Accurate Content-Based Video Copy Detection with Efficient Feature Indexing

Yusuke Uchida
KDDI R&D Laboratories Inc.
2-1-15 Ohara, Fujimino-shi
Saitama, Japan
ys-uchida@kddilabs.jp

Motilal Agrawal
SRI International
333 Ravenswood Avenue
Menlo Park, CA, USA
agrawal@ai.sri.com

Shigeyuki Sakazawa
KDDI R&D Laboratories Inc.
2-1-15 Ohara, Fujimino-shi
Saitama, Japan
sakazawa@kddilabs.jp

ABSTRACT

We describe an accurate content-based copy detection system that uses both local and global visual features to ensure robustness. Our system advances state-of-the-art techniques in four key directions. (1) Multiple-codebook-based product quantization: conventional product quantization methods encode feature vectors using a single codebook, resulting in large quantization error. We propose a novel codebook generation method for an arbitrary number of codebooks. (2) Handling of temporal burstiness: for a stationary scene, once a query feature matches incorrectly, the match continues in successive frames, resulting in a high false-alarm rate. We present a temporal-burstiness-aware scoring method that reduces the impact from similar features, thereby reducing false alarms. (3) Densely sampled SIFT descriptors: conventional global features suffer from a lack of distinctiveness and invariance to non-photometric transformations. Our densely sampled global SIFT features are more discriminative and robust against logo or pattern insertions. (4) Bigram- and multiple-assignment-based indexing for global features: we extract two SIFT descriptors from each location, which makes them more distinctive. To improve recall, we propose multiple assignments on both the query and reference sides. Performance evaluation on the TRECVID 2009 dataset indicates that both local and global approaches outperform conventional schemes. Furthermore, the integration of these two approaches achieves a three-fold reduction in the error rate when compared with the best performance reported in the TRECVID 2009 workshop.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*Indexing methods*

General Terms

Algorithms, design, performance

Keywords

Near-duplicate detection, content-based copy detection, approximate nearest neighbor search

ICMR '11, April 17-20, Trento, Italy

1. INTRODUCTION

Digital multimedia content, computer, and Internet technologies have become ubiquitous, with digital videos used extensively in many applications. Copyright infringement poses a significant issue for one of the applications — online video-sharing services. Because many upload video clips to these sites without proper copyright release, an automated system that detects copies of copyrighted video is needed.

In such an automated system, content holders register copyrighted content with the operators of video-sharing sites in advance. The operators extract features from the copyrighted content and store them in a database. When a user uploads a video clip, features are extracted from the uploaded video clip in the same way and the database is searched for a match. If the database contains matching content, the uploaded content is considered to be a copy of copyrighted content and filtered out, or some another action is taken in compliance with the content holder's intentions.

Given a test collection of videos and a set of queries, the goal of content-based copy detection (CBCD) technology is to determine for each query the place, if any, that some part of the query occurs, with possible transformations, in the test collection. In recent years, CBCD has attracted considerable research attention. The TRECVID [1] workshop series encourages research in content-based retrieval of digital video by providing a large test collection, uniform scoring procedures, and a forum for organizations interested in comparing their results.

For an automated CBCD system to be usable, it is important that it has the following properties:

Robustness to various transformations. The video may have been subject to severe transformations, including the addition of patterns, embedding in a different video, deletion of audio channels, and geometric transformations.

Computationally efficient. The system must be sufficiently efficient so that its application to a large collection of videos requires only modest computational power.

Low false alarms. A system with too many false detections is annoying and requires ongoing operator intervention to filter out the false alarms.

Our CBCD system has made significant advances in these directions. For accurate detection, our system is based on two types of CBCD schemes: one uses local invariant features, and the other

uses dense-sampled global features, thereby achieving robustness to various transformations. We present two indexing schemes to index these feature vectors efficiently in an inverted index scheme, based on a bag-of-features representation. We show that integration of these two types of features greatly improves results.

Local approach. Our first scheme is based on SIFT like local invariant features [4], in which a novel version of the product-quantization-based method [12] is integrated with an inverted index to perform an accurate nearest neighbor search (NNS). Our novelty lies in the ability to use an arbitrary number of codebooks to quantize the residual vectors. Doing so enables the product quantizer to switch codebooks according to feature distributions, thereby increasing NNS accuracy. We also point out a temporal burstiness problem and present a temporal-burstiness-aware scoring method that reduces the impact from similar features, thereby reducing these false alarms.

Global approach. Conventional global features suffer from lack of distinctiveness and invariance to nonphotometric transformations. Our second scheme is based on novel densely sampled global SIFT features, which are more discriminative and robust against logo or pattern insertions. Two feature descriptors are extracted at each sampled feature location, resulting in bigram feature representations. We perform multiple assignments for bigram feature representations on both the reference and query sides to improve the recall. We also present a technique to handle the Picture-In-Picture (PIP) transformation using global features.

2. RELATED WORK

Any CBCD system has two major components. The first component involves generating compact and discriminative signatures for each reference video that are invariant to various transformations. The second component is the similarity search algorithm, which uses the signatures to efficiently search for near-duplicate video keyframes. To date, many algorithms have been developed for each of the two components.

2.1 Local-feature-based CBCD

Given the success of local invariant features in the area of image retrieval [18, 20, 10], local features have also been adopted to CBCD systems [26, 5]. In local-feature-based CBCD systems, interest points [15, 16] are extracted from each video keyframe and summarized by their feature descriptors.

One challenge in local CBCD systems is efficient feature indexing for similarity search. Because hundreds of millions of local features are extracted in a large-scale system, feature indexing is critical for efficiency. To date, many indexing methods have been proposed such as ANN [3], LSH [2], randomized kd-tree algorithm [20, 21] and FLANN [17]. For a large-scale system consisting of millions of video keyframes, the indexing methods mentioned above are not suitable because they require the feature vectors themselves to be stored in indices [12]. Therefore, many CBCD schemes are developed over a bag-of-visual words + inverted index (BoVW+II) framework [22]. In BoVW+II, each feature vector is quantized into a visual word and stored in a inverted index with a time stamp or other information related to the feature. In addition to the efficiency that the inverted index structure confers, storage of feature vectors in the index is not needed. Because a naive BoVW+II approach suffers from many false matches of local features, embedding methods [10, 25, 12] are integrated into BoVW+II. Embedding methods encode feature vectors into more compact signatures (e.g., 32–128 bit code) and filter out most false matches according to the signatures. Although a product-quantization-based method [12] has been shown to achieve the best performance among the methods mentioned above, it still suffers from large quantization errors because it encodes all features with a single codebook and does not consider the divergence of feature distributions among visual words.

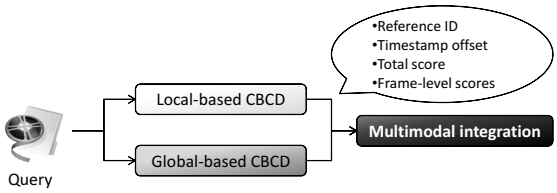


Figure 1: Framework of our CBCD system.

Another difficulty for local features is the temporal burstiness effect, in which incorrect query and reference feature pair matches in successive frames, especially in static regions like background, cause many false alarms in the final detection results. A few methods have been proposed to suppress non-consistent feature matches using spatial information such as weak geometric consistency (WGC) constraints [10] or scale-rotation invariant pattern entropy (SR-PE) [26]. However, because these bursty matches usually have the same spatial information over frames, the matches are always consistent with each other, and spatial verifications do not help suppress them.

2.2 Global-feature-based CBCD

Global features have been traditionally used in CBCD systems because of their efficiency and robustness against photometric transformations such as blur, compression, and gamma change. Global signatures summarize the entire frame into a single descriptor. An ordinal measure (OM) [9, 24], one of the major global descriptors, has been shown to be robust to changes in resolution and illumination. In [19], OM is extended to include temporal information for more robustness. Recently, gradient-based features [14] have also been shown to achieve good robustness and pairwise independence.

One difficulty in global feature approaches is a lack of distinctiveness. Many schemes adopt low-dimensional global features and require a sequential search because each global feature is not distinctive enough [13]. Sequential search is infeasible in a large-scale system, however, because it requires a lengthy processing time proportional to reference and query size.

3. OVERVIEW OF THE PROPOSED CBCD SYSTEM

Figure 1 provides an overview of the proposed CBCD system. Each video is resampled at a fixed frame rate (1 Hz for all our experiments) to extract keyframes. That rate makes our reference database size tractable and also helps us deal with frame rate changes.

For each keyframe of both reference and query videos, our system extracts local and global features. The features from the query video are used to produce query results independently in the form of a reference video identifier, a corresponding timestamp offset, a total score, and frame-level scores. The results of each of the two types of features are integrated in a postprocessing stage to make the results reliable and accurate. It is easy to see that additional

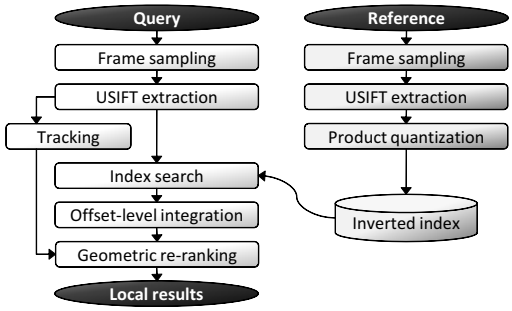


Figure 2: Overview of local-feature-based CBCD.

modalities such as audio-based queries can be easily integrated into our system.

Local and global feature integrations for CBCD task have been used because each feature works well for different transformations, which are in some sense complementary: local features are robust even against large geometric transformations, whereas global features are more robust against nongeometric (photometric) transformations. In addition, global features are much faster to compute.

The following sections describe each of the two modalities in some detail, as well as the integration framework and evaluation results.

4. LOCAL-FEATURE-BASED CBCD

Figure 2 illustrates a functional diagram of our local-feature-based CBCD scheme. We use local scale invariant features and their bag-of-features representations, which have been used widely in image/video retrieval areas [22, 20, 10, 5]. The key ideas here are (1) the use of the upright SIFT (USIFT) feature descriptor for distinctiveness, (2) a novel variant of product-quantization-based indexing for more accurate NNS and (3) consideration of the temporal burstiness of local features in scoring to suppress false alarms.

4.1 USIFT extraction

For each keyframe, we detect the SIFT features and use USIFT as the feature descriptor. USIFT is more distinctive and faster to compute than SIFT [4]. Because most video transformations include a small rotation, USIFT can be used without degrading the robustness of the CBCD system. Each key frame is then represented by a set of features points (bag-of-features). Each feature point contains the following information: video identifier *id* (only for reference video frame), timestamp *ts*, position of feature point (x, y), and USIFT feature vector $\mathbf{f}$.

4.2 Product-quantization-based indexing

Because each frame has a large number of USIFT feature descriptors, the indexing method must be accurate. We use a novel version of the product-quantization-based method [12] to index USIFT feature vectors obtained in the USIFT extraction step. Our approach increases NNS accuracy at a small increase in memory overhead.

In the product-quantization-based scheme [12], a reference vector is first quantized by a coarse quantizer with the size of N (50K in this paper); the residual vector from the corresponding centroid is then decomposed into S (8 in this paper) residual subvectors. Finally, the S residual subvectors are independently encoded (product quantization) into a short code using codebooks for product quantization. This can be integrated with an inverted index, referred to as IVFADC in [12].

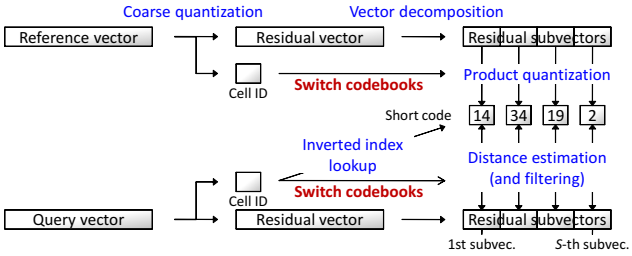


Figure 3: Modified product-quantization-based indexing.

In our CBCD system, we modify the original product-quantization-based method to use multiple codebooks in product quantization, whereas a residual subvector is quantized by a single codebook in the original algorithm. The framework of our product-quantization-based indexing method is shown in Figure 3. Our scheme allows us to switch codebooks depending on the cell that the input vector falls into in the coarse quantization step. The memory requirements for using a different codebook for each of the N cells becomes prohibitively large. Instead, we use an arbitrary number M ($M \ll N$) of codebooks for each of the s-th residual subvectors created by the following procedure:

1. For each $n = 1, \cdots, N$, create a set of s-th residual subvector $\mathcal{R}_{s,n}$ from all training vectors assigned to the n-th cell in a coarse quantizer and assign $\mathcal{R}_{s,n}$ to a random cluster $m \in \{1, \cdots, M\}$.

2. For each $m = 1, \cdots, M$, update codebook $C_{s,m}$ by clustering all s-th residual subvectors assigned to the cluster m.

3. For each $n = 1, \cdots, N$, assign $\mathcal{R}_{s,n}$ to the cluster $\hat{m}$, such that the codebook $C_{s,\hat{m}}$ achieves minimum error in quantization of $\mathcal{R}_{s,n}$. The identifier $\hat{m}$ is also stored in a table $T_{s,n}$.

4. Repeat Step 2 and Step 3 until convergence occurs.

Now, M codebooks can be switched according to the table T: s-th residual subvectors that belong to n-th cell in coarse quantization is to be quantized by the codebook with the identifier $T_{s,n}$ in product quantization.

In the case of $M = 1$, our codebook becomes identical to the codebook used in [12]. Use of larger M reduces the quantization error in product quantization and improves NNS accuracy at the cost of the additional memory requirements. Here, we decompose residual vectors into 8 residual subvectors, and set the number of codebooks M used in product quantization to 256. The size of each product quantization codebook is set to 256. In this case (for 128-dimensional USIFT vectors), only 256×256×128~8M byte memory is required to store codebooks, whereas 256×50K×128~1.6G byte memory is required if N codebooks are used. Each feature is encoded into 8×8=64 bits code, and the code is stored in the inverted index for distance estimation.

4.3 Index search

In the search step, USIFT features in a query video are efficiently matched with reference features using an inverted index [22]. In addition, we can filter out many false matches through distance calculation between a residual vector of a query feature and a short code of a reference feature [12]. Here, we filter out reference features with a distance larger than 0.3 (assuming USIFT features are normalized to a norm of 1.0). The result of the index search step is a set of matched keypoint pairs (Q, R), where each query keypoint Q has timestamp ts_q and coordinate (x_q, y_q), and each reference keypoint R has video identifier id, timestamp ts_r and coordinate (x_r, y_r).

4.4 Offset-level integration

Feature-level results obtained in the index search step are integrated into offset-level results using a voting scheme [13, 5]. Every matched keypoint pair (Q, R) votes for the corresponding bin $b[id][ts_r - ts_q]$ in the 2D Hough space. After performing non-maxima suppression and thresholding, we obtain the top 200 hypothesis represented by $(id, offset)$, where $offset = ts_r - ts_q$. Each hypothesis $(id, offset)$ has a list of matched keypoint pairs that have voted for the hypothesis, and this information is used for geometric verification. We also tried to incorporate the WGC method [10] using only scale information[1], but found that it did not contribute to accuracy of our experiments probably because WGC based on scale is less useful than one based on the orientation shown in [23].

4.5 Keypoint tracking

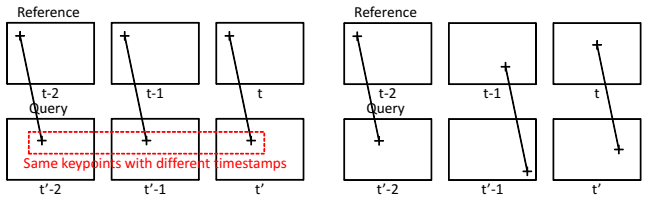
In the tracking step, query keypoints are tracked against keypoints in one and two previous frames. Then, each query keypoint Q_i has a list of keypoints Q s.t. $0 < ts_q - ts'_q \leq 2$, $(x_q - x_{q'})^2 + (y_q - y_{q'})^2 < r^2$ and $\|\mathbf{f}_q - \mathbf{f}_{q'}\|^2 < th^2$. Here r is the maximum distance between two tracked feature points and th is the maximum distance between their feature vectors. The lists are used in the geometric re-ranking step to alleviate the temporal burstiness effect.

4.6 Geometric re-ranking and handling temporal burstiness

Geometric verification and re-ranking are performed on the top 200 results (candidates) obtained in the offset-level integration step. For each result, we estimate the transformation matrix with 4 degrees of freedom [20] between the query video and the reference video using random sample consensus (RANSAC) algorithm and obtain the score according to the number of inliers. Instead of simply counting the number of inliers, our scoring scheme takes temporal burstiness into consideration in a manner similar to [11, 6].

Figure 4 shows an example of temporal burstiness phenomenon of local features. In the bursty case (Figure 4a) an incorrect match propagates to successive frames. This is different from the non-bursty case (Figure 4b) where different features match in successive frames. Eventhough the number of feature matches is the same in both cases, the bursty case is more likely to be a false match. Temporal burstiness of local features is seen more frequently than that of global features [6] because it occurs even in scenes with a dynamic foreground and static background, which is not the case of global features. In a naive scoring method, the bursty keypoint matches between irrelevant videos often get higher scores than true matches, which results in a high false-alarm rate. To alleviate this problem, we introduce a new scoring method to reduce the scores associated with the bursty matches.

[1]Orientations of matched keypoint pairs are always *consistent* in the USIFT-based scheme.



(a) Bursty case (b) Non-bursty case

Figure 4: An example of the temporal burstiness effect. (a) The same feature is matched on successive frames, which might occur even between irrelevant videos, especially in stationary scenes. (b) Different features are matched on successive frames, which rarely happens between irrelevant videos.

For each query point Q_i, the number of successive matches associated with similar points of Q_i is counted by $c_i = \max_{j \in \mathcal{T}_i} c_j + 1$, where $\mathcal{T}_i$ denotes a list of keypoint identifiers in one and two previous frames tracked by Q_i. Then, the score is added using a monotonic increasing function f:

$$\text{score} += f(c_i) - f(c_i - 1). \tag{1}$$

We experimented with three functions — f_1 f_2, and f_3:

$$f_1(c) = c, \quad f_2(c) = \sqrt{c}, \quad \text{and} \quad f_3(c) = \begin{cases} 0 & \text{if } c = 0 \\ 1 & \text{otherwise.} \end{cases} \tag{2}$$

Apparently, f_1 produces the same score as the conventional voting scheme does, wherein each point match increments the score by one. For f_2 and f_3 on the other hand, the added score becomes smaller when the number of matches in successive frames becomes larger. Thus bursty cases with long tracks of feature points will get smaller scores. The most extreme function, f_3, corresponds to removal of multiple matches from the scoring function.

Finally, we obtain a list of results based on local features. Each result includes the reference video identifier, corresponding timestamp offset, the updated score and frame-level scores as described in Section 3.

5. GLOBAL-FEATURE-BASED CBCD

Figure 5 presents a functional diagram of our global-feature-based CBCD system. The system relies on dense sampling of a video frame at fixed locations, enabling the adoption of BoVW+II framework and making it more robust against pattern insertions or other weak geometric transformations. The scale of the features dictates the window size to be used for computing the descriptor at that feature location. We have used scales that correspond to window sizes greater than or equal to one-third the total width or height. Densely sampled global features have been used previously for image classifications [7] or 3D model retrieval [8], with all sampled features quantized into visual words and summarized in one histogram. In our global-feature-based system, however, all sampled features are used in the same manner as in the local-feature-based system instead of summarizing in one histogram.

For a given scale, feature locations are chosen so that the neighbors in the x and y directions overlap by at least 75%. A total of 121 windows results, although our approach for TRECVID used a subset of only 40 such windows at the higher scales. Figure 6a shows two such neighboring feature locations for a scale of one-third the width and height of the image. In comparison with local features,

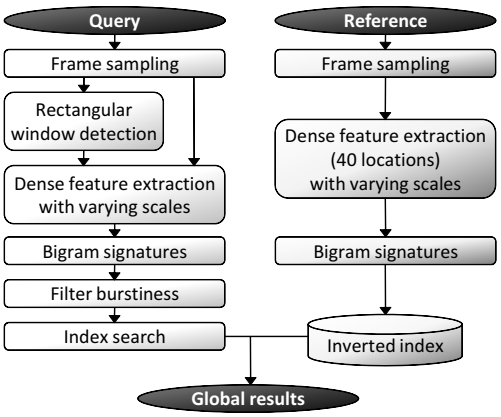
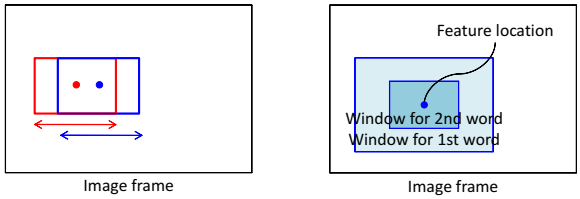


Figure 5: Overview of global-feature-based CBCD.



(a) Overlapping windows (b) Windows for bigram

Figure 6: (a) shows two neighboring windows corresponding to a scale of one-third the width and height. The feature locations (shown as circles) are such that there is a 75% window overlap between neighbors; (b) shows the two nested windows for making a bigram feature descriptor at a given feature location and scale.

feature detection and scale selection of the selected features are unnecessary, thereby increasing speed.

The indexing scheme for local-features-based on multiple codebook product quantization to increase accuracy is not directly applicable to global features primarily because each image frame has a small number of global features (40 in our case). In comparison, for local features, each frame has a few thousand features. Given the small number of features, emphasis on global features shifts to an increase in recall rather than accuracy. To improve recall, we propose multiple assignments on both the query and reference sides. Another consequence of the small number of global features is that the support region for each global feature is larger than the support region for local features. Therefore a 128-dimensional SIFT descriptor is not discriminative enough. We extract two SIFT descriptors from each location and use their bigram along with multiple assignments for our BoVW+II-based indexing scheme.

5.1 Bigrams as feature descriptors

Conventional global descriptors such as rank-based features [9, 24] are not distinctive enough for our dense sampling strategy. SIFT descriptors [15] on the other hand are known to perform well. We used bigrams of SIFT descriptors to make it more distinctive.

Given a feature location and scale, we extract two 128-dimensional SIFT descriptors from the window corresponding to that scale and use the bigrams as the descriptor for that feature location and scale. The first descriptor is extracted from the full window, and the second descriptor is extracted from a subwindow centered on the feature location with a width and height half of that of the full window. We quantize each of the two descriptors into 10,000 words independently, with the global descriptor for that feature location and scale then represented by their bigrams. Figure 6b shows the two nested windows for bigram computation of a feature location and scale.

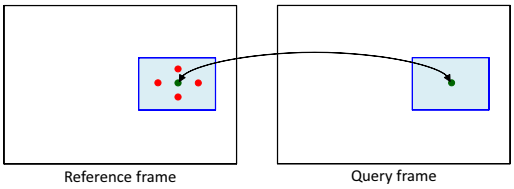


Figure 7: Automatic correspondence determination based on position and scale. The green features are corresponding locations. The red feature locations are neighbors in the reference frame and are included to make the matching robust to small shifts.

Essentially, this process is same as that for performing product quantization on larger 256-dimensional vectors. Each of the 128-dimensional SIFT descriptors can be seen as a subvector of the larger 256-dimensional vector. As in product quantization, each of the two 128-dimensional vectors is quantized independently. Therefore this approach is efficient because the codebooks that need to be stored in memory are smaller and the quantization step also involves fewer distance computations. A vocabulary size of 10,000 for each of the words results in a bigram vocabulary size of 100 million. We have found that using the bigram results in much better performance and also results in faster execution time.

5.2 Indexing with multiple assignments

We use the bigrams to make an inverted index and index each frame of the reference video. The use of bigrams results in a large vocabulary size; to increase recall, we use multiple assignments for both the query and reference sides. Each descriptor in the bigram is assigned to five words. Therefore, each bigram in both the reference and query side is assigned to 25 words.

5.3 Automatic geometric correspondence

On the query side, global features are computed in a manner similar to that for the reference video. When querying, each feature location in the query video corresponds to the same feature location and scale in the reference video. Therefore, we do not have to perform matching steps for the feature locations. To make the approach robust to small shifts, we also match it with its four neighbors in the reference video with the same scale. Figure 7 demonstrates these four neighbors in the reference video.

5.4 Special case: PIP detection

Because our global features are not scale-invariant, they do not work well when the frame undergoes drastic changes in scale. Therefore, the approach described so far does not work well for PIP transformation. Instead, to detect PIP transformations, we detect rectangular windows in the query video by accumulating image gradients in the x and y directions over time. We run edge detection in each video frame with a very low threshold and then accumulate the number of edges in each row and column temporally. Rows and columns with a sufficient number of accumulated edges are candidates for the edges of the window. The intersection of a pair of row and column edges is a candidate corner for the rectangular window.

Figure 8: Examples of detected PIP window.

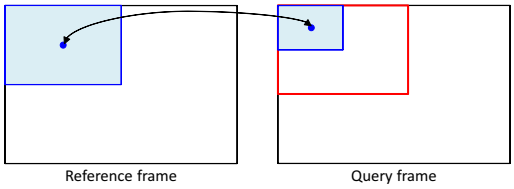


Figure 9: Geometric correspondence for the PIP transform. Matching is performed relative to the detected window.

Finally, those windows with a sufficient number of edges in at least two of the four sides are the candidate detected windows.

In our experiments, the thresholds we set result in our system missing very few such windows; however, the threshold also results in the system occasionally detecting substantially more windows than are actually present. Figure 8 show the detected PIP windows in a few frames.

If a rectangular window is detected in the query window (for a PIP transformation), the feature locations and scale are determined relative to the detected rectangular region instead of the whole frame (see Figure 9). When a PIP window is detected, querying is performed with the detected window in addition to the whole frame. This takes care of false PIP detections.

5.5 Querying

For each query frame, we use the 40 global features to find a corresponding match in the reference frames. The score of a match is simply the count of the number of global feature locations matched. Each match results in a vote for the time offset corresponding to that match. Temporal burstiness is also considered in the voting step using an approach similar to that described in the local part: we compare the bigrams from two consecutive features at the same location; if either of the two words is the same, we do not use that feature for our matching. This approach has similar effect to the scoring method using f_3 described in Section 4.6. After voting, non-maxima suppression and thresholding are also performed to obtain the list of global results.

6. LOCAL-GLOBAL INTEGRATION

Each of the global and local modalities discussed above produces a sorted list of results consisting of the reference video identifier, corresponding timestamp offset, total score, and frame-level scores. These results are sorted by the total scores, with the best result having the highest total scores. These individual results are integrated to produce a final list of results.

6.1 Integration and re-scoring

First, for each modality, all total scores are normalized by the second top score to emphasize distinctiveness and normalize scores among all queries. Then, any two modalities that indicate the same

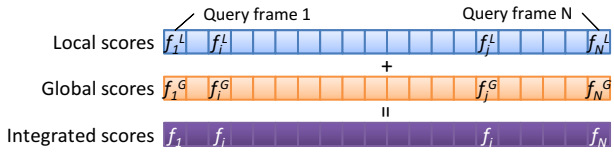


Figure 10: Integration of frame-level scores. Frame-level scores are normalized so that $\sum_{t=1}^{T} f_t^L = 1.0$ and $\sum_{t=1}^{T} f_t^G = 1.0$ before integration.

id and *offset* are integrated by simply summing the total scores. Frame-level scores are also integrated frame-by-frame after they are normalized so that the sum of frame-level scores becomes 1.0 as shown in Figure 10. We have found that this normalization step slightly improves the accuracy of segment localization compared with simple linear weighting.

6.2 Segment localization

Finally, start frame $\hat{i}$ and end frame $\hat{j}$ of the copied segment in the query video is determined by

$$\arg\max_{i,j} S_{i,j}. \tag{3}$$

$S_{i,j}$ is a partial sum of frame-level scores from frame i to j normalized by the segment length:

$$S_{i,j} = \frac{\sum_{t=i}^{j} f_t}{\sqrt{j-i+1+\alpha}+\beta} \tag{4}$$

where T denotes the number of keyframes in a query video and $f_1, \cdots, f_T$ indicate frame-level scores. Although this is a brute-force computation, use of integral image can greatly accelerate the computation. In this paper, we set $\alpha = 40$ and $\beta = 40$.

7. EXPERIMENTS

We evaluated our CBCD system using the TRECVID 2009 dataset [1]. To evaluate the video-only queries, we chose the 2009 dataset rather than the most recent 2010 dataset because all queries in the more recent dataset have both video and audio, precluding evaluation of video-only queries. Furthermore, because the 2010 dataset includes only Internet videos, we believe that the 2009 dataset is more representative for copyright content protection. The 2009 dataset includes 838 reference videos (about 400 hours in total) and 1,407 query videos. Each query has been edited by the seven transformations listed in Table 1, including both photometric transformations and geometric transformations.

In the framework of the TRECVID CBCD task, a CBCD system is characterized by three key performance measures:

Detection accuracy. Normalized detection cost rate (NDCR)[2] is used to evaluate the detection accuracy fairly among different systems. NDCR measures the trade-off between the cost of false negatives and false positives, and is defined by a weighted mean of the two errors. There are two profiles, referred to as BALANCED and NOFA, that are related to NDCR weighting. Although the former assigns balanced weights to both false negatives and false positives, the latter assigns much greater weight to false positives. In this paper, we show results only for the NOFA profile because NDCR

[2]http://www-nlpir.nist.gov/projects/tv2010/Evaluation-cbcd-v1.3.htm

Table 1: Query transformations.

T2	Picture in picture
T3	Insertions of pattern
T4	Strong re-encoding
T5	Change in gamma
T6	Decrease in quality (combinations of 3 transformations from blur, gamma, frame dropping, contrast, compression, ratio, and noise)
T8	Post production (combinations of 3 transformations from crop, shift, contrast, caption, flip, insertion of pattern, and picture in picture)
T10	Combinations of 5 transformations from T2-T8

values become almost the same for BALANCED and NOFA profiles on the TRECVID 2009 settings.

Localization accuracy. The accuracy of localization is measured by the F-measure, which is the harmonic mean of the precision and recall of the detected copy location relative to the true video segment. It is calculated only for correctly detected queries, and it reflects the overlapped area between the detected query and reference segments.

Efficiency. Efficiency is evaluated by the mean processing time per query.

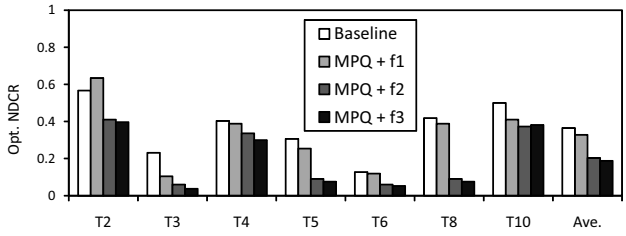


Figure 11: NDCR measures of a local CBCD system for a NOFA profile. Lower values mean better results. Performance of the baseline scheme (see text) is compared with Multiple Product Quantization (MPQ) and the three scoring schemes $f1$, $f2$ and $f3$.

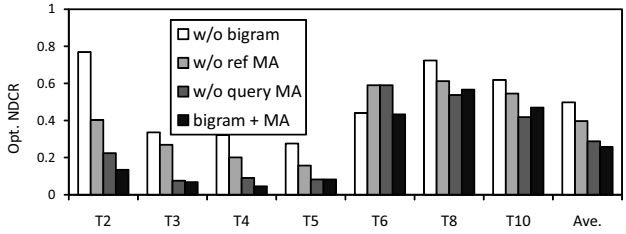


Figure 12: NDCR measures of a global CBCD system for the NOFA profile. Lower values mean better results. Results from our proposed scheme (bigram+MA) are compared with schemes without multiple assignments (MA) on the reference (w/o ref) and query side (w/o query) and also without bigram (w/o bigram).

7.1 Performance evaluation of the local approach

Figure 11 compares the results of different local schemes using the NDCR measure for different video transformations. Baseline scheme (described in Section 4) does not have multiple codebooks for product quantization and neither does it handle temporal burstiness. This scheme is quite similar to the framework described in [5], with minor differences arising because we use USIFT features and also product-quantization-based indexing. MPQ+f1, MPQ +f2, and MPQ+f3 correspond to our proposed scheme with multiple-codebook-based product quantization (MPQ) and temporal-burstiness-aware scoring using f_1, f_2, and f_3 measures described in Section 4.6. As mentioned, f_1 produces the same score as the conventional voting scheme. Therefore, the improvement in MPQ+f1 when compared with Baseline demonstrates the contribution of multiple codebooks for product quantization. It is also clear that handling temporal burstiness significantly affects performance (f_2 and f_3), especially in T5 and T8 transformations. It is surprising that the most extreme scoring function f_3 achieved the best performance, because it was not the best choice in the case of spatial burstiness[3]. It implies that temporal burstiness appears much more frequently and is more important for accuracy than spatial burstiness. On an average, our proposed scheme results in an almost two-fold decrease in the NDCR measure against the baseline scheme.

7.2 Performance evaluation of the global approach

Figure 12 shows the evaluation results of various global schemes. Bigram+MA is our proposed scheme with bigrams and multiple assignments (MA) as presented in Section 5. Without bigram (w/o bigram) uses a single 128-dimensional SIFT descriptor. The codebook size in this case is 10,000, and indexing is done with five multiple assignments both on the query and reference sides. It is clear that bigram contributes significantly to accuracy. To assess the contribution of multiple assignments, we also evaluate two schemes based on bigrams but without multiple assignments: w/o ref MA does *not* have multiple assignments on the reference side and similarly w/o query MA does *not* have multiple assignment on the query side. It is also clear from Figure 12 that multiple assignments are also crucial for a good NDCR. In addition, the performance of w/o query MA is quite close to our scheme. Because w/o query MA has multiple assignments in the reference, it is clear that multiple assignments in the reference video make the largest contribution to accuracy. As in local approach, on average, our proposed scheme with bigrams and multiple assignments results in an almost two-fold decrease in the NDCR measure.

[3] f_3 corresponds to multiple match removal (MMR) in [11].

7.3 Performance evaluation of the integrated results

Figure 13 shows the results of both local (MPQ+f3), global-feature-based (bigram+MA) and integrated schemes for the NOFA profile. The best scores among all participants of the TRECVID 2009 CBCD task for each video transformation are also shown. In all cases, except T6 and T8, the global features perform better than the best reported results. Furthermore, our global features perform better than local features for transformations T2 and T4 and are almost on a par with those for T5. Whereas we expect global features to perform better for the photometric transformations T4 and T5, the strong performance for T2 is primarily due to our special PIP handling (Section 5.4). Our local features consistently outperform the best reported system, and they make the greatest contribution in complex transformations such as T6 and T8. Finally, it is clear that the integration of the two substantially improves accuracy, especially for transformations T3 through T6. That improvement demonstrates that the results of local and global schemes are consistent, thereby increasing accuracy. For other transformations,

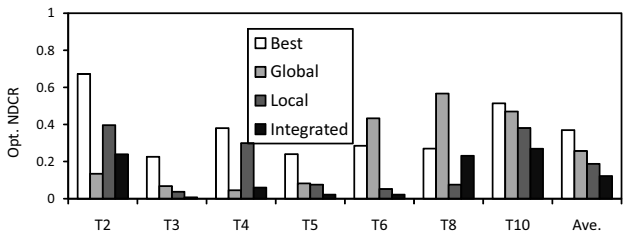


Figure 13: NDCR measures for the NOFA profile. Results are shown for our local, global and integrated scheme. The best scores from participants of TRECVID 2009 CBCD task are also shown.

Table 2: Localization accuracy (F-measure).

	T2	T3	T4	T5	T6	T8	T10
Global	0.937	0.938	0.936	0.940	0.939	0.936	0.941
Local	0.943	0.920	0.934	0.930	0.913	0.931	0.931
Integrated	0.960	0.952	0.949	0.961	0.946	0.957	0.956

the integration of these two schemes results in performance somewhere midway between the performances of either scheme. On average, our integrated system results in a three-fold improvement in accuracy when compared with the best reported results.

7.4 Evaluations based on other criteria

Table 2 shows the localization accuracy of our schemes. It is clear that our schemes have achieved almost perfect performance in terms of segment localization criteria (perfect performance corresponds to a score of 1.0). Our localization accuracy is high because the copied segments are localized at a later stage, after the video id and offsets have been determined. It can also be seen that integration of local and global features increases localization accuracy.

For all our experiments, we used a Windows XP system with a Core i7 2.93 GHz CPU and 24 GB main memory. In terms of computational cost, on average our scheme required 121 seconds per query. Our time is longer than the median for all participants in TRECVID 2009 (median was about 80 seconds), whereas the global part required only 15 seconds out of 121 seconds.

8. CONCLUSION

Our system for CBCD of video is multimodal and integrates both local and densely sampled global features to produce robust results. We have proposed novel indexing schemes to efficiently and accurately perform retrieval using both these features. Advances in local indexing include optimized multiple-codebook-based product quantization to increase NNS accuracy. On the global side, we have used bigrams along with multiple assignments in indexing. We have validated our choice of indexing with extensive experimental results. We have also demonstrated that it is important to handle temporal burstiness to suppress false alarms and have presented a scoring method that accounts for temporal burstiness.

Results on a large dataset of over 400 hours of video demonstrate that both our global and local schemes outperform the best algorithm presented in the TRECVID 2009 workshop. The integration of these two schemes results in a three-fold reduction in the error rate. Furthermore, our scheme yields very good segment localization results and its computational time is almost the same as other state-of-the-art algorithms. Future work includes integrating these two features more tightly at the frame level and also adding audio features in our framework to handle very difficult video transformations. We are also investigating specialized algorithms for hard video transformations such as camcording.

9. REFERENCES

[1] `http://www-nlpir.nist.gov/projects/trecvid/`.

[2] A. Andoni. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *Proc. of FOCS*, pages 459–468, 2006.

[3] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu. An optimal algorithm for approximate nearest neighbor searching in fixed dimensions. *JACM*, 45(6):891–923, 1998.

[4] G. Baatz, K. Koser, D. Chen, R. Grzeszczuk, and M. Pollefeys. Handling urban location recognition as a 2d homothetic problem. In *Proc. of ECCV*, 2010.

[5] M. Douze, H. Jégou, and C. Schmid. An image-based approach to video copy detection with spatio-temporal post-filtering. *IEEE Trans. on Multimedia*, 12(4):257–266, 2010.

[6] M. Douze, H. Jégou, C. Schmid, and P. Pérez. Compact video description for copy detection with precise temporal alignment. In *Proc. of ECCV*, 2010.

[7] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *Proc. of CVPR*, 2005.

[8] T. Furuya and R. Ohbuchi. Dense sampling and fast encoding for 3d model retrieval using bag-of-visual features. In *Proc. of CIVR*, 2009.

[9] X. Hua, X. Chen, and H. Zhang. Robust video signature based on ordinal measure. In *Proc. of ICIP*, pages 685–688, 2004.

[10] H. Jégou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *Proc. of ECCV*, pages 304–317, 2008.

[11] H. Jégou, M. Douze, and C. Schmid. On the burstiness of visual elements. In *Proc. of CVPR*, 2009.

[12] H. Jégou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. *IEEE Trans. on PAMI*, 33(1):117–128, 2011.

[13] J. Law-To, L. Chen, A. Joly, and I. Laptev. Video copy detection: a comparative study. In *Proc. of CIVR*, pages 371–378, 2007.

[14] S. Lee and Y. H. Suh. Video fingerprinting based on orientation of luminance centroid. In *Proc. of ICME*, pages 1386–1389, 2009.

[15] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[16] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. A comparison of affine region detectors. *IJCV*, 60(1-2):43–72, Nov. 2005.

[17] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *Proc. of VISAPP*, 2009.

[18] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *Proc. of CVPR*, 2006.

[19] S. Paisitkriangkrai, T. Mei, J. Zhang, and X. S. Hua. Scalable clip-based near-duplicate video detection with ordinal measure. In *Proc. of CIVR*, pages 121–128, 2010.

[20] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial

matching. In *Proc. of CVPR*, 2007.

[21] C. Silpa-Anan and R. Hartley. Optimised kd-trees for fast image descriptor matching. In *Proc. of CVPR*, 2008.

[22] J. Sivic and A. Zissermane. Video google: A text retrieval approach to object matching in videos. In *Proc. of ICCV*, pages 1470–1478, 2003.

[23] S. S. Tsai, D. M. Chen, G. Takacs, V. Chandrasekhar, R. Vedantham, R. Grzeszczuk, and B. Girod. Fast geometric reranking for image based retrieval. In *Proc. of ICIP*, 2010.

[24] M. Usman and C. Kim. Real time video copy detection under the environments of video degradation and editing. In *Proc. of ICACT*, pages 1583–1588, 2008.

[25] J. Wang, S. Kumar, and S. F. Chang. Semi-supervised hashing for scalable image retrieval. In *Proc. of CVPR*, 2010.

[26] W. L. Zhao and C. W. Ngo. Scale-rotation invariant pattern entropy for keypoint-based near-duplicate detection. *IEEE Trans. on Image Processing*, 18(2):412–423, 2009.