INRIA LEAR-TEXMEX: Copy detection task
Introduction

- INRIA participation in 2008: top results on all transformations
  - focus on accuracy + localization

- Video:
  - same system as in 2008:
    - *An image-based approach to video copy detection with spatio-temporal filtering*
    - + parameter’s optimization

- Audio: new system (no audio in 2008’s evaluation)
  - audio descriptors computed with standard package (spro)
  - novel approximate nearest neighbor search method

- In this talk:
  - brief overview of our video and audio systems
  - focus on our ANN method
  - comments on our results
Short overview of our video system: key components

- Local descriptors: CS-LBP
  - Heikkila et al., PR’2010

- ANN search: Hamming Embedding
  - Jégou et al., ECCV’08

- Score regularization:
  \[ s_i = s_i \times \left( \frac{s_i}{\max_j s_j} \right)^\alpha \]

- Weak geometric consistency
  - Jégou et al., ECCV’08

- Burstiness strategy + Multi-probe
  - Jégou et al., ICCV’09

- Spatio-temporal fine post-verification
  - Douze et al., IEEE TMM’10
Short overview of our audio system: key components

- **Descriptors**
  - Filter banks
  - Compounding
  - Energy invariance
  - 1 vector /10 ms

- Online package: [https://gforge.inria.fr/projects/spro](https://gforge.inria.fr/projects/spro), filter banks, MFCC, etc

- Novel ANN search based on compression paradigm: see next slides

- Temporal integration: Hough voting scheme (votes in histogram $\Delta t = t_b - t_q$)
Video parameter optimization

OBJECTIVE: improve precision with "reasonable" cost w.r.t. efficiency

- Decreasing detector threshold
  - number of descriptors $\uparrow$
  - complexity $\uparrow$
  - precision $\uparrow$ (with HE)
  - threshold: T200 or T100

- Describe flip/half-sized frames
  - on database side only
  - threshold: H200 or H100

- Multiple assignment (=multi-probe)
  - on query side only

**Observation:**
- half sized and flipped frame help a lot
- small multi-probe (x3) is sufficient

**Table:**

<table>
<thead>
<tr>
<th>query</th>
<th>T200</th>
<th>T200 +H200</th>
<th>T200 +H100</th>
</tr>
</thead>
<tbody>
<tr>
<td>T200</td>
<td>0.483</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T100</td>
<td>0.514</td>
<td>0.568</td>
<td>0.583</td>
</tr>
<tr>
<td>T100+flip</td>
<td>0.627</td>
<td>0.719</td>
<td>0.738</td>
</tr>
<tr>
<td>T100+flip, MA10</td>
<td>0.683</td>
<td>0.749</td>
<td>0.737</td>
</tr>
<tr>
<td>T100+flip, MA3</td>
<td>0.650</td>
<td>0.755</td>
<td>0.761</td>
</tr>
</tbody>
</table>

Note: generic system
- only flipped is specifically to mAP on a validation dataset
Huge volumes to index: approximate nearest neighbor search

<table>
<thead>
<tr>
<th>index size (database)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video, T200</td>
</tr>
<tr>
<td>Video (half, H100)</td>
</tr>
<tr>
<td>Audio</td>
</tr>
<tr>
<td>d=128</td>
</tr>
<tr>
<td>d=128</td>
</tr>
<tr>
<td>d=144</td>
</tr>
<tr>
<td>2.48 billion descriptors</td>
</tr>
<tr>
<td>0.97 billion descriptors</td>
</tr>
<tr>
<td>140 million descriptors</td>
</tr>
</tbody>
</table>

→ Need for powerful approximate search

- Locality Sensitive Hashing: memory consuming, need for post-verification on disk, not very good trade-off between precision/efficiency

- FLANN: excellent results, memory consuming, need for post-verification (on disk given the dataset size)

- We used:
  - Video: Hamming Embedding with 48 bits signature (10B/descriptors+geometry)
  - Audio: Compression based approach → Product quantization method
Indexing algorithm: searching with quantization [Jegou et al., TPAMI’11]

Purpose: approximate NN search with limited memory (and no disk access)

- Search/Indexing = distance approximation problem
- The distance between a query vector \( x \) and a database vector \( y \) is estimated by
  \[
  d(x, y) \approx d(x, q(y))
  \]

  where \( q(.) \) is a fine quantizer

\[ \rightarrow \text{vector-to-code distance} \]

- Distance is approximated in compressed domain
  - typically 8 table look-ups and additions per distance estimation (for SIFTs)
  - proved statistical upper bound on distance approximation error
Indexing algorithm: searching with quantization [Jegou et al., TPAMI’11]

- Combination with inverted file: coarse quantizer to avoid scanning all elements
- Here: MA=3

Efficient search: searching in 2 billion SIFT vectors (with MA=1)
  - This method: 3.4 ms / query vector
  - HE: 2.8 ms / query vector

Fine representation: $2^{64}$ centroids per cell (typically for SIFTs)
Comparison with FLANN [Muja & Lowe’09]

- Tested on 1 million SIFTs

- 1.5 to 2 faster than FLANN for same accuracy

- Memory usage for 1M vectors (according to “top” command):
  - FLANN: > 250MB
  - Ours: < 25MB
NDCR: Comparison between 2008 and 2010

- Huh?! What’s the problem?

  “Bug”: a few false positive videos are returned frequently with very high scores
Results on Trecvid: sub-optimality of our approach

- Problem with audio: pseudo-white segments → corrupts similarity measure
- Fusion based on invalid assumptions:
  - two first runs: audio and video assumed to have similar performance
  - two last runs: audio assumed to be better than video
Conclusion

- We have learned many things this year:
  - actual decision threshold: need for « cross-databases » setting method
  - audio helps a lot (when working)
  - fusion module is very important
    - audio ≠ video, room for improvement by score normalization
    - strong bonus when both agree

- What’s might interest the other participants in what we have done
  - approximate nearest neighbor method for billion vectors

- Online resources:
  - spro: library for audio descriptors
  - Matlab toy implementation of our compression based search method
  - BIGANN: a billion sized vector set to evaluate ANN methods
  - GIST descriptor in C: OK for several copy transformations
    [Douze et al., CIVR’09, IBM Trecvid’10]