The France Telecom Orange Labs (Beijing) Video Semantic Indexing Systems – TRECVID 2012 Notebook Paper

Kun Tao[1], Yuan Dong [2], Yunlong Bian[2], Xiaofu Chang[1,2], Hongliang Bai[1], Wei Liu[1], Feng Zhao[1], Peng Li[1], Chengbin Zeng[1]

[1]France Telecom Orange Labs (Beijing), Beijing, 100190, P.R.China
[2]Beijing University of Posts and Telecommunications, Beijing, 100876, P.R.China
kun.tao@orange.com yuandong@bupt.edu.cn

ABSTRACT

In this paper, we describe FTRDBJ's systems and experiments for TRECVID2012 SIN task. This year, a new type of runs named "concept pair" was evaluated, for which we tried many different models. We submitted two "full" runs and two "pair" runs. For the "full" runs, our systems are very similar to the old ones used in 2011. A 6-feature and a 9-feature composite-kernel SVM systems were used to detect the concepts. For the "pair" runs, we submitted a confidence based run and an OWA based run. The final results show that the first one worked better because a simpler model usually works more robust.

1. INTRODUCTION

In TRECVID2012 Semantic Indexing task, the size of IACC data corpus kept increasing and more annotations were provided, which can help the participants to get more reliable assessment of their systems. LIG & LIF provided a new optimized annotation corpus. Using relationships between concepts, the annotations can be greatly expanded after propagation [1]. Our experiments proved that using the new annotation corpus can improve the system performance and all our submitted runs are based on it.

Although the number of concepts is still 346, the subsets selected for light runs and for final evaluation are quite different from those of 2011 [2]. Based on the early fusion strategy which has been proved in our previous works, two "full" runs were submitted. One of the "full" runs is based on 6-feature composite-kernel SVM. For the reason of time limitation, we only chose 100 concepts to train the 9-feature SVM models. Combining the results from 9-feature models and 6-feature models, another run was submitted. Although no new technologies were added, the systems showed reliable performance.

This year, two new versions of SIN task were added: "concept pair" and "no annotation". We are very interested in "concept pair" task because it's similar to the real application scenes that users may submit queries with multiple keywords. Regarding this task as a late fusion problem, we tested different fusion strategies and finally submitted two "pair" runs. The basic information of our 4 runs is shown below:

- F_A_FTRDBJ-SIN-1_1: Composite Kernel SVM with 9 features or 6 features. MAP = 0.206.
- F_A_FTRDBJ-SIN-2_2: Composite Kernel SVM with 6 features. MAP = 0.195.
- P_A_FTRDBJ-SIN-3_3: Pair concepts fused by confidence. MAP = 0.071.
- P_A_FTRDBJ-SIN-4_4: Pair concepts fused by OWA. MAP = 0.032.

More details about our systems and experiments will be introduced in the sections below.

2. SINGLE CONCEPT INDEXING

In last several years, the early fusion method based on composite-kernel SVM was proved to be very effective [3]. Therefore we kept using it for our SIN systems in TRECVID2012. Totally 9 kinds of visual features were extracted and their dimensions are show in Table 1.

Table 1. Visual Features

Feature Name	Dimension
CCV	360
GCM	108
LBP	2180
HOG	512
sift~no_orientations.hists	512
dense-opsift.hist	512
sift.vw.hists	512
dense-sift.vw.2L-pyramid-hist	2560
hog.vw.2L-pyramid-hist	2560

Using the first 6 features, the models were trained for our second run F_A_FTRDBJ-SIN-2_2. During the training of composite-kernel SVMs, the kernels were given

different weights which are proportional to the single feature performances [3].

Last year, our experiments show that using 8 features including 2-level pyramid BOW features will be better than only using 5 features. But using more features will bring several times more calculation cost. This year only 100 concepts were selected (including the 50 concepts for Light runs and the concepts in concept pairs) by us and for which all 9 features were used to train the models. Above 100 models and the other 246 models trained by 6 features were than used in our first run F_A_FTRDBJ-SIN-1_1.

3. CONCEPT PAIR FUSION

The "concept pair" task aims to detect pairs of unrelated concepts. Intuitively, it's similar to the textual queries using multiple keywords. It can be dealt as a fusion problem or a re-ranking problem. The final goal of this task is to find a combination of concepts that do better than just combining the output of individual concept detectors. But final results show that it's harder than expected to reach above goal because there're two big obstacles:

1) The lack of training samples, especially positive-positive samples (P-P samples, both of the two concepts are labeled as positive). With the increasing of IACC shot numbers, the annotations can no longer cover all shots in data corpus. Although for two individual concepts there're enough annotated labels, their overlap part that can be used to train/evaluate a "concept pair" model might be much smaller. Especially for the pair 905 and 906, there's only 1 and 0 P-P sample. Thus our experiments were all taken on the other 8 concept pairs, and the parameters for the final models of 905 and 906 were calculated by averaging the parameters of the other concept pairs.

2) The precision unbalance of two individual detectors. In most cases, a more common concept with more training samples can get a better detector than that of a rare concept with little samples. The precisions of two detectors sometime can reach a difference of more than 10 times. The estimated MAPs of 16 individual detectors used in our experiments are shown in Table 2. Such unbalance should be taken seriously in fusion models.

Table 2. Performance of Single Detectors

Concept Pair	AP of 1st Detector	AP of 2nd Detector
901	0.202	0.324
902	0.076	0.057
903	0.065	0.182
904	0.029	0.295
907	0.9	0.1149
908	0.321	0.087
909	0.124	0.154
910	0.168	0.048

Regarding the concept pair combination as a late fusion problem, different methods were tried in our experiments. First, the single detectors were trained by 9-feature SVM. Then their outputs were transformed into 4 kinds of scores which can be used as the input of fusion models: The original scores of SVM, ranking scores in all tested shots and the normalized scores of them. Finally, we used 3 kinds of weighted average strategies and 4 kinds of study based fusion strategies to accomplish the fusion processing:

1) Equal weight average.

2) Using confidence coefficients as the weights. Confidence is widely used association analysis [4]. It has a simple form like:

$$c(X_i \to X_{i,j}) = \frac{\sigma(X_i \cup X_j)}{\sigma(X_i)} = \frac{Num(concept1, concept2)}{Num(concept1)}$$

X_i means the existing of concept i, $X_i \cup X_j$ means the existing of both concepts.

3) Using $c(X_i \to X_{i,j}) * AP_i$ as the weights.

That means not only the confidence but also the reliability of the detectors is considered.

4) Logistic regression.

The weights were trained by logistic regression using LIBLINEAR [5].

5) Ordered weighted averaging (OWA)

OWA has been proved to be valuable in many applications [6]. First, the inputs are sorted by values. Then the weights will be given to the inputs in corresponding sorting positions instead of the inputs from corresponding sources:

$$F(a_i \cdots a_n) = \sum_{j=1}^{n} w_j b_j$$

Where b_j is the j_{th} largest of a_i. The weights for sorted inputs were also trained by LIBLINEAR.

6) Rankboost.

Rankboost is also a widely used fusion algorithm, especially for retrieval applications. More details about it can be found in [7].

7) SVM.

The linear kernel was used to train the SVM based fusion models.

Using above 4 kinds of inputs and 7 kinds of algorithms, we tried different strategies. In order to compare the effects of score based inputs and rank based

inputs, two runs were selected for final submission: one is based on normalized score and confidence, while another is based on ranking score and OWA.

4. EXPERIMENT RESULTS

4.1 Annotation Corpus Selection

This year two kinds of annotation corpus were provided. One is the raw version containing all and only the direct annotations. The other is the full version after propagation using relations. The total number of raw annotations from 2010 to 2012 is about 5M, and after propagation it increased to about 20M. Regarding that the propagation can enrich the label coverage and eliminate some error labels, it usually brings better results.

A comparison experiment was taken before testing the other algorithms. The labels of 50 concepts (for light runs) in two annotations corpuses were all divided into training parts (70%) and testing parts (30%). Using 6-feature composite kernel SVM, the effectiveness of two corpuses was evaluated.

Table 3. Annotation Corpus Testing

MAP	Trained by Raw	Trained by Full
Tested on Raw	0.345	0.363
Tested on Full	0.293	0.305

The results in Table 3 show that using full version annotation is better. Thus all our systems were trained on it.

4.2 Single Concept Indexing

Our systems for single concept indexing are based on 6 features or 9 features. The system performances based on above two strategies were tested on 50 concepts for light runs and are shown in Table 4. (60% of all developing data are used for training, 40% for testing)

Table 4. 6-Feature/9-Feature Comparision

MAP	C=0.1	C=1.0	C=10.0
6-Feature	0.241	0.305	0.308
9-Feature	0.269	0.313	-

In final NIST evaluation, the first run of 9-feature got a MAP of 0.206, and the MAP of second run is 0.195. As we mentioned above, only part of the concepts in the first run were really trained by 9 features. 21 of them were finally included in the 46 full run concepts evaluated by NIST. Considering the MAP of the 21 concepts, the MAP of 9-feature increases to 0.223, while the MAP of 6-features is only 0.198. That means although using more features is more time-costing, it can bring better performance.

4.3 Concept Pair Fusion

In section 3 seven different strategies are mentioned, which are all late fusion models. In fact, we also trained two kinds of models as the baselines which were directly trained by regarding "concept pair" as one complex concept. First positive-positive samples were regarded as positive. Then for Type-1 baseline only negative-negative (N-N) samples were regarded as negative, while for Type-2 all N-N, N-P and P-N samples were regarded as negative. The two baselines were trained by 9-feature SVM and tested on 8 concept pairs. The corresponding results are shown in Table 5.

Table 5. Baselines for Concept Pair

MAP	C=0.1	C=1.0
Type-1	0.0764	0.0815
Type-2	0.0784	0.082

The results of different late fusion methods for concept pair detection are shown in Table 6. All MAPs are only for 8 concept pairs.

Table 6. Results of Late Fusion Methods

MAP	Score	Score-Norm	Rank	Rank-Norm
Equal-Weight	0.074	0.08	0.068	0.068
Confidence	0.081	**0.082**	0.07	0.07
Confidence*AP	0.068	0.072	0.067	0.067
L-R	0.081	0.001	0.067	0.068
OWA	0.072	0.001	**0.076**	0.067
Rankboost	-	-	0.06	-
SVM	0.064	0.07	-	0.069

For above two tables, the corresponding single detectors or baseline models were trained on 60% of the developing data, then the other 20% were used to train fusion models if needed and the last 20% were used to test the final system performance.

Among results in Table 6 we can found someone whose performance is very close to the baselines. But the more complex models based on rankboost and SVM didn't perform well. That was caused by overfiting. In order to evaluate the different types of inputs, we finally submitted a best system using scores (Score-Norm/confidence) and a best system using ranks (Rank/OWA).

In the submitted runs evaluated by NIST, our run P_A_FTRDBJ-SIN-3_3 got the 2nd high MAP. Analysing the delivered result files, we found that most runs failed on the concept pair 907 (Person + Underwater). That might be

caused by the precision unbalance of single detectors because the concept "Person" is very popular. Fortunately, using confidence can avoid such problem. The system based on confidence also showed good robustness on the other concept pairs while many other methods were puzzled by overfitting and sometimes even worse than only using simple combinations.

In 2012 the evaluation of concept pair SIN was severely effected by the lack of annotations, since most complex models should be trained on enough samples. Anyway, there're some simple models which are more robust and can be used in similar cases. With the accumulation of data corpus, many study based models will show their abilities in the future.

5. CONCLUSION

In the last two years, our systems based on composite-kernel SVM kept working well. But at the same time some new features and classifiers were proposed by other participants and show very good performance. Thus we plan to pay more attention on studying new features in the future and try to improve our systems step by step. In the concept pair task, the results of 2012 show that using a simple model will bring more robustness in case that the samples are insufficient. We're also trying to find a better way to overcome the difficulties cased by insufficient samples and detector precision unbalance.

6. REFERENCES

[1] G. Quénot etc. "TRECVID 2012 Collaborative annotation", http://mrim.imag.fr/tvca2012/

[2] "Guidelines for the TRECVID 2012", http://www-nlpir.nist.gov/projects/tv2012/tv2012.html.

[3] Y. DONG, etc. "The France Telecom Orange Labs (Beijing) Video Semantic Indexing Systems – TRECVID 2011 Notebook Paper," http://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.11.org.html, 2011.

[4] Pang-Ning Tan, etc. "Introduction to Data Mining", Addison-Wesley, 1 edition , 2005.

[5] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," Journal of Machine Learning Research 9(2008), 1871-1874.

[6] Yager, R. R. and Kacprzyk, J., "The Ordered Weighted Averaging Operators: Theory and Applications", Kluwer: Norwell, MA, 1997.

[7] Y. Freund etc. "An Efficient Boosting Algorithm for Combining Preferences", Journal of Machine Learning Research 4 (2003).

FRANCE TELECOM ORANGE LABS (BEIJING) AT TRECVID 2012: INSTANCE SEARCH

Hongliang Bai[†]*,Yuan Dong*[‡]*, Lezi Wang*[‡]*, Chong Huang*[‡]*, Nan Zhao*[‡]*, Shusheng Cen*[‡]*, Kun Tao*[†]

[†]France Telecom Research & Development - Beijing, 100190, P.R.China
[‡]Beijing University of Posts and Telecommunications,100876, P.R.China
{hongliang.bai,kun.tao}@orange.com
{yuandong,wanglezi,huangchong661100,zhao.nan07,censhusheng}@bupt.edu.cn

ABSTRACT

The framework of TRECVID INS2012 task is introduced by France Telecom Orange Lab (Beijing). It is the first time that we participate in the very challenging task. One interactive and three automatic runs have been submitted, namely:

F_X_NO_FTRDBJ_1: SIFT feature, Vocabulary Tree(VT) clustering, Bag of Words(BOW), video indexing and searching, confuser extraction, geometry verification.

F_X_NO_FTRDBJ_2: SIFT feature, VT clustering, BOW, video indexing and searching, geometry verification.

F_X_NO_FTRDBJ_3: SIFT feature, Approximate K-Means(AKM) clustering, BOW, video indexing and searching, confuser extraction, geometry verification.

I_X_NO_FTRDBJ_4: SIFT feature, random walk based relative feedback.

After experiments, the mAP performances of above four runs are 0.105, 0.081, 0.071 and 0.251 respectively. The interactive run is better than the automatic runs. It is consistent with our experience.

Index Terms— TRECVID, INS, Vocabulary Tree, Approximate K-Means, Relative Feedback

1. INTRODUCTION

Video explosion is happening as the large number of movies, TV program streams, personal-made clips are uploaded into the web. The requirements for searching some special videos are more and more strong because the videos are delighting, instructive or useful. Americans viewed a record 16.8 billion videos online in 2009 April driven largely by surge in viewership at YouTube [1].

Many state-of-art methods or algorithms have been proposed to meet with the above requirements in the recent years. They mainly include videos crawbing, feature extraction, feature encoding, video search and reranking. In the feature extraction stage, the local feature is most frequently used. Its extraction basically has two steps; one is feature detectors, such as Harris detector, Harris Laplace detector, Hessian Laplace, Harris/Hessian Affine detector, and the other is feature descriptors, such as Scale Invariant Feature Transformation (SITF) [2], Shape Context, Gradient Location and Orientation Histogram [3], Speeded Up Robust Features(SURF) [4], DAISY [5]. The dimension of feature can be reduced by PCA. Feature encoding can improve the search efficiency, such as the spatial pyramid, BOW [6], fisher vector [7]. The video search can be implemented by k-d tree, hashing-based, LSH [8], product quantization [9], and so on. The reranking usually regarded as machine learning problems, such as Rank SVM [10], IR SVM, AdaRank. The learning-based algorithm integrates both the initial ranking and visual consistency between images [11].

It is necessary to measure the cons and pros of the different methods in the same database. TREC Video Retrieval Evaluation(TRECVID) INS [12, 13] provides us a good platform to demonstrate and compare the different methods.

The rest of paper will introduce our work to implement the INS task. Section 2 and 3 are our automatic and interactive pipelines and algorithm details. The experiments and discussion are in the section 4. Finally, we will propose the future work to improve current performance.

2. AUTOMATIC INSTANCE SEARCH

2.1. System Pipeline

The basic structure of our automatic instance search consists of two stages: off-line and online stages, shown in Fig. 1.

In the off-line stage, a large-scale codebook tree is trained and the reference video dataset is processed. Frames are firstly sampled from each video corpus and then visual features are extracted from each frame. With the SIFT library, we averagely sample the descriptors, which are used to build a vocabulary tree [14] or k-d tree [15]. The reason we utilize the hierachical structure for codebook is that a large scale codebook is allowed, as well as the retrieval efficiency could be improved. Next we project the SIFT descriptors of reference dataset into the tree in video-based format for VT and in frame-based format for k-d tree. At the same time TF-IDF weighting strategy is applied to the quantization value. In the

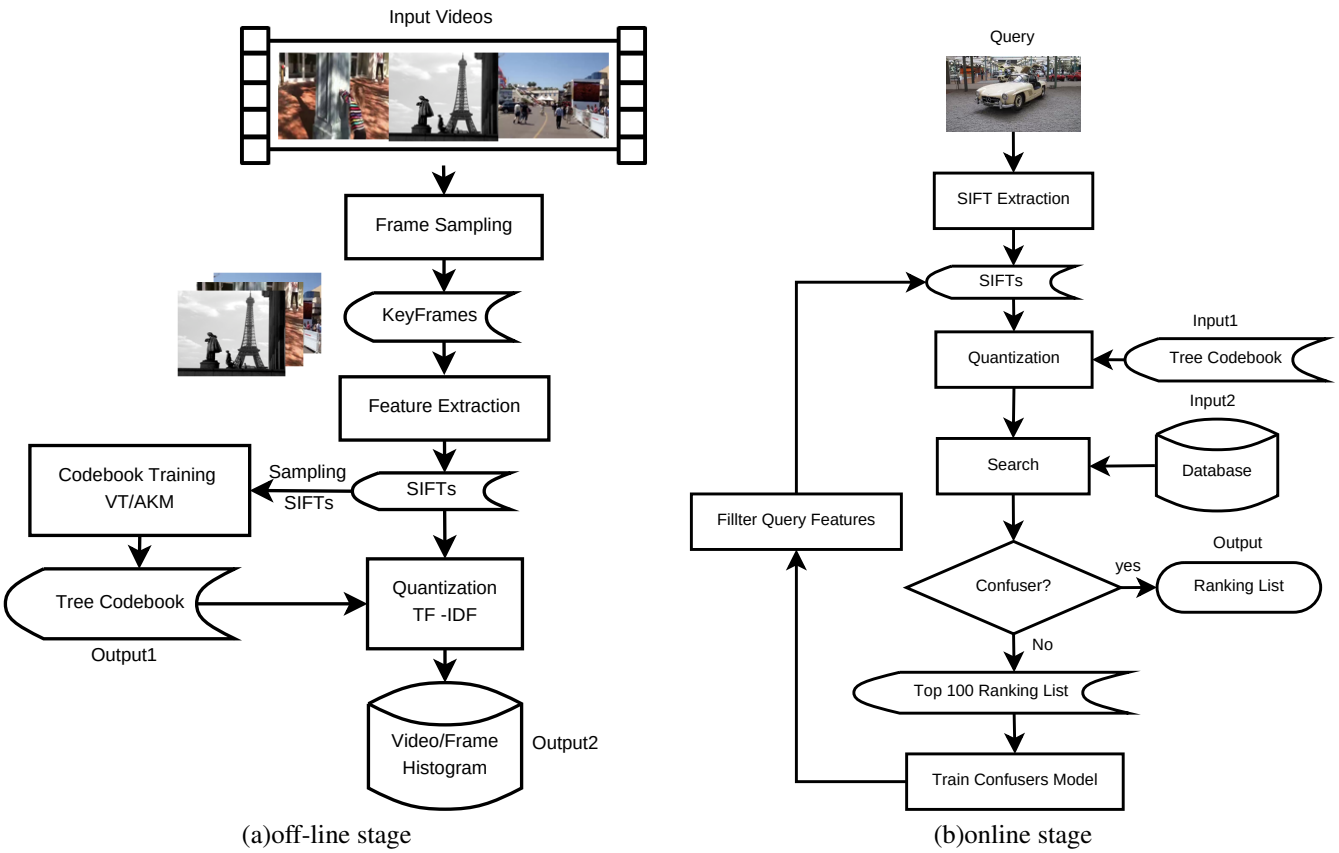


Fig. 1. Automatic instance search pipeline

end the system outputs a large scale codebook tree and a high-dimensional histogram for each video or frame.

In the online stage, given a query image, we first extract its SIFT descriptors and project them into the codebook tree using the same bin-weighting strategy of reference keyframes. Then, an initial ranking list is generated by measuring the similarity between the query and items in the reference database. Through our experiments, we figure out that a large amount of noises exist in the database, for example, grass or ocean without any salient object in an image, which leads to the TF-IDF failure. To get rid of the noise interference, we want to obtain a clean query object by eliminating the irrelevant features. Thereby, the top 100 initial ranking items are fetched to train a confuser model as done in [16]. After that, the query is filtered by the model to remove the irrelevant features. Meanwhile, features located in the mask region are threw back for the quantization to avoid over-filtered case. The later process is repeated to get the final ranking list.

2.2. Feature Extraction

We extract 3 frames per second from the 74958 videos (totally 2,082,277 frames) in TRECVID INS 2012. The frame rates of the videos vary in 320×240, 640×360, 640×480, 640×1138 and 160×120. The normalized size of each frame is 352×288. The Harris-Laplace/MSER and SIFT local features are extracted as detectors and descriptors of above frames, respectively.

2.3. Vocabulary Tree-based Image Retrieval

VT [14] is adopted in our system to improve retrieval efficiency for a large scale database. In the offline step, 70K keyframes are randomly sampled from reference database, which extracts around 75M SIFT descriptors, 35M and 40M for MSER and Harris-Laplace detectors respectively. Then, a tree is built with 100-branching and 3-level factors, resulting in 1M leaf nodes. All the SIFT descriptors of each video are projected into the VT to form a high sparsity histogram. At each leaf node, the TF-IDF weighting strategy is adopted, based on L1 normalization. In this way, each video in the gallery are represented by a high-sparsity bin-weighted histogram.

In the online phase, from the given query image the two sorts of SIFT descriptors are extracted, which are then projected into the already built vocabulary tree. Therefore, the prior query image is represented by a bin-weighted histogram.

It is followed by measuring the L1 metric based similarity between each topic with every video clip. Finally, a ranking list for each topic is accessed.

2.4. AKM-based Image Retrieval

In this run we used Bag-Of-Words encoding algorithm at the frame level. SIFT descriptors are extracted each frame, which is densely extracted from the video set. The set of descriptors is clustered and quantified by approximate k-means clustering. Each frame is represented by a sparse high-dimension histogram with adopting the TF-IDF weighting. On the online searching phrase, we consider maximized similarity at frames level as the similarity between query and reference videos, which is computed by the histogram intersection. Details about our algorithm is presented as follow:

Offline Indexing: we randomly sample 50M descriptors for the clustering. Because the vast majority of computation time is spent on calculating the nearest neighbours between the points and cluster centers, the exact k-means based nearest neighbour is computationally expensive. We adopt the approximate k-means(AKM) [17] with 1M clusters, which could speed up the computation of the assignments in each iteration by an approximate nearest neighbor (ANN) method. The multiple k-d trees are selected as the data structure storing clusters. In this way, the assignment of data is turned into the approximate nearest cluster problem. A forest of 8 randomized k-d trees is used to built over the cluster centers at the beginning of each iteration to increase the speed. In each k-d tree, each node splits the dataset using the dimension with the highest variance for all the data points falling into that node and the value to split on is found by taking the median value along that dimension (although the mean can also be used).

The way of feature quantization is similar with clustering. We use bin-weighted histogram to present the set of quantified feature: each bin multiplied by the number of data assigned to corresponding cluster and TF-IDF weighting. For memory consideration, we take the sparsely index-value storage method (5k bytes per frame in average).

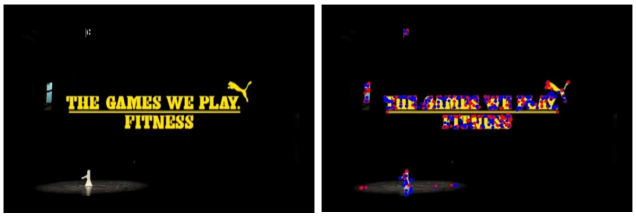
Online Searching: Given that all the frames are represented by the set of sparse histogram, the similarity of frames between query and reference is measured by histogram intersection metric. Then we compute the similarity of videos by choosing the maximized similarity at the frame level. Finally, the similarities among videos is ranked to select the top 1000 as our initial result.

2.5. Video Reranking by Confuses Detector

The VT-based or AKM-based BOW encoding and DF-IDF weighting generates the initial results. The spatial verification can improve the search performance in the query object region and database images. RANSAC [18] is used in spatial verification. Our confuser detector is inspired by Total recall II [16]. The confuser model words W_c:

$$W_c = \{w|P(w|S)/p(w) > \gamma_0\} \quad (1)$$

where S is the learned shortlist, γ_0 is the predefined threshold, here is set 3. In our system, local and global confusers are learned. The global confuser is used to erase the common background noise, such as sea wave, leaves, grass. The local confusers find the special noise for some querying image. In Fig. 2, the most confusers exists in the text regions.



(a) (b)

Fig. 2. In some query frame (a), the confuser points are shown (b). NOTE: blue points are the confuser points, and red ones are general SIFT points

3. INTERACTIVE INSTANCE SEARCH

The objective of our interactive search system is taking advantage of users' feedback to boost the performance by bridging the gap between high-level semantics and low-level image features. The proposed approach is based on Random Walk Algorithm which is introduced by Leo Grady [19] for image segmentation, and we model the search task as a graph-theoretic problem. The system pipeline is shown in Fig. 3 and details of each module is described as follows.

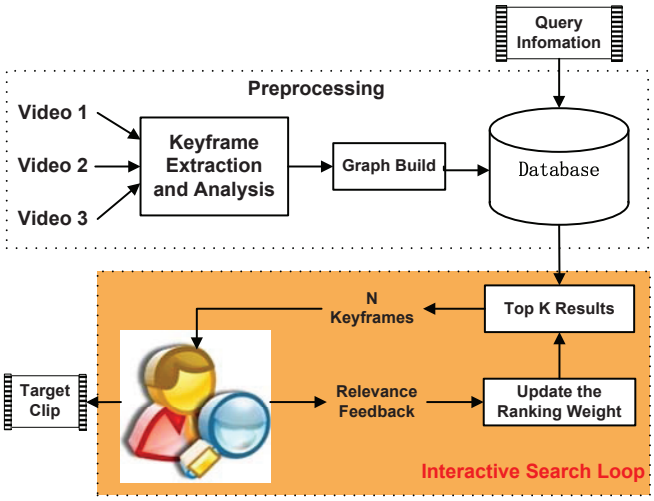


Fig. 3. Interactive instance search pipeline

3.1. Pre-processing

In pre-processing step, the content of each reference video is represented by a vector and the graph used for the interactive indexing is built based on those vectors.

keyframes extraction and analysis: keyframes are temporal sub-sampled with uniform sampling rate of 1 frame per second and feature are extracted, forming a vector to describe the video content as Section 2.3 described.

Graph build: A graph is a pair $G = (V, E)$, where V is set of nodes (video vectors) and E is the set of edges connecting two nodes. We assign a weight w to specific edge(ij), where the w is defined as the similarity of two vectors, and generate the adjacent matrix $W = (w_{ij})$. And the weighted Laplacian matrix can be written as $L = D - W$, where $D = (d_{ij})$ is a diagonal matrix with $d_{ii} = \sum_{j \in V} w_{ij}$.

3.2. Relevance Feedback

Given a query topic, the system returns the initial top $K(K = 2000)$ results with the highest low-level similarities and enters the interactive search loop. Users are shown keyframes of top $N(N = 100)$ videos based on the ranking weight X_i^r at the r-th feedback round. At each round, users should label each video as "relevant" or "non-relevant", where the ranking weight of relevant file equals 1 and 0 non-relevant. Users' feedback is used for computing the ranking weight for unlabeled videos, by solving the following equation:

$$L_{UU} X_U = -L_{UM} X_M \tag{2}$$

where laplacian matrix L can be written as

$$L = \begin{bmatrix} L_{UU} & L_{UM} \\ L_{MU} & L_{MM} \end{bmatrix}$$

The ranking vector $X_i \in [0, 1]$ and X can be written as $X = [X_U, X_M]$, and U stands for unlabeled and M indicates labeled. The submitted result is the ranked K video files based on the ranking weights computed at the last round.

4. EXPERIMENTS

4.1. Database Description

TRECVID is a laboratory-style evaluation that attempts to model real world situations or significant component tasks involved in such situations. The INS task is to find more video segments of a certain specific person, object, or place, given a visual example. In 2012, 21 topics are included, in Fig.4. One person, 14 landmarks and 6 objects/logos are in the query list. Automatic and interactive systems will use the same set.

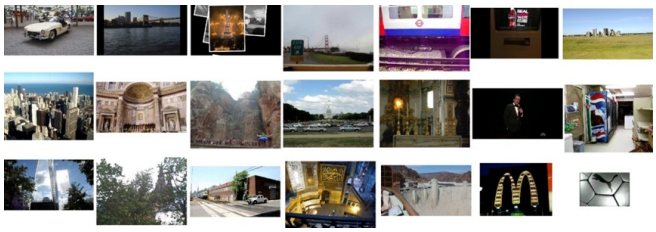
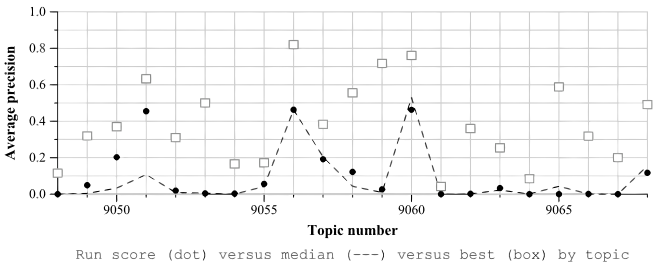


Fig. 4. Querying samples in INS2012,with index 9048∼9069

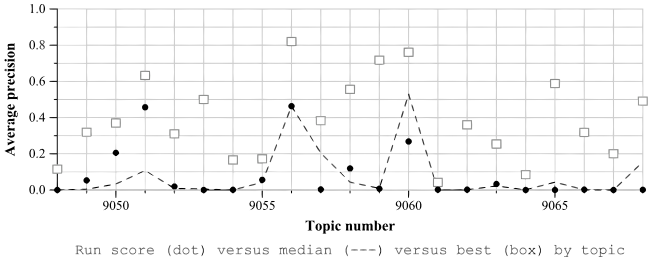
4.2. VT-based Query Performance

In this algorithm, we submitted two runs. The difference between them is whether the confuser detector is used. When the confusers are imported into the system, and the mAP is improved from 0.081 to 0.105, in 5. In Fig.(a), the worse topics are in querying for 9048(Benz), 9061(Pepsi), 9062(World Trade Center), 9064(tower), 9065(indoor of a unknown landmark), 9066(dam), 9067(McDonald). The mAPs of the worst topics are closed to 0. The object regions are small in the query 9048, 9061, 9066 and 9067. The number of local features is low in the query 9062 and 9066.

The best querying is 9051(Golden Bridge), 9056(indoor of a unknown landmark) and 9060(actor). The mAPs of the best topics are 0.455, 0.463 and 0.463 respectively.



(a) Confuser detector is used



(b) Confuser detector is not used

Fig. 5. VT-based automatic query results

4.3. AKM-based Query Performance

In this year, the mAP performance of AKM-based query is 0.071, which is worse than VT-based one. The worst querying topics are similar with VT-based one. The best querying

is only 9051(Golden bridge) and 9056(indoor of a unknown landmark). We also compared the performances of AKM-based and VT-based algorithms in 2011, whose mAPs are 0.458 and 0.416 respectively.

We also can see there is a big gap between the mAP 0.458 in 2011 and 0.071 in 2012. The 2011 querying images and reference videos are duplicate or near duplicate. So the searching tasks in 2011 are quite simpler than in 2012.

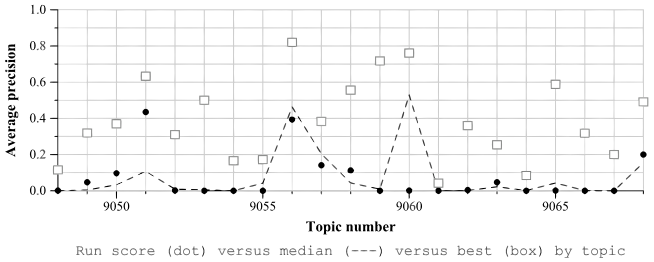


Fig. 6. AKM-based automatic query results

4.4. Interactive INS Performance

The interactive INS task is very interesting and useful because it can solve the lower performance problem by the user feedback style. The feedback can improve the performance. The mAP is up to 0.251 in Fig.7. In many query topics, the interactive run is best among our submitted runs, such as 9048(Benz) and 9067(McDonald).

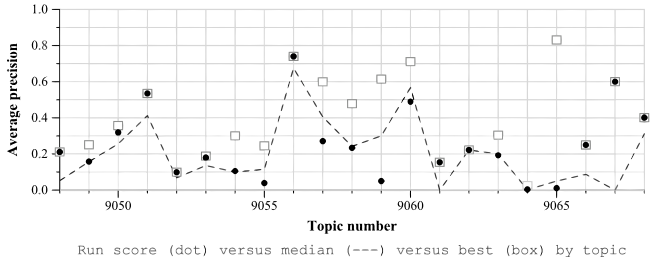


Fig. 7. Interactive query results

5. CONCLUSIONS AND FUTURE WORKS

The VT-based and AKM-based video retrievals are state-of-art algorithms. When they are used, the baseline performance is achieved in TRECVID INS2012. From the experiment, the interactive INS are better than the automatic INS. In the future, the codebook generation, multiple codebook fusion and reranking will be our research focus.

6. REFERENCES

[1] A. Lipsman, "Americans viewed a record 16.8 billion videos online in april driven largely by surge in viewership at youtube," http://www.comscore.com/Insights/Press_Releases/2009/6/Americans_Viewed_a_Record_16.8_Billion_Videos_Online_in_April, JUN 2009.

[2] D. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[3] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *PAMI*, vol. 27, no. 10, pp. 1615–1630, 2005.

[4] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[5] E. Tola, V. Lepetit, and P. Fua, "Daisy: an Efficient Dense Descriptor Applied to Wide Baseline Stereo," vol. 32, no. 5, pp. 815–830, May 2010.

[6] F. Li and P. Pietro, "A bayesian hierarchical model for learning natural scene categories," in *CVPR*, 2005.

[7] H. Jegou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, and C. Schmid, "Aggregating local image descriptors into compact codes," *PAMI*, vol. 34, no. 9, pp. 1704–1716, 2012.

[8] Piotr Indyk and Rajeev Motwani, "Approximate nearest neighbors: towards removing the curse of dimensionality," in *STOC '98: the thirtieth annual ACM symposium on Theory of computing*, 1998, pp. 604–613.

[9] H. Jegou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *PAMI*, vol. 33, no. 1, pp. 117–128, 2011.

[10] R. Herbrich, T. Graepel, and K. Obermayer, "Large margin rank boundaries for ordinal regression," in *Advances in Large Margin Classifiers*, 2000, pp. 115–132.

[11] Y. Jing and S. Baluja, "Visualrank: Applying pagerank to large-scale image search," *PAMI*, vol. 30, no. 11, pp. 1877–1890, 2008.

[12] Alan F. Smeaton, Paul Over, and Wessel Kraaij, "Evaluation campaigns and trecvid," in *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, New York, NY, USA, 2006, pp. 321–330, ACM Press.

[13] Paul Over, George Awad, Martial Michel, Jonathan Fiscus, Greg Sanders, Barbara Shaw, Wessel Kraaij,

Alan F. Smeaton, and Georges Quenot, "Trecvid 2012 – an overview of the goals, tasks, data, evaluation mechanisms and metrics," in *Proceedings of TRECVID 2012*. NIST, USA, 2012.

[14] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *CVPR*, 2006, vol. 2, pp. 2161–2168.

[15] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, no. 9, pp. 509–517, 1975.

[16] O. Chum, A. Mikulik, M. Perdoch, and J. Matas, "Total recall ii: Query expansion revisited," in *CVPR*, 2011, pp. 889–896.

[17] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman., "Object retrieval with large vocabularies and fast spatial matching," *In Proc. CVPR*, 2007.

[18] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, June 1981.

[19] G. Leo, "Random walks for image segmentation," *PAMI*, vol. 28, pp. 1768–1783, 2006.