ORAND Team: Instance Search and Multimedia Event Detection Using k-NN Searches

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 2013.

The INS participation considered four submissions, namely: orand-1sift, orand-2sift, orand-graph, and orand-interactive. The first two submissions follow the approximate k-NN search approach we presented at TRECVID 2012, the last two submissions use a static similarity graph between shots to propagate scores. In general, our submissions achieve satisfactory performance: their MAP are higher than the median in every topic, and they achieved the highest MAP in two topics.

The MED participation considered one submission to prespecified events and one submission to ad-hoc event detection. The first submission follows a naive BOW approach and achieved low performance. The second submission follows the approximate k-NN search approach and achieved higher performance. This is our first participation at MED, hence we still need more work in order to achieve competitive performance in this task.

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 2013 [12]. TRECVID is an evaluation sponsored by the National Institute of Standards and Technology (NIST) with the goal of encouraging research in video information retrieval [14].

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 2013 evaluated 30 topics (26 objects and 4 persons) with four visual examples per topic. The reference video collection was the BBC EastEnders collection, 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).

Currently, the most common approach used to address the Instance Search problem is the well-known Bag-of-Visual-Words or codebook approach. It was introduced as a technique to perform efficient similarity searches in large video collections [13]. This approach first extracts local descriptors from a sample of video frames, then it defines the codebook as the set of centroids computed by a clustering algorithm. Many systems following this approach show high performance at Instance Search and other related problems like video classification, copy detection, event detection, object recognition, etc.

However, two main issues arise when following the codebook approach: the high computational cost required by the codebook computation, and the loss of information due to quantization. Many techniques have been developed either to improve the performance of the codebook computation and/or to improve the quality of the information stored in descriptors, e.g. soft assignment [16], hamming embedding [7], spatial pyramids [8], histogram of distances by codeword [4], hierarchical k-means [9], and many others.

2.1 System Description

This participation is the progression of our work at TRECVID 2012 [5]. We are currently interested in studying the effectiveness that can be reached when no quantization is applied to local descriptors. 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: first, the videos are sampled at a regular-step, then two types of local descriptors are computed for selected frames: SIFT and CSIFT. The local descriptors are partitioned into subsets, and for each subset a k-NN search is performed. The partial results for every subset are merged in order to de-

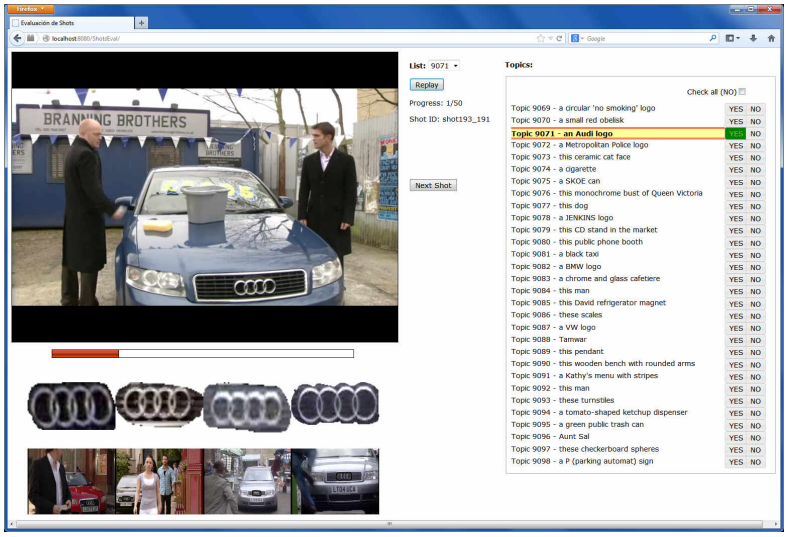

Figure 1: Instance Search, interactive system. The user must enter *Yes* or *No* whether the topic 9071 (an Audi logo) is visible or not in the displayed shot. Programme material ©BBC.

termine the actual k-NN. Thereafter, the shots are ranked according to the number of nearest neighbors they contain in the k-NN lists.

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 one frame per second, and for each frame we computed CSIFT implemented by *FeatureSpace* software [3], and SIFT descriptor implemented by *VLFeat* software [17].

Let $\mathcal{R}$ be the set of descriptors for reference videos, and $\mathcal{Q}$ be the set of descriptor for visual topic examples, the sizes of these sets were:

- CSIFT:
  - $|\mathcal{Q}| = 1.8 \times 10^5$ vectors 192-d.
  - $|\mathcal{R}| = 1.5 \times 10^9$ vectors 192-d.
- SIFT:
  - $|\mathcal{Q}| = 1.2 \times 10^5$ vectors 128-d.
  - $|\mathcal{R}| = 1.7 \times 10^9$ vectors 128-d.

2.1.2 Similarity Search

The similarity search consisted in retrieving for each x in $\mathcal{Q}$ the k Nearest Neighbors (k=50) in $\mathcal{R}$ according to distance:

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

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. Unlike our last participation, this year we solved the approximate search using the FLANN library [11], computing two kd-trees per subset and the search limits the number of visited leaves. We implemented the similarity search using the P-VCD software [2].

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 Aggregation of scores

Two different lists of candidates can be merged to produce a single list by summing the scores of common shots and selecting the top 1000 scores. We used this technique to combine the candidate shots obtained from CSIFT and SIFT. We also used this technique in the voting algorithm to fix the internal parameters, i.e., instead of selecting a single method to weight votes (e.g. gaussian, linear, or sigmoid weighting) we used many methods separately producing a list of candidates for each method, and then we merged all the lists producing a single list. Hence, we used the score aggregation as a consensus algorithm between different methods and parameters.

2.1.5 Similarity Shot Graph

A video shot is a series of interrelated consecutive frames taken contiguously by a single camera and representing a continuous action in time and space [6]. A shot division of a video may produce fine-grained segmentation of videos. In fact, the shots provided by NIST have an average length 3.3 seconds per shot and many shots are just a few milliseconds length. A topic is usually visible in many shots from the

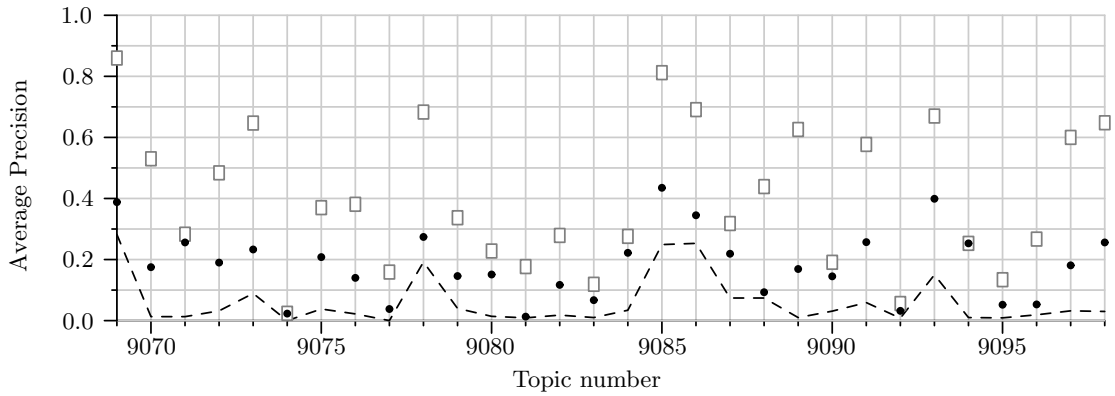

Figure 2: Instance Search, results achieved by orand-graph submission. The dots show the achieved AP at each topic, the boxes the best AP achieved by any submission, and the dashed line is the median value.

same scene, hence when a shot has high score for a topic, it is likely that some other shots from the same scene will also contain the topic.

We extracted a global descriptor for each shot (color histogram and edge histogram [10]). Then, we computed a similarity graph by creating a node for each shot and the weighted edges are computed by calculating the distance between shot descriptors. We used the similarity graph to improve the scoring of shots by propagating votes between similar shots. Additionally, we also used the similarity graph in the interactive run to propagate the decision of the user whether a shot contains some topic or not.

2.2 Submissions and Results

Each run was evaluated by NIST, computing the average precision by topic (30 topics for automatic runs and 25 topics for interactive runs). Two kinds of submissions were evaluated: interactive (where a user can correct the results and give feedback to the system), and automatic (where the system gives the results without user interaction). In total 65 automatic submissions and 9 interactive submissions were evaluated. We submitted three automatic runs and one interactive run. Their description and Mean Average Precision are:

- orand-1sift: is the list of shots obtained from computing CSIFT descriptors, approximate search, and voting algorithm (as described above). MAP=0.183, 15^{th} of 65 runs.
- orand-2sift: is produced by merging orand-1sift with list of shots obtained from using SIFT descriptors. MAP=0.177, 17^{th} of 65 runs.
- orand-graph: is produced by propagating the score of candidates in orand-2sift to other similar shots according to the similarity graph. MAP=0.184, 14^{th} of 65 runs.
- orand-interactive: the runtime of orand-graph was subtracted from the runtime limit of 15 minutes defined by NIST. A 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=0.215, 3^{rd} of 9 interactive runs. Figure 1 shows a screenshot of the interactive system.

The decrease in effectiveness between orand-1sift and orand-2sift evinces some problem in the score aggregation when the lists are produced from different modalities. On the other hand, the increase in effectiveness between orand-2sift and orand-graph shows that the similarity shot graph helps at increasing the rank of correct shots.

Comparing the results by topic with other teams, our submissions achieved a AP above the median at almost every topic, and achieved the first place at topics *9071–an Audi logo* (MAP=0.283 by orand-1sift) and *9094–a tomato-shaped ketchup dispenser* (MAP=0.253 by orand-graph). Also a high rank were achieved at topics *9074–a cigarette*, *9087–a VW logo*, and *9090–this wooden bench with rounded arms*. Figure 2 shows the performance by topic achieved by orand-graph.

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 for this year [15] consisted in 98.119 search videos, 1.2 TB (PROGAll dataset). Optionally, a team may choose to evaluate the system only in a subset of approximately 32.000 videos (PROGSub dataset).

This was our first participation in this task.

3.1 Pre-specified Events

In the case of pre-specified events, we were able to submit just one run using a naive implementation of Bag-Of-Words approach. We extracted a sample of 10 frames per video evenly distributed. For each sampled frame, we computed SIFT descriptors using VLFeat software [17]. The descriptors from 100Ex-videos were clusterized with k-means algorithm in order to compute 1000 centroids. Thereafter, a summarization vector per video was computed. Using the BOW vectors for training videos we built a SVM model per

event. The training and validation datasets corresponded to 100Ex-videos. Finally, we were only able to successfully complete the classification step in the PROGSub dataset.

This submission achieved a low effectiveness: MAP=0.6%, compared to other submissions we obtained the 17^{th} of 18 participant teams. In this submission we suffered from our lack of experience in this task. In order to overcome this problem for the ad-hoc evaluation, we quickly adapted the engine we used at Instance Search task (see Section 2.1) to the requirements of MED.

3.2 Ad-hoc Events

In the case of ad-hoc events, we finally submitted a run following the approach of k-NN searches we had used at Instance Search task, however due to time restriction we had to use low quality images and highly approximated searches. We extracted a sample of 5 frames per video evenly distributed, and we computed SIFT and CSIFT descriptors (the frames were scaled down to 150 pixels height). We processed all the videos in PROGAll dataset and training videos in event-kits. Thereafter, for each video in PROGAll dataset, we loaded its local descriptors and for each descriptor we performed an approximate k-NN search (k=4) in the set of descriptors of training videos. The voting algorithm consisted in processing the k-NN lists, and summing one vote to the event-kit that owns each retrieved NN. The voting algorithm was run separately for SIFT and CSIFT, 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=3.8% in PROGAll dataset and MAP=5.4% in PROGSub dataset. These results clearly show an improvement compared to the achieved MAP at pre-specified events. However, they still show a poor performance compared to other submissions: 11^{th} of 14 teams in PROGAll, and 13^{th} of 16 teams in PROGSub. Therefore, more work is needed in order to obtain satisfactory performance at this task.

4. CONCLUSIONS

In this report we detail our submissions and achieved results in INS and MED tasks at TRECVID 2013. Our submissions were based on performing k-NN searches in the full set of descriptors without applying quantization nor summarization. The approach shows promising results: in Instance Search task, the system achieved competitive performance compared to other teams, and at some topics achieved the highest performance. However, we still need to analyze the benefits and drawbacks of this approach compared to codebooks. In Multimedia Event Detection task, our lack of experience in this task affected our results. The submissions to this task achieved low performance, thus we still need more work to address MED challenges and improve effectiveness. The submissions for both tasks were completed on a single machine Intel Core i7-4770K (3.50GHz, 8 cores), 32 GB RAM, 7 TB disk, Linux.

5. REFERENCES

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

[2] P-VCD. `http://sourceforge.net/projects/p-vcd/`.

[3] Feature Detection Code., 2010. `http://www.featurespace.org/`.

[4] S. Avila, N. Thome, M. Cord, E. Valle, and A. Araujo. Bossa: Extended bow formalism for image classification. In *Proc. of the int. conf. on Image Processing (ICIP)*, pages 2909–2912. IEEE, 2011.

[5] J. M. Barrios and B. Bustos. Prisma-orand team: Instance search based on parallel approximate searches. In *Proc. of TRECVID*. NIST, USA, 2012.

[6] A. Hanjalic. Shot-boundary detection: unraveled and resolved? *IEEE Transactions on Circuits and Systems for Video Technology*, 12(2):90–105, 2002.

[7] H. Jégou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *Proc. of the european conf. on Computer Vision (ECCV)*, pages 304–317. Springer, 2008.

[8] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. of the intl. conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2169–2178 Vol.2. IEEE, 2006.

[9] D.-D. Le, C.-Z. Zhu, S. Poullot, V. Q. Lam, D. A. Duong, and S. Satoh. National institute of informatics, japan at trecvid 2011. In *Proc. of TRECVID*. NIST, USA, 2011.

[10] B. S. Manjunath, J.-R. Ohm, V. V. Vasudevan, and A. Yamada. Color and texture descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):703–715, 2001.

[11] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *Proc. of the int. conf. on Computer Vision Theory and Application (VISSAPP)*, pages 331–340. INSTICC Press, 2009.

[12] P. Over, G. Awad, M. Michel, J. Fiscus, G. Sanders, W. Kraaij, A. F. Smeaton, and G. Quéenot. Trecvid 2013 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proc. of TRECVID*. NIST, USA, 2013.

[13] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proc. of the IEEE int. conf. on Computer Vision (ICCV)*, pages 1470–1477. IEEE, 2003.

[14] 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.

[15] S. Strassel, A. Morris, J. Fiscus, C. Caruso, H. Lee, P. Over, J. Fiumara, B. Shaw, B. Antonishek, and M. Michel. Creating havic: Heterogeneous audio visual internet collection. In *Proc. of the int. conf. on Language Resources and Evaluation (LREC)*. European Language Resources Association (ELRA), 2012.

[16] J. van Gemert, J.-M. Geusebroek, C. Veenman, and A. Smeulders. Kernel codebooks for scene categorization. In *Proc. of the european conf. on Computer Vision (ECCV)*, pages 696–709. Springer, 2008.

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