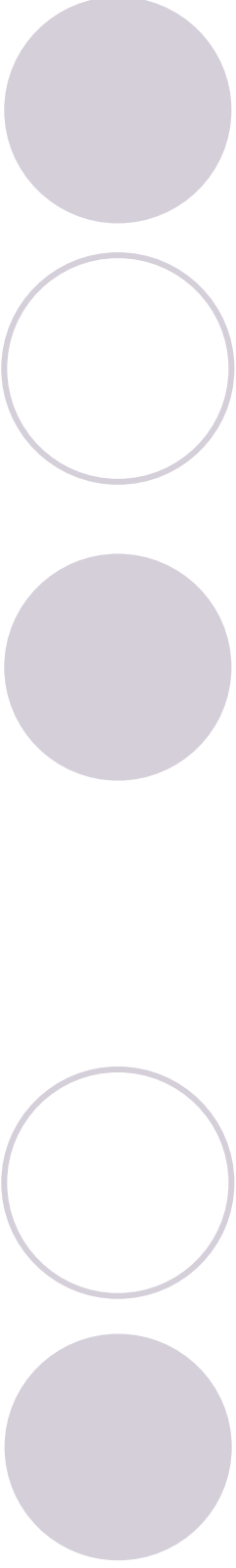
Conclusions – Future Work (cont.)

- Explore color temperature feature further
 - evaluate the similarity metric more extensively
 - develop a method to combine it with color histogram
- Test individual features in similarity tasks
 - *MPEG-7* color descriptors
 - auto-correlation matrix
 - compare with auto-correlation coefficients usually used
- Reduce complexity
 - texture auto-correlation, *MPEG-7* features



Bibliography

- [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.*
- [Nesv05] J. Nesvadba et al., “Real-Time and Distributed AV Content Analysis System for Consumer Electronics”, in *Proc. of Int’l Conf. on Multimedia and Expo*, pp.1549-1552, Amsterdam, The Netherlands, June 6-8, 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.
- [NL05] J. Nesvadba, Y.S. Joshi and S. Pfundtner, “Parallel Shot Detector”, *Patent No. NL004360, 2005.*



Questions ?