ORAND at TRECVID 2014: Instance Search and Multimedia Event Detection

Juan Manuel Barrios
ORAND S.A.
Santiago, Chile
juan.barrios@orand.cl

Jose M. Saavedra
ORAND S.A.
Santiago, Chile
jose.saavedra@orand.cl

Felipe Ramirez
ORAND S.A.
Santiago, Chile
felipe.ramirez@orand.cl

David Contreras
ORAND S.A.
Santiago, Chile
david.contreras@orand.cl

ABSTRACT

ORAND S.A. is a Chilean company focused on developing applied research in Computer Science. This report describes the participation of the ORAND team at Instance Search task (INS) and Multimedia Event Detection task (MED) in TRECVID 2014.

The INS participation consisted in three submissions to automatic detection and one submission to interactive detection. All the submissions used the four samples for each topic (type D). Our submissions were based on computing approximate k-NN search between local descriptors (without using any quantization), and computing a static similarity graph between shots to propagate scores.

The MED participation consisted in two submissions to pre-specified events (010Ex and 100Ex) and two submissions to ad-hoc event detection (010Ex and 100Ex). Both submissions used the MED14EvalSub (32K videos) and noPRF. The submissions considered only low-level features (gray and color SIFT) and performed approximate k-NN searches between them without computing a codebook.

1. INTRODUCTION

ORAND is a Chilean software company focused on developing applied research in Computer Science. This paper describes our participation at Instance Search (INS) and Multimedia Event Detection (MED) tasks at TRECVID 2014 [6]. TRECVID is an evaluation sponsored by the National Institute of Standards and Technology (NIST) with the goal of encouraging research in video information retrieval [7].

2. INSTANCE SEARCH

Instance Search task (INS) consists in retrieving the shots that contain a given entity (object or person) from a video collection. The target entity, called a *topic*, is defined by visual examples and a brief textual description. A visual example is a still image (extracted from a sample video) and a mask, which delimits the region of the image where the topic is visible. INS 2014 evaluated 30 topics (24 objects, 1 location and 5 persons) with up to four visual examples per topic [6]. The reference video collection was the BBC EastEnders collection (same as INS 2013), which consists in 244 videos with a total extension of 435 hours (39 million frames approx.). Additionally, the list of shots for each video was predefined and given to each team (a total number of 471,526 shots). Each participant system had to submit the list of shots that most probably show each topic (with a maximum length of 1000 shots per topic).

2.1 System Description

This participation is the progression of our work at TRECVID 2013 [5]. We are currently interested in studying two aspects: the effectiveness that can be reached when no quantization is applied to local descriptors, and the propagation of scores between similar shots. Unlike the codebook approach, we follow the k-NN approach on the full set of descriptors. In this case, the main issue is to efficiently perform several k-NN searches in a very large set of vectors.

As a general overview, our approach follows these steps: video frames are sampled at a regular-step and local descriptors are computed for the selected frames. The extracted local descriptors are partitioned into subsets, and for each subset a k-NN search is performed. The partial results for all subsets are merged in order to determine the actual k-NN. A voting algorithm ranks each shot according to the number of nearest neighbors they contain in the k-NN lists. Finally, the scores are propagated between shots according to a pre-computed similarity shot graph.

2.1.1 Feature Extraction

The videos in the collection are TV quality: 576i/25. In particular, interlaced videos show unnatural horizontal lines that may affect the quality of local descriptors. In order to reduce this effect, all the videos were re-encoded and deinterlaced using FFmpeg software [1].

Then, every video was sampled at five frames per second, and for each frame we computed CSIFT implemented by *FeatureSpace* software [4], and SIFT descriptor implemented by *VLFeat* software [8].

2.1.2 Similarity Search

The similarity search consisted in retrieving for each x in $\mathcal{Q}$ the k Nearest Neighbors in $\mathcal{R}$ according to distance:

$$L_1(\vec{x}, \vec{y}) = \sum_{i=0}^{d} |x_i - y_i|$$

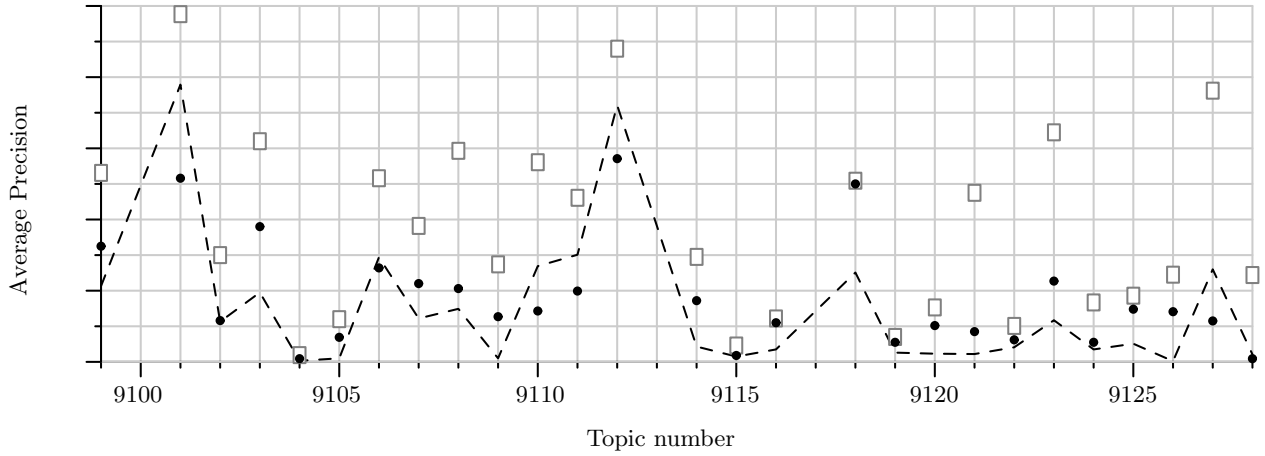

Figure 1: Instance Search, results achieved by F_D_4 submission. The dots show the achieved AP at each topic, the boxes the best AP achieved by any submission, and the dashed line is the median value. Note that this submission was based on an incomplete search.

In order to solve these searches, we partitioned $\mathcal{R}$ into several subsets $\{\mathcal{R}_1, ..., \mathcal{R}_n\}$, i.e.:

$$\mathcal{R} = \bigcup_{i=1}^{n} \mathcal{R}_i \quad , \quad \forall\, i \neq j, \mathcal{R}_i \bigcap \mathcal{R}_j = \emptyset$$

Thereafter, for each x in $\mathcal{Q}$ an approximate k-NN search is performed at every $\mathcal{R}_i$. The final k-NN are determined by merging the n partial results and selecting the top k. The similarity search was implemented using the MetricKnn library [2]. The similarity search was resolved using resources from the NLHPC [3].

2.1.3 Voting algorithm

In order to score shots, a voting algorithm traverses the lists of k-NN for each local descriptor at each example image, and sums one vote to the shot that contains the frame that produced the NN. Each votes is weighted according to the distance to the mask and the rank in the k-NN list of the voter. The sum of votes produces the final score for each shot, and the top 1000 are selected for each topic.

2.1.4 Similarity Shot Graph

A video shot is a series of interrelated consecutive frames. Usually, a shot division produces fine-grained segmentation of videos. In fact, the shot division provided by NIST produces shots with average length 3.3 seconds, and many shots are just a few milliseconds length. If the topic is static in the background, it may be expected that the topic will also be visible in other shots from the same scene.

The Similarity Shot Graph (SSG) is created by computing the similarity between every pair of shots in the collection. Let S be the number number of shots in the collection (S is about 500.000), the SSG is a directed weighted graph with S nodes and the edge between two nodes represents the similarity between the two shots. The similarity between two shots was computed by sampling three frames from each shot (start/middle/end) and extracting a global descriptor to each frame. Then, a k-NN search was resolved for each frame and the similarity between shots is computed by counting common NNs. We implemented the similarity search using the MetricKnn library [2].

In addition to propagate scores in shots, the SSG is also used to propagate user decisions for the interactive run. When a user states that a candidate shot is correct or incorrect, that decision is also propagated to other shots following the edges in the SSG.

2.2 Submissions and Results

Two kinds of submissions were evaluated: interactive and automatic. All our submissions were type D (four visual examples). Each submission was evaluated by NIST, computing the average precision by topic (30 topics for automatic runs and 25 topics for interactive runs) and we computed the MAP by averaging all the results. We created three automatic runs and one interactive run:

- F_D_1: The same as F_D_2 plus propagation of scores using a similarity shot graph. MAP=0.139.
- F_D_2: SIFT descriptors extracted every 5 frames per second, approximate 20-NN search using 5 kd-tree. MAP=0.137.
- F_D_4: CSIFT descriptors extracted every 5 frames per second, approximate 20-NN search using 5 kd-tree. MAP=0.183. Incomplete Search.
- I_D_3: Interactive Run. The run F_D_1 was evaluated and the user decisions were propagated using shot similarity graph. The user reviewed the top-score shots up to complete the total runtime limit and classified them into correct/incorrect shots. Every user decision was also propagated to similar shots following the similarity graph. MAP=0.174.

Unfortunately, during our participation we had a problem with the infrastructure during the search phase, which forced us to build the F_D_4 submission with just an 80% completed. However, once we submitted the runs, we continued and completed the search, and re-evaluated it using the released ground truth. The results for the complete search were MAP=0.220. Also, if the interactive run had been based on F_D_4, the interactive run would have achieved MAP=0.250.

3. MULTIMEDIA EVENT DETECTION

Multimedia Event Detection (MED) consists in deciding whether a given event is present in a video clip. The event is specified by an "event-kit", which contains a textual description of the event plus 100, 10 or 0 example videos. The evaluation considered two scenarios: *pre-specified events*, i.e., the event-kits are a priori known by the team thus it is possible to manually adjust a specific detector for each event; and *ad-hoc events*, i.e., the event-kits are a priori unknown by the team, thus the system must have a generic search engine that takes the event-kit as input. The reference video collection consisted in 200K search videos, and a team may choose to evaluate the system only in a subset of approximately 32K videos (MED14EvalSub) [6].

The MED participation consisted in two submissions to pre-specified events (010Ex and 100Ex) and two submissions to ad-hoc event detection (010Ex and 100Ex). Both submissions used the MED14EvalSub without any feedback process (noPRF). The submissions considered only low-level features and were based on the approach of approximate k-NN searches that we used for the Instance Search task.

3.1 Pre-Specified Events

In the case of Pre-Specified Events, for each training video in event-kits (010Ex and 100Ex) and background videos we sampled 5 frames per video (evenly distributed), and we computed CSIFT descriptors scaling down the images to 200x150 pixels. Thereafter, for each video in MED14EvalSub the same features were extracted and approximate k-NN searches (k=10) were performed (i.e., locating the descriptors from training videos and background videos that were most similar to the descriptors in the test video). The voting algorithm consisted in processing the k-NN lists, and summing one vote to the event-kit that owns each retrieved NN. The spatial restriction were applied in order to reduce noise: there must be at least 5 votes in the same frame in order to count the votes. The classification output corresponded to the most voted event, and the confidence score was given by the difference to the second most voted event. The results achieved by this submission were MAP=1.2% (010Ex) and MAP=5.0% (100Ex), which is a poor performance compared to submissions from other teams.

3.2 Ad-hoc Events

In Ad-hoc Event Detection, the training videos (010Ex and 100Ex) and background video were sampled at 10 frames per video, each sampled frame was scaled to 400x225 pixels and three types of descriptors were computed: SIFT, CSIFT at MSER keypoints, and CSIFT at Hessian-Laplace keypoints. Thereafter, for each video in MED14EvalSub the same features were extracted and an approximate k-NN searches (k=1) were performed. The voting algorithm was run separately for each type of descriptor, and the total votes were merged and normalized to sum 1. The classification output corresponded to the most voted event, and the confidence score was given by the difference to the second most voted event. The results achieved by this submission were MAP=4.3% (010Ex) and MAP=10.0% (100Ex), which is an improvement compared to our submission in Pre-Specified Events, but still a poor performance compared to submissions from other teams.

All the submissions for MED were completed on a single machine Intel Core i7-4770K (3.50GHz, 8 cores), 32 GB RAM, 7 TB disk, Linux. We should note our system needed very small resources either in training and testing (it does not need any clustering process), thus our submissions were between the fastest ones. Therefore, the poor detection performance might also be due to the small resources assigned to the system.

4. CONCLUSIONS

In this report we described our submissions to INS and MED tasks at TRECVID 2014. They were based on resolving k-NN searches in the full set of descriptors without applying quantization nor summarization to them.

In Instance Search task, the system achieved competitive performance compared to other teams. The similarity shot graph may be useful to improve the MAP either in automatic and interactive search. However, we should note in some topics it may harm the precision. More research is needed in order to understand the scenarios were the graph can be successfully applied.

In Multimedia Event Detection task, our submissions achieved in general a low performance. We still need more work in order to evaluate whether the system requires more resources to properly work, or the approach of k-NN searches is not able to fulfill the semantic needs of MED task.

The library MetricKnn was built during this participation. It contains implementation for different indices, distances and search algorithms. The library is under construction but a preview version is available in its website [2].

5. ACKNOWLEDGEMENTS

This research was partially supported by the supercomputing infrastructure of the NLHPC (ECM-02) and by CONICYT Project PAI-78120426.

6. REFERENCES

[1] FFmpeg. `http://www.ffmpeg.org/`.

[2] MetricKnn. `http://www.metricknn.org/`.

[3] National Laboratory for High Performance Computing (NLHPC). `http://www.nlhpc.cl/`.

[4] Feature Detectors and Descriptors: The State Of The Art and Beyond. Feature Detection Code., 2010. `http://kahlan.eps.surrey.ac.uk/featurespace/web/`.

[5] J. M. Barrios, J. M. Saavedra, F. Ramirez, and D. Contreras. Orand team: Instance search and multimedia event detection using k-nn searches. In *Proc. of TRECVID*. NIST, USA, 2013.

[6] P. Over, G. Awad, M. Michel, J. Fiscus, G. Sanders, W. Kraaij, A. F. Smeaton, and G. Quéenot. Trecvid 2014 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proceedings of TRECVID 2014*. NIST, USA, 2014.

[7] A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and TRECVid. In *Proc. of the int. workshop on Multimedia Information Retrieval (MIR)*, pages 321–330. ACM, 2006.

[8] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms, 2008. `http://www.vlfeat.org/`.