A BoF Model Based CBCD System Using Hierarchical Indexing and Feature Similarity Constraints

Nan Nan
The School of Electronic and Information Engineering
Xi'an Jiaotong University
Xi'an, China
nannan@mail.xjtu.edu.cn

Guizhong Liu
The School of Electronic and Information Engineering
Xi'an Jiaotong University
Xi'an, China
liugz@xjtu.edu.cn

Chen Wang
The School of Electronic and Information Engineering
Xi'an Jiaotong University
Xi'an, China
wangchen615@gmail.com

ABSTRACT

Recently, local interest points (also known as key points) are shown to be useful for content based video copy detection. The state-of-art local feature based methods usually build on the bag-of-visual-words model and utilize the inverted index to accelerate search process. In this paper, we offer a detailed description of a novel CBCD system. Compared with the existing local feature based approaches, there are two major differences. First, besides the descriptors, the dominant orientations of local features are also quantized to build the hierarchical inverted index. Second, feature similarity constraints are used to refine the matching of visual words. Experiments performed on a reference video dataset of 50 hours show that our system can deal with 9 types of common video transformations, and due to the hierarchical indexing and feature similarity constraints, the computational costs are reduced as well.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*information filtering, search process*; I.4.9 [**Computing Methodologies**]: Image Processing and Computer Vision—*Applications*

General Terms

Algorithms

Keywords

Video copy detection, hierarchical indexing, feature similarity constraints

1. INTRODUCTION

Nowadays, the copy infringement of digital videos is becoming more serious, which makes the study on the techniques for maintaining intellectual property receive increasing attention. Generally, there are two genres of the techniques to detect video copies: digital watermarking and content based copy detection (CBCD). Digital watermarking needs to embed a digital signature(watermark) in the original video, therefore it is inconvenient to use. Meanwhile, it is vulnerable facing content editing, such as color adjustments, logo / text insertions, and format changes (The TREC Video Retrieval Evaluation (TRECVID) [14] lists several common video transformations [1]). Unlike digital watermarking, Content based copy detection relies only on a similarity comparison of content between the original video and its various possible copies. As long as the video signatures are generated properly, these methods can be robust to many video attacks. Consequently, the techniques used in CBCD are now more prevalent.

In recent years, a number of content based copy detection approaches have been proposed. Based on the features used for generating video signatures, they can be categorized into two types: global features and local features based methods.

Global features describe the frames by using statistical and distribution information of the color, gray scale, edge, texture, etc. In general, they are easy to compute, compact in storage, but vulnerable to many video attacks [16]. The most popular ones are the ordinal measure [2] and color histogram [6]. A recent trend is to combine global features with sequence matching [16]. But due to the inherent deficiency of global features, these methods are still not robust enough for video attacks like crop, letterbox, PIP, etc.

Local features, on the contrary, are more robust to video attacks, but they are of high computational complexity and with enormous quantity of feature data. Nowadays, using the bag-of-features (BoF, also known as bag-of-visual-words) model becomes more popular. The work [13] first introduced this model to deal with searching in a large corpus of images. After that, many methods apply such model to image copy detection, image objects retrieval and video copy detection [10, 11, 15, 5, 12, 4, 9, 3, 7]. Some of them use alternative clustering methods such as the hierarchical k-means [10], approximate k-means [11], or a regular lattice [15], to avoid the high time complexity when the volume of vocabulary is huge. Some others increase the discriminative power of the visual words by soft assignment for descriptors [5, 12] or by adding additional signatures to descriptors [4, 9]. Some works utilize the feature layout information to re-rank the search results [12, 4, 3]. All these methods just quantize the descriptors to build the inverted index, so the information of the features is actually not fully utilized. Furthermore,

ICIMCS '10 December 30-31, 2010, Harbin, China

unlike object retrieval, in CBCD, a video frame is commonly copied as a unit (or at least the most part of it), so the layout information of features should be specially considered under the circumstances.

In this paper, we propose a CBCD system complementary to those approaches mentioned above. In the quantization part, the orientations of the features are also quantized to build the first level of our hierarchical inverted index, by which the time complexity of clustering the descriptors and computing the visual words is reduced. In the search part, we integrate several novel feature similarity constraints within the inverted file to refine the matching and speed up the search.

The paper is organized as follows. A flowchart of the proposed system is given in Section 2. Section 3 offers a brief description of our strategy for key frames selection and SIFT feature extraction. Our primary contribution, a novel search engine (including the hierarchical indexing and feature similarity constraints) is described in Section 4. Section 5 presents the experimental results and Section 6 concludes the paper.

2. FLOWCHART OF OUR SYSTEM

As shown in Figure 1, the proposed system consists of two parts, namely, the offline module and the online module. In the offline module, we first segment the reference videos and select key frames from those video segments. SIFT features are then extracted from the key frames and vocabularies are generated by clustering a training dataset of the SIFT descriptors. All the SIFT features are then quantized according to the vocabularies and stored in a inverted table. In the online module, key frames of query videos are obtained by uniform frame sampling. The feature extraction and quantization steps are then carried out, which are basically identical to those in the offline module. The search engine returns result lists by a voting strategy.

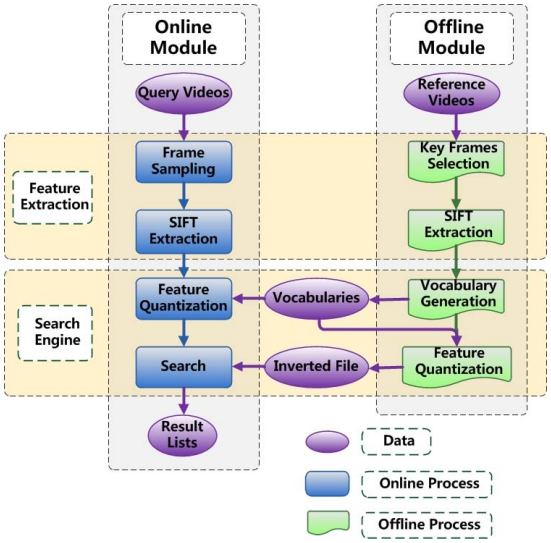


Figure 1: The flowchart of the proposed system.

3. FEATURE EXTRACTION

In this section, we describe the feature extraction in our system, including the key frames selection and SIFT extraction.

In common video retrieval systems, key frames selection is a necessary pre-processing step. For reference videos, the purpose of key frames selection is to select the most representative frames while maximally drop the redundancy in videos. In our system, we first employ a shot boundary detection method based on blocked based orientation gradient histogram and color histogram to segment a video into nearly still or gradually changing scenes (segments), then by comparing frame by frame luminance differences and spatial luminance variances in each video segment, we select the most stationary frame, which is neither blurred by fast motion nor too dark in luminance. For query videos, the purpose of key frames selection here is to speed up the searching process, so we simply use uniform sampling to extract key frames from a query video.

For the key frames from both the query videos and the reference videos, we utilize the DoG blob detector [8] to detect interesting regions in a single frame and extract a 128-d SIFT feature [8] from each region.

4. SEARCH ENGINE

In this section, we give a description of our hierarchical indexing method and the voting strategy using feature similarity constraints.

4.1 Hierarchical Indexing

The term 'hierarchical indexing' used in [10] refers to the vocabulary tree built by hierarchical k-means, which is different from our work. In that work, an initial k-means was first run on a training data set of feature descriptors, defining k cluster centers. By assigning each descriptor to its closest center, the training data was then partitioned into k groups. The same process was recursively applied to each group to determine the tree level by level. In the online phase, each descriptor was propagated down the tree by means of comparing the descriptor to the cluster centers at each level and then choosing the closest one. This hierarchical indexing structure has multiple levels, but the defining of k children for each level is similar to that in the non-hierarchical methods[13, 12]. Moreover, the entities to build the indexing are only the descriptors of the features.

In the extraction of local features, such as SIFT, the orientation of a key point is defined as the orientation corresponding to the highest peak of the orientation histogram formed for a neighboring region around the key point [8]. The orientation assignment is a key step for the key point to achieve invariance to image rotation. However, under the circumstance of video copy transformations, the frame rotation rarely occurs. Other video copy transformations either have limited influence on the orientations (such as crop, noise, blur, shift, etc.), or change them in a foreseeable way (such as flip.). In one word, the key points in video frames usually have orientation constancy.

Based on the above analysis, an improvement on the BoF model is taken for the copy detection application by considering dominant orientations of the key points. In the clustering stage, for a large set of the training data, this approach uses two steps to generate the vocabularies. Firstly, the numerical range of the orientations of the key points, namely the interval $[0, 2\pi]$ is uniformly divided into n in-

tervals. According to the division, the orientations of the key points are quantized and the data is partitioned into n groups, each of which consists of the key points with orientations in the same interval. Secondly, the vocabulary for each of the groups is generated by quantizing the descriptors of the key points inside the group. The choice of the quantization method can be various (k-means, hierarchical k-means, or the other methods.), depending on the amount of the data in the group. Since it is reduced dramatically as a consequence of the first step, here we use the k-means method for all the groups.

To compute the visual word for a key point, its orientation is first quantized to an integer between 1 and n , and the visual word closest to the descriptor is chosen in the vocabulary which is determined by the integer. Once the quantization is finished, the feature data can be indexed in a hierarchical way as Figure 2 shows.

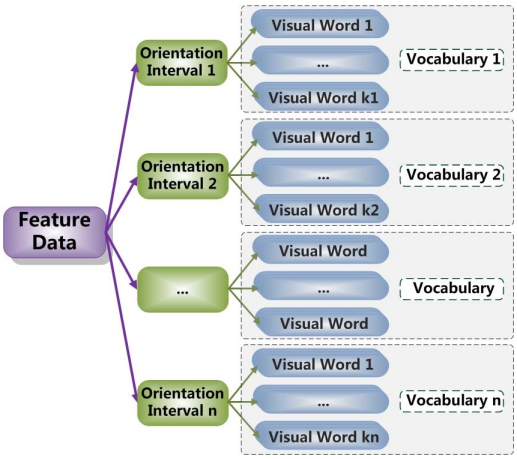


Figure 2: The hierarchical indexing of feature data.

4.2 Feature Similarity Constraints

The traditional BoF model combines the robustness of the local features and the efficiency of the inverted file to achieve a fast approximation to the one-by-one image / frame comparison. However, it does not fully utilize the information the local features provide for two reasons.

First, the discriminative power of the feature descriptors is reduced. Any two descriptors in the same cluster have no difference between each other. When using a small vocabulary for a large set of features, there will be a lot of noise descriptors inside each cluster (Voronoi cell). Second, the spatial information of the features is not utilized at all in the visual word matching. This strategy may work fine in image object retrieval but is not accurate and efficient enough for video copy detection. Video copy transformations usually have small influence on the position of a key point in a frame (except for PIP). Therefore, the spatial information of the features is quite useful in the circumstances.

Figure 3 demonstrates these two situations. In Figure 3(a), we can see that the descriptors of the key points A and B lie in the same Voronoi cell, but considering the difference between them, the probability of mismatching is high. In Figure 3(c), although the descriptors of the key points A and C lie close in the feature space (as shown in Figure

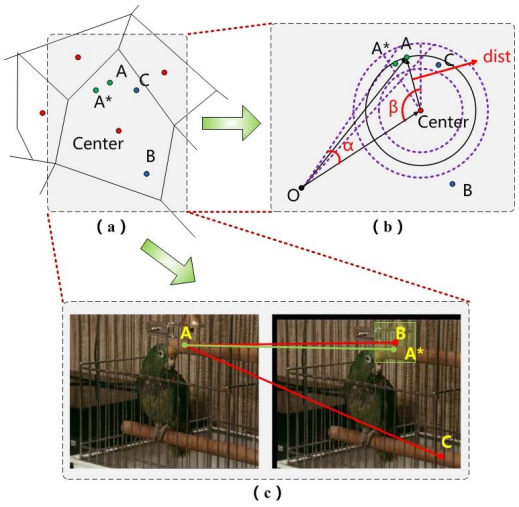


Figure 3: Feature similarity constraints (a) The descriptors of several features in a Voronoi cell. (b) The visualization of the deviation representation in a 2-dimensional feature space. (c) Spatial position constraints for features in video frames.

3(a)), if the spatial information is considered, they are still mismatched.

In this section, we present an approach to refine the matching of the visual words as well as to speed up the search process by using additional information of the features, which are the locations of the feature descriptors in the feature space and the spatial positions of feature points in video frames.

Since each descriptor is a 128-d vector, it could be deemed as a point in the high-dimensional space, whose coordinates (location) are represented by the 128 values of the vector. After being assigned to a visual word, a descriptor lies in a specific Voronoi cell and its matching descriptors should be the closest ones among all the descriptors in that Voronoi cell. However, finding them by direct comparisons of the descriptors is impractical, because this process not only needs all the descriptors to be stored in the inverted file, but also adds extra computational effort required for computing the distances between the descriptors.

As formula (1) shows, a descriptor can be completely specified by the center of its attributive Voronoi cell and its deviation from the center.

$$desVec = cenVec - devVec \tag{1}$$

where $desVec$, $cenVec$ and $devVec$ are the descriptor, the center and the deviation respectively. When two descriptors are assigned to the same cluster center, their difference can be represented by the difference of their deviations from the center. So matching two descriptors in a Voronoi cell can be converted into matching the deviations of the two descriptors. As the deviation of a descriptor is a 128-d vector, which is still inconvenient to store and compare, we here use 3 quantities to describe it: $dist$, α and β. Here $dist$ is the magnitude of the deviation. α is the angle between the descriptor and the center, while β is the angle between the deviation and the center. In Euclidean geometry, the angle

Table 1: The contents for a feature record in the inverted file

Orientation Interval ID	
Visual Word ID	
1	Video Name ID
2	Frame Number ID
3	$dist$
4	α
5	β
6	n_x
7	n_y

θ between two vectors a and b is defined by formula (2):

$$\theta = \arccos(\frac{a \cdot b}{|a||b|}) \quad (2)$$

where $\cdot$ denotes the dot product and $|\ |$ denotes the length of a vector.

From a statistical perspective, the angle between two vectors reflects the correlation of them, so when a specific center is given, these three quantities describe the descriptor and the its deviation from the center in an approximate way (The visualization of this representation for a 2-dimensional feature space is shown in Figure 3(b)).

For the spatial position, we use the normalized coordinates of the feature, which is defined by formula (3).

$$\begin{cases} n\_x = x/W \\ n\_y = y/H \end{cases} \quad (3)$$

where W and H are the width and the height of the frame in which the feature is detected. The reason of normalizing the coordinates is to avoid the influence caused by the variations of the frame size.

The five quantities, $dist$, α, β, n_x, n_y, describe the location of a feature in both the feature space and the pixel domain, so they can be used to measure the similarity of two features in video copy detection. In our system, they are integrated into our inverted file system (as Table 1 shows)and used as search constraints. In feature space, the matching candidates are selected from a small space determined by $dist$, α and β rather than the whole Voronoi cell (as Figure 3(b) shows); in pixel domain, the search range is limited to a square centered at (n_x, n_y) rather than the whole frame (as Figure 3(c) shows).

4.3 Voting Strategy

In our system, since videos are represented by key frames and these key frames are represented by sets of features, copy detection based on the BoF model can be interpreted accordingly as a two-step voting strategy.

For a query video, the first step is to obtain a set of matching frame pairs between the query video and the reference videos. Suppose that a frame of a query video is represented by M local features $qf_m (1 \leq m \leq M)$, and all the reference videos have a total of N frames, each of which is represented by local features $rf_{n,k} (1 \leq n \leq N$ and $1 \leq k \leq K$, where K is the number of features in the corresponding reference frame). Then this step can be described in the following substeps.

1) The scores s_n of all the frames from the reference videos are set to 0.

2) For each qf_m, find a set of $rf_{n,k}$ in the inverted file, which satisfies the following two conditions:

a. Both the quantized orientations and the quantized descriptor indices (visual words) of qf_m and $rf_{n,k}$ are identical.

b. The constraints of feature similarity between qf_m and $rf_{n,k}$ are satisfied, as formula (4) shows.

$$|quantity_{qf_m} - quantity_{rf_{n,k}}| \leq quantity\_threshold \quad (4)$$

Here $quantity$ represents the five quantities defined in Section 4.2, i.e. $dist$, α, β, n_x and n_y, and $quantity\_threshold$ represents the corresponding threshold value. These threshold values delimit the search ranges of $rf_{n,k}$ in both the feature space (as Figure 3(b) shows) and the pixel domain (as Figure 3(c) shows).

3) For each $rf_{n,k}$ found by the previous process, update the score s_n of the corresponding frame by

$$s_n := s_n + \frac{1}{\sqrt{MK}} \quad (5)$$

4) After all the qf_m are searched, rank the scores of all the frames of the reference videos and choose the frame with the highest s_n as the most matching one.

In the second step, The scores of the frames from the same reference videos are accumulated and ranked in descending order to generate the list of the matching reference video segments.

5. EXPERIMENTAL EVALUATIONS

5.1 Dataset

Here we present the different datasets used in our experiments.

Dataset1 (Reference videos: 109, key frames: 41,557, features: 33M). The reference dataset contains all the videos of tv.2007.sv.test in TRECVID [14]. There are 109 videos with the size of total 29.2G and lasts more than 50 hours. We select 41,557 key frames (using the method introduced in section 3)from them and extract more than 33M SIFT features from the key frames.

Query Dataset (Query videos: 90) 10 video segments selected from **Dataset1** are used to generate the query videos. Each query video is generated by taking one from the 10 segments and then optionally applying one transformation using random parameters to the entire segment. In our experiments, 9 transformations (including insertions of logo/text, reencoding, change of gamma, blur, contrast, noise, crop, shift, letterbox) are used, therefore a total of 90 query videos (lasting 1 hour) are generated. The descriptions of the transformations, the parameters and the limits of their values can be found in [1]. For the query videos, the uniform sampling ratio is 1 frames every 4 seconds.

Dataset2 (Key frames: 180, features: 100k). 90 pairs of original frames and their transformed versions (10 pairs for each transformation mentioned above) are selected from **Dataset1** and **Query Dataset**. We extract about 100k SIFT features from them.

All the reference and query videos are in MPEG1 format, 352x288, 25 fps and our experiments are carried out on a 2.33 GHz two-core computer with 2 GB memory.

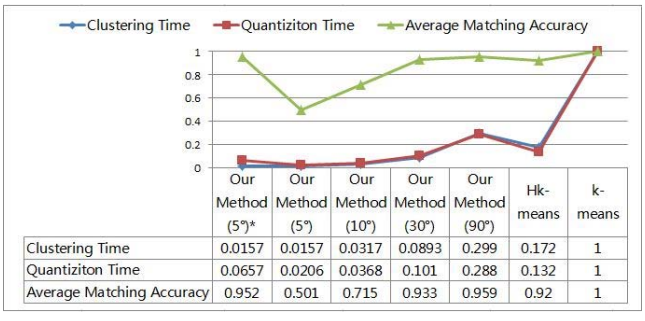
5.2 Hierarchical Indexing

The goal here is to test our hierarchical indexing method and we evaluate it by using three quantities: the clustering time, the quantization time and the average matching accuracy. The clustering time is the time for generating visual vocabularies from the training data, and the quantization time is the time for quantizing all the reference video data for the purpose of building the index. Together they reflect the computational cost of an indexing method. The matching accuracy is defined as follows: suppose that two frames has n pairs of correctly matching descriptors, and after feature quantization, m pairs among them have identical visual words, then the matching accuracy is computed by m/n.

Dataset2 is used to carry out this evaluation. The step sizes for quantizing the orientations of the features are 5°, 10°, 30°, 90°, and hence produce 72, 36, 12, and 4 orientation intervals respectively. Our implementations of other indexing methods using k-means [13] and hierarchical k-means [10] are used for comparison. All the methods use 500 visual words. In the case of the hierarchical k-means, a two-level indexing structure is constructed (10*50).

Figure 4 shows the results of the computational cost and the average matching accuracy respectively. All the results are normalized by their counterparts of the method using k-means for comparison. It can be seen that the method using k-means has the highest matching accuracy, but its computational cost is much higher than the rest of the methods, which will become unacceptable as the amount of testing data increases. The method using hierarchical k-means reduces the high time complexity of k-means at the sacrifice of average matching accuracy (drop by 8%). For our hierarchical indexing method, although the decrease of the step size (from 90° to 5°) causes a reduction in matching accuracy by 48%, it dramatically reduces the clustering time (more than 19 times less) and the quantization time (more than 13 times less), which would be very helpful when facing a large amount of data.

In order to compensate for the loss of matching accuracy in the case of 5°, we adjusted our quantization strategy by assigning the orientation of each descriptor to 3 intervals, that is, the original interval and its two neighboring ones (Our method (5°)*). In this way, a high matching accuracy is obtained along with the lowest clustering time and a medium quantization time.



	Our Method (5°)*	Our Method (5°)	Our Method (10°)	Our Method (30°)	Our Method (90°)	Hk-means	k-means
Clustering Time	0.0157	0.0157	0.0317	0.0893	0.299	0.172	1
Quantiziton Time	0.0657	0.0206	0.0368	0.101	0.288	0.132	1
Average Matching Accuracy	0.952	0.501	0.715	0.933	0.959	0.92	1

Figure 4: The comparison of computational cost and average matching accuracy for different indexing methods.

Table 2: The thresholds for feature similarity constraints

/	Average Difference	Threshold
$dist$	13.68	15
α	1.76°	2°
β	1.67°	2°
n_x	/	$16/W$
n_y	/	$16/H$

5.3 Feature similarity constraints

The thresholds of feature similarity constraints are determined based on statistical studies and priori knowledge. For $dist$, α and β, we have examined over $50k$ pairs of matching descriptors, computed the differences of the values and studied the distribution characters of them. It turns out that these differences are densely distributed around their average values. Therefore, we set the 3 thresholds to 3 values slightly larger than the average differences respectively and guarantee that over 95% of matching descriptor pairs satisfy the constraints. For n_x and n_y, the priori knowledge of video transformations is considered and a search region of 16*16 in pixel domain is used to confront the influences of key point position changes caused by shift and letterbox. All the thresholds are listed in Table 2.

5.4 Comparison with other methods

We now evaluate the performance of our system and compare it with other methods. **Dataset1** and **Query Dataset** are used to carry out this evaluation. The methods we test are as follows: (1) SIFT + hierarchical indexing + feature Similarity constraints; (2) SIFT + hierarchical indexing; (3) SIFT + k-means; (4) SIFT + hierarchical k-means. The hierarchical indexing here refers to our method introduced in Section 4.1 and the step size for quantizing the orientations of the features is 5°. Each method has a vocabulary of 10,000 (In the case of the hierarchical k-means, a two-level indexing structure is constructed (100*100).) and employs the same voting strategy (Section 4.3).

Figure 5 compares the results returned by the different methods. Since the SIFT descriptor is robust to various video copy transformations, it is not surprising that all methods perform well in most situations. However, comparing the results of method (1) and method (2), we can clearly see that using feature similarity constraints indeed improves the performance of the search engine. The performance of method (2) and method (3) are almost the same and superior to that of method (4), which is another proof of the effectiveness and efficiency of our indexing method. It is worth noticing that method (1) fails to detect several video copies of the shift transformation. The reason is that because it integrates feature similarity constraints in its search process, which delimit the search range by using feature locations in the pixel domain. When the shift transformation has huge influence on the frames, our method may not work well.

Figure 6 shows the average search time (in seconds) for one frame in the 41,557 key frames of the reference dataset. It can be seen that, the time of method (1) is much shorter than the others, which indicates that by using feature similarity constraints, the computational cost of the search process is largely reduced. Meanwhile, among those methods which do not use feature similarity constraints, method (2)

has the shortest search time, indicating that our hierarchical indexing is more efficient than the others.

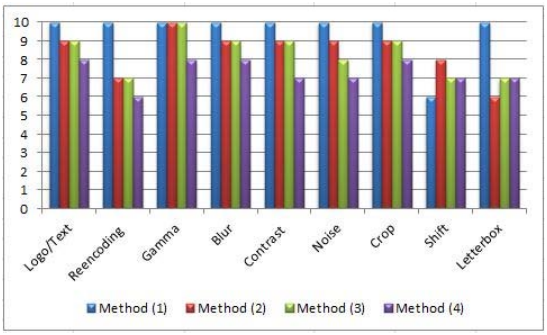


Figure 5: The comparison of search results of different methods.

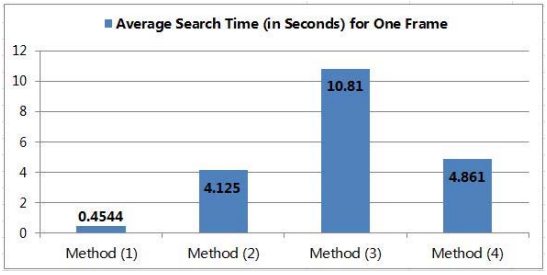


Figure 6: The average searching time (in seconds) for one frame

6. CONLUSION

We have introduced a novel CBCD system, which improves the standard BoF models in two ways. First, a new hierarchical indexing method has been proposed, which combines both the orientations and the descriptors of local features. It can help to reduce the clustering time for generating vocabularies, and at the same time maintain competitive performance. Second, feature similarity constraints are used to refine the matching of visual words and speed up the search process. Experiments show that our system can deal with 9 types of common video copy transformations, and due to the hierarchical indexing and feature similarity constraints, the computational costs are reduced substantially as well. In future works, the audio information could be considered to facilitate the video copy detection.

7. ACKNOWLEDGEMENTS

This work is supported in part by National High Tech. Project No.2009AA01Z409 and National 973 Project No.2007 CB311002.

8. REFERENCES

[1] Final list of transformation. *http://www-nlpir.nist.gov/projects/tv2008/final.cbcd.video.transformations.pdf.*

[2] D. Bhat and S. Nayar. Ordinal measures for image correspondence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(4):415, 1998.

[3] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *Proc. ICCV*, volume 2, 2007.

[4] H. Jegou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. *Computer Vision–ECCV 2008*, pages 304–317.

[5] H. Jegou, H. Harzallah, and C. Schmid. A contextual dissimilarity measure for accurate and efficient image search. In *IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR'07*, pages 1–8, 2007.

[6] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, 2006.

[7] Z. Li, G. Liu, H. Jiang, and X. Qian. Image copy detection using a robust gabor texture descriptor. In *Proceedings of the First ACM workshop on Large-scale multimedia retrieval and mining*, pages 65–72, 2009.

[8] D. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[9] M. Mei, Z. Zhao, A. Cai, and X. Xie. Rapid search scheme for video copy detection in large databases. In *The IEEE International Conference on Intelligent Computing and Intelligent Systems*, pages 448–452, 2009.

[10] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, 2006.

[11] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR'07*, pages 1–8, 2007.

[12] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008*, pages 1–8, 2008.

[13] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Ninth IEEE international conference on computer vision, 2003. Proceedings*, pages 1470–1477, 2003.

[14] A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and trecvid. In *MIR'06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, pages 321–330, 2006.

[15] T. Tuytelaars and C. Schmid. Vector quantizing feature space with a regular lattice. *IEEE International Conference on Computer Vision*, 2007.

[16] M. Yeh and K. Cheng. Video copy detection by fast sequence matching. In *Proceeding of the ACM International Conference on Image and Video Retrieval*, pages 1–7, 2009.