Chapter 14

Multimedia Information Retrieval

The Challenges
Content-based Image Retrieval
Retrieving and Browsing Video
Fusion Models: Combining it All
Segmentation
Compression and MPEG Standards

What is Multimedia?

- We face an ever-growing mountain of digital data
  - sharing through cable, satellite, mobile phones
  - uploading through personal cameras, laptops, and mobile phones
  - trend accelerated by mobile phones with cameras
- Need to develop better management methods and tools for all this multimedia data
  - **Multimedia** is essentially any digital data, including plain text, mostly unstructured, that we use to communicate or capture information

Multimedia IR

- Most general form of the multimedia retrieval problem
  - The retrieval of text, image, video and sound data related to the interest of the user and their ranking according to a similarity degree
- Similarity degree should be computed to improve likelihood that user will find answers relevant
- For searching, user could describe a scene in video by typing

```
Keanu Reeves avoiding bullets in a
helicopter crash in the movie The Matrix
```

Multimedia IR

- Multimedia information retrieval (MMIR) encompasses different sub-areas
  - content representation and multimedia object representation
  - feature extraction
  - query formulation to map high-level semantic concepts into low-level features
  - query-by-example
  - relevance feedback, interactive queries
  - efficient feature indexing and cataloguing
  - integrated searching and browsing
  - techniques for searching multimedia based on their contents

Text IR versus Multimedia IR

- Several aspects that make text retrieval different from image, audio or video retrieval
  - in text, words are readily available as basic units and structure is provided by punctuation and paragraphs
  - in contrast, multimedia data is typically an uninterrupted stream, a linear story with few delimiters
- For non-text media, defining the semantic unit is a fundamental step to attain high-quality search
- In video, for instance, time is important—content changes with time

Text IR versus Multimedia IR

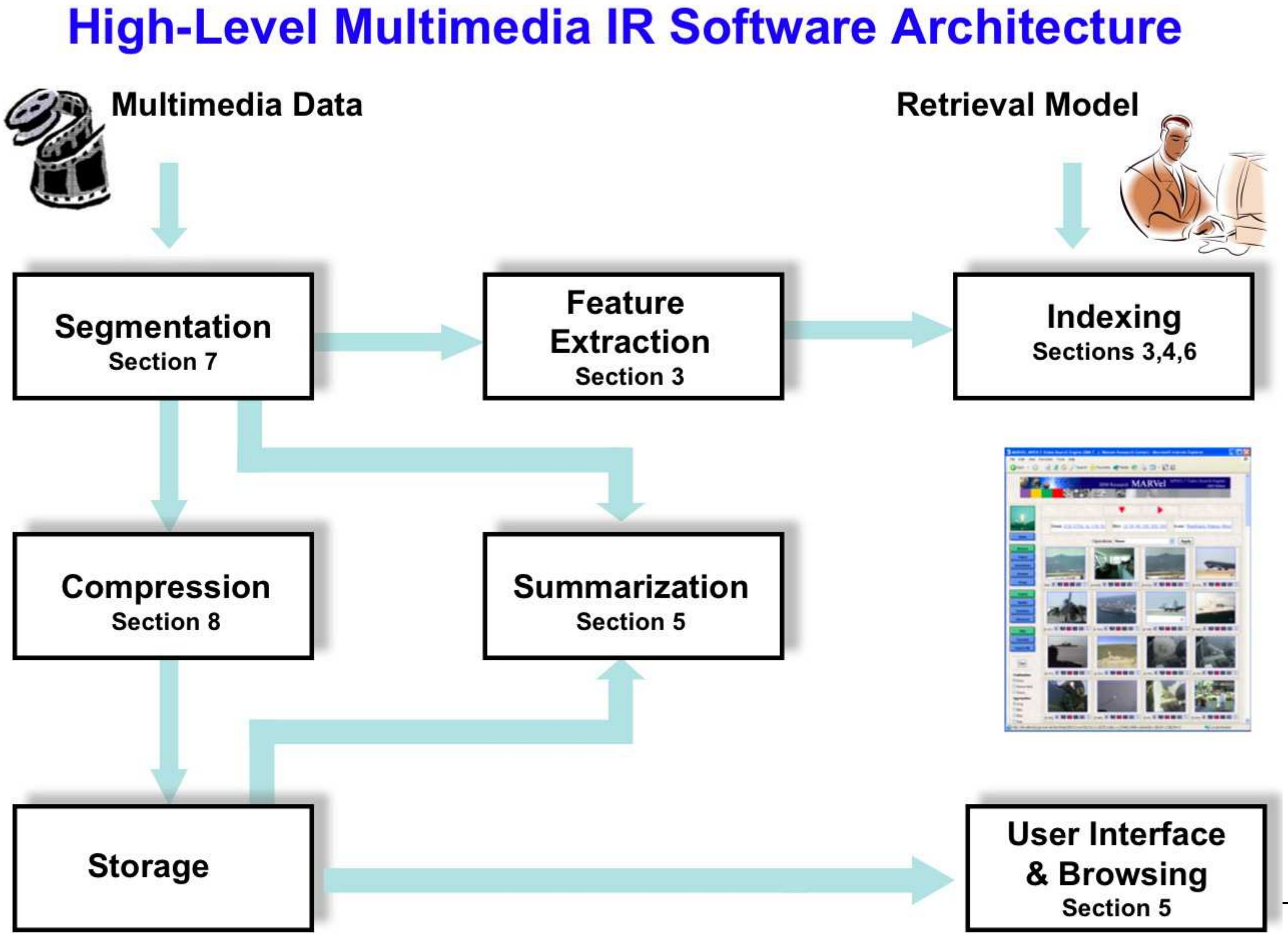
- Advances in speech recognition allow the generation of good quality speech transcripts
- However, even a good transcript lacks punctuation, paragraphs, and all the elements that provide structure
- Although retrieval based on a speech transcript seems very close to text retrieval, in practice it is not
    - time associated with every word in the speech transcript can be a valuable information for dealing with this problem

Text IR versus Multimedia IR

- Sheer differences in sizes of text documents and multimedia objects
  - 75-minute audio signal compressed in MP3: 60M bytes
- We have a strong technological culture around words
  - concepts of summarizing and highlighting are much better understood for text
  - for multimedia there is no canonical or universally agreed notion of what a summary is
- Multimedia retrieval is a relatively new discipline
- Even though, growth of image and video search engines is here to stay

Text IR versus Multimedia IR

- Information flow in a multimedia retrieval system



High-Level Multimedia IR Software Architecture

Multimedia Data

Retrieval Model

Segmentation
Section 7

Feature Extraction
Section 3

Indexing
Sections 3,4,6

Compression
Section 8

Summarization
Section 5

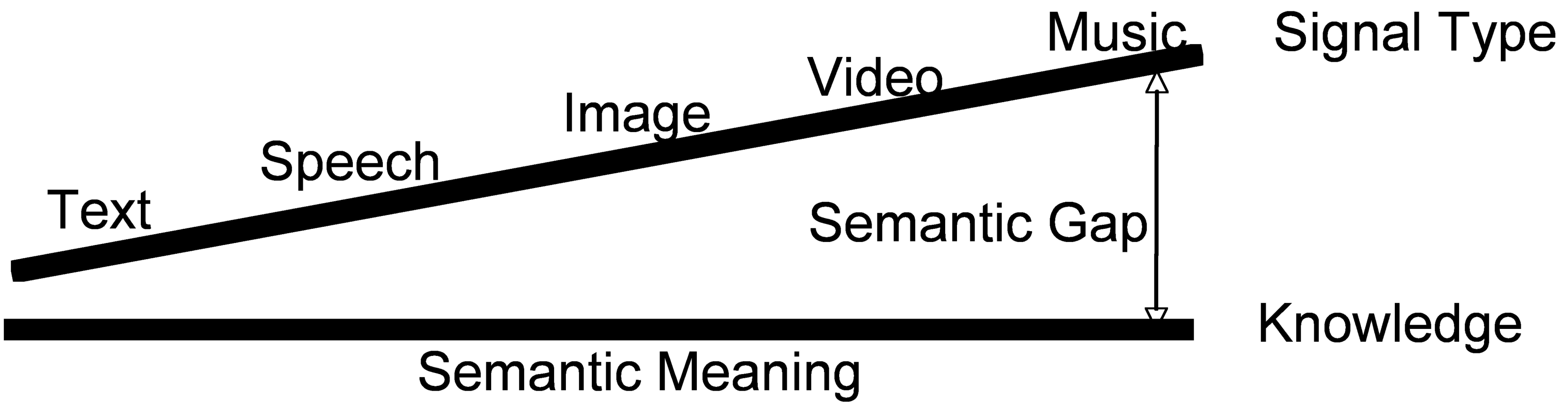
Storage

User Interface & Browsing
Section 5

The Challenges

The Semantic Gap

- Large gap between contents of a multimedia signal and its meaning
- Usually referred to as the **semantic gap**



The Semantic Gap

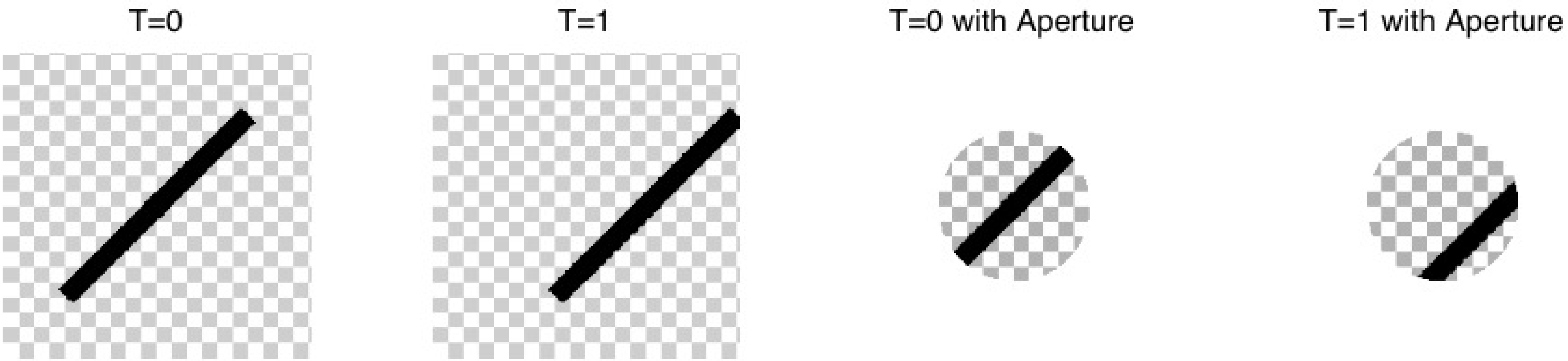
- **Object recognition:** hard problem in image and audio processing
  - humans can look at an image and identify faces and objects
  - automatically labeling components of an image or analyzing the sounds in a waveform are unsolved problems
- Multimedia IR systems make heavy use of human-generated words
  - almost ignore the content features to generate an answer for the user

The Semantic Gap

- Image or audio signal carry subjective and emotional interpretations
  - difficult for computers to reproduce
  - in speech, non-semantic information conveyed by the prosody of the signal
  - prosody allows distinguishing between “don’t stop" and “Don’t! Stop!"

Feature Ambiguity

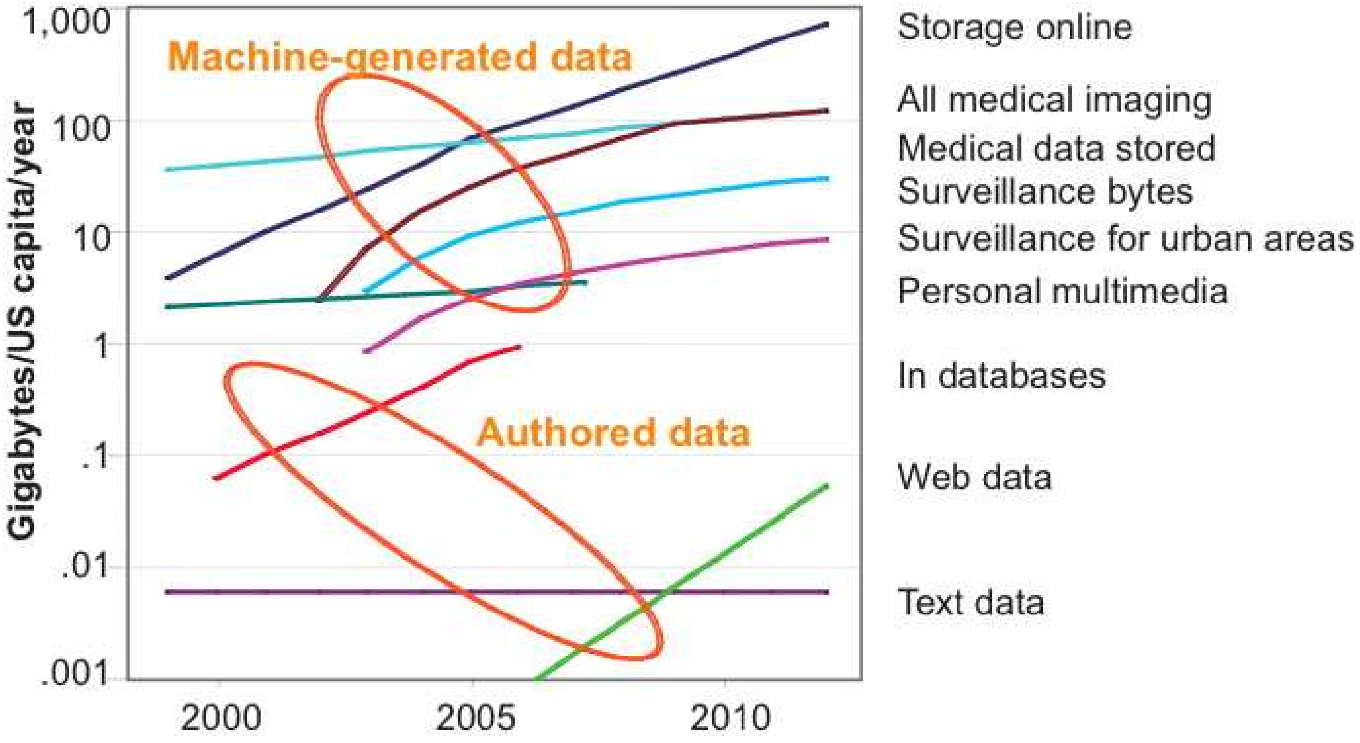
- Aperture problem



  - bar is moving to the right, which cannot be properly interpreted with aperture
  - for efficiency, simple motion detector only measures portion of the image—**the aperture**
  - aperture limits decision to small portion of image
  - lack of global information on image makes interpretation difficult

Machine-generated Data

- Growth of data is big challenge



Content-based Image Retrieval

Content-based Image Retrieval

- Idea: identify and extract features related to image contents
- The problem: **content-based image retrieval** is the task of retrieving images based on their contents
- Query-by-example (QBE)
  - user supplies an image and the system finds other images that are similar to it
  - ignores semantic information associated with images
- Best ranking functions based on image properties that are not affected by variables
  - pose, camera focal length and focus, lighting, camera viewpoint, and motion

Color-Based Retrieval

- Common QBE solution: feature summaries across entire image
  - average color: treat color as a global feature
  - does not depend on image resolution
    - even though, location of colors is very relevant
  - compare color histograms of different pictures
    - colors are quantized into one of N bins
    - number of pixels in each bin are compared

Color-Based Retrieval

- Color histogram is independent of image resolution and viewing angle
- No need to perform foreground–background segmentation
- Histogram of color c_i in image I is defined as

$$h_I(c_i) = P(color(p) = c_i | p \in I)$$

  - $P(color(p) = c_i | p \in I)$: probability that pixel p randomly selected from image I has color c_i

Color-Based Retrieval

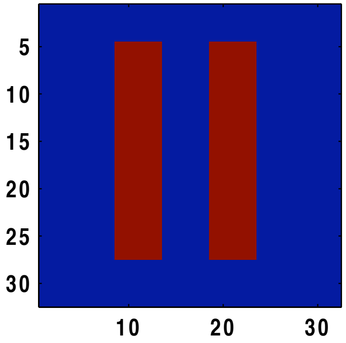
- To improve color histogram include information on relative locations of each color
- Build color autocorrelogram by counting pixels

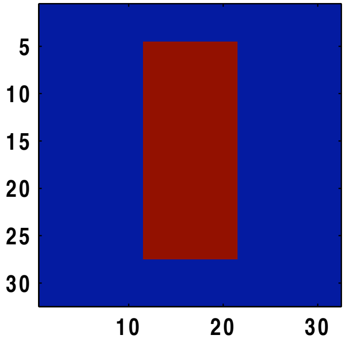
$$h_I(c_i, c_j, r) = P(color(p_1) = c_i \wedge color(p_2) = c_j | r = d(p_1 - p_2))$$

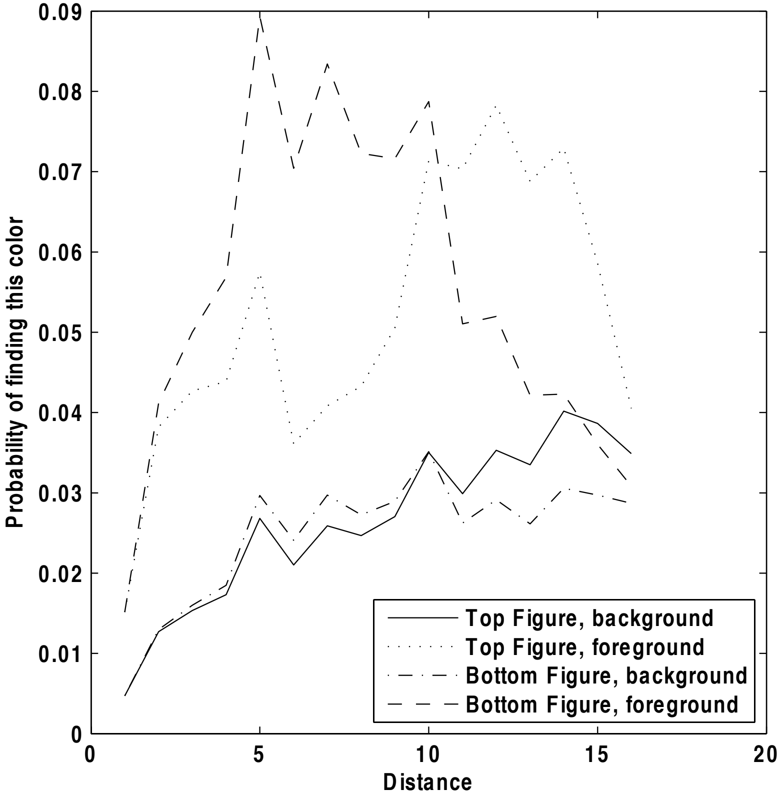
  - (c_i, c_j): color pair
  - $d(p_1 - p_2)$: distance between two pixels p_1 and p_2
  - pixels p_1 and p_2 randomly selected from image I

Color-Based Retrieval

- Large differences between autocorrelograms of distinct images that have identical color histograms

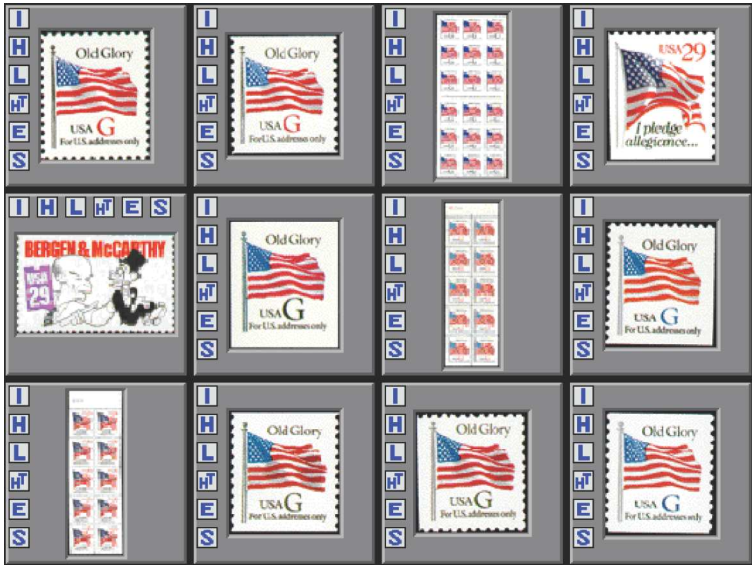


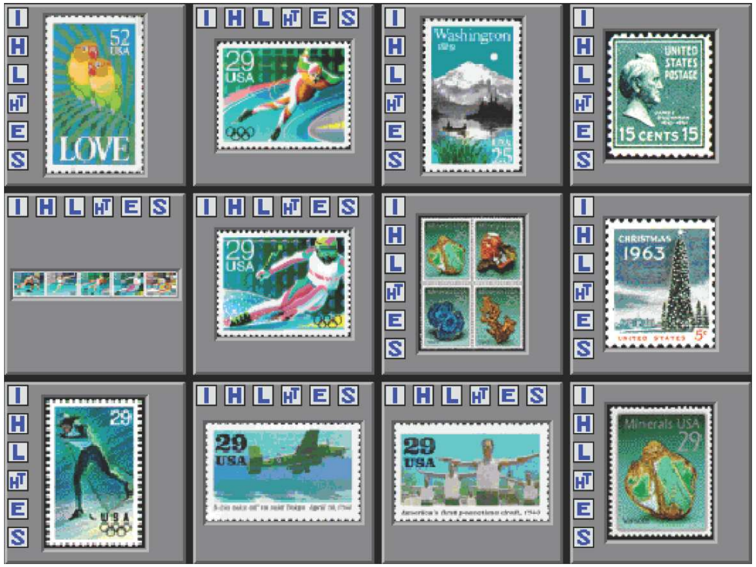




Example 1

- Example of application of the technique





Example 1

- **Color constancy:** perceptual property associated with a color
  - problem with retrieval based on color histograms
  - human viewers recognize color of an object, almost without regard to incident light
  - an apple looks red, either in daylight or under indoor light
  - humans are good at perceiving the same colors
  - a color histogram is not so forgiving

Texture

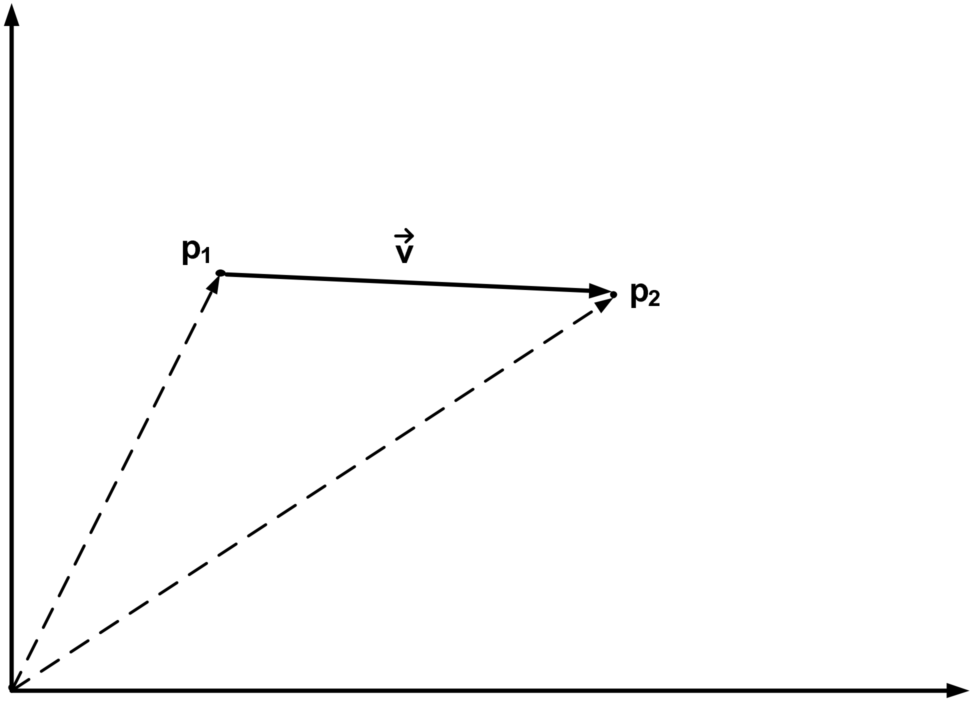
- **Texture:** a measure of the repetitive elements in the image
- A perceptual phenomenon, easily detected by humans
- Challenging to describe mathematically
- Characterizes the repeating patterns of image intensity that are too fine to be distinguished as separate objects
- Most texture measures are invariant to intensity and orientation

Co-occurrence Texture Measures

- Simplest texture measure: uses co-occurrence matrix called gray-level co-occurrence matrix (GLCM)
- Summarizes information about patterns of light in pairs of image pixels
- Pixel pairs to use are determined using a vector $\vec{v}$
  - pixel pair (p_1, p_2) that has directionality and distance determined by $\vec{v}$ is said to be $\vec{v}$-aligned
  - establishes directionality and distance for pixels in each pair

Co-occurrence Texture Measures

- Given a vector $\vec{v}$, pixel pairs $[p_1, p_2]$ for which $\vec{p_1} - \vec{p_2} = \vec{v}$ are said to be $\vec{v}$-aligned
- $\vec{v}$-aligned pixels are considered in GLCM matrix computations



Co-occurrence Texture Measures

- $P_I(c_i, c_j, \vec{v})$: probability of finding $\vec{v}$-aligned pixel pairs in image I, associated with colors c_i and c_j

$$P_I(c_i, c_j, \vec{v}) = P(color(p_1) = c_i, color(p_2) = c_j | \vec{p}_2 - \vec{p}_1 = \vec{v})$$

- Statistics to summarize information in a GLCM
  - energy, entropy, contrast, and homogeneity

Co-occurrence Texture Measures

- **Energy:** measure of brightness of $\vec{v}$-aligned pixels

$$\mathcal{E}_I(c_i, c_j, \vec{v}) = \sum_i \sum_j P_I(c_i, c_j, \vec{v})^2$$

- **Entropy:** measure of non-uniformity of $\vec{v}$-aligned pixels

$$\Psi_I(c_i, c_j, \vec{v}) = \sum_i \sum_j P_I(c_i, c_j, \vec{v}) \log P_I(c_i, c_j, \vec{v})$$

- **Contrast:** measure of differences between pixel light intensities ϕ_i of pixels in $\vec{v}$-aligned pairs

$$\mathcal{C}_I(c_i, c_j, \vec{v}) = \sum_i \sum_j (\phi_i - \phi_j)^2 \;\; P_I(c_i, c_j, \vec{v})$$

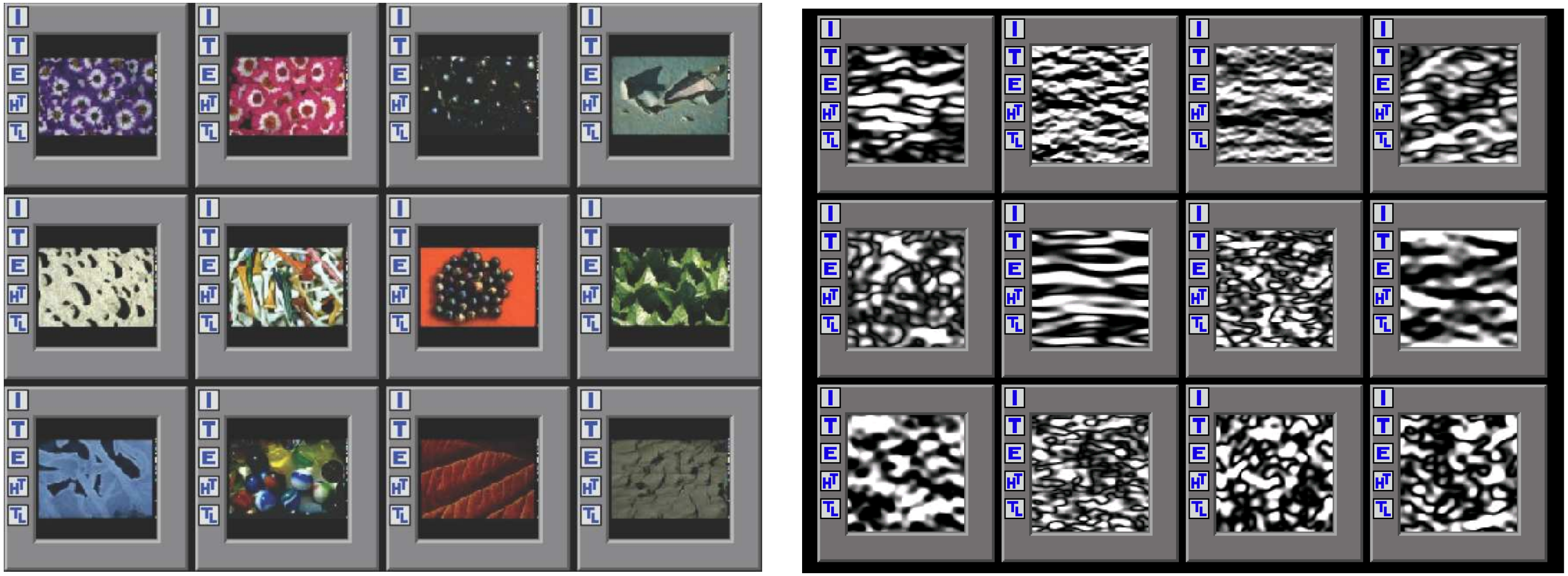
Co-occurrence Texture Measures

- **Homogeneity:** measure of similarity of pixels

$$\mathcal{H}_I(c_i, c_j, \vec{v}) = \sum_i \sum_j \frac{P_I(c_i, c_j, \vec{v})}{1 + |\phi_i - \phi_j|}$$

Example 2

- Example of texture retrieval using the QBIC system



Salient Points

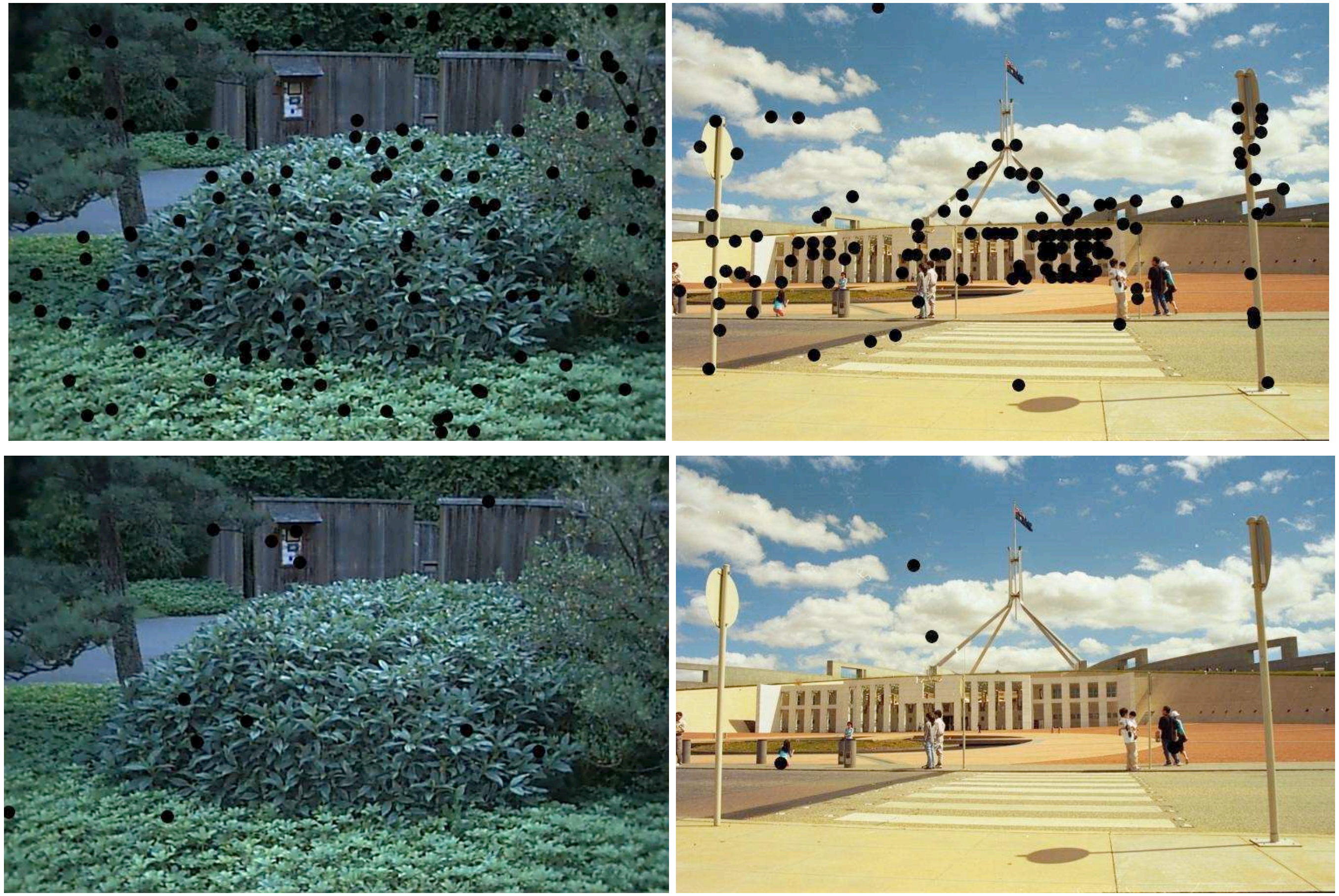
- Algorithms for color and texture-based retrieval use histograms over entire image
- More sophisticated approach builds a feature model
  - combine color and spatial frequency information, at “interesting” image regions
- Analyze image looking for points that are especially distinct

Salient Points

- **Salient points**: technique that finds image features that are persistent across a number of scales
- Especially robust to changes in lighting, position of camera, and even object’s angle
- Typical operations at salient points include key points, stable orientation, and local geometry on texture

Salient Points

- Salient points tend to be associated with corners or distinct places in the image

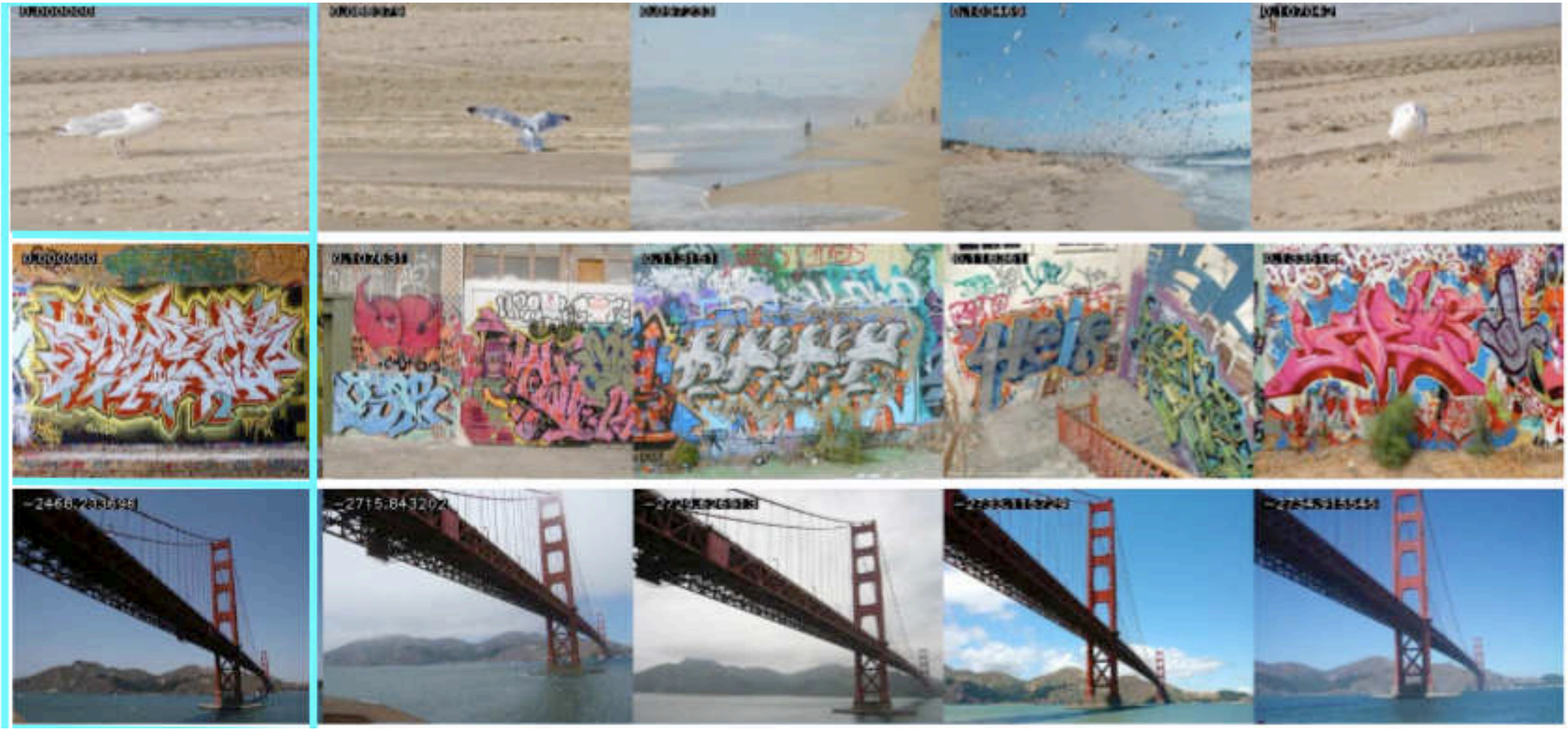


Example 3

- Image similarity computed by summarizing statistics of salient points
- Image characteristics near salient point defined by simple spectral filters
- Values are clustered using k-means to determine “words” in the language
- Algorithm like pLSA used for image matching

Example 3

- Results produced content retrieval using salient points



Audio and Music Retrieval

The Problem

- The audio retrieval problem

  The retrieval of audio tracks that match a vaguely specified audio-information need.

- This problem takes many forms such as:
  - **fingerprinting:** given a small snippet of sound, find an audio object that matches it
  - **speech recognition:** given an audio track, recognize the text it contains
  - **speaker identification:** given an audio track, recognize the speaker(s) it contains
  - **spoken document retrieval:** given a text query, retrieve spoken documents that match the query

Fingerprinting

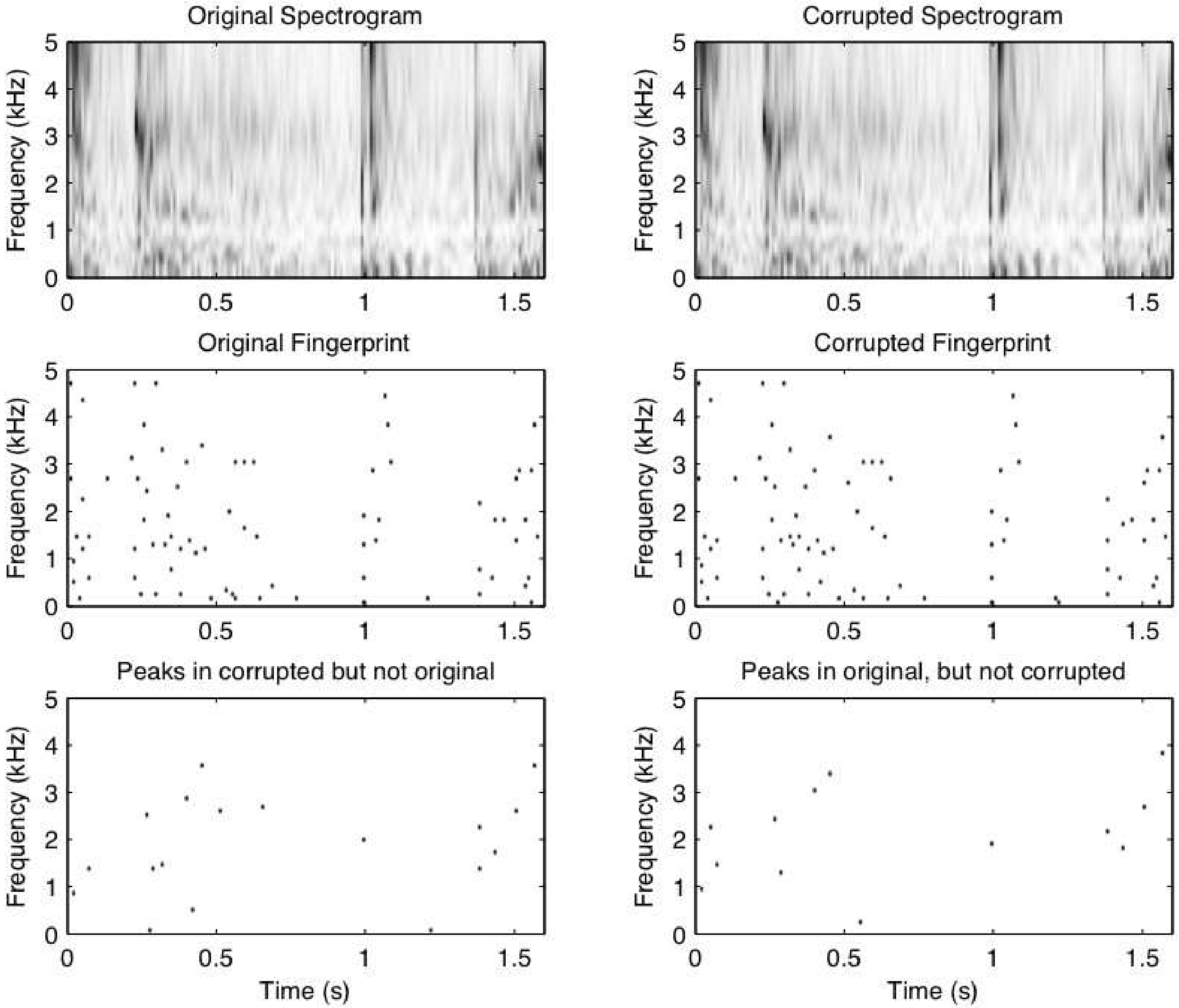
- Audio fingerprinting is a commercially successful IR task
- Use a small snippet of sound to query a large database and look for an exact match
- Process is complicated because the query is often corrupted
  - Typical case: snippet of sound captured by a cell phone in the noisy environment of a pub

Fingerprinting

- One solution approach:
  - look for changes in the spectrogram
  - encode the most salient portions of the audio
  - **spectrogram:** spectral-temporal distribution of sound
- Difficulty:
  - make process robust to common abuses in audio signals
  - loud background noise, inexpensive microphones on a cell phone, and compression algorithms optimized for voice and not music
- Location of a peak relatively stable even when noise added
- Constellation of peaks constitutes a **fingerprint** that can be used to identify a section of the audio piece

Fingerprinting

- Fingerprinting using Madonna’s song “Borderline”



Speech Recognition

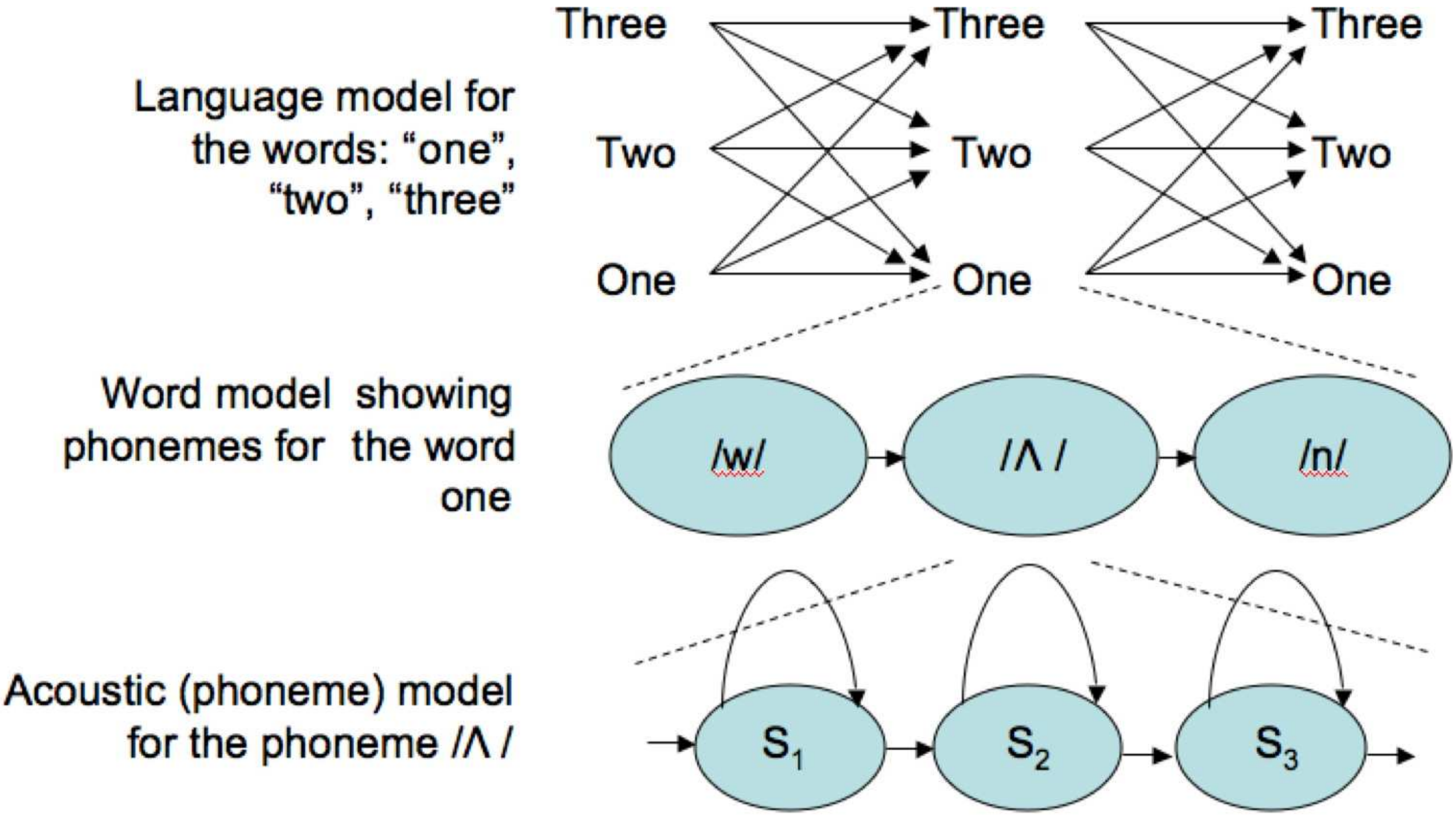
- Recognize the words contained on audio track
- Works well when two conditions are met:
  - constrained acoustic environment: single voice and no background noise or music
  - well defined task: limited number of words need to be recognized at any point in time
- Unfortunately, multimedia signals usually violate both conditions

Hidden Markov Models

- Used to find sequence of word models that best explain the audio
- Include information on the legal phoneme sequences and their pronunciation
- All information is tied together within a single probabilistic framework
- Model estimates probability that a set of phonemes, corresponding to a word, sounds like what was heard

Hidden Markov Models

- Simple HMM illustrating these two models



Language model for the words: “one”, “two”, “three”

Word model showing phonemes for the word one

Acoustic (phoneme) model for the phoneme /Λ /

Hidden Markov Models

- Speech signal as a sequence of static states
  - the signal is assumed to be constant and when it changes the HMM moves to a new state
- Each state models a portion of a speech signal with a probabilistic density function
- To handle the dynamics of speech, each acoustic model is composed of three to five states
- Each state describes the likely MFCC (mel-frequency cepstral coefficient) vectors with a Gaussian Mixture Model (GMM)

Gaussian Mixture Models

- Many ways to pronounce the phoneme /a/ in the word cat
  - to handle this, use a GMM for each phoneme
- A GMM is a probability density function modeled with a small number of Gaussian bumps, which in this case lead to a 39-D space

Gaussian Mixture Models

- Basic form of this multidimensional Gaussian model

$$G(x,\mu,\Sigma) = \frac{1}{(2\pi)^{N/2}\left|\Sigma\right|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$$

where

  - x is the N-dimensional data point
  - μ is the location of the Gaussian mean
  - $(\cdot)^T$ represents matrix transpose
  - Σ is a matrix that describes the covariance of the data

Gaussian Mixture Models

- We create a mixture of these Gaussians by adding a number of them
- Each component represents the probability of a different portion of the acoustic space

$$GMM(x, \{\mu\}, \{\Sigma\}) = \sum_i A_i G_i(x, \mu_i, \Sigma_i),$$

where

  - G_i is a single multidimensional Gaussian
  - A_i is a weighting coefficient

- Generally these covariance matrices are diagonal

Language and Acoustic Models

- Language model makes speech recognition work by constraining number of possible words, which greatly reduces chances of mistakes
  - very simple solution: only words that are allowed are the ten digits
  - in this case, we say that the vocabulary size is 10 and that the language has a **perplexity** of 10
- Typical large-vocabulary speech-recognition systems have a perplexity of 60
- *Speech recognition* works well even over lousy communication channels such as a cell phone

Speaker Identification

- Consists of determining who is speaking regardless of the words they are saying
- Two common approaches
  - speaker-dependent speech recognition
  - GMMs density estimation
- **Speaker-dependent speech recognition:** unique models tuned to the pronunciation peculiarities of each speaker
  - collecting speaker-dependent information for a large population is impractical

Speaker Identification

- More general model like a single GMM used to capture all sounds produced by a speaker
- GMM might require up to 2,000 components to properly model the way each speaker speaks
  - large number of components is necessary because system is not trying to recognize individual words
- Speaker identification using GMMs often needs more than 10 seconds of speech to make a reliable decision

Spoken Document Retrieval

- Retrieve spoken documents that fit a text query
- Two speech-specific approaches are most commonly used
  - keyword spotting
  - phonetic recognition
- Both approaches are more robust for IR than normal speech-to-text using a speech recognizer

Spoken Document Retrieval

- **Keyword spotting:** recognize pre-selected keywords inside spoken documents
  - each keyword contains a lot of information
  - presence of one of them is highly informative
  - approach is limiting because users must include those keywords in their queries

Spoken Document Retrieval

- **Phonetic recognition:** perform retrieval at the phoneme level
- Key issue: needs to deal with mismatches at the level of underlying sounds
  - Using conventional IR techniques the words “bat" and “bet" are completely different
  - But phonetically the /a/ and the /i/ in these two words are very easy to confuse

Audio Basics

- Analyzing the audio signal, to extract basic information, is an important part of an audio-retrieval system
- Audio is recorded as a waveform
    - measures of the changes in air pressure along the wave over time
    - if sound wave is produced by combination of multiple sources, signal is complex
    - each object in a sound landscape has three primary dimensions
        - loudness, pitch, and timbre
    - For IR, we can ignore the overall loudness of the signal
    - Pitch and timbre carry different kinds of information

Audio Basics

- **Pitch:** attribute of sound that describes musical melody
  - psychoacousticians define it based on what we perceive
  - speech researchers define it based on what the glottis in the throat is doing
  - engineers define it based on the harmonicity of the signal
- Here we will use the musical definition—we are most interested in which notes are played
- For our purposes, we define

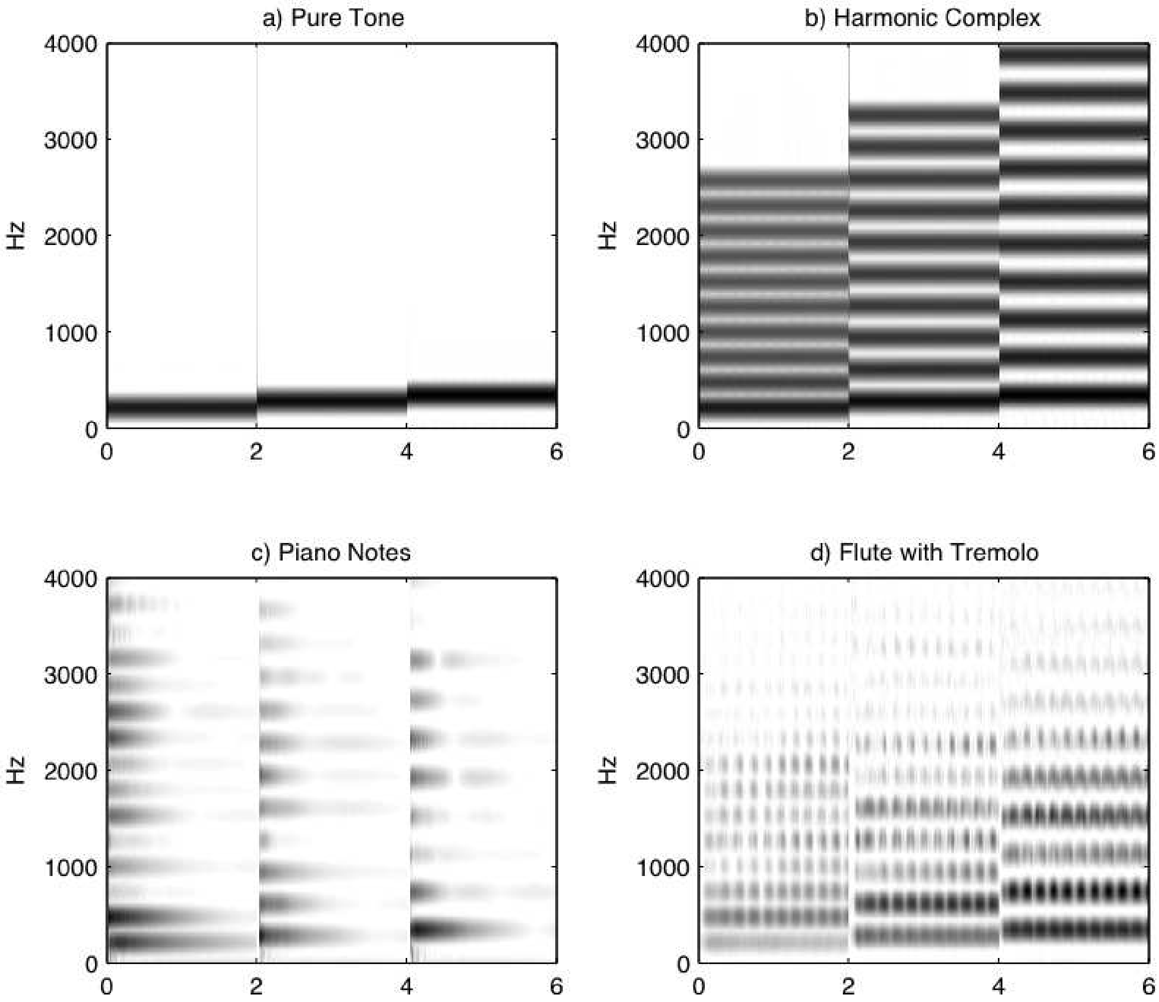
  pitch (or note): the lowest frequency in the harmonic complex

Audio Basics

- While the pitch is often ignored in speech processing, it is an important cue for
  - Auditory Scene Analysis
  - understanding the emotional content of the signal
- **Timbre:** property of the sound that allows identifying the type of musical instrument that is playing
  - separate dimension of sound that we define as everything except for the loudness and pitch information
  - allows understanding emotional and musical content in a signal
- To understand the words, we look at the timbre

Sound Spectrograms

- Describe how frequency of signal changes over time

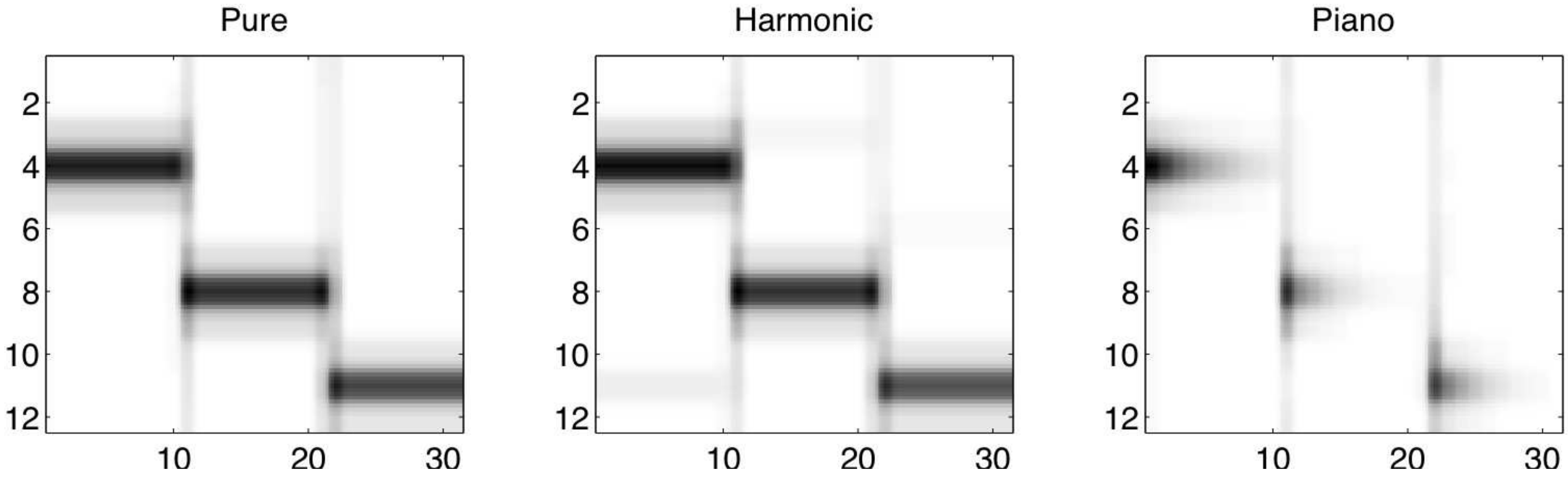


Sound Chromagrams

- Music IR systems depend on a representation of the sound known as the chromagram
- **Chroma:** cyclic metric that assigns a same value to two tones separated by an integral number of octaves
- **Chromagram:** formed from the spectrogram by combining multiple octaves into a single 12-D vector
  - if base octave is from 65 to 123 Hz, information from each octave are combined to find the estimate of the 12 notes in the chromagram
- Resulting chromagram represents the notes (or chroma) of the music as a function of time

Sound Chromagrams

- 12-dimensional chromagram, as a function of time, for 3 of the notes (cases a, b, c) shown in the last figure (2 slides back)



Mel-Frequency Cepstral Coefficient

- **MFCC:** most common representation for timbre
  - operates on each frame of the spectrogram
  - converts detailed spectral information into a (usually) 13-dimensional vector that captures the broad shape of the spectrum

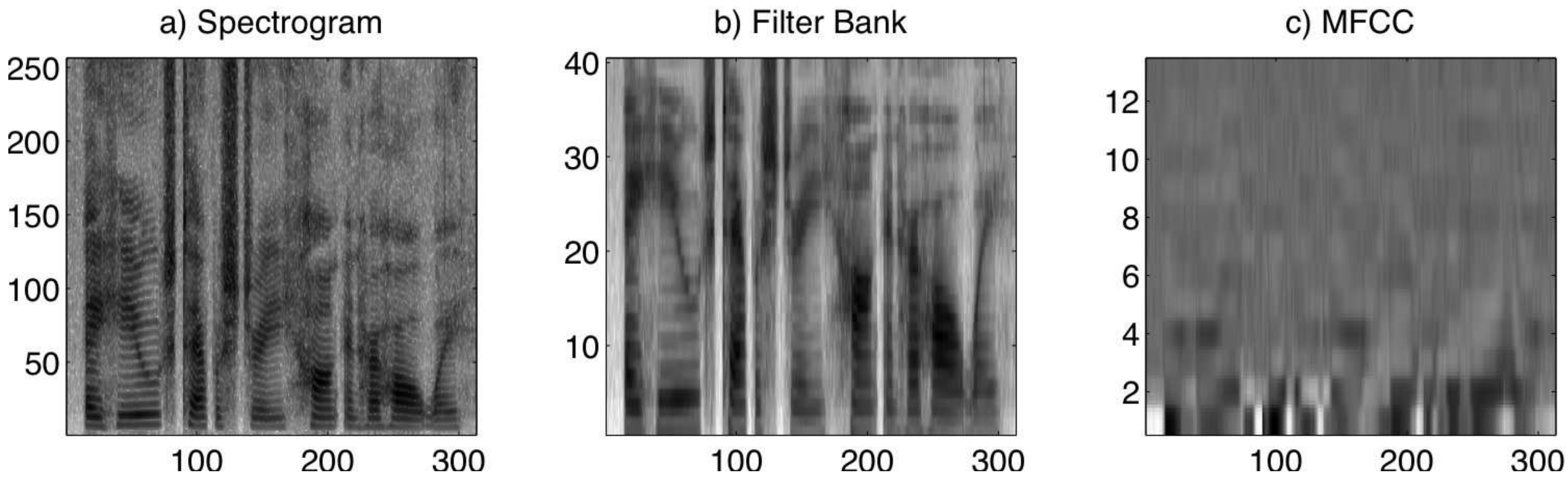
Mel-Frequency Cepstral Coefficient

- Processing steps compute MFCC of the following speech signal: "a huge tapestry hung in her hallway"

  a) spectrogram

  b) rescaling to convert to a mel-scale filter bank

  c) DCT to reduce the dimensionality to 13



Retrieving and Browsing Video

Video Abstracts

- Video representation that captures content concisely and efficiently
  - for a user unfamiliar with a video, it should be easier to assimilate abstract than original video
- Abstract covers video content when it captures all the salient topics or events of the original video
- Video-abstraction typically requires to
  - analyze and segment the original video into manageable units
  - rank such units using various combinations of visual, audio, textual, and other features extracted from the original stream
  - select the relevant units/segments that define the summary
  - generate the visualization for such summary

Video Abstracts

- Visualization schemes can be divided into two types
  - static (frame-based)
  - dynamic (video-based)
- Dynamic summaries are constructed by generating a new video sequence, typically a much shorter one, from the source video

Static Summaries

- **Static display**: something that can be printed on paper
- Simplest video summary is its title
- Next in complexity, visual summaries are based on a subset of still images (key-frames)
- Static summaries provide a compact alternative to a full video because they are assembled from static images

Static Summaries

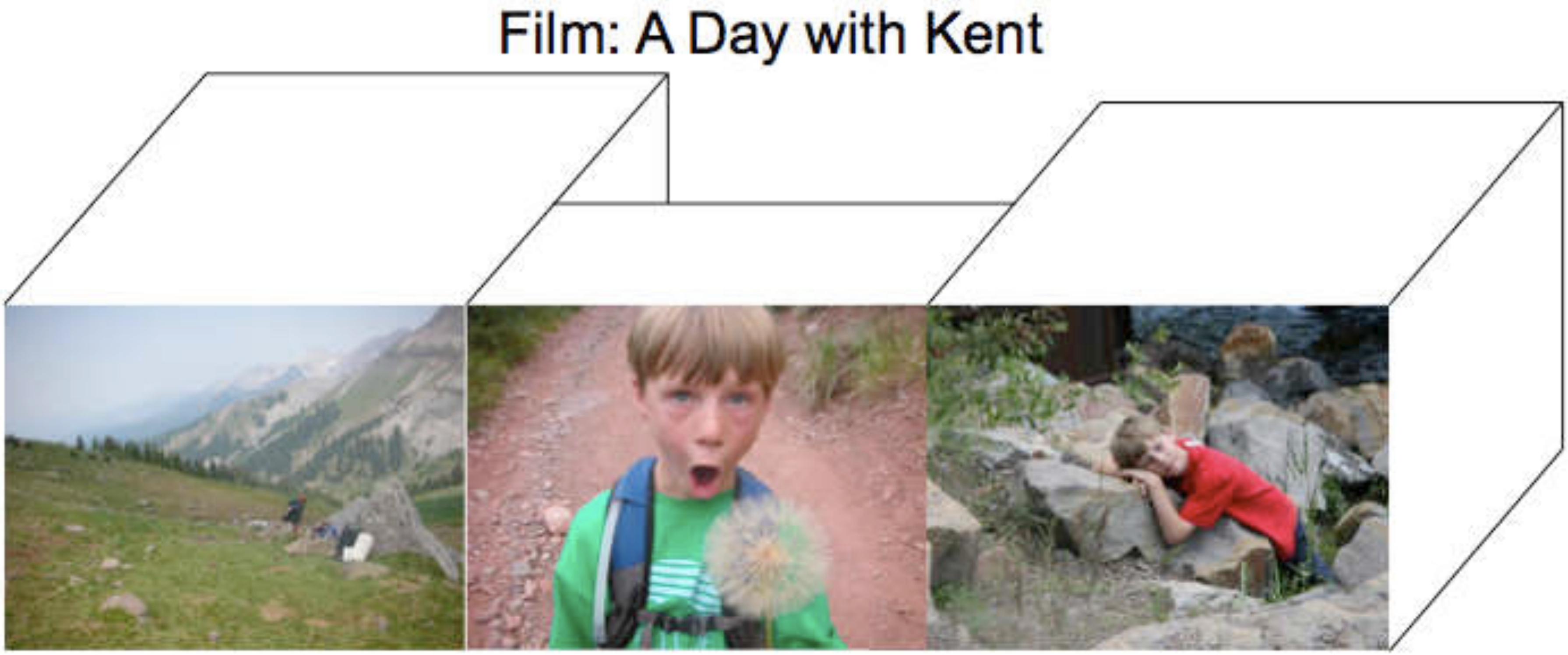
- In movie making, storyboards describe action to be shot, camera angles
  - provide a summary of the entire film
- In video summarization, storyboards are composed of an array of thumbnails in chronological order
  - early storyboard approaches were very simple
  - key-frames were selected either:
    - randomly, or
    - at certain time intervals
  - main disadvantage is that they do not provide context

Static Summaries

- More sophisticated approaches
  - extract the key-frames based on shots or scenes
  - to select key-frames, use a combination of low-level features such as color, texture, and motion
- Despite their drawbacks, static storyboards are widely used in video-retrieval systems and commercial products like iMovie

Static Summaries

- Visualization with time information into filmstrip



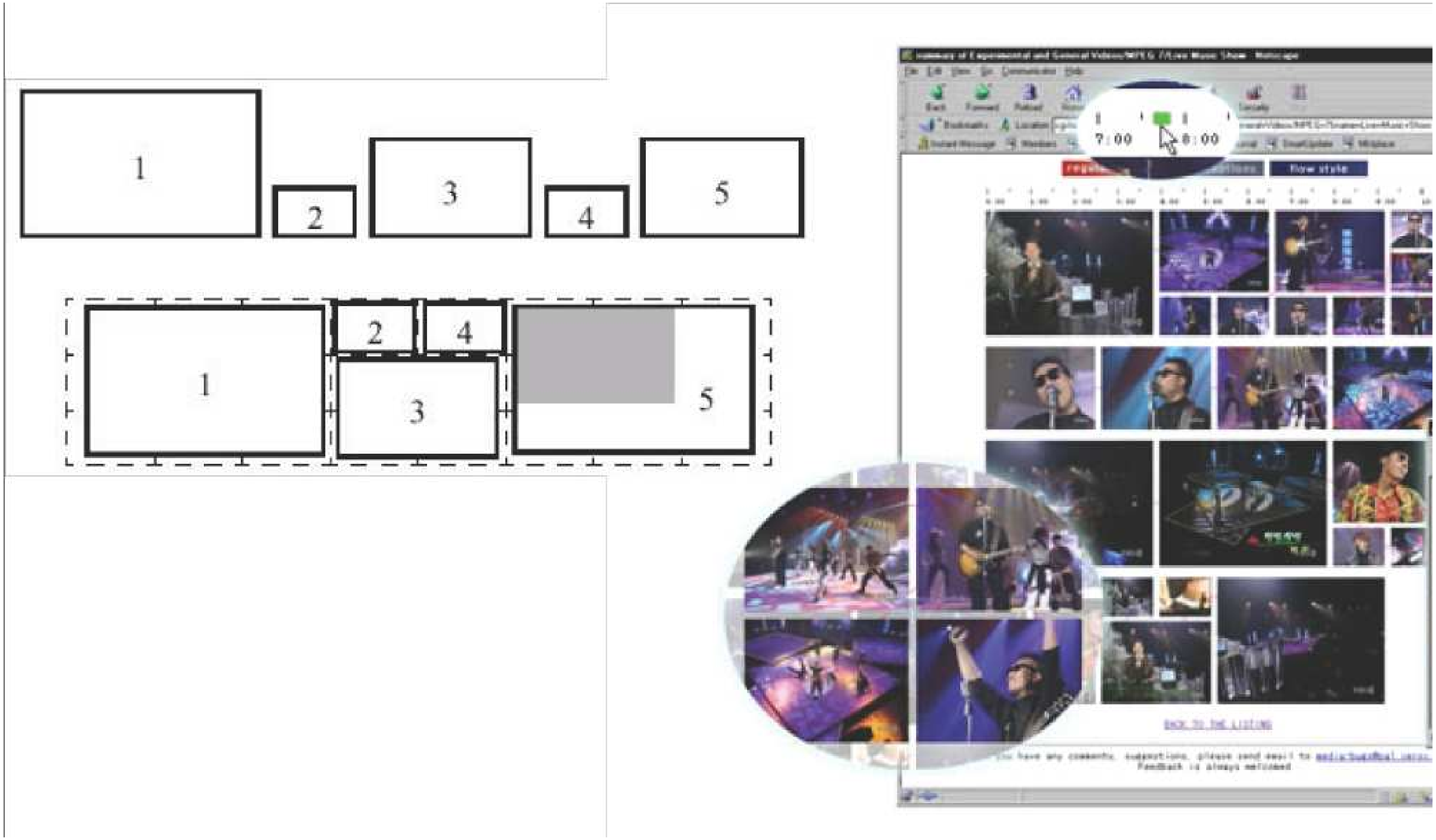
- Cuboid associated with each thumbnail has depth proportional to duration of the shot

Sophisticated Storyboards

- In traditional storyboards, thumbnails have same size
- In two-dimensional storyboards
    - thumbnails of different sizes
    - relative size indicates importance of key frame
    - Example: Video Manga
        - inspired by Manga, represents one type of storyboard
        - thumbnails of different size packed in visually pleasing form analogous to style used in comic books
    - Challenge: efficient layout of variable-size thumbnails

Sophisticated Storyboards

- Manga: size of thumbnails reflect importance of key frames

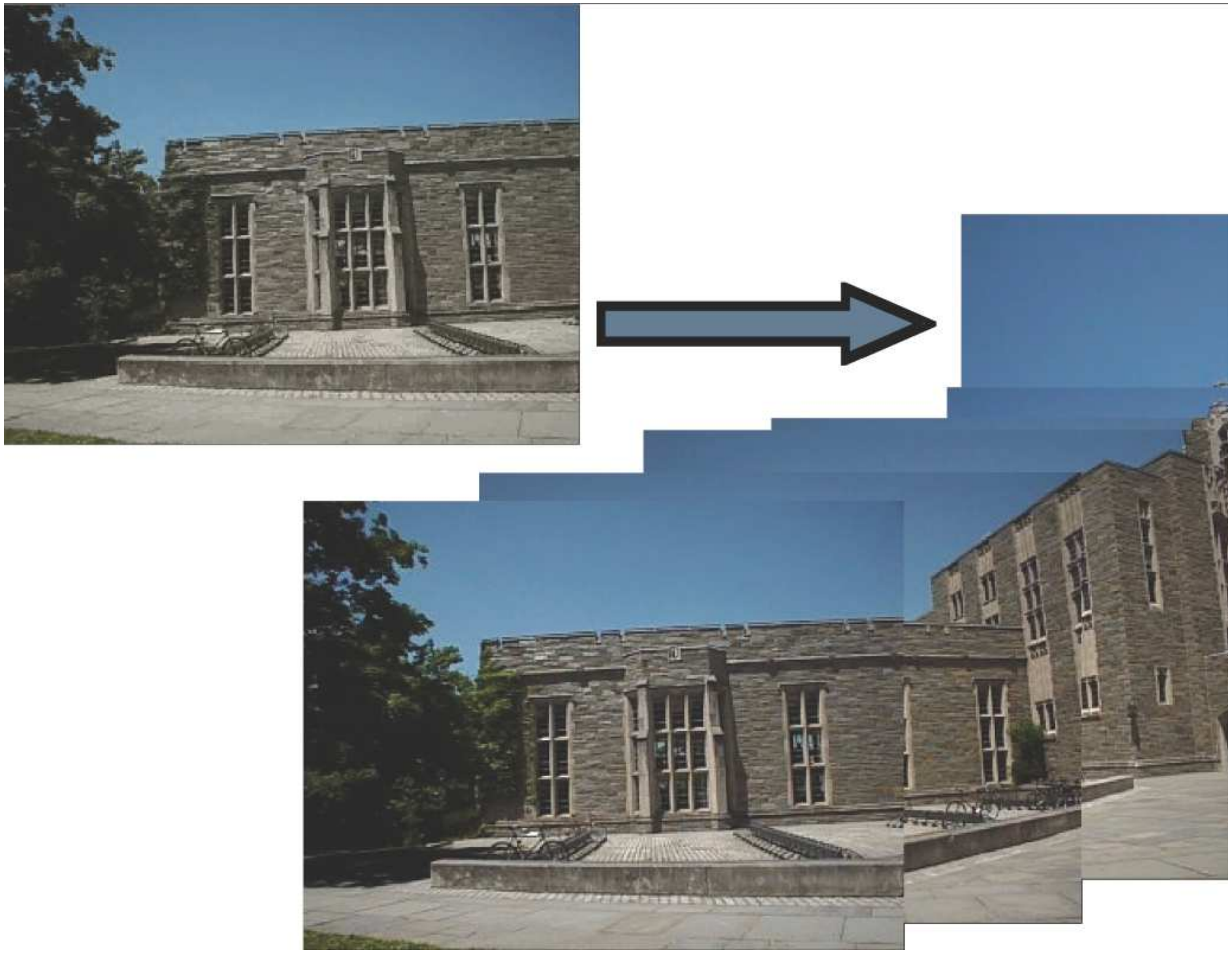


Mosaics and Salient Stills

- Shots can include moving objects and camera motion
  - tilting and panning, zooming and changes of depth field
- Shot represented by synthetic panoramic images denoted salient stills or mosaics
- **Salient Stills**
  - class of composite images that aggregate temporal changes in a shot
  - three types, depending on whether motion introduced by camera or object
    - pan
    - zoom
    - timeprints

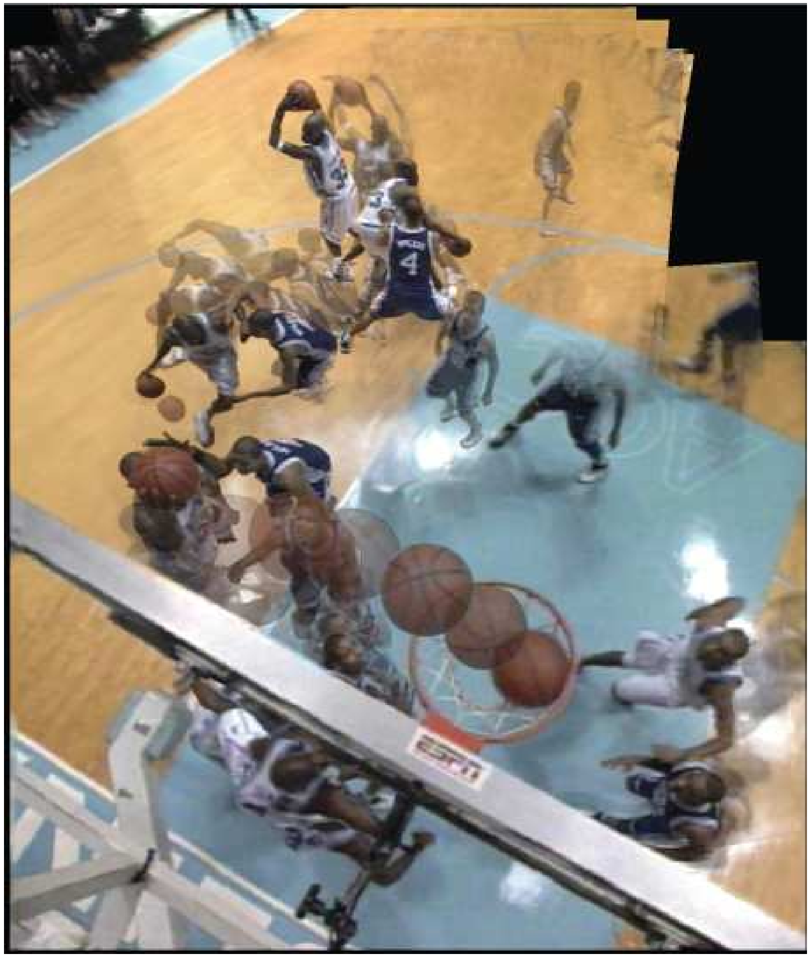
Mosaics and Salient Stills

- **PanningMosaic:** find overlap between different images in time and combine them into one image



Mosaics and Salient Stills

- **Timeprint:** multiple video frames combined into single image that shows motion



Mosaics and Salient Stills

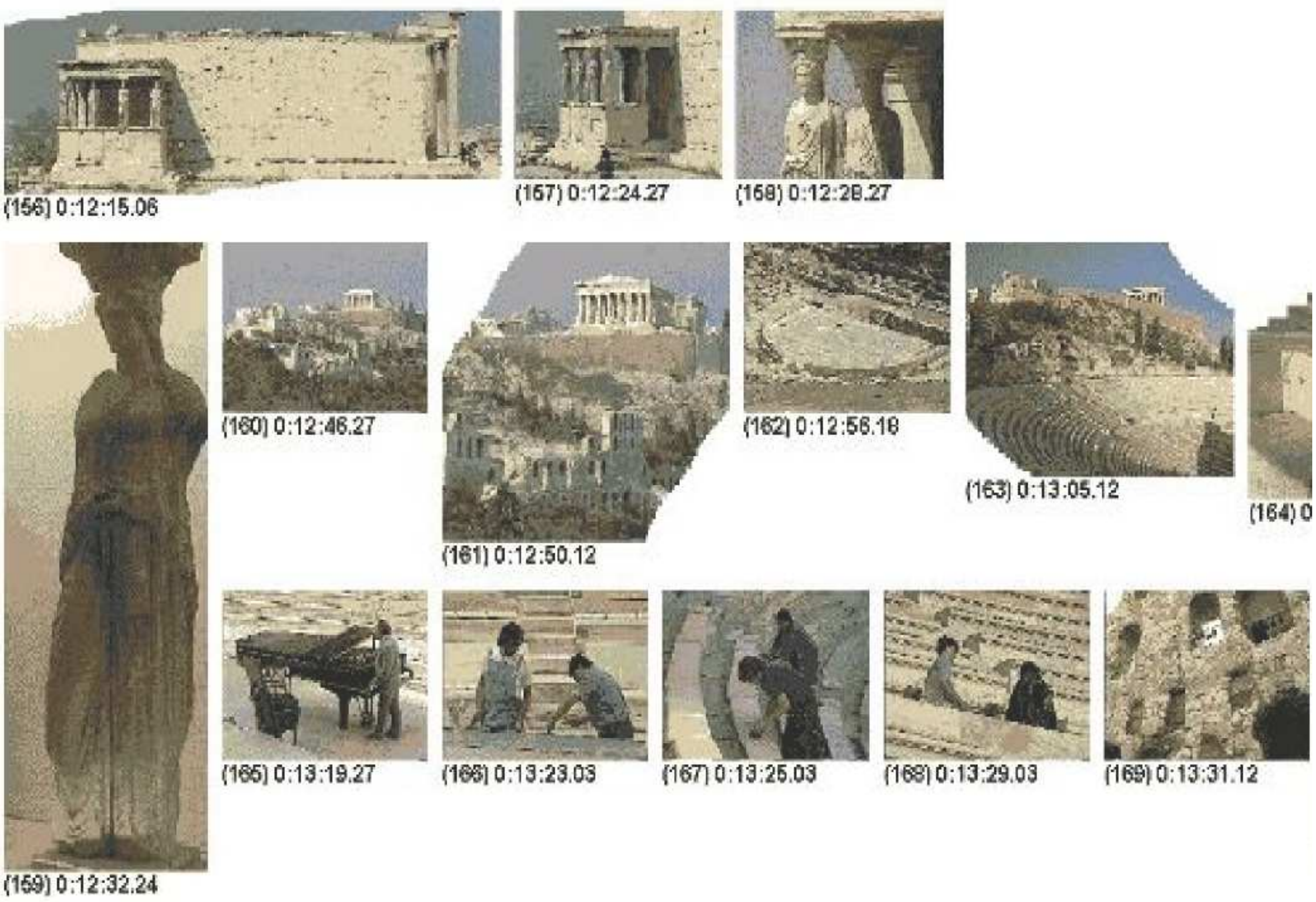
- Generation of salient stills requires two major steps:
  - modeling
  - rendering
- **Modeling:** estimate correspondence between frames
- **Rendering:** select
  - frame of reference
  - frames to render
  - how objects will be handled in relation to the background image
  - what type of temporal operator should be applied

Mosaics and Salient Stills

- For salient stills from panning
  - compute camera motion from frame to frame
  - create single panoramic still image as composite of all the frames in the shot
  - once salient stills computed for all shots, users can quickly grasp video content
- Salient stills for zoom
  - combine multiple key-frames into a single multi-resolution image
- **Timeprint**
  - salient still from zoom or pan
  - incorporates objects in the scene creating an aggregate of the background and objects positions

Mosaics and Salient Stills

- Storyboard that combines mosaics and traditional key-frames



Dynamic Summaries

- Static summaries
  - not suitable for videos where most of the information resides in audio track
- Dynamic summaries
  - incorporate time and audio
  - provide compactness and non-static representation
- Examples of these summaries:
  - slide shows
  - moving storyboards
  - movie trailers

Dynamic Summaries

- **Slide Shows** display key frames at a fixed rate and includes play controls and a time bar
    - to select the key frames composing a slide show, different algorithms can be used
- **Moving Storyboard (MSB)**
    - slide show synchronized with version of original audio track
    - can have the same duration of original audio track
    - one or more key frames per shot are extracted and displayed during the entire duration of the shot

Dynamic Summaries

- More advanced interfaces result from combining several modalities
  - speech recognition
  - image processing
  - natural language understanding to process video automatically
- Movie Content Analysis (MoCA) Project
  - generates movie trailers using several modalities
  - **movie trailer:** short version of longer video intended to attract viewer’s attention

Dynamic Summaries

- MoCA creates a video abstract in 3 steps

  1. segment video to understand the shots and identify faces, dialog, and extra text from the titles
  2. select clips that best represent the movie
  3. assemble clips by ordering them and select the right transitions

- Emotional content of the story not considered by automatic means

Interactive Summaries

- **Apple Video Magnifier**
  - early interface for video browsing
  - hierarchical view of entire movie
    - starting with row of key frames, every frame is expanded into another row to provide the next level of detail

Interactive Summaries

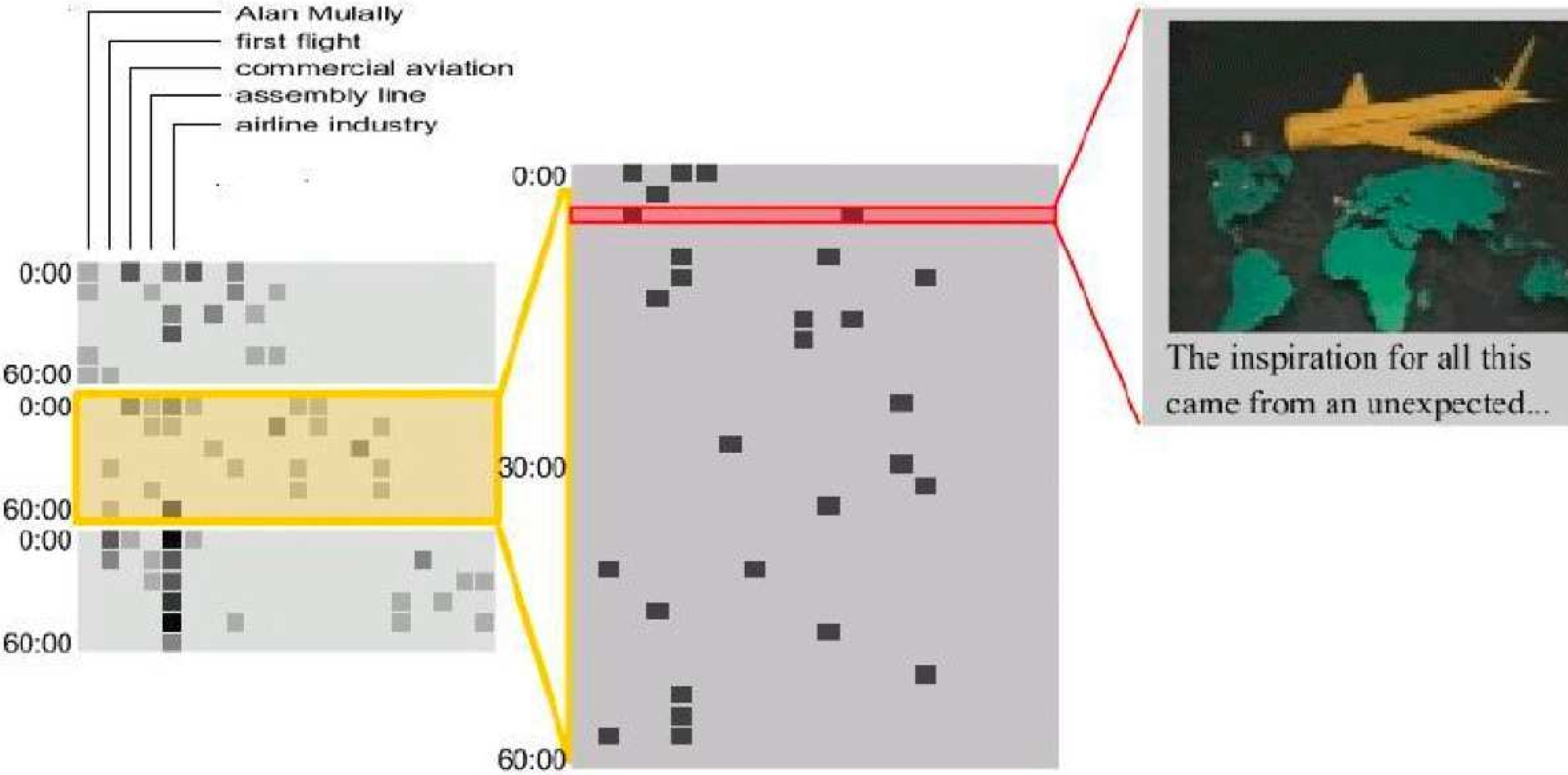
- Even sophisticated storyboards do not work for both videos and collections of videos
- One solution: **movieDNA**
  - visualization for video, video collections and linear data in general
  - $2D$ image where image graphically resembles a DNA fingerprint
  - requires segmentation of video by
    - straight-forward approach, or
    - more sophisticated content-based approach
  - time (in one or more different videos) flows down the image
  - each pixel says which **feature** is present in video
    - presence of a person
    - presence of a topic
    - type of audio
    - any other kind of metadata

Interactive Summaries

- In movieDNA, user can quickly
  - see what is in the video
  - see when it occurs
  - jump to the appropriate segment
- **HMDNA: Hierarchical movieDNA**
  - aggregation of several movieDNAs
  - provides high-level overview of a video collection, at-a-glance

Interactive Summaries

- 2-Level Hierarchical MovieDNA



Visual vs. Audio Browsing

- Humans are much more efficient at browsing visual content than browsing speech or audio content
- Defining audio unit equivalent to thumbnail image unit is challenging
- For starters, need to listen to one audio stream at a time
- Two approaches for speeding up audio:
  - time-scale modification (TSM)
  - speech summarization

Visual vs. Audio Browsing

- TSM algorithms
  - generate comprehensible speech signal by shortening signal in a way that preserves pitch, timbre, and voice quality of the signal
  - speech can be sped up by a constant factor of 2.5 and still be comprehensible to an average user
  - alternative: analyze the words and select only some of the phrases or sentences for playback; for this use
    - speech-recognition algorithms to extract the text and timings
    - text-summarization algorithms to select the most important phrases

Evaluating Summaries

- No universal definition of a set of evaluation metrics to determine quality of a summary
- In most cases, evaluation is subjective
  - determining whether a user can successfully perform specific tasks while using a summary instead of the original video
- Key point: evaluating quality of a summary depends on the questions you ask subjects

Fusion Models: Combining it All

Fusion Models: Combining it All

- Multimedia fusion
  - combining different kinds of data to make a better decision for a multimedia-retrieval task
  - two distinct kinds of fusion
    - recognizing one domain based on information from the other
    - using both domains for simultaneous recognition of content of interest
- Recognizing one domain based on the other
  - build a joint probability model that fuses a multimedia signal with a textual model
  - allows using audio to label faces, images, and audio

Fusion Models: Combining it All

- Use both domains for simultaneous recognition of content
  - use different kinds of information to better understand the signal
  - best example: audio–visual speech recognition
    - improve speech recognition by reading the lips in a video

Naming Faces

- The Web, and especially news articles, are filled with pictures and their captions
- Inherent problem: extract, from the captions, names for faces in the pictures
- Berg and her colleagues solve this problem in three stages

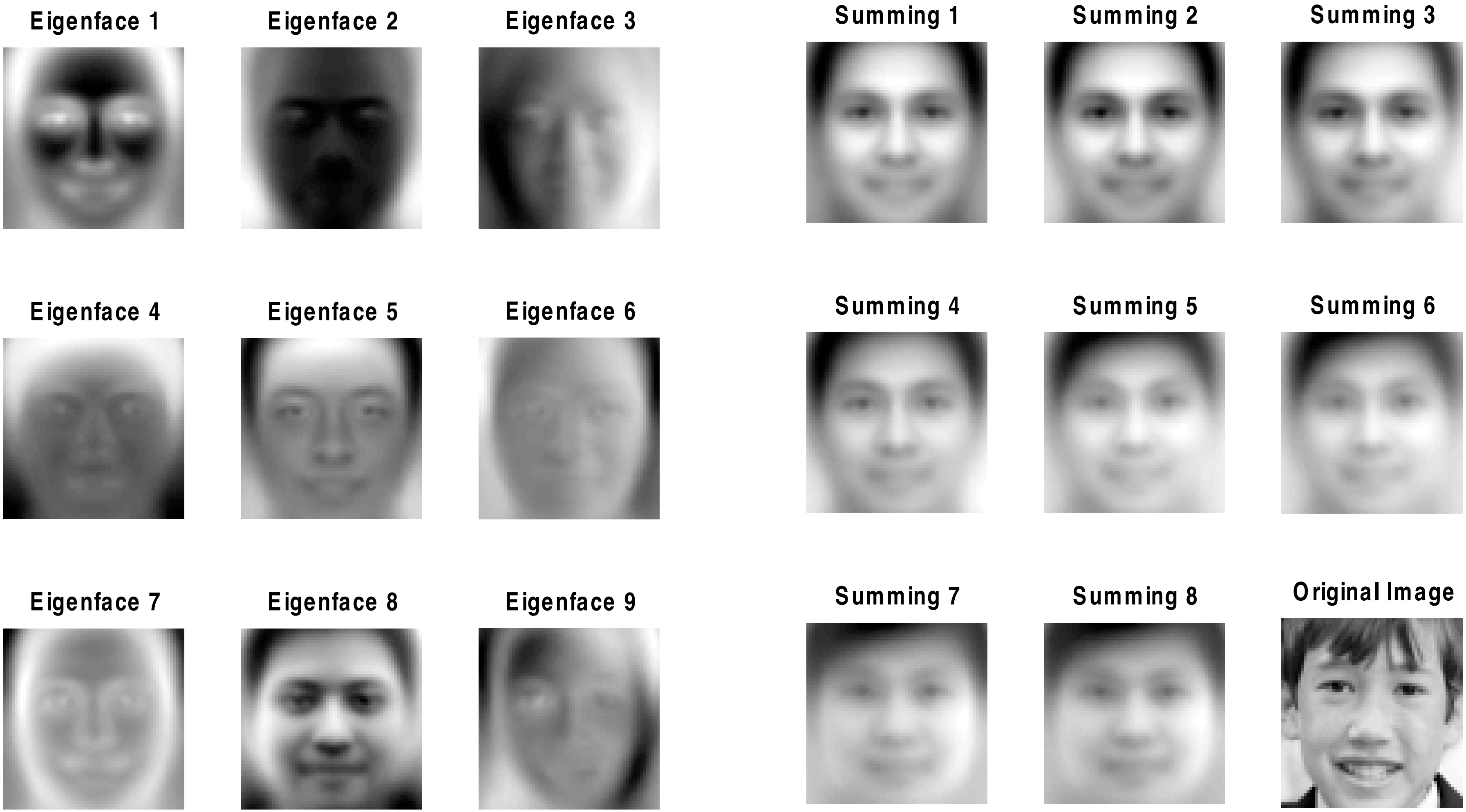
1. use a technique based on principle-components analysis (PCA) to find faces in image
2. use simple named-entity detectors to look in the caption for proper names
3. cluster all the facial images that are labeled with each name

Naming Faces

- Faces come in a multitude of styles and poses
- Yet, large range of images preserve common features
- **EigenFaces**
  - important tool for recognizing common features in faces
  - find optimal subspace using principle components analysis (PCA)
  - all (training) images of faces are aligned so that eyes and other features of the face are always in the same spot
  - image brightness is then read out of the image, composing a single vector of size $N \times M$
  - each facial image forms one point in high-dimensional space
  - discriminate the portion of the space that corresponds to faces from the portions that do not

Naming Faces

- Eigenface representations



Naming Faces

- Named-entity detector extracts common names from the captions associated with each image
- Difficulties
  - proper names that do not correspond to a single face, such as an organization
  - faces in the image that do not have a name listed in the caption

Naming Faces

- Problem: establish correspondence between proper names and images
- Solution: use a combination of clustering and expectation–maximization
  - Berg builds probabilistic model that divides EigenFace space
  - expectation–maximization (EM) algorithm is used
    - estimate a probabilistic model that connects EigenFace space to each potential name
    - use either maximum-likelihood or an average estimate to assign a name (or null) to each face image
  - repeat until name-image assignment converges
  - Berg gets approximately 78% accuracy on an identification task over 1000 images on the Web

Naming Images

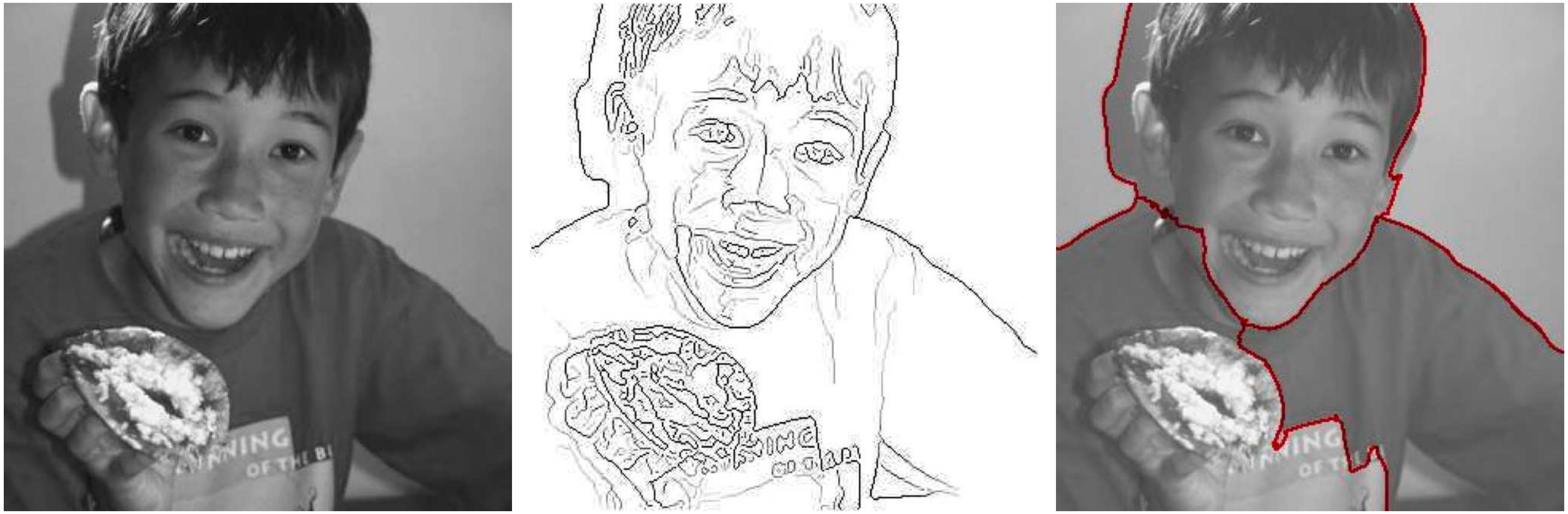
- More general approach: fuse images and words
  - use a generalized language model
  - any number of words are used to describe portions of an image
- Barnard proposes solution based on machine-translation
  - like translating from one language to another, connect image features to words
  - use hierarchical image clustering
  - label each cluster with a set of words

Naming Images

- First task when analyzing images
  - identify different regions of image that correspond to different objects
  - for this, use normalized cuts
- **Normalized cuts**
  - graph that connects each pixel to each other pixel is built
  - weight of the edge is a function of how similar the two pixels are
  - function describes how spatially separated two pixels are in the original image
  - can be formulated as a singular-value decomposition (SVD) problem

Naming Images

- Image segmentation performed by normalized cuts



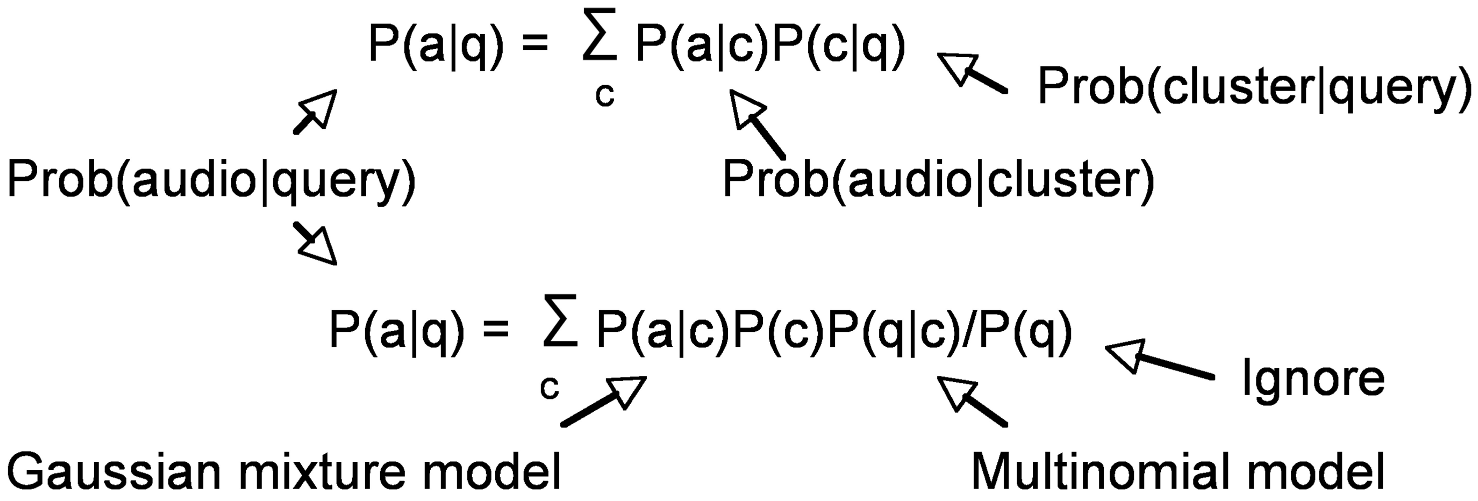
- Given an image, we can
  - query the word–image probability model to estimate the words that are most likely to be associated with that image, or
  - find the image features that best correspond to any word

Naming Audio

- Slaney studied analogous approach
  - but aimed at connecting audio and words
  - each sound file assumed to contain just one sound
  - no segmentation is needed
  - sounds from two different sound-effects libraries were linked with their textual description
    - anchor space represents sounds as points, or anchors, that correspond to distances in an ensemble of sound models
  - distances from the query sound to each of the anchor models compose a vector
  - distances are computed using GMMs, much like the models of a speaker in speaker-identification

Naming Audio

- Math involved in semantic–audio retrieval
  - second equation follows from application of Bayes Rule to first equation



$$P(a|q) = \sum_{c} P(a|c)P(c|q)$$

Prob(cluster|query)

Prob(audio|query)

Prob(audio|cluster)

$$P(a|q) = \sum_{c} P(a|c)P(c)P(q|c)/P(q)$$

Ignore

Gaussian mixture model

Multinomial model

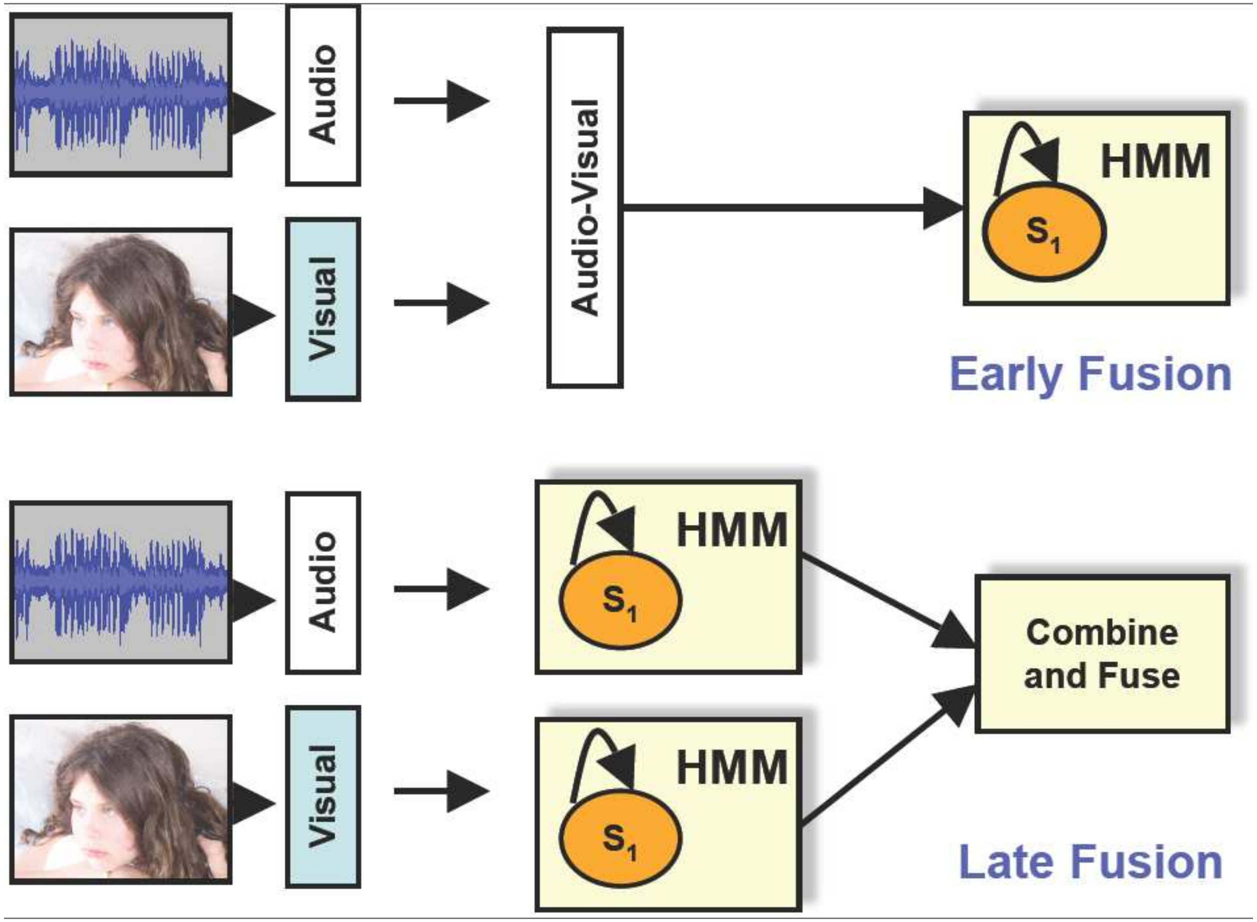
Combining Audio and Video for AVSR

Combining Audio and Video AVSR

- Audio–visual speech recognition (AVSR)
  - combines acoustic and visual information on the face of the talking person
  - even in great acoustic conditions, easier to distinguish with visual information
- Represent visual evidence with pixel-based or shape-based features
  - **pixel-based features:** image pixels, often transformed much like EigenFaces, form a feature vector
  - **shape-based features:** based on finding the location of facial features such as lip positions and jaw outline
  - either type of feature, or a combination of them, can be used as input to a AVSR system

Combining Audio and Video (AVSR

- Two different approaches to the AVSR problem

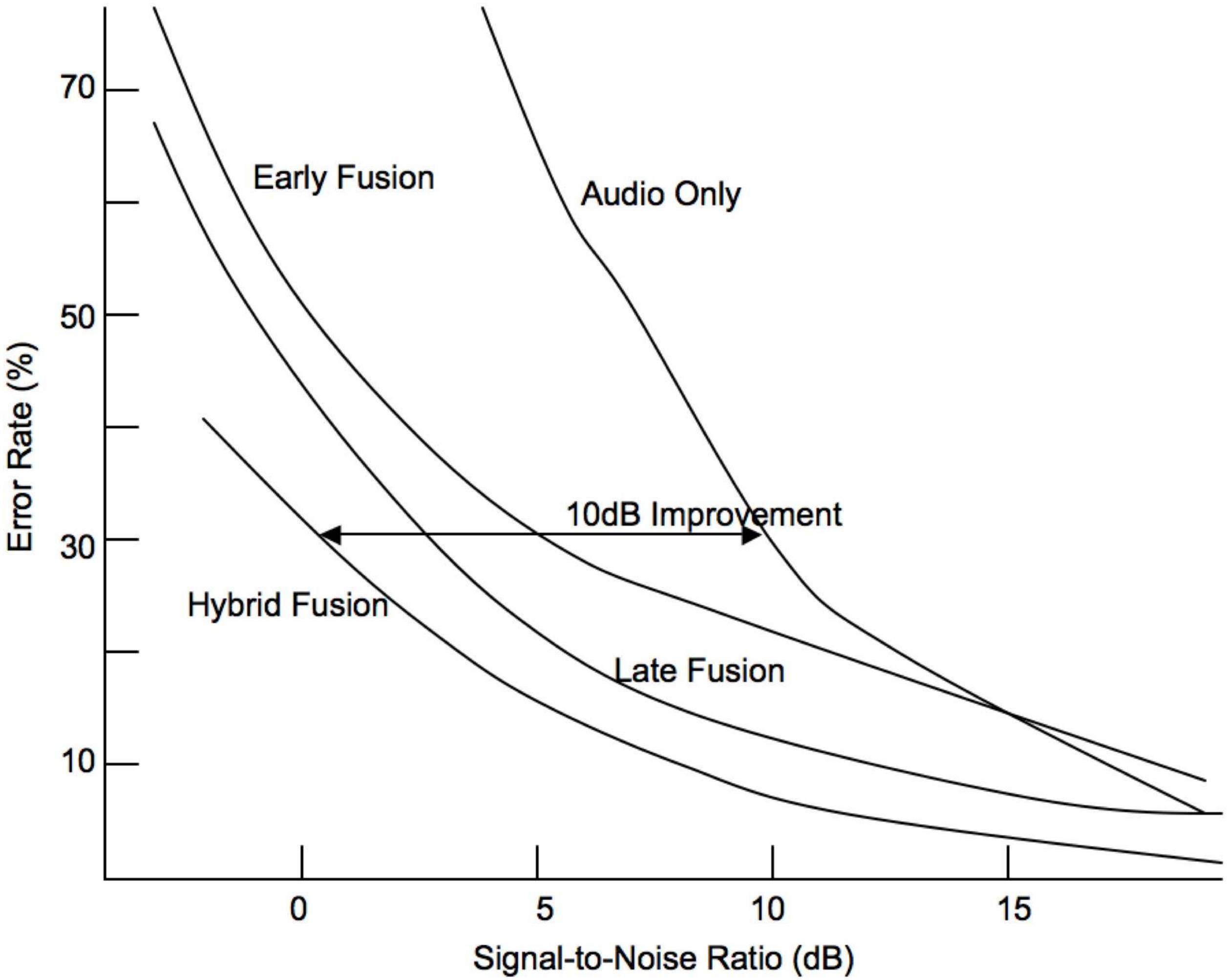


Combining Audio and Video (AVSR

- **Early fusion**
  - acoustic and visual features are concatenated
  - provided as input to a conventional recognizer
  - information combined at early stage before decision made
- **Late fusion**
  - use separate recognizers to make decisions about acoustic and visual information
  - the two decisions are then fused to decide which word is present
- Conventional speech recognizers might use various phonemes for English
- Visual counterpart is a **viseme**
  - characteristic visual pattern that represents one position of lips

Combining Audio and Video (AVSR)

- Typical results for audio and audio–visual speech recognition



Combining Audio and Video (AVSR)

- It seems logical that a recognizer should do better with an early fusion approach
  - early fusion has all information it needs to understand correlations and other oddities of both data
- Yet, AVSR works better with late fusion instead of early fusion
- One hypothesis is that joint probability model is too complicated for a single recognizer to learn
  - maybe because model cannot capture the nuances of a joint distribution
  - maybe because there is not enough data
- In either event, late fusion, or a hybrid approach that combines early and late decisions, works better

Combining Audio and Video for Multimedia

Combining for Multimedia

- AVSR can also be used to solve the more general audio–visual recognition problem
- Example: multimedia system proposed by IBM researchers
  - best audio-only model had a precision of 30%
  - best visual-only model (face recognition) had a precision of 29%
  - best combined system, based on late fusion, had a precision of 47%

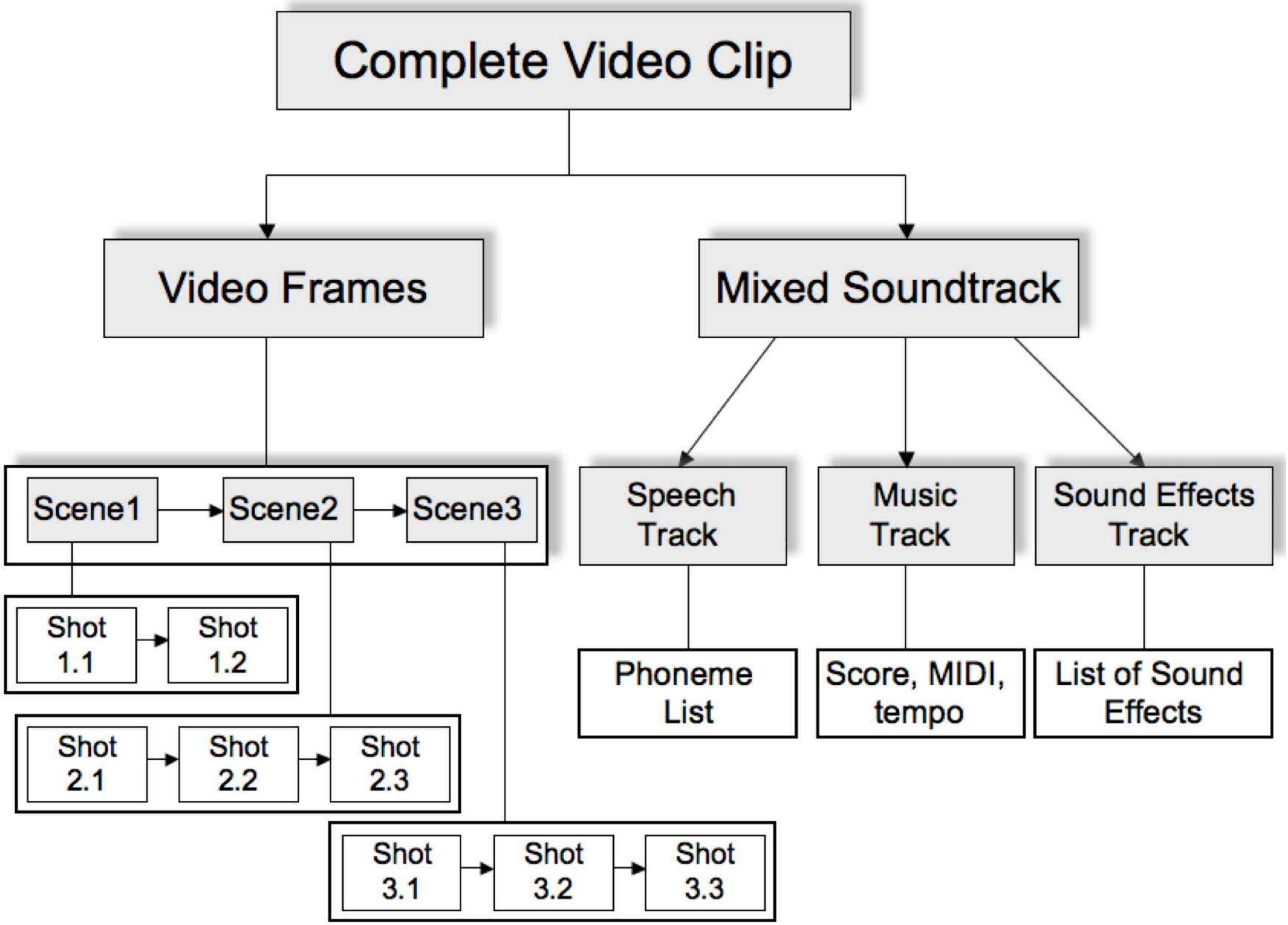
Segmentation

Segmentation

- Task of dividing multimedia objects into smaller objects, prior to processing multimedia queries
- A video is composed of a number of scenes
  - **scene:** sequence of adjacent shots that are semantically coherent
  - related scenes grouped into higher semantic units, called **segments** or **stories**

Segmentation

- Hierarchy of objects in a film or video



Segmentation

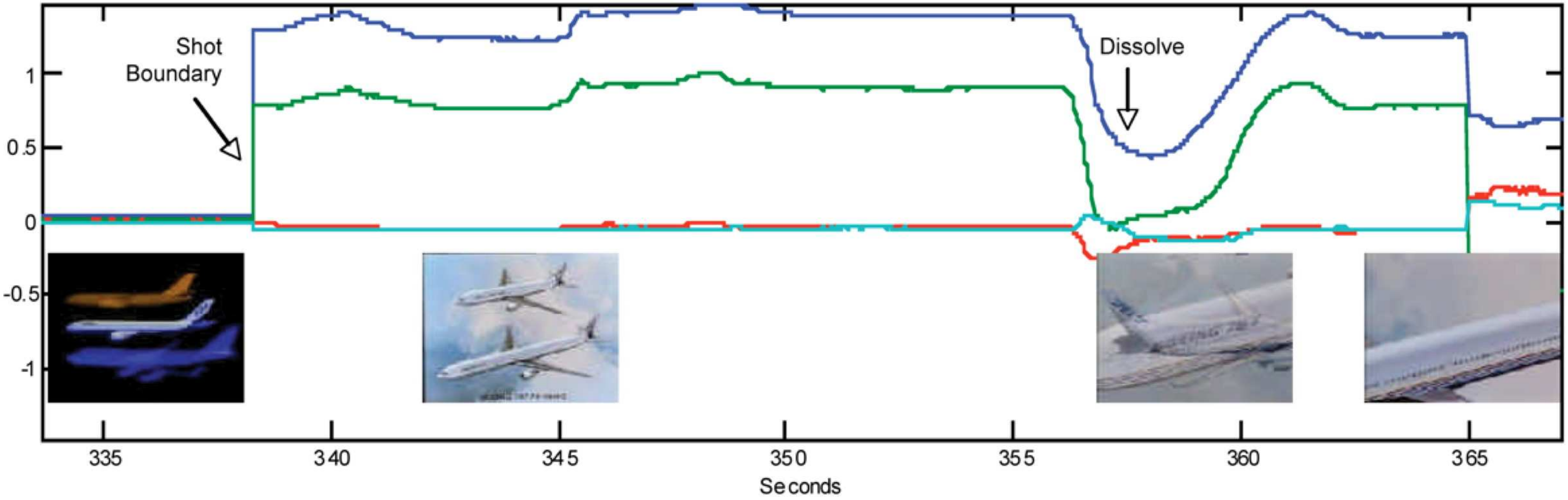
- **Transition**
  - a video change from one shot to the next
- **Cut**
  - abrupt transition
  - easy to detect because entire image changes
- Other transitions
  - fades, dissolves, and wipes
  - occur slowly over time
- Segmentation algorithms can be of various types
  - pixel-based, statistical differences, histogram-based, edge-based, DCT-based, and motion-based

A Video Segmentation Example

- Shot-boundary detection
  - algorithm for video segmentation
  - looks at summary statistics to determine instants of major changes in video
  - one simple global statistic of value: color histogram of image
- **Color histogram**
  - computed by counting number of pixels of each color in image
  - color often represented as three 8-bit numbers
  - number of colors is too large to count directly
  - 512 different color types: ($8 \times 8 \times 8$ colors)

A Video Segmentation Example

- Summary of color information in a 35-second long section of the "21st Century Jet" video
  - sharp transitions in the signals correspond to shot boundaries
  - large and slow transitions indicate a dissolve



A Video Segmentation Example

- Best low-rank approximation to the signal
  - singular-value decomposition (SVD) algorithm
  - in previous figure, notice small changes in signal during middle of a shot
  - for instance, due to motion of object in the frame (at 345 seconds)
  - shot changes are clearly visible as a dramatic change in the color signal, for example at time 338
  - near 357 seconds, two related images are superimposed to provide a smooth transition between video clips—a **dissolve**

Segmentation Schemes for Video

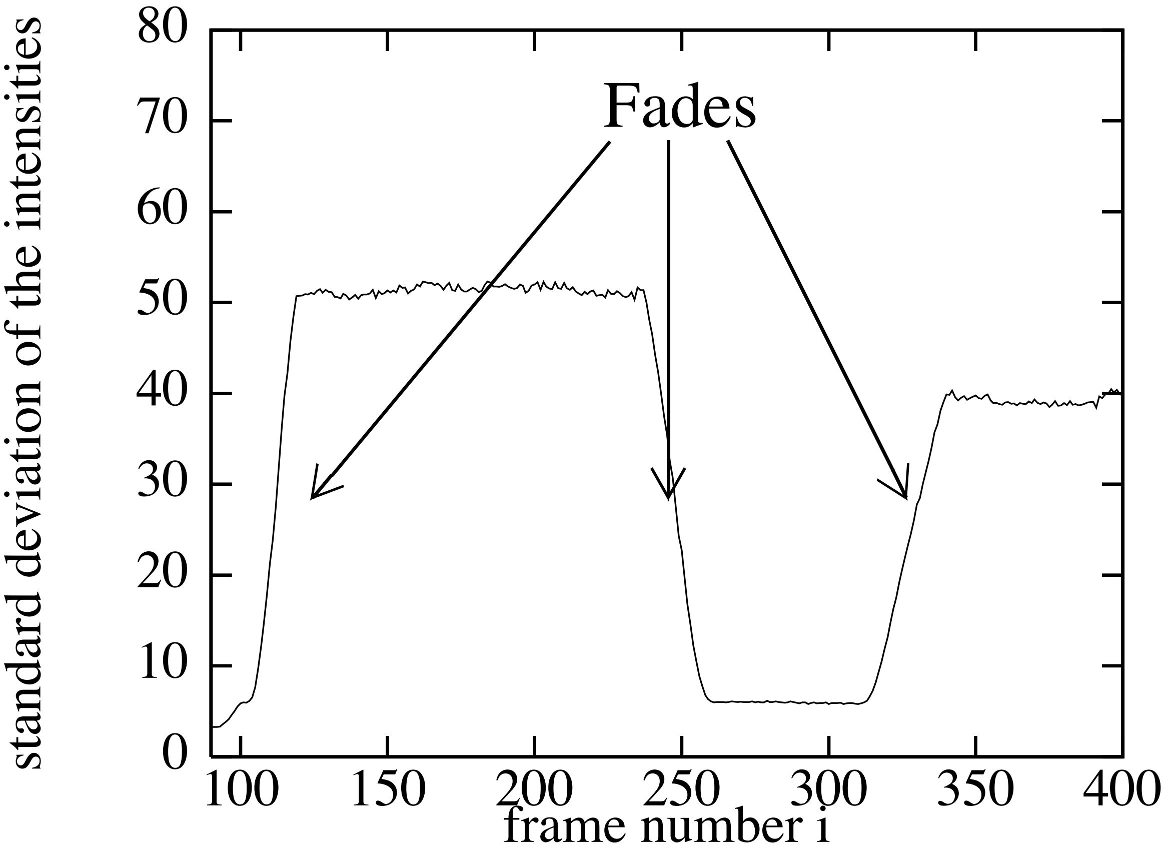
- Histogram method works well
  - because changes to video caused by camera or object movement
- Detecting fades is more difficult
  - because it represents a change in the video over time
  - starting with a normal image, video linearly decays to black and then grows into new image
- Simple fade detector looks for frames that have a constant color

Segmentation Schemes for Video

- **Dissolves**
    - hardest segment boundary to detect
    - image slowly changed from one scene to another by cross-fading the pixels
    - for example, performing linear interpolation of consecutive images on a pixel-by-pixel basis
    - to detect such changes, measure overall variance of luminance of each image frame
    - during a dissolve, two images are blended and combination reduces overall variance

Segmentation Schemes for Video

- To detect a dissolve, look for dip, lasting several seconds, in mean luminance variance



Segmentation Schemes for Video

- More robust approach to find dissolves
  - build explicit model as done by Covell
  - given any two points within a dissolve, intermediate points are simply a linear interpolation of endpoints
  - sample pairs of frames in video at intervals of 1 second and check if intermediate frame is predicted by a linear interpolation of endpoints
  - prediction error gives estimate of how likely a dissolve is at this point
  - expand the region with low-prediction error to find beginning and end of the dissolve

Video Segmentation with Edges

- **Edges**
  - sharp discontinuities in the luminance of an image
  - interesting because they are robust to lighting changes and camera motion
  - Zabih proposes to combine motion estimation, edge detection to find edge-change fraction
  - basic idea: look for edges that do not appear in next image in the sequence (and vice versa)

Video Segmentation with Edges

- First, register the two images to remove any global motion
  - done by finding the shift $\Delta x, \Delta y$ that maximizes pixel to pixel correlation $\mathcal{C}_{rr}$ between current frame (I_1) and next frame (I_2)

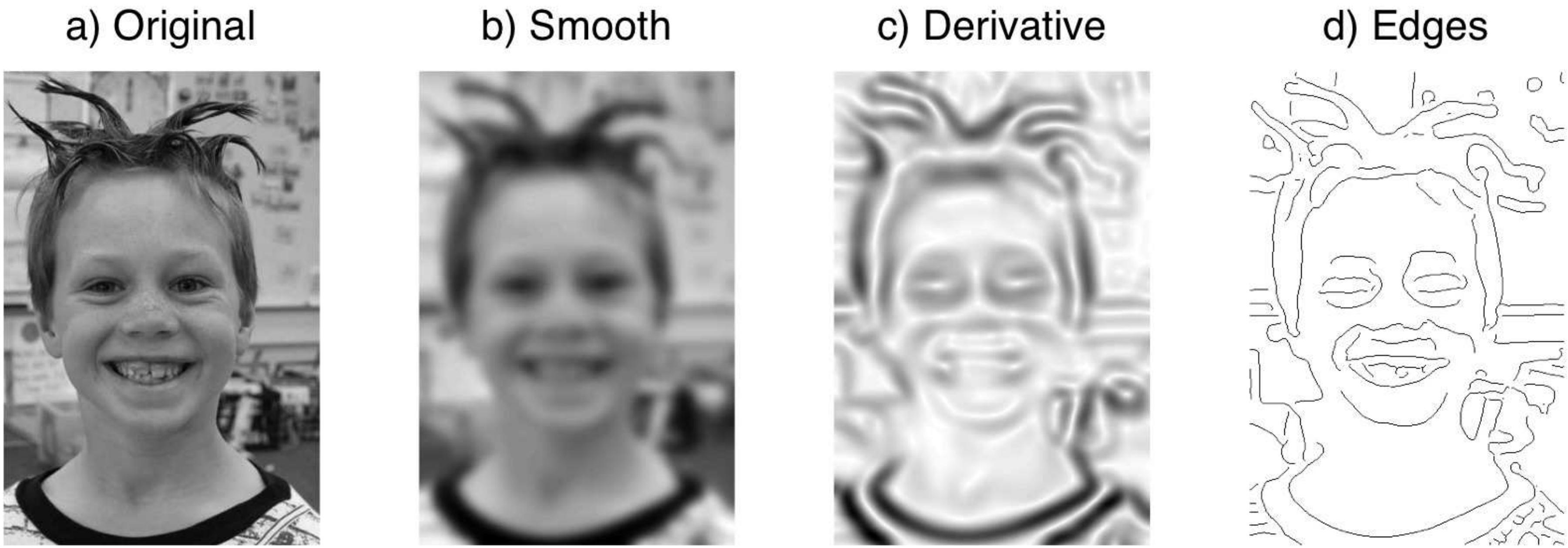
    $$\sum_{x,y} \mathcal{C}_{rr}(I_1[x + \Delta x, y + \Delta y], I_2[x, y])$$

    where x and y are pixel coordinates
  - given offset required to bring the two frames into alignment, we have two images that are roughly aligned
  - as a result, can find and match the edges

Video Segmentation with Edges

Canny edge detector

- finds important (for the purposes of this algorithm) points in an image
- process of computing edges using Canny's method:



Video Segmentation with Edges

- Scene breaks (of all kinds)
  - found by counting edges that come and go between frames
  - for each edge location, look for a corresponding edge in a small region of the other image
  - fraction of edges that are found in two images provide measure of image similarity
  - when entire scene changes, measure registers a low similarity—a **scene break**
  - edge-based detection is less sensitive to motion and chromatic changes than histogram-based detection

Speech Segmentation

- Segmentation boundary
  - characterized by a decision that something has changed in the signal
  - probabilistic way to make this decision is:
    - to build a model of the first portion of the signal
    - to advance the model through the signal
    - to detect the point at which the model no longer fits or explains the data
  - this (one-sided) calculation is error-prone because a new point might not fit the model
    - point can be caused by a noise

Speech Segmentation

- Can use double-sided approach that compares models on both sides of a potential boundary
- **Bayesian Information Criteria (BIC)**
  - build a model for signal
  - segment signal into two smaller pieces
  - build two different models
  - given model M_i and data D_i with $i = 1, ..., N$

$$BIC(M_i) = \log P(D_1, D_2, ..., D_N | M_i) - \frac{1}{2d_i} \log N$$

  - d_i: number of independent variables in model M_i
  - first term: log likelihood that model explains data
  - second term: penalizes models that are more complicated because they take more parameters to describe them

Segmentation Evaluation

- Shot boundary detection
  - relatively mature area of research
  - general purpose algorithm might require several passes through video
  - approaches based on global statistics involve a threshold (or set of thresholds)
    - set either manually or automatically
    - in practice, only automatic adaptive thresholds make sense
- Challenge: develop a single-pass algorithm that can robustly detect cuts and transitions in real-time

Compression
MPEG Standards

Compression and MPEG Standards

- Unlike text documents, one almost never sees an uncompressed multimedia object
- Most multimedia formats remove redundant information that the human brain can not perceive
  - human eye can more readily perceive changes in intensity than changes in color
- Compression enables use of digital video in applications with restricted bandwidth requirements
  - video-on-demand (VOD)
  - video conferencing

Compression and MPEG Standards

- Five key procedures for making image and video compression efficient
  - color subsampling
  - removing spatial redundancy with discrete cosine transform (DCT)
  - entropy coding
  - motion compensation
  - removing temporal redundancy

Intensity and Sampling

- Color and intensity are the most basic elements of a picture
- Intensity of light is sampled at discrete points as function of time and space
  - if image is sampled too coarsely, information is lost
  - If it is sampled too finely, there is unnecessary (redundant) information in the image
- Intensity information is usually captured uniformly, no matter what the color

Color

- Color is a basic feature of an image
  - can be perceived and distinguished by humans
  - visible wavelengths are in the range of 400 to 700 nanometers (nm)
  - each color corresponds to a narrow band in this range
  - human eye can distinguish 400,000 colors
  - humans generally perceive color with photo sensors that are sensitive to three different bands of color
    - gamut of all colors are produced with red, green and blue (RGB) phosphors in graphics displays

Color

- Color is represented in terms of RGB intensities
- Not how human visual system perceives it
- Other color systems are used to represent color information
- **Hue, Saturation and Value (HSV)**
  - popular alternative color description scheme
  - basic colors (red, green, purple) are encoded in the value of hue
  - value (or brightness) is the overall intensity or energy of the light source
  - amount of saturation determines whether the color is pink or a deep red—its vibrancy

Color

- YC_bC_r
  - color system known used as basis of image (JPEG) and video systems (MPEG and DVD)
  - like HSV, YC_bC_r system encodes color with three values
    - a luminance or Y
    - a blue chroma signal C_b
    - a red chroma value C_r

Color

- YC_bC_r values computed as follows

$$\begin{aligned} Y &= K_r \times R + (1 - K_r - K_b) \times G + K_b \times B \\ C_b &= \frac{1}{2} \times \frac{B - Y}{1 - K_b} \\ C_r &= \frac{1}{2} \times \frac{R - Y}{1 - K_r} \end{aligned}$$

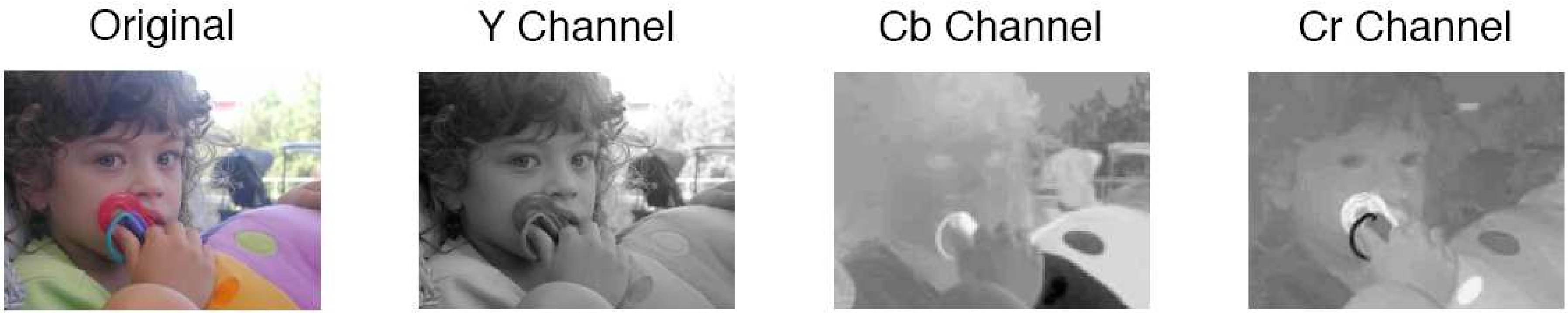
  - variables R, G, and B represent the intensities of red, green, and blue in the RGB scheme
  - K_r and K_b are constants given by $K_r = 0.299$ and $K_b = 0.114$

Color

- Downsampling color or chrominance information is important step in image compression
- Our eyes are much better at detecting spatial changes in luminance than at in chrominance
- In YCbCr scheme
  - Y signal is kept unaltered
  - C_b and C_r signals are each downsampled by a factor of 2 or 4

Color

- Effect of downsampling on color image and its three components



Original

Y Channel

Cb Channel

Cr Channel

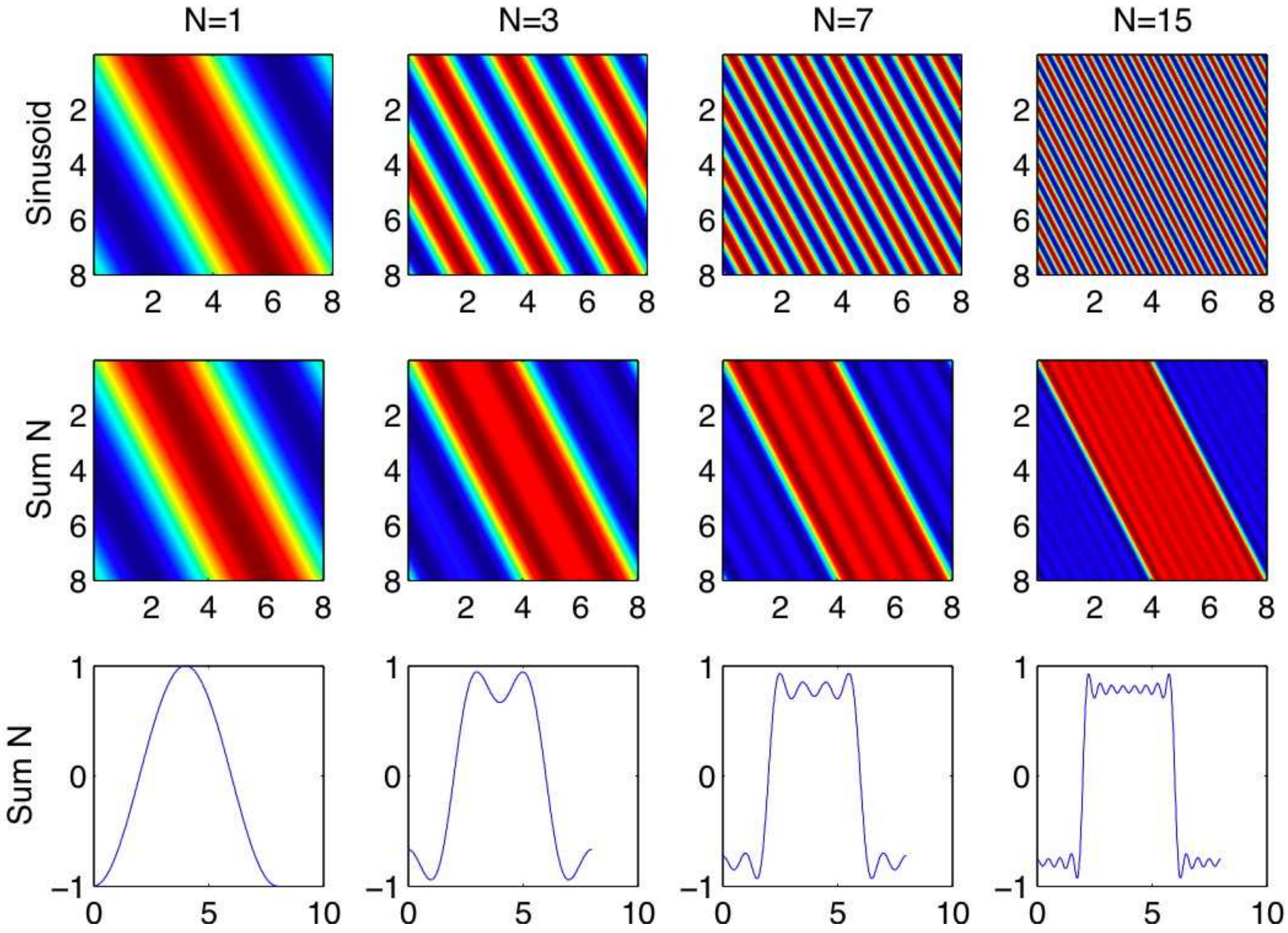
- Note compression artifacts, best seen by looking for the jagged diagonal lines in the Cb and Cr images

Lossy Compression

- After conversion of image to perceptually relevant color space (YC_bC_r), two kinds of compression
  1. lossy stage throws away information eye cannot perceive
  2. lossless stage removes statistical redundancies in signal
- Sensitivity of eye
  - ability to perceive different frequencies
  - beyond roughly 6 cycles per visual degree, ability to perceive a pattern is quickly reduced
- Sort frequency content, keep low-frequency changes
- Images described in terms of spectral content
- Image decomposed into spectral components using a discrete Fourier transform (DFT)

Lossy Compression

- DFT represents image in terms of weighted sum of spatial sinusoids

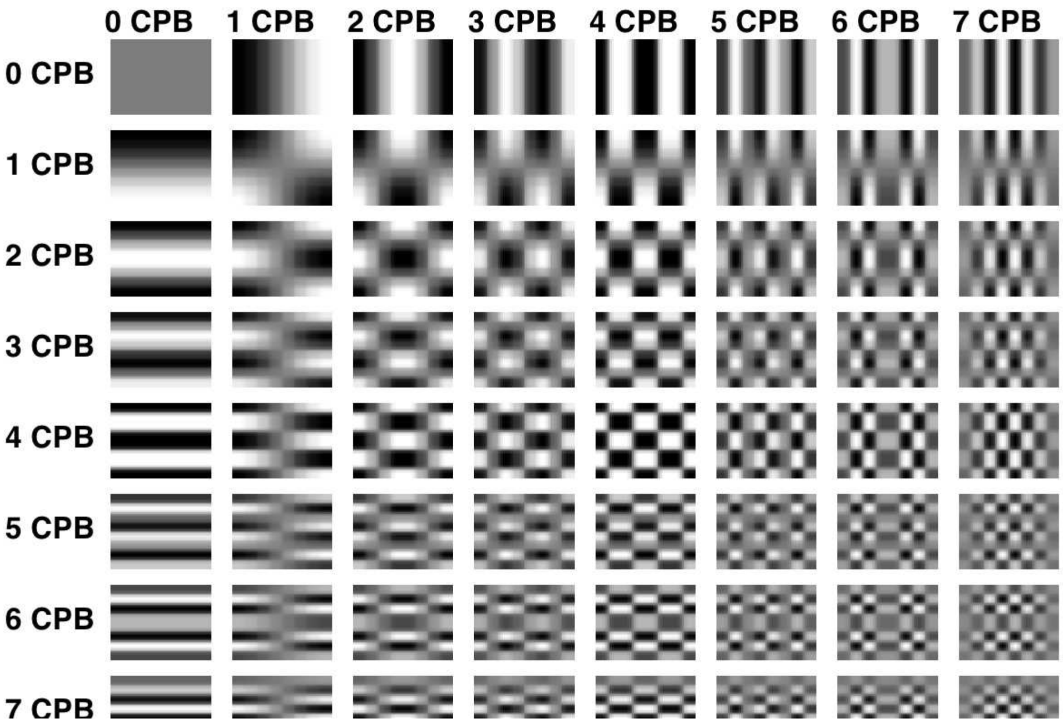


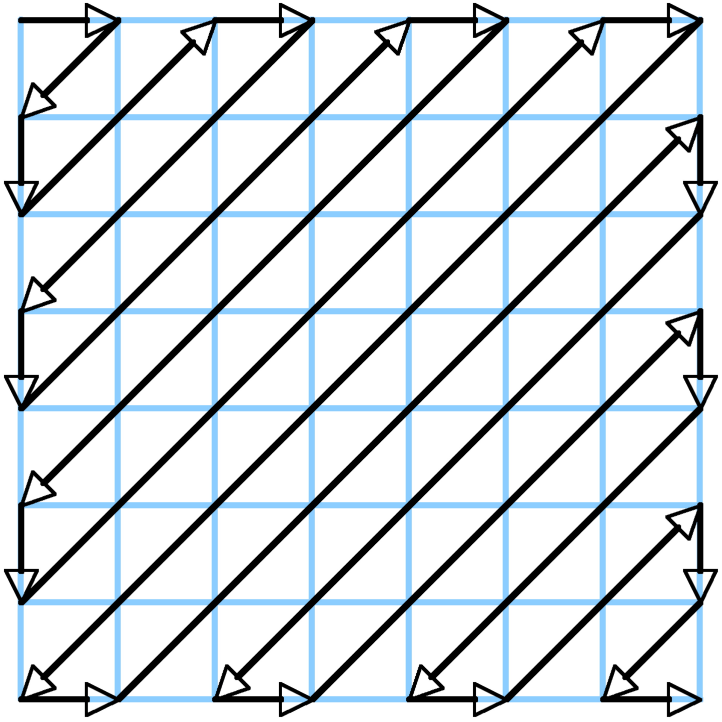
Lossy Compression

- Since eyes are most sensitive to low spatial frequencies, transmit coefficients of these frequencies with higher precision
- Spectral analysis accomplished using discrete cosine transform (DCT)
  - most important frequencies transferred with highest fidelity
- Image partitioned into as many blocks of 8x8 pixels as needed to fully cover image
- DCT represents each block of pixels with 64 different base functions
  - each function represents different combination of horizontal and spatial frequencies

Lossy Compression

- 64 base functions





  - compute DCT of block, 64 coefficients, one per base function
  - compute DCT again, inconsequential rounding errors

Lossless Compression

- Lossless compression
  - applied after lossy compression
  - further compresses the data
  - does not introduce any errors in the representation
- After removing redundancies in signal, aim at removing statistical patterns in the numbers
- Two common approaches
  - run-length encoding
  - entropy coding

Lossless Compression

- **Run-Length Encoding (RLE)**
  - during DCT processing, rearrange pixels in 8×8 DCT block in order of their importance
  - transmit these coefficients by transmitting a value and a count of how many times the same value is used
- **Entropy Coding**
  - second stage of compression
  - uses entropy (randomness) of coefficients to design optimal coding scheme

Temporal Redundancy

- In video compression, do two additional and related types of redundancy removal
  - motion estimation
  - image prediction
- Important in video because one video frame often looks very similar to the next
- Transmit just the changes between one frame and the next, which is a delta image

Temporal Redundancy

- Transmit only the first image in a scene
  - then transmit delta images
  - an image can only be reconstructed if we first decompress all preceding images
  - very sensitive to errors, which makes skipping through the video more difficult
- Alternative is provided by MPEG compression
  - delta images can be computed in either forward and backward directions relative to fully-transmitted images known as I-frames

Motion Prediction

- Given two images, compress the first image and then use it to predict the second image
    - use a pixel in the first image to predict the exact same pixel in the second image
    - using subtraction alone does not necessarily decrease the amount of transmitted data
- Video compression uses a more powerful technique known as **motion prediction**

Motion Prediction

- Each 16×16 block of pixels predicted in new frame based on nearby blocks in prior frame
- Involves search for best match in preceding frame
- Look for a translation that minimizes difference function $E(\Delta x, \Delta y)$ between two consecutive frames I_1 and I_2

$$E(\Delta x, \Delta y) = \sum_{x,y} (I_1[x + \Delta x, y + \Delta y] - I_2[x, y])^2$$

- Each 16×16 block of image that is highly similar to a neighboring block
  - represented by a predictive displacement function

Motion Prediction

- Frames in the stream are compressed
  - independently, or
  - relatively to neighboring frames
- **I-frames**
  - compressed standalone
  - rate of I-frames not necessarily related to compression
  - in streaming video, I-frames play a key role for scrubbing
  - live streams have a lower frequency of I-frames
    - reason you cannot change your digital channel instantly

Motion Prediction

- Each 16×16 block of image compressed by analyzing and saving prediction error

$$I_e(x, y) = I_t(x, y) - I_i(x + \Delta x, y + \Delta y)$$

where

- I_t: current frame in the video stream,
- I_i: reference frame
- Δx and Δy: motion prediction vectors for this macro block
- I_e: 16×16 block image error

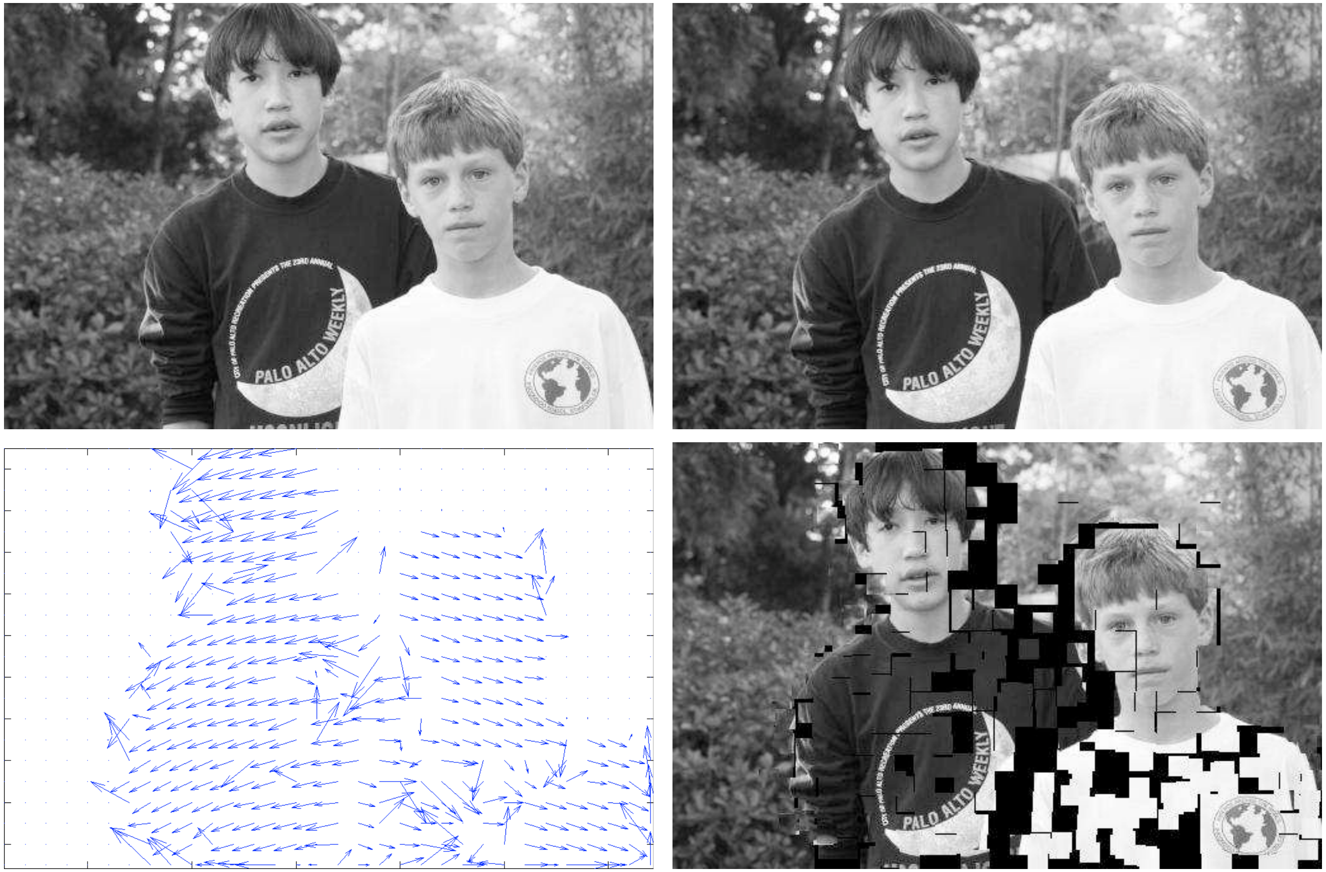
Motion Prediction

- Function $E(\Delta x, \Delta y)$ and motion prediction hide three important details

  1. summation usually carried out over a macro block, yielding an estimate of which pixel values are best found in the other image
  2. in brute-force implementation, cost grows with square of maximum distance considered
  3. best motion-prediction vectors for any one macro block are independent of any other block

Motion Prediction

- Example of motion prediction vectors between two images



MPEG Standards

- Much of audio-visual content in multimedia systems encoded in **MPEG**
  - standard for compression and delivery of multimedia
  - created by the International Standards Organization (ISO) and the International Electro-Technical Commission (IEC)
  - not just a standard but a family of standards: MPEG-1, MPEG-2, MPEG-4, MPEG-7, MPEG-21

MPEG-1

- Standard started in 1988 and approved in late 1992
- In 1988 video did not fit on common storage media
  - applications like Video CD and CD-ROM drove development of MPEG-1
  - challenge: fit audio and video in storage media then used exclusively for audio
  - interactivity needed to support random access

MPEG-1

- Designed to achieve video quality comparable to VHS
  - 1.5M bps (bits per second)
  - a frame size of 352x240
  - 29.97 frames per second
  - stereo audio at 192 bps
- Efficient compression algorithm that can be decoded in real-time
- Widely adopted, playable in most computers and DVD players
- **MP3**
  - level 3 MPEG-1
  - most popular audio-compression standard

MPEG-2

- Developed to provide higher quality and bandwidth than MPEG-1
- Bit rates
  - 3 – 15 Mbps for broadband
  - 15 – 30 Mbps for HDTV
- Efficiently compresses interlaced video
  - most significant enhancement from MPEG-1
- MPEG-2 scales well to HDTV resolution and bit rates
  - makes an MPEG-3 standard unnecessary
- MPEG-2 decoders also decode MPEG-1 bit streams
  - also, provides multi-channel surround sound coding

MPEG-2

- MPEG uses information from neighboring areas to compress specific areas of a frame
- Motion vector captures movement of target area and makes prediction easier
- Prediction involves more than just looking at previous frames
- Three types of frames are used: I, P, and B
  - I-frames are denoted **intra frames**
  - P-frames and B-frames are denoted **inter frames**

MPEG-2

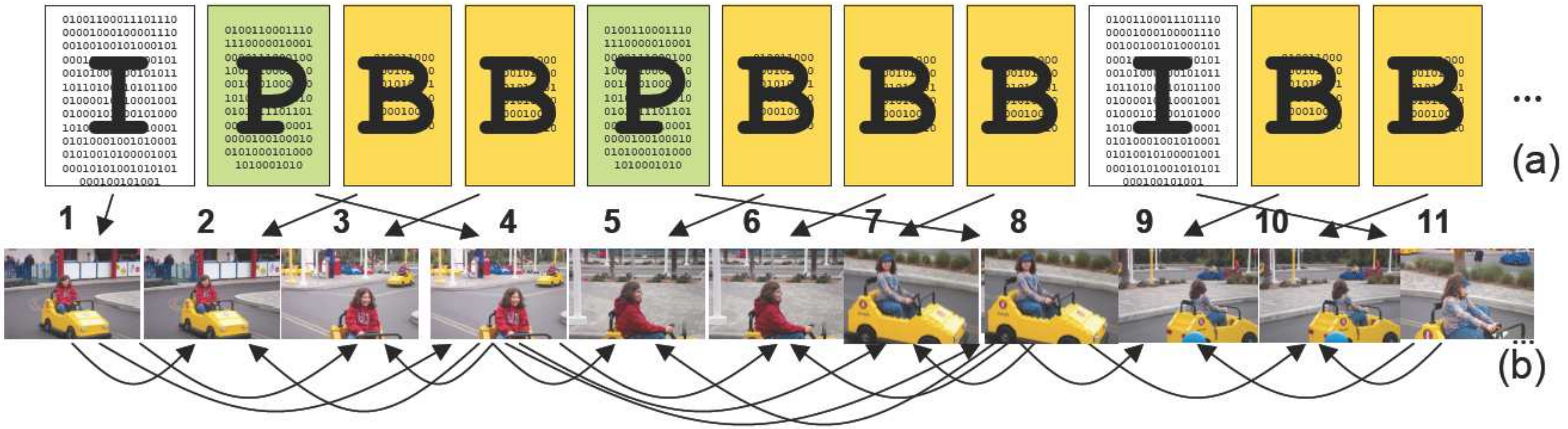
- I-frames do not reference any other frame
  - simply coded as a still image
  - consequently, decoding can start at any I-frame
- I-frames provide anchors into the video stream
  - constitute the entry points for random access
  - provide a fresh start from the point of view of error recovery

MPEG-2

- P-frames are compressed
  - reconstructed using forward prediction
  - reconstruction requires either previous I-frame or previous P-frame
  - from one of previous frames, along with motion prediction vectors, calculate the new frame
- B-frames, or bidirectional frames, are unique
  - use both forward and backward predictions
  - reconstructed from closest past I-frame or P-frame, and closest I-frame or P-frame in the future
- MPEG prediction can be described from the point of view of the encoder or the decoder

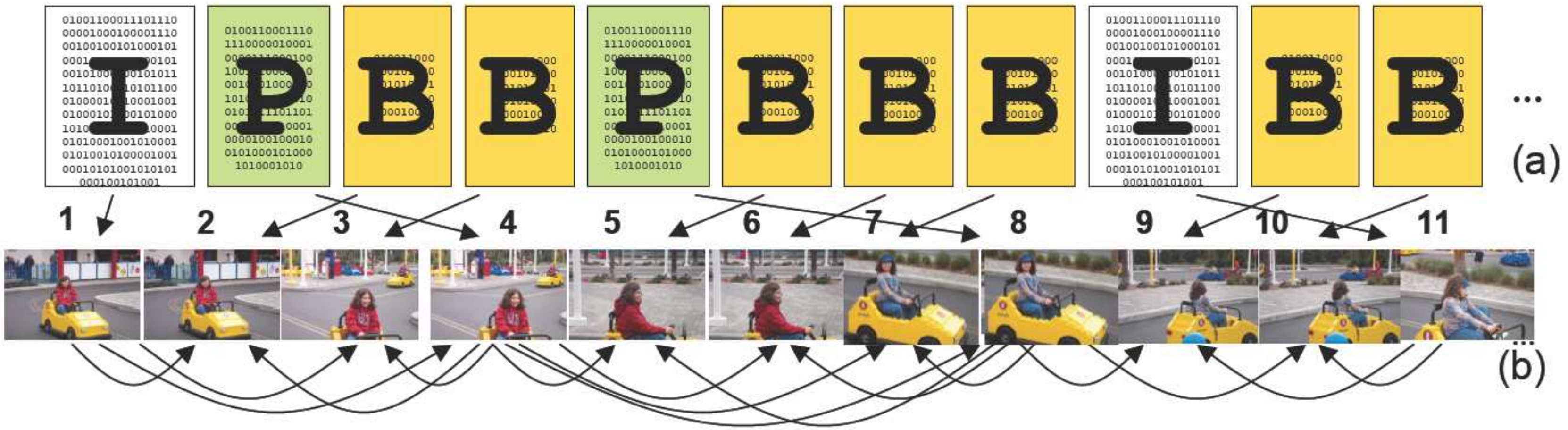
MPEG-2

- Typical sequence of frames in encoded stream and frame dependency



MPEG-2

- Coding/transmission order of frames must be different from display/playback order
  - otherwise, decoder would have to suspend reconstruction of B-frames until reference P or B frames arrive
- Display sequence displayed can be transmitted as $IBBPBBBI$



MPEG-2

- Decoder needs three buffers
  - one for forward prediction
  - one for backward prediction
  - one for the reconstructed image
- Each block in a P-frame can be intra-coded or predicted
- Each block in a B-frame can be intra-coded or predicted

MPEG-4

- Originally targeted at low-bit rate video applications
  - scope was afterwards expanded
- MPEG-4 scales
  - can perform under a wide range of bit-rates, from a couple of Kbits/sec to 10Mb/sec
  - can use object-based compression
    - first standard to reach beyond block-based compression

MPEG-4

- Vision for MPEG-4 was
  - to provide a bridge between Web media and conventional media
- MPEG-4
  - enables interaction with objects within the scene
  - supports integration of natural and synthetic media
  - provides compression for speech, audio and video

MPEG-7

- First MPEG standard that is not about compression
  - about semantics of media
  - describes metadata about the content, not the content itself
  - can be seen as a content description standard
  - uses a Description Definition Language (DDL) and is defined in XML

MPEG-21

- Open framework for multimedia delivery and consumption
  - addresses the challenges of describing the intellectual properties rights associated with a piece of multimedia
- Fundamental unit of transaction is a **Digital Item (DI)**
  - combination of audio, images, video, and text metadata
  - captures relationship among these components

MPEG-21

- **Rights Expression Language**
  - standard to allow sharing digital rights information among various players involved
- MPEG-21 provides framework where two people can interact with one another
  - they can manipulate, trade, consume,and access, a Digital Item smoothly and efficiently
- Hope is that such transparent interaction discourages illicit file sharing