

# Image and Video Databases: Who Cares?

**Edward J. Delp**

**Purdue University**

**School of Electrical and Computer Engineering  
Video and Image Processing Laboratory (*VIPER*)**

*ace@ecn.purdue.edu*

*http://www.ece.purdue.edu/~ace*

*http://www.ima.umn.edu/~delp*



# *VIPER* Research Projects

- Scalable Video and Image Compression
  - color still image compression (*CEZW*)
  - high and low bit rate video compression (*SAMCoW*)
  - wireless video and streaming media
- Error Concealment
- Content Addressable Video Databases (*ViBE*)
- Multimedia Security: Digital Watermarking
- Embedded Real-Time Image and Video Processing
- Analysis of Mammograms



# Outline

- Who Cares?
- The content-based video retrieval problem
  - MPEG-7
- What is *ViBE*?
- Temporal segmentation of video sequences
- Pseudo-semantic labeling of shots
- Future work



# Penetration of TV/Video

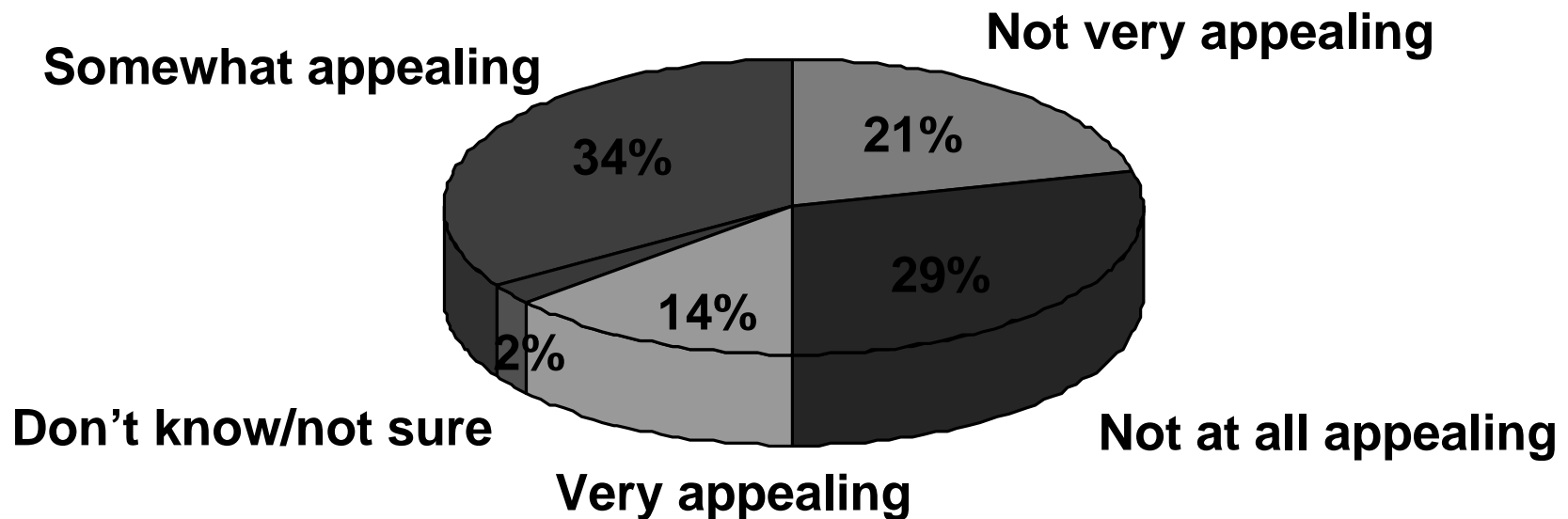
- **Percentage of US households with**
  - **at least one TV set: 98%**
  - **two TV sets: 34%**
  - **three or more TV sets: 40%**
  - **at least one VCR: 84%**
- **The average American watches 3 hours and 46 minutes of TV each day**

**Source:** A.C. Nielsen Co. (1998), [http://www.oc-profam-net.org/media/tv\\_statistics.htm](http://www.oc-profam-net.org/media/tv_statistics.htm)



# Maybe TVs Should Just Stay Dumb?

## How appealing is Interactive TV?



Source: Angus Reid Group, *Red Herring* August 2000, out of 1000 Americans

IMA March 1, 2001

Slide 5



# What Do Users Want?

- **Time-shifting programs 47%**
- **Video conferencing 36%**
- **Video on demand 35%**
- **Getting many more channels 33%**
- **Being able to control camera angles 30%**
- **Using TV to surf the web 24%**
- **Using TV to write and receive email 24%**
- **Play games with groups of people who have iTV 14%**
- **Shopping over TV 12%**

Source: Angus Reid Group, *Red Herring* August 2000, out of 1000 Americans

*IMA March 1, 2001*

Slide 6



# Video Database Problem

- **How does one manage, query, and browse a large database of digital video sequences?**
- **Problem Size**
  - **One hour of MPEG-2 is 1.8 GB and 108,000 frames**
- **Goal - browse by content (how do you find something?)**
  - **applications include digital libraries**
- **Need for compressed-domain processing**
  - **What type of compression should be used?**
- **Network Services: QoS?**



# Content-Based Access Applications

- **Professional**
  - Large video archives
  - Surveillance video archiving
- **Educational**
  - Multimedia libraries
  - Distance education by video streaming
- **Consumer applications:**
  - Content filtering and time shifting
  - Home video database???



# Goals

- **Management of a large video database**
  - database issues, scalability, etc.
- **Browsing video data in the database**
  - how best to present data to the user
  - user must get an idea about the whole database
- **Searching video data in the database**
  - query languages
- **Given a video sequence, how can one rapidly get an idea about its content? (video summarization)**



# Application Models

- **Consumer Model**
- **Video-on-Demand Model**
- **Digital Library Model**



# Consumer Model

- **Scenario: Consumers will acquire more images and video using “cheap” digital cameras**
  - hence, there will be a market for home image and video database management products
- **Everybody has a computer, web page, digital camera, and editing software**



# Consumer Model

- Applications:
  - search your database for images of your children as they grow
  - find the wedding pictures



**In the next 10 years more than 90% of images and video in your life will be “digital”**



# Consumer Model

- **Wrong!**
  - More than 60 billion photos taken each year
    - each image is looked at *less than one time*
    - same applies to video
  - How do most consumers do it now?
    - Shoe box (will having pixels change this?)
- What is the payoff for consumers to manage their images?
- Industry will not be able to sell enough systems to people, other than techno-geeks, to make any money

IMA March 1, 2001

Slide 13



# Video-on-Demand

- **Scenario: consumers need a video database to search for entertainment videos**
  - database system can generate a customized preview/synopsis for the viewer based on preference information
- **System would be available on the Internet or cached by the cable system**



# Video-on-Demand

**Query: “Show me an Arnold Schwarzenegger movie where 20 people are killed in the first 11 minutes and 20 seconds. I want to see a preview of all the deaths now.”**



# Video-on-Demand

- **This is a dream!**
- **The average person chooses a movie based on: topic, actors, director, previews, advertising, and reviews**
  - much of this is text-based
- **It is not obvious that people want to interact with their entertainment**
  - interactive entertainment other than video games for boys has been a big loser
- **What is needed: a really good program guide**



# Digital Library Model

- **Scenario: a networked-based image and video database exists to provide educational value to a user and/or capitalistic advantage to a company**
- **Model differs from above in that the system is managed by professionals**
  - **model similar to a university or corporate research library**
  - **system may provide entertainment value *but* it is not solely organized for this purpose**



# Digital Library Model

- **Users (even consumers) use the system in “research” mode**
- **For example: NBA video database of all NBA games**
  - **sportscasters use for reports**
  - **highlight videos**
  - **used for scouting by teams**
  - **fan use: “Show me all the clips of Michael Jordan doing a reverse slam dunk where he pushes off on his left foot”**



# Digital Library Model

- People will be users of our systems *NOT* managers
- This model has the most payoff for the user and the research community



# More Comments

- **Some of the “applications” used to justify why we are doing some of the research are ludicrous and ill-conceived**
- **What is the correct application model?**
- **Is there hope.....you bet!**
- **By the way, what is the “killer app” for our work? I know one but it is not nice!**



# Goals

- **Management of a large video database**
  - database issues, scalability, etc.
- **Browsing video data in the database**
  - how best to present data to the user
  - user must get an idea about the whole database
- **Searching video data in the database**
  - query languages
- **Given a video sequence, how can one rapidly get an idea about its content? (video summarization)**



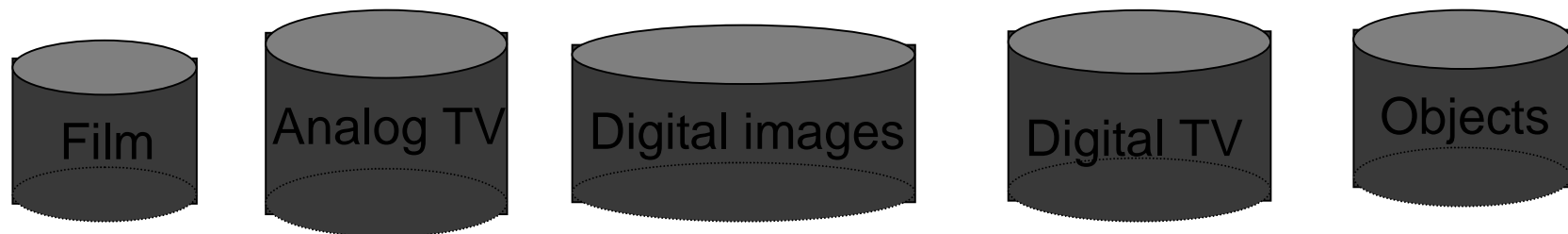
# Further Motivation

- **Digital libraries**
- **“Intelligent” TV**
- **More than 6 million hours of feature films and video archived worldwide (increasing 10% per year)**
- **The indexing effort is estimated to be 10 hours of work per one hour of video data**



# MPEG-7

## Multimedia Content Description Interface



Reusability of content for many applications

Digital studios  
Internet  
Mobile applications



# What is “Content”?



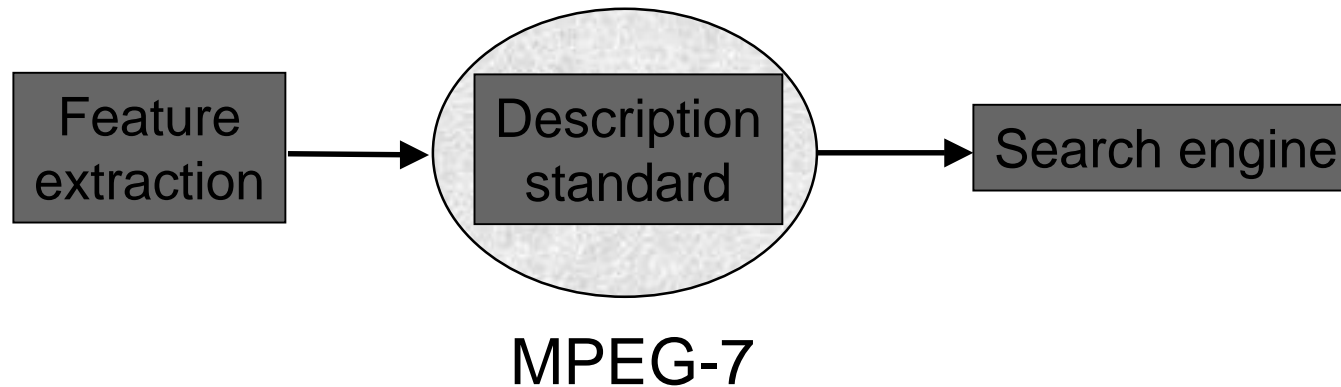
- “Play it again Sam!”
- Man facing woman
- Casablanca
- Ingrid Bergman
- Humphrey Bogart
- Famous movies
- Close-up shot
- “Not Sports”

**Content is dependent on the particular user group querying the system**



# MPEG-7 Framework

- **MPEG 7 will provide standardized descriptions of various types of multimedia information - it is not a video compression standard**



- **MPEG 7 will not provide tools to extract the multimedia information**



# ***ViBE*: A Video Database Structured for Browsing and Search**

*IMA March 1, 2001*

Slide 26



# ViBE Research Team

- **Purdue University**
  - **Charles Bouman, Cuneyt Taskiran, and Edward Delp**
- **Universidad Politecnica de Valencia (Spain)**
  - **Alberto Albiol**
- **Universidad Politecnica de Catalonia (Spain)**
  - **Luis Torres**

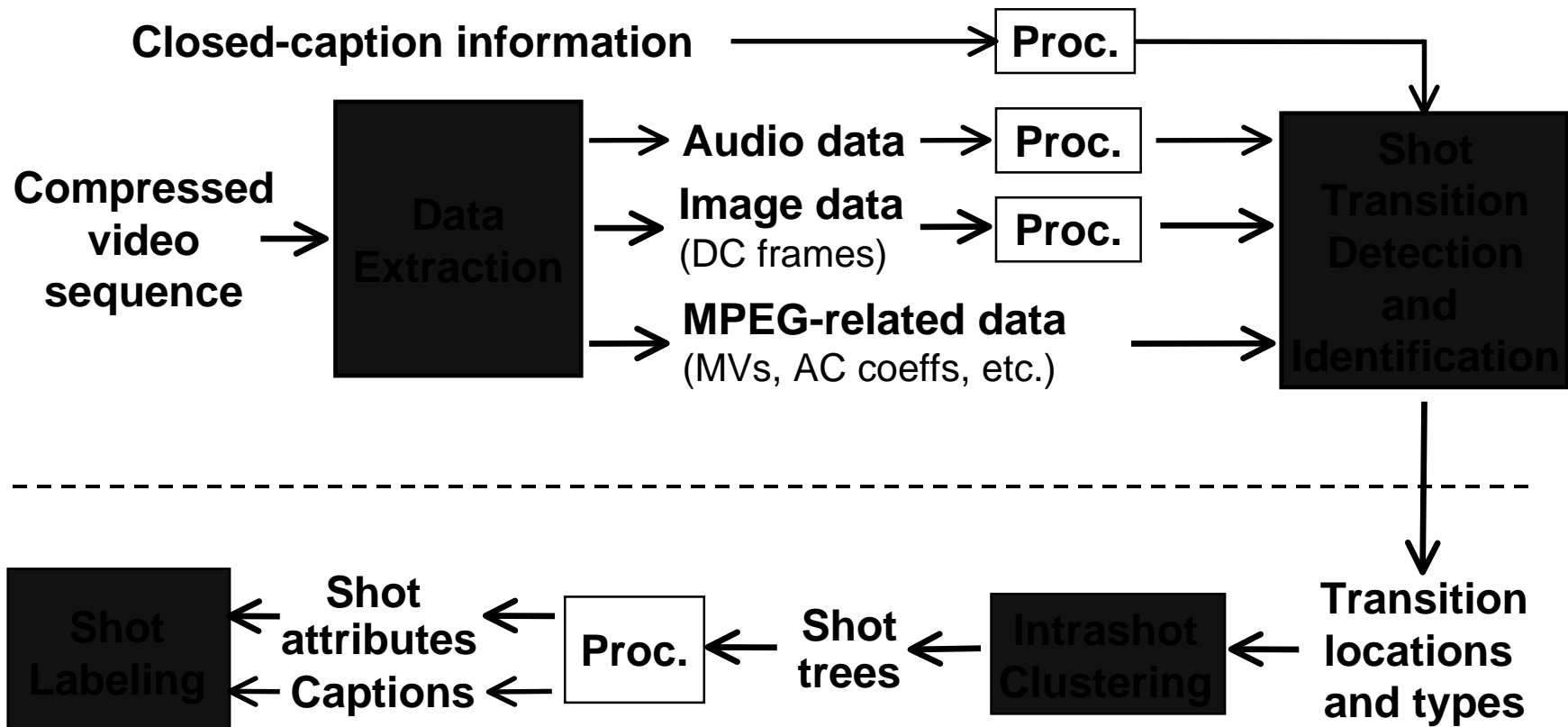


# The Problem

- How does one manage, query, and browse a large database of digital video sequences?  
(search/browse)
- Given a video sequence, how can one rapidly get an idea about its content? (video summarization)
- Problem Size
  - One hour of MPEG-2 is 1.8 GB and 108,000 frames
  - 0.5K - 1K shots per hour



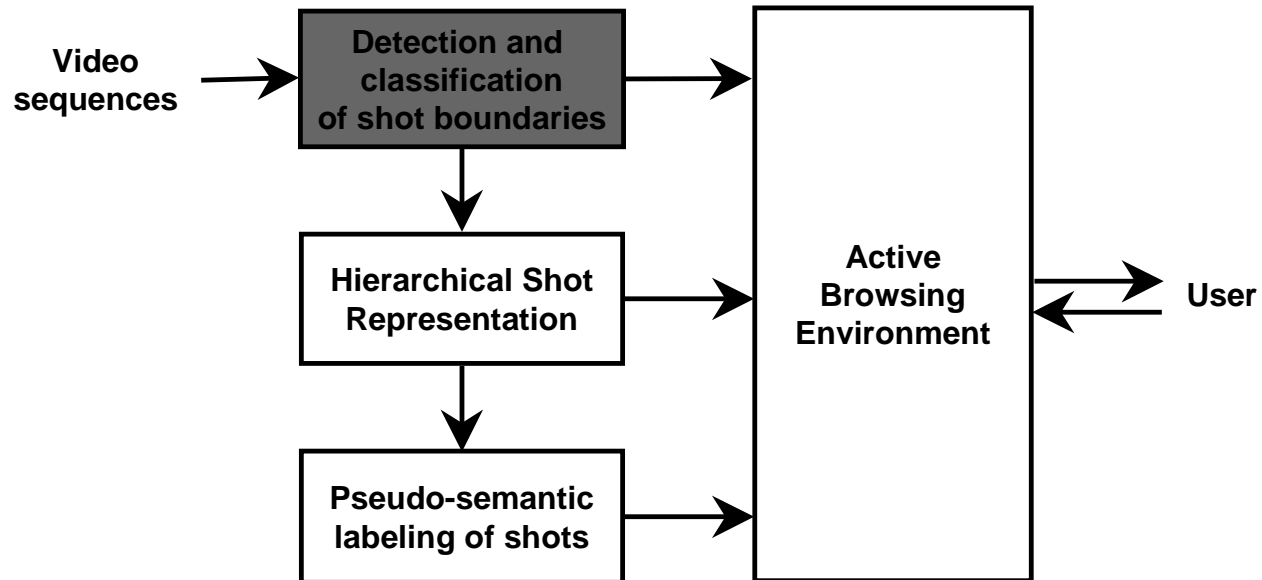
# Video Analysis: Overview



# ***ViBE*: A New Paradigm for Video Database Browsing and Search**

- ***ViBE* has four components**
  - shot boundary detection and identification
  - hierarchical shot representation
  - pseudo-semantic shot labeling
  - active browsing based on relevance feedback
- ***ViBE* provides an extensible framework that will scale as the video data grows in size and applications increase in complexity**





# Temporal Segmentation of Video Sequences

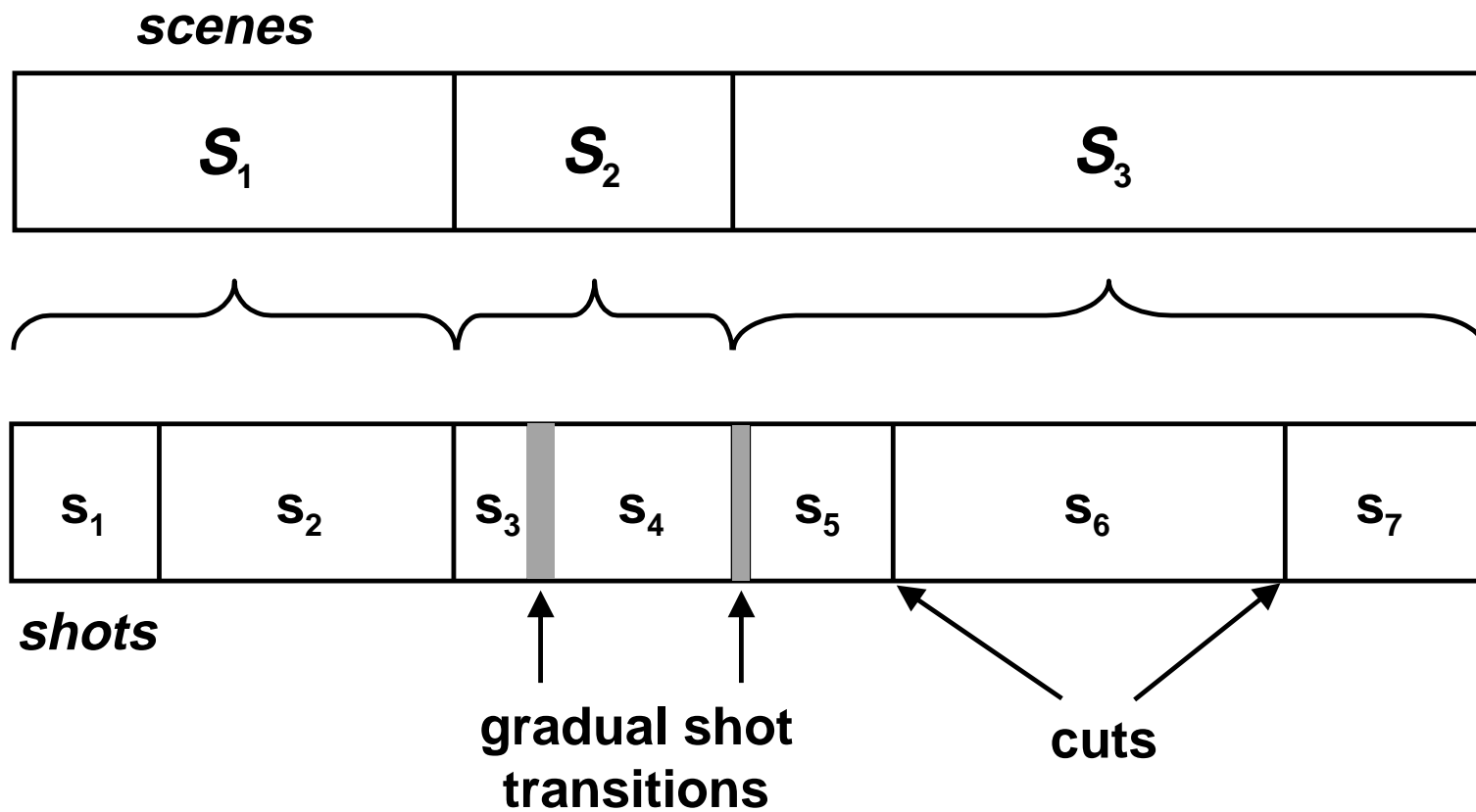


# The Temporal Segmentation Problem

- Given a video sequence, segment it into groups of frames that have continuity in some general conceptual or visual sense
  - the segmented units will usually correspond to *shots* in the sequence
- This task requires the detection of shot boundaries
- Identification of the *types* of shot boundaries is also important



# Hierarchical Structure of Video



# Examples of Some Shot Transitions



**Dissolve**



**Fade-out**



**Wipe**



# Why is Temporal Segmentation Important?

- **Breaks video up into manageable “semantic chunks”**
- **Shot transitions give valuable information relative to shot content**

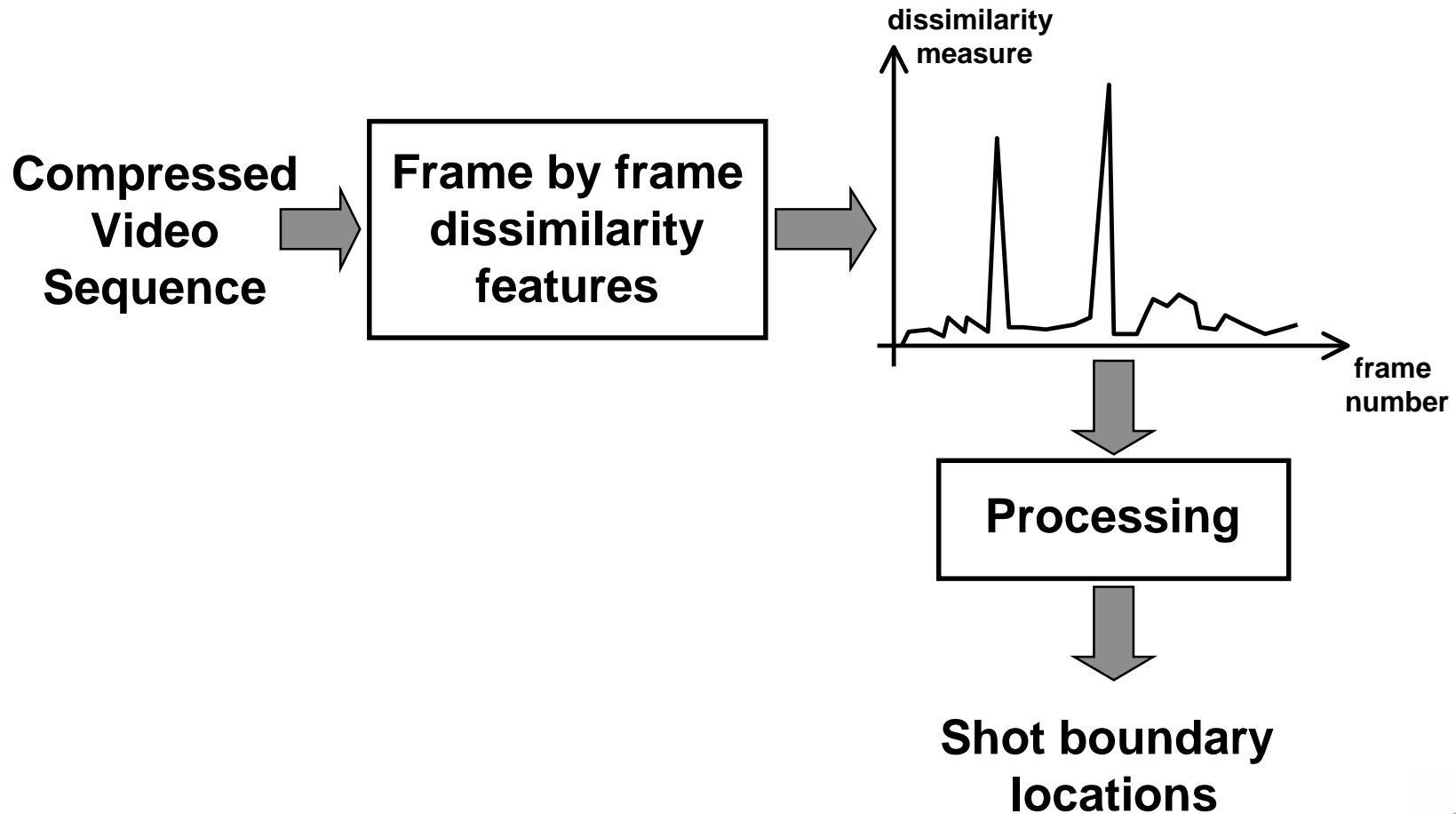


# Previous Work

- **Many methods**
  - **Pixelwise frame differences** (Y comp of DC frames)
  - **Color histograms** (histogram intersection Y, U, and V comps)
  - **Dimensionality reduction techniques** (VideoTrails, KLT)
  - **Edge detection** (entering and exiting edge pixel ratios)
  - **Motion vector information** (types of MBs)
  - **Model-based methods** (shot duration modeling)



# Common Approach to Temporal Segmentation



# Problems With This Approach

- What type of feature(s) should be used?
- How do we choose threshold(s) robustly?
- Classes (cut vs. dissolve, etc.) may not be separable using one simple feature

Using a multidimensional feature vector may alleviate these problems



# Working in the Compressed Domain

- **Advantages**
  - Working with any reasonable amount of video is impossible in the uncompressed domain
  - Compressed data stream contains many useful features that were computed by the encoder
  - No need for recompression after analysis
- **Disadvantages**
  - Resolution of DC frames too low for some operations
  - MPEG features may be encoder dependent
  - The MVs may not be reliable in some cases



# The DC Sequence

- The *DC coefficient* of a DCT block is given by

$$X(0,0) = \frac{1}{8} \sum_{m=0}^7 \sum_{n=0}^7 x(m,n)$$

- A *DC frame* is created by the DC coefficients of the 2-D DCT of a frame in the compressed sequence
- Set of DC frames for a sequence is known as the *DC sequence*
  - For I frames, DC coefficients are already present
  - For P and B frames, DC coefficients estimated using the MVs using Shen and Delp's (1995) method

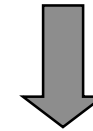


# An Example of an Extracted DC Frame



↑  
**Decompressed original frame**  
**352 × 240**

**DC Frame**  
**44 × 30**



# The Generalized Trace (GT) / Regression Tree Methodology

- Given a compressed sequence, first the DC sequence is derived
- Features are extracted from each DC frame in the DC sequence and these are placed in a feature vector known as the *Generalized Trace*
- Uses features which are readily available from the MPEG stream with minimal computation
- Regression tree (Gelfand, Ravishankar, and Delp 1991) is used to process the GTs and detect and classify shot transitions



# List of Features

- The GT feature vector consists of
    - $g_1$  : Y component
    - $g_2$  : U component
    - $g_3$  : V component
    - $g_4$  : Y component
    - $g_5$  : U component
    - $g_6$  : V component
    - $g_7$  : Number of intracoded MBs
    - $g_8$  : Number of MBs with forward MV
    - $g_9$  : Number of MBs with backward MV
    - $g_{10} - g_{12}$  : Frame type binary flags
- histogram intersections
- frame pixel variances
- Not applicable to all frames



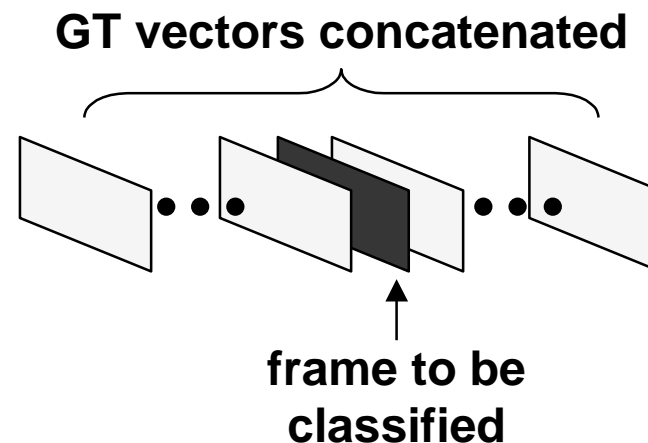
# Decision Trees

- **Advantages**
  - Has a form which is simple and relatively easy to understand
  - Increases efficiency by not testing a sample against all classes
  - Feature subsets at each node can be optimized locally
  - Avoids the “curse of dimensionality” for small sample size
- **Disadvantages**
  - Data fragmentation due to decrease in subset size
  - Overlap between terminal nodes may reduce efficiency



# Feature Windowing

- Place a window centered around the frame to be classified
- Place the GT vectors in the window into one large vector and use this vector as input to the regression tree
- Increases classification robustness

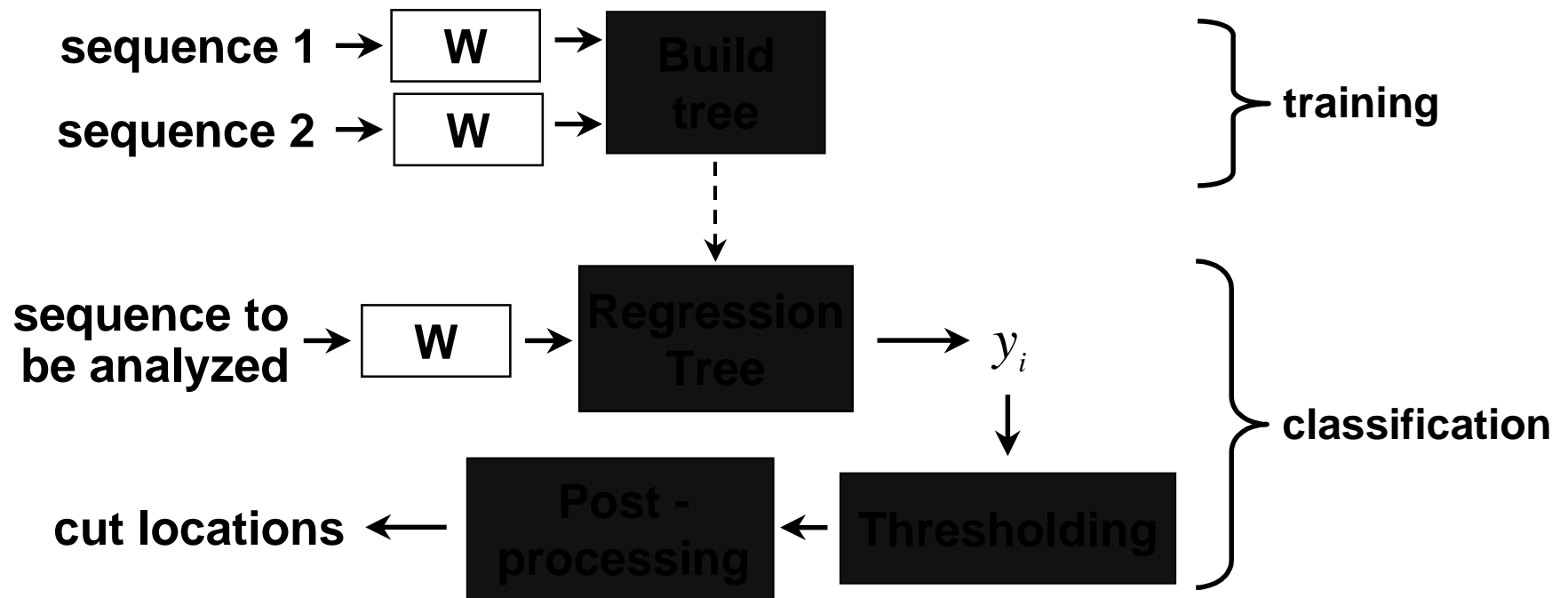


# **Advantages of the GT/Regression Tree Methodology**

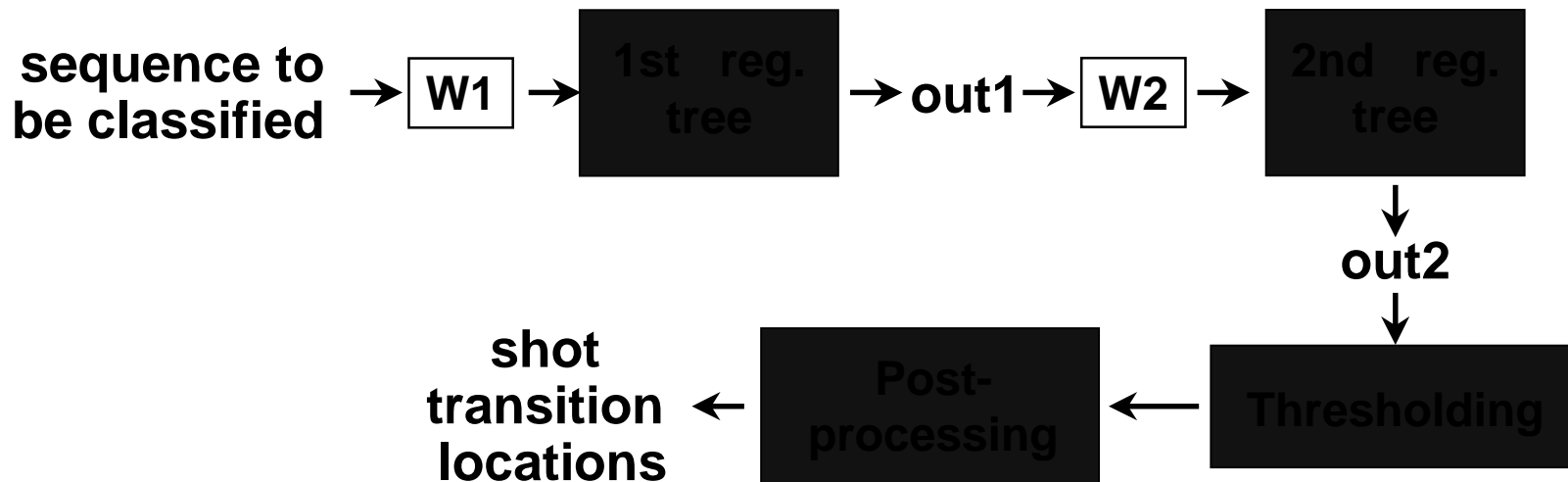
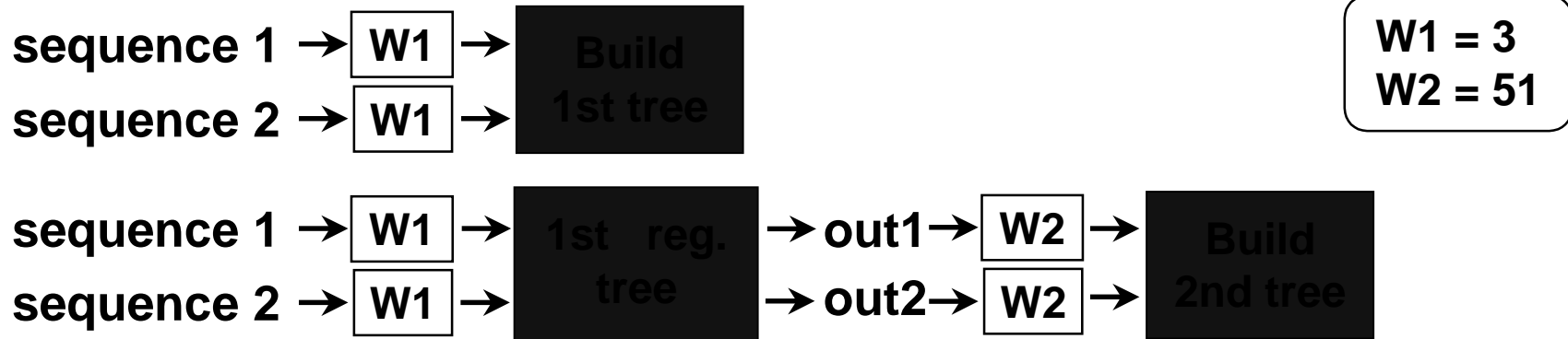
- **Regression tree provides normalized output which allows consistent thresholding of different sequences**
- **The tree weighs the features from the GT to suit the window of frames being analyzed**
- **A multitude of different features are collectively used to detect shot boundaries**
- **The method is highly extensible to include other features**
- **A unified framework is provided to deal with different kinds of shot transitions**



# Detecting Cuts



# Detecting Gradual Transitions



# Postprocessing of Results

- **Cuts**
  - If two cuts are closer than 10 frames, delete one
- **Dissolves and fades**
  - If two transitions are closer than 30 frames, they are combined
  - If the length of a transition is less than 3 frames or greater than 200 frames, it is deleted



# The Data Set for Cut Detection Experiments

- Digitized at 1.5Mb/sec, CIF format (352x240), MPEG-1
- Contains more than 10 hours of video
- 6 different program genres
- 10 min clips were recorded at random points during the program and commercials were edited out
- A single airing of a program is never used to obtain more than one clip (except *movies*)



# Data Set Statistics

	<b>frames</b>	<b>cuts</b>	<b>dissolves</b>	<b>fades</b>	<b>others</b>
<i>soap opera</i>	67582	337	2	0	0
<i>talk show</i>	107150	331	108	1	6
<i>sports</i>	78051	173	45	0	29
<i>news</i>	58219	297	7	0	6
<i>movies</i>	54160	262	15	6	1
<i>cspan</i>	90269	95	19	0	0
<b>TOTAL</b>	<b>455431</b>	<b>1495</b>	<b>196</b>	<b>7</b>	<b>42</b>



# Experimental Procedure

- **Use a cross-validation procedure to determine performance**
  - for each genre  $G$  {soap, talk, sports, news, movies, cspan}
  - for  $i = 1$  to 4
    - randomly choose  $S1$  and  $S2$ , both not in  $G$
    - train regression tree using  $S1$  and  $S2$
    - process all sequences in  $G$  using this tree
    - average performance over  $G$
  - average the four sets of values to find performance for  $G$
- **Window size = 3 frames; threshold = 0.35**



# Results for Cut Detection

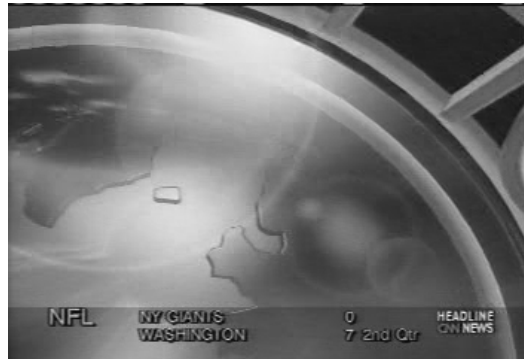
	<i>Tree Classifier</i>			<i>Sliding Window</i>			<i>Simp. Thresholding</i>		
	<b>Detect</b>	<b>FA</b>	<b>MC</b>	<b>Detect</b>	<b>FA</b>	<b>MC</b>	<b>Detect</b>	<b>FA</b>	<b>MC</b>
<b><i>soap</i></b>	0.941	13.3	0	0.916	99	0	0.852	24	0
<b><i>talk</i></b>	0.942	32.3	7.5	0.950	45	1	0.968	171	15
<b><i>sports</i></b>	0.939	82.5	34.8	0.785	59	1	0.925	251	73
<b><i>news</i></b>	0.958	38.0	0.75	0.886	61	0	0.926	212	1
<b><i>movies</i></b>	0.821	43.3	2	0.856	25	0	0.816	25	3
<b><i>cspan</i></b>	0.915	54.3	8.5	0.994	40	0	0.943	3	20

Fairly constant performance across video genres

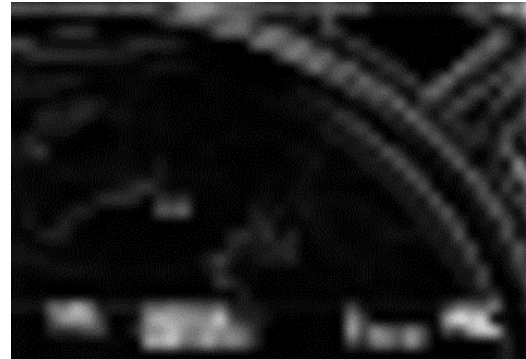


# Current Work

- Investigating the use of a feature based on the edge image obtained from the AC coefficients



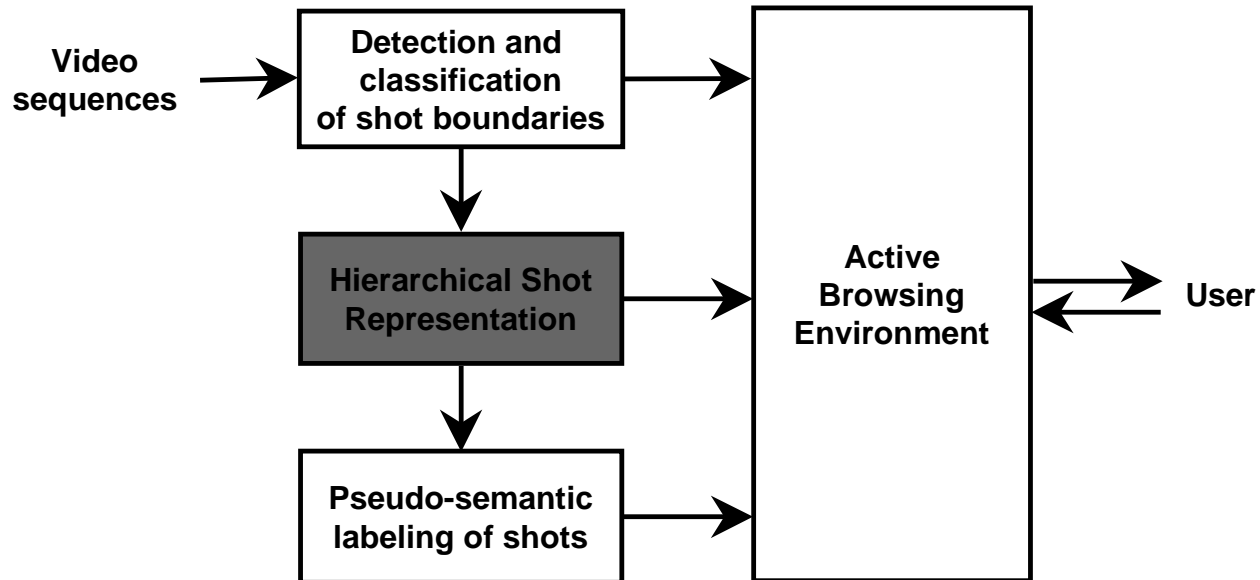
decompressed frame



edge image

- Enhancing the performance in detecting gradual transitions
- Compare with some other popular techniques



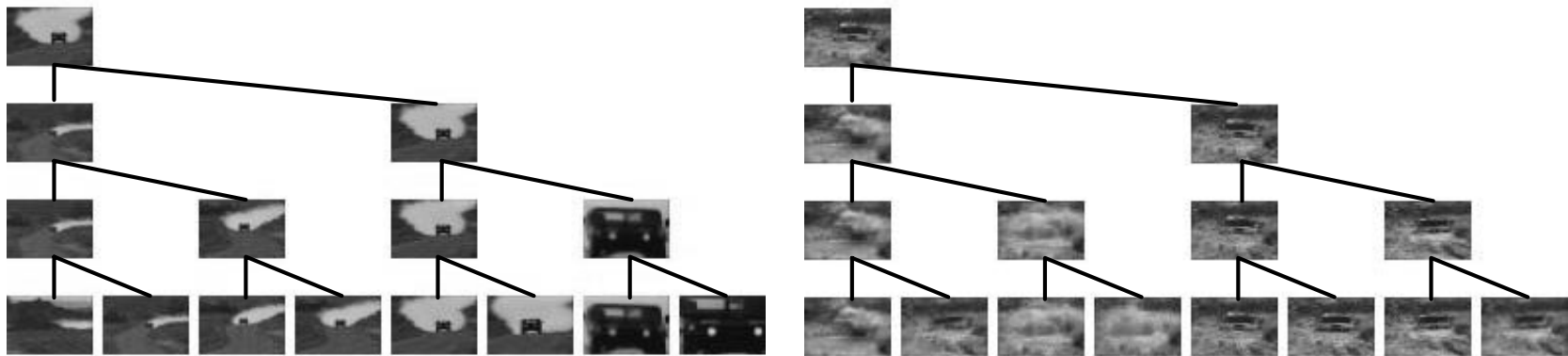


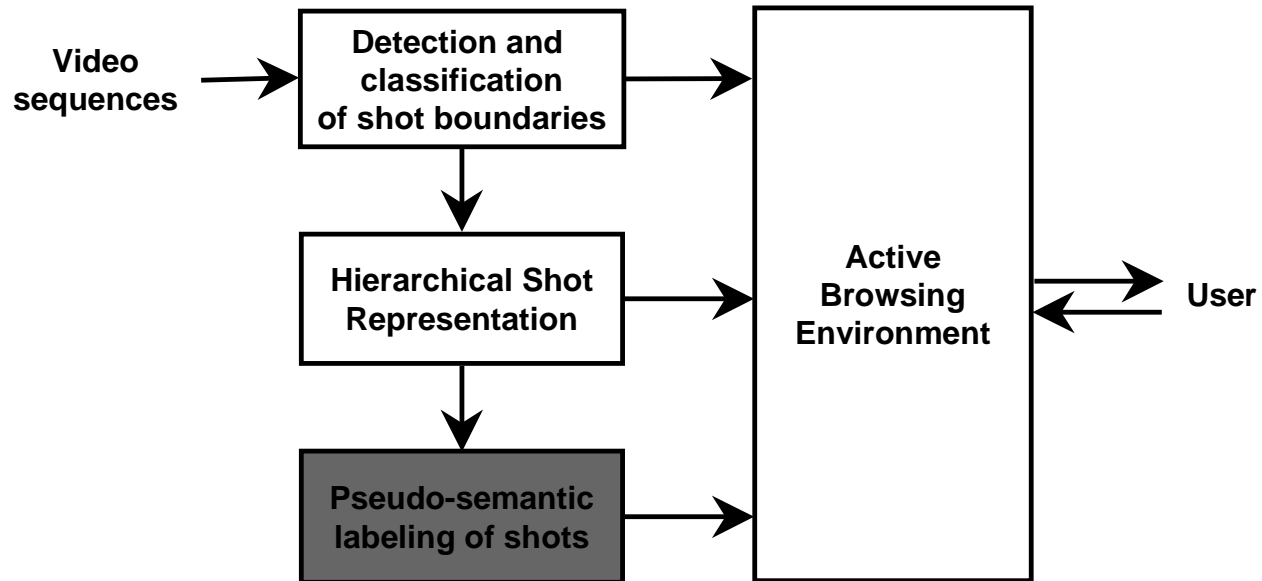
# HIERARCHICAL SHOT REPRESENTATION



# Tree Representation of Shots

- **Single keyframe is not adequate for shots with large variation**
- **Agglomerative clustering is used to build tree representation for shots**
- **211 dimensional feature vector is extracted from each DC frame containing color, texture and edge features**





# PSEUDO-SEMANTIC SHOT LABELING



# Pseudo-Semantic Labeling

*A bridge between low level and semantic description of scene content*

- **Semantic description example:**
  - *Michael Jordan doing a reverse slam dunk when pushing off on his left foot*
- **Low level description example:**
  - *Search for images with blue areas on the top and green areas on the bottom*



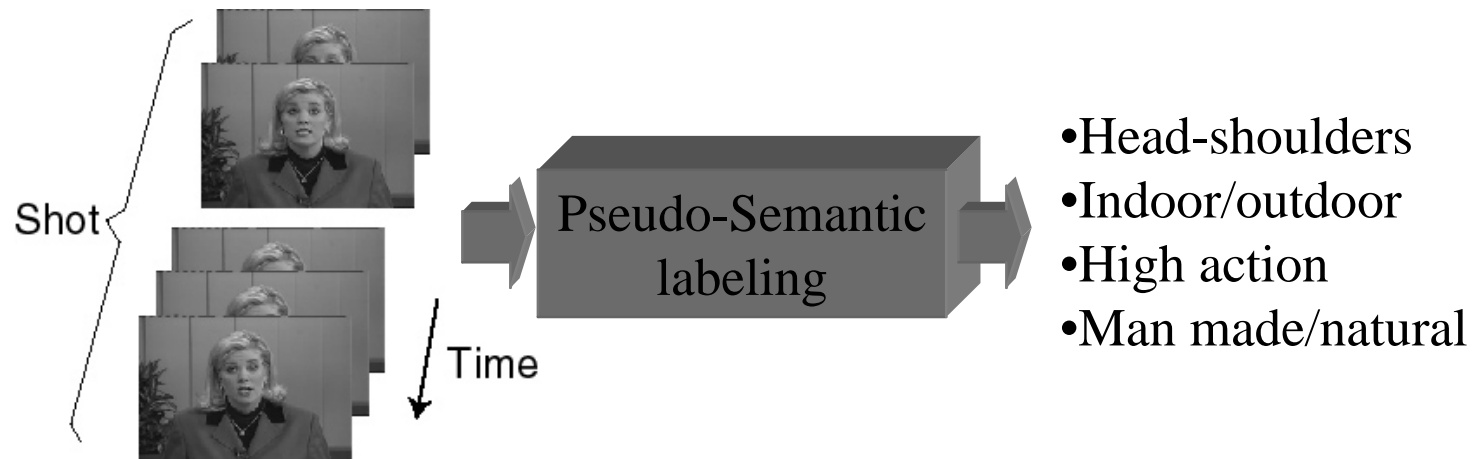
# Pseudo-Semantic Labeling Problem

- **Given a shot, derive a label using mid- and low-level features which correlates well with the high-level description of the shot**
- **Examples: head and shoulders, indoor/outdoor, high action**
- **Should use as little uncompressed information as possible**
- **Should be simple and fast**
  - **Coarse classification *without* image understanding**



# Pseudo-Semantic Label

- Based on low level and easy to derive features extract semantic information of the shot



# Head and Shoulders Feature Label

- From a shot-based point of view, we want to indicate if there is a talking head in a shot
- The first goal is to extract skin-like regions from each frame
- With motion and texture information, each region along the shot will be labeled as a face candidate or not



# Skin Detection

- **Extracts regions which potentially correspond to face regions based on color**
- **Skin and no skin classes are modeled using normalized histograms in the YCbCr color space.**
- **Neyman-Pearson test is used to classify each pixel into the skin and no skin classes**



# Skin Detection Examples



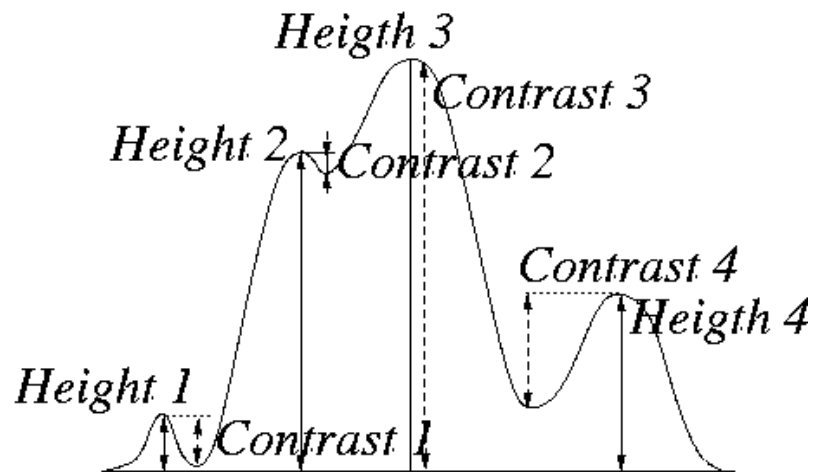
# Unsupervised Segmentation

- Skin detection produces non-homogeneous regions containing more than one object
- Unsupervised segmentation is used to segment the skin detected areas into homogeneous regions



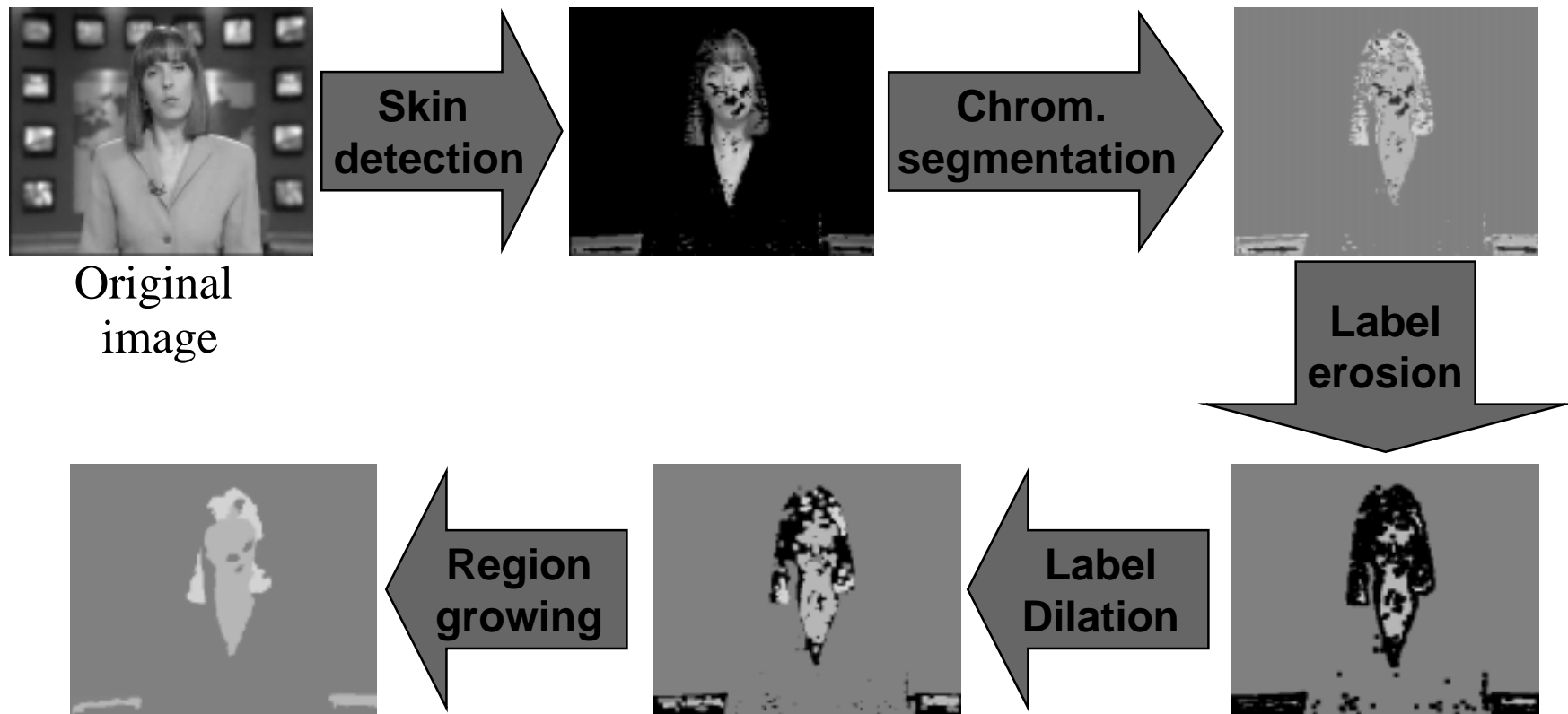
# Unsupervised Segmentation Using Chrominance

- The color space is clustered using the CbCr histogram of the skin detected pixels
- The histogram is treated as a gray scale image and then the watershed algorithm is used to cluster it



Markers for watershed  
are local maxima with  
high normalized contrast

# Example of Unsupervised Segmentation Using Chrominance

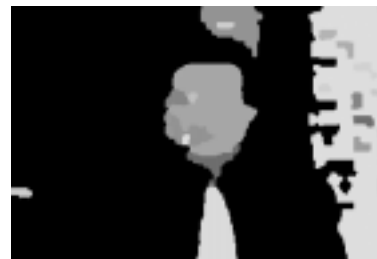


# Unsupervised Segmentation Using Luminance

- After segmentation using chrominance, each class is resegmented using the luminance information
- For each class the luminance histogram is clustered and again morphological filtering and region growing is used



# Unsupervised Segmentation Examples



# Region Merging

- **Unsupervised segmentation is likely to partition face areas in various connected regions**
- **Regions will be merged in a pair-wise way**



# Region Extraction and Characterization

- **Connected regions are extracted and feature vectors are computed for all of them**
- **Each feature describes a characteristic regarding, texture or color of the region**



# Face Extraction Results



# Face Recognition Results

	<b>Shots</b>	<b>Faces</b>	<b>Detect (%)</b>	<b>FA(%)</b>	<b>Correct(%)</b>
<i>news1</i>	231	76	73.7	16.1	80.5
<i>news2</i>	72	29	93.1	23.3	83.3
<i>news3</i>	78	33	87.9	15.6	85.9
<i>news4</i>	103	42	90.5	13.1	88.3
<i>news5</i>	188	51	76.5	13.9	83.5
<i>movie</i>	142	92	84.8	28.0	80.3
<i>drama</i>	100	90	94.4	20.0	93.0
<b>total</b>	<b>914</b>	<b>413</b>	<b>85.2</b>	<b>17.0</b>	<b>84.0</b>



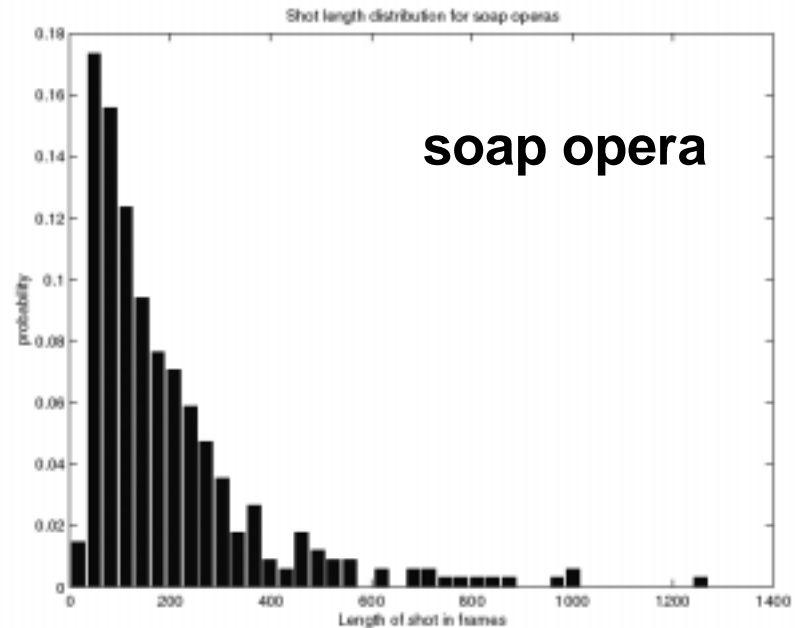
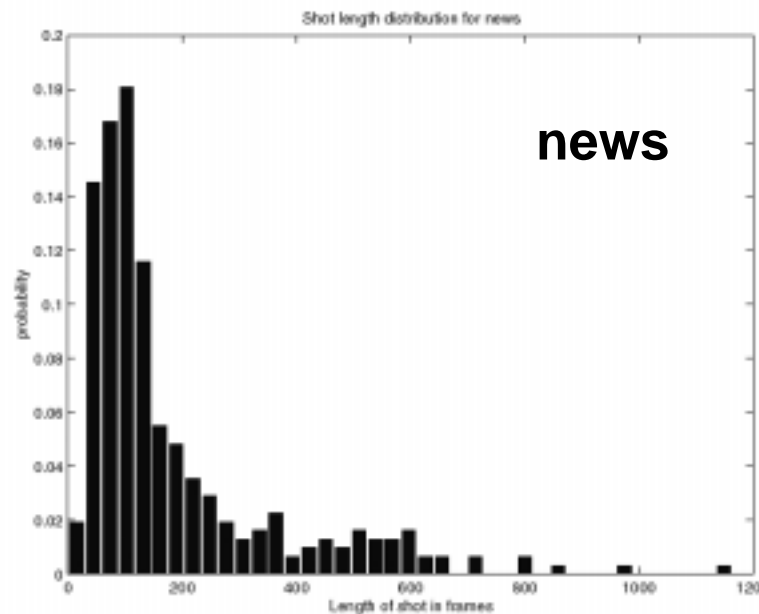
# **“Indoor/Outdoor” Feature Label**

- **Extract dominant orientation from texture (Gorkani and Picard 1994)**
- **Hidden Markov Models on image blocks (Yu and Wolf 1995)**



# Shot Length Feature

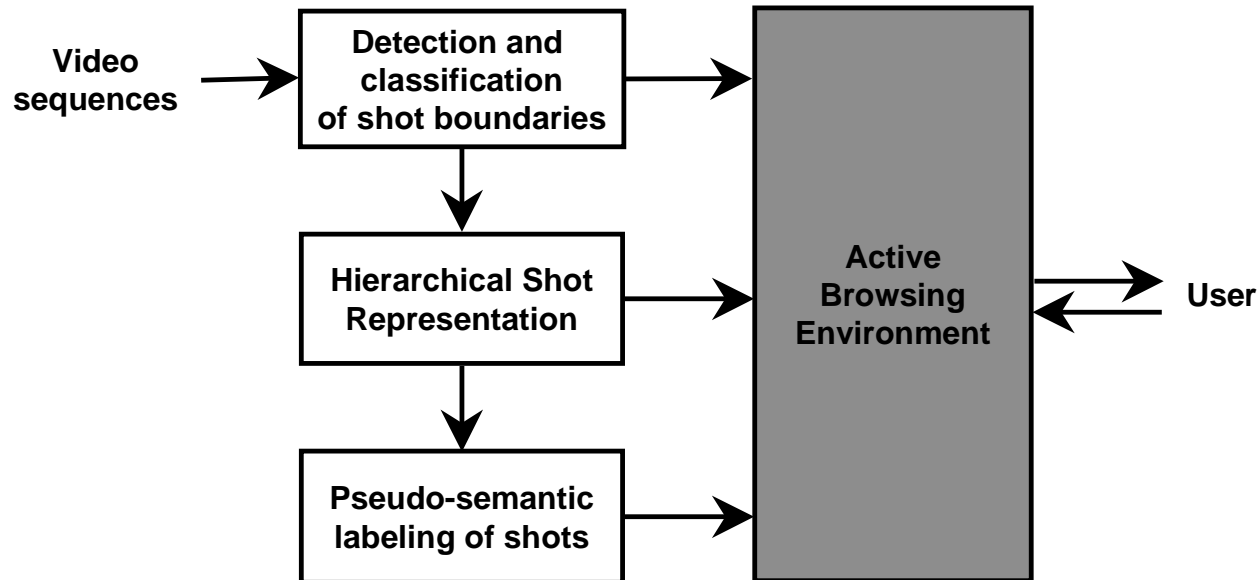
- Shot boundaries are “man-made” according to editing rules
- Shot length is an indication of editing pattern
- Shot length distributions for different genres are different



# Current Work

- Investigate the feasibility of deriving the “indoor/outdoor” label from compressed data
- Find a suitable measure of motion to be used in deriving the “high-motion” label
- Try to increase the performance of the “head and shoulders” label classifier



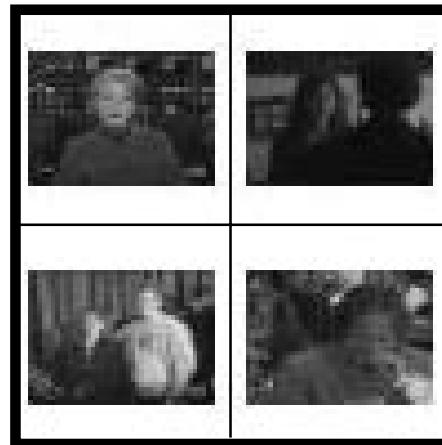
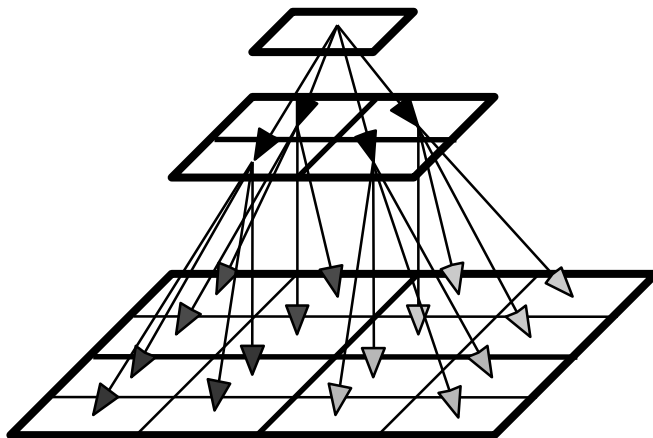


# BROWSING AND SEARCHING ENVIRONMENT

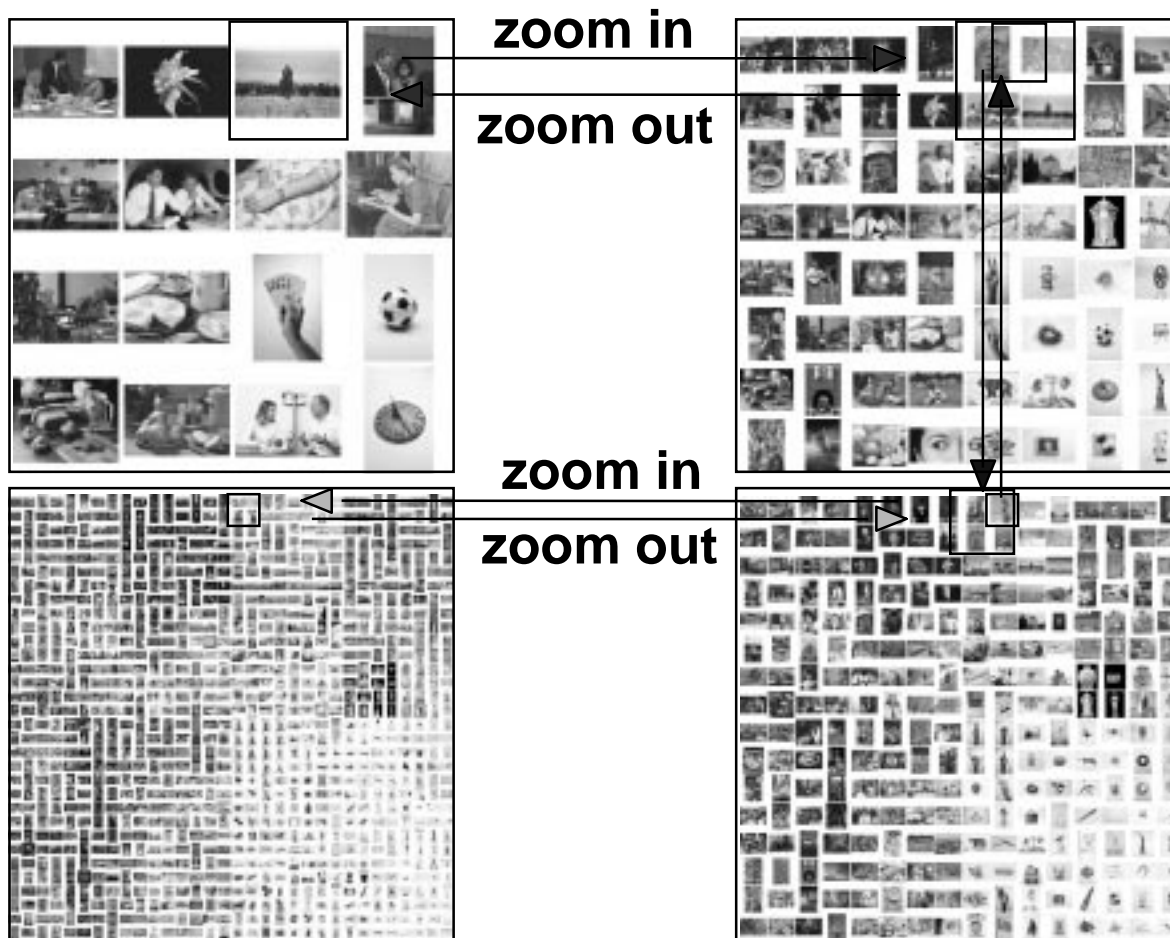


# Browsing with a Similarity Pyramid

- Organize database in a pyramid structure
  - Top level of pyramid represents global variations
  - Bottom level of pyramid represents individual images
- Spatial arrangement makes similar images neighbors
- Embedded hierarchical tree structure



# Navigation via the Similarity Pyramid



# Browser Interface



Similarity Pyramid

Control Panel Relevance Set



# Relevance Feedback

- **Previous research:**
  - search-by-query only
  - iterative update of the dissimilarity function
- **Our method : Use relevance feedback to**
  - prune database using cross validation method
  - reorganize database based on optimized dissimilarity function



# Database Reorganization Based on the Relevance Set

- Distance function,  $D_{\hat{\epsilon}}(s_i, s_j)$ , between shots is parametrized by the feature vector  $\theta$
- $D_{\hat{\epsilon}}(s_i, s_j)$  contains the distances based on the shot tree, temporal position, motion, and pseudo-semantic features
- Search for the feature vector,  $\theta$ , that maximizes the separability of the shots in relevance set from the ones in the rest of the database
- Conjugate gradient optimization is used



# **Video Genre Classification Using the Pseudo-Semantic Trace**

*IMA March 1, 2001*

Slide 82



# Using Hidden Markov Models to Analyze Video Sequences

- Distribution of shot length indicates the editing pattern used in the sequence
- Different genres have different editing patterns.
- Previous work
  - Using HMMs on audio for genre classification
  - Using shot labels like *medium shot* to detect dialogues



# The Pseudo-Semantic Trace

- We have used two features for these experiments
- The components of the pseudo-semantic feature vector for the  $n^{\text{th}}$  shot with length  $L_n$  frames are given by

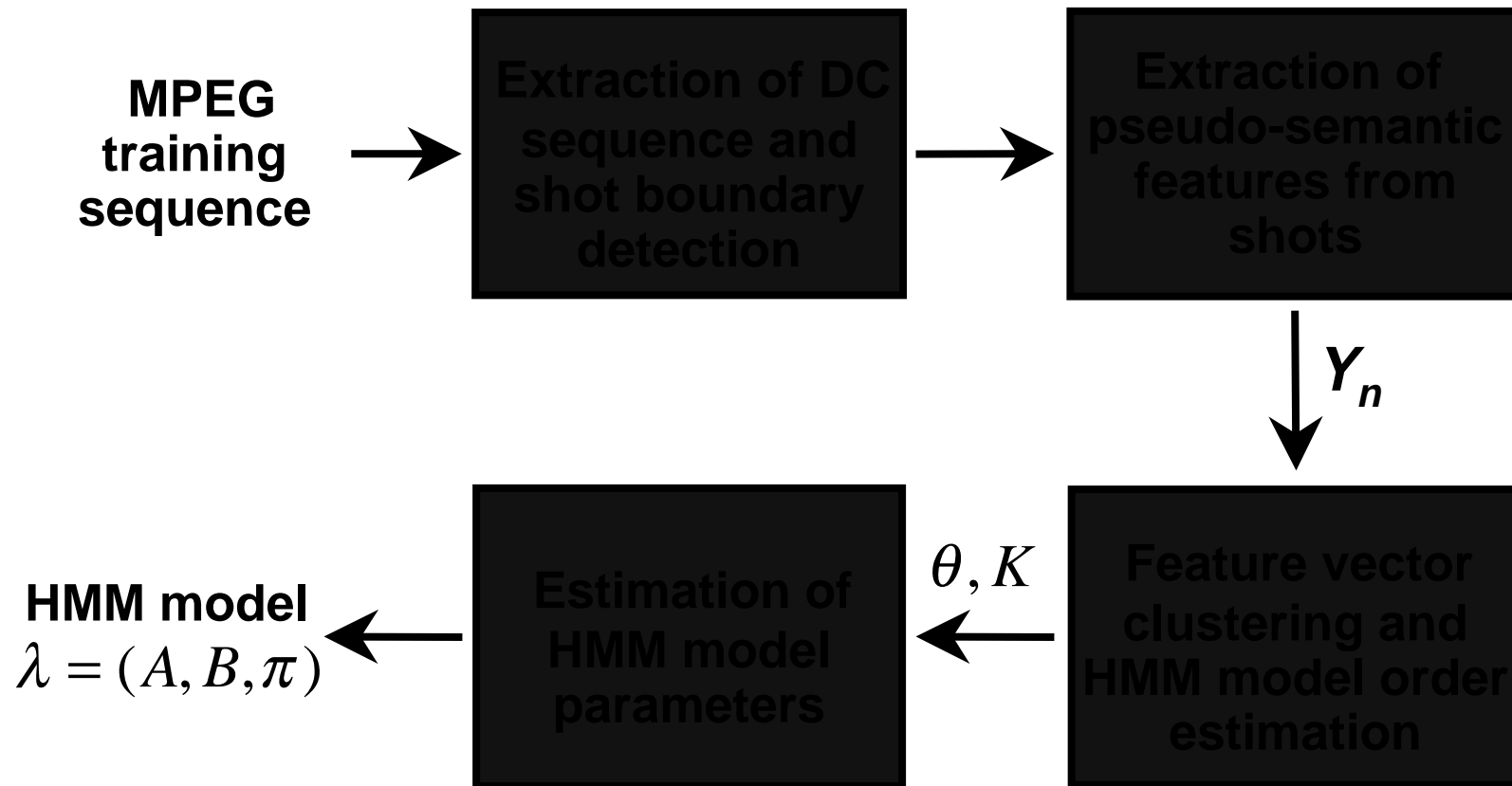
$$Y_{n1} = \frac{\sum_{k \in s_n} m_k}{L_n} \quad \text{and} \quad Y_{n2} = L_n$$

$m_k = (\text{\#forw. MB}) + (\text{\#back. MB}) + 2(\text{\#forw.-back. MB})$

- We call the sequence of feature vectors for a sequence the pseudo-semantic trace



# HMM Training Procedure for Genres



# Clustering Feature Vectors Using a Gaussian Mixture Model

- The feature vectors in the pseudo-semantic trace are clustered using a Gaussian mixture model

$$\log p_y(y|K, \theta) = \sum_{n=1}^N \log \left( \sum_{k=1}^K p_{y_n|x_n}(y_n|k, \theta) \pi_k \right)$$

- Maximum likelihood is used to estimate mixture parameters

$$\hat{\theta}_{ML} = \arg \max_{\theta} \log p_y(y|K, \theta)$$

- The number of clusters =  $K$  = HMM model order is estimated by minimizing the minimum description length criterion

$$MDL(K, \theta) = -\log p_y(y|K, \theta) + \frac{1}{2} P \log(NM)$$



# Building Hidden Markov Models for Video Genres

- After clustering, the symbol sequence corresponding to each shot is determined using the ML estimate

$$v_n = \arg \max_{k \in [1, K]} p_{y_n | x_n}(y_n | k, \theta)$$

- The symbol sequences are then used to train an ergodic HMM for each video genre



# Classification of Sequences

- For a sequence  $S$ , first the symbol sequence is estimated
- Then the model symbol probabilities,  $p(v_1, \dots, v_n | \lambda_i)$ , are estimated for each of the  $L$  genre models using the forward-backward procedure
- The given sequence is then classified to a genre class using the ML rule

$$\text{genre of } S = \arg \max_{i \in [1, L]} p(v_1, \dots, v_n | \lambda_i)$$



# Distance Between Genre Models

- Two HMMs may look very different but may be statistically very similar
- Distance between two HMMs is defined as

$$D(\lambda_i, \lambda_j) = \frac{1}{T} \left[ \log P(O^{(i)} | \lambda_i) - \log P(O^{(i)} | \lambda_j) \right]$$

$O^{(i)} = O_1 O_2 \dots O_T$  are observations generated by  $i^{th}$  model

- We can define a *symmetric* distance as

$$d(\lambda_i, \lambda_j) = \frac{D(\lambda_i, \lambda_j) + D(\lambda_j, \lambda_i)}{2}$$



# Results: Model Distances

- Distances were computed using symbol sequences of length 5000 generated using HMM models

	soap	talk	sports	cspan
soap	0	2.033	1.788	3.991
talk		0	2.094	3.241
sports			0	3.424
cspan				0



# Results: Genre Classification

- HMMs of order 6 were used in the classification

## *Classifier Output*

<i>True Label</i>	<b>soap</b>	<b>talk</b>	<b>sports</b>	<b>cspan</b>
<i>soap</i>	0.583	0.250	0.167	0
<i>talk</i>	0	0.833	0.167	0
<i>sports</i>	0.333	0.083	0.583	0
<i>cspan</i>	0	0	0.083	0.917



# Future Research

- Investigate the use of genre profiles to classify video sequences
  - video grammar
- Add UPC face identification module
- Use information from the closed-caption signal and captions to extract content for shots
- Use audio channel
- Use watermarking for feature binding
- Internet delivery

