National Institute of Informatics, Japan at TRECVID 2009

Duy-Dinh Le [1], Sebastien Poullot [1,2], Michel Crucianu [2], Xiaomeng Wu [1]
Michael Nett [1,3], Michael E. Houle [1], Shin'ichi Satoh [1]

[1] *National Institute of Informatics*
2-1-2 Hitotsubashi, Chiyoda-ku, Japan 101-8430
[2] *CEDRIC - CNAM*
292 rue St Martin
75141 Paris cedex 03, France
[3] *Faculty of Mathematics, Computer Science and Natural Sciences*
The RWTH Aachen University
D - 52056 Aachen, Germany
ledduy,wxmeng,meh,satoh@nii.ac.jp,
poullot.sebastien@free.fr, michel.crucianu@cnam.fr
michael.nett@rwth-aachen.de

***Abstract*—This paper reports our experiments for TRECVID 2009 tasks: high level feature extraction, search and content-based copy detection. For the high level feature extraction task, we used the baseline features such as color moments, edge orientation histogram, local binary patterns and local features trained with SVM classifiers and nearest neighbor classifiers. For the search task, we used . Concerning content based video copy detection (CBVCD), using local features leads to good robustness to most types of photometric or geometric transformations. However, to achieve both good precision and good recall when the transformations are strong, especially occlusions, feature *configurations* should be taken into account. This usually leads to complex matching operations that are incompatible with scalable copy detection. We suggest a computationally inexpensive solution for including a minimal amount of configuration information that significantly improves the balance between overall detection quality and scalability.**

I. High Level Feature Extraction

A. Method Overview

In our framework, features are extracted from the input keyframe images representing for shots. We extracted five keyframes per shot that are spaced out equally within the provided shot boundary. In the training stage, we used these features to learn SVM classifiers and nearest neighbor classifiers. These classifiers were then used to compute the raw output scores for each test image in the testing stage. These output scores were further fused by taking the average for computing the final output score. In order to return K shots most relevant for one concept query that then are evaluated and compared in TRECVID benchmark, all normalized final output scores of shots are sorted in descending order and top K shots are returned. In the case of a shot consisting of several sub-shots, only the maximum score among subshots' scores is used for that shot.

As for feature extraction, we used types of global features color moments, color histogram, edge direction histogram and local binary patterns; and local features using SIFT (c.f. Table I). These features are extracted from a nxn grid of the input image, normalized to zero mean and unit standard deviation and then stored for training and testing. Specifically, the normalized vector $x^{norm} = (x_1^{norm}, x_2^{norm}, ..., x_N^{norm})$ of an input raw vector $x^{raw} = (x_1^{raw}, x_2^{raw}, ..., x_N^{raw})$ is defined as follows:

$$x_i^{norm} = \frac{(x_i^{raw} - \mu)}{\sigma}$$

where x_i^{norm} and x_i^{raw} is the i-th element of the feature vectors x^{norm} and x^{raw} respectively, N is the number of dimensions. μ is the mean $\mu_i = \frac{1}{N}\sum_{i=1}^{N} x_i^{raw}$ and σ is the standard deviation

$$\sigma_i = \sqrt{(\frac{1}{N}\sum_{i=1}^{N} x_i^{raw} - \mu_i)^2}$$

The number of positive shots for each concept is small, so we enlarged the positive set by extracting 5 keyframes from each positive shot and manually revised annotation of these keyframes. As for the concepts used in the TRECVID 2008, we used the annotations shared by LIG and ICT-CAS. As for the new concepts of TRECVID 2009, we only used 100 positive shots per concept.

LibSVM [1] is used to train SVM classifiers with RBF kernel. The optimal (C, g) parameters are found by conducting a grid search with 5-fold cross validation on a subset of 1,500 samples stratified selected from the original dataset.

As for nearest neighbor classifiers, firstly, we construct a SASH [1] indexing structure on top of the set of all training examples. The SASH takes a test item as input and retrieves k training examples, that are considered similar in terms of

[1] http://www.csie.ntu.edu.tw/∼cjlin/libsvm

Feature	Description
nsc.cCV_MIXED.g3.qx.g_lbp20+g_cm3 +g3.qx.g_eoh36+g3.qx.g_ch16	Early fusion of features such as color moments, local binary patterns, edge orientation histograms, and color histograms
nsc.cCV_MIXED.g3.qx.g_lbp20+g_cm3+g3.qx.g_eoh36	Early fusion of features such as color moments, local binary patterns, and edge orientation histograms
nsc.cCV_MIXED.g3.qx.g_lbp20+g_cm3	Early fusion of features such as color moments, and local binary patterns extracted from 3x3 grid image
nsc.cCV_MIXED.g5.qx.g_lbp30+g_cm3	Early fusion of features such as color moments and local binary patterns extracted from 5x5 grid image
nsc.cCV_GRAY.g5.q30.g_lbp	Local binary patterns extracted from 5x5 grid image and quantized into 30 bins per region
nsc.cCV_HSV.g5.q3.g_cm	Color moments extracted from 5x5 grid image in HSV color space
nsc.cCV_RGB.g5.q3.g_cm	Color moments extracted from 5x5 grid image in RGB color space
nsc.DoG.sift.part-1000.M1000.m50.bow	Bow model using SIFT and tf for weighting. The keypoint detector provided by D. Lowe was used to extract keypoints in images

TABLE I
THE FEATURES USED FOR THE HLF TASK.

the Euclidean distance of the item's feature vectors, in time $\mathcal{O}(k + \log n)$.

Given a test item we retrieve 20 similar items from the SASH. In order to cope with noise we derive a class label by a majority vote, that is, the class label observed most often among those 20 near neighbors is elected.

Since the fraction of positive and negative examples in the 20 retrieved items only provides a very coarse weighting, we use the distance Δ between the test item and the closest training example (from among the 20 examples retrieved from the SASH) to determine our vote's certainty: If we vote for a positive example, we apply the weight $1/\Delta$, otherwise $-1/\Delta$.

B. Result

We submitted 6 runs and the results are shown in Table II. There are two type a runs among the 6 runs that only use TRECVID 2005 dataset. Our best run is NII.SECODE.R1 (MAP 0.110). However, the run using only two features of color moments and local binary patterns can achieve 0.096 MAP. In addition, our trial on using nearest neighbor classifiers is bad. It concludes that K-NN is not suitable for this task.

II. SEARCH

A. Method Overview

We used the following methods to return shots for each query:

- Visual search using SVM: We trained concept detectors corresponding to queries. In other words, each query is served as one concept. We used the same approach with the HLF task to train concepts where example keyframes for each query (5 keyframes were extracted for each example shot) are positive samples, and keyframes picked randomly from the training set of the HLF task are negative samples. The trained concept detectors were then used to predict test keyframes and top 1,000 shots were selected based on the prediction score.

- Visual search using KNN: We used KNN described in the HLF task to rank keyframes.

- Concept selection using visual feature: We used 30 concept detectors of the HLF task in 2008 and 2009, and did prediction on query video examples. The prediction scores for each keyframe that are above the threshold of 0.5 were selected and the concepts corresponding to these scores were selected. The fused detection scores of these concept detectors on test shots were used to rank these shots. For example, using our method, as for query 270 'Find shots of a crowd of people, outdoors, filling more than half of the frame area', there are two concepts detected in video examples that are Demonstration_Or_Protest and Hand. For each keyframe in the test set, we fused the prediction scores of the two concepts and used it for ranking the keyframes.

- Concept selection using textual description: We used the textual description of each query and that of the 30 concepts described above for matching. After obtaining the candidate concepts, the same method with concept selection using visual feature was used to rank keyframes in the test set.

As for high precision runs (P run), we simply picked top 10 shots returned by the corresponding N-run.

B. Result

We submitted 10 runs (fully automatic) and the results are shown in III. Our best run (F_A_N_NII.SEVIS.R1) achieves MAP 0.065. The run based on concept selection using visual feature(F_A_N_NII.SEVIS.R3) is effective with MAP 0.050. The performances of runs (F_A_N_NII.SEVIS.R5 and F_A_N_NII.SEVIS.R6) using example keyframes to train detectors are not good. It indicates that pre-trained concept detectors are good for the search task.

III. CONTENT-BASED COPY DETECTION

A. Introduction

Experience with reused video content shows that the most frequent transformations concern gamma and contrast changes, scaling, cropping, blurring, compression artifacts and video

RunID	Description	MAP
A_NII.SECODE.R1	Fusion of 8 features: 7 global features (color moments, local binary patterns, color histogram and edge orientation histogram and early fusion of these features) and 1 local features	0.110
A_NII.SECODE.R2	Fusion of 5 global features derived from color moments and local binary patterns only	0.096
A_NII.SECODE.R3	Local feature using DoG+SIFT as keypoint detector+descriptor. The vocabulary is formed by using RSC clustering with 725 clusters. BoW model uses tf for weighting	0.040
a_NII.SECODE.R4	Fusion of 8 features as A_NII.SECODE.R1. However the training set is different. It used TV2005 dataset (US news video)	0.041
a_NII.SECODE.R5	Fusion of 5 global features as A_NII.SECODE.R2. However the training set is different. It used TV2005 dataset (US news video)	0.040
A_NII.SECODE.R6	KNN-based classifier using early fusion of features such as color moments, local binary patterns, edge orientation histogram	0.013

TABLE II
THE PERFORMANCE OF NII'S RUNS FOR THE HLF TASK.

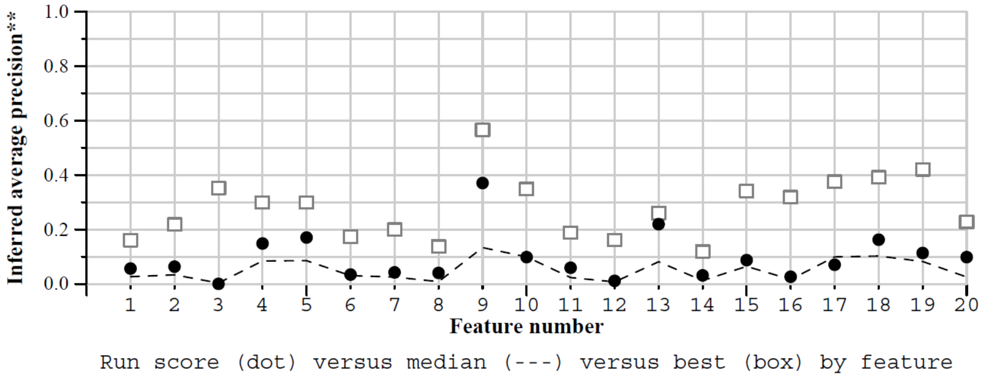

Fig. 1. The performance of the run using color moments and local binary patterns.

RunID	Description	MAP
F_A_N_NII.SEVIS.R1	Fusion of the methods such as visual search using SVM, concept selection using visual feature, and concept selection using textual description	0.065
F_A_N_NII.SEVIS.R2	Fusion of the methods such as visual search using SVM, and concept selection using visual feature	0.048
F_A_N_NII.SEVIS.R3	Concept selection using visual feature	0.050
F_A_N_NII.SEVIS.R4	Fusion of the methods such as concept selection using visual feature, and concept selection using textual description	0.051
F_A_N_NII.SEVIS.R5	Visual search using SVM	0.032
F_A_N_NII.SEVIS.R6	Visual search using KNN	0.003
F_A_P_NII.SEVIS.R7	Fusion of the methods such as visual search using SVM, concept selection using visual feature, and concept selection using textual description	0.22
F_A_P_NII.SEVIS.R8	Fusion of the methods such as visual search using SVM, and concept selection using visual feature	0.13
F_A_P_NII.SEVIS.R9	Concept selection using visual feature	0.16
F_A_P_NII.SEVIS.R10	Fusion of the methods such as concept selection using visual feature, and concept selection using textual description	0.14

TABLE III
THE PERFORMANCE OF NII'S RUNS FOR THE SEARCH TASK.

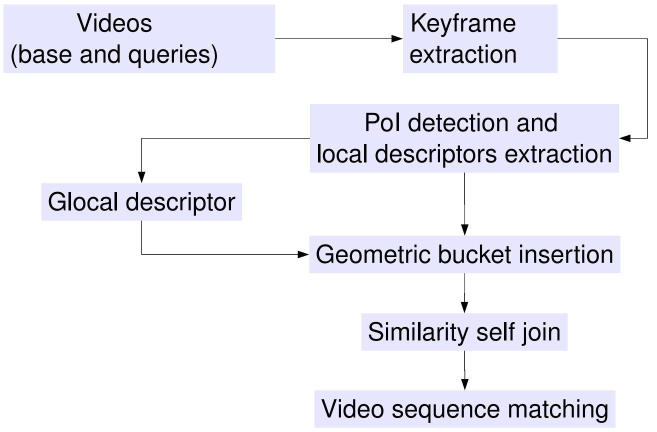

Fig. 2. Steps for database construction and self query.

inlays (logos, frames, text). Trecvid contest also includes noise addition, change of ratio, and picture in picture transformations. In Trecvid the amplitude of the transformations is large, and some may be seen as extrem. Concerning TV09 CBVCD we aim to propose a solution offering a good balance between accuracy and computation costs so as to be scalable.

Note that we are using a framework for video database mining. We construct a database using descriptors from reference database and queries then a similarity self join is performed. However we here added a constraint on the ID of the videos in order to make impossible any match between two videos from the reference database or between two videos from the queries. Figure 2 sums up the processings used for TV09 CBVCD. The different steps are detailed in the following sections.

B. Video Copy Detection

1) Features extraction: For obtaining a good accuracy (precision and recall) we have prefered local descriptions, which have shown the best results in many papers. However this choice has basically many drawbacks concerning time consumption and so scalability. First, the extraction of the descriptors is usually very expensive. For performing it faster we have chosen to use only keyframes (about 3000 per hour), which implies a keyframe detector. We also have limited the number of extracted descriptors to 150 per keyframe.

Concerning the detector of Point of Interest (PoI) we have used the Improved Harris Corner Detector [13]. It runs faster than the DoG detector or the Hessian detector. Then we compute the local descriptor from [11] at these positions. It is a compact descriptor, only 20 dimensions. Its computation is far faster than SIFT. However note that this descriptor is not invariant to scale and so we did not expect to find any copies based on transformation 2 (Picture in picture type 1), what is confirmed by our results.

This first set of choices allows to make scalability possible. The whole extraction process needs 1/20 of real time with a 2.4Ghz simple core and 1/35 using the two cores.

However indexing of the local descriptors is also necessary for performing querying the database in a reasonnable time.

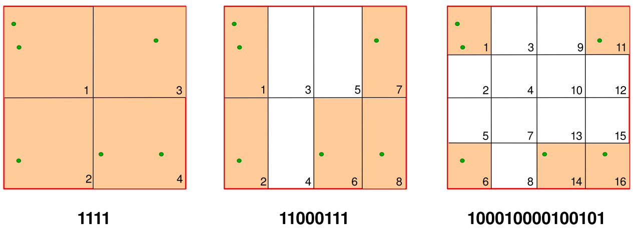

Fig. 3. Glocal signatures for a set of 6 local features with 3 different quantizations (at a depth of 2, 3 and 4) of a 2D description space.

Indexing methods for local descriptors, such as [7], [11] could have been used. But to further improve the scalability of the method, we have chosen to use a frame descriptor that embeds the local descriptions. This description relies on a quantization of the description space and is described in next section. It is quite close to the well known bag of features []. Such a description allows to simplify the voting process needed with local descriptors after querying, thus reducing the computation costs.

2) The Glocal description: We start by briefly presenting the description and indexing method put forward in [12], that serves as basis for our approach. The detection of the video sequences that occur more than once (with various modifications) in a video database begins with the extraction of keyframes from all the videos, using an algorithm finding the maxima of the global intensity of motion (leading, on average, to 1 keyframe / second). Then, a similarity self-join operation is performed on the set of keyframe descriptions, based on a specific indexing method. Eventually, these links between individual keyframes allow to find the matching video sequences.

Instead of directly using the set of signatures (descriptions) of the local features extracted from a frame, in [12] this set is first embedded into a fixed-length binary vector. The embedding procedure is: (i) given the local features of a set of frames, the description space (not the image plane) is adaptively partitioned at a limited depth h, which produces 2^h cells that are numbered according to some consistent rule (see Fig. 3); (ii) for each frame, its Glocal signature is the binary vector where the bit i is set to 1 only if the description (signature) of at least one local feature of the frame falls within cell i.

The Dice coefficient was employed to measure similarity between Glocal signatures, $S_{\text{Dice}}(\mathbf{g}_1, \mathbf{g}_2) = \frac{2\,|\mathcal{G}_1 \cap \mathcal{G}_2|}{|\mathcal{G}_1|+|\mathcal{G}_2|}$, where $\mathcal{G}_i$ is the set of bits set to 1 in the signature $\mathbf{g}_i$ and $|\cdot|$ denotes set cardinality.

For the similarity self-join operation, the database of Glocal signatures is divided into overlapping *buckets* (stored as inverted lists) such that, in each bucket, any two signatures are sufficiently similar. A self-join is then independently performed within each bucket. Following [12], a bucket is defined by a specific set of 3 bits that are set to 1 in at least one Glocal signature in the database. Every signature has several bits set to 1 (depending on the number of local features). Each

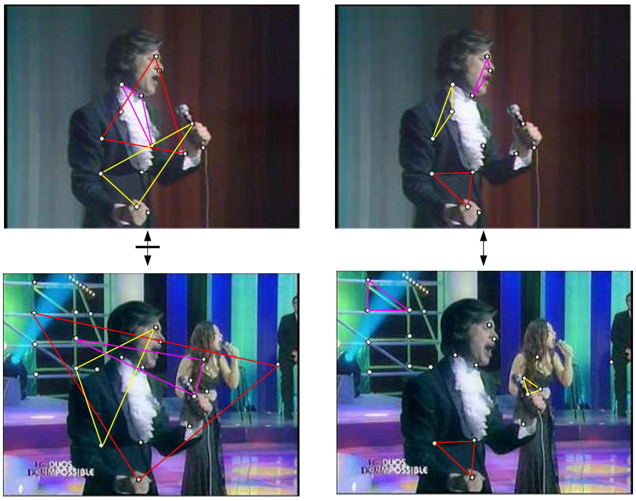

Fig. 4. Feature triplets selected in the original frame (top line) and in the copy (bottom line), with the previous rules (left) or with the new locality constraint (right).

signature can be stored in all the buckets that are defined by all the combinations of 3 bits set to 1 in the signature. This produces a redundant index. Next section describes how the combinations are selected.

To find the pairs of similar keyframes, the similarities (Dice coefficients) between Glocal signatures are computed within each bucket; if the similarity is above a decision threshold θ, the identifiers of both keyframes are stored as a link. Here we ser $\theta = 0.1$ for the BALANCED run and $\theta = 0.3$ for the NOFA run. At the end of this self-join operation, all the resulting pairs of connected keyframes are eventually used for recovering the matching video sequences.

3) Locality-based bucket definition: Some of the triplets selected by these rules are represented by triangles on the left side of Fig. 4, for an original keyframe (top) and for a copy (bottom) where the video inlay replaced a large part of the frame. It can be seen that the triplets link local features that are quite distant in the image plane and are unlikely to be preserved by strong cropping or video inlays. Since such transformations alter the longer-range structure of the frame but maintain part of the short-range structure, we here take into account *locality* in the image plane when selecting the triplets that define the buckets where the Glocal signature of the frame is indexed. The impact of the locality constraint is obvious when comparing with the left side of the same figure.

The selection and indexing procedure is: for each local feature f_i in the frame, (i) find its 10 nearest neighbors (10NN) in the *image plane*; (ii) the first triplet consists of f_i together with its 2NN and the corresponding bucket is identified by the numbers of the cells in *description space* where f_i and its 2NN are found; (iii) the second triplet consists of f_i together with its 3rd and 4th nearest neighbors, and so on, defining then 5 triplets for f_i; (iv) store (or index) the Glocal signature of the frame into these three buckets. Two local features f_i anf f_j may share a triplet, if so, one or the other is withdrawn (they are identical). This selection rule thus exploits *both* the positions of the features in description space and their neighborhood in the image plane.

For the locality constraint to be meaningful and in order to cover well all the small salient areas of a frame, the number of local features considered in the frame should be high enough. But an increase in the number of local features considered has an impact on the time and space complexity of the CBVCD-based mining operations. Indeed, the number of buckets necessarily increases with the number of local features per frame. Also, having more features per frame may require a finer partitioning of the description space, which implies longer Glocal signatures that take more space and require more time for computing Dice coefficients. At the same time, the length of the individual buckets is likely to diminish. After various experiments we have set $L = 150$ the maximum number of local features used. Quality improvements from $L = 100$ to $L = 150$ was significant, but no more from $L = 150$ to $L = 200$.

4) Use of simple configuration information: Locality constraints reinforce robustness of the indexing scheme to transformations that alter the longer-range structure of the frame while keeping part of the short-range structure. Additional local geometric information should improve discrimination power and thus allow to reach both better detection precision and better recall.

A bucket is identified by using two neighbors (among the 10NN) of a local feature f_i in the frame. It is then natural to associate in that bucket, to the Glocal signature of the frame, data describing the relations between the feature f_i and the two neighbors. The data we add is the ratio between the shortest side and the longest side of the triangle formed in the image plane by the feature f_i and the two neighbors considered. This simple information is robust to translation, rotation and (isotropic) scaling, but not to more general affine transforms like scaling with very different ratios in two different directions (anisotropic scaling). An equivalent choice would have been the angle $\text{neighbor}_1\widehat{f_i}\text{neighbor}_2$, but the computation of the length ratio is less expensive. Since this information only considers the positions of the local features in the image plane and not their individual descriptions, it can be employed even with local descriptions that do not include any orientation information. Also, it is only dependent on the robustness of the local feature detector and not on the robustness of the feature description.

According to our indexing scheme, the Glocal signature of a frame is stored in every bucket selected by the locality-based rule, together with the ratio between the shortest side and the longest side of the triangle between the local features identifying that bucket. This is shown in Fig. 5. A similarity self-join is then performed in each bucket independently of the other buckets. This operation now involves a joint condition, including both the similarity between the Glocal signatures and the similarity between their corresponding ratio data. The threshold θ_r on the difference between ratios is given by the expected error of the local feature detector and by the required

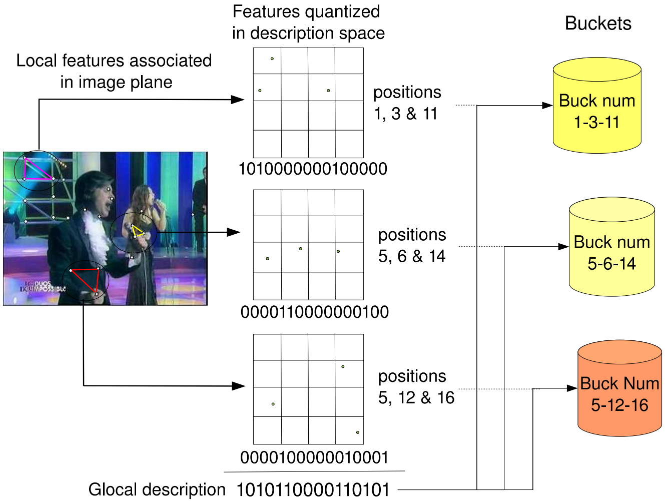

Fig. 5. Bucket selection for a frame signature, using both feature location in description space and locality constraints in image plane.

robustness to anisotropic scaling. This ratio information can be stored in low precision. We here used $\theta_r = 0.1$ both for the BALANCED and NOFA runs.

The threshold θ_s on the Dice coefficient above which two Glocal signatures are considered to match is the decision threshold and has a key role in defining the balance between precision and recall. Two keyframes are considered to be in "copy" relation if their Glocal signatures collide in at least one bucket, their Dice coefficient is above θ_s and the difference between ratios in that bucket is lower than θ_r. Actually, the ratios are compared *first* and then, if their difference is $< \theta_r$, the Dice coefficient between the two Glocal signatures is computed. Since the comparison of two small precision numbers is much less expensive than the computation of the Dice coefficient between the two Glocal signatures (especially for long signatures), this pre-filtering using simple local configuration information actually saves significant computation time, at the expense of little additional space.

5) Video Sequences Matching: Finally, we obtain a set of pairs of matching keyframes between all couples of videos. For each couples we then look for some time consistency between these pairs using the time codes. We here have a set of contraints. First, a copy sequence must contain at least C_L pairs of keyframes ($C_L = 2$ for the BALANCED run and $C_L = 3$ for the NOFA run). Second, two following matching pairs must appear within C_T seconds ($C_T = 5$ for the BALANCED run and $C_T = 3$ for the NOFA run). Third, there can be an time offset between two following pairs but it must be shorter than C_O seconds ($C_0 = 0.4$ for the BALANCED run and $C_0 = 0.2$ for the NOFA run). The copy sequences respecting these constraints are kept.

6) Results: The following graphs show our results (Fig. 6). Dots represents our run, squares the better runs and dashes the median score. Concerning the BALANCED run, our result are quite satisfying, always better than median one, and sometimes not so far from the best one. The mean processing times are almost the better ones. The NOFA run is less interesting, the parameter set seems to strict but however let some false alarms occured, which strongly penalyzes the score. Processing times are still the same.

C. Video + Audio Copy Detection

The audio-only copy detection scheme is based on the fingerprint extraction method proposed by Haitsma [14]. The confidence score is defined by $1 - BER$, where BER denotes the Bit Error Rate between two audio segments. We fuse the video-only and audio-only copy detection results at decision level. We tried two operators, one is **AND** and the other is **OR**. In the case of **AND**, a pair of two segments is determined as a copy if and only if it is detected by both video-only and audio-only methods; in the case of OR, a pair of two segments is determined as a copy if it is detected by either video-only or audio-only methods. For the fusion of the confidence score, we used the weighted average method. We assume that the confidence of the audio-only method is higher than that of the video-only one. Therefore, a higher weight (0.65) is associated to the confidence score of the audio-only results. The weight of the video-only results is thus $1 - 0.65 = 0.35$.

We have two types of video-only results, which can be respectively denoted by **BALANCED-Video** and **NOFA-Video**. For the audio-only method, we have only one result list. Then we used two operators. Thus, we have four runs, which are respectively denoted by **BA**, **BO**, **NA**, and **NO**. The explanation of these four runs is as follows.

BA: **BALANCED-Video** result and **AND** operator are used
BO: **BALANCED-Video** result and **OR** operator are used
NA: **NOFA-Video** result and **AND** operator are used
NO: **NOFA-Video** result and **OR** operator are used

We submitted all of these four runs for both **BALANCED** and **NOFA** application profiles. In other words, we finally have eight runs. As we expected, **BO** produced the lowest Min_NDCR for both **BALANCED** and **AND** application profiles (Fig. 7). This is because, the number of false alarms is very small for both **BALANCED-Video** and **NOFA-Video** results, while the number of misses of **NOFA-Video** results is much larger than that of **BALANCED-Video** ones. On the other hand, the usage of **AND** operator rejected many correct results, so the runs using **OR** operator produced lower Min_NDCR than those using **AND** operator.

D. Conclusion

TRECVID CBVCD task includes several transformations with various strenght. Since transformations like strong cropping and video inlays alter the longer-range structure of the frame but maintain part of the short-range structure, we suggest to take into account *locality* in the image plane when indexing the video keyframes. We further include in the indexing and matching processes simple local geometric data, involving the nearest neighbors of a feature in the image plane. This data is selected to be as robust as possible to the most common types of image transformations. This method

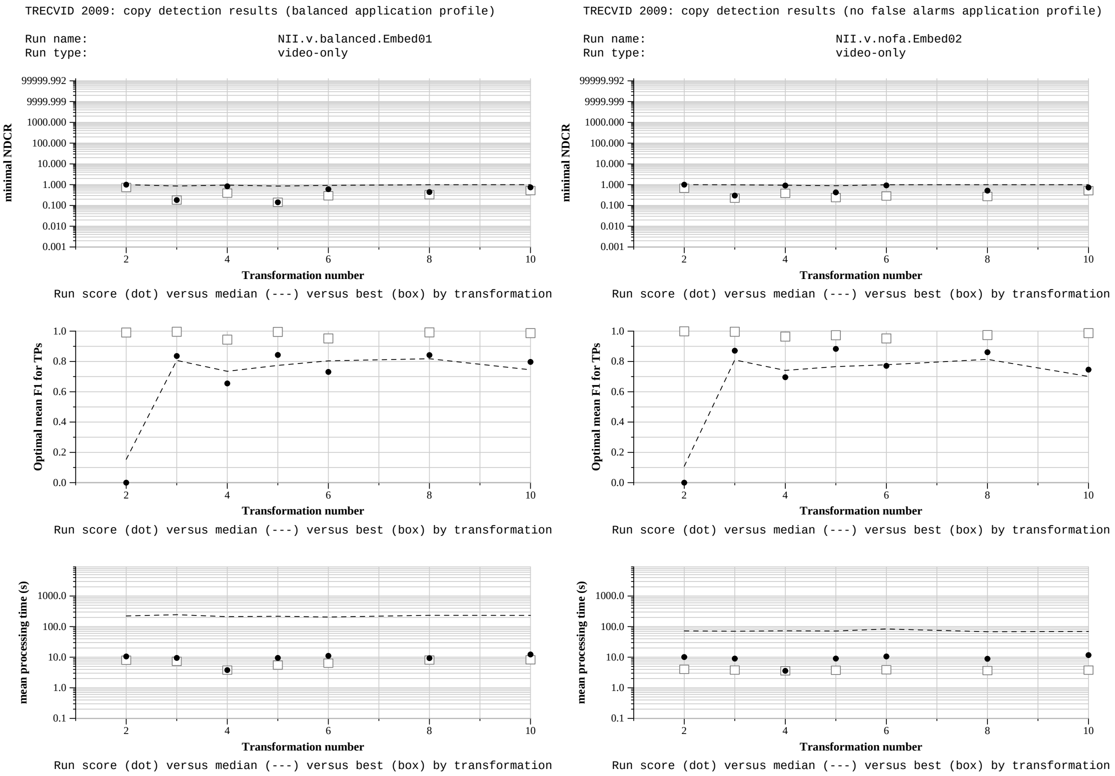

Fig. 6. BALANCED (left) and NOFA (right) runs of video copy detection

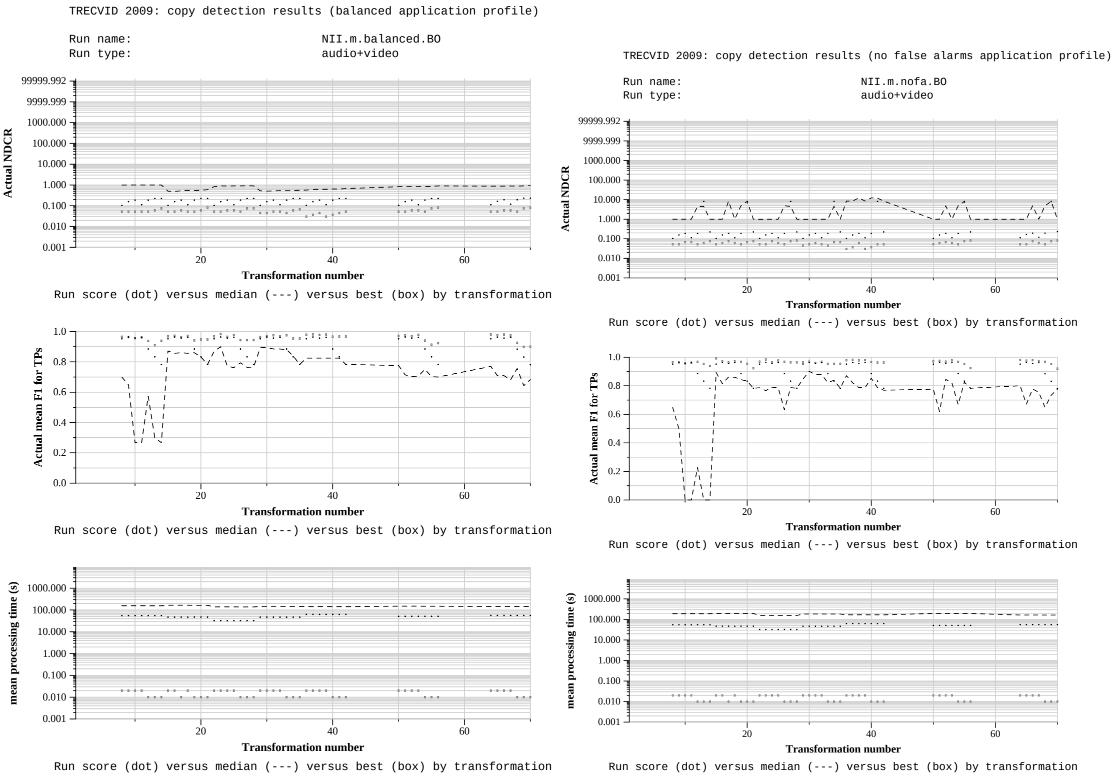

Fig. 7. BALANCED (left) and NOFA (right) runs of video + audio copy detection (BO)

give a good balance between quality of the results, both for precision and recall, and is a scalable solution.

References

[1] M. E. Houle and J. Sakuma, “Fast approximate similarity search in extremely high-dimensional data sets,” in *Proc. Int. Conf. on Data Engineering (ICDE)*, 2005, pp. 619–630.

[2] C.-W. Ngo, W. Zhao, and Y.-G. Jiang. Fast tracking of near-duplicate keyframes in broadcast domain with transitivity propagation. In ACM Multimedia, pages 845–854, 2006.

[3] E. Shechtman and M. Irani. Space-time behavior based correlation. In CVPR, pages 405–412, 2005.

[4] Fuminori Yamagishi, Shin’ichi Satoh, Takashi Hamada, and Masao Sakauchi, “Identical Video Segment Detection for Large-Scale Broadcast Video Archives,” *Proc. of International Workshop on Content-Based Multimedia Indexing (CBMI’03)*, pp. 135–142, 2003.

[5] Fuminori Yamagishi, Shin’ichi Satoh, and Masao Sakauchi, “A News Video Browser Using Identical Video Segment Detection,” *Proc. of Pacific-Rim Conference on Multimedia (PCM2004)*, pp. 205–212, Vol. II, 2004.

[6] Sheng Tang, et al., “TRECVID 2008 High-Level Feature Extraction By MCG-ICT-CAS,” *Proc. TRECVID 2008 Workshop,Gaithesburg, USA , Nov. 2008*.

[7] Aristides Gionis and Piotr Indyk and Rajeev Motwani. Similarity Search in High Dimensions via Hashing. *VLDB’99: Proceedings of the 25th International Conference on Very Large Data Bases*, 518–529, 1999.

[8] A. Joly, O. Buisson, and C. Frélicot. Content-based copy detection using distortion-based probabilistic similarity search. *IEEE Trans. on Multimedia*, 9(2):293–306, 2007.

[9] Qin Lv and William Josephson and Zhe Wang and Moses Charikar and Kai Li. Multi-probe LSH: efficient indexing for high-dimensional similarity search. *VLDB ’07: Proceedings of the 33rd international conference on Very large data bases*, 950–961, 2007.

[10] Alexis Joly. New local descriptors based on dissociated dipoles. *CIVR ’07: Proceedings of the 6th ACM international conference on Image and video retrieval*, 573–580, 2007.

[11] Sébastien Poullot and Olivier Buisson and Michel Crucianu. Z-grid-based Probabilistic Retrieval for Scaling Up Content-Based Copy Detection. *(CIVR’07: Proceedings of the ACM International Conference on Image and Video Retrieval*, 348–355, 2007.

[12] S. Poullot, M. Crucianu, and O. Buisson. Scalable mining of large video databases using copy detection. In *MM’08: Proc. 16th ACM intl. conf. on Multimedia*, pages 61–70, New York, NY, USA, 2008. ACM.

[13] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(5):530–535, 1997.

[14] J. Haitsma and T. Kalker. A Highly Robust Audio Fingerprinting System. 3rd International Conference on Music Information Retrieval, 2002.