A Content-based Video Copy Detection Method with Randomly Projected Binary Features

Chenxia Wu Jianke Zhu Jiemi Zhang
College of Computer Science, Zhejiang University, China
chenxiawu@hotmail.com, jkzhu@zju.edu.cn, jmzhang10@gmail.com

Abstract

Video copy detection has been actively studied in a wide range of multimedia applications. This paper presents a novel content-based video copy detection method using the randomly projected binary features. A very efficient sparse random projection method is employed to encode the image features while retaining their discrimination capability. By taking advantage of the extremely fast similarity computation of binary features using Hamming distance, we present a keyframe-based copy retrieval method that exhaustively searches the copy candidates from the large video database without indexing. Moreover, an effective scoring and localization algorithm is proposed to further refine the retrieved copies and accurately locate the video segments. The experimental evaluation has been performed to show the efficacy of the proposed randomly projected binary features. The promising results in the TRECVID2011 [14] content-based copy detection task demonstrated the effectiveness of our proposed approach.

1. Introduction

During the past decade there has been an exponential growth of online videos due to the huge amount of user-contributed multimedia contents through abundant video sharing websites. The massive publishing and sharing of videos raises the problem of a large amount of near-duplicate copies. Video copy detection is essential to many real-world applications, including copyright protection, law enforcement investigations, business intelligence, advertisement tracking and redundancy removal.

Generally, a video copy is a segment of video sequence that is transformed from another one by means of inserting patterns, compression, change of gamma, decrease in quality, camcording, etc. In this paper, we focus on the content-based copy detection using visual features which offers an alternative solution to the traditional watermarking.

The key of content-based video copy detection is to extract the robust and discriminative features from video frames. Many previous methods [9, 21, 12, 5] extracted the local feature such as SIFT [11] to deal with occlusions and cropping problem. However, it is quite computational expensive to extract the robust keypoint descriptors and perform pairwise matching of a large number of local features frame by frame. Alternatively, the complicated indexing scheme is employed to reduce the total number of comparisons among the local feature descriptors. Although such method obtains some promising results, it is still computationally intensive and require a large amount of memory to store the vocabularies. Moreover, such indirect comparison method may inevitably miss some video copies.

In contrast to the local features, the global features, such as color histogram, pyramid histogram of oriented gradients (PHOG) [3], GIST [13] are efficient to compute and quite compact to store. Taking advantage of some preprocessing techniques [2], the global feature can also deal with the tough transformations, such as cropping, flipping, and picture in picture.

In this paper, we propose a novel content-based video copy detection method using the randomly projected binary features. To this end, both PHOG and GIST are encoded by a very efficient sparse random projection method [10], which greatly retains the discrimination capability of the original features. By taking advantage of the extremely fast similarity computation of binary features using Hamming distance, we present a keyframe-based copy retrieval method that exhaustively searches the copy candidates from the large video database without indexing. Moreover, an effective scoring and localization algorithm is proposed to further refine the retrieved copies and accurately locate the video segments. The experimental evaluation has been performed to show the efficacy of the proposed randomly projected binary features. The promising results in the TRECVID2011 [14] content-based copy detection task demonstrated the effectiveness of our proposed approach.

In summary, the main contributions of this paper are: (1) a novel scheme for the content-based video copy detection using binary features; (2) a random projection approach to

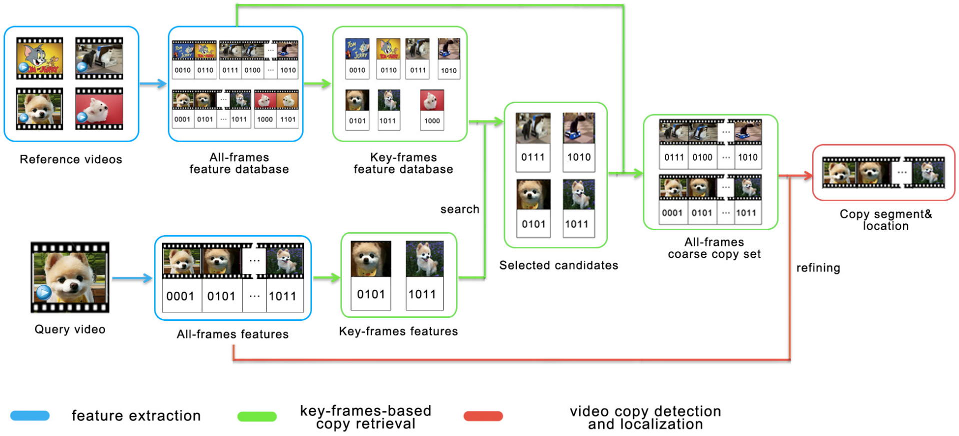


Figure 1. Overview of our proposed approach to content-based video copy detection.

encoding the binary features for GIST and PHOG; (3) an effective video copy scoring and localization method.

2. Related Work

As content-based video copy detection is beneficial for many real-world applications, there are numerous research efforts on this topic [9, 7, 12, 5, 16].

A large number of research tasks [9, 7, 12, 5] try to find the video copies based on the local salient points and local fingerprints using the complex indexing scheme. Joly et al. [7] proposed an approximate similarity search technique, in which the probabilistic selection of the feature space regions was based on the distribution of the feature distortions. Then, a post-processing technique considering only the geometrically consistent matches was used to enhance the video copy retrieval performance. Law-To et al. [9] introduced an indexing method through building trajectories of local points in the video sequence. A scale and rotation invariant local descriptor for corner points was proposed based on a generalized Radon transform [12]. Douze et al. [5] estimated a spatio-temporal model between the query video and the potentially corresponding video segments. In [16], a multiple feature hashing method is presented to retrieve the near-duplicate videos.

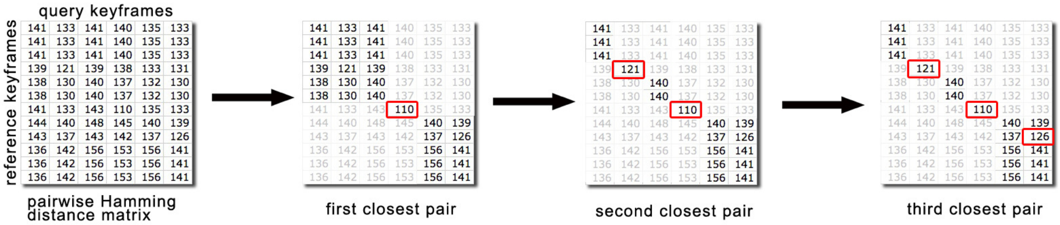
A number of testbeds have been built for evaluating the performance on video copy detection. The content-based copy detection competition in TREC Video Retrieval Evaluation (TRECVID) [15] is one of the most important benchmarking platform, in which many techniques have been evaluated. Here we briefly review several typical methods. Juan et al. [2] produced the copy candidates by comparing video segments using the combination of the visual global features. Yusuke et al. [17] employed a bag-of-global visual features using the DCT-sign-based feature. They performed multiple assignment in the temporal, feature and spatial domain, then adopted inverse document frequency weighting and temporal burstiness-aware scoring to emphasize distinctive visual words. Jiang et al. [6] detected video copies with a cascade of multi-modal features and a temporal pyramid matching method. Note that all these methods depend on the audio content, we focus on the method using visual information only in this paper.

3. Video Copy Detection with Binary Features

Given a database of the *reference videos*, and a set of *query videos* are generated by applying some transformations on the corresponding reference videos. Our task is to detect the correct copy or claim no copy can be found for each query video from the reference video database using the content information. To this end, we propose a novel approach which employs two kinds of randomly projected binary features to represent each frame. As illustrated in Fig. 1, the whole process can be divided into three steps: 1) feature extraction; 2) keyframe-based copy retrieval; 3) video copy detection and localization. In the following, we introduce each step in detail.

3.1. Feature Extraction

Typically, it is very efficient to extract the global features from the image. We further extract the binarized global features from each frame in videos, which can greatly reduce the computational cost and storage requirement. In this paper, we introduce two kinds of binary features by encoding two conventional global features: Pyramid Histogram of Oriented Gradients (PHOG) [3] and GIST [13]. We name the binary PHOG feature as "BPHOG", and name Binary GIST feature as "BGIST", respectively. In our implementation, we first detect the copies using the two kinds of binary features separately, and then simply combine the results by selecting the ones with the higher confidence score. Some video preprocessing techniques are also used before the feature extraction to deal with some tough transformations including black borders, irrelevant frames, flip, and picture-in-picture, which can substantially improve the



141	133	141	140	135	133
141	133	141	140	135	133
141	133	141	140	135	133
139	121	139	138	133	131
138	130	140	137	132	130
138	130	140	137	132	130
141	133	143	110	135	133
144	140	148	145	140	139
143	137	143	142	137	126
136	142	156	153	156	141
136	142	156	153	156	141
136	142	156	153	156	141

Figure 2. Find three representative keyframe pairs using pairwise Hamming distance.

performance of the global feature representation. For first three transformations, we employ the similar idea of inverse transformation described in [1]. To effectively detect the picture-in-picture transformations, we detect the right angle corners of the front picture using the position constraints.

3.1.1 Sparse Random Projection

We target to project original features into an m-dimensional space such that the pairwise Euclidean distance is preserved. It turned out that an ϵ-approximate embedding can be found using a very sparse random projection matrix whose entries consist of $\{\sqrt{s}, 0, -\sqrt{s}\}$ with probabilities $\{0.5/s, 1-1/s, 0.5/s\}$ when $s \geq 3$ [10]. Then, we can binarize each projection into single bit according to its sign. One bit quantization of the projected vectors approximates the angle between the original vectors. We empirically set s to $d/2$, where d is the dimensionality of its original feature. Thus, each row of the projection matrix consists of single 1 and one -1 in our setting. The remaining elements are all set to zeros. In other words, we compare two randomly selected item for each dimension from the original feature to obtain its corresponding binary feature. If the previous one is larger, the bit is set to one, otherwise zero.

3.1.2 Binary PHOG

PHOG feature consists of a histogram of orientation gradients (HOG) over each subregion in image at each resolution level. As in [3], the distance between two PHOG image features can reflect the extent to which the images contain similar shapes and correspond in their spatial layout. Since the variations of a copy may not significantly change the distribution of intensity gradients, we propose an effective Binary PHOG feature representation for each frame.

To efficiently extract the BPHOG feature, we directly represent each subregion by binarizing the HOG descriptor [4] without detecting the edges as the original PHOG implementation [3]. More specifically, we firstly compute the gradient (g_x, g_y) for each pixel (x, y) in the subregion, and then calculate the gradient magnitude G_m and orientation bin G_b as $G_m(x, y) = \|(g_x, g_y)\|, G_b(x, y) = \lfloor N_b \arctan(g_y/g_x)/\pi + (1/2)N_b \rfloor$, where N_b is the total number of the orientation bins. The k-th dimensional of the HOG descriptor is computed as: $HOG(k) = \sum_{(} x, y)G_m(x, y), G_b(x, y) = k$.

Secondly, N_b-dimensional feature vector is projected onto an m-dimensional space by an $m \times N_b$ projection matrix, and each projection is binarized into one bit by its sign. Finally, we extract BPHOG feature for each frame by concatenating all binarized HOG codes from each subregion.

3.1.3 Binary GIST

GIST feature is a set of perceptual dimensions (naturalness, openness, roughness, expansion and ruggedness) that represent the dominant spatial structure of a scene [13]. It has been widely used to represent the image for classification, copy detection, and object recognition. Similarly, we binarize the extracted GIST features using the random projection method introduced in Section 3.1.1. Comparing to binarizing HOG feature in each subregion, we quantize the GIST feature in each block into the binary codes.

3.2. Keyframe-based Copy Retrieval

To retrieve the relevant reference videos for a query, in this step, we employ an exhaustive brute-force search by taking advantage of fast Hamming distance computation between binary features. To further reduce the computational cost, we extract keyframes to efficiently measure the similarity between videos. Specifically, we first compute the Hamming distance between the binary features for each pair of neighbor frames. The largest $\phi - 1$ distances are selected as the segment boundaries. Then, the video is segmented into ϕ segments. The first frame and the last frame of each segments are selected as the keyframes.

To measure the similarity between two videos using the keyframes, we select three representative keyframe pairs one by one with the scheme illustrated in Fig. 2. The closest pair is selected at first. Then we mask the keyframe pairs that violate the temporal order of the current selected pair. This process is repeated until the three representative keyframe pairs are found. The keyframe-based distance between two videos is defined as the average Hamming distances of these three representative keyframe pairs. To consider both the efficiency and effectiveness, three representative keyframe pairs are sufficient to filter out most irrelevant reference videos in our empirically study. Finally, a coarse

Algorithm 1 Copy Confidence Scoring and Localization

```
Input: Extracted features for $Q$ and $R$, distance threshold $\theta_d$, minimum matched segments length $\theta_l$.
Output: Confidence Score $S$, $(Q_{first}, R_{first}, R_{last})$.
Initialize matching segments index set:
$V = \{1, \cdots, q + r - 1\}$;
Initialize the first and the last frame index vectors:
$Q_f = q + 1, Q_l = 0, R_f = r + 1, R_l = 0$;
Initialize the matched frame pairs counter $M = 0$;
Compute the pairwise Hamming distance matrix $H$;
for $i = 1$ to $q$ do
  for $j = 1$ to $r$ do
    if $H(i, j) < \theta_d$ then
      Current sequence index $k = i - j + r$;
      Update the first and the last indexes:
      if $Q_f(k) > i$ and $R_f(k) > j$ then
        $Q_f(k) = i, R_f(k) = j$;
      end if
      if $Q_l(k) < i$ and $R_l(k) < j$ then
        $Q_l(k) = i, R_l(k) = j$;
      end if
      Increase the matched frame pairs counter:
      M(k)=M(k)+1;
    end if
  end for
end for
Compute matched segments length $L = Q_l - Q_f$;
Remove the segments with the length below $\theta_l$ from $V$;
Calculate the confidence score for each segment:
$C(k) = M(k)/L(k) + L(k)/q, k \in V$;
Find the most matched segments from $V$:
$n = \arg\max_k C(k), k \in V$; $S = C(n)$
$(Q_{first}, R_{first}, R_{last}) = Time(Q_f(n), R_f(n), R_l(n))$.
return $S, (Q_{first}, R_{first}, R_{last})$
```

copy set with a few top nearest reference videos for a query video is constituted.

3.3. Video Copy Detection and Localization

Once retrieving a coarse set of video copies for a query video Q, the next step is to evaluate the copy confidence score for each reference video R in the coarse copy set. We employ a 3-tuple $(Q_{first}, R_{first}, R_{last})$ structure to localize the copy segments. They respectively denote the time of the first frame in the query, the first and the last frame in the reference video. Typically, the time shift between the found copies is supposed to be a constant value here.

Taking advantage of the efficient Hamming distance computation, we are able to compute the copy confidence score and find the exact locations of copies frame by frame. To compute the confidence score and the location of the reference video R for the query video Q, we design a search algorithm based on the pairwise Hamming distance matrix $H \in R^{q \times r}$ between all q frames in Q and all the r frames in R, as summarized in Algorithm 1.

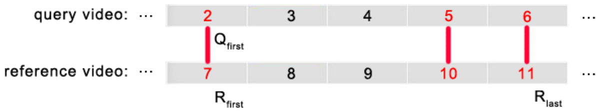


Figure 3. Example of a possible matching sequence: the 1st frame in the query video is aligned to the 6th frame in reference video.

We employ Algorithm 1 to evaluate every possible matching sequence, i.e., every diagonal of H like the sequence $(1, 6), (2, 7), \cdots, (q, q + 5)$ illustrated in Fig. 3 where $q < r$ and (i, j) denotes the indexes of the corresponding frame pair from Q and R respectively. To find the most confident segment of video copies from all these possible sequences, we employ the following rules. Two frames with the Hamming distance below a threshold θ_d are treated as a matched pair of frames. As the example shown in Fig. 3, the $(2, 7)$ is the first matched pair and $(6, 11)$ is the last matched pair. The segments between the first matched pair and the last matched pair are treated as the matched segments, such as $(2, 7), \cdots, (6, 11)$ in the example. Thus, we have the most confident matched segment for each possible matching sequence. Then, we define the confidence score for the matched segments as $m/l + l/q$, where m denotes the number of matched frame pairs, l is the length of the matched segments, and q is the length of the query video. In this example, $m = 3$, $l = 5$. In the score function, the first term awards more matched frames and the second term awards longer matched segments. Moreover, the length of the matched segments l should be longer than a threshold θ_l. Finally, the segment with the maximum confidence score is treated as the confidence score of this reference video. This process can be done by simply traversing the matrix H only once according to Algorithm 1. By repeating this process, we can compute the copy confidence score and the localization result of each reference video in the coarse copy set.

We select the video with the largest confidence score as the copy of the query video. Along with the confidence score, the localization result is obtained simultaneously. Moreover, we make use of the ratio-distance to remove the large number of false positives. Specifically, we only count the video with the score ratio between the first and the second largest ones larger than 1.3 as the true positive; otherwise, we claim that no video copy is found for the query.

4. Experiments

4.1. Feature Comparisons

We compare the performance of seven kinds of features with different coding approaches on a widely-used Columbia's TRECVID2003 dataset [20], which consists of 600 keyframes with 150 near-duplicate image pairs and 300

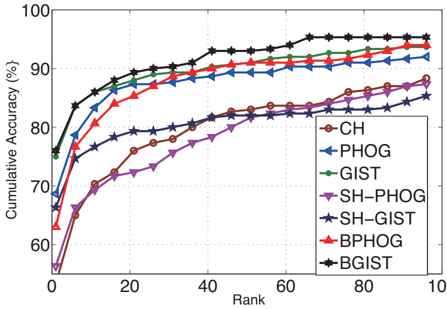


Figure 4. Comparisons on Columbia's TRECVID2003 dataset

non-duplicate images extracted from the TRECVID2003 corpus. Cumulative accuracy [22, 23] using different features by ranking with their corresponding similarity/distance measures are evaluated to compare the performance. We list the detailed information on compared methods and their settings as follows:

Color Histogram (CH) We compute the histogram for HSV channels with H:64 bins, S:64 bins and V:32 bins.

PHOG We extract the PHOG feature [3] without the edge detection, and use three levels with $4^0 + 4^1 + 4^2 = 21$ subregions and 24 orientation bins for each region.

GIST We extract the GIST feature [13] on 3 scales with $8, 8, 4$ orientations respectively. $4 \times 4 = 16$ blocks are used to account for the spatial variations.

SH-PHOG Spectral Hashing [18] is used to convert the 504-dimensional PHOG feature into 504-bit binary code. All the images are used to train the hash functions.

SH-GIST Spectral Hashing is used to project the 320-dimensional GIST feature onto 512-bit binary code.

BPHOG We binarize the HOG feature for each subregion to 24-bit leading to 504-bit BPHOG feature.

BGIST We binarize the GIST feature for each block to 32-bit leading to 512-bit BGIST feature.

We plot the cumulative accuracy results in Fig. 4 for all the compared methods on the Columbia's TRECVID2003 dataset. First of all, we can observe that BGIST achieves the best result, which even perform better than the original GIST feature. Moreover, the performance of BPHOG method is only slightly worse than the original PHOG feature. Furthermore, the binary coding using Spectral Hashing is much worse than the original features. It can also be seen that GIST method obtains the best performance among the non-binary features, and PHOG performs better than color histogram. Thus, we can conclude that our used random projection approach to binary coding not only preserves the discriminative capability of the original features but also greatly reduce the computational cost by taking advantage of Hamming distance.

4.2. TRECVID2011 CCD Task Results

The TRECVID2011 [14] content-based copy detection task requires to perform $11,256$ queries in a reference video database consisting of around 12K videos amount to about 400 hours. The query video is transformed by 8 types of visual transformation and 7 types of audio transformation from 201 videos. Since we focus on the visual contents in this paper, $8 * 201 = 1608$ visual queries are used in our evaluation. 8 types of visual transformation is summarized as follows: (T1) Simulated camcording; (T2) Picture in picture; (T3) Insertions of pattern; (T4) Strong re-encoding; (T5) Change of gamma; (T6) Decrease in quality; (T8) Post production; (T10) Combination of three randomly selected transformations chosen from T2-T5, T6 and T8.

We have submitted two runs using our proposed approach described in Section 3. One only employed the BGIST feature (bgist) and the other used the combination of BGIST and BPHOG features (bhg). Empirically, the number of keyframes is set to 10 with $\phi = 5$ and 50 with $\phi = 25$ respectively for the query and the reference video. θ_d and θ_l in Algorithm 1 are set to 170 and 15, respectively.

As defined in CCD task [8], we employ the Normalized Detection Cost Ratio (NDCR) to measure the accuracy of the copy detection. The lower NDCR, the better performance is. We also use F1 measure to evaluate the accuracy of finding the exact extent of the copy in the reference video, once the system has correctly detected a copy. The higher F1 measure, the better performance is.

Table 1 shows the NDCR and F1 measure results of our two runs. From NDCR results, it can be seen that our two runs obtain the promising copy detection performance for the most of transformations. It is worthy of mention that we only take consideration of the visual contents while most of the leading teams find the copies using both visual and audio information. Our proposed binary global features, keyframe-based copy retrieval and scoring algorithm together contribute to the desirable copy detection performance. Moreover, the proposed method never fails to a certain transformation, which shows the great generalization capability of the BPHOG and BGIST features with the presented preprocessing techniques. Respectively, the proposed approach does not perform well for the insertions of pattern (T3) because the inserted patterns may affect the the global feature representation.

Obviously, our localization algorithm shows the leading performance according to the F1 measure results for all the transformations. Traversing the Hamming distance matrix to numerate all possible matching sequence is the key to the success of the proposed method. From above all, we can conclude that the desirable results are mainly due to the fast computation for Hamming distance of the binary features.

5. Conclusion and Future Work

In this paper, we presented a novel randomly projected binary feature approach to the content-based video copy detection using visual information. We introduced two kinds

Table 1. NDCR (lower is better) and F1 measure (higher is better) results and ranks of our two runs and the median. Note that we only consider the visual content leading to the desirable results while most teams make use of both visual and audio contents.

	NDCR					F1				
	bgist	bhg	median	bgist-rank	bhg-rank	bgist	bhg	median	bgist-rank	bhg-rank
T1	0.888	0.881	93.146	9/32	8/32	0.958	0.958	0.796	1/32	1/32
T2	0.731	0.687	107.688	8/32	7/32	0.957	0.943	0.882	3/32	8/32
T3	0.649	0.470	91.737	13/32	12/32	0.952	0.958	0.903	8/32	4/32
T4	0.515	0.448	91.754	10/32	8/32	0.957	0.958	0.887	5/32	4/32
T5	0.343	0.284	107.019	9/32	7/32	0.952	0.949	0.893	5/32	8/32
T6	0.500	0.425	69.322	8/32	6/32	0.956	0.952	0.890	4/32	6/32
T8	0.731	0.590	91.786	10/32	8/32	0.959	0.949	0.885	4/32	8/32
T10	0.657	0.575	122.496	9/32	8/32	0.958	0.950	0.887	4/32	5/32
Average	0.626	0.545	107.2	8/32	7/32	0.956	0.952	0.809	2/32	4/32

of binary global features to extract the effective binary features from video frames. Accordingly, we developed a keyframe-based video copy retrieval algorithm to exhaustively search the video copy candidates and a scoring and localization algorithm to refine the search results. The extensive experiments demonstrated the effectiveness of our proposed copy detection method. In the future, we will try to apply the semi-supervised hashing scheme [19] to effectively extract the binary features.

Acknowledgments

The work was fully supported by National Natural Science Foundation of China under Grants (61103105 and 91120302) and Fundamental Research Funds for the Central Universities.

References

[1] J. M. Barrios and B. Bustos. Content-based video copy detection: Prisma at trecvid 2010. In *Proc. TRECVID 2010*, 2010.

[2] J. M. Barrios, B. Bustos, and X. Anguera. Combining features at search time: Prisma at video copy detection task. In *Proc. TRECVID*, 2011.

[3] A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. In *CIVR*, 2007.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[5] M. Douze, H. Jegou, and C. Schmid. An image-based approach to video copy detection with spatio-temporal post-filtering. *Multimedia, IEEE Transactions on*, 2010.

[6] M. Jiang, S. Fang, Y. Tian, T. Huang, and W. Gao. Pku-idm @ trecvid 2011 cbcd: Content-based copy detection with cascade of multimodal features and temporal pyramid matching. In *Proc. TRECVID*, 2011.

[7] A. Joly, O. Buisson, and C. Frelicot. Content-based copy retrieval using distortion-based probabilistic similarity search. *Multimedia, IEEE Transactions on*, 2007.

[8] W. Kraaij, P. Over, J. Fiscus, and A. Joly. Cbcd evaluation plan. In *Proc. TRECVID*, 2011.

[9] J. Law-To, O. Buisson, V. Gouet-Brunet, and N. Boujemaa. Robust voting algorithm based on labels of behavior for video copy detection. In *ACM Multimedia*, 2006.

[10] P. Li, T. J. Hastie, and K. W. Church. Very sparse random projections. In *KDD*, 2006.

[11] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.

[12] E. Maani, S. Tsaftaris, and A. Katsaggelos. Local feature extraction for video copy detection in a database. In *ICIP*, 2008.

[13] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 2001.

[14] P. Over, G. Awad, M. Michel, J. Fiscus, W. Kraaij, and A. F. Smeaton. Trecvid 2011 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *TRECVID*, 2011.

[15] A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and trecvid. In *MIR*, 2006.

[16] J. Song, Y. Yang, Z. Huang, H. T. Shen, and R. Hong. Multiple feature hashing for real-time large scale near-duplicate video retrieval. In *ACM Multimedia*, 2011.

[17] Y. Uchida, K. Takagi, and S. Sakazawa. Kddi labs at trecvid 2011: Content-based copy detection. In *TRECVID*, 2011.

[18] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In *NIPS*, 2008.

[19] C. Wu, J. Zhu, D. Cai, C. Chen, and J. Bu. Semi-supervised nonlinear hashing using bootstrap sequential projection learning. *IEEE Transactions on Knowledge and Data Engineering*, 99(PrePrints), 2012.

[20] D.-Q. Zhang and S.-F. Chang. Detecting image near-duplicate by stochastic attributed relational graph matching with learning. In *ACM Multimedia*, 2004.

[21] W.-L. Zhao, C.-W. Ngo, H.-K. Tan, and X. Wu. Near-duplicate keyframe identification with interest point matching and pattern learning. *Multimedia, IEEE Trans. on*, 2007.

[22] J. Zhu, S. C. Hoi, M. R. Lyu, and S. Yan. Near-duplicate keyframe retrieval by nonrigid image matching. In *ACM Multimedia*, 2008.

[23] J. Zhu, S. C. H. Hoi, M. R. Lyu, and S. Yan. Near-duplicate keyframe retrieval by semi-supervised learning and nonrigid image matching. *ACM Trans. Multimedia Comput. Commun. Appl.*, 7(1):4:1–4:24, Feb. 2011.