Nanjing University at TRECVID 2010

Content-based Copy Detection Task

Ying Lin[1], Yang Yang[1], Kang Ling[1], Jinwei Xiao[3] , Pei Yang[3], Gangshan Wu[2]

State Key Laboratory for Novel Software Technology, Nanjing University

keller0618@163.com, gswu@nju.edu.cn

{yang,xiaojw,lingkang}@graphics.nju.edu.cn

ABSTRACT

This year, we participated in TRECVID 2010 content-based copy detection task. In this notebook paper we will describe our work in details. Different from last year when we just used SURF (speeded up robust feature) as visual feature, this year we employed a combination of four different features for our rough detection process: global SURF, center SURR, global color correlogram and center correlogram. What highlights our work is that , to achieve high accuracy detection ,we proposed a method using Non-orthogonal Binary Subspace in our accurate detection process. Finally we submitted 4 different runs in CBCD task, their description are as follows:

1.NJU.m.nofa.norank1:pre-process+feature extraction+search+post process 1

2.NJU.m.balanced.rank1:pre-process+feature extraction+search+post process 2

3.NJU.m.nofa.comp2:pre-process+feature extraction+search+post process 3

4.NJU.m.balanced.comp2:pre-process+feature extraction+search+post process 3

The only difference among these runs is that when producing the final result, we adpoted different post process methods.

1.INTRODUCTION

Nowadays with the rapid development of the modern science, the digital and multimedia technology are improving at full speed, which results in the accumulation of voluminous broadcasting of multimedia contents. As a result, the search of copies in large video databases has become a new critical issue. It is essential for many applications, such as copyright enforcement, web search improvement, advertisement monitoring, video retrieval by examples, redundancy reducing, concept tracking, etc[1].

Consequently, content based copy detection (CBCD) has come into focus for researchers to solve this challenge.

A copy is a segment of video derived from another video, usually by means of various transformations such as addition, deletion, modification (of aspect, color, contrast, encoding, ...), camcording, etc. [1]

This paper proposed a novel way to cope with the CBCD task at TRECVID 2010. In this proposed system framework, generally speaking, our system mainly consists of five key steps: query pre-process, feature extraction, rough detection, accurate detecion and post process. Firstly we coped

with several types of transformation in query videos, than we employed a combination of four different features as visual feature for our rough detection process.After that, what highlight our job, we adopted a method using NBS(Non-orthogonal Binary Subspace) in our accurate detection process to produce a list of similar frames to each query. Finally, based on these similar frame result list, we carried out a schema of post process to generate final results.

The rest of paper is organized as follows: The framework of our system is presented in section 2, and this part also describes the details of the methodology and technology of our approach. Section 3 gives the video copy detection evaluation results and explanation. Finally, the conclusion and future work are given in section 4.

2.OUR COPY DETECTION SYSTEM

Figure 1 illustrates the system framework of our approach.

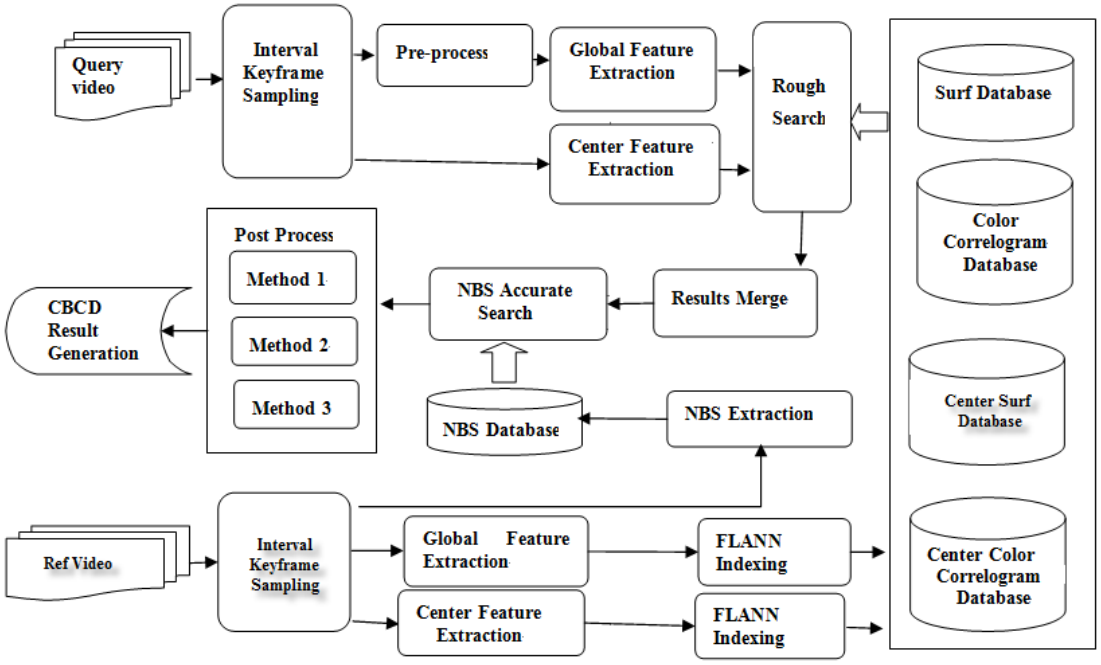

Figure 1: system framework

We process the query videos and reference videos separately. For reference videos, we firstly adopted interval frame sampling, and then we extract 4 types of visual features. After that we construct Fast Library for Approximate Nearest Neighbour (FLANN)[2] for these 4 types of feature vectors separately for subsequent retrieval.For query videos, firstly we adopted the same frame sampling schema, and then we did a pre-process to cope with several.transformations and produce a new version for each query video. We'll discuss this part intensively in section 2.2. Next we extract the visual features and conduct the rough detection process. The detail is in section 2.4.1. After that, for accurate detection, we used NCC (normalized cross correlation)[5] to filter some invalid frames which had been returned in the last step. At last, we post processed the frame level results and generate the final results. We'll explain each step of our approach below in details:

2.1 Interval Frame Sampling

If we extract every frames of the query and reference video, it'll cost too much computational

and processing expense. To handle this problem, we just extracted keyframes from both the query and reference videos at intervals of one second, and subsequent processes were just applied to these extracted keyframes. To those videos whose FPS was not integer, we rounded it up into integer. This not only greatly reduces the voluminous redundant data and computational time consuming, but also to some extent skips over the dropping frames provided by NIST.

2.2 Pre-Process for Query

One of the critical parts of query process is to detect and deal with the difference types of transformation. In this pre-process part, we did some anti-transform job. Since the visual features SURF [3] we used is claimed to be robust against different image transformations and quite resistant to image deterioration and coding artifacts, also the color correlogram feature is stable to gamma, blur etc as well, so we did not pre-process particularly the transformations which just simply modify the video, such as gamma, blur, noise, and re-encoding, but just dealt with those transformations automatically. To those transformations which really change the video content, we must process them specially. However it is still not realistic to cope with all the other transformations. In this paper, we mainly focused on black box and insertion of pattern detection.

2.2.1 Black Box Detection

The black box in the query video may derive from crop, shift and ratio. We did not differentiate crop and shift which do not change the length-width ratio, but just simply detected and removed the black box. For ratio, we not just removed the black box, but also provided a original non-scaled version as well.

For each frame of query video, we applied edge detection for black box detection and found the maximal rectangle in a single keyframe, and we assumed the largest one of all the rectangles we had found in each keyframe to be the video size we supposed to figure out. And then we computed the coordinate of the four vertexes of the rectangle, so that we could infer the widths of the four-side black boxes. As shown in figure 3, if the widths of the upper and under side were the same and the widths of the left and right side both closed to zero，we affirmed that the transformation is ratio，and we resized it to its original size. If not, we just directly remove the black box, it is shown in figure 2. It had been found through the experiment that our approach was effective for most of the videos.

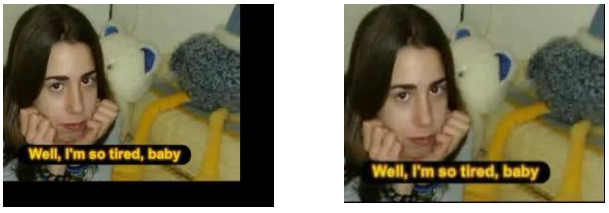

Figure 2:crop or shift process

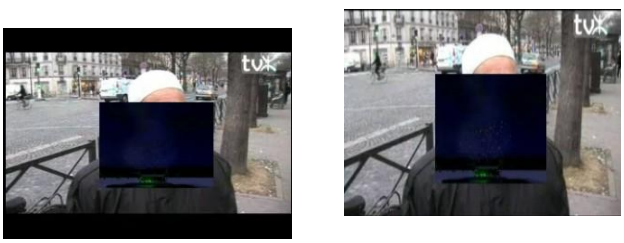

Figure 3:ratio process

2.2.2 Insertion of Pattern Detection

After removing the black box, we detected the insertion of pattern in the query videos, and compute a mask file for each query video. To deal with various types of insertion (text or still picture), we computes a mask file for each query video from the change of gray level of each pixel. As shown in the figure 4, in this file the areas which we found to be the artificially inserted content were masked so that we could ignore these areas when we extracted the visual features.

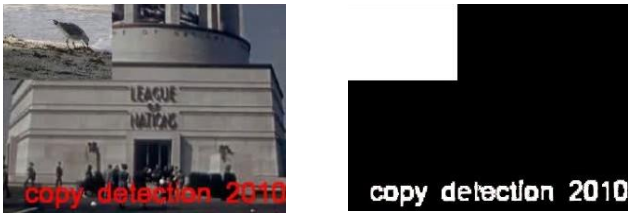

Figure 4: insertion of pattern process

2.3 Feature Extraction

This year, we choose SURF and color correlogram as our visual features, and we also presented the novel method to use NBS as a feature for our accurate detection.

2.3.1 NBS(Non-orthogonal Binary Subspace)

Feng Tang proposed a method to use Haar-like binary box functions to represent a single image or a set of images, as showed in figure 5. A desirable property of these box functions is that their inner product operation with an image can be computed very efficiently[5].

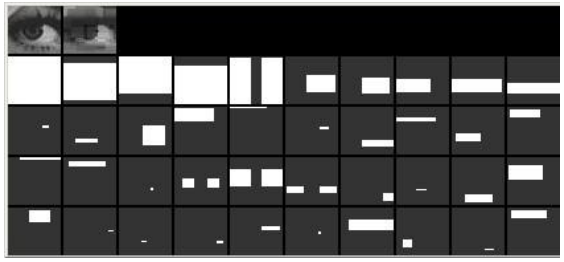

Figure 5: The first 30 non-orthogonal binary base vectors for an eye image

The problem of searching for the best subspace representation in a set of predefined non-orthogonal base vector dictionary is proved to be NP-hard. So we can use greedy solutions to find a suboptimal solution. In this system we employed the OOMP to search for the best subspace representation for the reference videos.

2.3.4 Feature Extraction schema

For reference videos, we extracted center color correlogram, center SURF feature, global color correlogram, global SURF feature and NBS feature.Since we did not deal with some transformations such as PIP, to reduce the influence of these transformations, we used a different feature extraction schema for query videos. For original query videos which had not been pre-processed, we extracted color correlogram and SURF feature of the center area of the keyframes.For pre-processed query

videos, ignoring the areas provided by the mask files we had computed during pre-process, we extracted global color correlogram, global SURF feature.

2.4 Detection Strategy

Our search process consists of two main parts, that is rough search and accurate search. During the rough search process, we aimed to using FLANN sketchily find out some similar reference frames to query as candidates for subsequent accurate search. Then during the accurate search process, we used NCC (normalized cross correlation) to filter the similar frames we just got in order to achieve a more accurate search result..

2.4.1 Rough Detection

Fast Library for Approximate Nearest Neighbour (FLANN) is a library for performing fast approximate nearest neighbour searches [2].In the rough search, firstly we constructed the FLANN of the feature vectors we had extracted from the reference videos for indexing and then conducted the rough frame detection process.

For the feature vectors extracted from the reference video, we establish the FLANN indexing separately for each kind of feature. For each query feature vector, we figure out the its distance from the feature vector of the reference videos, and save the most similar 30 reference feature vectors for future process. Simply merged the 30 most similar feature vectors which we had gained separately for the 4 different types of feature of one single query vector, we got a new merged result of 120 similar vectors for one query vector. If there was same vector, just ignored it and filled it with null.

2.4.2 Accurate Detection

In this paper, we proposed a novel method using Normalized cross correlation (NCC) to carry out our accurate detection. NCC is recently widely used in finding image two dimension patterns. Figure 6 is an example of application of NCC. And utilizing NCC we can make it possible to carry out our accurate search.

Firstly we used MSER[6] image segmentation approach to segment the reference keyframes into several patches. And then we extracted the NBS feature of these patches, next we used NCC to check that if there were same patches between the query keyframes and the reference keyframes we had gained in rough search process. If the number of the same patches was beyond a predefined threshold, we affirmed this reference keyframe to be an accurate result, otherwise we considered it to be a valid result and dropped it. Thus we can have a more accurate reference similar keyframe list for each query keyframe.

Figure 6: example of application of NCC

2.5 Post-process

In this part, we were inspired by the work which had been done by ATT in TRECVID 2009 CBCD task. After what we had done as discussed above, we got several similar reference keyframes to each query video. And then we could determine the list of best matching videos for it and generate the final result. In our approach we use three kinds of post process method, and these are the only difference among different runs.

In our first post process method, we used the similar method proposed by ATT[7], however what was different is that we used relevance NBS score instead of the relevance score that was used in their paper. Using this method we produced one runs: NJU.m.nofa.norank1. In the second post process method, we still adopted the method used by ATT to normalize the revelance NBS score and produced NJU.m.balanced.rank1. In the third post process method, we made some improvement. After doing what we had done in the first method, we used the result we had got and did NBS accurate detection process again to achieve a better result. Using this method we produced other two runs: NJU.m.nofa.comp2 and NJU.m.balanced.comp2. In each run for each query, we return no more than 8 similar reference videos.

3.RESULTS EVALUATION

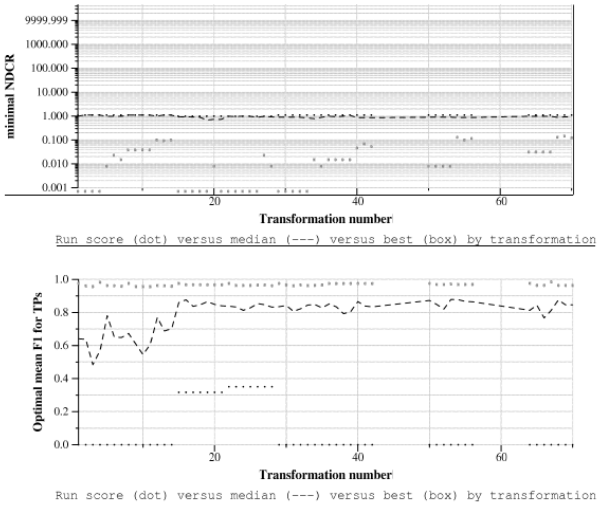

Figure 7: optimal NDCR and F1 for NJU.m.balanced.comp2

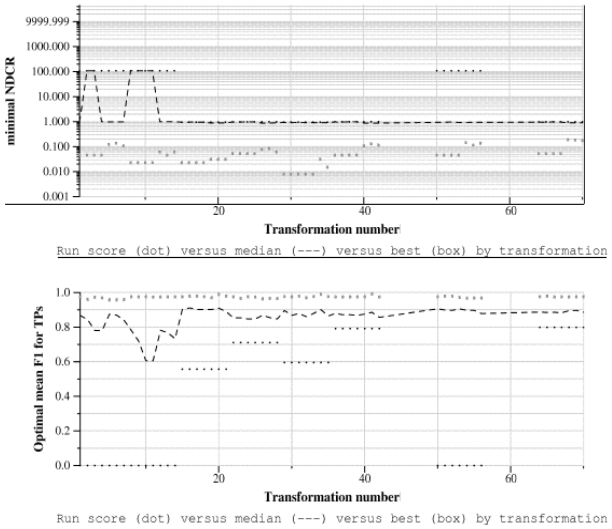

Figure 8: optimal NDCR and F1 for NJU.m.nofa.norank1

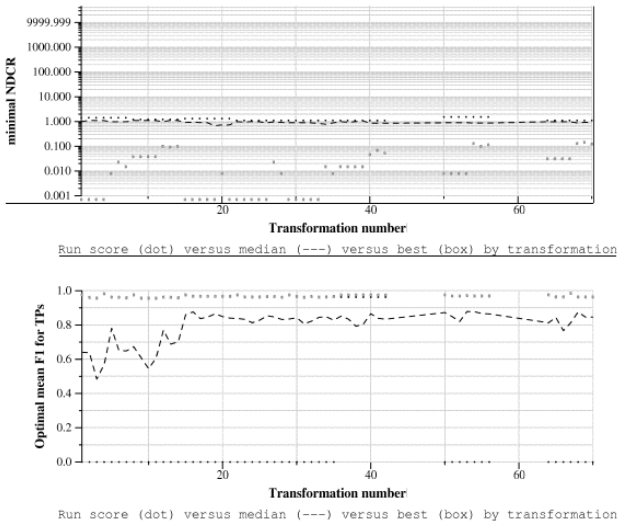

Figure 9:optimal NDCR and F1 for NJU.m.balanced.rank1

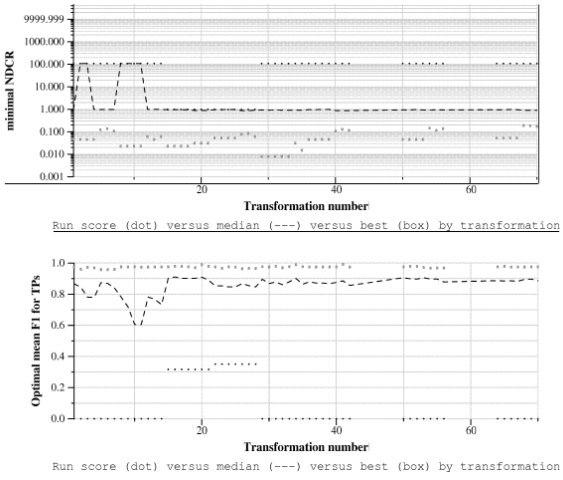

Figure 10: optimal NDCR and F1 for NJU.m.nofa.comp2

Figure 7-10 show the performance of our system in this task. Our system performance is not good this year. We carefully studied the results and found out two steps in our approach may lead to this situation. First, different number of feature vectors we returned in rough detection stage may have a great influence to the final results. Second, the threshold we set in NBS accurate detection stage and post process stage may not be appropriate. And most important, we put the reference video id wrong and lead to this fail result. Although we did not have good performance in this task, still some ideas we proposed in our approach were innovative and to some extent effective.

4.ONCLUSION AND FUTURE WORK

In this paper, we proposed a novel method to deal with the contend-based copy detection problem. In this method, we employed a combination of four different features as visual features for rough detection process, and then we use NBS to achieve a more accurate detection result. At last, we used three different post process methods and submitted 4 different runs in TRCVID 2010 CBCD task. The evaluation results show that our system achieve good performance in detection cost, but bad for detection accuracy. This is still a lot to improve in our system, and we'll do some further research in the future.

References:

[1] http://www-nlpir.nist.gov/projects/tv2010/tv2010.html#ccd

[2] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithmic configuration. In Proc. VISAPP, 2009.

[3] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," in Computer Vision - ECCV 2006, (A. Leonardis, H. Bischof, and A. Pinz, eds.), vol. 3951, pp. 404-417, of Lecture Notes in Computer Science, Springer, 2006.

[4] Jing Huang , Ramin Zabih. Color-spatial image indexing and applications. 1998

[5] F. Tang , R. Crabb and H. Tao. Representing images using nonorthogonal Haar-like bases. IEEE Trans. Pattern Anal. Mach. Intell., vol. 29, pp. 2120 2007.

[6] Michael Donoser , Horst Bischof, Efficient Maximally Stable Extremal Region (MSER) Tracking, Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, p.553-560, June 17-22, 2006

[7] Zhu Liu,Tao Liu,Behzad Shahraray. AT&T Research at TRECVID 2009 Content-based Copy Detection. TRECVID 2009.

Instance Search Task

Jinwei Xiao[1], Yang Yang[2], Kang Ling[1], Ying Lin[3], Pei Yang[3], Gangshan Wu[2],

NJU Multimedia Technology Laboratory

`[1]{jinwei.xiao, lingkang1988}@gmail.com,[2]gswu @nju.edu.cn

[3]{keller0618,yypp_918}@163.com,[4]yangy@graphics.nju.edu.cn

ABSTRACT

This paper provided an overview of Instance search task submitted in TRECVID 2010. NJU Multimedia Technology Laboratory participated in two tasks at TRECVID 2010: the content-based copy detection (CBCD) task and the instance search (IS) task. The Instance search task is a pilot task

and we adopted local visual features with region detector to represent video content, appending k-means algorithm to cluster video shots to reduce the huge quantity of local visual features. Besides, we employed FLANN index techniques to obtain the reliability and repeatability of our visual features.
Keyword instance search query instance test videos sift mser

1. Introduction

Instance search is a pilot task of TRECVID 2010. Instance search is essential for many applications, for example, discovering some special videos from mass of archive videos, organizing personal videos, protecting brand or logo from being illegal used, etc.

According to the IS task description from TRECVID 2010, the goal of IS task is find all the occurrence of the given object (a person or something else that is delimited in queries) in a given collection of test videos. The occurrence of delimited object in a query may have a great difference with the query delimited object. The occurrence may have different viewpoint, different pose, different color, different background, etc. For example, if the given query object is a person, then the occurrence of this person in the test videos will be with different cloth, different pose, different scene, different shape, sometimes only contains a part of the person. In TRECVID 2010, the queries will be provided with special format. The following is the query description of IS task in TRECVID 2010.
Each query will be likely consist of a set of

- 5 or so example frame images drawn at intervals from a video containing the item of interest. For each frame image:
  - the rectangular region within the frame image, containing the item of interest
  - a binary mask of an inner region of interest within the rectangle
  - the frame image with the inner region outlined in red
  - a list of vertices for the inner region
- the video from which the images were selected
- an indication of the target type taken from this set of strings {PERSON, CHARACTER, LOCATION , OBJECT}

In the proposed IS framework, we employed local visual features (Scale Invariant Feature Transform -SIFT) and region detector (Maximally Stable Extremal Regions) to match the content among query and test videos. FLANN (Fast Library for Approximate Nearest Neighbors) and Random Sample Consensus (RANSAC) techniques are supplied to maintain the scalability and increase the robustness of our approach. In total, we submitted three IS runs: one for processed queries, two for unprocessed queries with different parameters.

Though, our evaluation result is not good , we think our approach has something to recommend. More effort is required to improve the precision and scalibility of our apporach in future.

2. Overview

The following figure 1 shows the process of our approach. The processing consists of two parts displayed with different colors. The blue part shows the processing for test videos, while the gray one shows the processing for the query frame.

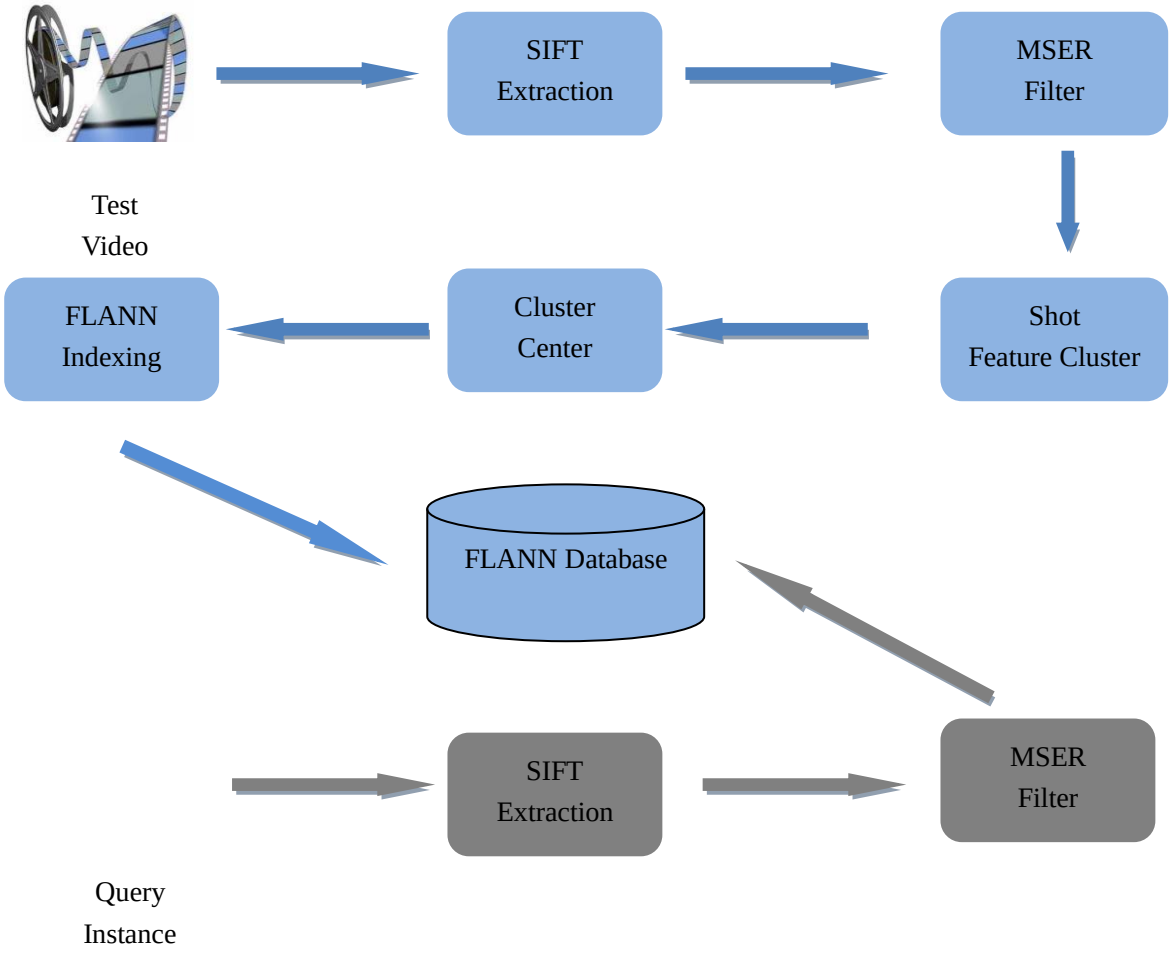

Figure 1. Overview of the proposed IS framework

The processing of Test Videos is as follows: We first extract SIFT features from Test Videos. Since extract each Test reference frame is computationally impossible and time consume, we only extract SIFT features at a rate of one reference frame per second in order to reduce the SIFT features. Besides we use MSER region detector to remove unstable SIFT features and this will improve the match precision of video content. Though we reduce the SIFT features by sampling, the magnitude of generated SIFT features are still staggering. Further data reduction is made by applying a cluster algorithm. We make an assumption that the video content of a shot has high similarity. So we cluster the SIFT features of a shot and represent the shot with the cluster centers. At the end, FLANN is adopted for efficient indexing and the indexing result is saved in the FLANN database.

For each query instance, we first apply SIFT algorithm to every occurrence of query instance in

queries data. Then we also employ MSER region detector to remove unstable SIFT features. Afterwards, we apply two different methods. For run one, we combine all the SIFT features of all the occurrence of one query instance to represent a query instance and query from the FLANN database. For run two and three, we cluster all the SIFT features of all the occurrence of one query instance and represent the query instance with the cluster center. Then retrieval will be made on the FLANN database.

3. Instance search processing system

3.1. SIFT Extraction

Scale Invariant Feature Transform (SIFT) is a local feature proposed by David Lowe at 2004 which has been proved to be efficient and robust in occluded object recognition and image matching. Besides, it can also achieve good performance under multi-view and rotation conditions. In the paper, we adopted SIFT feature as our visual features to match the video content. We use the C++ SIFT code implemented by Rob Hess according to the algorithm of David Lowe. The parameters of SIFT we used are same with the default parameters in Lowe's paper. Since SIFT is a distinctive local feature, it can get quantities of features. So we sample the Test Video and extract SIFT features at a rate of per frame one second.

3.2. MSER Filter

Since SIFT keypoints are maximum of the difference of Gaussian which will generate a lot of unstable keypoints. These unstable keypoints will make false and unstable results. Maximally Stable Extremal Regions (MSER) is proved to be a most stable region detector in many applications. We apply MSER to remove unstable SIFT features. Let $B = \{b_1, ..., b_n\}$ as the regions detected of a image. Let $S = \{s_1, ..., s_m\}$ Indicate the SIFT features set we extracted from the image. We define a SIFT feature is unstable as follows.

If a SIFT feature s_i doesn't belong to any region b_j of B, then we think s_i is unstable and remove it from the feature set S. We take the remaining SIFT features as our candidate features for further process.

3.3. Shot Feature Cluster

Though we extracted SIFT features per frame one second and employed MSER filter method to reduce the SIFT features, the quantity of SIFT features is still awesome. So we cluster the SIFT features using K-means algorithm. Before that we make an assumption that the frames in one shot have high similarity. At this assumption we cluster all the SIFT features in one shot and represent the

shot with the cluster centers. Empirically we set the cluster number k as 300 and describe the shot with 300 cluster centers. Normalization is made to avoid data inconsistency.

3.4. FLANN Indexing

For many computer vision problems, the most time consuming component consists of nearest neighbor matching in high-dimensional spaces. There are no known exact algorithms for solving these high-dimensional problems that are faster than linear search. Approximate algorithms are known to provide large speed ups with only minor loss in accuracy. In this paper we adopted FLANN as our indexing. FLANN is a library for performing fast approximate nearest neighbor searches in high dimensional spaces. We build FLANN indexing structures using 128-dimension cluster centers. The indexing structure will be saved with cluster center identifications (a string that is composed of the reference video ID, the shot ID) in an index file.

Querying the index file for a query SIFT feature (or 128-dimension cluster center) is very efficient since the index structure is a tree and the complexity is in the order of Log (N). For a query instance, the task is to find all reference frames with matching SIFT features and using the number of matching SIFT pairs as the relevance score for ranking purpose. To reduce the computational complexity for the following process, we only keep a few top reference SIFT features for one query SIFT feature in the match list. For run 2, we keep top 30 SIFT features for one query SIFT feature while for run 3, we keep 50.

Vote on a query frame will be made to get the reference shots that relevance to the query instance. Then the number of matching SIFT pairs combining with the threshold (Empirically value and in this paper it is set 5% of the number of SIFT features of query instance) will be the relevance score to rank the result. We only keep top 1000 reference shots for each query instance.

4. Result evaluation

TRECVID 2010 IS dataset contains 22 query instances and397 Test Videos (from TRECVID 2009 CBCD dataset sound & vision of 2009). The performance of our approach is not good. Average precision of most query instance is zero. But we glance over results of other Teams that all the other teams also perform badly. It shows this task is really a challenge.

We carefully study our approach and find some points that led up to the bad results. First the assumption about shot we made may be incorrect. Second cluster based on shot reduce the distinctive of each SIFT feature and enlarge the noise. Lastly, additional features should be added to advance the match. According to this further effort will be made to improve our performance on this task.

5. Conclusion

In this paper, we presented an Instance Search system. The system is based on SIFT features and relies on FLANN for scalability. MSER and K-means are used for data reduction and unstable feature removing. The evaluation results show that our system achieves good performance for query time cost but bad for query accuracy. More effort is needed to further improve the scalability of our approach.

6. References

[1] David G. Lowe, "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, 60, 2 (2004), pp. 91-110.

[2] Wu, Z. and Ke, QF and Isard, M. and Sun, J. "Bundling features for large-scale partial-duplicate web image search" CVPR, 2009.

[3] R. Hess. "An Open-Source SIFT Library". *ACM MM*, 2010.

[4] Foo, J.J. and Zobel, J. and Sinha, R. and Tahaghoghi, SMM. "Clustering near-duplicate images in large collections", 2007, Proceedings of the international workshop on Workshop on multimedia information retrieval.

[5] Marius Muja, David G. Lowe. "Fast approximate nearest neighbors with automatic algorithm configuration". International Conference on Computer Vision Theory, 2009.