PRISMA-ORAND team: Instance Search Based on Parallel Approximate Searches

Juan Manuel Barrios[1,2] **and Benjamin Bustos**[2]

[1] ORAND Research Center, Chile.

[2] PRISMA Research Group, Department of Computer Science, University of Chile.

Instance Search Task, TRECVID.
November 27, 2012

ORAND

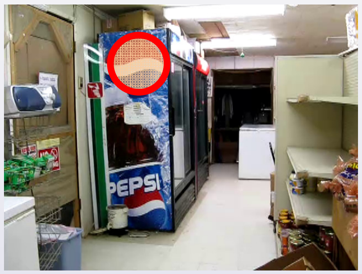
- Chilean private company: http://www.orand.cl
- Research Center in Computer Science + Software Development.
- Links academy and industry in order to address challenging problems (R&D projects).
  - Search and/or detect problems in the industry.
  - Study the state-of-the-art and develop new techniques in collaboration with universities/research groups.
  - Apply software engineering to produce a solution for the end user.

Instance Search 2012

- **Objective**: To find videos of a specific person, object, or place, given visual examples.
- **Video dataset**:
  - Dataset totals: 75.958 videos, 188 hours, 19 million frames, 46 GB.
  - Average video: 9 sec. length, 647 KB, width x height= 573 x 398.
- **21 Topics**:
  - 15 Objects (6 logos, 9 buildings), 5 Locations, 1 Person.
  - On average 4.9 visual examples per topic.

Example

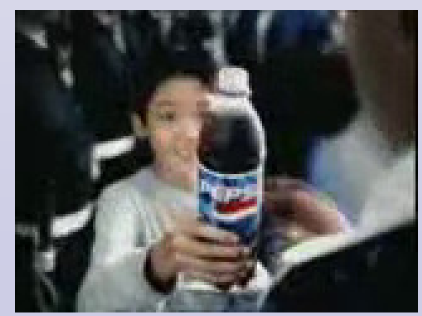
- Topic 9061: “Pepsi logo - circle” (OBJECT)





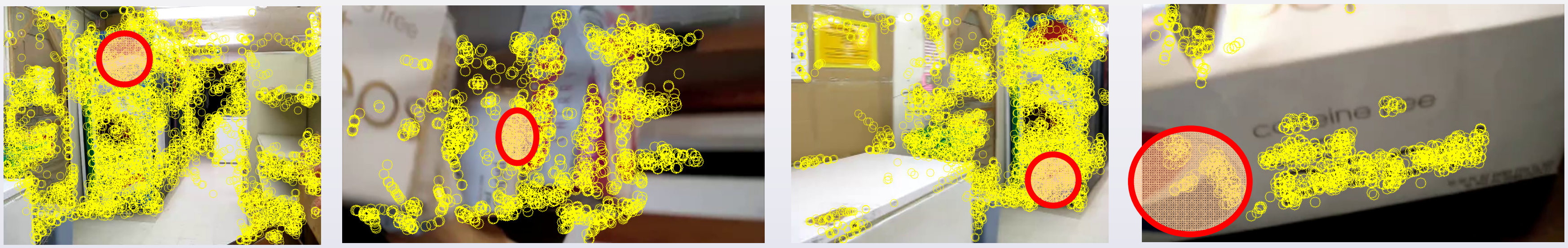
- Expected results (videos in ground truth):





Computing local descriptors

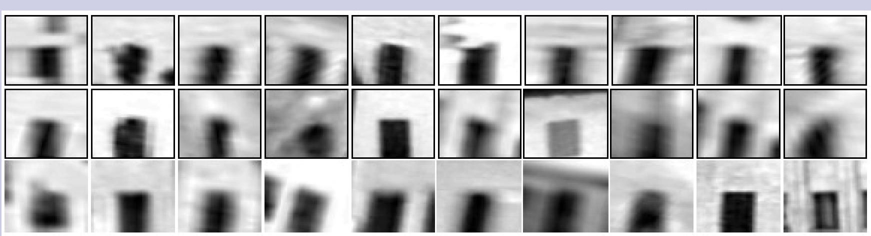
- Topic 9061: “Pepsi logo - circle” (OBJECT)

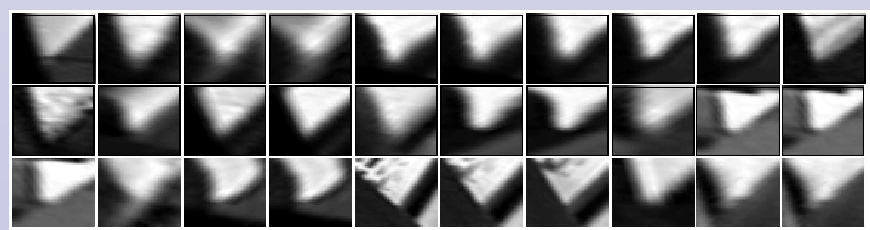


- Expected results (videos in ground truth):

Bag-of-Visual-Words

- The most common approach for Instance Search (and many other problems) is the well-known **Bag-of-Visual-Words** (BOVW) approach.
- It was introduced as a technique to perform efficient similarity searches in large video collections [Sivic and Zisserman, 2003].
  - The visual vocabulary (codebook) enables to create an inverted index.
  - The inverted index retrieves similar descriptors by locating collisions.
- Enables the perform similar searches in “immediate run-time”.





Bag-of-Visual-Words

- BOVW implementations usually follows three main steps:
  1. Extract local descriptors for the whole dataset (or some subset).
  2. Determine a codebook by calculating representative vectors for the dataset.
     - K-means algorithm due to its efficiency at large datasets.
  3. For each video frame calculate a histogram with the occurrences of each codeword.
     - Every local descriptor is quantized to its nearest codeword.
- Many variants and improvements.
- BOVW achieves satisfactory results at image classification, semantic indexing, object recognitions, etc.

Issues for BOVW approach

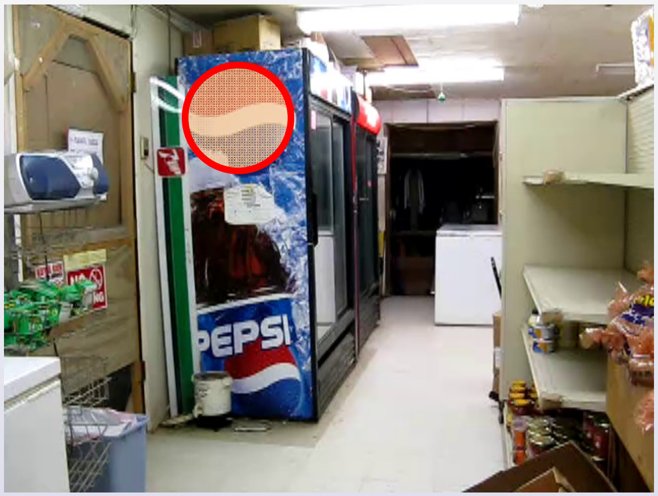
- Quantization of local descriptors produces loss of information.
  - Many techniques focuses on reducing this loss:
    - Soft-assignment [Van Gemert et al., 2008].
    - Hamming embedding [Jegou et al., 2008].
    - Spatial pyramids [Lazebnik et al., 2006].
    - Histogram of distances by codeword [Avila et al., 2011].
    - Many others..
- The codebook computation is expensive:
  - K-means algorithm can take several hours or days to complete.
  - It is an offline process (does not use queries), hence its processing time is not reported.

Research question

- **Question: Can the similarity search using the whole set of local descriptors achieve better effectiveness than BOVW?**
  - If quantization produces loss of information, then avoiding quantization might improve the effectiveness.
  - The online phase will be slower (at least will not be “immediate”)
  - The offline phase will not consider a expensive clustering process.

1. **Scenario 1: Naïve search outperforms BOVW.**
   - BOVW is a technique that improves efficiency but loses information in the quantization.
2. **Scenario 2: BOVW outperforms naïve search.**
   - The occurrences of the codewords create new information that is not provided by original descriptors.
   - “mid-level features” [Boureau et al., 2010; Martinet el al.].

System Overview



Topic

Dataset

System Overview (Step 1)

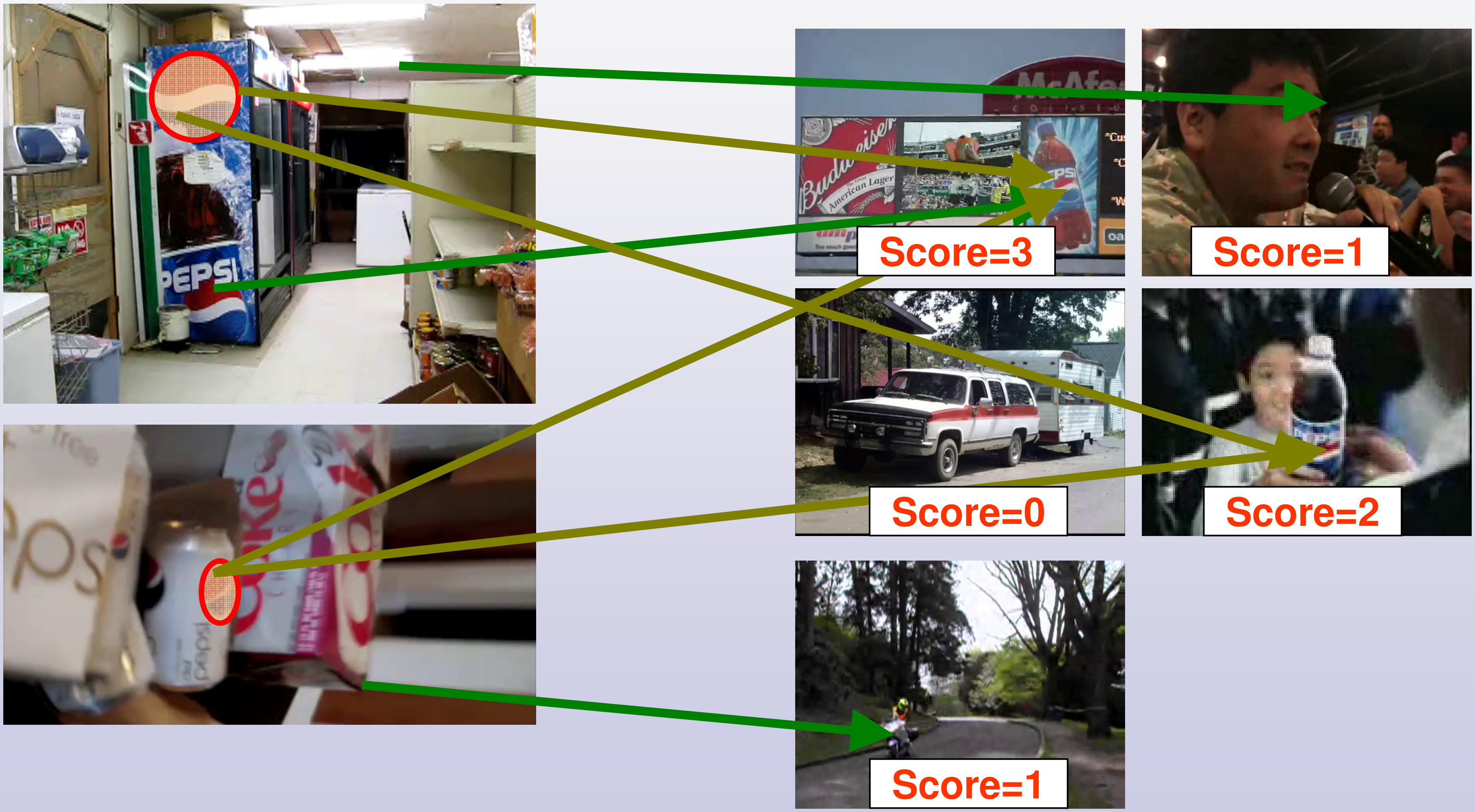
Q

R

System Overview (Step 2)

Q *R*

System Overview (Step 3)



Topic

Dataset

System Overview

1. **Feature Extraction.**
   - Computation of local descriptors for topic images and mirrored versions (*Q*).
   - Computation of local descriptor for sampled frames of dataset videos (*R*).
2. **Similarity Search**. For each object in *Q* perform a k-NN search in *R*.
   - Partition *R* in *m* subsets $R=\{R_1,\ldots,R_m\}$.
   - In parallel, using *m* different machines from **Amazon EC2**:
     - For each object in *Q* perform an approximate k-NN search in R_i.
     - Approximate search using the metric space approach.
   - Merge partial results to produce the k-NN for each object in *Q*.
3. **Instance search based on k-NN results**.
   - Voting algorithm based on the videos owning each NN.

Step 1: Feature extraction

- Keyframe selection by constant sampling.
- Two methods for interest point detection:
  - Hessian-Laplace (HL).
  - Maximally Stable Extremal Regions (MSER).
- Reduction of interest points by reducing frames size.
- CSIFT local descriptors (192d) for each interest point.
- “Feature Detection Code” http://www.featurespace.org/

- Submission **prisma-one180px**:
  - 1 frame every 1.5 seconds → 480.000 frames.
  - Images scaled to 180 pixels height → 345 HL/frame.
  - CSIFT → 192-d vectors.
  - ***Q*= 75.000** descriptors, ***R*= 166.000.000** descriptors.

Step 2: Similarity Search

- Submission **prisma-one180px**:
  - Naïve exact search → unaffordable (a few months to complete).
  - Partition *R* into *m*=10 subsets and resolve them in parallel by different machines.
  - ***Q*= 75.000** descriptors, **R_i= 16.600.000** descriptors.
  - Parallel exact search → several days to resolve *Q* searches.

- Search using the Metric space approach:
  - Similarity search and Indexing structure are based exclusively on distances between objects: *d(x,y)*.
  - Adaptation to local descriptors of the approximate search with pivots used at TRECVID 2011 [1].

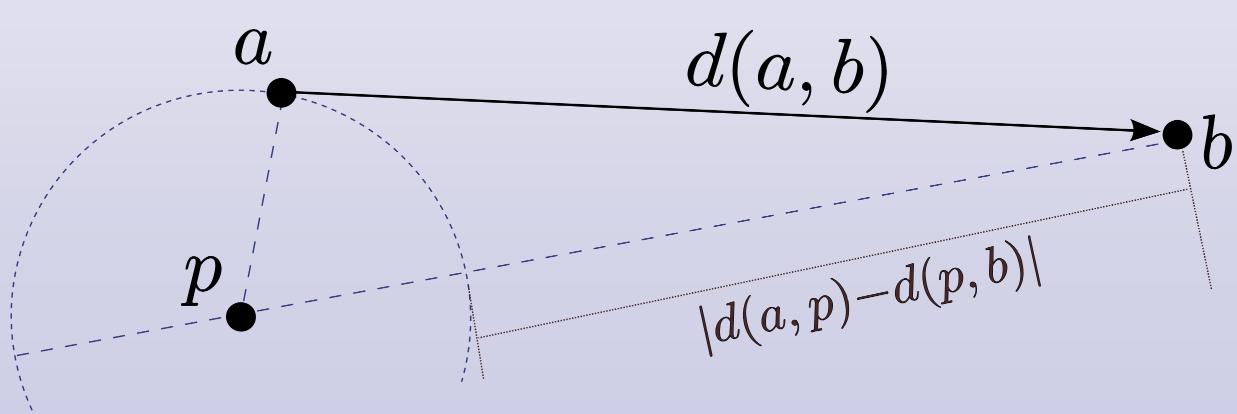
[1] J.M.Barrios and B.Bustos. *Competitive content-based video copy detection using global descriptors*. Multimedia Tools and Applications. Springer, 2011.

Step 2: Similarity Search

- Distance function *d* must satisfy the metric properties:
  - **Non-Negativity**, **Symmetry**, and **<u>Triangle Inequality</u>**.

$$d(a,b) \leq d(a,p) + d(p,b)$$

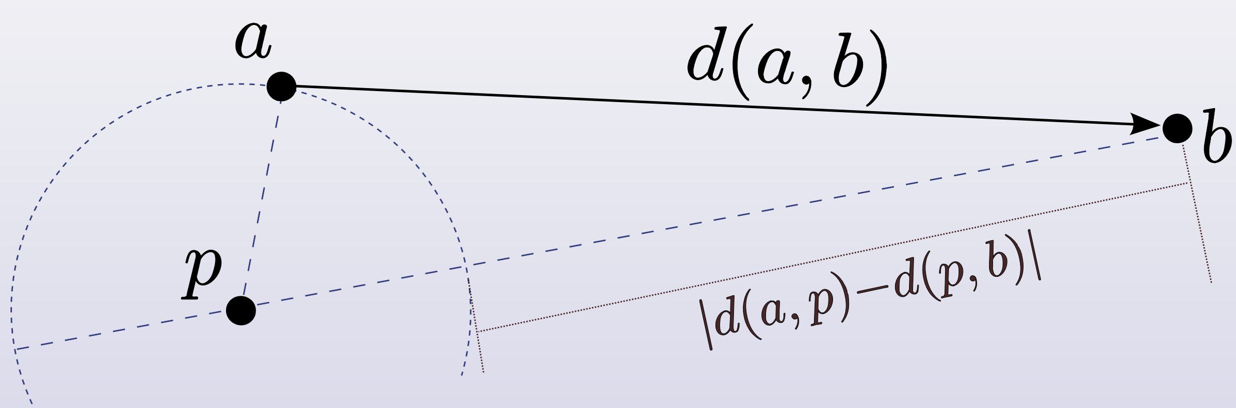
- Using a static object (called pivot), a **lower bound** for the distance *d(a,b)* can be computed:



Lower bound: $|d(a,p) - d(p,b)| \leq d(a,b)$

Step 2: Similarity Search

- Distance approximation:
  - Use the lower bound as a fast estimator of *d(a,b)*:

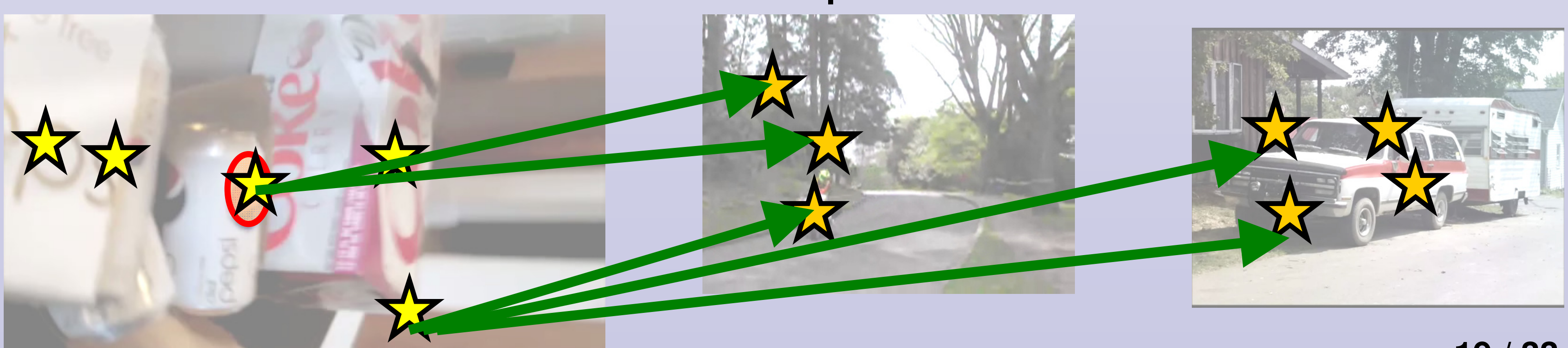


$$d(a,b) \approx |d(a,p) - d(p,b)|$$

  - Evaluate *d(a,b)* only for *T%* objects with lowest lower bound.
  - Estimation can be improve with more pivots.
- Submission **prisma-one180px**:
  - Parallel approximate search (T=0.5%) → a few hours to resolve *Q* searches.

Step 3: Instance Search

- For each object in *Q* the *k*=50 nearest neighbors are retrieved.
- Each NN votes in favor of the video that owns it.
- The vote is weighted according to the rank of the NN.
- Votes corresponding to a query object inside the mask are weighted higher (*2).
- Detection score is the sum of votes.
- Late fusion (sum of scores) for candidates proceeding from different local descriptors.



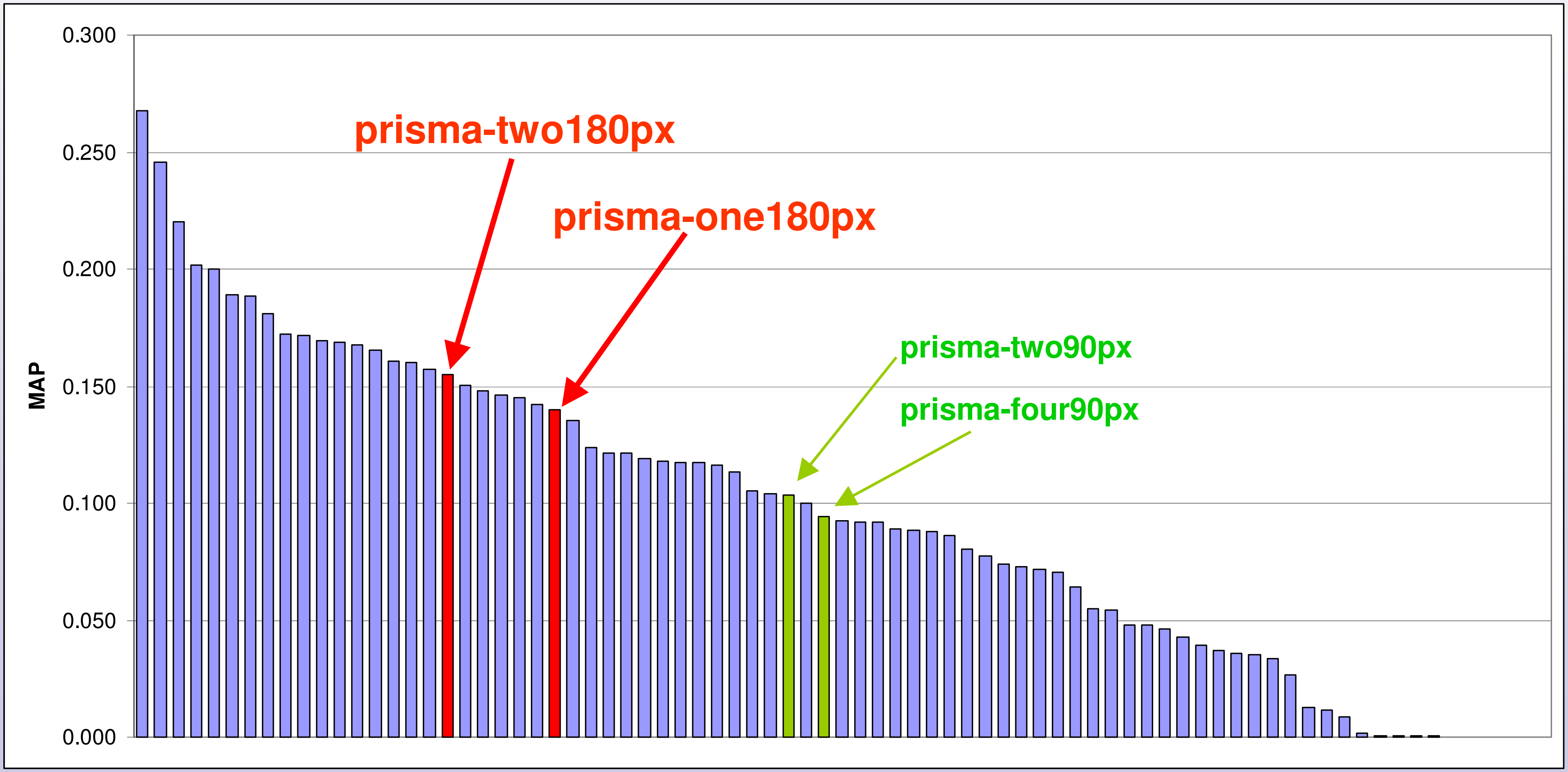
RESULTS

Results

- 24 teams, 79 automatic submissions.
- **prisma-one180px:**
  - 1 frame every 1.5 seconds.
  - Each frame scaled to 180 pixels height.
  - Extracts CSIFT at HL interest points.
  - **Q=75.000, R=166.000.000 objects.**
  - Parallel search in 10 machines.
  - Approximate search evaluating 0.5% of distances.
  - MAP=0.140 (24^{th} / 79)
- **prisma-two180px:**
  - Same as previous.
  - Extracts CSIFT at HL and CSIFT at MSER interest points.
  - **Q_{HL}=75.000, R_{HL}=166.000.000 objects.**
  - **Q_{HL}=44.000, R_{HL}=95.000.000 objects.**
  - Parallel search in 20 machines.
  - MAP=0.155 (18^{th} / 79)

Overall Results

- MAP for the 21 topics:

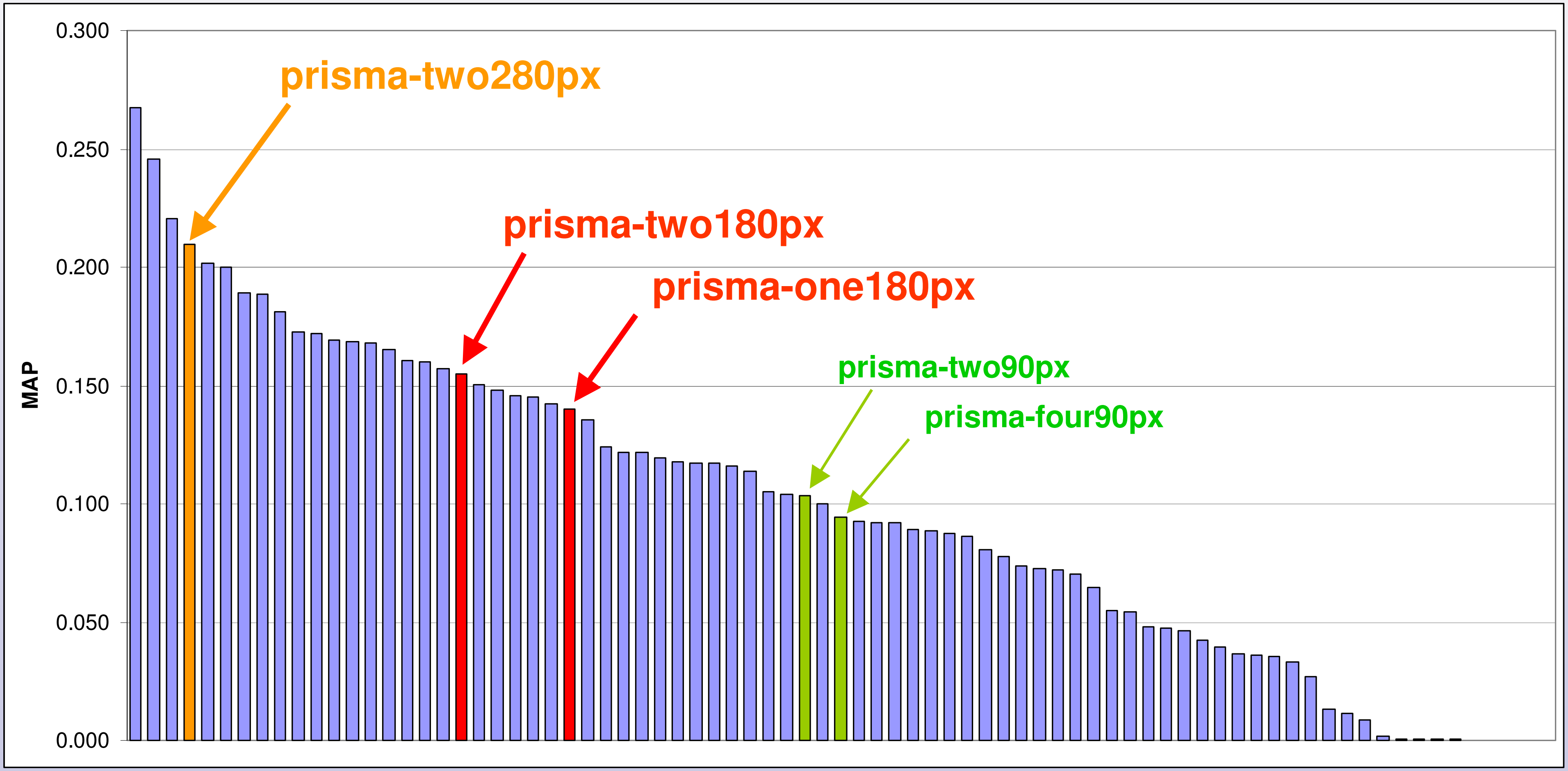


New Submission

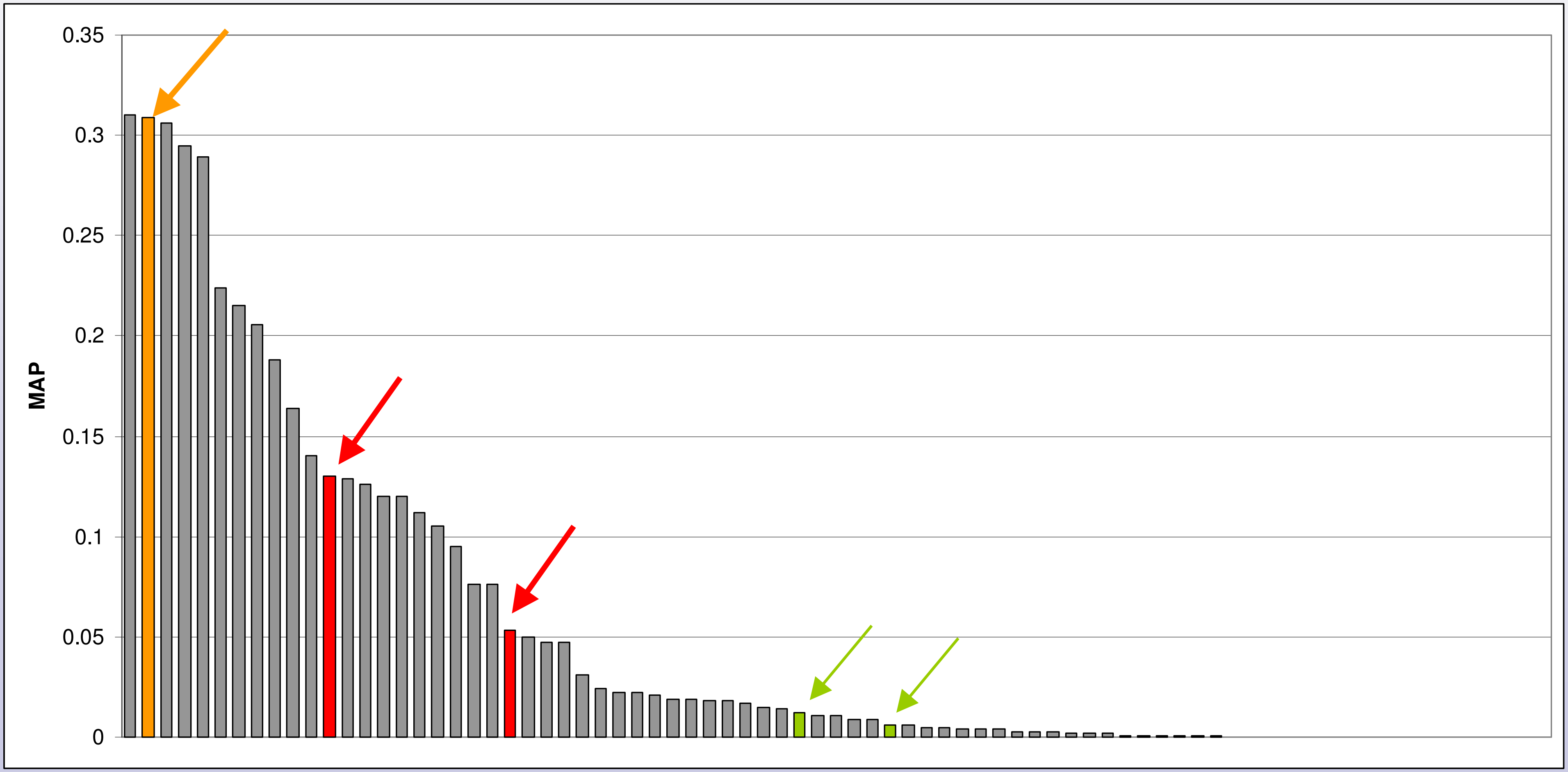
- **prisma-two280px (not submitted):**
  - 1 frame every 0.5 seconds.
  - Each frame scaled to 280 pixels height.
  - Extracts CSIFT at HL and CSIFT at MSER interest points.
  - **Q_{HL}=155.000, R_{HL}=973.000.000 objects.**
  - **Q_{HL}=94.000, R_{HL}=543.000.000 objects.**
  - Parallel search in 120 machines (in fact, 20 machines with 6 consecutive processes each one).
  - Approximate search evaluating 1% of distances.
  - MAP=0.210 (4^{th} / 79)

Overall Results

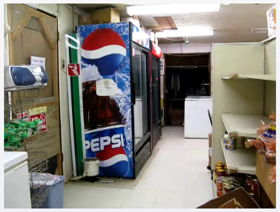
- MAP for the 21 topics.

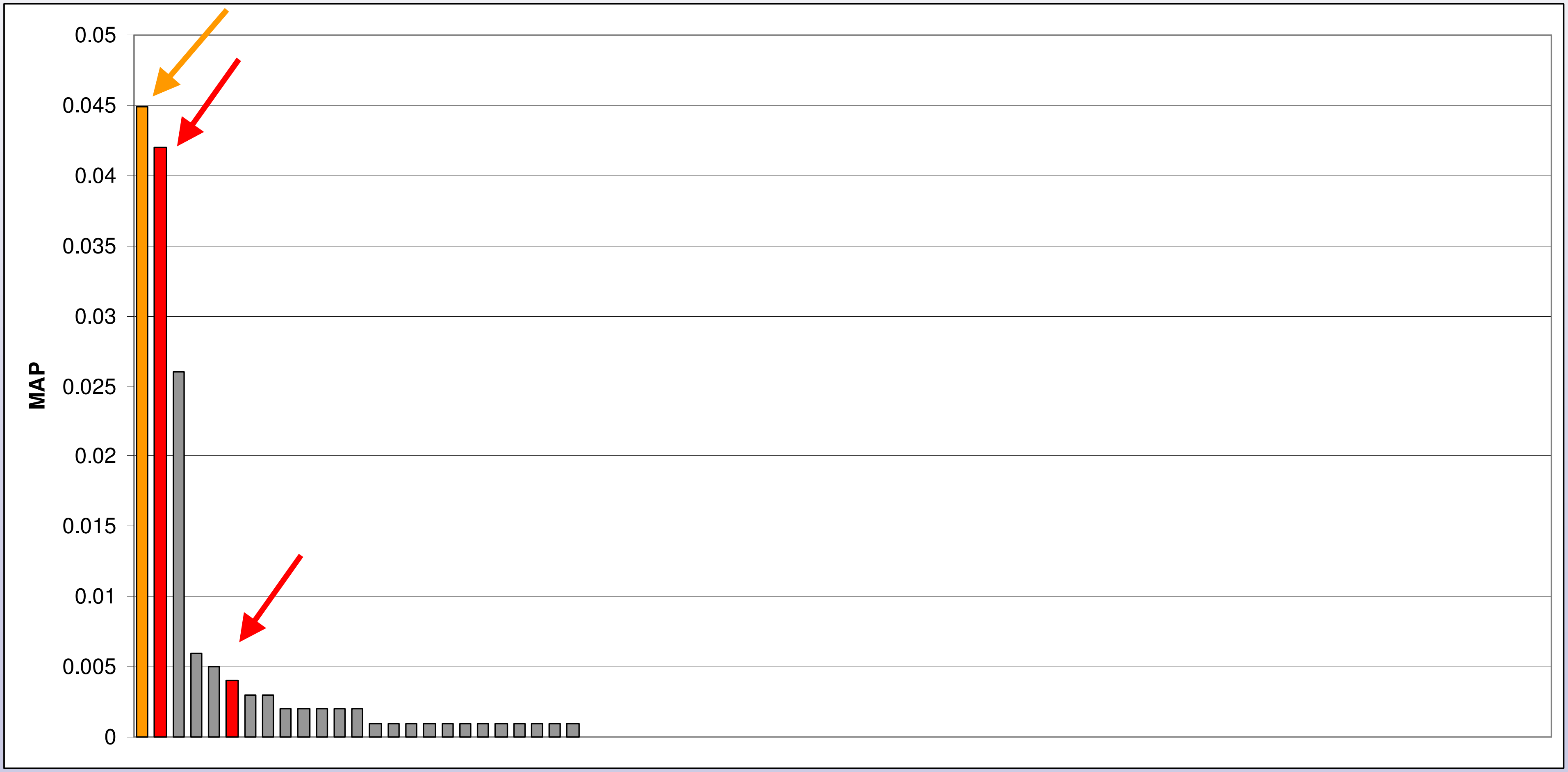


9052 London Underground logo

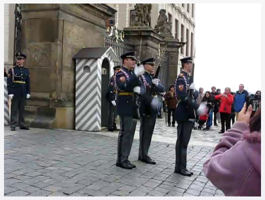


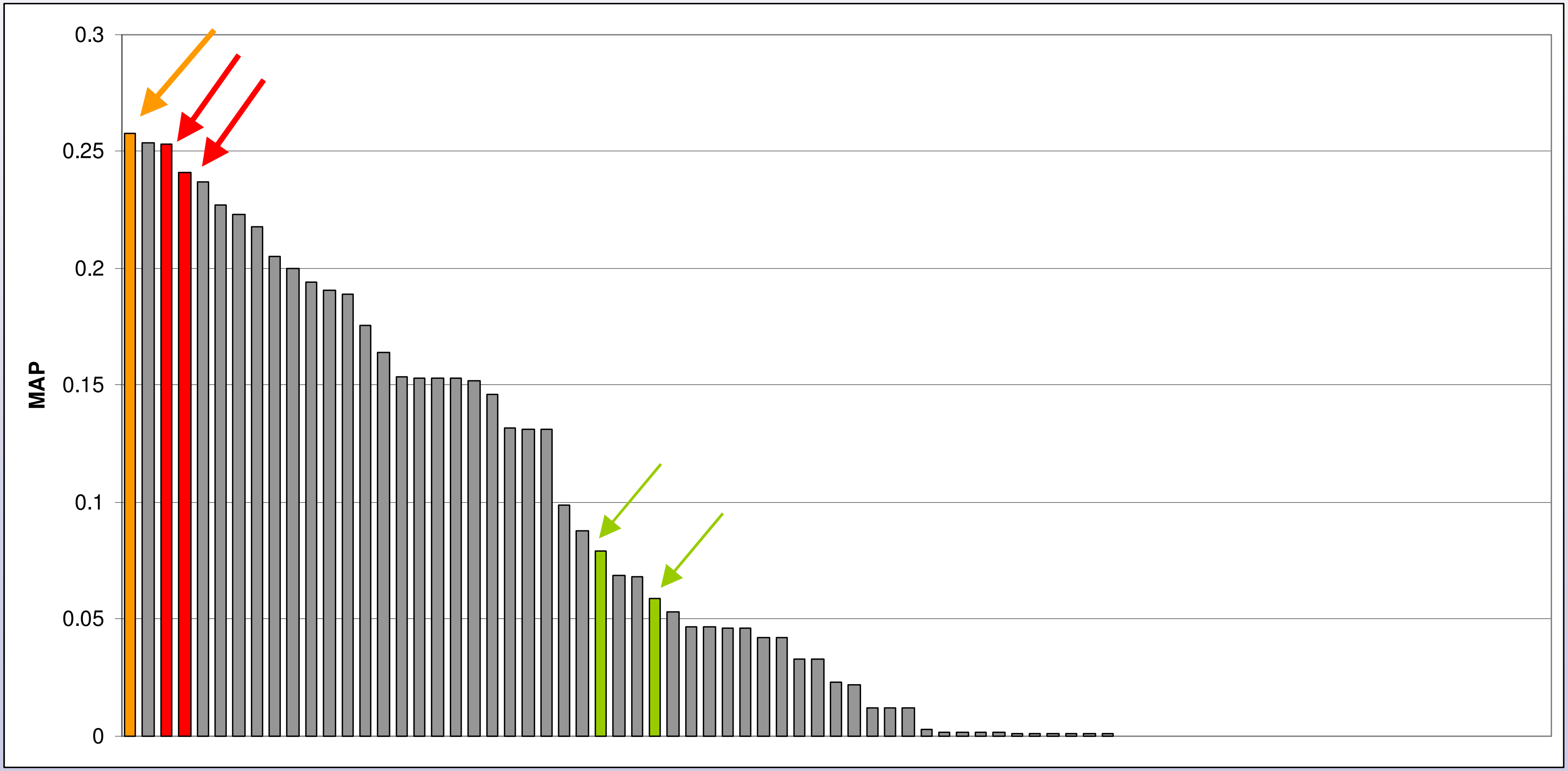
9061 Pepsi logo - circle



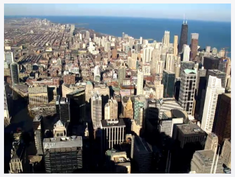


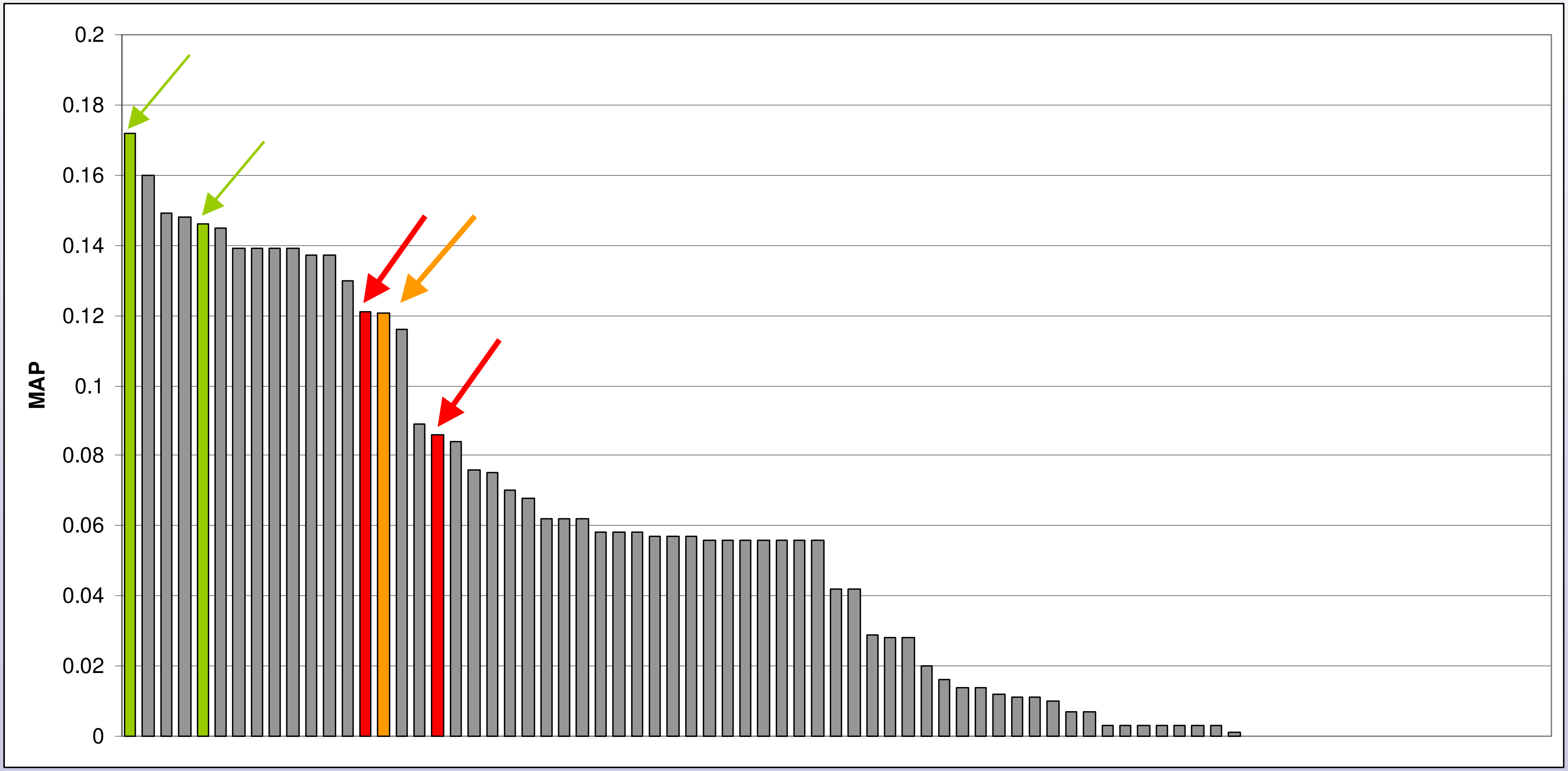
9063 Prague Castle



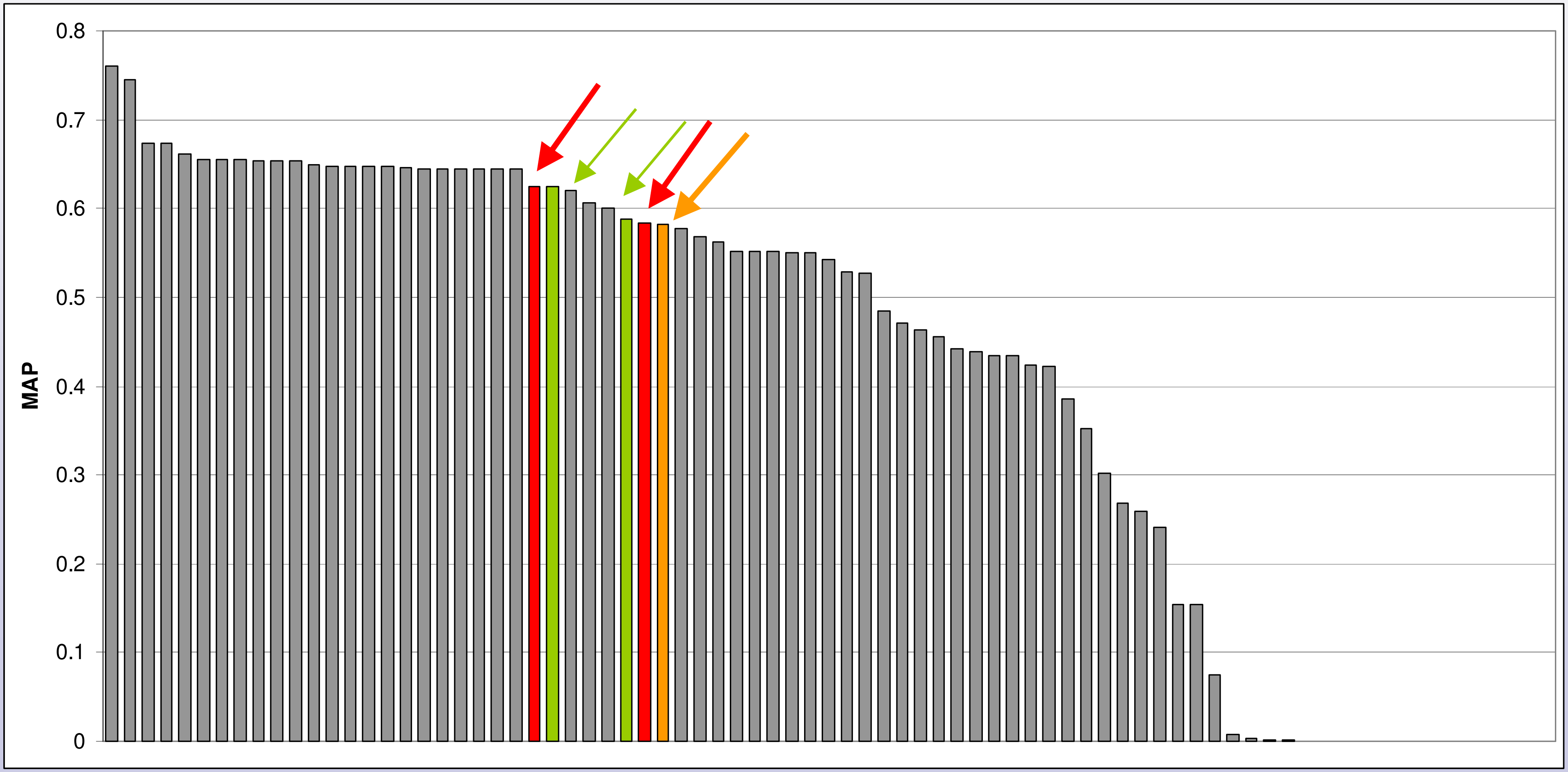


9055 Sears/Willis Tower



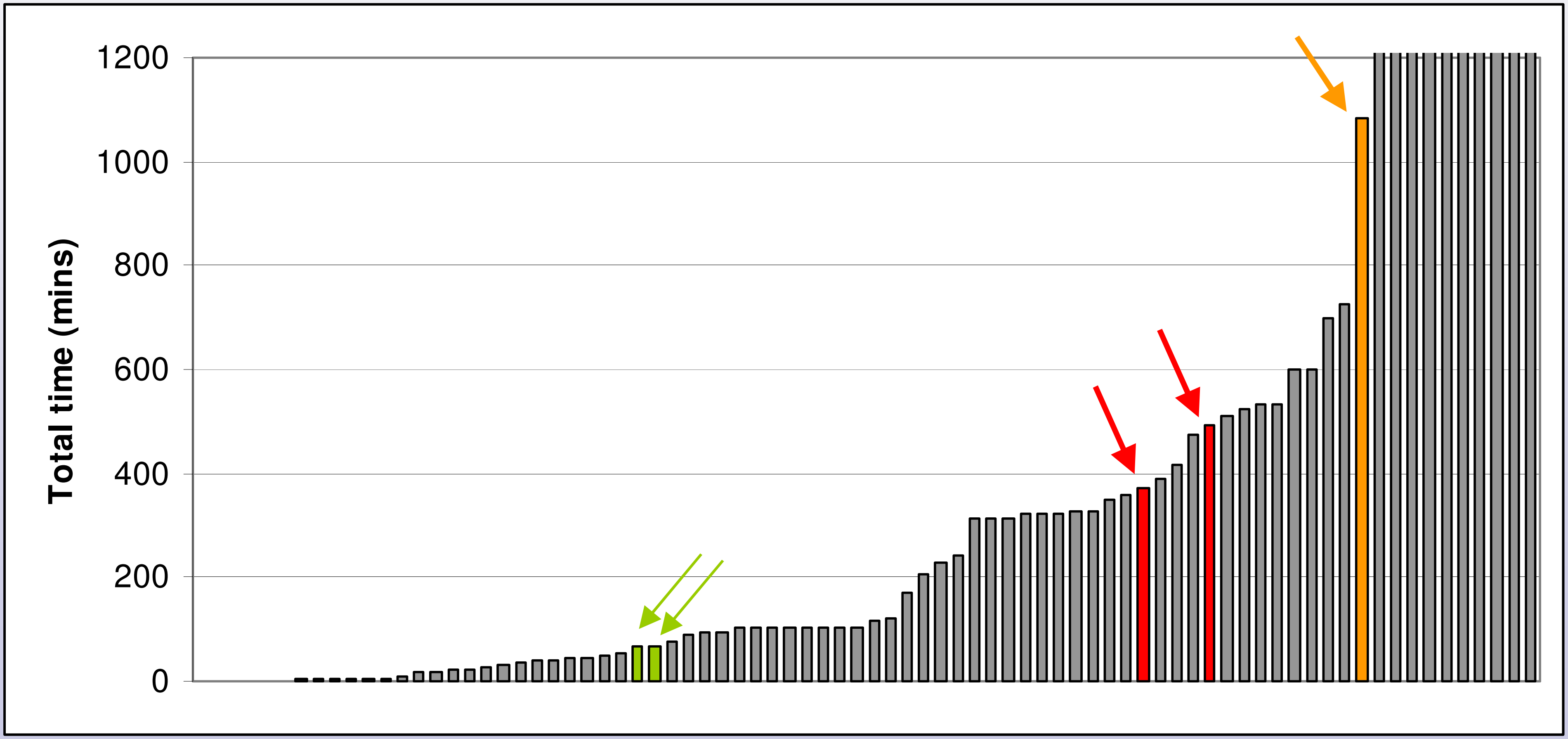


9060 Stephen Colbert



Time

- Sum of time for all topics:



Conclusions

- In this work we have shown an alternative approach for the BOVW method that may achieve high effectiveness at the Instance Search problem.
- In order to achieve high efficiency and effectiveness we perform several parallel approximate searches.
- The search method can easily be divided and distributed into a network of independent machines.
- We have tested our approach using the **Amazon Elastic Compute Cloud** (EC2).

Conclusions

- Does the similarity search on the whole set of local descriptors achieves better effectiveness than BOVW?
  - The results are not conclusive.
  - The dataset was not ideal to test this statement.
- Conjecture:
  - Similarity Search with no-quantization may achieve higher effectiveness <u>when the problem is based on duplicates</u>, like CCD and instance search (some topics).
  - BOVW can achieve higher effectiveness <u>when the problem is based on generalizations or related objects</u>, like semantic indexing, instance search (some topics), MED.

P-VCD

- P-VCD is an open source software with GPL license written in C.
  - http://sourceforge.net/projects/p-vcd/
- It contains the implementations for different search methods using the metric space approach.
- It was originally designed as an engine for content-based video copy detection. Now we have extended it to address the Instance Search problem.
- Its development is currently supported by ORAND, Chile.
- The project is still immature, but we encourage researchers and advanced users to test its performance.