Content-Based Temporal Processing of Video

Rob Joyce
Princeton University
August 5, 2002

Advisor: Bede Liu
Readers: Wayne Wolf, Bradley Dickinson
Motivation

• Much of the “content” information we want to extract from video is temporal

• Benefits in
  - Transmission
  - Browsing
  - Search engines
  - Compression / transcoding

• Manual annotation often impractical
  - Live streams
  - Multi-stream setups
  - Low-budget productions
  - One-time use

• Temporal info allows for “assisted” manual annotation
Thesis Outline

- Gradual Transition Detection in Video
- VBR Bandwidth Prediction
- Multimodal Processing Issues
- Association Matrices
- Temporal Structure
- Hierarchical Visualization
- Conclusions & Future Work
Gradual Transition Detection

Gradually change every pixel in the same way

Abruptly change evolving subsets of pixels

Dissolve / Fade

\[ f_k(x, y) = \alpha_k h_k(x, y) + (1 - \alpha_k) g_k(x, y) \]

Compute correlations of frame differences; will be 1 during ideal dissolves.

“Wipe”

\[ F_k(p) = \frac{||I_k|| + E_{G,k}(p)}{N} G_k(p) + \left(1 - \frac{||I_k|| + E_{H,k}(p)}{N}\right) H_k(p) \]

Compute correlations of frame histogram differences; will be 1 during ideal wipes.

VBR Bandwidth Prediction

- Use shot boundaries as renegotiation points \cite{Bocheck98}
  - Traffic after boundary has little relation to that before
- Use short-term observation of traffic and content statistics (AC coeffs., MV magnitudes, etc.)
- Determine traffic descriptor with neural network:

\[ \text{traffic prediction neural network} \]

• Multimodal Processing (segment distance metrics, norm.)
• Association Matrices (representation, sequence det.)
• Temporal Structure (transitive links, threading)
• Hierarchical Visualization
Multimodal Processing

- Segment audio & video independently
- Audio segmentation: speaker-based [Gish, Wyse, Siegler ’92-’97]
  Difficult without speaker training, due to variance in cepstral coeffs., plus addition of noise/music.

- Audio (speaker) segment distance metric:
  - Somewhat better, given segment boundaries
  - $L^2$ distance between cepstral mean of each segment
  - Still only detect ~30% of the same/“similar” speaker pairs

- Temporal video segment distance metric:
  - Distinct from a search engine shot distance metric
  - Does shot $k$ proceed from $j$? Two key frames per shot:
    \[ D_{j,k} = d(K_{exit}(j), K_{enter}(k)) \quad j < k \]
    \[ \text{regional-histogram image distance} \]
A/V Distance Normalization

• Need to make meaningful comparisons between audio and video distance metrics

⇒ Normalize such that an audio segment distance of $d$ is perceptually equivalent to a video distance of $d$
  – Determining exact mapping between measurements or statistics and perceived distance difficult

• Roughly quantize into three classes, then assign nominal normalized distances to each
  1. “Same”: same source in the same context (0.0).
  2. “Similar”: same source, recorded in different manners or different conditions (0.3).
  3. “Different”: No clear relationship between the segments (1.0).

• Detection problem; priors depend on separation
Association Matrices

- Want single representation of distance information using multiple modalities and metrics
- Audio distance matrices via short-term statistics [Foote '99]
  - Visualization of self-similarity, links, and common sequences
- Motivated by Foote’s distance matrices, formulate a general “association matrix” among segments of same & different modalities:

```
segment set vector \( S = [S_{m_1} \quad S_{m_2} \quad \ldots \quad S_{m_K}] \)

association matrix element \( a_{i,j} = D_{i,j}(s_i, s_j) \)
```
Simplified A/V Association Matrix

- From here on, use $K=2$:
  - $m_1$ is the two-key-frame video shot distance
  - $m_2$ is the cepstral-mean audio segment distance

$$A = \begin{bmatrix} D_{VV} & D_{AV}^T \\ D_{AV} & D_{AA} \end{bmatrix} \quad S = \begin{bmatrix} S_V & S_A \end{bmatrix}$$

- $D_{AV}$ is $1$ minus the fraction (of the shorter segment) that the corresponding audio and video segments overlap in time (other metrics possible…)
- $D_{VV}$ and $D_{AA}$ symmetric (for careful definition of $m_1$, $m_2$)
- $D_{AV}$ almost all ones except near diagonal

- For comparisons to be meaningful, all distances must be perceptually normalized!
Example A/V Matrix

7 minute segment of the “Charlie Rose” PBS talk show

Video shots
1-10: conversation between host and guest 1
11: logo
12-15: host speaks
16-30: guest 2
31-38: mostly guest 2 speaking with game screens as video

Colors:
- “same”
- “similar”
- “different”
Time Normalized “Matrix”

Columns/rows scaled according to segment duration

0-55s: conversation between host and guest 1
55-65s: logo
65-110s: host speaks
110-300s: mostly guest 2 speaking with game screens as video

Colors:
- “same”
- “similar”
- “different”

FPO: Content-Based Temporal Processing of Video
Rob Joyce, Princeton University
Superimposed A/V Matrices
Superimposed A/V Matrices

8.5 minutes of CBS’s “The Late Show with David Letterman”

- Monologue
- Desk shots
- Interview (mostly host speaking)
- Pre-recorded sequence

Correlated Video Audio Both
Superimposed A/V Matrices

7.5 minutes of CBS-2 local evening news

[Graph showing superimposed A/V matrices with labels for Newsdesk, Anchor shots, Correlated Video, Audio, Both]
Idiomatic Sequence Detection

- Can interpret local temporal properties of streams as matrix properties, allowing easy detection algorithms

E.g.:

DIALOG

ACTION

- “different” segment pairs (+0.5)
- “same” segment pairs (-0.5)
- don’t care pairs (0.0)

Use regional correlations along diagonal to check (find largest subsequences matching prototype)
Idiomatic Sequence Detection

• Not all prototypes are “self-similar”:

  \[
  \begin{array}{c}
  \times \times \times \times \times \\
  \times \times \times \times \times \\
  \times \times \times \times \times \\
  \times \times \times \times \times \\
  \times \times \times \times \times \\
  \hline
  \times \times \times \times \times \\
  \times \times \times \times \times \\
  \times \times \times \times \times \\
  \times \times \times \times \times \\
  \times \times \times \times \times \\
  \end{array}
  \quad \text{then}
  \begin{array}{c}
  \times \times \times \times \times \\
  \times \times \times \times \times \\
  \times \times \times \times \times \\
  \times \times \times \times \times \\
  \times \times \times \times \times \\
  \hline
  \times \times \times \times \times \\
  \times \times \times \times \times \\
  \times \times \times \times \times \\
  \times \times \times \times \times \\
  \times \times \times \times \times \\
  \end{array}
  \]

  □ - “different”
  ■ - “same”
  × - don’t care

• Not all prototypes are “local”:

  \[
  \begin{array}{c}
  \times \times \times \times \times \\
  \times \times \times \times \times \\
  \times \times \times \times \times \\
  \times \times \times \times \times \\
  \times \times \times \times \times \\
  \hline
  \times \times \times \times \times \\
  \times \times \times \times \times \\
  \times \times \times \times \times \\
  \times \times \times \times \times \\
  \times \times \times \times \times \\
  \end{array}
  \quad \text{then}
  \begin{array}{c}
  \times \times \times \times \times \\
  \times \times \times \times \times \\
  \times \times \times \times \times \\
  \times \times \times \times \times \\
  \times \times \times \times \times \\
  \hline
  \times \times \times \times \times \\
  \times \times \times \times \times \\
  \times \times \times \times \times \\
  \times \times \times \times \times \\
  \times \times \times \times \times \\
  \end{array}
  \]

  Low-threshold
  (at least one seg.)
## Sequence Detection Results

25 minutes of digitized television

<table>
<thead>
<tr>
<th>Idiomatic Sequence</th>
<th>against ground truth</th>
<th>against assoc. matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$P(D)$</td>
<td>#FA</td>
</tr>
<tr>
<td>Dialog</td>
<td>(video)</td>
<td>6/6 (100%)</td>
</tr>
<tr>
<td></td>
<td>(audio)</td>
<td>4/7 (57%)</td>
</tr>
<tr>
<td>Action</td>
<td>(video)</td>
<td>6/6 (100%)</td>
</tr>
<tr>
<td></td>
<td>(audio)</td>
<td>3/4 (75%)</td>
</tr>
<tr>
<td>Return to Anchor</td>
<td>(video)</td>
<td>5/11 (45%)</td>
</tr>
<tr>
<td></td>
<td>(audio)</td>
<td>0/2 (0%)</td>
</tr>
<tr>
<td>Character Introduction</td>
<td>(video)</td>
<td>15/23 (65%)</td>
</tr>
<tr>
<td></td>
<td>(audio)</td>
<td>11/19 (58%)</td>
</tr>
<tr>
<td>Character Departure</td>
<td>(video)</td>
<td>14/23 (61%)</td>
</tr>
<tr>
<td></td>
<td>(audio)</td>
<td>11/19 (58%)</td>
</tr>
<tr>
<td>Independent Event</td>
<td>(video)</td>
<td>2/3 (67%)</td>
</tr>
<tr>
<td></td>
<td>(audio)</td>
<td>8/13 (62%)</td>
</tr>
<tr>
<td><strong>Totals</strong></td>
<td></td>
<td>97/157 (62.2%)</td>
</tr>
</tbody>
</table>
Multimodal Temporal Structure

- Beyond idiomatic sequences, how does plot manifest itself in connections between a/v shots and scenes?
- Need some method of associating visually/aurally distinct segments that are topically related ⇒ transitivity (e.g., V1 → V4 → A3 → A9 → V6)
- Many streams don’t have admit a simple segmentation into shots and scenes as groups of shots (e.g., sports)
- Ideally, want to determine (transitive) chains of association to infer plot characteristics
Prior Joint A/V & Structure Work

- Summarization of video-only streams by clustering temporal sequences
  - Image-based dialog, action, etc. sequences [Yeung 1996]
  - Motion/histogram-based clustering [Rui 1998]

- Detection and visualization of self-similarity in audio
  - Distance matrices from short-term statistics [Foote 1999]

- Use of video shot boundaries and audio
  - Coincidence of audio and video boundaries [Sundaram 2000]
  - Audio classification (speech, silence, music, “noise”) and heuristic rules to find scene breaks and commercials [Saraceno 1998]
  - Audio classification + low-resolution frames for dialog, action, “story” sequences [Saraceno 1999]
Association “Graphs”

- Transitive links between segments important

- Motivates graphical interpretation of assoc. matrices:
  - Each segment (audio or video) is a node
  - Edge weight between nodes is the normalized distance

- Shortest Path/Dijkstra algorithm: what sequence of events led from A to B?

- Breadth-first search: which segments are “related” to this one, ignoring the number/edge weights of intervening segments

- Possible edge/path restrictions:
  - Forward-time only, reverse-time only (causal)
  - Contiguous/overlapping segment pairs only (breaks bad)
  - Direct/transitive weights below some threshold (say 0.9)
Pruning Graphs

• Potentially large number of links
  – Problematic & misleading in plots, analysis
  – Even worse if transitive paths are added as “links”

• Use a “memory-based” model:
  For a given segment, claim that
  – The most recent same/similar segment is likely the first one recalled by a human viewer (even if via transitive links)
  – Segment introducing the type like the current one, or an important similar segment in the past, may also be recalled (former better for clustering, latter difficult to define)

• Implementation:
  – For each segment, find most recent same/similar segments using breadth-first search on edges below some threshold
  – Also run on time-reversed stream if possible (delayed causal impl. possible)
Plot Thread Model

• Ideally, the diverging and converging paths of the memory-based graph will follow the semantic chains of plot through the media stream.
• Aural/visual cues added by director help (detectable?)
• Call independent yet simultaneous chains of association “threads”:
• Merge/split nodes often particularly important
Threading Heuristic

• Assign thread numbers to nodes:
  – Start with the “memory-based” pruned association graph
  – If a node $j$ has a single parent, and
    • The parent has only one child ($j$), assign $j$ to the same thread as the parent
    • Otherwise, the parent is a split node, and assign $j$ to a new thread
  – If a node $j$ has a multiple parents (or none), assign $j$ to a new thread

• This scheme over-allocates threads, but transitive links make it difficult in general to
  – Know if the child in a merge should be associated with a particular parent, or none at all (a new thread)
  – Know if the parent in a split should be associated with a particular child, or none at all (a separate thread)
Thread Reassignment

• To combat the over-allocation of thread numbers, re-use thread numbers where they are no longer used.
• Use greedy (fast but sub-optimal) procedure guaranteeing that once a thread starts, it stays on the same parallel line and is never interrupted:
  – Determine the first and last node number occupying each thread.
  – For each thread number \( t \), find the lowest thread number \( < t \) that has a last-occupancy time less than \( t \)'s first-occupancy time; if one exists, reassign \( t \) to this new (lower) thread and update the lower thread’s last-occupancy time.
  – Eliminate any unused thread numbers.
• May well put different “plot” threads on the same parallel line, but the alternative is unwieldy graphs.
Hierarchical Visualization

• Conflicting goals in visual summaries:
  – Should be compact (“at a glance”) and intuitive
  – Should be capable of answering rather detailed questions

• Naively plotting whole graphs/trees is unwieldy, even after the memory-based pruning algorithm
  ⇒ Use hierarchical methods

• Other goals:
  – Intuitively show temporal progression
  – Automatic graph layout
  – Clearly show which segments are concurrent
  – Concurrent plot “threads” should be in parallel
Prior Visualization Work

- Scene transition graphs and clustering of shots, semi-automatic graph layout [Yeung 1996]
- Linear browser with speaker tracks alongside, for editing applications [Toklu 2000]
- Hierarchical feature-presence vs. time plots for fixed or shot segments [Ponceleon 2001]
- Complementary iconic and episodic pair of interfaces (and associated semantic issues) [Davis 1994]
- More generally, multiple streams of cause & effect (non-video) [Tufte 1997]
Generating the Hierarchy

- Select nodes for display at each “level” of hierarchy
- When “zooming in” from a node at level $l$, present a level $l + 1$ graph centered on the selected node
  - Graph edges determined by memory-based transitive path search (including hidden nodes)
  - Allow user to easily pan to other areas, like a map
  - Alternative: show only nodes near the one clicked-on
- Rank nodes by “importance” to determine in which graphs they appear
  - Inclusion in idiomatic sequences, particularly introductions, merges, and splits
  - Alt.: Non-temporal characteristics (motion, audio volume, …)
- Level $l$ graph includes all nodes of rank $\leq l$
## Node Rankings

Our rank assignment:

<table>
<thead>
<tr>
<th>Rank</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>First and last audio and video segments</td>
</tr>
<tr>
<td>2-4</td>
<td>Character introduction segments</td>
</tr>
<tr>
<td>5-7</td>
<td>Path merge segments</td>
</tr>
<tr>
<td>8-10</td>
<td>Path split segments</td>
</tr>
<tr>
<td>11-13</td>
<td>Topic change sequences’ first segments (i.e., where the change is)</td>
</tr>
<tr>
<td>14-16</td>
<td>Return-to-anchor sequences’ first segments (i.e., the “anchor”)</td>
</tr>
<tr>
<td>17-19</td>
<td>Interlude/commercial sequences’ first and last segments</td>
</tr>
<tr>
<td>20-22</td>
<td>Character departure segments</td>
</tr>
<tr>
<td>23-25</td>
<td>Action sequences’ first and last segments</td>
</tr>
<tr>
<td>26-28</td>
<td>Dialog sequences’ first two segments (i.e., both participants)</td>
</tr>
<tr>
<td>29</td>
<td>Segments aligned in audio and video</td>
</tr>
<tr>
<td>30</td>
<td>Video shots &gt;7 seconds and audio segments &gt;10 seconds</td>
</tr>
</tbody>
</table>

(2-4: 2 for coincident audio and video, 3 for video only, 4 for audio only)
Rank Equalization

• May have too many ranks, or a number of empty ranks
• Set a constant growth factor $\gamma$: enforce that there are exactly $4^{\gamma^{k-1}}$ nodes of rank $k$
  – Order nodes by rank, shuffling nodes of equivalent rank
  – Select first 4 nodes as rank 1, next $4\gamma$ as rank 2, etc.
  – Highest rank may not be full, but corresponding hierarchy level contains all nodes

• Logarithmic nature flattens even the longest streams into a few hierarchy levels ($\leq 8$ for up to 1000 segs.)
• We select $\gamma = 2.0$
Graph Layout (Time)

- Intuitively show temporal progression & concurrence: use time as the horizontal dimension

  Constraints:
  - Minimum/fixed node size (to include thumbnail, times, etc.)
  - Don’t want shortest segment to force extremely wide graphs
  - Nodes should not overlap
  - Desire to align audio and video in time

  ⇒ time dimension will be nonlinear

- Algorithm:
  - First, pack all video nodes in order by start time, left to right
  - Place all audio nodes by interpolating video timestamps
  - Working from left to right, where two audio nodes overlap, shift all video and audio nodes to the right to make space
  - Use faint lines to indicate constant time intervals
Graph Layout (Vertical)

- Segments’ **vertical positions determined by thread numbers** (in each modality)
- For simplicity, place video and audio nodes independently, all video above all audio
- Cross-modality information is implicit, because each modality’s edges are determined using the memory-based transitive links
- Further cross-modality information: indicate overlapping audio/video segments explicitly with edges
Plotting the Graph

- DC+2AC thumbnails for video segments
- “Thumbnails” for audio segments?
- Incorporate other segment/edge info?
- Use scalable vector graphics (SVG) W3C standard for uniform web-based interface with easy panning
Hierarchical Graph Demo

SVG Graph Examples

Rank 3 (All Segments) Summary for nbcnews5

NBC Nightly News, June 21 1999

Stream duration: 56.92 seconds
Graph creation date: 28 Jun 2002
Hierarchy depth: 3 levels (zoom factor is 2x)

Video segments: 11
Audio segments: 6

Quick navigation:  Top level summary | Show all segments | Highlight previously-selected segment

Click on a node to go deeper in the hierarchy, or hold down alt/option and drag to pan.

Red nodes are video shots, blue are audio (speaker) segments. Grid lines are percentages (in time) through the duration of the video.

Rob Joyce, robjoyce@princeton.edu, SDate: 2002/06/07 13:09:10 8
Summary

- Gradual transition detection in video: wipe/dissolve
- Application to VBR bandwidth prediction per shot
- “Perceptual” normalization of audio and video segment distances, multimodal cross-distances
- Association matrix representation of seg. distances
- Detection of idiomatic sequences from assoc. matrix
- Graph interpretation, incorporating transitive links
- Memory-based pruning, assignment to “plot” threads
- Node ranking ⇒ Hierarchical graph representation of multimedia streams
Some Future Directions

- Incorporation of long-term structure info in VBR bandwidth prediction (similar shots, similar traffic)
- Better audio segmentation, distance metrics
- More interesting cross-modality distance metrics (face detection/lip movement…?); other modalities
- Analysis of the effects of segmentation errors on the distance matrices and idiomatic sequence detection
- Smarter use of transitive links in threading
- Incorporate other information in node ranking (non-temporal statistics, DB lookup for characters, etc.)
Thanks!

Thanks to:

Profs. Bede Liu, Wayne Wolf, Bradley Dickinson, S-Y Kung

Contributors (discussion, ideas, testing/code):
Min Wu, Peng Yin, Scott Craver, Prof. Perry Cook

Graph Demo:
http://www.ee.princeton.edu/~robjoyce/res/svg/