The University of Sheffield and Harbin Engineering University at TRECVID 2012: Instance Search

Manal Al Ghamdi[†]*, Muhammad Usman Ghani Khan*[†]*, Lei Zhang*[‡] *and Yoshihiko Gotoh*[†]

[†] Department of Computer Science, University of Sheffield, UK [‡] Harbin Engineering University, PRC

ABSTRACT

This paper describes our contribution to instance search (INS) task for TRECVID 2012. We present four approaches for this task, (i) histograms of SIFT features as feature vectors and Bhatacharya distance for similarity detection (ii) feature vector is combination of SIFT features alone, while for matching we used a basic descriptor matching algorithm (iii) IR based approach using SIFT features and (iv) affine invariant SIFT features as feature vectors.

Index Terms— video retrieval, instance search task, video indexing

1. INTRODUCTION

The INS task is a pilot task introduced in TRECVID 2010 campaign. Yearly, different testing video and query images are released to the participants for the INS task. In TRECVID 2011, the testing data was produced form the rushes collection. They automatically decomposed each video in the dataset into short and equally length clips with different names from the original video file. There were a total number of 20,982 test video clips and 25 image test queries. Some image transformations were also applied to random test clips. The task includes recurring queries with people, location and objects in the rushes.

This year, there were 21 topics and more than 70000 short clips as testing data collected from the Flicker. The main objectives from our participant was to explore the task definition and the evaluation issues.

2. APPROACHES OVERVIEW

The main aspects of our four approaches are presented in details below.

2.1. Run 1: PHOW descriptors and BoW approach

2.1.1. Offline Indexing

We extracted one frame per second from every video clips. Then we densely computed the PHOW descriptors on a regular grid across the image and vector quantised them into visual words. The codebook size is set to 500. We used the SIFT code available from the *VLFeat toolbox* [1]. The frequency of each visual word is then recorded in a histogram for each tile of a spatial tiling. The final feature vector for the image is a concatenation of these histograms.

2.1.2. Online Indexing

The framework of online searching is presented in part of Figure 1. Given the image set of topic, we extracted the Region of Interest (ROI) using the related mask. Then the feature vector consists of SIFT features computed on a regular grid across the image. Finally, the extracted SIFT features are projected to the vocabulary tree. One histogram is then generated as final representation for each topic. For similarity measurement, distances between each topic and every video clip is computed using the Bhattacharyya matching as following:

$$d_{Bhattach}\left(H_1, H_2\right) = \sqrt{1 - \sum_i \frac{\sqrt{H_1\left(i\right) \cdot H_2\left(i\right)}}{\sum_i H_1\left(i\right) \cdot \sum_i H_2\left(i\right)}} \quad (1)$$

where H_1 and H_2 are the query topic and the video clip histograms. The distances are sorted and the first 1000 lowest scores are returned as good matches. Figure 2 shows the performance of this run.

2.2. Run 2: Baseline run with SIFT only

2.2.1. Offline Indexing

Similar to the first run, one frame per second are extracted from every video clips and used to compute PHOW descriptors. We also used the SIFT code available from the *VLFeat toolbox* [1]. The descriptors are computed from 4×4 cells and with 8 bins for histogram of oriented gradients (HOG).

2.2.2. Online Indexing

The framework of online searching is presented in part of Figure 1. Given the image set of topic, we extracted the Region of Interest (ROI) using the related mask. Then the feature vector consists of PHOW descriptors are computed. For the search, each SIFT keypoint in the query topic is matched to its corresponding descriptors in the video clip database as proposed in [2]. The computed scores based on the squared Euclidean

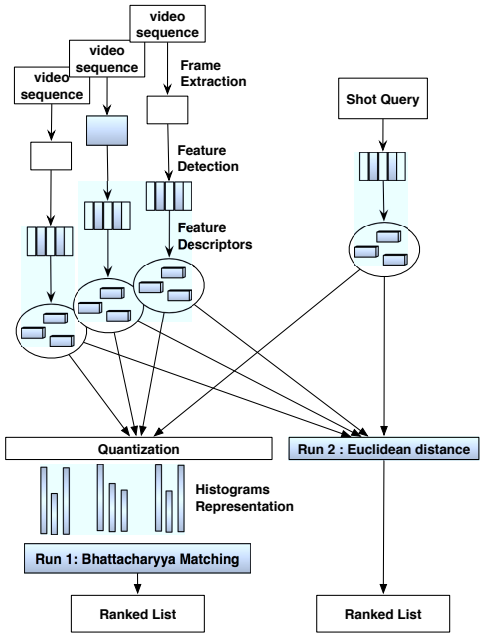

Fig. 1. Framework of online searching in first and second run

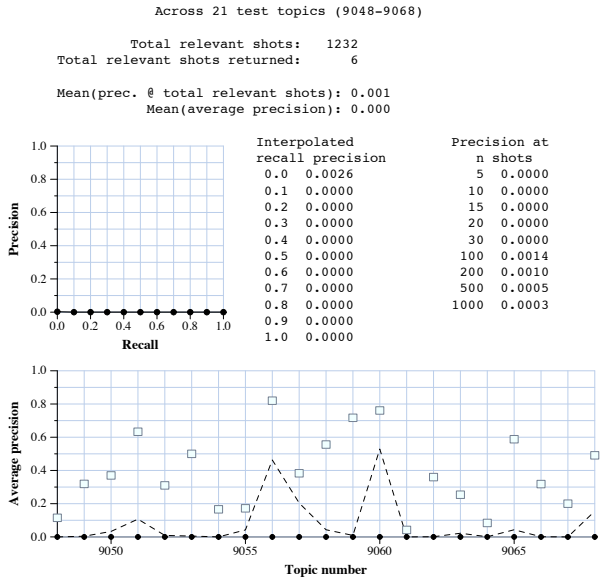

Fig. 2. Performance of the run R1

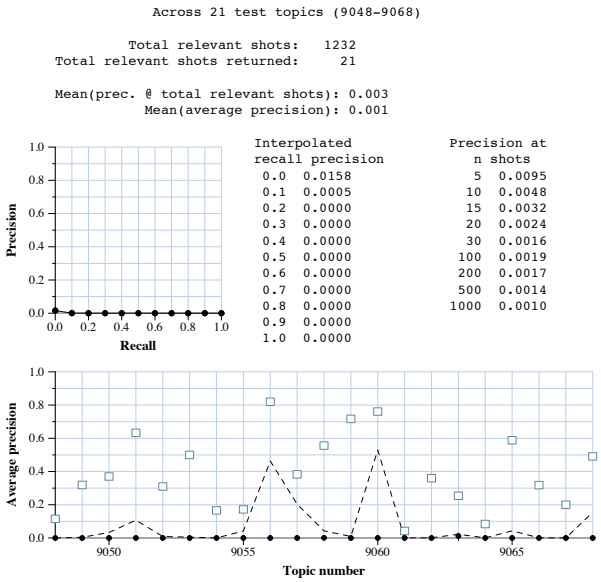

Fig. 3. Performance of the run R2

distance between the query topic descriptor and the closest descriptor in the video database. Finally, the highest scores are used as rank in the final result. The performance of this run is presented in Figure 3.

2.3. Run 3: IR based Approach

An IR-based framework is proposed to efficiently retrieve candidate images from large source collections. The source collection is indexed off line. The testing image is split into smaller queries. The index is queried against each query from the testing image to retrieve a set of potential source video segments. The top N images are selected for each testing image and the results of multiple queries merged using a score-based fusion approach [3] to generate a ranked list of source videos. The top K images in the ranked list generated by CombSUM are marked as potential candidate images.

Figure 4 shows the proposed process for retrieving candidate images using an IR-based approach. The source collection is indexed with an IR system (an offline step). The candidate retrieval process can be divided into four main steps: (1) pre-processing, (2) query formulation, (3) retrieval and (4) result merging. These steps are described as follows:

1. **Pre-processing:** This is the step for feature generation. Similar to the first two runs, for each of the suspicious document, SIFT features are calculated and histograms of those features are generated. These histograms are considered as sentences of any text document.

2. **Query Formulation:** Sentences from the suspicious document are used to make a query. The length of a

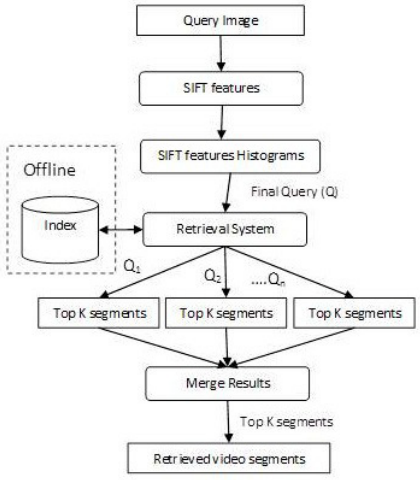

Fig. 4. Process of candidate document retrieval

query can vary from a single sentence to all the sentences appearing in a document, i.e. the entire image. A long query is likely to perform well in situations when large portions of image are similar. On the other hand, small portions of similar images are likely to be effectively detected by a short query. Therefore, the choice of query length is important to get good results.

3. **Retrieval:** Terms are weighted using the *tf.idf* weighting scheme. Each query is used to retrieve relevant source documents from the source collection.

4. **Result Merging:** The top N source documents from the result sets returned against multiple queries are merged to generate a final ranked list of source documents. In a list of source documents retrieved from a query, document(s) at the top of the list are likely to be the similar videos. In addition, portions of text from a single source document can be reused at different places in the same video segment. Therefore, selecting only the top N documents for each query in the result merging process is likely to lead to the original source document(s) appearing at the top of the final ranked list of the documents.

   A standard data fusion approach called CombSUM method [3] is used to generate the final ranked list of documents by combining the similarity scores of source documents retrieved against multiple queries. In the CombSUM method, the final similarity score, $S_{finalscore}$, is obtained by adding the similarity scores of source documents obtained from each query q:

$$S_{finalscore} = \sum_{q=1}^{N_q} S_q\left(d\right) \qquad (2)$$

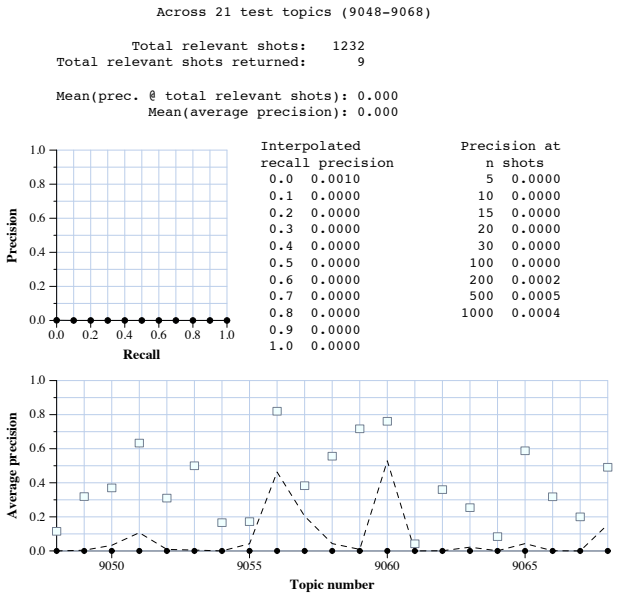

Fig. 5. Performance of the run R3

where N_q is the total number of queries to be combined and $S_q\left(d\right)$ is the similarity score of a source document d for a query q.

The top K documents in the ranked list generated by the CombSUM method are marked as potential candidate source documents.

2.3.1. Implementation

Two popular and freely available Information Retrieval systems are used to implement the proposed IR-based framework: (1) Terrier [4] and (2) Lucene [5]. In both Terrier and Lucene, terms are weighted using the *tf.idf* weighting scheme. In Terrier, documents against a query term are matched using the TAAT (Term-At-A-Time) approach. Using this approach, each query term is matched against all posting lists to compute the similarity score. In Lucene, the similarity score between query and document vectors is computed using the cosine similarity measure.

2.4. Run 4: Affine SIFT only

2.4.1. Pre-processing

There are two steps for pre-processing. One is for testing video. In order to reduce the data size, for one video, only four frames from start, middle and end position are selected to represent the content of this video. Furthermore, this four frames are composed into one frame by zooming the size of each frame to proper level. The other step is for queries. The mask image is adopted to remove the background of each

Fig. 6. Pre-processing for testing video

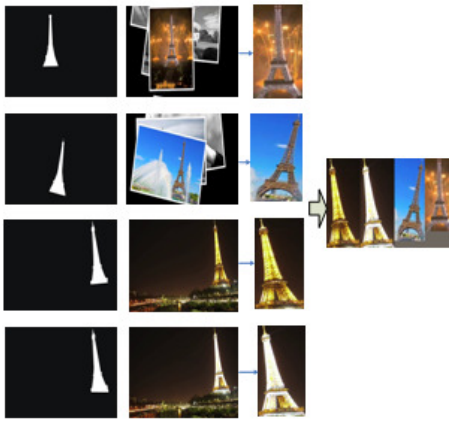

Fig. 7. Pre-processing for query

query. By shrinking the image size, four to five images of one query are also represented by one big image. The whole procedure is shown in Figure 6 and Figure 7.

2.4.2. Feature extraction and distance matching

The framework of searching is presented in Figure 8. We adopted the Affine-SIFT code available from [6] and extracted ASIFT feature for every testing frame and query image. Then we matched testing frame and query image by the fully affine invariant image comparison method [7].

3. CONCLUSION

In this paper we presented our experiments performed in the TRECVID 2012 instance search task. This participation rewarded us an experience in our researches and in finding new ideas and directions in the domain of object-based video retrieval.

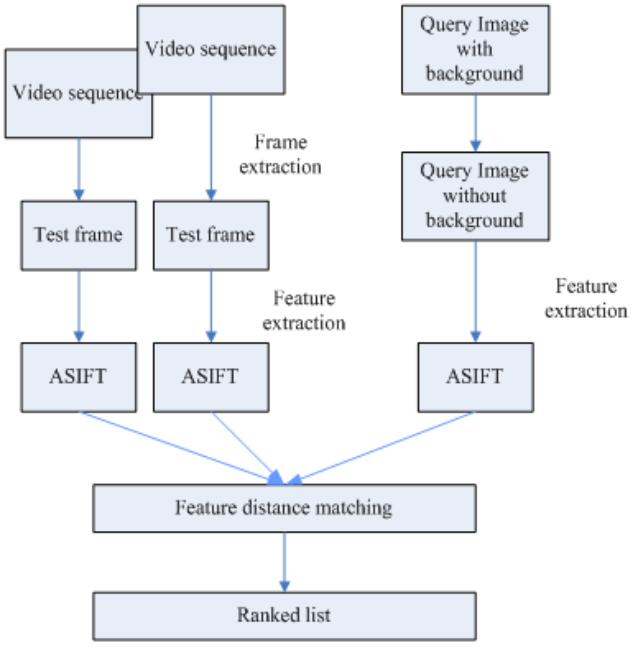

Fig. 8. Matching framework

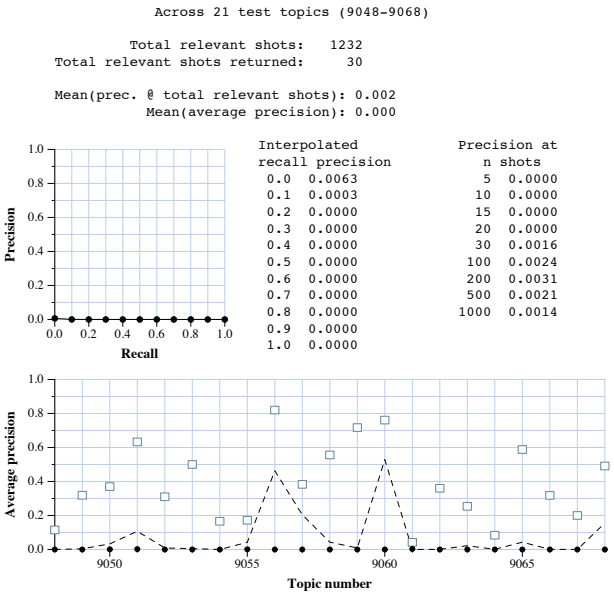

Fig. 9. Performance of the run R4

4. REFERENCES

[1] Andrea Vedaldi and Brian Fulkerson, "Vlfeat: an open and portable library of computer vision algorithms," in *Proceedings of the international conference on Multimedia*, New York, NY, USA, 2010, pp. 1469–1472, ACM.

[2] David G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, pp. 91–110, 2004.

[3] E. Fox and J. Shaw, "Combination of multiple searches," *NIST SPECIAL PUBLICATION SP*, pp. 243–243, 1994.

[4] I. Ounis, G. Amati, Plachouras V., B. He, C. Macdonald, and Johnson, "Terrier Information Retrieval Platform," in *Proceedings of the 27th European Conference on Information Retrieval*. 2005, pp. 517–519, Springer.

[5] E. Hatcher, O. Gospodnetic, and M. McCandless, *Lucene in Action*, Manning Publications, 2004.

[6] Guoshen Yu, Jean-Michel Morel, "ASIFT: An Algorithm for Fully Affine Invariant Comparison," *Image Processing On Line*, 2011, http://dx.doi.org/10.5201/ipol.2011.my-asift.

[7] Guoshen Yu and J.-M. Morel, "A fully affine invariant image comparison method," in *Acoustics, Speech and Signal Processing, ICASSP, IEEE International Conference on*, 2009, pp. 1597 –1600.