<table>
<thead>
<tr>
<th>Authors</th>
<th>Presenters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wei Jiang</td>
<td>Armin Samii</td>
</tr>
<tr>
<td>Courtenay Cotton</td>
<td></td>
</tr>
<tr>
<td>Shih-Fu Chang</td>
<td>Images from the interwebs, 2009 paper, and 2010 slides from Dan Ellis</td>
</tr>
<tr>
<td>Dan Ellis</td>
<td></td>
</tr>
<tr>
<td>Alexander C. Loui</td>
<td></td>
</tr>
</tbody>
</table>
Goal of the Work

- Categorize generic videos
  ('Semantic Concept Detection')
- Uses Kodak's consumer benchmark video set
Method Overview

1) Extract features
   - Video contents
   - Motion tracking
   - Audio

2) Create codebook
   - Prototypical features in a concept

3) Predict
   - Match new examples to items in codebook
Label is penguin, but most of the image has nothing to do with a penguin.
Quality, perspective, and lighting varies across videos.
Motivating Problems (3)

Need a way to relate sound and video
Motivating Problems (4)

Sounds can come from objects out of the frame.
Downsides to Related Work

- Treat audio and video separately
  → Important to relate them (e.g. the cat)
  Solution: combine audio+video into one unit

- Look at static regions in the image
  → No movement data
  Solution: track movement of regions over time

(details to come)
Main Contribution: S-AVAs

- 'Short-term Audio-Visual Atoms'
  - A 1-second slice of video:
    - 137 video features
    - 4 motion features
    - 152 audio features

- Purpose:
  Relate a small segment of the video to a concept
Framework

Short-term Video Slice

Shot-term Region Tracks by STR-PTRS

Visual Features from Short-term Region Tracks (color, texture, edge, motion)

visual atom visual atom

visual atom

Short-term Window

MP Bases

dictionary

Psychoacoustic Pruning

Audio Representation

Joint Audio-Visual Codebook by Multiple Instance Learning
Goal:
- Detect and track objects

What doesn't work (previous attempts):
- Blob-based tracking (subtracts out the background) → Fails with shaky cameras
- Model-based tracking (manual user initialization) → Too many videos
- Tracking raw pixels → Fails if lighting or perspective change
Tracking

- Alternative: Image segmentation
  - Frame-by-frame
  - Link similar segments ('regions')
  - Track each region independently
Tracking

- Each video slice is 1 second
- 10 frames extracted per slice
- Track all regions over the 10 frames
- Similar regions have similarly-moving points
Short-Term Point Track

- Image features found
  - "Corners, etc."
  - Things that can be easily locked on
- Kanade-Lucas-Tomasi (KLT) Tracker
  - Tracks based on image gradient
  - Robust against changes in lighting
  - Allows affine image changes
- Results in a set of feature tracks
Short-Term Region Track

- Segmentation by color and texture
  - Jseg tool
  - Each frame segmented independently
- Robust against mistakes
  - Errors are averaged across all 10 frames
Short-Term Region Tracks

- Common mistakes
  - One segment split into two
  - Two segments combine into one
- Solution: replication
  - Keep all potential tracks
Short-Term Region Tracks

- To find matching regions across frames:
  - For each region in frame T>1,
    - Calculate KLT distance to all regions in time T-1
    - Choose shortest distance
    - If distance is within some threshold:
      - Mark the two as the same region
    - Else
      - Create a new region track starting at time T
  - Remove all short region tracks
Negligible failures:

- Objects entering in middle of video slice
  → The object will be there in next slice
  → If not, the object probably wasn't important

- Scene changes
  → Most of the data will be discarded by remove
Short-Term Region Tracks

- Choosing length of a video slice

**Trade-off:**
- Longer → more audio/video information
- Shorter → more accurate motion tracking
Recap

- What we have so far:
  - A set of regions for every frame
  - Links between regions marked as a match (and an *implicit* motion vector)

- Still nothing for a machine learning algorithm
Visual Features

- Color moments in HSV space
  - Mean, standard deviation, skewness
  - 9 dimensions
- Gabor texture
  - 48 dimensions
- Edge direction histogram
  - 73 dimensions
Where We Are

- For each video slice:
  - For each frame in each slice:
    - For each region in each frame:
      - Extract visual features

- Total: 130 features per region

- For each region:
  - Average across frames to obtain feature vector (for the region track)
Trajectory Features

- Calculate optical flow for each pixel
  - Lucase Kanade method
- Divide optical flow direction into four quadrants
- Use 4-bin histogram as a 4D feature

Bins: \{4, 1, 0, 0\}
Audio Representation

- Need to be:
  - Compact
  - Robust against noise
  - Able to match similar audio in different environments

- Algorithm:
  - Matching Pursuit decomposition
Matching Pursuit Decomposition

- Have a dictionary of basis functions
- Try to describe signal with a set of them

Algorithm:
- Find function that best matches current signal
- Subtract that signal
- Repeat 500 times

Properties:
- Focuses on energy peaks
- Relatively invariant to background noise
Matching Pursuit Dictionary

- Use Gabor function:

\[ g_\gamma(t) = K(\gamma)e^{-\pi\left(\frac{t-u}{s}\right)^2}\cos(\omega(t-u)+\phi) \]

- By modifying \( u, s, w \):

Some generated Gabor functions
Updated parameters:

\[ g_\gamma(t) = K(\gamma)e^{-\pi \left(\frac{t-u}{s}\right)^2} \cos(\omega(t-u) + \phi) \]

Updated parameters:
- \( \phi \): Translate in time
- \( \omega \): Scale in length

updated by eight powers of two
at 16kHz, 2ms to 256ms
Finding the "best match" function

- For each function in dictionary:
  - Subtract from the current signal
  - Calculate the new energy
- Pick the function that minimizes the new energy
- Save it, subtract, and repeat
Psychoacoustic Masking

- **Principle**
  - Humans cannot perceive a low-energy signal if a high-energy signal of similar frequency is present.

- Remove 30% of the functions that are less noticeable (from the original 500)
Turning MP into features

- Each of the eight length scales gets a histogram
- Each histogram has 19 bins
- Bin sizes are one-third of an octave wide

- 8 histograms * 19 bins = 152 features
- Every region segmentation gets the same audio feature vector
Summary: S-AVA

Short-term Audio-Visual Atom (S-AVA)

<table>
<thead>
<tr>
<th>Short-term region track $r = {r^t}, t = 1, \ldots, T$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_{vis}$ : $d_{vis}$ dimensions (visual color/texture/edge)</td>
</tr>
<tr>
<td>$f_{mt}$ : $d_{mt}$ dimensions (visual motion)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Short-term audio window</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_{audio}$ : $d_{audio}$ dimensions (audio MP hist. &amp; energy)</td>
</tr>
</tbody>
</table>

Figure 6 in the paper
Recap: Framework
Recap: Framework

Short-term Video Slice  \rightarrow \text{Shot-term Region Tracks by STR-PTRS}  \rightarrow \text{Visual Features from Short-term Region Tracks (color, texture, edge, motion)}  \rightarrow \text{visual atom}  \rightarrow \text{Joint Audio-Visual Codebook by Multiple Instance Learning}

Short-term Window  \rightarrow \text{MP Bases}  \rightarrow \text{Psychoacoustic Pruning}  \rightarrow \text{Audio Representation}

?
Available Data

- **Labels**
  - One or more per video
  - In general, only applies to a region

(Not the same concepts in the paper)
What We Want From ML

- For each concept, we want:
  - Several examples of "prototypical" S-AVAs
  - A way of mapping each video (many S-AVAs) to a single binary decision
  - Ideally, if any S-AVA votes "aye," the entire video should be labeled "aye."

One sheep said "aye"
Multiple Instance Learning

- Put 10 video slices into one "bag"
  - 10 seconds of video
  - 100 frames
  - Varying number of regions per 1-second slice
  - For a single concept, bag has a binary label
- Treat each concept separately
- For each concept, will have many "yes" bags, and many more "no" bags
**Intuition**

- Each S-AVA in a bag is a **marble**
- For each concept, want to find **marbles** which best represent the concept

**Want a magic marble** that is:

- Similar to at least one **marble** in every "yes" bag
- Dissimilar from every **marble** in every "no" bag
Multiple Instance Learning

- Formula to do that:

\[ Q_l = \frac{1+y_l}{2} - y_l \prod_{j=1}^{N_l} (1-e^{-||f_{lj}-f^*||^2_{w^*}}) \]

- \( f^* \) is the magic marble from the previous slide
- Optimize \( f^* \) with Expectation-Maximization
- With optimal \( f^* \), formula will:
  - For a negative bag: Be larger if \( f^* \) is far away from ALL instances
  - For a positive bag: Be larger if \( f^* \) is close to ANY instance
Multiple Instance Learning

- We now have an $f^*$ which describes each concept
- Need to learn boundary:

Key: Positive labels  Negative labels  $f^*$
Support Vector Machines:
- Find the hyperplane around the point that separates positive and negative instances

Key:
- Positive labels
- Negative labels
- $f^*$
- Separating hyperplane
Multiple Instance Learning

- But, not always so easy:

  - Need more hyperplanes to describe this (and thus more $f^*$s)

Key:
- Positive labels
- Negative labels
- $f^*$
- Separating hyperplane
Multiple Instance Learning

- Can create by weighting each feature differently
- Boosting:
  - If an algorithm is better than random, can "boost" it
  - Initialize all weights to be equal
  - Run algorithm
  - Reweight all mistakes to be more important
  - Repeat until satisfaction

Convergence of AdaBoost
Call each $f^*$ a **codeword**, and each set of them a **codebook**. (Each concept has a codebook.)

**Key:**
- Positive labels
- Negative labels
- $f^*$
- Separating hyperplane
Recap: MIL

- We look at every 10 second segment of video
- Each has many S-AVAs
- If an S-AVA is close enough to a prototype, label it "yes" for that concept.
Results

- Ran trials on Kodak Consumer Benchmark Video Set

- AP: Average Precision
  - How well they performed on one concept

- MAP: Mean Average Precision
  - How well they performed across all concepts
Results

- Over 120% MAP increase
  (compared to static region segmentation without temporal tracking)

- 8.5% MAP increase when combining audio and video
  (compared to self)

- Many concepts achieve more than 20% correctness
Results: against others
Results: against self
Some Magic Marbles
Some Magic Marbles
Ideas for Future Work

Their ideas
- Sounding regions
- Multiple-region tracks
- Smart segmentation of video slices to avoid hitting scene transitions

My ideas
- Overlapping S-AVA segments
- Use multiple lengths of S-AVAs to get both temporal data and A/V details
Questions?