THU-IMG at TRECVID 2010

Chen Sun, Jianmin Li, Bo Zhang, Qingtian Zhang
Intelligent Multimedia Group
State Key Laboratory of Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology (TNList)
Department of Computer Science and Technology, Tsinghua University, Beijing, P.R. China

ABSTRACT

Content-based Copy Detection

ID	Profile	Brief Description
dragon	BALANCED	Baseline: SURF + PiP Detection + VBH Indexing + Post Processing
tiger	NOFA	Baseline with different parameters
linnet	BALANCED	Baseline with different parameters and flipped query
tortoise	NOFA	Baseline with different parameters and flipped query

1. INTRODUCTION

Intelligent Multimedia Group in Department of Computer Science and Technology, Tsinghua University took part in TRECVID 2010 and submitted the results for Content-Based Copy Detection (CBCD) task. In this paper, the approach we adopted for this task is presented.

2. OVERVIEW

Our system is a frame based video-only copy detection system using the bag-of-feature model. Despite the parameter setting and component differences, each run can be separated into two parts, i.e. the offline part and online part. For the offline part, frames are sampled from reference video clips at a constant rate, and then a set of Speeded-Up Robust Features (SURF) [1], a kind of local feature were extracted for each frame. A two-level vocabulary is then trained with K-Means [2] at the first level and Vocabulary Based Hashing (VBH) [3] at the second level to quantize virtual words for feature descriptors. In the last step, we build an inverted index for each virtual word. For the online part, frames are sampled from query video clips at the same constant rate as the offline part, and then we pre-process the frames to detect Picture in Picture (PiP) regions, and extract features for the whole image as well as detected PiP regions. By querying the index built offline, reference frames with the same virtual words are retrieved, and additional geometric consistency as well as temporal consistency checking is applied to make the final copy segment estimation.

A block diagram for the whole system is illustrated in Fig. 1.

3. CBCD SYSTEM COMPONENTS

3.1 PREPROCESSING AND FEATURE EXTRACTION

Since the amount of data is quite large, it’s impossible to build the index for every frame from all reference video clips. For the simplicity of sampling and temporal consistency checking, the sample rate is fixed as **one frame per second** for both the reference data and query data. There are eight types of transformation for query video this year. To deal with these transformations, we add a preprocessing stage before extracting local features for queries.

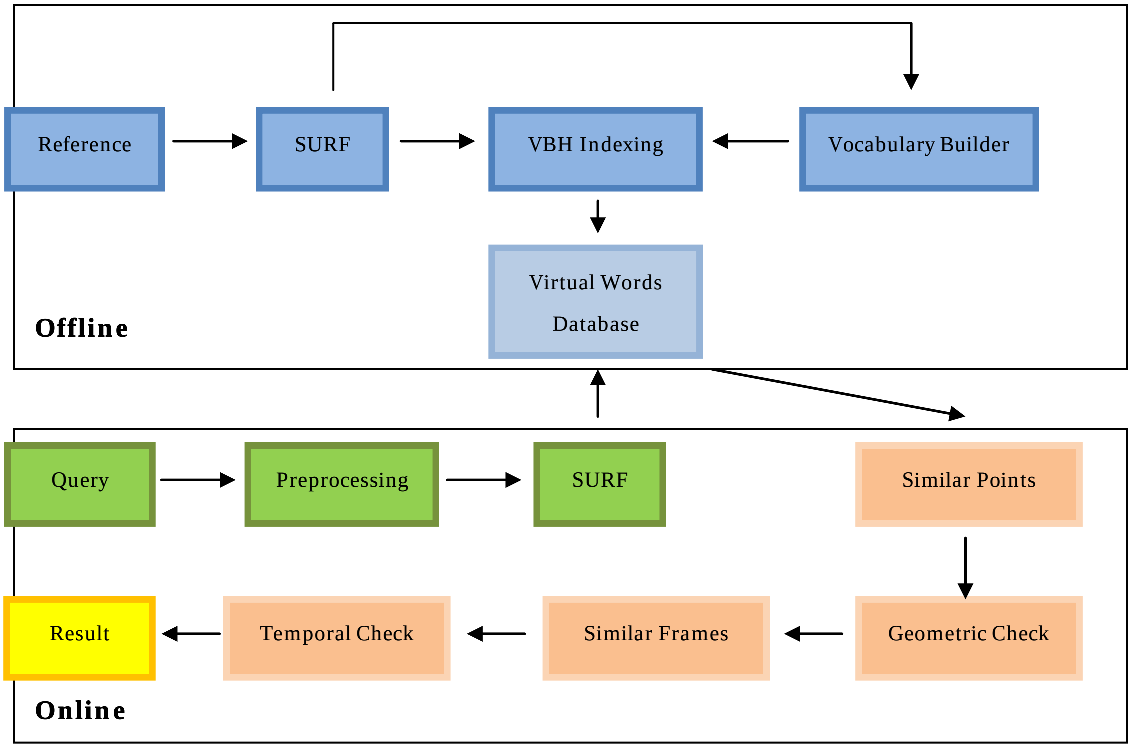


Fig. 1 System overview

Input: A query video V with n frames

Output: A PiP region represented by its upper left point and lower right point or NULL

S1. Apply Canny Edge Detection for each frame of V, and get $F = \{F_1, F_2, \dots, F_n\}$

S2. For every consecutive k frames F_i,..., F_{i+k-1}, calculate the average of them, drop the pixels smaller than θ and get smoothed image I_i

S3. For each I_i, perform Hough line transform, merge the shorter lines and remove lines that are not horizontal or vertical and add four virtual lines that represent the boundary of the image, get line set L_i

S4. For each L_i, search the edge groups that forms a rectangle, add the groups into set E

S5. For all edge groups in E, calculate its average position, drop the edge groups which are too far from the average, if at least N groups remains, return their average position, otherwise return NULL

Fig. 2 Picture in Picture region detection algorithm

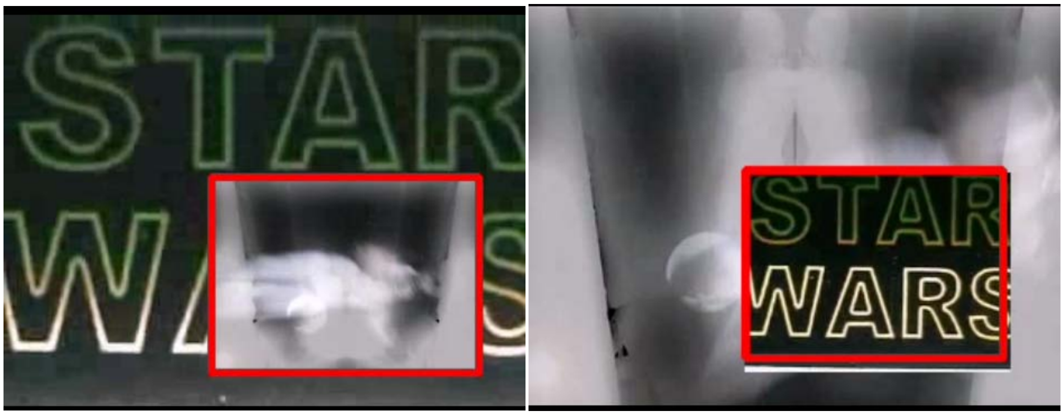


Fig. 3 Detected PiP regions (Marked by red boundaries)

Specifically, Picture in Picture Type 1 (T2) and Flip (T8) are processed in this stage, which can't be handled well by mere SURF features. For Picture in Picture Type 1, since the PiP region is fixed in a single query, we exploit the temporal information to boost our detection performance. Fig. 2 illustrates the algorithm in detail. Fig. 3 shows the performance on videos with PiP regions.

Since SURF is not mirror robust, for all the query video clips, we generate another flipped version. In summary, we have three types of queries: the original one, the PiP one and the flipped one.

After the preprocessing stage, SURF features are extracted for each sampled frame, a 64 dimension feature vector is generated for each interest point.

3.2 VOCABULARY TRAINING AND INDEX BUILDING

Once the features are extracted, Vocabulary-based hashing (VBH) index is built for reference video features. VBH combines the popular bag-of-word and Locality Sensitive Hashing (LSH) [4] approach. Each VBH vocabulary defines a hashing function, which maps the input feature points into the ID of nearest virtual word in it. Different vocabularies define different hashing functions, and form a hashing function family. The hashing function family is incorporated into the LSH approach results in the VBH index.

In practice, instead of applying VBH directly, we use a two-level vocabulary with K-Means at the first level and VBH at the second level. Given the reference feature points, we use the algorithm described in Fig. 4 to generate the two-level vocabulary.

Input: Reference local feature vector set *D*

Output: A two level vocabulary *V1* and *V2*

S1. Randomly select *m* feature vectors from *D*, apply the K-Means algorithm to generate *k* clusters, the centers of these clusters give the first level vocabulary *V1* with *k* entries

S2. For each virtual word in *V1*, randomly select n feature vectors from *D*, find the vectors belong to it, using these vectors to train the VBH hash functions, which yields the second level vocabulary *V2*

Fig. 4 Algorithm to generate the vocabulary

In our system, the size of the first level vocabulary is 20000, and the size of the second level vocabulary is 256, which means the number of hash functions is 8. Four second level vocabularies are used together to achieve higher performance.

Based on the vocabulary, an inverted index is built by attaching each virtual word the label of the feature point, which is identified by the triple: video id, frame id and point id. To facilitate the geometric consistency checking, the x coordinate and y coordinate are also stored in the label.

Since we found that image scale affects the SURF feature extraction, we build an additional index for the reference data after half scaling the frames.

3.3 QUERY

Given a set of feature vectors from a query frame, we find the approximate nearest neighbors in the database by traversing the two level vocabularies and find its own virtual words, by the inverted index, the nearest reference points can be retrieved. The retrieved points are passed to the geometric and temporal consistency checking unit to generate the final results.

3.4 DETERMINATION

After the query step, for each feature point of each query frame, a set of reference labels are retrieved from the database, and we can apply a simple voting mechanism by calculating the number of labels for each frame. However, since the bag-of-word model drops the geometric information, it’s found that there are irrelevant frames with high voting scores. To deal with this problem, with the extra geometric information in each label, we apply geometric consistency checking with a RANSAC [5] like algorithm, which is illustrated in Fig. 5.

```
Input: Matched coordinate pairs $C$ between two frames
Output: Decision score for the reference frame
S1. Randomly select 3 pairs from $C$, calculate the affine transformation matrix $M$
S2. For each pair $(a, b)$ in $C$, apply $M$ to $a$ and get $a'$, calculate the number of pairs which $dist(a'-b)$ is less than $\theta$, denoted as $N$
S3. If $N$ is greater than some certain threshold $\varphi$, return $N$ as the decision score, if the algorithm has already run for more than $L$ iterations, return 0, else go back to S1
```

Fig. 5 Algorithm to check geometric consistency

In practice the threshold φ is set as a percentage of the number of feature points of the query frame.

Once we get the decision score for each retrieved frame, an adjusted version of the time sequence consistency method [6] is adopted to locate the matched segments between the query and the reference video. Like the original method, our version finds the time consistent similar frame path with maximum weight. However, in the original method, the weight of the path is defined as the sum of all the weights of the similar frames in the consistent path, which benefits the paths with many low-weighted frames. In our version, the weight is defined as the sum of the logarithms of all the weights of the similar frames in the consistent path, which favors the paths with a session of significant-weighted frames.

After this stage, the final decision store for each retrieved reference clip is generated.

4. EXPERIMENTS

We submit 4 runs for the task, all with video information only.

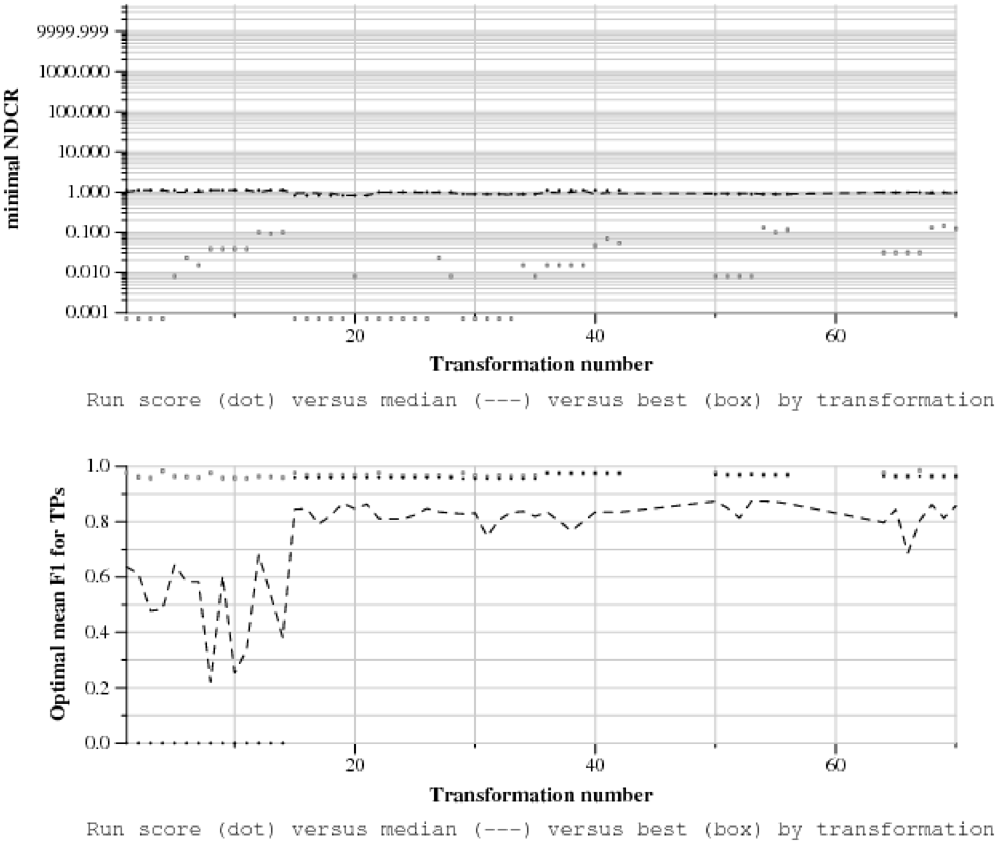
The *dragon* run belonging to the BALANCED profile uses 4 VBH vocabularies for the normal query and the PiP query, the geometric consistency threshold φ is 0.15, 8 normal query results and 3 PiP query results for each query are remained in the final result list.

The *tiger* run belonging to the NOFA profile uses 4 VBH vocabularies for the normal query and the PiP query, the geometric consistency threshold φ is 0.2, 5 normal query results and 3 PiP query results for each query are remained.

The *linnet* and *tortoise* run adds 2 flipped query results for each query on the basis of the *dragon* and *tiger* run, respectively.

Due to time limitation, we only adjust parameters based on a small set of query videos generated on the reference video dataset by ourselves.

The optimal result for the *dragon* run can be seen in Fig. 6



Run score (dot) versus median (---) versus best (box) by transformation

Run score (dot) versus median (---) versus best (box) by transformation

Fig. 6 Optimal result for *dragon* run

The result shows that our system has an outstanding performance on location accuracy (F1) on most of the transforms, showing that our determination stage is effective. However, our system has difficulty dealing with camcording transformations, which means SURF may not be robust under this transformation and it should be taken special care in the pre-processing stage.

Moreover, we also find that our PiP detection stage has improved NDCR for the video T2 significantly, which means the PiP detection algorithm works well in our framework and mere SURF alone is not robust under this transformation.

ACKNOWLEDGEMENT

This work is supported by the National Key Foundation R&D Projects under the grant No.2007CB311003.

REFERENCE

[1] Herbet Bay et al. Speeded-Up Robust Features. ECCV 2006

[2] www.eecs.northwestern.edu/~wkliao/Kmeans/index.html

[3] Yingyu Liang, Jianmin Li, Bo Zhang. Vocabulary-based Hashing for Image Search. In Proceedings of the seventeen ACM international conference on Multimedia

[4] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In SCG, 2004

[5] Martin A. Fischler and Robert C. Bolles (June 1981). "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". Comm. Of the ACM 24: 381–395

[6] Yongdong Zhang, Ke Gao, etc. TRECVID 2008 Content-Based Copy Detection By MCG-ICT-CAS. In Proceedings of TRECVID 2008 workshop

[7] Yingyu Liang, Binbin Cao et al. THU-IMG at TRECVID 2009. In Proceedings of TRECVID 2009 Workshop

[8] Zhu Liu, Tao Liu, Behzad Shahraray. AT&T Research at TRECVID 2009 Content-based Copy Detection. In Proceedings of TRECVID 2009 Workshop