KDDI LABS AT TRECVID 2011: CONTENT-BASED COPY DETECTION

Yusuke Uchida, Koichi Takagi, Shigeyuki Sakazawa

KDDI R&D Laboratories, Inc.
2-1-15, Ohara, Fujimino-shi, Saitama, Japan

ABSTRACT

We describe our systems for a content-based copy detection (CBCD) task submitted to TRECVID 2011. In this year, focusing on non-geometric transformations, we use only a global visual feature for efficiency. This paper, we describe a fast, accurate content-based video copy detection scheme based on bag-of-global visual features, which is characterized by (1) utilizing an efficient DCT-sign-based feature to enhance fast detection; (2) performing multiple assignment in the temporal domain in addition to the feature and spatial domain, to ensure repeatability in segment-level matching; and (3) adopting inverse document frequency weighting and temporal burstiness-aware scoring to emphasize distinctive visual words. The baseline system processes queries 60 times faster than real-time. The system integrating four baseline systems processes queries 20 times faster than real-time, and it achieves a false negative rate of 1.5% against transformations 3 and 5 without any false positives.

1. INTRODUCTION

In recent years, content-based video copy detection (CBCD) technology has attracted considerable research attention. Given a test collection of videos and a set of queries, the goal of the CBCD task in TRECVID is to determine for each query the place, if any, that some part of the query occurs, with possible transformations, in the test collection. For an automated CBCD system to be usable, it must have the following properties:

- Computationally efficient: The system must be sufficiently efficient because many video clips are uploaded to video sharing sites every day.
- Robustness: The video may have been subject to editing or degradation, including the addition of captions or patterns, a change of resolution, compression, and so on. The system should be able to detect even these altered videos robustly.
- Low false alarms: A system with too many false detections is annoying and requires ongoing operator intervention to filter out the false alarms.

This year, focusing on non-geometric transformations, we use only a global visual feature to enhance efficiency. Our system satisfies the requirements described above by (1) utilizing an efficient DCT-sign-based feature for fast detection; (2) performing multiple assignment in the temporal domain in addition to the feature and spatial domain to ensure repeatability in segment-level matching; and (3) adopting inverse document frequency weighting and temporal burstiness-aware scoring to emphasize distinctive visual words (VWs), resulting in the suppression of false positives. Furthermore, the multiple baseline systems overviewed above are combined to reduce false positives. We submitted 4 runs based on three different systems for NOFA and BALANCED profile:

- `KDDILabs.m.nofa.base`: a baseline system using a DCT-based global feature is used.
- `KDDILabs.m.nofa.2sys`: a system integrating two baseline systems is used.
- `KDDILabs.m.nofa.4sys`: a system integrating four baseline systems is used.
- `KDDILabs.m.balanced.4sys`: the same system as `KDDILabs.m.nofa.4sys` is used.

All systems are based on only a global visual feature. In this paper, all the systems are described in Section 2, and results and a discussion are presented in Section 3.

2. SYSTEM OVERVIEW

Our system is based on the bag-of-global visual features framework [1]. We advance in the following three directions:

- Utilizing the DCT-sign-based feature, which demonstrates extremely fast extraction and quantization.
- Performing multiple assignment in the temporal domain, in addition to the feature and spatial domain, which improves the tradeoff between repeatability and filtering rate in segment-level matching.
- Adopting inverse document frequency weighting and temporal burstiness-aware scoring for global features to emphasize distinctive VWs.

The system consists of the following steps: feature extraction, feature quantization, indexing using an inverted index, and searching using the voting function.

2.1. Feature extraction and multiple assignment

Multiple assignment is powerful tool to improve repeatability of VW-based feature matching by assigning multiple VWs to a single feature or keyframe [1, 2]. In this paper, multiple assignment is defined to assign multiple VWs to a single, short segment, not to a keyframe. In this section, multiple assignments in the feature, spatial, and temporal domain are introduced. First, both reference and query video clips are divided into short segments with fixed durations in the temporal domain (0.3 sec in this paper). From each of the segments, fixed number 2^{mt} of frames are subsampled at a uniform interval (multiple assignment in the temporal domain). Subsequently, these subsampled frames are divided into 2^{ms} blocks[1] (multiple assignment in the spatial domain). Finally, feature vectors are extracted from these blocks. For our system, we adopt the DCT-sign-based feature [3]; each block is resized into 8x8 pixels, and 2D-DCT is performed. Top-v AC coefficients in the zigzag scan order are used as a feature vector. If the sum of their absolute values is less than a predefined threshold, the feature vector is discarded to ignore black frames. This is because black frames are matched to each other, causing many false positives. Subsequently, they are quantized into a v-bit binary string by taking the sign of the AC coefficients. The resulting binary strings of length v define VWs with a size of $N = 2^v$. Multiple assignment in the feature domain can be performed by toggling the most unreliable mf-bits [4]. With the multiple assignment in the feature domain, each feature is assigned to 2^{mf} VWs. The reliability of each bit is defined by the absolute value of the corresponding AC coefficient. Finally, t-th reference segment is represented by $\mathcal{R}_t = (r_{t,1}, \cdots, r_{t,w}, \cdots, r_{t,W})$, where $W(= 2^{ms})$ denotes the number of blocks and $r_{t,w}$ denotes a set of VWs associated with w-th block. We also denote s-th query segment by $\mathcal{Q}_s = (q_{s,1}, \cdots, q_{s,w}, \cdots, q_{s,W})$.

2.2. Indexing and searching inverted index

For simplicity, we explain the indexing and search step only when there is a single reference video clip. This limitation is easily overcome by considering reference video identifiers or by handling many video clips as a single, long video clip. In the indexing step, for each segment of reference video, the segment and block identifiers (t, w) are stored in the vw-th list of an inverted index for all $vw \in r_{t,w}$. In the search step, segment-level matching is efficiently performed by inverted index lookups. Two segments are matched if and only if they share the same VW(s) in at least one block. The function $m(\mathcal{Q}_s, \mathcal{R}_t)$ judges whether a query segment $\mathcal{Q}_s$ is matched with a reference segment $\mathcal{R}_t$:

$$m(\mathcal{Q}_s, \mathcal{R}_t) = \begin{cases} 1 & \text{if } {}^{\exists}w \text{ s.t. } q_{s,w} \bigcap r_{t,w} \neq \emptyset \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

[1]Divied into 2x1, 2x2, 4x2, and 4x4 blocks for ms = 1, 2, 3, and 4.

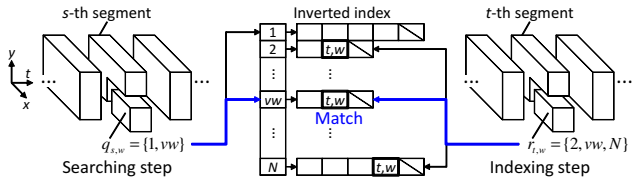


Fig. 1: Indexing and searching process using an inverted index structure.

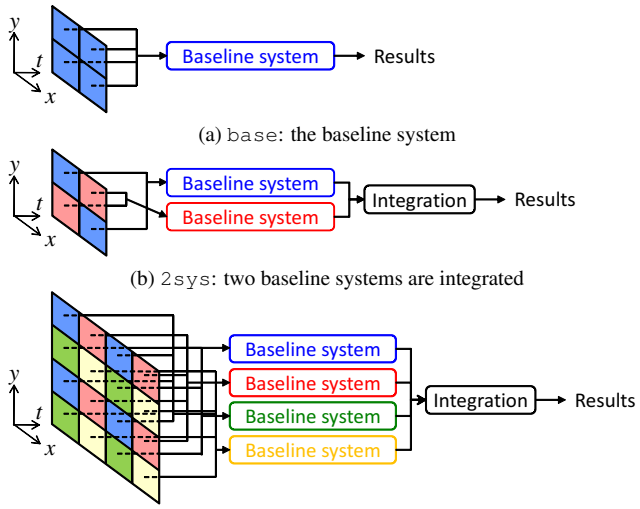
The indexing/searching process is summarized in Figure 1.

2.3. Offset-level integration

Segment-level matching results obtained by inverted index lookups are integrated into offset-level results using a voting framework [1, 5]. Every matched segment pair $(\mathcal{Q}_s, \mathcal{R}_t)$ votes for the bin $\mathrm{B}[t-s]$ corresponding to the offset $t-s$ in a 1-D Hough space. In voting, since our scheme is based on the BoVW framework, the inverse document frequency (IDF) weighting [6] can be applicable to emphasize distinctive VWs. We adopts a squared IDF [7] as a score. Though the IDF scoring has been used only for local features, it has also worked well for global features in our preliminary experiments. Performing non-maxima suppression and thresholding to the voting table after voting, we obtain a set of offset hypotheses. Each hypothesis indicates the offset between copied segments in the query and reference clips. Each offset has segment-level matching results associated with the offset represented by a set of tuples (s, vw, w). After sorting the tuples according to a query segment identifier s, they are divided into groups to localize the copied segments. A sequence of the tuples are divided if successive two tuples (s, vw, w) and (s', vw', w') satisfy $s'-s>th$. The scores of the segmented tuples are calculated by summing up the IDF weights of VWs appearing in the tuples. Temporal burstiness-aware (TBA) scoring [1] is also adopted, in which individual VWs contribute to the score only once even if a VW is shared in consecutive query and reference segments. The beginning and ending timestamps of copied segments are calculated from min and max of s in the tuples. Finally, all results are sorted according to their scores, and their scores are divided by the second best score to normalize scores among queries and transformations [8].

2.4. Integration of baseline systems

In order to get highly reliable results, especially to suppress false positives, we utilize a consensus of multiple systems. In order to improve detection performance, a common approach is to integrate results from systems based on different modalities (e.g., global, local, and audio [9]). In contrast, for our system, we combine the same systems: each input query is divided in the spatial domain into non-overlapping segments,



(a) `base`: the baseline system

(b) `2sys`: two baseline systems are integrated

(c) `4sys`: four baseline systems are integrated

Fig. 2: Three different systems used in our submission.

and they are used as subqueries of baseline systems. In `2sys`, each query is divided into 2x2 regions, and two of the four regions are given to a baseline system, and the other two regions are given to the other baseline system. In `4sys`, each query is divided into 4x4 regions, and four different regions are used as a query for each of the baseline systems. The three different systems used in our submission are summarized in Figure 2. Each of the baseline systems outputs the top 20 results for each query, and these results are integrated using a simple approach: two results from different baseline systems indicating overlapping segments of the same reference video clip are integrated. The integrated score is the sum of the two results to be integrated. The timestamps of the integrated result are averages of corresponding timestamps of the two results to be integrated. After integration, all results are sorted again, and only the result having the top score is returned as a final result. While the threshold for the baseline system is determined using the dataset used in the TRECVID 2009 workshop, the threshold for `4sys` is set to 3.9, a value which is not exceeded even when four results with scores less than 1.0 are integrated. A score of 1.0 corresponds to the second best score of the result from each baseline system.

3. RESULTS AND ANALYSIS

In this section, the results of our preliminary experiments and our submitted runs are shown. 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) measures the tradeoff between the cost of false negatives and false positives, and is defined by the weighted mean of two errors.

Table 1: NDCR scores for the `base` system and the previous year's system. The dataset used in the TRECVID 2009 workshop is used for the evaluation.

	T2	T3	T4	T5	T6	T8	T10
`base`	1.000	**0.007**	**0.000**	**0.000**	**0.000**	0.843	0.821
`[1]-1`	**0.134**	0.067	0.045	0.082	0.433	0.567	0.470
`[1]-2`	0.239	**0.007**	0.060	0.022	0.022	**0.231**	**0.269**

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

Our experiments were performed on a machine with a Core i7 970 CPU and 24 GB of main memory.

3.1. Comparison with previous year's systems

Table 1 shows the results of a preliminary experiment using the TRECVID 2009 dataset, which is the training dataset for this year's submission. In the table, `base` represents our baseline system with multiple assignment defined by the parameter $(0, 2, 3)$, while `[1]-1` and `[1]-2` represent a system based on global features and a system based on both local and global features as described in [1], respectively. The latter two systems were submitted to the TRECVID 2010 workshop. This year's baseline system achieves an NDCR score of 0.002 on average in transformations T3 to T6, which corresponds to a false negative rate of 0.2% (one false negative against 536 positive examples) without any false positives.

3.2. Results of submitted runs

Figure 6 shows the evaluation results of our run `KDDILabs.m.nofa.base` for the NOFA profile. We can see that the baseline system achieves reasonable accuracy against transformations 3 to 5, and attains almost perfect results in localization accuracy. The detection speed of the baseline system is extremely fast. It processes each query in 1.45 seconds on average (60 times faster than real-time), which includes a decoding time of 0.42 seconds. Figure 7 shows the evaluation results of our run `KDDILabs.m.nofa.2sys` for the NOFA profile. The accuracy with optimal thresholds is improved compared with the baseline system. This might be because each baseline system can use only two blocks in a frame in matching. Figure 8 shows the evaluation results of our run `KDDILabs.m.nofa.3sys` for the NOFA profile. The system used for the run `KDDILabs.m.nofa.3sys` achieves excellent results in transformations 3 to 5 even with the actual threshold. This is because the scores greatly depend on the consensus of multiple systems, suppressing accidental detections. The system requires 4.25 seconds to process each

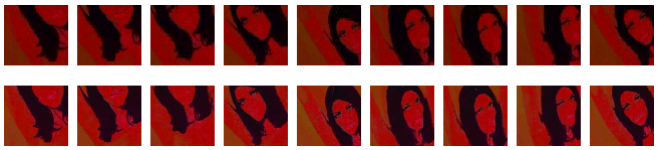


Fig. 4: Thumbnails of the query geometrically transformed and the corresponding groundtruth video clip.

Fig. 5: Thumbnails of the query corresponding to the last false negative and the groundtruth video clip.

query on average. It is still 20 times faster than real-time. Figure 3 shows processing time and actual NDCR values of all submitted runs for the NOFA profile. It can be seen that our systems have achieved good tradeoffs between processing time and accuracy in transformations 3 to 6.

3.3. Error analysis

In the run `4sys`, there are four false negatives indicated for each of the transformations 3 and 5 queries. For two of them, we returned the identifiers of duplicated videos that include the same content as the groundtruth videos. This is because our system returns at most one result for each query. If we count these two false negatives as true positives, the number of actual false negatives becomes 2 against transformations 3 and 5. This corresponds to a false negative rate of 1.5%. We found that a query corresponding to one of the remaining two false negatives is geometrically transformed, although by definition transformations 3 and 5 do not include any geometric transformations. Figure 4 shows thumbnails of the query in question and the corresponding groundtruth video clip. Figure 5 shows thumbnails of the query corresponding to the last false negative and the groundtruth video clip. We can see that the copied segment consists of almost black frames. Our system may have missed the query because it drops black frames in feature extraction.

4. CONCLUSION

In this paper, we introduced our CBCD system submitted to the TRECVID 2011 workshop. The baseline system processed queries 60 times faster than real-time. The system integrating four baseline systems processed queries 20 times faster than real-time, and it achieved a false negative rate of 1.5% against transformations 3 and 5 without any false positives. As the proposed system is extremely lightweight, it can be efficiently combined with other systems, such as local feature-based or audio feature-based systems, which are complementary to global feature-based systems.

5. REFERENCES

[1] Y. Uchida, M. Agrawal, and S. Sakazawa, "Accurate content-based video copy detection with efficient feature indexing," in *Proc. of ICMR*, 2011.

[2] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *Proc. of CVPR*, 2008, pp. 1–8.

[3] J. Barr, B. Bradley, and B. T. Hannigan, "Using digital watermarks with image signatures to mitigate the threat of the copy attack," in *Proc. of ICASSP*, 2003, pp. 69–72.

[4] J. Haitsma and T. Kalker, "A highly robust audio fingerprinting system," in *Proc. of ISMIR*, 2002, pp. 107–115.

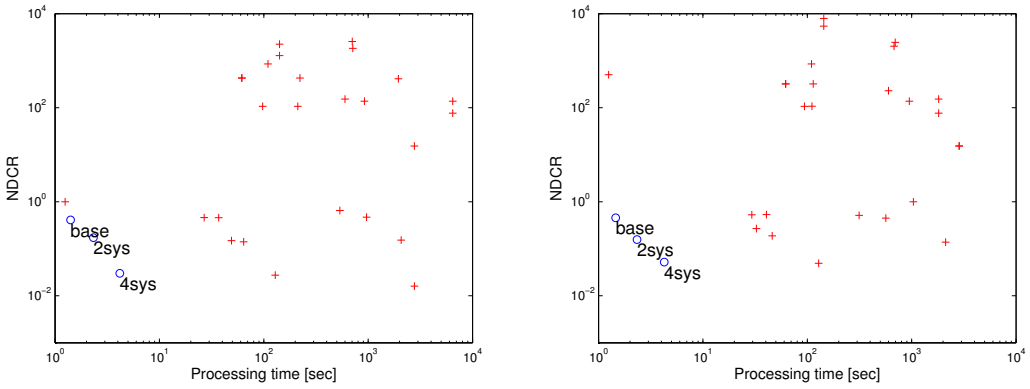
[5] J. Law-To, L. Chen, A. Joly, and I. Laptev, "Video copy detection: a comparative study," in *Proc. of CIVR*, 2007, pp. 371–378.

[6] J. Sivic and A. Zissermane, "Video google: A text retrieval approach to object matching in videos," in *Proc. of ICCV*, 2003, pp. 1470–1477.

[7] H. Jégou, M. Douze, and C. Schmid, "Improving bag-of-features for large scale image search," *IJCV*, vol. 87, no. 3, pp. 316–336, 2010.

[8] Z. Liu, T. Liu, and B. Shahraray, "AT&T research at TRECVID 2009 content-based copy detection," in *Proc. of TRECVID*, 2009.

[9] Y. Uchida, M. Agrawal, M. Akbacak, and S. Sakazawa, "KDDI labs and SRI International at TRECVID 2010: Content-based copy detection," in *Proc. of TRECVID*, 2010.



(a) T3 (b) T4

(c) T5 (d) T6

Fig. 3: Processing time and actual NDCR values of all submitted runs for the NOFA profile. Our runs are represented by 'o' and all the other runs are represented by '+'. From upper right to lower left, the tradeoff between processing time and accuracy improves.

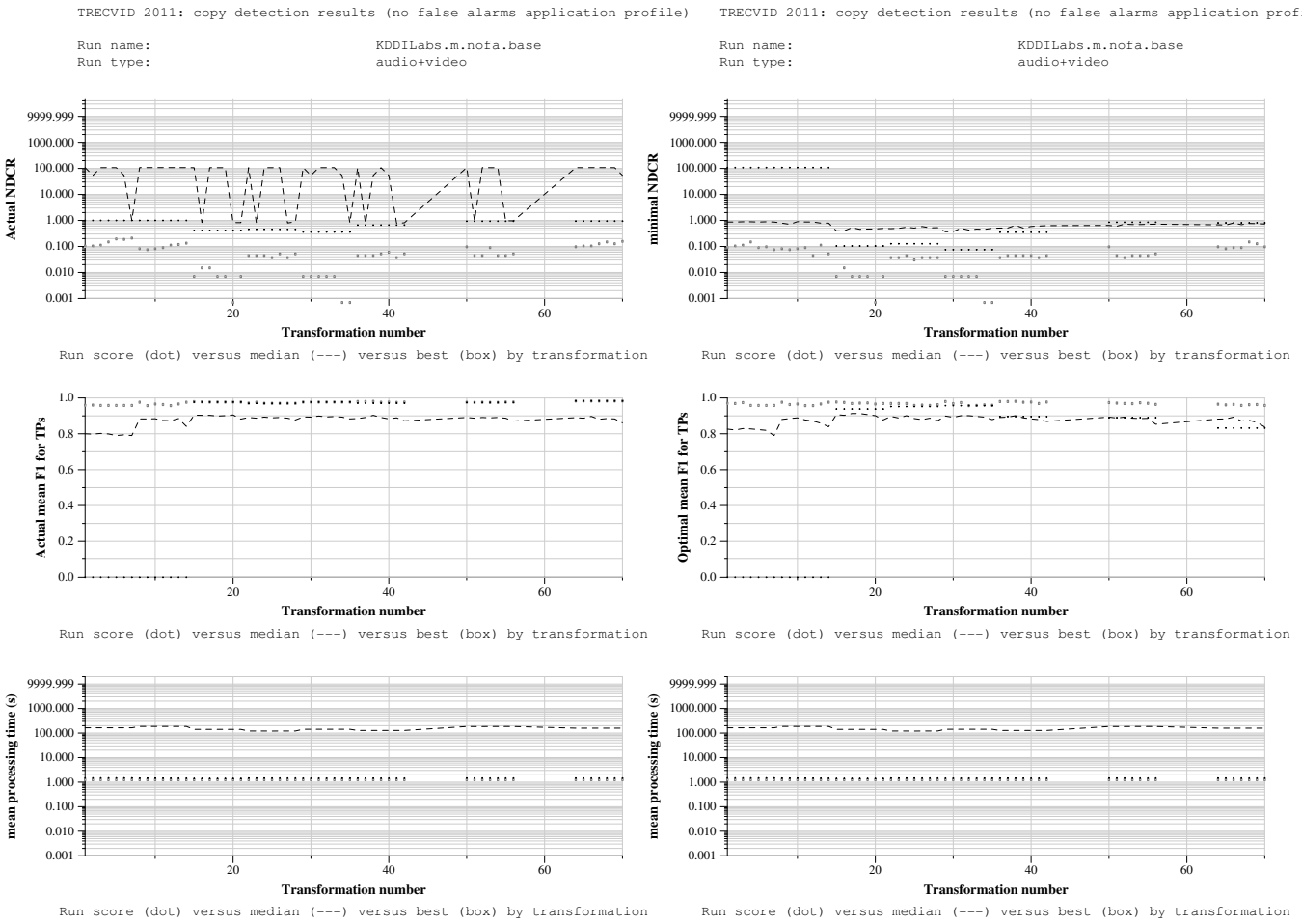


Fig. 6: The actual (left) and optimal (right) results of the `base` run.

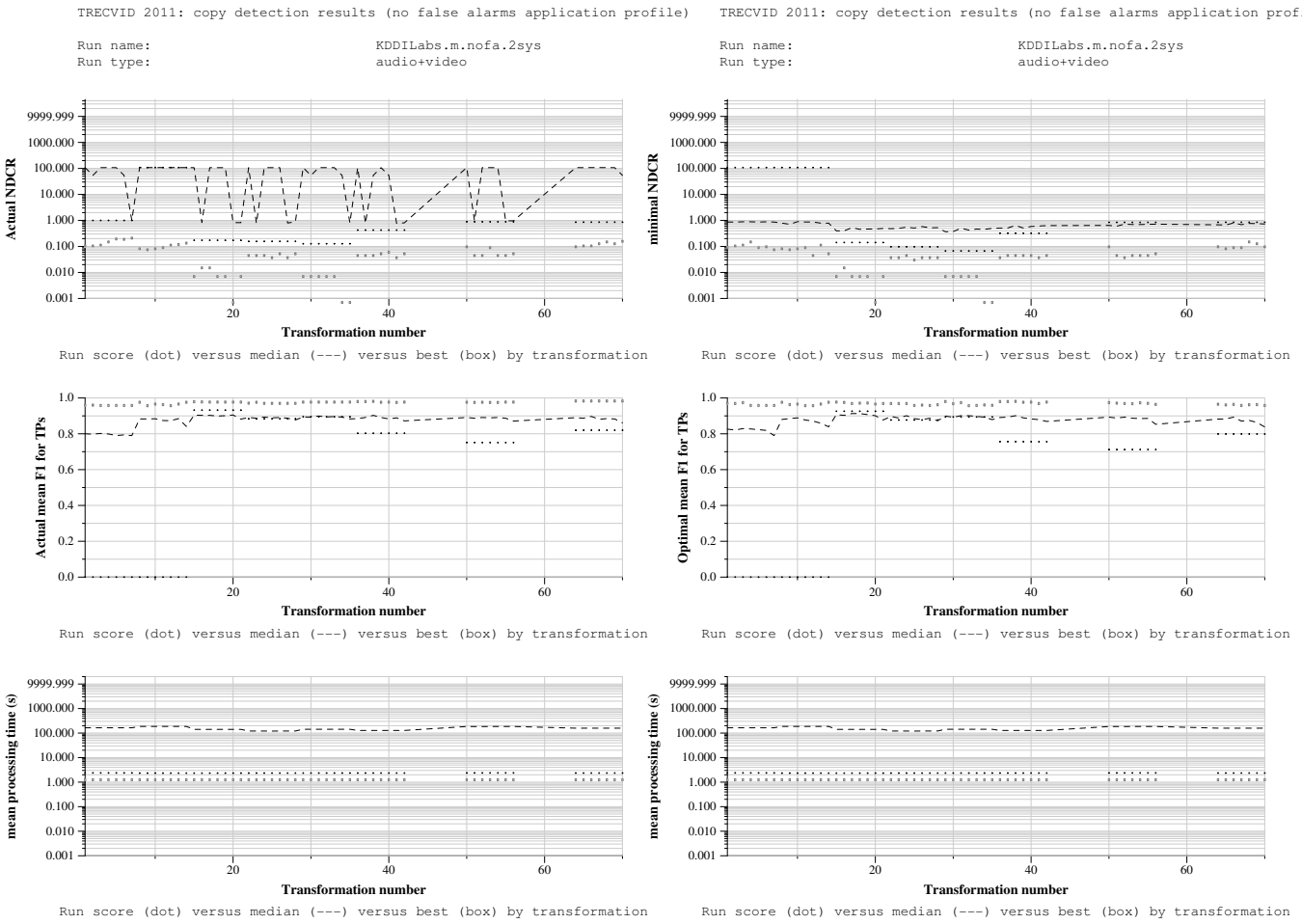


Fig. 7: The actual (left) and optimal (right) results of the `2sys` run.

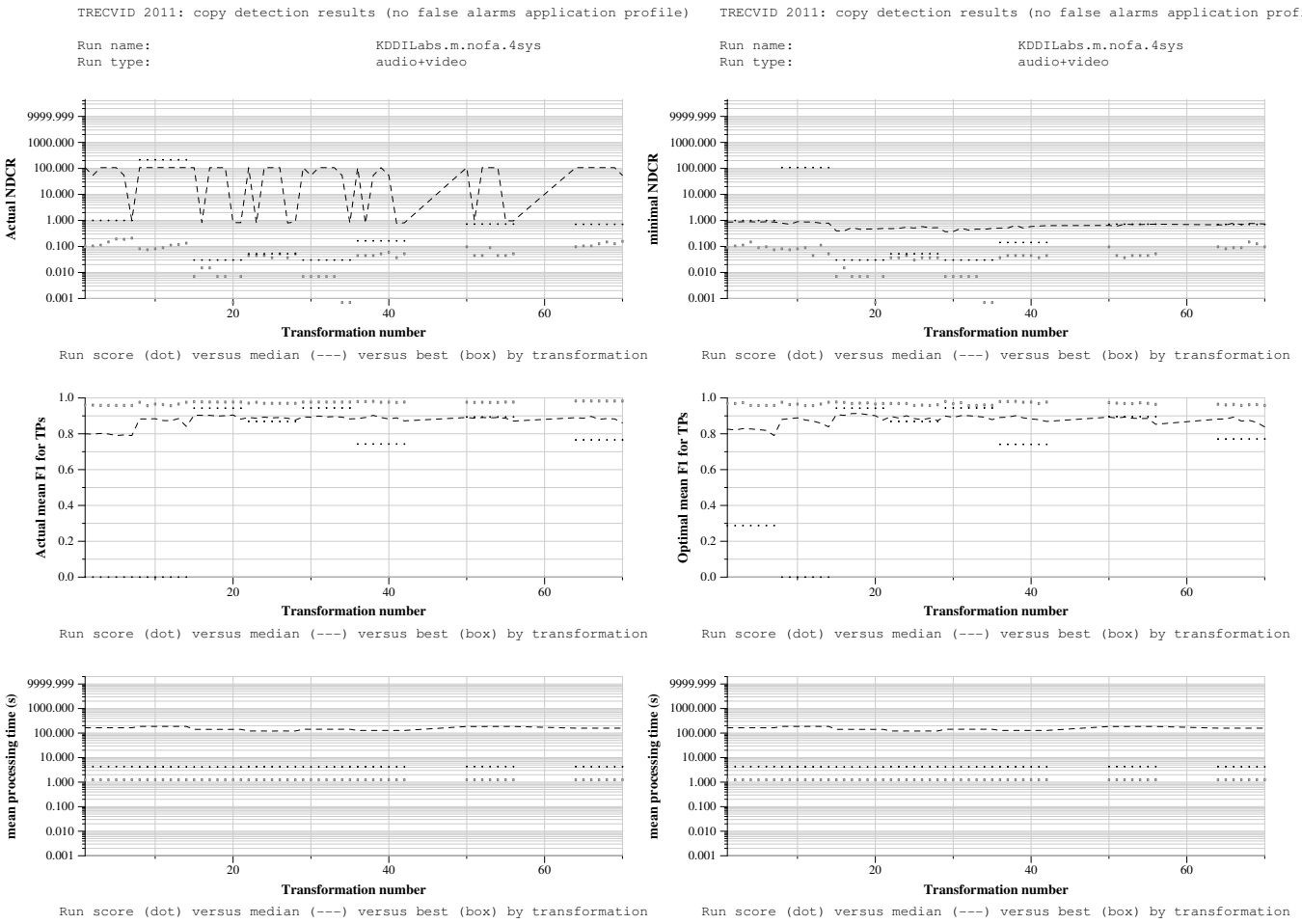


Fig. 8: The actual (left) and optimal (right) results of the `4sys` run.