ORAND team at Instance Search

Juan Manuel Barrios, Jose M. Saavedra
Felipe Ramirez, David Contreras
ORAND Research Center, Chile.

Instance Search Task, TRECVID.
November 10, 2014
Chilean private company:
- [http://www.orand.cl](http://www.orand.cl)

Research Center in Computer Science + Software Development.
Collaboration

- Collaboration with chilean and international institutions.
Objective: To retrieve shots that contain a given topic (object, person or location) from a video collection.

Video dataset:
- BBC EastEnders collection: 244 videos, 435 hours, 39 million frames, 287 GB, 768x576.

27 Topics:
- 5 Persons (background characters).
- 1 Location
- 21 Objects (10 “this”-object, 11 “a”-object)

Visual examples per topic:
- Video frame + Object Mask in the frame.
- Search Types {A, B, C, D} for {1, 2, 3, 4} visual examples, respectively.
Example

- **Topic 9125**: “this wheelchair with armrests”

- **Expected results (shots in ground truth):**
System Overview

1. Feature extraction
2. Similarity search
   - K Nearest Neighbors Searches.
3. Voting algorithm
   - Computes a score for each shot.
4. Score aggregation
   - Combines result for different modalities.
5. Score propagation
   - Scores are propagated between similar shots.
Example

Set $Q$

Set $R$

Programme material © BBC.
Feature Extraction

Set $Q$

Set $R$
Similarity Search

Set $Q$
Visual examples

Set $R$

K=100
Voting Algorithm

OUTSIDE MASK

NEAR MASK

INSIDE MASK

{ 1-NN, 2-NN, 3-NN, ... 100-NN }
Feature Extraction

- Compute local descriptors for all videos (set \( R \)).
  - Videos sampled at 5 frames/second (~7.8 million frames).
  - CSIFT at Hessian-Laplace keypoints
    - ~1000 vectors/frame, 192-d.
  - CSIFT at MSER keypoints
    - ~700 vectors/frame, 192-d.
  - SIFT at DoG keypoints
    - ~1800 vectors/frame, 128-d.
- Compute local descriptors for visual examples (set \( Q \)).
  - Search type \{A, B, C, D\} require \{30, 60, 90, 120\} images
  - Same three descriptors for each visual example.
Similarity Search

- For each vector in $Q$ locate the k-NN in $R$.
  - Approximate search, $K=100$.
- K-NN search was resolved in a cluster of 64 nodes.
  - Collection $R$ is partitioned into $244 \times 64$ segments.
  - Chilean National Laboratory for High Performance Computing
    - NLHPC [http://www.nlhpc.cl/]
- On each node:
  - Extract vectors on-the-fly from a segment of $R$.
  - Build a kd-tree index and perform approximate K-NN search.
    - FLANN [http://www.cs.ubc.ca/research/flann]
    - MetricKnn [http://www.metricknn.org/]
  - Save the k-NN list and discard vectors and indexes.
- Merge partial results to produce the actual k-NN lists.
Voting Algorithm

- The K=100 nearest neighbors for each vector in $\mathbf{Q}$ are retrieved from shots in $\mathbf{R}$.
- Each nearest neighbor adds one vote to the corresponding shot.
- The vote is weighted by:
  - The rank of the NN:
    - $w_1=0.99^k$ for $k$ in $\{1,\ldots,100\}$.
  - The spatial position of the query vector:
    - Using context: $w_2$ in $\{5, 3, 1\}$ for inside / border / outside the mask.
    - Without context: $w_2$ in $\{2, 1.5, 0\}$ for inside / border / outside the mask.
    - A smooth gaussian weight achieved similar performance than discrete weights.
Score Aggregation

- The aggregation of votes for all visual examples produces the result for a topic:

```
{shot1, shot2, shot3, ...}
```

Topic 9125
Score Propagation

- A scene in television is commonly comprised of shots produced by different static cameras.

- If the object is also static, all the shots from the same camera may contain the object.
Similarity Shot Graph (SSG)

- SSG contains the similarity between any pair of shots in the collection.
- Let $S$ be the number number of shots in the collection
  - SSG is a directed weighted graph with $S$ nodes.
  - The edge between two nodes represents the similarity between the two shots.
- SSG is computed off-line, prior to any topic search.
- BBC EastEnders
  - NIST provided the set of shots
  - $|S| = 471.526$. 
Similarity Shot Graph (SSG)

- SSG is produced by computing the self similarity for shots in the collection.
  - Near duplicated shots according to a weak Video Copy Detector (VCD).

- Weak VCD to compute the SSG:
  - Sample three frames per shot (start/middle/end).
  - Compute a global descriptor for the selected frames.
    - E.g. Color histogram, Gradient histogram, Ordinal Measurement.
  - For each frame locate similar frames (k-NN search).
    - MetricKnn http://www.metricknn.org/
  - Convert distances to similarities
    - Histogram of distances.
  - Aggregation of frame similarity in the same shot.
## Similarity Shot Graph (SSG)

<table>
<thead>
<tr>
<th></th>
<th>-</th>
<th>0.9</th>
<th>0.75</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9</td>
<td>-</td>
<td>-</td>
<td>0.8</td>
</tr>
<tr>
<td>0.9</td>
<td>0.9</td>
<td>-</td>
<td>0.66</td>
</tr>
<tr>
<td>0.75</td>
<td>0.66</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Similarity Shot Graph (SSG)
Score propagation using the SSG

- SSG edges represent the similarity between shots
  - Edge weight is a number between 0 and 1.
- A minimum similarity threshold can be defined to produce a sparse graph.
- It is not guaranteed SSG to be double linked nor the similarity matrix be symmetrical.
  - But it can be forced to be double linked and symmetrical.
- For a given topic, the computed scores are propagated to similar shots according to the SSG:

\[
\text{For each edge in SSG } (\text{shot}_a \rightarrow \text{shot}_b) : \\
\text{score(shot}_b) += \text{score(shot}_a) \times \text{sim(shot}_a,\text{shot}_b)
\]
SSG can also be used to propagate user decisions on interactive systems:

- If user rejects a shot, the SSG decreases the score of similar shots.
- If user accepts a shot, the SSG increases the score of similar shots.
RESULTS AT
INSTANCE SEARCH 2014
Submissions

- Due to an inconvenient with the infrastructure during our participation we were not able to complete the k-NN search.
  - The submissions were built with just 80% of the search.
  - Submitted run CSIFT achieved MAP=0.183 (type D)
- The following results show the MAP achieved by the complete submission [1].
  - MAP was computed using the released ground truth (qrels)
  - Complete k-NN search CSIFT achieves MAP=0.220
  - Score aggregation CSIFT+SIFT achieves MAP=0.223
  - Score propagation by SSG achieves MAP=0.225
  - Interactive submission achieves MAP=0.251

Overall Results

- MAP for the 27 topics, type D (four examples):
Overall Results

- MAP for the 27 topics, type C (three examples):
Overall Results

- MAP for the 27 topics, type B (two examples):
Overall Results

- MAP for the 27 topics, type A (one example):
Overall Results

- MAP for the 27 topics, Interactive:
  - User evaluates first shots (up to the time limit) and the decision is propagated to other shots by the SSG.
Results by Topic

- Topic 9111 “this dartboard”
Results by Topic

- Topic 9108 “these 2 ceramic heads”
Results by Topic

- Topic 9125 “this wheelchair with armrests”
Results by Topic

- Topic 9103 “a red, curved, plastic ketchup container”
Results by Topic

- Topic 9101 “a Primus washing machine”
Search Time

- Average time for all topics:
Conclusions

- We have shown an approach that uses k-NN searches without clustering to descriptors.
  - The search method can easily be divided and distributed into a network of independent machines.
  - We have tested our approach using the Chilean NLHPC.
- The construction of a Similarity Shot Graph can be useful to improve the MAP either in automatic and interactive search.
  - In some topics it may harm the precision.
  - More research is needed in order to understand the scenarios were SSG can be successfully applied.
- The results show the feature extraction and similarity search are the critical processes.
  - Voting algorithm and score propagation are useful but with less impact in the global result than k-NN search.
- This research was partially supported by the supercomputing infrastructure of the NLHPC (ECM-02).
MetricKnn

- MetricKnn is an Open Source Library for performing efficient k-NN search.
  - [http://www.metricknn.org/](http://www.metricknn.org/)
  - BSD License
- It is based on the metric space approach (a generalization of vector spaces).
- It provides an API (written in C) for using Metric Access Methods (MAMs) with predefined or custom distances.
- It can resolve approximate and exact searches:
  - MAMs outperform multidimensional indexes at exact searches.
- Custom distances give more flexibility to define new similarity models, e.g. distance combination [2].